Pour une description complète, formelle et calculatoire des circuits d'optique linéaire quantique

Nicolas Heurtel

▶ **To cite this version:**

Nicolas Heurtel. Pour une description complète, formelle et calculatoire des circuits d'optique linéaire quantique. Théorie et langage formel [cs.FL]. Université Paris-Saclay, 2025. Français. NNT : 2025UP-ASG035 . tel-05245900

HAL Id: tel-05245900

https://theses.hal.science/tel-05245900v1

Submitted on 9 Sep 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

université PARIS-SACLAY

Towards a Complete Formal and Computational Description of Linear Optical Quantum Circuits

Pour une description complète, formelle et calculatoire des circuits d'optique linéaire quantique

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580, Sciences et Technologies de l'Information et de la Communication (STIC) Spécialité de doctorat : informatique
Graduate School : Informatique et Sciences du Numérique
Référent : Université de Versailles-Saint-Quentin-en-Yvelines

Thèse préparée dans l'unité de recherche **Laboratoire Méthodes Formelles** (Université Paris-Saclay, CNRS, ENS Paris-Saclay, Inria), sous la direction de **Pablo ARRIGHI**, directeur de recherche INRIA, et le co-encadrement de **Benoît VALIRON**, professeur CentraleSupélec, et de **Shane MANSFIELD**, Chief Research Officer de Quandela.

Thèse soutenue à Gif-sur-Yvette, le 11 juin 2025, par

Nicolas HEURTEL

Composition du jury

Membres du jury avec voix délibérative

Eleni DIAMANTI Directrice de Recherche, CNRS, Sorbonne Université	Présidente
Ernesto GALVÃO Directeur de Recherche, International Iberian Nanotechnology Lab	Rapporteur & Examinateur
Emmanuel JEANDEL Professeur, Université de Lorraine	Rapporteur & Examinateur
Miriam BACKENS ISFP, INRIA, Université de Lorraine	Examinateur·rice
Ulysse CHABAUD Chargé de Recherche, INRIA, École Normale Supérieure de Paris	Examinateur

THÈSE DE DOCTORAT
NNT : 2025UPASG035

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Pour une description complète, formelle et calculatoire des circuits d'optique linéaire quantique
Mots clés : Optique linéaire quantique – Languages graphiques – Simulation – Calcul quantique – Théories équationnelles complètes

Résumé : Les ordinateurs quantiques exploitent les principes de la physique quantique pour encoder et traiter l'information, avec le potentiel de surpasser les capacités des ordinateurs classiques dans certains domaines. Cette thèse explore l'une des technologies majeures du calcul quantique : l'optique linéaire, où les calculs sont effectués à l'aide de photons traversant des circuits d'optique linéaire.

Si ces circuits sont bien compris et simulables à petite échelle, leur véritable intérêt apparaît à grande échelle lorsqu'ils atteignent des régimes hors de portée du calcul classique. Cette thèse s'intéresse à cette transition sous deux aspects : sur le plan calculatoire d'une part, en développant des modèles plus efficaces pour repousser les limites des simulations classiques, et sur le plan formel d'autre part, en modélisant à un niveau plus abstrait les circuits optiques et leurs interactions.

En simulation, l'un des défis majeurs est le calcul efficace des probabilités de sortie des circuits. Nous présentons dans cette thèse de nouveaux algorithmes pour calculer plusieurs de ces probabilités, où l'utilisation de la mémoire est optimisée pour significativement réduire le temps de calcul. Ce travail mène au développement de Perceval, une plateforme logicielle open-source dédiée au calcul quantique photonique.

Sur le plan théorique, nous introduisons un cadre formel et diagrammatique qui capture pleinement la structure des processus d'optique linéaire. Plus précisément, nous utilisons le formalisme des langages graphiques pour modéliser ces circuits. Nous établissons une théorie équationnelle complète, démontrant que deux circuits équivalents d'optique linéaire peuvent toujours être réécrits localement avec notre ensemble d'équations.

Enfin, nous montrons comment nos contributions ont des applications à la fois en physique et en informatique quantique théorique. En particulier, nous étendons notre théorie équationnelle complète des circuits optiques à une théorie équationnelle complète pour les circuits quantiques, résolvant ainsi un problème ouvert des dernières décennies.

Title: Towards a complete formal and computational description of linear optical quantum circuits
Keywords: Quantum linear optics – Graphical languages – Simulation – Quantum computing – Complete equational theory

Abstract: Quantum computers leverage the principles of quantum physics to encode and process information, offering promising applications that may overcome the capabilities of classical computers. This thesis explores one of the leading technologies for quantum computation: linear optics, where computations are performed with photons passing through linear optical circuits.

While linear optical circuits are well understood and computationally tractable when they are small, their real significance emerges at larger scales, where they transition into regimes beyond classical computational reach. This thesis addresses both aspects: first, by developing a more detailed computational description to push the limits of classical simulations; and second, by establishing a formal framework to analyze these systems from a higher-level perspective.

On the simulation side, a key challenge is computing output probabilities efficiently. We develop new algorithms to compute several of these probabilities, where memory usage is optimized to significantly reduce computation time. This work leads to the development of Perceval, an open-source software platform specifically designed for photonic quantum computing.

On the theoretical side, we construct a formal and diagrammatic framework that fully captures the structure of linear optical processes. Specifically, we introduce graphical languages, widely used in quantum computing, to model linear optical circuits. We establish a complete equational theory, proving that any two equivalent linear optical circuits can always be rewritten locally using our set of equations.

Finally, we show how our contributions provide tools to both physics and theoretical quantum computing problems. In particular, we extend our complete equational theory for linear optical circuits to derive a complete equational theory for general quantum circuits, resolving a long-standing open problem.

Remerciements

"I really mean it. Merci." — NH.

Voilà. C'est fait. Enfin, presque. La thèse n'aurait pas été possible sans de nombreuses personnes, personnes qui m'ont aidé, accompagné, inspiré... Elle ne peut donc pas être aboutie sans les remercier.

Honnêtement, il m'est très difficile d'écrire ces acknowledgements. Comment pourrais-je vraiment retranscrire toute la reconnaissance que j'ai, sur l'entièreté d'une aventure telle qu'une thèse ? I won't be able to express how thankful I am to all the people that have been directly or indirectly part of this journey. Still, they need to be mentioned. Les noms cités et l'ordre sont donc arbitraires, et soumis aux contraintes de l'exercice. I do mean it when I write I'm deeply thankful and grateful to all of you. Merci.

First, I would like to thank Eleni, Ernesto, Emmanuel, Miriam and Ulysse, for being part of the jury. Especially thank you to the reviewers, who have deeply examined the manuscript. If there are any typos/mistakes left, it is fully on me.

Benoît, Merci d'avoir cru en moi. Merci d'avoir été présent dans les moments à célébrer, dans les moments difficiles, d'avoir poussé ce *young naive student* à prendre son envol. Merci pour cette profonde bienveillance ; tu es un modèle de professeur à suivre. Une grande partie de cette aventure a été possible grâce à toi.

Shane, thanks for all your unconditional support in Quandela since the internship. I really enjoyed discovering linear optics with you. Thank you for the trust, freedom and healthy working environment. You gave me far more than scientific supervision, and this journey has been the right choice thanks to you.

Quandela, merci de m'avoir accueilli et accompagné tout au long du parcours. Merci à l'équipe théorie pour avoir été si chaleureuse et stimulante. Merci à toi PE, pour tous ces échanges et retours. Merci Boris, pour cet accompagnement dans cette dernière ligne droite. Merci Eric et Marion, pour tous les moments partagés et les CNot dans la cuisine. Merci Fatima, Maxence et Michel pour tout le soutien depuis mes débuts en entreprise.

Big thank you to all my sushi friends. Thank you Rawad, for all the kind teasing and scientific chats ; most of my future slack is dedicated to you. Merci Tim, pour avoir rendu ces semaines et mois de travail sur LO-SLAP beaucoup plus amusantes et stimulantes. Merci Grégoire pour cette amitié, mais aussi pour la passion et l'énergie que tu transmets au quotidien. Andreas, merci d'avoir tous ces titres d'amis, collaborateur, organisateur, etc. De toutes ces années, on ne retiendra pas que le Jacobien.

Thank you, QuaCS, for providing me such a joyful and kind academic environnement. Thank you Cole, Mathieu, Marin, Octave and Vladimir for all the support, chats and beers. Merci Pablo, pour la supervision et ces échanges sur la vulgarisation. Merci Marc, pour tous ces retours précieux et toujours pertinents. Alexandre, merci d'avoir été là. J'ai beaucoup appris à tes côtés ; I'm glad you came to

QuaCS. Merci Renaud, pour ces moments précieux en conférence, et les nombreuses heures au tableau à échanger. Je ne pourrai plus t'embêter dans ton bureau, mais je n'abandonne pas la fameuse future collaboration ! Merci Louis, pour avoir été mon voisin, mon oreille, et de m'avoir permis de continuer notre aventure à Èdinbourg.

Merci à mes co-auteur·rice·s. Nadia, merci pour ces échanges et cette bienveillance. Merci Simon, pour tous ces conseils et discussions. Merci Robert, pour tous ces moments en conférence, et d'avoir été là pour le research proposal. Jean, merci d'avoir été le père de Perceval, d'avoir cette flamme si inspirante, et d'avoir cru en le potentiel de SLOS.

Merci Elias et Massilya pour votre soutien. Merci Christina pour cette inspiration et les encouragements. Merci aux « étudiants du café », Camille, Jade, Julien, Kevin pour tous ces moments. Arthur, merci d'avoir été présent de si nombreuses fois.

Merci Antoine, Camille, Donatien, Jeremy et Nathalie pour toutes les histoires et souvenirs inoubliables.

Thank you Emily for being a bright fellow doctor. Merci Salomé, pour toutes ces discussions et pour le soutien mutuel.

Merci Alexandre pour toutes ces années d'amitié. Merci Céline, pour tous ces appels transatlantiques. On réussira nos projets !

Merci aux *copains*, Claire, Samuel, Vanessa et Victor. Je chéris tous les escapes, soirées jeux, sorties et voyages que nous avons pu faire. Merci Théo, pour nos échanges toujours sincères. Merci Anne-So, pour toutes ces aventures temporelles. Je vous attends tous·tes à Edinburgh !

À mes amis d'enfance Quentin, Mathias. Merci d'avoir été présent la veille de la soutenance. Quentin, merci d'être cette épaule infaillible. J'espère qu'on continuera d'évoluer ensemble. On a encore beaucoup de voyages ; on retournera au Japon en tant que docteurs !

À mes anciens colocs, à qui je dois beaucoup. Taha, merci pour toutes ces magic aventures. Merci Badr, pour toutes ces années, ces fous rires, et ces courtes nuits. Merci Julie, pour être si forte et inspirante. Toutes ces journées à la BU et ta présence à la soutenance ont été si importantes. El Mehdi, aka Doctor H ; merci pour cette 85 coloc. C'était un îlot de sérénité irremplacable, rempli de 3 ans de souvenirs. I'm glad to have been your partner in crime.

Merci à toute ma famille ; mes grands-parents, mes oncles, mes tantes, ma marraine et mes cousins, pour tout le soutien et l'amour à mon égard. Merci Marine, pour avoir été une grande soeur qui a veillé sur moi. Merci à mes parents, qui m'ont donné cette liberté et qui ont toujours été présents dans les moments importants. Le petit Nico grandit et vous remercie.

À ma grand-mère, dont le souvenir est gravé ici.

Contents

Introduction **15**

I Background **21**
I.1 Mathematical definitions and notation . . . 21
I.1.1 Hilbert space . . . 21
I.1.2 Operators and their properties . . . 22
I.1.3 Dirac notation . . . 22
I.2 Quantum mechanics . . . 23
I.2.1 Postulates of quantum mechanics . . . 23
I.2.2 Heisenberg picture . . . 26
I.2.3 Second quantization: Fock space . . . 26
I.2.4 Number, annihilation and creation operators . . . 27
I.2.5 Quantum computing and quantum circuits . . . 28
I.3 Linear optics . . . 30
I.3.1 Quantum model of light . . . 30
I.3.2 Linear optical evolution . . . 34
I.3.3 Linear optical circuits: basic components . . . 36
I.3.4 Universality of spatial-mode linear optics . . . 40
I.3.5 Simulation of discrete-variable quantum linear optics . . . 41
I.3.6 Computation and approximation of the matrix permanent . . . 45
I.4 Graphical language . . . 47
I.4.1 Towards a formalization of diagrams . . . 47
I.4.2 String diagrams . . . 48
I.4.3 Category theory: straight to the PROP . . . 50
I.4.4 Manipulating diagrams with rewriting theory and logic . . . 53

II Simulation of linear optical circuits **57**
II.1 Motivations and notation . . . 58
II.1.1 Simulation problems and use cases . . . 58
II.1.2 Definitions and notation . . . 60
II.2 SLOS: Strong linear optical simulator . . . 61
II.2.1 Algorithm . . . 62
II.2.2 Complexity analysis . . . 66
II.2.3 Implementation . . . 70
II.2.4 Benchmark . . . 73

II.2.5 Extension to feedforward and hybrid simulation . . . 75
II.3 LO-SLAP: Linear optical simulator through lattice of polynomials . . . 80
II.3.1 Algorithm . . . 80
II.3.2 Traversal time optimization with Steiner trees . . . 83
II.3.3 Complexity Analysis . . . 86
II.3.4 Implementation . . . 88
II.3.5 Benchmark with SLOS: memory trade-off and feedforward simulation . . . 91
II.3.6 Extension to noisy simulation . . . 95
II.4 Boson sampling for generic input and with feedforward . . . 96
II.4.1 Gate-by-gate simulation . . . 97
II.4.2 SBS: Superposition Boson Sampling . . . 100
II.4.3 Generalized Ryser's formula . . . 103
II.4.4 EBS: Entangled Boson Sampling . . . 105

III Graphical language for linear optical circuits **111**
III.1 $\mathbf{LO_{PP}}$: Linear optical polarizing-preserving circuit . . . 112
III.1.1 Syntax and single-photon semantics . . . 112
III.1.2 Minimal complete equational theory of $\mathbf{LO_{PP}}$ . . . 114
III.1.3 Universality and triangular $\mathbf{LO_{PP}}$-circuits . . . 119
III.1.4 Rewriting system . . . 123
III.1.5 Completeness of $\mathbf{LO_{PP}}$ . . . 127
III.2 $\mathbf{LO_{v}}$: Linear Optical circuits with vacuum sources and postselection . . . 132
III.2.1 Syntax and semantics . . . 133
III.2.2 Complete equational theory of $\mathbf{LO_{v}}$ . . . 135
III.2.3 From polarization to spatial modes . . . 137
III.2.4 Completeness of $\mathbf{LO_{v}}$ . . . 139
III.2.5 Compact universal forms of $\mathbf{LO_{v}}$-circuits . . . 142
III.3 $\mathbf{LO_{fi}}$: Linear optical circuit with finite-number-photon sources and detectors . . . 146
III.3.1 Syntax and many-photon semantics . . . 147
III.3.2 Complete equational theory of $\mathbf{LO_{fi}}$ . . . 149
III.3.3 Strongly normalizing rewriting system . . . 151
III.3.4 Unique normal forms . . . 157
III.3.5 Conclusion . . . 167

IV Applications of the theoretical contributions **169**
IV.1 Completeness of quantum circuits . . . 169
IV.1.1 Context and scope of this thesis . . . 170
IV.1.2 Quantum circuits: syntax, semantics and multi-controlled gates . . . 171
IV.1.3 First complete equational theory . . . 175
IV.1.4 Completeness: Sketch of the proof . . . 180
IV.2 Perceval: Software platform for discrete-variable linear optical quantum computing . . . 186
IV.2.1 Overview . . . 187
IV.2.2 Main Classes . . . 188
IV.2.3 Walkthrough tutorial with Perceval . . . 191
IV.3 Mitigating the crosstalk of linear optical circuits . . . 212
IV.3.1 Crosstalk: definition and model . . . 213
IV.3.2 Extended crosstalk model: induced phases . . . 214
IV.3.3 Reducing the crosstalk matrix with invariant transformations . . . 215

IV.3.4 Finding invariant phase transformations 217
IV.3.5 Graph-theory criteria for crosstalk robustness: acyclic graph and circuit rank 218

Conclusion and outlook **223**

Résumé en français

L'informatique est une pierre angulaire de la science et de la technologie modernes ; elle nous permet de traiter des données, de résoudre des problèmes mathématiques et de simuler des systèmes complexes. En informatique, le calcul consiste à encoder l'information dans des systèmes physiques et à les manipuler afin d'effectuer des opérations telles que des opérations arithmétiques, logiques ou de stockage et de récupération de données. Depuis 1945, [1] les ordinateurs sont principalement implémentés à l'aide de circuits électroniques, dans lesquels l'information est encodée par des niveaux de tension électrique et traitée par des composants tels que les transistors [PH13]. Ces systèmes suivent généralement un modèle de calcul fondé sur des opérations logiques appliquées à des bits, des chiffres binaires ne prenant que deux valeurs, habituellement représentées par 0 et 1. Ce cadre, appelé calcul classique, constitue la base des algorithmes et des architectures informatiques actuels.

Bien que particulièrement performants, les ordinateurs classiques se heurtent à des limitations fondamentales. Certains problèmes, comme la simulation de réactions chimiques clés, la résolution d'équations différentielles ou la factorisation d'entiers en nombres premiers, semblent nécessiter des milliards d'années avec des algorithmes classiques. Cela suggère de rechercher de nouvelles façons de traiter l'information et plus efficaces pour ces problèmes. Et si, à la place de composants électroniques régis par la mécanique classique, nous utilisions des systèmes physiques obéissant aux lois de la mécanique quantique pour encoder et manipuler l'information ?

Cette question a conduit à un nouveau paradigme; l'informatique quantique, conçue pour dépasser les limitations des ordinateurs classiques. Cette approche exploite les principes de la mécanique quantique, telles que la superposition, l'intrication et la mesure, pour réaliser des opérations de manière fondamentalement différente [Fey82; DP85; Llo96; NC10]. Par exemple, au lieu des deux états d'un bit, les ordinateurs quantiques utilisent des qubits, systèmes quantiques pouvant se trouver dans une superposition de deux états. Cela a débouché sur l'élaboration d'un cadre formel pour le calcul quantique [DP85], jetant les bases de l'exécution d'algorithmes quantiques sur un ordinateur quantique généraliste. Depuis lors, le domaine des algorithmes quantiques a évolué rapidement, proposant des accélérations allant du quadratique [Gro96] à l'exponentiel [Sho94; HHL09], ainsi que des heuristiques prometteuses et des applications en chimie, optimisation et apprentissage automatique [Lan+10; Per+14; FGG14; Bia+17; AL18; Pre18; Cao+19]. [2] Ces avancées ont conduit à un développement fort et remarquable de l'informatique quantique ces dernières années [Pre23].

Alors que les algorithmes quantiques présentent des avantages théoriques bien définis et prometteurs,

1. Le premier ordinateur électronique numérique a été développé en 1945 par John P. Eckert et John W. Mauchly. Il est connu sous le nom d'ENIAC (Electronic Numerical Integrator and Computer) [GG46; Har46].

2. Il convient de noter que la plupart de ces gains sont des conjectures. Par exemple, le meilleur algorithme classique connu pour la factorisation prime est sous-exponentiel [LL93] en fonction du nombre de bits de l'entier. Bien que son pendant quantique s'exécute en temps polynomial [Sho94], rien ne prouve qu'un algorithme classique plus efficace n'existe pas.

ils doivent encore être exécutés sur des dispositifs physiques pour devenir réellement utiles. Cela nécessite des systèmes quantiques bénéficiant d'un contrôle suffisant et capable de passer à l'échelle [DiV00]. Diverses plateformes ont ainsi été explorées pour réaliser des ordinateurs quantiques [NC10], chacune offrant des approches différentes. Cette diversité a mené à une compétition active entre plusieurs technologies, notamment les systèmes supraconducteurs [NPT99; Aru+19; Kja+20; Bra+24; Ach+24], les ions piégés [CZ95; Bru+19; Pae+24] et les atomes ultrafroids [Jak+99; Saf16; GB17].

L'un des candidats les plus prometteurs pour l'ordinateur quantique est l'optique linéaire à variables discrètes [Kok+07; OFV09; Rud17], qui utilise les photons, particules de lumière, pour encoder l'information quantique. Indépendamment de sa puissance de calcul, cette technologie se distingue pour transmettre l'information quantique, avec des applications en communication sécurisée [BB84; BB14] et en réseaux quantiques [Kim08; WEH18]. Pour le calcul quantique, [KLM01] a montré que des circuits d'optique linéaire avec adaptivité, l'utilisation de résultats intermédiaires pour adapter dynamiquement les opérations ultérieures du circuit, suffisent pour exécuter efficacement tout algorithme quantique. Cette découverte a ouvert la voie à de nombreuses études et améliorés exploitant l'adaptivité [YR03; Nie04; BR05; Bar+23; Gli+24].

Défis

Il existe plusieurs défis pour construire des dispositifs quantiques photoniques fiables. Comme tout système quantique physique, les circuits d'optique linéaire sont très sensibles aux erreurs. Cette exigence de précision implique une caractérisation poussée, visant à identifier le comportement et les paramètres spécifiques du système. L'objectif est de comparer les sorties réelles à celles prévues par un modèle théorique, souvent issu de simulations, afin de corriger les imperfections. Il existe donc un besoin crucial d'algorithmes efficaces capables de gérer divers types d'erreurs, d'entrées et de configurations expérimentales.

Malgré ce besoin, aucun algorithme générique et efficace n'a été développé pour répondre aux exigences à court terme de la caractérisation des circuits optiques. En effet, les probabilités de sortie de l'optique linéaire sont liées au calcul des permanents de matrices, un problème difficile [Val79], où même la tâche d'échantillonnage est difficile pour les ordinateurs classiques [AA11; Bro+19]. Alors que le calcul d'une seule probabilité de sortie a été bien étudié et optimisé [Sch04; GIQ19; Shc20], aucun travail antérieur n'a traité de manière exhaustive le calcul efficace de plusieurs probabilités simultanément. De plus, bien que l'adaptivité soit essentielle pour les architectures photoniques robustes aux erreurs [KLM01; Bar+23; Gli+24], aucun algorithme connu de simulation ne prend en charge efficacement ce cadre.

Si la simulation numérique peut aider pour les expériences à petite échelle, elle devient très vite inadaptée pour aborder des systèmes de plus grande taille. Pour concevoir, tester et vérifier des architectures photoniques à grande échelle, il est indispensable de dépasser les méthodes purement numériques et de développer des modèles abstraits, composables et à un plus haut niveau d'abstraction.

Parmi les cadres formels dédiés à l'étude des systèmes quantiques, les langages graphiques, où les systèmes sont représentés par des diagrammes et des fils pouvant être à la fois interprétés visuellement et manipulés formellement, ont récemment gagné en popularité. Cette approche de haut niveau a conduit à de nombreux progrès pour les systèmes constitués de qubit. L'exemple le plus marquant est celui des diagrammes ZX [CD08; CD11; HNW18; JPV18], avec des applications en compilation [KW20; Bac+21; Wet+24], simulation [KWV22; KW22; KYW23] et vérification [DL14; GD18]. Cependant, ces techniques ont été développées exclusivement pour les systèmes basés sur les qubits et ne peuvent pas être appliquées directement au cadre photonique.

Contributions et plan de cette thèse

Dans cette thèse, nous abordons à la fois les défis à petite et à grande échelle. Pour la plus basse échelle, nous proposons un large éventail d'algorithmes efficaces, destinés à la fois à l'échantillonnage et au calcul efficace d'un grand nombre de probabilités de sortie dans les circuits d'optique linéaire avec adaptivité et divers types d'états d'entrée (Chapter II). Pour la plus grande èchele, nous introduisons des languages graphiques adaptés aux circuits d'optique linéaire, englobant tous les principaux composants optiques et révélant de nouvelles propriétés et opérateurs (Chapter III). Ces travaux ont donné lieu à plusieurs publications et contribué à des applications majeures (Chapter IV), allant d'une question théorique majeure et ouverte depuis une vingtaine d'années sur les circuits quantiques, jusqu'à l'atténuation efficace d'erreurs physiques, tout en jouant un rôle central dans le développement d'un logiciel open-source.

Le plan, ainsi que les publications issues de ce travail, sont détaillés ci-après.

- **Chapitre I** (**Contexte**). Ce chapitre passe en revue les résultats clés et les fondements sur lesquels repose cette thèse. Nous commençons par un aperçu des notations et notions mathématiques essentielles (Section I.1). Ensuite, nous introduisons la mécanique quantique (Section I.2), en décrivant les postulats, les opérateurs pertinents et le modèle de circuit quantique. Nous présentons ensuite l'optique linéaire (Section I.3). Nous expliquons ce que nous entendons par optique linéaire et photons, détaillons tous les principaux composants et leur comportement, et décrivons les méthodes d'état de l'art pour simuler les évolutions d'optique linéaire. Enfin, nous proposons une introduction concise aux langages graphiques et aux systèmes de réécriture (Section I.4), en exposant leurs fonctionnalités. Notre démarche est de définir clairement tous les concepts et distinguons soigneusement postulats, définitions et propositions de nature physique.
- **Chapitre II** (**Vers une simulation efficace et une description calculatoire des circuits d'optique linéaire**). Ce chapitre présente plusieurs nouveaux algorithmes de simulation efficaces conçus pour répondre à diverses configurations (Section II.1).

  Nous introduisons d'abord SLOS (Section II.2), une méthode qui exploite une grande quantité de mémoire pour calculer très efficacement de nombreuses probabilités de sortie. Nous proposons également une variante qui réduit l'usage mémoire tout en conservant d'excellentes performances, dépassant l'état de l'art théorique pour le calcul d'une seule probabilité de sortie, et extensible à la simulation avec adaptivité.

  Nous présentons ensuite LO-SLAP (Section II.3), une nouvelle approche offrant un compromis mémoire-temps par rapport à SLOS, consommant moins de mémoire au prix d'un temps de calcul accru. Cette méthode se révèle particulièrement efficace pour les scénarios avec adaptivité, et nous proposons une comparaison détaillé des deux algorithmes.

  Enfin, dans la dernière section (Section II.4), nous proposons un algorithme d'échantillonnage inspiré de techniques de simulation de circuits quantiques. Cet algorithme prend en charge divers états d'entrée ainsi que des circuits avec adaptivité, et nous l'appliquons à deux cas d'usage majeurs. Nous introduisons également une nouvelle formule généralisée pour le calcul de sommes de permanents dérivés de la même matrice. SLOS et LO-SLAP ont é introduits dans les articles suivants.

  - **[Heu+23b]** N. Heurtel, S. Mansfield, J. Senellart, B. Valiron : Strong Simulation of Linear Optical Processes, *Computer Physics Communications (2023)*. https://doi.org/10.1016/j.cpc.2023.108848.
  - **[GH25]** T. Goubault de Brugière, N. Heurtel : Fast and memory-efficient strong simulation of noisy adaptive linear optical circuits, *preprint (2025)*.

https://arxiv.org/abs/2503.05699.

- **Chapitre III** (**Vers une formalisation et une description formelle des circuits d'optique linéaire**). Dans ce chapitre, nous introduisons trois langages graphiques pour modéliser les circuits d'optique linéaire, chacun étant accompagné d'une théorie équationnelle complète permettant une caractérisation intégrale, où toutes les manipulations et raisonnements peuvent être effectués dans le formalisme graphique.

  Nous présentons d'abord le **LO$_{\mathbf{PP}}$**-calculus (Section III.1), un langage graphique incluant tous les circuits composés de miroirs semi-réfléchissant et de déphaseurs, deux composants préservant la polarisation, très utilisés en photonique intégrée. Nous fournissons un système de réécriture déterministe, une théorie équationnelle complète, ainsi que des formes particulières de circuits possédant des propriétés d'unicité essentielles et servant de base aux deux langages suivants.

  Nous introduisons ensuite le **LO$_{\mathbf{v}}$**-calculus (Section III.2), une extension de **LO$_{\mathbf{PP}}$** intégrant des sources de vide et la post-sélection, ainsi que des lames à retard et des séparateurs de faisceau polarisants, c'est-à-dire des composants agissant sur la polarisation des photons. Nous proposons une théorie équationnelle complète ainsi que de nouvelles formes compactes pour les circuits impliquant la polarisation et la position comme degrés de liberté.

  Nous présentons enfin le **LO$_{fi}$**-calculus (Section III.3), une extension de **LO$_{\mathbf{PP}}$** avec des sources et des détecteurs permettant des états photoniques de dimension finie arbitraire. Nous proposons une procédure de réécriture déterministe menant à de nouvelles formes uniques pour décrire de tels circuits optiques.

  Ces travaux sont introduits dans les deux publications suivantes.

  - **[Clé+22]** A. Clément, N. Heurtel, S. Mansfield, S. Perdrix, B. Valiron : LO$_{\mathrm{v}}$-Calculus: A Graphical Language for Linear Optical Quantum Circuits, *MFCS (2022)*.
    https://doi.org/10.4230/LIPIcs.MFCS.2022.35.
  - **[Heu25]** N. Heurtel : A Complete Graphical Language for Linear Optical Circuits with Finite-Photon-Number Sources and Detectors, *CSL (2025)*. Cet article a également été présenté à *QPL24*, où il a reçu le prix du meilleur article étudiant.
    https://doi.org/10.4230/LIPIcs.CSL.2025.38.

- **Chapitre IV** (**Applications des contributions théoriques**). Dans ce chapitre, nous détaillons comment les travaux précédents nous permettent d'aborder plusieurs applications clés.

  Nous montrons d'abord comment la complétude du **LO$_{\mathbf{PP}}$**-calculus permet de dériver une théorie équationnelle complète pour les circuits quantiques (Section IV.1), résolvant ainsi un problème ouvert majeur.

  Nous présentons ensuite Perceval (Section IV.2), un logiciel open-source optimisé, développé par Quandela et dédié à l'optique linéaire à variables discrètes. Ce logiciel trouve son origine dans les travaux sur SLOS et intègre désormais SLOS et LO-SLAP comme algorithmes principaux de simulation.

  Enfin, nous montrons comment des outils issus de la théorie des graphes, associés aux règles de réécriture de **LO$_{\mathbf{PP}}$**, permettent d'atténuer les erreurs physiques dans les circuits d'optique linéaire (Section IV.3). Plus précisément, nous nous intéressons aux phases induites par la diffusion de chaleur dans le circuit. Nous proposons un algorithme efficace pour atténuer ces erreurs, en prouvant l'optimalité et en établissant un critère graph-théorique pour les circuits robustes à ces perturbations.

  Ces travaux sont décrits dans les trois articles suivants.

  - **[Clé+23]** A. Clément, N. Heurtel, S. Mansfield, S. Perdrix, B. Valiron : A Complete Equational Theory for Quantum Circuits, *LICS (2023)*.

https://doi.org/10.1109/LICS56636.2023.10175801.

- **[Heu+23a]** N. Heurtel, A. Fyrillas, G. de Gliniasty, R. Le Bihan, S. Malherbe, M. Pailhas, E. Bertasi, B. Bourdoncle, P. Emeriau, R. Mezher, L. Music, N. Belabas, B. Valiron, P. Senellart, S. Mansfield, J. Senellart : Perceval: A Software Platform for Discrete Variable Photonic Quantum Computing, *Quantum (2023)*.
  https://doi.org/10.22331/q-2023-02-21-931.
- **[Fyr+25]** A. Fyrillas, N. Heurtel, S. Piacentini, N. Maring, J. Senellart, N. Belabas : Resource-efficient crosstalk mitigation for the high-fidelity operation of photonic integrated circuits with induced phase shifters, *preprint (2025)*.
  https://arxiv.org/abs/2506.05988

Pour un aperçu global du manuscrit, on peut également se référer à la table des matières. Pour un indicateur des notions introduites, veuillez consulter la table des index. Mes contributions personnelles sont détaillées dans chaque section.

Introduction

Computing is a cornerstone of modern science and technology, enabling us to process data, solve mathematical problems, and simulate complex systems. At its core, computation consists in encoding information into physical systems and manipulating them to perform operations, such as arithmetic, logic, or data storage and retrieval. Since 1945, [3] computers were primarily implemented using electronic circuits, where information is encoded in voltage levels and processed through components such as transistors [PH13]. These systems typically follow a model of computation based on logical operations acting on bits — binary digits that take only two values, usually represented as 0 and 1. This framework, known as classical computing, forms the foundation of modern algorithms and computer architectures.

Although extremely useful, classical computers face fundamental limitations. Certain current problems — such as simulating key chemical reactions, solving differential equations or factoring large integers in prime factors — seem to require billions of years to be solved using classical algorithms. This suggests the need for new ways of processing information, outside the usual computational framework and more efficient for these hard problems. What if, instead of electronic components governed by classical mechanics, we use physical systems ruled by the laws of quantum mechanics to encode and manipulate information?

That question led to a new paradigm — quantum computing — designed to overcome the limitations of classical computers. This approach leverages the principles of quantum mechanics — such as superposition, entanglement, and measurement — to perform operations in fundamentally different ways [Fey82; DP85; Llo96; NC10]. For instance, instead of the two states of a bit, quantum computers use qubits, quantum systems that can be in a superposition of two states. This led to the development of a formal framework for quantum computation [DP85], laying the foundation for how quantum algorithms can be run on a general-purpose quantum computer. Since then, the field of quantum algorithms has evolved rapidly, offering speedups ranging from quadratic [Gro96] to exponential [Sho94; HHL09], along with promising heuristics and applications in chemistry, optimization, and machine learning [Lan+10; Per+14; FGG14; Bia+17; AL18; Pre18; Cao+19]. [4] These capabilities have led to the strong and prominent development of quantum computing in recent years [Pre23].

While quantum algorithms have well-defined and promising theoretical advantages, they still need to be executed on physical devices to become practically useful. This requires quantum systems with sufficient control and scalability [DiV00]. Various platforms have thus been explored for realizing quantum hardware [NC10], each offering different approaches to implementation. This has led to an

3. The first general-purpose electronic digital computer was developed in 1945 by John P. Eckert and John W. Mauchly. It is known as ENIAC (Electronic Numerical Integrator and Computer) [GG46; Har46].

4. Note that most of the speedups are conjectures. For instance, the best-known classical algorithm for prime factorization is sub-exponential [LL93] in the number of bits of the given integer. While its quantum counterpart only runs in polynomial time [Sho94], there is no proof that a better classical algorithm cannot exist.

active competition among technologies, including superconducting systems [NPT99; Aru+19; Kja+20; Bra+24; Ach+24], trapped ions [CZ95; Bru+19; Pae+24], and cold atoms [Jak+99; Saf16; GB17].

One prominent candidate for quantum computation is discrete-variable linear optics [Kok+07; OFV09; Rud17], that utilizes single photons, the particles of light, to encode quantum information. Notably, regardless of its computation power, this technology stands out as a natural carrier of quantum information, leading applications in secured communication [BB84; BB14] and quantum networks [Kim08; WEH18]. For quantum computation, [KLM01] demonstrated that linear optical circuits with feedforward — the process of using outcomes from one stage to dynamically adapt the operations performed at a later stage in the circuit — are sufficient to efficiently perform any quantum algorithm. This finding has paved the way for many scalable and improved schemes using feedforward [YR03; Nie04; BR05; Bar+23; Gli+24].

Challenges

Building reliable photonic quantum devices presents several challenges, both at small and large scales. Like any physical quantum system, linear optical circuits are highly susceptible to errors and require precise implementation to operate as intended. This demand for precision necessitates extensive characterization, aimed at identifying the system's specific behavior and parameters. The objective is to compare the actual outputs with those predicted by a theoretical model, often derived from simulations, in order to correct any imperfections. Consequently, there is a critical need for efficient algorithms capable of handling various types of errors, inputs, and experimental configurations.

Despite this need, no efficient and generic algorithm has been developed to meet the near-term requirements of hardware characterization. This is because linear optical output probabilities are tied to the computation of matrix permanents, a notoriously hard problem [Val79], and even the task of sampling is classically hard [AA11; Bro+19]. While the computation of a single output probability has been well studied and optimized [Sch04; GIQ19; Shc20], no prior work has comprehensively addressed the efficient computation of many such probabilities simultaneously. Furthermore, although feedforward is essential for fault-tolerant photonic architectures [KLM01; Bar+23; Gli+24], no simulation algorithm in the literature has efficiently supported this setting.

While numerical simulation can assist in small-scale experiments, it becomes completely impractical for addressing mid- or large-scale systems. To design, test, and verify large-scale photonic architectures, we must move beyond purely numerical methods and develop high-level, abstract, and compositional models.

Among the formal frameworks for studying quantum systems, graphical languages have recently gained prominence for describing quantum processes — where systems are depicted as diagrams and wires that can be both visually interpreted and formally manipulated. This high-level approach has led to numerous advances in the qubit model. The leading example is ZX-diagrams [CD08; CD11; HNW18; JPV18], with applications in compilation [KW20; Bac+21; Wet+24], simulation [KWV22; KW22; KYW23], and verification [DL14; GD18]. However, these techniques have been developed exclusively for qubit-based systems, and cannot be directly applied to the photonic setting.

Contributions and outline of this thesis

In this thesis, we address both small- and large-scale challenges. For the former, we provide a wide range of state-of-the-art algorithms for both sampling and efficiently computing many output probabilities of linear optical circuits with feedforward and various types of input states (Chapter II). For the latter, we introduce graphical frameworks fitted for linear optical circuits, encompassing all

key optical components and exhibiting new properties and operators that were previously unknown (Chapter III). These works have resulted in several publications and contributed to key applications (Chapter IV), ranging from a longstanding theoretical open question in quantum circuits, to an efficient mitigation of hardware physical errors, while also playing a key role in the development of a well-maintained open-source software.

The outline and the research publications that have emerged from this work are described as follows.

- **Chapter I** (**Background**). This chapter reviews key results and state-of-the-art foundations upon which this thesis is built. We first do an overview of the notations and key mathematical notions (Section I.1). Then, we introduce quantum mechanics (Section I.2), describing the postulates, relevant operators, and the quantum circuit model. Then, we present linear optics (Section I.3). We explain what we mean by linear optics and photons, all the key components and their behavior, while describing the state-of-the-art methods for simulating linear optical evolutions. Finally, we provide a straight-to-the-point introduction of graphical languages and rewriting systems (Section I.4), exhibiting their powerful features. We adopt a careful and formal approach, clearly defining all concepts and carefully distinguishing between postulates, definitions, and propositions of physical concepts.
- **Chapter II** (**Towards efficient simulation and a computational description of linear optical circuits**). This chapter presents several efficient new simulation algorithms designed to meet a range of practical needs (Section II.1).

  We begin by introducing SLOS (Section II.2), a method that leverages a large amount of memory to achieve highly efficient computation of many output probabilities. We also propose a masked variant that reduces memory usage while still offering strong performance — remarkably outperforming the theoretical state-of-the-art for computing a single output probability, and extendable to feedforward simulation.

  Next, we introduce LO-SLAP (Section II.3), a new approach that offers a compelling memory-time trade-off compared to SLOS, using less memory at the cost of increased computation time. This method proves particularly effective for feedforward scenarios, and we provide a detailed benchmark comparing the two algorithms.

  Finally, in the last section (Section II.4), we present a general-purpose sampling algorithm inspired by techniques from quantum circuit simulation. This algorithm supports a variety of input states and feedforward, and we apply it to two prominent use cases that we detail thoroughly. In doing so, we also introduce a novel generalized formula for computing sums of permanents derived from the same matrix. SLOS and LO-SLAP are described in the following papers.

  - **[Heu+23b]** N. Heurtel, S. Mansfield, J. Senellart, B. Valiron: Strong Simulation of Linear Optical Processes, *Computer Physics Communications (2023)*.
    https://doi.org/10.1016/j.cpc.2023.108848.
  - **[GH25]** T. Goubault de Brugière, N. Heurtel: Fast and memory-efficient strong simulation of noisy adaptive linear optical circuits, *preprint (2025)*.
    https://arxiv.org/abs/2503.05699.
- **Chapter III** (**Towards a formalization and fundamental description of linear optical circuits**). In this chapter, we introduce three closely-related graphical languages for modeling linear optical circuits, each accompanied by a complete equational theory, enabling a full characterization where all manipulations and reasoning can be carried out entirely within the graphical formalism.

  First, we introduce the **LO$_{\text{PP}}$**-calculus (Section III.1), a graphical language that captures all circuits composed of beam splitters and phase shifters — two polarization-preserving components

that constitute the most widely used circuits in current integrated photonics. We present a deterministic rewriting system, a complete equational theory, along with particular forms of circuits that exhibit key uniqueness properties and will serve as a foundation for the two following languages.

Then, we present the $\mathbf{LO}_{\mathsf{v}}$-calculus (Section III.2), an extension of $\mathbf{LO}_{\mathsf{PP}}$ with vacuum sources and postselection along with wave plates and polarizing beam splitters, i.e. components acting on the polarization of the photons. We provide both a complete equational theory and new compact forms for circuits involving both polarization and position as degrees of freedom.

Finally, we introduce the $\mathbf{LO}_{fi}$-calculus (Section III.3), an extension of $\mathbf{LO}_{\mathsf{PP}}$with sources and detectors allowing arbitrary finite-dimensional photonic states. This language is particularly useful, as the addition of ancillary sources and detectors describes a whole new set of quantum logic operations. We present a deterministic rewriting procedure leading to unique and novel forms for describing such optical circuits. Remarkably, the associated proofs lead to the introduction of new operators that uniquely decompose these circuits and exhibit particularly insightful and useful properties.

These works are presented in the two following publications.

- **[Clé+22]** A. Clément, N. Heurtel, S. Mansfield, S. Perdrix, B. Valiron: LO_{v}-Calculus: A Graphical Language for Linear Optical Quantum Circuits, *MFCS (2022)*.
  https://doi.org/10.4230/LIPIcs.MFCS.2022.35.
- **[Heu25]** N. Heurtel: A Complete Graphical Language for Linear Optical Circuits with Finite-Photon-Number Sources and Detectors, *CSL (2025)*. This paper has also been presented at *QPL24*, distinguished with the best student paper award.
  https://doi.org/10.4230/LIPIcs.CSL.2025.38.

- **Chapter IV** (**Applications of the theoretical contributions**). In this chapter, we detail how the previous works enable us to tackle several key applications.

  First, we present how the completeness of the $\mathbf{LO}_{\mathsf{PP}}$-calculus can be used to derive a complete equational theory for quantum circuits (Section IV.1), thus solving a longstanding open problem.

  Then, we introduce Perceval (Section IV.2), an open-source and optimized software developed by Quandela, dedicated to discrete-variable linear optics. This software originally stemmed from the work on SLOS, and now includes both SLOS and LO-SLAP as simulation backends.

  Last but not least, we show how graph-theoretic tools, together with the rewriting rules of $\mathbf{LO}_{\mathsf{PP}}$, can help mitigate hardware errors in linear optical circuits (Section IV.3). More precisely, we focus on crosstalk, i.e. unwanted interactions between phase shifters that arise due to heat diffusion in the circuit. We provide an efficient algorithm to mitigate these errors, prove its optimality, and establish a graph-theoretical criterion for circuits that are robust to crosstalk.

  These works are described in the three following papers.

  - **[Clé+23]** A. Clément, N. Heurtel, S. Mansfield, S. Perdrix, B. Valiron: A Complete Equational Theory for Quantum Circuits, *LICS (2023)*.
    https://doi.org/10.1109/LICS56636.2023.10175801.
  - **[Heu+23a]** N. Heurtel, A. Fyrillas, G. de Gliniasty, R. Le Bihan, S. Malherbe, M. Pailhas, E. Bertasi, B. Bourdoncle, P. Emeriau, R. Mezher, L. Music, N. Belabas, B. Valiron, P. Senellart, S. Mansfield, J. Senellart : Perceval: A Software Platform for Discrete Variable Photonic Quantum Computing, *Quantum (2023)*.
    https://doi.org/10.22331/q-2023-02-21-931.

- **[Fyr+25]** A. Fyrillas, N. Heurtel, S. Piacentini, N. Maring, J. Senellart, N. Belabas: Resource-efficient crosstalk mitigation for the high-fidelity operation of photonic integrated circuits with induced phase shifters, *preprint (2025)*.
  https://arxiv.org/abs/2506.05988

For an overview of the whole manuscript, one can also refer to the table of contents. For an indicator of the notions introduced, please refer to the index table. My personal contributions are detailed in each section.

Chapter I
Background

In this thesis, several fields come together. Grounded in quantum mechanics, our work explores linear optical circuits using graphical languages — a tool from category theory — as well as rewrite theory and logic.

This chapter contains all the foundations and state-of-the-art knowledge required to follow the rest of this thesis.

1. Section I.1. We begin with standard mathematical notions and the notations used throughout the thesis. This includes, for example, Hilbert spaces and Dirac notation $\langle \cdot | \cdot \rangle$.
2. Section I.2. We then introduce quantum mechanics: its postulates, the key spaces and operators, including the Fock space and the creation and annihilation operators. We conclude with quantum circuits, a widely used representation of quantum algorithms.
3. Section I.3. Next, we review linear optics and relevant state-of-the-art results. We define photons and linear optical systems, then dive into the key properties of linear optical circuits and how they can be simulated.
4. Section I.4. Finally, we present graphical languages. We begin with essential concepts from category theory in order to define PROPs — categories particularly well-suited for diagrammatic reasoning. We conclude with core ideas from rewrite theory and logic, which are critical for formal diagram manipulation.

The index table may be consulted at any time to revisit definitions or concepts introduced in this chapter.

I.1 Mathematical definitions and notation

All concepts introduced here can be easily found on lecture notes [NC10; Val24]. We still choose to highlight the key concepts involved for completeness purposes.

I.1.1 Hilbert space

A Hilbert space is a vector space $\mathcal{H}$ equipped with an inner product $\langle \cdot, \cdot \rangle$ that is complete with respect to the norm induced by this inner product. The norm of a vector $\boldsymbol{v} \in \mathcal{H}$ is given by

$$\|\boldsymbol{v}\| = \sqrt{\langle \boldsymbol{v}, \boldsymbol{v} \rangle},$$

which is also referred to as the ℓ^2-norm or 2-norm.

We assume Hilbert spaces to be defined over the complex numbers $\mathbb{C}$, unless stated otherwise.

Given a vector space V with an inner product, its Hilbert completion, denoted $\overline{V}$, is the unique Hilbert space that contains V as a dense subspace and where all Cauchy sequences in V converge.

The tensor product of two Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, written $\mathcal{H}_1 \otimes \mathcal{H}_2$, is a Hilbert space constructed from the formal linear combinations of vectors of the form $\boldsymbol{v} \otimes \boldsymbol{w}$, with $\boldsymbol{v} \in \mathcal{H}_1$, $\boldsymbol{w} \in \mathcal{H}_2$, and extended by linearity and completion. It allows us to describe composite quantum systems, where each subsystem lives in its own Hilbert space.

Given two finite-dimensional spaces $\mathcal{H}_1$ and $\mathcal{H}_2$ of dimensions n_1 and n_2, with respective bases $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n_1}\}$ and $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{n_2}\}$, the tensor product space $\mathcal{H}_1 \otimes \mathcal{H}_2$ has dimension $n_1 \times n_2$. A natural basis of this space is given by

$$\{\boldsymbol{v}_i \otimes \boldsymbol{w}_j \mid 1 \leq i \leq n_1,\ 1 \leq j \leq n_2\}.$$

We define the dual of a vector space V, denoted V^*, as the set of linear forms from V to a field $\mathbb{K}$:

$$V^* = \{f : V \to \mathbb{K} \mid f \text{ is linear}\}.$$

In this thesis, we work over the field $\mathbb{K} = \mathbb{C}$, the complex numbers.

I.1.2 Operators and their properties

An operator on a Hilbert space $\mathcal{H}$ is a linear map $A : \mathcal{H} \to \mathcal{H}$. In the finite-dimensional case, such operators are represented by matrices. An operator A is called bounded if there exists a constant $C \geq 0$ such that

$$\|A\boldsymbol{v}\| \leq C\|\boldsymbol{v}\|$$

for all $\boldsymbol{v} \in \mathcal{H}$. Boundedness guarantees that the operator is continuous and defined on all of $\mathcal{H}$. For any bounded linear operator $A : \mathcal{H} \to \mathcal{H}$, there exists [RS72] a unique bounded operator $A^\dagger : \mathcal{H} \to \mathcal{H}$, called the Hermitian adjoint of A, such that

$$\langle A\boldsymbol{v}, \boldsymbol{w}\rangle = \langle \boldsymbol{v}, A^\dagger \boldsymbol{w}\rangle \quad \forall \boldsymbol{v}, \boldsymbol{w} \in \mathcal{H}.$$

Note that in the matrix representation, i.e. in finite-dimensional spaces, the adjoint corresponds to the conjugate transpose of the matrix.

An operator A is said to be Hermitian (or self-adjoint) if $A = A^\dagger$. Hermitian operators have real eigenvalues and are associated with observable quantities in quantum theory. The eigenvalue equation is given by

$$A\boldsymbol{v} = \lambda \boldsymbol{v},$$

where $\lambda \in \mathbb{R}$ is the eigenvalue associated with eigenvector $\boldsymbol{v}$.

I.1.3 Dirac notation

To express vectors and their duals concisely, we use the Dirac notation introduced in [Dir39]. A vector $|\boldsymbol{v}\rangle$ in a Hilbert space is called a ket, and its Hermitian adjoint $\langle \boldsymbol{v}| = |\boldsymbol{v}\rangle^\dagger$ is called a bra, which is a linear form:

$$\langle \boldsymbol{v}| : |\boldsymbol{w}\rangle \mapsto \langle \boldsymbol{v}, \boldsymbol{w}\rangle.$$

In a finite-dimensional Hilbert space, $|\boldsymbol{v}\rangle$ corresponds to a column vector, while $\langle \boldsymbol{v}|$ corresponds to its conjugate transpose, i.e., a row vector. In this setting, we recover the familiar identity:

$$\langle \boldsymbol{v}|\boldsymbol{v}\rangle = \|\boldsymbol{v}\|^2.$$

This notation also allows us to express more general operator relations. For instance, $\langle \boldsymbol{v} | A | \boldsymbol{w} \rangle$ denotes the scalar $\langle \boldsymbol{v}, A\boldsymbol{w} \rangle$, representing how the vector $|\boldsymbol{w}\rangle$ evolves under the action of the operator A and is projected onto $|\boldsymbol{v}\rangle$. More generally, the outer product $|\boldsymbol{n}\rangle\langle\boldsymbol{k}|$ defines a linear operator that maps $|\boldsymbol{k}\rangle$ to $|\boldsymbol{n}\rangle$ and sends any vector orthogonal to $|\boldsymbol{k}\rangle$ to zero.

I.2 Quantum mechanics

Physics aims to explain how the nature behaves and describe its state and evolution by mathematical formulae. In particular, when it comes to describe the behavior of small systems like particles, the most accurate known theory is quantum physics explaining phenomena – for instance the black body radiation [Pla01] and the photoelectric effect [Ein05] – that the classical physics couldn't explain.

In this section, we first present the postulates [1][Dir30; Neu32] that form the foundation of quantum mechanics (Section I.2.1). Each postulate is complemented with notes, examples, and additional references to provide deeper insights [2]. We then introduce the Heisenberg picture (Section I.2.2), which is particularly relevant for describing linear optical transformations. Next, we define the Fock space and its basis (Section I.2.3), which contain the photonic states of interest. This is followed by an introduction to the number, annihilation, and creation operators (Section I.2.4). Finally, we conclude by presenting quantum circuits (Section I.2.5), the building blocks of quantum algorithms.

I.2.1 Postulates of quantum mechanics

A. Postulate I: Description of the state of a system

> **Postulate I.a** (State)**.** *The state of an isolated physical system, at a fixed time t, is represented by a state vector $|\psi\rangle$ belonging to a Hilbert space $\mathcal{H}$ called the state space.*

If there exists two distinct valid states $|\psi\rangle \in \mathcal{H}$ and $|\psi'\rangle \in \mathcal{H}$, then by linearity, $|\chi\rangle = \alpha|\psi\rangle + \beta|\psi'\rangle$ also belongs to $\mathcal{H}$ and is also a valid state. We say that $|\chi\rangle$ is a superposition of $|\psi\rangle$ and $|\psi'\rangle$. Note that a state vector is by definition a normalized state, i.e. $|\alpha|^2 + |\beta|^2 = 1$.

For describing the state space of a system of n particles, each of them living in $\mathcal{H}_i$, we use the tensor product space $\bigotimes_{i=1}^{n} \mathcal{H}_i$. That is a direct corollary of the following postulate.

> **Postulate I.b** (Composite system)**.** *The Hilbert space of a composite system is the Hilbert space tensor product of the state spaces associated with the component systems. For a non-relativistic system consisting of a finite number of distinguishable particles, the component systems are the individual particles.*

Note that a n-particle state $|\psi\rangle \in \mathcal{H}^{\otimes n}$ that can be decomposed as $|\psi\rangle = |\psi_1\rangle \dots |\psi_n\rangle$ where $|\psi_i\rangle \in \mathcal{H}$ is said to be separable. A state that is not separable is called entangled [Sch35; Sch36]. The interest for entangled states lies in two aspects. On the computational side, those systems are necessarily harder to classically simulate, because they can't be divided into smaller simulable systems. On the physical side, performing an operation (Postulate II.b and Postulate II.c) on one particle of a two entangled particle state, affect the full state, therefore affecting the other particle that may have been sent far away. Those kinds of experiments explicit the non-local properties of quantum mechanics [EPR35; Boh35; Ein36; Bel64].

1. We use the most common set of postulates, implicitly assuming the Copenhagen interpretation [Boh35]. Other sets of postulates and interpretations of quantum mechanics also exist [Rue11].

2. For a comprehensive pedagogical overview of quantum mechanics [Fey+66], one can refer to the three volumes by Cohen-Tannoudji, Diu, and Laloë [CDL19a; CDL19b; CDL19c].

At this point, we only described what the state *is*, which is not what we *see*. Indeed, we don't directly observe states in superposition. Observing a superposed state transforms it into an observable state, the one we *see*; that process is called the *measurement* (Postulate II.b). Before describing how measurement works and the resulting state after the measurement (Postulate II.c), we need to characterize what are the physical properties or quantities that can be measured. Those are called the observables, described in the following postulate.

B. Postulate II: Measurement

Postulate II.a (Observable)**.** *Every measurable physical quantity $\mathcal{A}$ is described by a Hermitian operator $\hat{\mathcal{A}}$ acting on the state space $\mathcal{H}$. This operator is an observable, meaning that its eigenvectors form a basis for $\mathcal{H}$. The result of measuring a physical quantity $\mathcal{A}$ must be one of the eigenvalues of the corresponding observable $\hat{\mathcal{A}}$.*

Since the observables are Hermitian operators, their eigenvalues are always real, which are indeed the values that can be read on a measuring device. If the spectrum of the observable is discrete, then the possible results are *quantized*, being the reason behind the name of quantum mechanics.

Note that compared to classical quantities, some quantum quantities can't be observed simultaneously. The observables associated to such quantities are called incompatible. Mathematically, let $[\hat{A}, \hat{B}] = \hat{A}\hat{B} - \hat{B}\hat{A}$ be the commutator of two observables: $\hat{A}$ and $\hat{B}$ are incompatible if their commutator is nonzero. This property is the foundation of the Heisenberg's indeterminacy principle [Hei27; Boh28], stating there is a limit to the precision we can know of two incompatible physical quantities. For instance, let $\hat{x}$ and $\hat{p}$ be the position and momentum observables. The canonical commutation relation $[\hat{x}, \hat{p}] = i\hbar$ leads to $\Delta\hat{x}\Delta\hat{p} \geq \frac{\hbar}{2}$, where $\Delta\hat{x}$ are $\Delta\hat{p}$ are respectively the uncertainties of $\hat{x}$ and $\hat{p}$.

Postulate II.b (Result of the measurement)**.** *When the physical quantity $\mathcal{A}$ is measured on a system in a normalized state $|\psi\rangle$, the probability of obtaining an eigenvalue (denoted a_n for discrete spectra and α for continuous spectra) of the corresponding observable $\hat{\mathcal{A}}$ is given by the amplitude squared of the appropriate wave function (projection onto corresponding eigenvector).*

$$\begin{aligned}
\mathrm{P}(a_n) &= |\langle a_n|\psi\rangle\rangle|^2 && \text{Discrete, non degenerate spectrum} \\
\mathrm{P}(a_n) &= \sum_i^{g_n} |\langle a_n^i|\psi\rangle|^2 && \text{Discrete, degenerate spectrum} \\
d\mathrm{P}(\alpha) &= |\langle\alpha|\psi\rangle|^2 d\alpha && \text{Continuous, nondegenerate spectrum}
\end{aligned}$$

where g_n denotes the degeneracy of the eigenvalue a_n.

This postulate, also known as the Born rule [Bor26], leads to the probabilistic behavior of quantum mechanics. We access to only one of the value with a given probability. We need another postulate to describe what is the outcome state after being measured.

Postulate II.c (Post-measurement)**.** *If the measurement of the physical quantity $\mathcal{A}$ on the system in the state $|\psi\rangle$ gives the result a_n, then the state of the system immediately after the measurement is the normalized projection of $|\psi\rangle$ onto the eigensubspace associated with a_n:*

$$|\psi\rangle \mapsto \frac{P_n|\psi\rangle}{\sqrt{\langle\psi|P_n|\psi\rangle}}$$

where $P_n = |a_n\rangle\langle a_n|$.

Example I.1. Let $\{|0\rangle, |1\rangle\}$ be two orthogonal states a particle can be in. A generic two-particle state is $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle = |0\rangle(\alpha|0\rangle + \beta|1\rangle) + |1\rangle(\gamma|0\rangle + \delta|1\rangle)$. If we measure the

state of the first particle, we obtain $|0\rangle$ with probability $|\alpha|^2+|\beta|^2$, and $|1\rangle$ with probability $|\gamma|^2+|\delta|^2$. If we obtain $|0\rangle$, then the resulted state is $\frac{|0\rangle(\alpha|0\rangle+\beta|1\rangle)}{\sqrt{|\alpha|^2+|\beta|^2}}$.

The measurement, being a probabilistic projection on one subspace, is therefore irreversible. If we don't know the distribution of $|\psi\rangle$, we can't come back to $|\psi\rangle$ after measuring it.

However, for an isolated system that is not measured, the evolution is always reversible. That is the consequence of the following postulate, describing the evolution of a quantum system.

C. Postulate III: Time evolution of a system

Postulate III (Schrödinger equation - Unitary evolution)**.** *The time evolution of the state vector $|\psi(t)\rangle$ is governed by the Schrödinger equation, where $\hat{H}(t)$ is the observable associated with the total energy of the system (called the Hamiltonian):*

$$i\hbar\frac{d}{dt}|\psi(t)\rangle = \hat{H}(t)|\psi(t)\rangle$$

Equivalently, the time evolution of a closed system is described by a unitary transformation on the initial state:

$$|\psi(t)\rangle = \hat{U}(t;t_0)|\psi(t_0)\rangle$$

where $\hat{U}(t;t_0) = \exp\left(-\frac{i}{\hbar}\int_{t_0}^{t}\hat{H}(s)ds\right)$

This formulation of quantum mechanics, where the state of a system is described by a wave function $|\psi(t)\rangle$ evolving according to the Schrödinger equation, is referred to as first quantization. Note the unitary transformations conserve the norms of the states [Wig31], thus keeping the probability distribution normalized, and are invertible operations, thus making every transformation reversible.

One could ask why we don't observe such probabilistic behaviors for large systems, where their evolutions seem to be described by classical and deterministic laws. The link between these two apparently different cases is explained by a phenomenon called decoherence [Zur03]. In an open system, the particles interact with their environment, i.e. numerous other particles, making the particles of the system entangled with the environment. This interaction causes the superposition of the state to vanish, making the system appear to behave classically, even though the underlying laws governing it remain quantum mechanical.

D. Postulate IV: Symmetrization

Postulate IV (Symmetrization)**.** *When a system includes several identical particles, only certain kets of its state space can describe its physical states. Physical kets are, depending on the nature of the identical particles, either completely symmetric or completely antisymmetric with respect to permutation of these particles. Those particles for which the physical kets are symmetric are called bosons, and those for which they are antisymmetric, fermions.*

The symmetrization postulate (see [MG64] for review) is particularly relevant when looking at particles with spins [UG26; DF28; Wig39]. In particular, we distinguish two types of particles [Pau40]: the fermions (antisymmetric wave function) following the Pauli exclusion principle [Pau25; DF25], and the bosons (symmetric wave function) following Bose-Einstein statistics [Bos24; Ein24].

The particles of interest in optics, a fortiori of interest for this thesis, are photons [Ein05; Lew26]. Those particles are bosons, and their behavior is described in the next section.

Towards simplification. It has been shown that some previous postulates can be derived from the others, under heuristic assumptions. Recent examples include the measurements postulates (Postulates II.a, II.b and II.c) that have been shown [MGM19] to be a consequence of the state postulate

(Postulate I.a), the composite postulate (Postulate I.b) and the unitary evolution postulate (Postulate III). In [CMA21], the composite system postulate (Postulate I.b) has been shown to be a consequence of the state postulate (Postulate I.a) and the measurement postulates (Postulates II.a, II.b and II.c).

I.2.2 Heisenberg picture

Before looking at another way to represent the states, known as the second quantization (Section I.2.3), it is important to notice there is also another way to characterize the evolution of quantum systems.

Postulate III is so-called the Schrödinger picture. In this formalism, the observables are constant and that is the state $|\psi\rangle$ which is evolving. Given an observable $\hat{\mathcal{O}}$ and a transformation $\hat{U}(t;t_0)$, the state evolves as $|\psi(t)\rangle = \hat{U}(t;t_0)$ and $\hat{\mathcal{O}}$ remains the same.

Another and equivalent [Dir30] way of describing the evolution is the so-called Heisenberg picture. In this formalism, the state remains constant at $|\psi(t_0)\rangle$, while the observable evolves as $\hat{U}^\dagger\hat{\mathcal{O}}\hat{U}$. The equivalence and the formula can be quickly shown thanks to the expectation value of $\hat{\mathcal{O}}$, that should remain the same in every formalism:

$$\langle\psi(t)|\,\hat{\mathcal{O}}\,|\psi(t)\rangle = \langle\psi(t_0)|\,\hat{U}(t;t_0)^\dagger\hat{\mathcal{O}}\hat{U}(t;t_0)\,|\psi(t_0)\rangle$$

Heisenberg picture is particularly relevant to study the evolution of linear optical systems (Section I.3.2), where we prefer using the evolution of the operators instead of the states to predict the outcome.

I.2.3 Second quantization: Fock space

In order to represent the quantum state of many indistinguishable particles, it is usual to use the second quantization [Dir27; JW28] rather than the first quantization that uses the wave function $\psi(r,t)$ for individual particles. As described by Postulate IV, there are only two types of particles. In this section, we only focus on one of the two: bosons.

Let us consider two indistinguishable bosons in two different states $|\psi_0\rangle$ and $|\psi_1\rangle$. By their indistinguishability and symmetry, both states $|\psi_0,\psi_1\rangle$ and $|\psi_1,\psi_0\rangle$ are *equally valid*. That is why the quantum state representing that system is symmetrized, being invariant through swaps: $|\psi\rangle = \frac{1}{\sqrt{2}}(|\psi_0,\psi_1\rangle + |\psi_1,\psi_0\rangle)$. For n bosons, we would need to consider the $n!$ permutations of the group S_n. For simplicity of the notations, we introduce the symmetric tensor product $\otimes_S$.

Definition I.2 (Symmetric tensor product)**.** Let $\otimes_S$ be the symmetric tensor product [a] defined as:

$$|\psi_1\rangle \otimes_S \ldots \otimes_S |\psi_n\rangle = \frac{1}{\sqrt{n!}}\sum_{\sigma\in S_n}|\psi_{\sigma(1)},\ldots,\psi_{\sigma(n)}\rangle$$

a. Note that $\otimes_S$ is multilinear and is therefore a well-defined tensor product.

If each particle lives in the Hilbert space $\mathcal{H}$, then the space for n-boson systems is $\mathcal{H}^{\otimes_S n}$. More generally however, the number of particles is not necessarily constant or bounded. To characterize the space of a many indistinguishable particles, e.g. bosons, we introduce the symmetric Fock space [Foc32].

Definition I.3 (Symmetric Fock space)**.** Let $\mathcal{H}$ be a Hilbert space. We define the symmetric Fock

space as the following.

$$\mathcal{F}(\mathcal{H}) \coloneqq \overline{\bigoplus_{n=0}^{\infty} \mathcal{H}^{\otimes_S n}}$$

where we recall the overline to be the Hilbert completion.

Remark I.4 (About fermions). Note that fermions, unlike bosons, obey the Pauli exclusion principle, which states that no two identical fermions can occupy the same quantum state. This is encoded mathematically by the antisymmetric tensor product, which is used to construct the antisymmetric Fock space.

Definition I.5 (Modes)**.** Let consider a finite Hilbert space $\mathcal{H}$. An orthogonal basis of $\mathcal{H}$, representing the states a particle can be, are called the modes. The modes can be understood as the *degrees of freedom* of a particle. In this thesis, the number of modes is noted with the letter m, where $m = \dim(\mathcal{H})$.

For a basis of $\mathcal{F}(\mathcal{H})$, rather than using the basis of $\mathcal{H}$ with $\otimes_S$, a more suited and used representation is the occupation number representation, where we indicate *"how many photons are there in each state"*. We consider the following basis states, denoted as Fock states.

Definition I.6 (Fock states)**.** Let $\mathcal{F}_m \coloneqq \{|n_1, n_2, \ldots, n_m\rangle, (n_1, n_2, \ldots, n_m) \in \mathbb{N}^m\}$. The state $|n_1, n_2, \ldots, n_m\rangle$ is called a Fock state, and represents a configuration where n_i photons occupy the i^{th} mode.

In practice, for a finite Hilbert space $\mathcal{H}$, as $\mathcal{H} \simeq \mathbb{C}^m$, it is tantamount and often simpler to consider $\mathcal{F}(\mathcal{H}) \simeq \mathcal{F}(\mathbb{C}^m) \simeq \ell_2(\mathcal{F}_m) \simeq \ell_2(\mathbb{N}^m)$.

I.2.4 Number, annihilation and creation operators

As detailed in Section I.3.2, in order to study the evolution of quantum transformations on the Fock states, it is common to look at the evolutions on operators instead.

Let first define the particle number operator $\hat{n}$, that is counting the number of particles in a mode.

Definition I.7 (Particle number operator)**.** Let $\hat{n}$ be the number operator, defined as follows:

$$\hat{n} \coloneqq \sum_{n=0}^{\infty} n \, |n\rangle \langle n|$$

We define the number operator on the i^{th} mode $\hat{n}_i \coloneqq \mathbb{I} \otimes \cdots \otimes \hat{n} \otimes \ldots \mathbb{I}$, where $\mathbb{I}$ is the identity operator.

Note that we have $\hat{n} = \hat{n}^\dagger$, therefore $\hat{n}$ is Hermitian and can be an observable. It aligns with our intuition that we can indeed measure and observe the number of particles of a system.

As the real eigenvalues of $\hat{n}$ are nonnegative, $\hat{n}$ is in particular semi-definite. This allows to decompose this operator into a product: $\hat{n} = \hat{a}^\dagger \, \hat{a}$. These two operators are the creation and annihilation operators.

Definition I.8 (Annihilation operator)**.** Let $\hat{a}$ be the annihilation operator, defined as follows:

$$\hat{a} \coloneqq \sqrt{n} \, |n-1\rangle \langle n|$$

where we note that $\hat{a} \, |0\rangle = 0 \neq |0\rangle$. We define the annihilation operator on the i^{th} mode

$\hat{a}_i \coloneqq \mathbb{I} \otimes \cdots \otimes \hat{a} \otimes \ldots \mathbb{I}$.

Definition I.9 (Creation operator)**.** Let $\hat{a}^\dagger$ be the creation operator, defined as follows:

$$\hat{a}^\dagger \coloneqq \sqrt{n+1}\,|n+1\rangle\langle n|$$

We define the creation operator on the i^{th} mode $\hat{a}_i^\dagger \coloneqq \mathbb{I} \otimes \cdots \otimes \hat{a}^\dagger \otimes \ldots \mathbb{I}$.

These two operators satisfy the following commutation relations:

$$[\hat{a}, \hat{a}] = 0 \quad [\hat{a}^\dagger, \hat{a}^\dagger] = 0 \quad [\hat{a}, \hat{a}^\dagger] = \mathbb{I} \tag{I.1}$$

where we recall the commutator definition $[\hat{A}, \hat{B}] = \hat{A}\hat{B} - \hat{B}\hat{A}$.

Any Fock state can be decomposed as a product of creation operators acting on the vacuum state:

$$|n_1, n_2, \ldots, n_m\rangle = \frac{(\hat{a}_1^\dagger)^{n_1}(\hat{a}_2^\dagger)^{n_2}\ldots(\hat{a}_m^\dagger)^{n_m}}{\sqrt{n_1! n_2! \ldots n_m!}}\,|0, 0, \ldots, 0\rangle \tag{I.2}$$

Note that as $\hat{n} = \hat{a}^\dagger\,\hat{a}$ is an observable, its evolution with the Heisenberg picture is described by $\hat{n} \to \hat{U}^\dagger \hat{n} \hat{U}$. As $\hat{U}^\dagger \hat{n} \hat{U} = \hat{U}^\dagger\,\hat{a}^\dagger\,\hat{a}\,\hat{U} = \hat{U}^\dagger\,\hat{a}^\dagger(\hat{U}\hat{U}^\dagger)\,\hat{a}\,\hat{U} = (\hat{U}^\dagger\,\hat{a}^\dagger\,\hat{U})(\hat{U}^\dagger\,\hat{a}\,\hat{U})$, for constituency and similarly we can describe the evolution of the creation operator in the Heisenberg picture:

$$\hat{a}^\dagger \to \hat{U}^\dagger\,\hat{a}^\dagger\,\hat{U} \tag{I.3}$$

We see in Section I.3.2 which operations $\hat{U}$ are characteristic of linear optical circuits. Using Equation (I.2) and Equation (I.3), we can deduce how the Fock states evolve through a linear optical circuit, as detailed in Section I.3.5.

I.2.5 Quantum computing and quantum circuits

To perform a computation with a computer, the information must first be encoded onto a physical system. For practical reasons, the most unit basic of information of classical computers, the bit [3], uses only two values [Sha48]. These values are often represented by the digits 0 and 1, so mathematically we have $\texttt{bit} \in \{0, 1\}$. Physically, the values of a bit correspond to the high and low voltage states of a transistor [PH13; HH15]. The logical operations performed on bits, such as NOT, AND, and OR, are based on Boolean algebra [Boo47; Boo54], and can also be implemented using transistors [Sha38].

By analogy with classical computers and also for practical reasons, quantum computers also usually have two quantum states as their unit of information. This unit is called a qubit, and the two states are usually denoted as $|0\rangle$ and $|1\rangle$. As these are quantum states, they can be a superposition, so mathematically we have $\texttt{qubit} = \alpha\,|0\rangle + \beta\,|1\rangle$, with $|\alpha|^2 + |\beta|^2$. In some cases, we may be interested in doing computation with qudits [Wan+20], where there are $d > 2$ orthogonal quantum states for the unit of information. Note that contrary to the classical implementation relying heavily on one technology (transistors), there are many systems pursuing the path of computing with qubits [NC10].

While classical computations on n bits can *only* map $\{0, 1\}^n \to \{0, 1\}^n$, a quantum algorithm on n qubits can represent any unitary transformation from $\mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$. In practice, however, a generic large unitary cannot be directly implemented, and operations are typically limited to logical gates acting on a few qubits. These quantum operations are referred to as gates, and their physical implementation depends heavily on the technology used.

3. The word bit is a portmanteau of binary digit.

Interestingly, even a finite set of gates can approximate any unitary operation [Bar+95; Boy+00; NC10]. Such sets are called universal, and they enable the realistic execution of quantum algorithms, provided the gates of a universal set are indeed implementable.

To represent a quantum algorithm, it is common to specify the sequential order in which the elementary gates are performed, as well as the qubits on which they are applied. A ubiquitous representation of such sequences is quantum circuits.

Definition I.10 (Quantum circuits (Informal))**.** A quantum circuit is a model for quantum computation where wires represent qubits and gates are depicted as boxes or symbols acting on these wires, as illustrated in Figure I.1.

Quantum circuits rely on sequential composition and parallel operations. The latter is captured by the tensor product $\otimes$. An n-qubit state resides in the space $\bigotimes_{i=1}^{n} \mathbb{C}^2 = \mathbb{C}^{2^n}$, which has dimension 2^n. A quantum gate acting on k qubits (wires) is thus represented by a unitary matrix of size $2^k \times 2^k$.

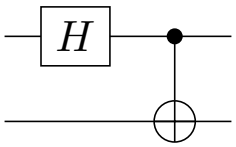


Figure I.1 – Quantum circuit on 2 qubits. Each qubit is initialized at the state $|0\rangle$. The Hadamard gate is applied on the first qubit, and then a controlled Not is applied on the two.

We introduce in Table I.1 some gates with their matrix and circuit representation, that are used in Section IV.1. With these definitions, the output state of Figure I.1 is the Bell state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

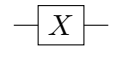
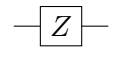
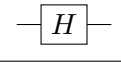
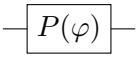
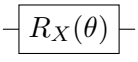
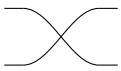
Operator	Gate	Matrix
Pauli X		$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Pauli Z		$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
Hadamard		$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
Phase		$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix}$
Rotation X		$\begin{pmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$
SWAP		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
Controlled Not		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

Table I.1 – Quantum gates and their corresponding unitary matrices.

Remark I.11 (Informal definition). Note the provided definition is informal and not mathematically rigorous. In Section I.4, we introduce PROP, a mathematical framework from category theory that formally defines quantum circuits (Example I.75).

I.3 Linear optics

Linear optics is the study of light passing through linear optical components [Lou00]. In this thesis, we study the evolution of photons through linear optical circuits.

In this section, we first build the physics foundation to understand what the photon is (Section I.3.1). Then, we introduce a formal definition of linear optics (Section I.3.2), before characterizing linear optical circuits and their core components (Section I.3.3). Next, we introduce spatial-mode linear optics and some key properties (Section I.3.4). We conclude with a state of the art on the simulation of linear optical circuits (Section I.3.5) and a key tool in their simulation: the computation of matrix permanents (Section I.3.6).

I.3.1 Quantum model of light

In order to introduce the quantum model of light and its properties, we first need to define the classical model of light, i.e. a wave (Section I.3.1.A.) in the classical electromagnetic field (Section I.3.1.B.). We then describe the quantum model of such a field, that is, the quantized electromagnetic field (Section I.3.1.C.). We finish by defining a photon as the quantum of such a quantized field (Section I.3.1.D.).

A. Classical wave

Definition I.12 (Wave)**.** A wave $\boldsymbol{u}(\boldsymbol{r},t)$ is a solution of the wave equation:

$$\frac{\partial^2 \boldsymbol{u}}{\partial t^2} = c^2 \left(\frac{\partial^2 \boldsymbol{u}}{\partial x^2} + \frac{\partial^2 \boldsymbol{u}}{\partial y^2} + \frac{\partial^2 \boldsymbol{u}}{\partial z^2} \right), \tag{I.4}$$

where $r = (x, y, z)$ denotes the Cartesian coordinates, t the time and c is a real non-negative number. When the wave is not stationary, c represents the wave propagation speed. In the case of electromagnetic waves in vacuum, the speed [4] is $c = 2,99792459.10^8 m.s^{-1}$.

Prominent waves that are key to introduce several important notions are monochromatic plane waves:

Definition I.13 (Monochromatic plane waves)**.** A monochromatic plane wave is

$$\boldsymbol{u}_{\boldsymbol{k}}(\boldsymbol{r},t) = \sum_{\sigma} A_p(t) \boldsymbol{\varepsilon}_{\boldsymbol{k},\sigma} \left(e^{i(\boldsymbol{k}\cdot\boldsymbol{r} - \omega t + \varphi_\sigma(t))} + e^{-i(\boldsymbol{k}\cdot\boldsymbol{r} - \omega t + \varphi_\sigma(t))} \right)$$

where:

- The vector $\boldsymbol{k}$ is the wave vector. It corresponds to the direction of propagation of the wave, since $\boldsymbol{u}_{\boldsymbol{k},\omega}(\boldsymbol{r}, t + \Delta t) = \boldsymbol{u}_{\boldsymbol{k},\omega}(\boldsymbol{r} + \Delta r, t)$. Note that we have $\|\boldsymbol{k}\| = \frac{2\pi}{\lambda}$ where λ is the wavelength, i.e. the distance between consecutive crests.
- The value $\omega_{\boldsymbol{k}}$ is the angular frequency. It is the number of cycles per unit of time. Note

4. Note that the value of the speed of light in vacuum is exact and not an approximation. Indeed, since the 17th General Conference on Weights and Measures (CGPM) in 1983, the meter is defined as the length of the path travelled by light in vacuum during a time interval of 1/299,792,458 of a second.

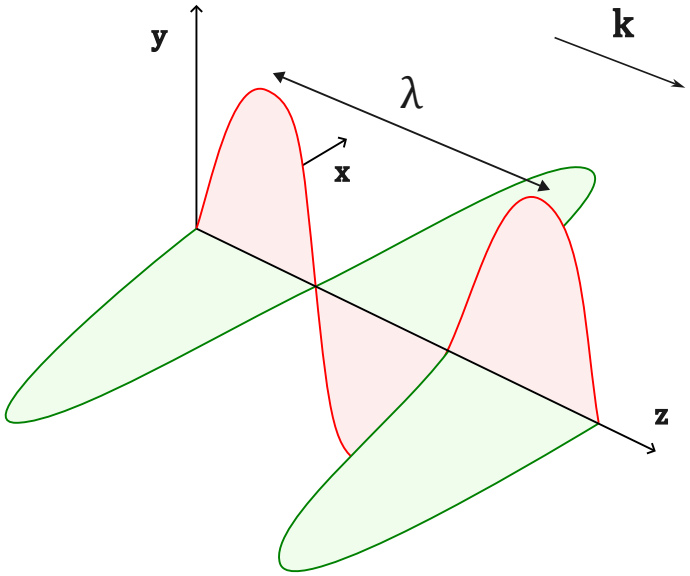


Figure I.2 – Representation of a plane wave propagating along the z axis, with two orthogonal polarizations depicted in red and green. The symbols λ and $\boldsymbol{k}$ represent the wavelength and the wave vector, respectively.

that we have $\omega_{\boldsymbol{k}} = c\|\boldsymbol{k}\|$ where c is the wave propagation speed and that $f = \frac{\omega_{\boldsymbol{k}}}{2\pi}$ is called the frequency of the wave.

- The real value φ is the phase shift. A phase is thus an angle-like quantity representing a fraction of the cycle covered.
- We sum over $\sigma \in \{V, H\}$, labeling the polarization orthonormal vectors $\boldsymbol{\varepsilon}_{\boldsymbol{k},V}$ and $\boldsymbol{\varepsilon}_{\boldsymbol{k},H}$. The polarization represents the orientations of the oscillations, that are labeled V for vertical and H for horizontal.
- The real value $A_{\boldsymbol{k},\sigma}(t)$, up to a normalization factor[5] of $1/2$, represents the amplitude of the wave.
- The configuration $(\boldsymbol{k}, \sigma)$ is called a mode.

A plane wave is illustrated in Figure I.2.

Remark I.14 (Real values). Note that even though we decompose quantities using complex numbers to simplify calculations, waves representing physical quantities are always real-valued. In particular, we recall Euler formula: $e^{iX} + e^{-iX} = 2\cos(X)$, explaining the sinusoidal form of Figure I.2.

Note that perfect plane waves in an infinite volume mean they would theoretically span an infinite space and therefore have an infinite energy[6]. Still, they remain a useful mathematical model as the general solution of Equation (I.4) can be expressed as a sum of plane waves. This is illustrated in the following example.

Example I.15 (Wave in a finite volume)**.** If $\boldsymbol{u}(\boldsymbol{r}, t)$ is confined in a finite volume $V = L^3$ we have:

$$\boldsymbol{u}(\boldsymbol{r}, t) = \sum_{\boldsymbol{k},\sigma} \boldsymbol{\varepsilon}_{\boldsymbol{k},\sigma} \left(A_{\boldsymbol{k},\sigma}(t) e^{i(\boldsymbol{k}\cdot\boldsymbol{r} - \omega_{\boldsymbol{k}} t)} + A^*_{\boldsymbol{k},\sigma}(t) e^{-i(\boldsymbol{k}\cdot\boldsymbol{r} - \omega_{\boldsymbol{k}} t)} \right) \tag{I.5}$$

5. Technically, the amplitude is $\frac{1}{2} A_{\boldsymbol{k},\sigma}(t)$, the factor coming from the Euler decomposition of the exponential.
6. The energy $E \propto \int_{\mathbb{R}^3} |A|^2 d^3\boldsymbol{r}$ diverges.

where we sum over $\mathbf{k} = \frac{2\pi}{L}(n_x, n_y, n_z)$, with $(n_x, n_y, n_z) \in \mathbb{N}^3$. Note contrary to Definition I.13, $A_{\boldsymbol{k},p}(t)$ is now a complex value as we incorporated the phase.

Remark I.16 (Fourier decomposition). A decomposition of a function into a sum of simple trigonometric functions can be seen as the Fourier coefficients of the Fourier series [Fou22; SS03].

Definition I.17 (Interference)**.** When two or more waves superpose, it can lead to constructive or destructive effects depending on their relative phase differences [You04]. Such phenomena is called interferences.

B. Classical electromagnetic field

In physics, a field is a function that assigns a physical quantity to each point in space and time. Fields returning a scalar value (resp. a vector) are called scalar (resp. vector) fields.

A fundamental example of a vector field is the electromagnetic field, which consists of two interrelated components:

- **Electric field.** The presence of charges (e.g., electrons) creates attractive or repulsive forces (Coulomb force [Cou85]) depending on their charge. The effect of these forces is described by the electric vector field, noted $\boldsymbol{E}$.
- **Magnetic field.** Moving charges generate forces that cannot be explained by electrostatic laws alone. This effect is described by the magnetic vector field, noted $\boldsymbol{B}$.

Note that these fields interact and affect each other. Their interaction is fully described by Maxwell's equations [Max65], and the study of these interactions is known as electrodynamics. From these equations, we can derive a wave equation (Equation (I.4)), explaining the general form of the electromagnetic field is very similar to Equation (I.5).

Definition I.18 (Classical electromagnetic field)**.** The classical electromagnetic field is the pair $(\boldsymbol{E}, \boldsymbol{B})$ of the electric and magnetic fields. Their evolutions are described by Maxwell's equations. The general form for the electromagnetic field is:

$$\begin{aligned}
\boldsymbol{E}(\boldsymbol{r}, t) &= \sum_{\boldsymbol{k},\sigma} \boldsymbol{\varepsilon}_{\boldsymbol{k},\sigma} \left(E_{\boldsymbol{k},\sigma}(t) e^{i(\boldsymbol{k}\cdot\boldsymbol{r} - \omega_{\boldsymbol{k}} t)} + E^*_{\boldsymbol{k},\sigma}(t) e^{-i(\boldsymbol{k}\cdot\boldsymbol{r} - \omega_{\boldsymbol{k}} t)} \right) \\
\boldsymbol{B}(\boldsymbol{r}, t) &= \sum_{\boldsymbol{k},\sigma} (\boldsymbol{k} \times \boldsymbol{\varepsilon}_{\boldsymbol{k},\sigma}) \left(B_{\boldsymbol{k},\sigma}(t) e^{i(\boldsymbol{k}\cdot\boldsymbol{r} - \omega_{\boldsymbol{k}} t)} + B^*_{\boldsymbol{k},\sigma}(t) e^{-i(\boldsymbol{k}\cdot\boldsymbol{r} - \omega_{\boldsymbol{k}} t)} \right)
\end{aligned}$$

where we sum over all the modes $(\boldsymbol{k}, \sigma)$, i.e. all the configurations that are solutions to Maxwell's equations, and where $(\boldsymbol{k} \times \boldsymbol{\varepsilon}_{\boldsymbol{k},\sigma})$ is the cross product. Similarly to Equation (I.5), $\boldsymbol{k}$ represent the wave vectors and σ the polarization.

We can now define what the classical model of light is.

Definition I.19 (Classical model of light)**.** In classical physics, light is a wave of the electromagnetic field.

Note that in the classical theory, the energy of the electromagnetic field is continuous.

Definition I.20 (Energy of an electromagnetic wave in vacuum)**.** The energy in a volume V at time t of an electromagnetic wave is:

$$\int_V \frac{\varepsilon_0}{2} \|\boldsymbol{E}(\boldsymbol{r}, t)\|^2 + \frac{1}{2\mu_0} \|\boldsymbol{B}(\boldsymbol{r}, t)\|^2 d^3\boldsymbol{r} \tag{I.6}$$

where the constant ε_0 and μ_0 are respectively the permittivity and the permeability of free space, and $\boldsymbol{E}$ and $\boldsymbol{B}$ are respectively the electric and magnetic fields.

C. Quantization of the electromagnetic field

As discussed in Section I.2, the classical description of light fails to explain phenomena like the photoelectric effect [Ein05]. It was observed that electrons could be ejected from a metal surface when light with a frequency above a certain threshold was shone on it. Surprisingly, the kinetic energy of the ejected electrons did not depend on the light's brightness (amplitude of the electromagnetic field), but instead increased with the frequency of the light. This contradicted classical wave theory, where energy is tied to the amplitude, not the frequency, cf. Equation (I.6).

To resolve this paradox, Einstein proposed a particle-like interpretation: light should be viewed as a stream of *particles* whose energy depends on frequency, according to Planck's relation [Pla01]. This model suggests that light exhibits discrete (quantized) behavior rather than a continuous spectrum of energy values like the amplitudes of the classical electric and magnetic fields. For a complete description of light, we therefore need to adapt our continuous model to account for this discrete, particle-like behavior.

The process of transitioning from a continuous to a discrete description is called quantization. In this framework, we replace the classical electromagnetic field of Maxwell's theory by a quantized electromagnetic field.

Definition I.21 (Quantized electromagnetic field)**.** The general expression of the quantized electromagnetic field at a fixed time t is:

$$\begin{aligned}\hat{\boldsymbol{E}}(\boldsymbol{r}) &= \sum_{\boldsymbol{k},\sigma} \boldsymbol{\varepsilon}_{\boldsymbol{k},\sigma} \left(E_{\boldsymbol{k},\sigma}\, \hat{a}_{\boldsymbol{k},\sigma}\, e^{i\boldsymbol{k}\cdot\boldsymbol{r}} + E^*_{\boldsymbol{k},\sigma}\, \hat{a}^\dagger_{\boldsymbol{k},\sigma}\, e^{-i\boldsymbol{k}\cdot\boldsymbol{r}} \right) \\ \hat{\boldsymbol{B}}(\boldsymbol{r}) &= \sum_{\boldsymbol{k},\sigma} (\boldsymbol{k} \times \boldsymbol{\varepsilon}_{\boldsymbol{k},\sigma}) \left(B_{\boldsymbol{k},\sigma}\, \hat{a}_{\boldsymbol{k},\sigma}\, e^{i\boldsymbol{k}\cdot\boldsymbol{r}} + B^*_{\boldsymbol{k},\sigma}\, \hat{a}^\dagger_{\boldsymbol{k},\sigma}\, e^{-i\boldsymbol{k}\cdot\boldsymbol{r}} \right)\end{aligned}$$

where $\hat{a}_{\boldsymbol{k},\sigma}$ and $\hat{a}^\dagger_{\boldsymbol{k},\sigma}$ are respectively the creation and annihilation operators (Section I.2.4) on the mode $(\boldsymbol{k},\sigma)$. Both $\hat{E}$ and $\hat{B}$ are operators acting on quantum states.

Remark I.22 (Observable). The quantized electric and magnetic fields are Hermitian. Thus, these operators are observable (cf Postulate I.2.1.B.) and correspond to quantities that can be measured (Postulate I.2.1.B.). This is in adequacy with the classical theory, where these fields are real functions that can be measured.

D. Photon

In the quantized electromagnetic field (Definition I.21), the creation operator corresponds to the addition of a discrete quantum of energy to the system, while the annihilation operator corresponds to its removal. This discrete behavior of energy levels of the electromagnetic field is exactly what characterizes photons.

Definition I.23 (Photon)**.** In quantum field theory, a photon is the fundamental quantum of the quantized electromagnetic field. Given a mode $(\boldsymbol{k},\sigma)$ of the quantized electromagnetic field, the smallest possible excitation of that mode is a photon of momentum $\boldsymbol{p} = \hbar\boldsymbol{k}$ and of polarization σ. Utilizing the Fock state representation (Section I.2.3), a photon in the mode $(\boldsymbol{k},\sigma)$ is represented

as the smallest excitation of the vacuum state $|0\rangle$ as follows:

$$|1_{\boldsymbol{k},\sigma}\rangle = \hat{a}^\dagger_{\boldsymbol{k},\sigma}\,|0\rangle$$

In this thesis, we consider a fixed frequency, i.e. the norm of $\boldsymbol{k}$ is constant, and only a finite number of modes. More precisely, we only consider the two following kind of modes.

- **Spatial mode.** The photon's spatial mode corresponds to the optical path or waveguide [7] in which it propagates. In this thesis, the spatial modes will be indexed by the integers [8], indexing the numbering of the path. The creation operators corresponding to the i^{th} spatial mode will be noted $\hat{a}^\dagger_i$. We remove all the annotations in the ket representation, so that for instance:

$$|1,0,1\rangle = \hat{a}^\dagger_1\,\hat{a}^\dagger_3\,|0,0,0\rangle$$

- **Polarization mode.** As before, we consider the two orthogonal polarization V and H. The creation operator of the polarization mode $|V\rangle$ (resp. $|H\rangle$) in the i^{th} spatial mode will be noted $\hat{a}^\dagger_{V_i}$ (resp. $\hat{a}^\dagger_{H_i}$). For instance, we have:

$$|V,0,H\rangle = \hat{a}^\dagger_{V_1}\,\hat{a}^\dagger_{H_3}\,|0,0,0\rangle$$

Remark I.24 (Generic notation). As defined in Section I.2.4, we also use the notation $\hat{a}^\dagger_i$ (resp. $\hat{a}_i$) to note the creation (resp. annihilation) operators for the i^{th} mode, whatever its type. Therefore, if the context does not specify it, it can refer to other modes than the position.

I.3.2 Linear optical evolution

Now, we aim to formally define what linear optics is by first specifying the form of the Hamiltonian [9] (Section I.3.2.A.), before showing that we can focus solely on the transformation of creation operators (Section I.3.2.B.).

A. Hamiltonian description of linear optics

Definition I.25 (Linear optics)**.** Linear optics [Kok+07] refers to the set of photonic state transformations where the Hamiltonian [10]is bilinear in creation and annihilation operators of same wave frequency:

$$\hat{H}_{\mathsf{LO}} = \sum_{k,\ell} h_{k,\ell}\,\hat{a}^\dagger_k\,\hat{a}_\ell \tag{I.7}$$

where $h_{k,\ell} = h^\dagger_{\ell,k}$ because $\hat{H}$ is Hermitian. From now on, we only consider a unique frequency. This leads to the following unitary transformation [11]:

$$\hat{U} = \exp\Big(-i\hat{H}_{\mathsf{LO}}\Big) \tag{I.8}$$

7. Waveguides are physical systems that can restrict the possible modes of the quantized electromagnetic to one.

8. This thesis focuses only on discrete-variable linear optics. Note that there are modes indexed by continuous parameters, such as those found in free wave-propagation.

9. We recall the reader that the Hamiltonian is one standard way to describe the evolution of quantum systems, cf. Postulate III.

Remark I.26 (Setting $\hbar$ to 1). As it is frequently done in mathematical physics and quantum field theory, and to avoid tracking constants in all subsequent formulae, we set $\hbar = 1$. Formally, this corresponds to a change of units so that the constant is absorbed and reduced to 1. Note that this can complicate conversions into measurable quantities, which is not the scope of this thesis.

Remark I.27 (Finite case). In the finite case with m modes, we have $\hat{H} = X^\dagger H X$ where H is a $m \times m$ Hermitian matrix and $X = (\hat{a}_1, \dots, \hat{a}_m)^T$.

Proposition I.28 (Conservation of the photon number)**.** *A linear optical transformation always conserves the total number of photons.*

Proof. Let $\hat{N} = \sum_i \hat{n}_i$ be the observable of the total number of photons, where we recall that $\hat{n}_i = \hat{a}_i^\dagger \hat{a}_i = \hat{a}_i \hat{a}_i^\dagger - \mathbb{I}$ and that annihilation and creation operators acting on different modes commute (Section I.2.3). For every $k \neq \ell$, we have:

$$\hat{a}_k^\dagger \hat{a}_\ell \hat{n}_k = \hat{a}_k^\dagger \hat{n}_k \hat{a}_\ell = \hat{a}_k^\dagger (\hat{a}_k^\dagger \hat{a}_k) \hat{a}_\ell = \hat{a}_k^\dagger (\hat{a}_k \hat{a}_k^\dagger - \mathbb{I}) \hat{a}_\ell = (\hat{a}_k^\dagger \hat{a}_k - \mathbb{I}) \hat{a}_k^\dagger \hat{a}_\ell = (\hat{n}_k - \mathbb{I}) \hat{a}_k^\dagger \hat{a}_\ell$$

Following that proof and with similar ones, we have:

- $\forall k \neq \ell, (\hat{a}_k^\dagger \hat{a}_\ell)\hat{n}_k = (\hat{n}_k - \mathbb{I})(\hat{a}_k^\dagger \hat{a}_\ell)$.
- $\forall k \neq \ell, (\hat{a}_k^\dagger \hat{a}_\ell)\hat{n}_\ell = (\hat{n}_\ell + \mathbb{I})(\hat{a}_k^\dagger \hat{a}_\ell)$.
- $\forall k \neq \ell, (\hat{a}_k^\dagger \hat{a}_\ell)(\hat{n}_k + \hat{n}_\ell) = (\hat{n}_k + \hat{n}_\ell)(\hat{a}_k^\dagger \hat{a}_\ell)$.
- $\forall k, \forall \ell, \forall j : (j \neq k \wedge j \neq \ell), (\hat{a}_k^\dagger \hat{a}_\ell)\hat{n}_i = \hat{n}_i(\hat{a}_k^\dagger \hat{a}_\ell)$.

The last two relations imply:

$$\hat{N}\hat{H}_{LO} = \left(\sum_i \hat{n}_i\right)\left(\sum_{k,\ell} h_{k,\ell}\, \hat{a}_k^\dagger \hat{a}_\ell\right) = \left(\sum_{k,\ell} h_{k,\ell}\, \hat{a}_k^\dagger \hat{a}_\ell\right)\left(\sum_i \hat{n}_i\right) = \hat{H}_{LO}\hat{N}$$

hence $[\hat{N}, \hat{H}_{LO}] = 0$. As $\hat{U} = \exp\left(\alpha \hat{H}_{LO}\right)$, we also have $[\hat{N}, \hat{U}] = 0$. Therefore $\hat{N}$ remains unchanged:

$$\hat{N} \to \hat{U}^\dagger \hat{N} \hat{U} = \hat{U}^\dagger \hat{U} \hat{N} = \hat{N}$$

□

Remark I.29 (Bogoliubov transformations). Physically, we can implement more general transformations that include quadratic terms, such as $(\hat{a}^\dagger)^2$ and $(\hat{a}^\dagger)^2$, in the Hamiltonian. These quadratic terms, known as squeezing, are nonlinear and do not necessarily conserve the number of photons.

B. Description of linear optics using only creation operators

We now present the following key theorem, which allows us to solely reason with creation operators from now on.

Theorem I.30 (Linear optical unitary transformation of creation operators)**.** *Linear optics refers*

11. Technically, the Hamiltonian is often written as $\hat{H}_{\text{LO}} = \sum_{k \neq \ell} h_{k,\ell}\, \hat{a}_k^\dagger \hat{a}_\ell + \sum_k h_{k,k}(\hat{a}_k^\dagger \hat{a}_k + \frac{1}{2})$. As a matter of convention, we can omit the $\frac{1}{2}$ term to simplify the expressions. Note that this term corresponds to the vacuum energy, which we arbitrary set to zero. More details are given in [Lit20].

to the set of unitary transformations of the creation operators:

$$\hat{a}_j^\dagger \mapsto \sum_i u_{i,j}\, \hat{a}_i^\dagger$$

Proof. With the Heisenberg picture, we know that $\hat{a}_j^\dagger \mapsto e^{i\hat{H}_{\mathsf{LO}}}\, \hat{a}_j^\dagger\, e^{-i\hat{H}_{\mathsf{LO}}}$. For any A, B, we can show the following:

$$e^A B e^{-A} = B + [A,B] + \frac{1}{2!}[A,[A,B]] + \frac{1}{3!}[A[A[A,B]]] + \dots$$

where we recall the definition of the commutator $[X,Y] = XY - YX$. Our goal is to show that for $A = i\hat{H}_{\mathsf{LO}}$ and $B = \hat{a}_j^\dagger$, each term of the serie only contains creations operators. The first term is calculated as follows:

$$[i\hat{H}_{\mathsf{LO}}, \hat{a}_j^\dagger] = i\sum_{k,\ell} h_{k,\ell}[\hat{a}_k^\dagger\, \hat{a}_\ell, \hat{a}_j^\dagger] = \sum_{k,\ell} h_{k,\ell}\, \hat{a}_k^\dagger [\hat{a}_\ell, \hat{a}_j^\dagger] = \sum_{k,\ell} h_{k,\ell}\, \hat{a}_k^\dagger\, \delta_{j,\ell} = \sum_k h_{k,j}\, \hat{a}_k^\dagger$$

. We can give an explicit formula for each coefficient. For instance, for the second-order and third-order term we have:

$$\begin{aligned}
[i\hat{H}_{\mathsf{LO}}, [i\hat{H}_{\mathsf{LO}}, \hat{a}_j^\dagger]] &= i^2[\hat{H}_{\mathsf{LO}}, \sum_k h_{k,j}\, \hat{a}_k^\dagger] \\
&= i^2 \sum_k h_{k,j}[\hat{H}_{\mathsf{LO}}, \hat{a}_k^\dagger] \\
&= i^2 \sum_{k,p} h_{p,k} h_{k,j}\, \hat{a}_p^\dagger
\end{aligned}$$

$$\begin{aligned}
[i\hat{H}_{\mathsf{LO}}, [i\hat{H}_{\mathsf{LO}}, [i\hat{H}_{\mathsf{LO}}, \hat{a}_j^\dagger]]] &= \cdots \\
&= i^3 \sum_{k,p,q} h_{q,p} h_{p,k} h_{k,j}\, \hat{a}_q^\dagger
\end{aligned}$$

Note that each term is only a linear sum of creation operator, therefore leading to:

$$\hat{a}_t^\dagger \mapsto \sum_t u_{s,t}\, \hat{a}_s^\dagger$$

where

$$u_{s,t} = 1 + ih_{s,t} + \frac{i^2}{2!}\sum_k h_{s,k}h_{k,t} + \frac{i^3}{3!}\sum_{k,p} h_{s,p}h_{p,k}h_{k,t} + \dots$$

is well-defined. In finite dimension, which is the scope of this thesis, $\hat{H}_{\mathsf{LO}}$ can be represented by a matrix H. We can express $u_{s,t}$ as the (s,t) coefficient of the matrix e^{iH} , i.e.

$$u_{s,t} = \left(e^{iH}\right)_{s,t}$$

Reciprocally, we can show that any unitary transformation of the creation operators is consistent with Definition I.25 and conserves the Hamiltonian bilinearity.

$$\hat{H}_{\mathsf{LO}} = \sum_{k,\ell} h_{k,\ell}\, \hat{a}_k^\dagger\, \hat{a}_\ell \mapsto \sum_{k,\ell} h_{k,\ell} \left(\sum_i u_{i,k}\, \hat{a}_i^\dagger\right)\left(\sum_i u_{\ell,i}^*\, \hat{a}_i\right) = \sum_{k,\ell} h'_{k,\ell}\, \hat{a}_k^\dagger\, \hat{a}_\ell$$

□

I.3.3 Linear optical circuits: basic components

We describe the four linear optical components used in this thesis, which form the building blocks of linear optical circuits.

A. Phase shifter

Light is characterized by several properties, including its wavelength and phase, see Section I.3.1.A.. When propagating through a medium, the speed of light changes. This speed change implies a change in the wave's phase.

Phase shifters are devices that change the phase of a wave by either making the path longer [Don+22] or changing its refractive index. The latter can be achieved by applying heat or voltage, altering the refractive index through thermal [Har+14] or electro-optic effects [Li+20].

Definition I.31 (Phase shifter)**.** A phase shifter of phase φ is one-spatial-mode component adding a phase φ to the quantum state:

$$|1\rangle \mapsto e^{i\varphi} |1\rangle$$

The diagrammatic representation of the phase shifter is —[φ]—.

Operators The creation operator $\hat{a}^\dagger$ of the spatial mode containing the phase shifter is transformed as follows:

$$\hat{a}^\dagger \mapsto e^{i\varphi}\, \hat{a}^\dagger = e^{i\varphi\hat{n}}\, \hat{a}^\dagger\, e^{-i\varphi\hat{n}}$$

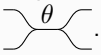
From that, we can deduce $\hat{U}_{PS} = e^{-i\varphi\hat{n}}$ and $\hat{H}_{PS} = \varphi\hat{n}$.

B. Beam splitters

Beam splitters [Zei81; PSM87; CST89; Leo03] are optical device splitting the light into a reflected and transmitted beam. These components are at the core of optics, for they allow interferences [You04; Mic81].

Definition I.32 (Beam splitter)**.** A beam splitter is two-spatial-mode component, where for each input port, the light can be both transmitter and reflected. When there is one photon, the effect on quantum states is:

$$|1,0\rangle \mapsto \cos(\theta)\,|1,0\rangle + i\sin(\theta)\,|0,1\rangle$$
$$|0,1\rangle \mapsto i\sin(\theta)\,|1,0\rangle + \cos(\theta)\,|0,1\rangle$$

where $|\cos(\theta)|^2 = R$ and $|\sin(\theta)|^2 = T$ are respectively the reflectivity and transmitivity. We say a beam splitter is balanced when $R = T = 1/2$. The diagrammatic representation of a beam splitter is 

.

Remark I.33 (Convention of beam splitter). There exist different conventions for the matrix of the beam splitter. Indeed, as soon as the matrix is of the form:

$$e^{i\phi_0}\begin{pmatrix} \cos(\theta)e^{i\phi_R} & \sin(\theta)e^{-i\phi_T} \\ \sin(\theta)e^{i\phi_T} & -\cos(\theta)e^{-i\phi_R} \end{pmatrix}$$

the matrix is a valid representation. The one we chose is so-called Loudon's convention [Lou00]. We chose this convention as the matrix is both symmetric and a rotation.

Operators The creation operators $\hat{a}_1^\dagger$ and $\hat{a}_2^\dagger$ of the two spatial modes containing the beam splitter are transformed as follows:

$$\begin{pmatrix} \hat{a}_1^\dagger \\ \hat{a}_2^\dagger \end{pmatrix} \mapsto \begin{pmatrix} \cos(\theta) & i\sin(\theta) \\ i\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} \hat{a}_1^\dagger \\ \hat{a}_2^\dagger \end{pmatrix} = e^{i\hat{H}_{BS}} \begin{pmatrix} \hat{a}_1^\dagger \\ \hat{a}_2^\dagger \end{pmatrix} e^{-i\hat{H}_{BS}}$$

where $\hat{H}_{BS} = \theta(\hat{a}_1^\dagger\, \hat{a}_2 + \hat{a}_1\, \hat{a}_2^\dagger)$.

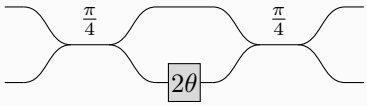
Example I.34 (Hong-Ou-Mandel effect)**.** When two seperate indistinguishable photons enter a perfectly balanced beam splitter, there is no chance for them to remain seperated. Instead, both photons either exit through the first port or the second port, each with a probability of one-half. This called the Hong-Ou-Mandel effect (HOM) [HOM87]. It can be shown as follows, by taking $\theta = \frac{\pi}{4}$:

$$|1,1\rangle = \hat{a}_1^\dagger \hat{a}_2^\dagger |0,0\rangle \mapsto \frac{1}{2}(\hat{a}_1^\dagger + i\,\hat{a}_2^\dagger)(i\,\hat{a}_1^\dagger + \hat{a}_2^\dagger)|0,0\rangle = \frac{i}{2}(\hat{a}_1^\dagger)^2(\hat{a}_2^\dagger)^2|0,0\rangle = \frac{i}{\sqrt{2}}(|2,0\rangle + |0,2\rangle)$$

This property is referred to as bunching [Spa+13].

In practice, a parameterized beam splitter is often implemented by balanced beam splitters and one parameterized phase shifter in the middle. This crucial setup is the Mach-Zehnder interferometer (MZI) [Zeh91; Mac92].

Definition I.35 (Mach-Zehnder interferometer)**.** A MZI is the optical setup constituted of two balanced beam splitters and one parameterized phase shifter in the middle. A parameterized beam splitter of reflectivity $R = |\cos(\theta)|^2$ and transmitivity $T = |\sin(\theta)|^2$ can be physically implemented with:



From now on and for simplicity, we omit the term parameterized and only use beam splitter.

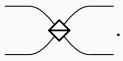
Remark I.36. Note that two beams interfere when they are identical, as it is implicitly described by the creation operators which don't include other modes than the positions. For instance, two beams of opposing polarizations act independently and don't interfere [AF19].

C. Polarizing beam splitter

In this thesis, we also study components that act on the polarization degrees of freedom of the photons. Let $|V\rangle, |H\rangle$ be the two orthogonal linearly polarized states. The state of one photon on polarization H in the 3^{rd} (out of 5) wire is noted: $|0,0,H,0,0\rangle$. The creation and annihilation operators of the i^{th} position with the vertical (resp. horizontal) polarization are respectively noted $\hat{a}_{V_i}^\dagger$ and $\hat{a}_{V_i}$ (resp. $\hat{a}_{H_i}^\dagger$ and $\hat{a}_{H_i}$).

Definition I.37 (Polarizing beam splitter)**.** A polarizing beam splitter (PBS) is a two-spatial-mode component swaping the horizontal polarization of the quantum state:

$$(\alpha|V\rangle + \beta|H\rangle)(\gamma|V\rangle + \delta|H\rangle) \mapsto (\alpha|V\rangle + \delta|H\rangle)(\gamma|V\rangle + \beta|H\rangle)$$

The diagrammatic representation of the PBS is  .

Operators The creation operators are transformed as follows:

$$\begin{pmatrix} \hat{a}_{V_1}^\dagger \\ \hat{a}_{H_1}^\dagger \\ \hat{a}_{V_2}^\dagger \\ \hat{a}_{H_2}^\dagger \end{pmatrix} \mapsto \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{a}_{V_1}^\dagger \\ \hat{a}_{H_1}^\dagger \\ \hat{a}_{V_2}^\dagger \\ \hat{a}_{H_2}^\dagger \end{pmatrix}$$

Putting in the Fock state basis, we have:

$$\hat{U}_{\mathsf{PBS}} = \begin{array}{c} \\ \langle H,0| \\ \langle V,0| \\ \langle 0,H| \\ \langle 0,V| \end{array} \begin{array}{c} \begin{array}{cccc} |V,0\rangle & |H,0\rangle & |0,V\rangle & |0,H\rangle \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{array}. \tag{I.9}$$

D. Wave plate

Definition I.38 (Wave plate)**.** A wave plate is one-spatial-mode component rotating the polarizations of the quantum state:

$$\begin{aligned} |V\rangle &\mapsto \cos(\theta)\,|V\rangle + i\sin(\theta)\,|H\rangle \\ |H\rangle &\mapsto i\sin(\theta)\,|V\rangle + \cos(\theta)\,|H\rangle \end{aligned}.$$

The diagrammatic representation of the wave plate is —▯θ—.

Operators The creation operators are transformed as follows:

$$\begin{pmatrix} \hat{a}_V^\dagger \\ \hat{a}_H^\dagger \end{pmatrix} \mapsto \begin{pmatrix} \cos(\theta) & i\sin(\theta) \\ i\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} \hat{a}_V^\dagger \\ \hat{a}_H^\dagger \end{pmatrix}$$

Remark I.39 (Convention for wave plate). Usually, the matrix for describing the effect of a wave plate is different from the one we chose. Indeed, it is usually preferred to describe the wave plate [CB21] as:

$$\widehat{U}_{\mathsf{WP}}(\delta,\xi) = \begin{array}{c} \\ \langle H| \\ \langle V| \end{array} \begin{array}{c} \begin{array}{cc} |H\rangle & |V\rangle \end{array} \\ \begin{pmatrix} i\sin(\delta)\cos(2\xi) + \cos(\delta) & i\sin(\delta)\sin(2\xi) \\ i\sin(\delta)\sin(2\xi) & -i\sin(\delta)\cos(2\xi) + \cos(\delta) \end{pmatrix} \end{array} \tag{I.10}$$

where δ is a parameter proportional to the thickness of the waveplate and ξ represents the angle of the waveplate's optical axis in the $\{|V\rangle, |H\rangle\}$ plane. We often distinguish the following two kinds of wave plate.

- Half-wave plates, which rotate linear polarisations in the $\{|V\rangle, |H\rangle\}$ plane. It is for $\delta = \pi/2$.
- Quarter-wave plates, which convert circular polarisations to linear ones (e.g. $|L\rangle = (|V\rangle + i\,|H\rangle)/\sqrt{2}$) and vice-versa. It is for $\delta = \pi/4$.

Note that up to a global phase, any 2×2 unitary can be implemented with two quarter-wave plates and one half-wave plate [Sha+07]. Note that the matrix we chose for the wave plate is the matrix of the half-wave plate up to phases.

E. Linear optical circuits

Definition I.40 (Linear optical circuits (Informal))**.** A linear optical circuit is constructed through the vertical and horizontal composition of basic linear optical components, connected by physical wires, as depicted in Figure I.3.

Note that, similar to the informal definition of quantum circuits (Definition I.10), we can employ the formalism introduced in Section I.4 to formally define linear optical circuits. This formalism is one of the contributions of this thesis and is further detailed in Chapter III.

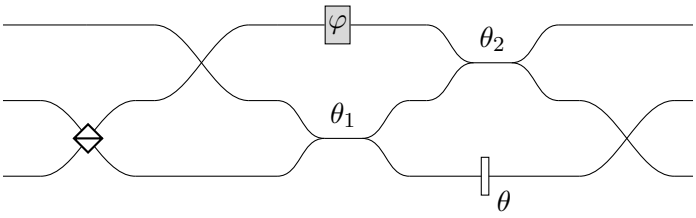


Figure I.3 – Example of a linear optical circuit.

I.3.4 Universality of spatial-mode linear optics

In linear optics, it is very usual to only consider the positions of the photons as degrees of freedom. We refer it as spatial-mode linear optics.

Definition I.41 (Spatial-mode linear optics)**.** Spatial-mode linear optic refers to linear optical transformations where annihilation and creation involved in the Hamiltonian are only acting on the spatial modes. In the scope of this thesis, the number of spatial modes is always finite.

It is also very usual to define spatial-mode linear optical operations as the set of interferometers than can be build from phase shifters and beam splitters. This is due to the following equivalence.

Theorem I.42 (Universality of phase shifters and beam splitters)**.** *Any unitary transformation of m spatial-mode creation operators can be implemented using at most $\frac{m(m+1)}{2}$ phase shifters and $\frac{m(m-1)}{2}$ beam splitters. Equivalently, spatial-mode linear optics consists of all transformations that can be realized with these two components.*

Proof. Similarly to using Householder reflections or Givens rotations, we can prove the following lemma by induction [Rec+94] (see [Cle+16] for another proof).

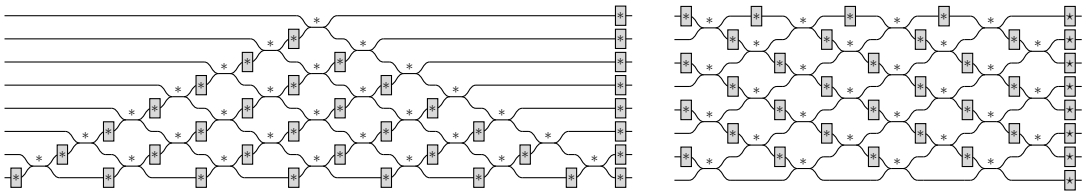
Lemma I.43 (Unitary decomposition into phases and rotations)**.** *Any $m \times m$ unitary matrix can be decomposed into the composition of $\frac{m(m+1)}{2}$ phase gates ($P_k(\varphi)$) and $\frac{m(m-1)}{2}$ 2×2 rotation gates ($BS_\ell(\theta)$) where:*

$$P_k(\varphi) \coloneqq \begin{pmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & \cdots & e^{i\varphi} & \cdots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \quad BS_\ell(\theta) \coloneqq \begin{pmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & \cdots & \cos(\theta) & i\sin(\theta) & \cdots \\ 0 & 0 & \cdots & i\sin(\theta) & \cos(\theta) & \cdots \\ \vdots & \vdots & & \vdots & & \ddots \\ 0 & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix}$$

By noticing P_k and BS_ℓ can be respectively implemented by a phase shifter on the k^{th} wire and by a beam splitter on the $(\ell, \ell+1)^{\text{th}}$ wires, phase shifters and beam splitters can thus implement any unitary on the spatial modes. Therefore, the proof of this theorem directly follows from Theorem I.30, as performing any unitary on the creation operators is the same as any linear optical operation. □

The first universal linear optical circuit has been introduced in [Rec+94], providing a triangular arrangement. More recently, a rectangular decomposition has been proposed [Cle+16]. In practice, the rectangular scheme is widely more used than its triangular counterpart as it is more compact and

possesses more symmetries. These two different shapes are illustrated in Figure I.4. Additionally, it is important to recall that parameterized beam splitters are implemented by MZIs. The MZI in [Cle+16] has one phase outside and one inside the two balanced beam splitters. A last improvement, putting both phases inside and therefore compacting and symmetrizing a bit more, has been developed in [BW21].



(a) Triangular form [Rec+94]. (b) Rectangular form [Cle+16].

Figure I.4 – Universal linear optical circuits. They can implement any unitary on the spatial modes.

Remark I.44 (Non-deterministic gates). Let us note that the unitary is arbitrary for the modes, but not an arbitrary unitary on the Fock space. For instance, we can prove there is no linear optical transformation mapping $|1,0,1,0\rangle = \hat{a}_1^\dagger \hat{a}_3^\dagger |0,0,0,0\rangle$ to the Bell state $\frac{1}{\sqrt{2}}(|1,0,1,0\rangle + |0,1,0,1\rangle) = \frac{1}{\sqrt{2}}(\hat{a}_1^\dagger \hat{a}_3^\dagger + \hat{a}_2^\dagger \hat{a}_4^\dagger)|0,0,0,0\rangle$. Indeed, no unitary operation on the spatial modes can perform the following:

$$\hat{a}_1^\dagger \hat{a}_3^\dagger \mapsto \left(\sum_k u_{k,1}\, \hat{a}_k^\dagger\right)\left(\sum_\ell u_{\ell,3}\, \hat{a}_\ell^\dagger\right) \neq \frac{1}{\sqrt{2}}(\hat{a}_1^\dagger \hat{a}_3^\dagger + \hat{a}_2^\dagger \hat{a}_4^\dagger).$$

Interestingly, such operations can be performed by adding nonlinearity. One practical nonlinearity effect is the measurement. Thus, there exist linear optical schemes that can, together with auxiliary resources and measurements looking for specific results, prepare Bell states and quantum logic gates [KLM01; Kni02; Ral+02; FSK21]. As a specific pattern in the outcome of a probabilistic measurement is required, such operations are probabilistic or non-deterministic.

I.3.5 Simulation of discrete-variable quantum linear optics

When considering simulation, it is important to define the settings of our system. This thesis focuses on the simulation of linear optical quantum circuits with inputs given by finite linear combinations of Fock states, known as the discrete-variable model (Section I.3.5.A.). We then present state-of-the-art algorithms for simulating these noiseless systems, distinguishing between two types of simulation: strong (Section I.3.5.B.) and weak (Section I.3.5.C.), before briefly describing simulation in the noisy regime (Section I.3.5.D.).

A. Discrete-variable model of linear optics

From now on, we only consider simulation of the discrete-variable model of quantum linear optics.

Definition I.45 (Discrete-variable quantum linear optics)**.** Discrete-variable quantum linear optics refers to the model where the system can be described with a finite number of Fock states.

Remark I.46 (Continuous-variable model). The discrete-variable (DV) model is often contrasted with the continuous-variable (CV) model, where the system is described by any vector in the Fock space, including infinite superposition of Fock states. For further reading, refer to [Cha20] for a more detailed

overview of continuous variables and optics, and to [MA23] for approximations of DV simulations using CV settings.

Additionally, given a fixed number of modes and number a photon, it is relevant to know the number of different output states.

Proposition I.47 (Size of the output space)**.** *There are* $\binom{m+n-1}{n}$ *configurations for distributing* n *indistinguishable photons among* m *modes.*

Proof. This is the same as counting the number of ways to put n indistinguishable balls into m bins [Fel68], which can be counted with the stars and bars formalism. Let $n = 3$ and $m = 4$. The configuration $|1, 0, 2, 0\rangle$ can be modeled with $\star||\star\star|$, where the n stars represent the balls and the $m - 1$ bars separate the bins. Note that there are $m + n - 1$ emplacements and n stars to place, thus there are $\binom{m+n-1}{n}$ different configurations. □

Remark I.48. Note that from the properties of the binomial coefficient, we have $\binom{m+n-1}{n} = \binom{m+n-1}{m-1}$.

B. Strong simulation

A strong simulation aims to exactly compute the output probabilities.

Definition I.49 (Strong simulation)**.** Let $|\psi\rangle$ be an input state entering an m-mode interferometer $\hat{U}$. The strong simulation computes the probabilities of measuring output states $|t\rangle$: $p(t) = |\langle t|\hat{U}|\psi\rangle|^2$.

Remarkably, when both the inputs and outputs are Fock states, the output amplitude is directly related to the matrix permanent.

Proposition I.50 (Single output amplitude)**.** *Let* $|s\rangle = |s_1, \dots, s_m\rangle$ *be a Fock state of* n *photons entering an interferometer of matrix* $U : m \times m$*. The output amplitude corresponding to the state* $|t\rangle$ *is:*

$$\langle t|\hat{U}|s\rangle = \frac{\text{Per}\,(U_{s,t})}{\sqrt{s_1! \dots s_m! t_1! \dots t_m!}} \tag{I.11}$$

where $U_{s,t}$ *is obtained from* U *by repeating* s_j *times its* j^{th} *column and* t_i *times its* i^{th} *row, and where* $\text{Per}\,(M) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} m_{i,\sigma(i)}$ *is the matrix permanent, summing over all the permutations of the* $1, \dots, n$*.*

Proof. This directly follows from the development of:

$$|s_1, \dots, s_m\rangle = \frac{\hat{a}_1^\dagger \dots \hat{a}_m^\dagger}{\sqrt{s_1!, \dots, s_m!}} |0, \dots, 0\rangle \mapsto \frac{\left(\sum_{i_1=1}^{m} u_{i_1,1}\, \hat{a}_{i_1}^\dagger\right) \cdots \left(\sum_{i_m=1}^{m} u_{i_m,1}\, \hat{a}_{i_m}^\dagger\right)}{\sqrt{s_1!, \dots, s_m!}} |0, \dots, 0\rangle$$

and by the definition of the matrix permanent [Sch04; AA11]. □

Proposition I.51 (Complexity of strong simulation)**.** *Computing one single output probability of a linear optical circuit is #P-hard in the worst case.*

Proof. From [Val79], it is known that computing the permanent of a generic $(0,1)$-matrix is #P-hard. Let A be an $n \times n$ $(0,1)$-matrix, and define $B = \frac{A}{\|A\|}$ where $\|A\|$ denotes the spectral norm of A. We can embed B into a $2n \times 2n$ unitary matrix U using a unitary dilation theorem [Sch55], as follows:

$$U = \begin{pmatrix} B & \sqrt{I - BB^\dagger} \\ \sqrt{I - B^\dagger B} & -B^\dagger \end{pmatrix}.$$

From Theorem I.42, we know that any unitary transformation can be implemented in spatial-mode linear optics. Therefore, we can physically realize $\hat{U}$. Computing the following output amplitude is therefore #P-hard:

$$\langle 1 \dots 1 0 \dots 0 | \hat{U} | 1 \dots 1 0 \dots 0 \rangle = \mathrm{Per}(B) = \frac{\mathrm{Per}(A)}{\|A\|^n}.$$

□

The state-of-the-art algorithms to compute or approximate the permanents of such matrices are detailed in Section I.3.6.

C. Weak simulation

The weak simulation (or sampling) aims to produce one sample from the probability distribution.

Definition I.52 (Weak simulation)**.** Let $|\psi\rangle$ be an input state entering an m-mode interferometer $\hat{U}$. The weak simulation randomly outputs the output state $|t\rangle$ with a probability $|\langle t| \hat{U} |\psi\rangle|^2$.

The weak simulation of linear optical experiments has been mainly studied for the case of a Fock state as an input. This case is known as the Boson Sampling problem, introduced in [AA11].

Definition I.53 (Boson Sampling)**.** Boson Sampling is the task of sampling from a spatial-mode linear optical circuit with one Fock state as input.

Remark I.54 (Convention for the definition). Even though the original paper [AA11] was more flexible in its definition, referring to sampling from the output distribution of a boson computer, the convention of considering Boson Sampling as involving only Fock states as inputs is commonly used. In some contexts, the convention is even stricter, considering only inputs of the form $|1 \dots 1\rangle$.

Remark I.55 (Other Boson Sampling settings). When the input consists of a squeezed state, a specific continuous-variable state, the sampling task is referred to as Gaussian Boson Sampling [Ham+17]. Additionally, there is nonlinear Boson Sampling, which arises when the circuit includes nonlinear elements [Spa+23].

Proposition I.56 (Complexity of Boson Sampling)**.** *Boson Sampling is not efficiently simulable by a classical computer unless the polynomial hierarchy collapses to the third level.*

Proof. This was proved in the seminal paper by Aaronson and Arkhipov [AA11]. Note that under some reasonable conjectures, they also prove that approximate sampling is also hard. □

The naive classical algorithm for performing Boson Sampling would be to compute all the probabilities for the different outputs, compute the cumulative distribution function F, and then sample using inverse transform sampling: uniformly sample $u \in [0,1]$, and return the smallest t such that $F(t) \geq u$. This algorithm successfully outputs a sample t with the correct probability.

However, for a circuit of n photons over m modes, this would require computing and storing $\binom{n+m-1}{n}$ probabilities, each requiring the computation of a matrix permanent, which could take $O(n2^n)$ operations in the worst case.

Much faster sampling algorithms have been proposed by Clifford and Clifford [CC18; CC24].

Proposition I.57 (Classical simulation of Boson Sampling)**.** *A sample from a Boson Sampling experiment with n photons can be obtained in $O(n2^n)$ classical operations.*

Proof. This directly follows from the algorithm of [CC18]. Note this algorithm has further been improved by the same authors in [CC24], taking into account the repetitions of the matrix rows due to the bunching of photons. For pedagogical purposes, we summarize the main ideas of [CC18] here [12].

The algorithm uses the probability chain rule to sequentially sample photon by photon. To lighten the notation, we respectively note t^k the output sampled at the step k, containing $1 \leq k \leq n$ photons. We note $p(t^k|t^{k-1}, s^k)$ the conditional probability to sample t^k conditioned on t^{k-1} — we know the position of $k-1$ out of the k photons — and conditioned on the input s^k. In this work, we have $s^n = |1, 1, 1, \ldots, 1, 0, \ldots, 0\rangle_m$.
The main idea is to avoid directly sampling from $p(t^k)$, by noticing we can equivalently sample from $p(t^k|s^k)$, where the inputs s^k are randomly chosen, cf (A) and (B). Then, to sample from $p(t^k|s^k)$, we use the chain rule to sample from $p(t^k|t^{k-1}, s^k)$ instead (C), which is easier to compute (D). The technical points can be summarized as follows [13]:

(A) We have $p(t^k) \propto \sum_{s^k} p(t^k|s^k) \propto \mathbb{E}_{s^k}\left\{p(t^k|s^k)\right\}$, with $p(t^k|s^k) = |\left\langle t^k\right| \hat{U} \left|s^k\right\rangle|^2$ and the expectation value summing uniformly over the $\binom{n}{k}$ states s^k (cf Lemma 1 and 3 of [CC18] [14]).

(B) Instead of sampling from $p(t^k)$, we know now it is equivalent to sample from $\mathbb{E}_{|s\rangle^k \leq |s\rangle^n}\left\{p(t^k|s^k)\right\}$. As the s^k are uniformly distributed, we can first uniformly sample the order $\left\{|s\rangle^k, 1 \leq k \leq n\right\}$. Now, we only need to sample from the $p(t^k|s^k)$ distribution.

(C) For a fixed order $s^k, 1 \leq k \leq n$ randomly sampled at the step (B), we have $p(t^n|s^n) = p(t^1|s^1)p(t^2|t^1, s^2)\ldots p(t^n|t^{n-1}, s^n)$ with the chain rule. At the step k, given the previous sample t^{k-1}, we sample the output t^k from $p(t^k|t^{k-1}, s^k)$.

(D) Given $t^{k-1} = \langle t_1, \ldots, t_m|$, there are only m different possibilities for t^k, consequence of where the k^{th} photon has arrived. To sample from $p(t^k|t^{k-1}, s^k)$, we only need to compute the following m probabilities:

$$\left\{|\left\langle t_1, \ldots, t_i + 1, \ldots, t_m\right| \hat{U} \left|s\right\rangle|^2, 1 \leq i \leq m\right\}$$

.

Clifford and Clifford provided an efficient way to compute the m probabilities at step k in $O(k2^k)$, leading to a complexity of $O\left(\sum_{k=1}^{n} k2^k\right) = O(n2^n)$. □

Note that state-of-the-art algorithms for Boson Sampling [CC18; CC24] are currently limited to single Fock states as inputs. One contribution of this thesis is to introduce a more general algorithm for simulating Boson Sampling with a broader range of DV input states, as described in Section II.4.

D. Simulation of noisy linear optical circuits

Linear optical experiments, like any quantum system, are highly sensitive to noise, i.e., physical errors that cause the desired quantum state to be lost. Accounting for these errors is crucial, as it can

12. See [NG24] for another introduction to Clifford and Clifford algorithm.
13. We use $\propto$ to lighten the formula and disposing of factorials coefficients depending on k, n.
14. In [CC18], the photon-to-mode encoding is used, where the positions of the i^{th} photon is described by $r_i \in [m]$. By using their notations, we have $p(r^k|r_1, \ldots, r_{k-1}) = p(t^k|t^{k-1})$. They use a parameter $\boldsymbol{\alpha}$ to parameterize the permutation of the columns, representing the order of the input photons. Considering a random permutation $\alpha_1, \ldots, \alpha_n$ is therefore equivalent to consider a random order s^k of the inputs.

be shown that a sufficient amount of noise leads to efficient classical weak simulations [RRC16; Shc19; Qi+20]. In practice, linear optical experiments are subject to the following errors:

- **Photon loss.** Photons can be *lost* at many parts of the system: at the source — characterized by its brightness —, through interactions with optical components, or at the detector.
- **Distinguishability.** To exhibit crucial quantum effects such as interferences, photons must be indistinguishable. Differences in the polarization and frequencies, for instance, lead to an absence of interference. The distinguishability of two photons is often characterized by a HOM experiment (see Example I.34), i.e. sending the two photons in a balanced beam splitter. Since two perfectly indistinguishable photons always bunch together, the probability of observing them in separate output modes directly reflects how different they are.
- **Multi-photon emission.** At the source, there is a nonzero probability of emitting more than one photon. Although this probability is very small, it is sometimes considered that sources emit two photons [Lou00].

This thesis mainly focuses on simulations in the perfect case. For a broader state of the art about noisy simulations, see [Ren+18; OB18; GRS19; BO20; SG21; Bul+24].

I.3.6 Computation and approximation of the matrix permanent

As described in Proposition I.50, all single-output amplitudes can be computed using the matrix permanent of a certain matrix. In this section, we present state-of-the-art algorithms for both computing (Section I.3.6.A.) and approximating (Section I.3.6.B.) matrix permanents.

A. Exact computation of the permanent

Since we know from [Val79] that computing the permanent of matrix in $\{0,1\}^{n\times n}$ is #P-complete, we can't expect to compute the permanent in poly(n) time. Still, computing the permanent of an $n\times n$ matrix using the $n!$ permutations involved in the sum is intractable. This complexity has been "run down" to $n2^n$.

Proposition I.58 (Computation of the permanent of an arbitrary matrix)**.** *The permanent of an arbitrary $n\times n$ complex matrix can be computed in $O(n2^n)$ complex operations.*

Proof. The first introduced formula in the literature based on the inclusion-exclusion principle, has been proposed by Ryser [Rys63]:

$$\mathrm{Per}\,(A) = (-1)^n \sum_{(e_1,\dots,e_n)\in\{0,1\}^n} (-1)^{e_1+\cdots+e_n} \prod_{i=1}^{n}\sum_{j=1}^{n} e_j a_{ij} \qquad \text{(Ryser's)}$$

As for each of the 2^n set S, there is a product of n terms, where each of the term takes $O(n)$ operations to be computed, the apparent complexity is in $O(n^2 2^n)$. However, by processing the sets S in a Gray code order [NW78], each sum $\sum_j e_j a_{ij}$ only differs of one coefficient from the one computed before, and thus can be computed in $O(1)$ instead of $O(n)$. The complexity of computing a permanent with Ryser's formula is therefore $O(n2^n)$.

With the same complexity, another formula has been proposed by Glynn [Gly10]:

$$\mathrm{Per}\,(A) = \frac{1}{2^{n-1}} \sum_{(\delta_1=1,\delta_2,\dots,\delta_n)\in\{-1,1\}^n} \left(\prod_{k=1}^{n}\delta_k\right) \prod_{i=1}^{n}\sum_{j=1}^{n} \delta_j a_{ij} \qquad \text{(Glynn's)}$$

The link between both formulae has been done in [Gly13], by looking at polarization identities of symmetric tensors. Up to today, no formula with a better complexity than $O(n2^n)$ has been found [15].

□

Furthermore, we can notice from Equation (I.11) that the matrices of interest have repeated columns and rows. One can use this property to derive a more efficient formula.

Proposition I.59 (Computation of the permanent of a matrix with repeated rows)**.** *Let A be a $n \times n$ matrix where there are k distinct rows, where the i^{th} is repeated t_i. Then the permanent of A can be computed in*

$$O\left(\frac{\prod_{i=1}^{m}(t_i+1)}{\min_{t_\ell \neq 0} t_\ell + 1} \times n\right)$$

complex operations.

Proof. We can extend Glynn's formula [Gly10] to exploit the properties of the roots of unity, see Appendix B of [Tic11] or Appendix D of [Shc13]. Let $R_t = \{1, \ldots, e^{\frac{2i\pi t}{t+1}}\}$ be the set of the t-th roots of unity, we have:

$$\text{Per}\,(A) = \frac{\prod_{j=1}^{m} t_j!}{\prod_{j=1}^{m}(t_j+1)} \sum_{r_1 \in R_{t_1+1}} \cdots \sum_{r_m \in R_{t_m+1}} r_1 \ldots r_m \prod_{i=1}^{k}\left(\sum_{j=1}^{n} r_j a_{ij}\right).$$

This formula requires the evaluation of $\prod_{i=1}^{n}(t_i+1)$ product of k elements, each taking $O(n)$ operations to compute. Thus, the total number of operations is

$$O\left(\prod_{i=1}^{n}(t_i+1) \times nk\right)$$

complex operations. With some extra optimizations [Shc20] this count can be reduced to

$$O\left(\frac{\prod_{i=1}^{m}(t_i+1)}{\min_{t_\ell \neq 0} t_\ell + 1} \times n\right).$$

□

Proposition I.60 (Time complexity for computing the full output distribution with matrix permanents)**.** *The permanent-based method computes the full distribution of an n-photon input state $|s_1, \ldots, s_m\rangle$ in $\Omega\left(\binom{2m+n-1}{n}\right)$ complex operations, i.e. at least $O\left(\binom{2m+n-1}{n}\right)$ operations.*

Proof. Giving the best algorithm for computing a permanent with repeated rows, we need to sum over all the possible outputs states the term $n\frac{\prod_{i=1}^{m}(t_i+1)}{\min_{t_l \neq 0}(t_l+1)}$:

$$\sum_{|t\rangle \in \mathcal{F}_m^n} \frac{\prod_{j=1}^{m}(t_j+1)}{\min_{t_l \neq 0} t_l + 1}$$

for which there is no known analytical formula. To simplify the expression of the sum, we use a lower bound, considering $n \geq 1$ so that $\min_{t_l \neq 0}(t_l+1) \leq n+1 \leq 2n$. We therefore have the following lower bound:

$$\frac{1}{2}\sum_{|t\rangle \in \mathcal{F}_m^n} \prod_{i=1}^{m}(t_i+1) \leq \sum_{|t\rangle \in \mathcal{F}_m^n} n\frac{\prod_{i=1}^{m}(t_i+1)}{\min_{t_l \neq 0}(t_l+1)}$$

15. Note that [Bjö16] gives an algorithm, not a formula, computing the permanent of a $n \times n$ matrix in poly(n)-bit integers in $O(2^{n-\Omega\sqrt{n/\log n}})$.

As shown in the Lemma 2 of [CC24], the sum of the left-hand side is equal to $\binom{2m+n-1}{n}$, giving our lower bound of $\Omega\binom{2m+n-1}{n}$. □

B. Approximation of the permanent

As the permanents need an exponential time to be computed, one may be interested in just knowing an approximate value, if the computed time is shortened and becomes polynomial. We define the two following approximations.

Definition I.61 (Additive approximation)**.** We say that $\widetilde{\text{Per}}(A)$ is an ϵ-additive approximation of $\text{Per}(A)$ if $||\widetilde{\text{Per}}(A) - \text{Per}(A)|| < \epsilon$, where $||.||$ is the largest singular value norm.

Definition I.62 (Multiplicative approximation)**.** We say that $\widetilde{\text{Per}}(A)$ is an ϵ-multiplicative approximation of $\text{Per}(A)$ if $||\widetilde{\text{Per}}(A) - \text{Per}(A)|| < \epsilon||\text{Per}(A)||$, where $||.||$ is the largest singular value norm.

Note that if X, the value we want to approximate, is such that $|X| < c$, then a $\frac{\epsilon}{c}$-multiplicative approximation gives a ϵ-additive approximation. The inverse is not necessarily true, especially when the value $|X|$ is small compared to ϵ.

Contrary to the exact computations, there exists efficient randomized classical algorithms for efficiently approximating some class of permanents. One breakthrough was done by Jerrum, Sinclair and Vigoda [JSV04], who give a fully polynomial-time randomized approximation scheme (FPRAS) to multiplicatively approximate the permanent of matrices with nonnegative entries.

For matrices with arbitrary complex values, or more specifically for submatrices of unitary matrices, there is no FPRAS to date. The most used algorithm known for our case of interest is due to Gurvits [Gur05], which gives an additive approximation for arbitrary matrix. For $\epsilon > 0$ and an $n \times n$ matrix A, Gurvits' algorithm approximate $\text{Per}(A)$ to within an additive error $\pm\epsilon||A||^n$ in $O(\frac{n^2}{\epsilon^2})$ time. If A is a submatrix of a unitary, then $||A|| \leq 1$, and therefore Gurvits' algorithm gives an ϵ-additive approximation of $\text{Per}(A)$. A description of Gurvits' algorithm is provided in [AA11; AH14].

I.4 Graphical language

The simpler and more intuitive a tool is, the more accessible it becomes for understanding and manipulating complex ideas. Prominent and particularly useful models are diagrams, making processes visual and easy to grasp. Such representations are all the more powerful when they are formally defined and can be freely manipulated diagrammatically.

In this section, we present the first emerged graphical languages and when category theory arose for formalizing them (Section I.4.1). Then, we introduce an informal presentation of string diagrams and their key features (Section I.4.2), before formally defining them with category theory (Section I.4.3). In particular, we introduce the notion of PROPs, that is the core framework of the graphical languages introduced in Chapter III of this thesis. Finally, we finish by covering all notions of rewriting theory and logic that help manipulate such graphical languages (Section I.4.4).

I.4.1 Towards a formalization of diagrams

One powerful and widely used representation is diagrams and drawings. From simple geometric problems to more complex notions as with Feynmann diagrams [Fey49], we know they can help to

grasp ideas and solve problems. They are even more appealing when rigorous calculations and proofs can be derived diagrammatically. A significative contribution to the development of such rigorous graphical frameworks is Penrose's work on tensor networks [Pen71b; Pen71a].

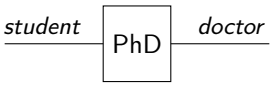
Still, at that time, each graphical language needed its own mathematical formalism. There was no generic framework to quickly introduce the properties of one's graphical languages, nevertheless a framework to unify all graphical languages with similar properties was lacking — which is expected for formalizing physical systems that naturally possess similar structures. It is therefore of great interest to look for a unified and high-level model to define graphical languages, exhibiting the fundamental structure of equivalent processes.

Interestingly, during the same period, a field capturing mathematical structures and their interactions emerged: category theory [Mac65; Mac71]. The feat of using all the abstract tools of category theory to unify and formalize graphical languages has arguably been introduced first by Joyal and Street in [JS91]. More precisely the string diagrams — the graphical framework from which we build our work on — can be seen as monoidal categories [16].

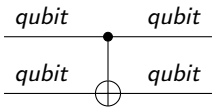
In this thesis, as many other graphical languages [Sel11], we use monoidal categories to formalize our diagrams with optical elements. More particularly we use PROPs, a special kind of monoidal category, that we define in the next section.

I.4.2 String diagrams

String diagrams represent processes, taking some things as inputs and some things as outputs. In this thesis, we choose the convention of having inputs on the left and output on the right. The objects involved in the processes can be of very different natures:

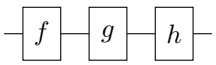


The natures of these objects can be seen as *data types* in computer science. In the previous example, PhD would be the process that takes a student as an input and outputs a doctor. Note that the boxes can take many forms, e.g. the quantum logic gate CNot:



Note that in that example, there is no need to label the wires to indicate their type, as there is no ambiguity. Indeed, for quantum circuits, all the wires represent the same objects: *qubits*. For simplicity and as later all the wires also correspond to one single object, we keep all the wires unlabeled from now on.

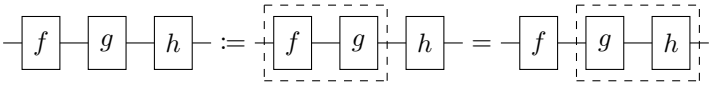
Let us consider generic processes f, g and h. These can be composed in sequence. We note this composition $\circ$. Diagrammatically, it corresponds to a horizontal composition:



(I.12)

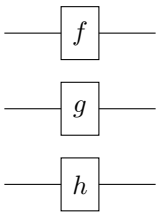
Note that $\circ$ is associative, i.e. $h \circ g \circ f := h \circ (g \circ f) = (h \circ g) \circ f$. Diagrammatically, we have:

16. Note that monoidal categories have been prominent in linking concepts between different fields (physics, topology, logic, etc) [BS11]. More generally, category theory has brought a new insight and new tools for categorizing quantum mechanics, leading to categorical quantum mechanics [AC04; Sel07; AC09] (see [HV19] for an introduction).



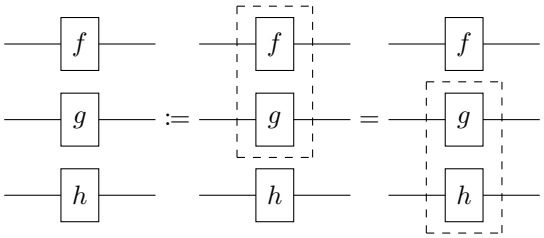
(I.13)

Similarly, these processes can be composed in parallel. We note this composition $\otimes$. We adopt the convention that wires are ordered from top to bottom, with the top wire corresponding to the first input (or output), the next to the second, and so on. Diagrammatically, a vertical composition is:



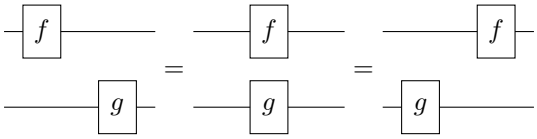
(I.14)

Note that $\otimes$ is associative, i.e. $f \otimes g \otimes h \coloneqq f \otimes (g \otimes h) = (f \otimes g) \otimes h$. Diagrammatically, we have:



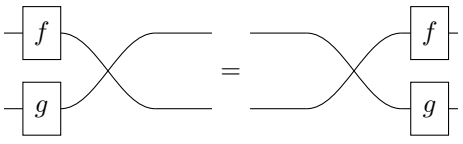
(I.15)

Furthermore, this vertical composition could assume a perfect synchronized of the processes in parallel. Very often in physical processes such as the ones we study, there is no need for such constraints. The order of independent and separate processes do not change the outcome, meaning the following three diagrams represent the same process:

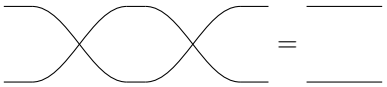


(I.16)

Furthermore, it is very common that the wires can be freely bent. We say that *only the connectivity matters*. Thus, we also have the following equalities:



(I.17)



(I.18)

The goal of the next section is to provide a mathematical framework fit to capture all the previous properties. More precisely, we gradually capture and formalize:

- The associative horizontal composition $\circ$ of processes (Equation (I.12) and I.13). This is captured in the category (Definition I.63).
- The associative vertical composition $\otimes$ of processes (Equation (I.14), I.15 and I.16). This is captured in the strict monoidal category (Definition I.68).

- The displacements of operators through SWAPs (Equation (I.17)). This is captured in the strict braided monoidal category (Definition I.71).
- The simplifications of SWAPs with identities (Equation (I.18)). This is captured in the strict symmetric monoidal category (Definition I.72).
- The wires representing the same objects. This is captured in the PROP (Definition I.74).

Coherence. To assure the category formalism is fitted for the graphical relations we want to describe, we need to be sure that:

- The graphical representation does not express less than the category. If certain equations or transformations hold in the category, the graphical language should be able to express them.
- The graphical representation does not express more than the category. We do not allow for diagrammatic operations or relationships that are not valid in the category.

This alignment between the graphical representation and the underlying category is called coherence and has been demonstrated in [JS91] (see [Sel11] for a survey).

I.4.3 Category theory: straight to the PROP

This section aims to define all the notions that are essential for formalizing the graphical languages that have the properties introduced in Section I.4.2. The goal is to use category theory.

One way of thinking about category theory — that can help to understand the following definitions — is to see category theory as a refinement, a generalization, or *categorification* of ordinary algebra. In the following, we introduce many definitions that can be seen as generalizations of well known structure in algebra. In particular and among other things, we can see sets as small categories (Remark I.64), product category as the categorification of the Cartesian product (Definition I.68), and a monoidal category as categorification of monoids (Definition I.70).

Definition I.63 (Category)**.** A category $\mathcal{C}$ is a collection of objects $\text{ob}(\mathcal{C})$ and a collection of morphisms (or arrows) $\text{mor}(\mathcal{C})$, such that:

- To each morphism f is associated two objects A and B in $\text{ob}(\mathcal{C})$ called the domain and the codomain. We write $f : A \to B$, also noted $A \xrightarrow{f} B$, if A is the domain (or source) and B is the codomain (or target).
- For every pair $f : A \to B$ and $g : B \to C$ in $\text{mor}(\mathcal{C})$, their composite $g \circ f : A \to C$ is also in $\text{mor}(\mathcal{C})$. Furthermore, the binary operation $\circ$ (composition) is associative:

$$h \circ (g \circ f) = (h \circ g) \circ f$$

- For every object A, we have the identity morphism $id_A : A \to A$ in $\text{mor}(\mathcal{C})$, satisfying the following right and left unit laws:

$$\forall f : A \to B, f \circ id_A = f$$
$$\forall g : B \to A, id_A \circ g = g$$

Remark I.64. The term *collections* is used to denote the objects and morphisms of a category. This is to prevent from making any assumption on the nature of this collection, which could be a set, a class, a type, etc. A category where the collections are sets is called a small category. Remarkably, the generalization of the nature of the objects makes possible to consider the categories of all the small category, noted as Cat — an impossible feat for sets due to Russell's paradox.

Definition I.65 (Isomorphism)**.** Let $\mathcal{C}$ be a category, and $A, B \in \mathrm{Ob}(\mathcal{C})$. If $f : A \to B$ and $g : B \to A$ are such that $g \circ f = \mathrm{id}_A$ and $f \circ g = \mathrm{id}_B$, then f and g are *isomorphisms*. We say that g is an inverse of f (and vice versa). Both f and g are called *invertible*, and A and B are said to be *isomorphic*. The inverse of f is unique and can also be written as f^{-1}.

Definition I.66 (Dual of a category)**.** Let $\mathcal{C}$ be a category. The category $\mathcal{C}^{\mathsf{op}}$, called the *dual* or *opposite category* of $\mathcal{C}$, is defined as follows:

- $\mathrm{Ob}(\mathcal{C}^{\mathsf{op}}) = \mathrm{Ob}(\mathcal{C})$.
- If $f : A \to B$ is a morphism in $\mathcal{C}$, then $f^{\mathsf{op}} : B \to A$ is a morphism in $\mathcal{C}^{\mathsf{op}}$.
- The composition is defined such that $g^{\mathsf{op}} \circ f^{\mathsf{op}} := (f \circ g)^{\mathsf{op}}$.

Remark I.67. We don't use the notation C^* as C^* is usually used for C^*-algebras.

As detailed in Section I.4.2, we would like to have a tensor product $\otimes$ of objects. We need to properly define the binary operator $\otimes : \mathsf{ob}(\mathcal{C}) \times \mathsf{ob}(\mathcal{C}) \to \mathsf{ob}(\mathcal{C})$. Before defining the tensor product as a functor, i.e. a morphism between categories, we first define the category containing the objects $A \times B$: the product category.

Definition I.68 (Product category)**.** For two categories $\mathcal{C}$ and $\mathcal{D}$, the product category $C \times D$ is the category satisfying the following.

- The objects are the ordered pair (A, B) with $A \in \mathsf{ob}(\mathcal{C})$ and $B \in \mathsf{ob}(\mathcal{D})$, i.e. $\mathsf{ob}(\mathcal{C} \times \mathcal{D}) = \mathsf{ob}(\mathcal{C}) \times \mathsf{ob}(\mathcal{D})$.
- The morphisms are the ordered pairs (f, g) with $f \in \mathsf{mor}(\mathcal{C})$ and $g \in \mathsf{mor}(\mathcal{D})$, i.e. $\mathsf{mor}(\mathcal{C} \times \mathcal{D}) = \mathsf{mor}(\mathcal{C}) \times \mathsf{mor}(\mathcal{D})$.
- The composition of the morphisms is component wise, i.e. $(f, g) \circ (f', g') = (f \circ f', g \circ g')$.

For understanding what a functor is, let us clarify the subtle difference in vocabulary between a morphism and a homomorphism. A homomorphism between two structures is a map that preserves the structure, whereas a morphism is a more general concept that does not necessarily preserve the structure. For instance, a homomorphism ϕ between two magmas, i.e. a set of elements with a binary operation $\cdot$, satisfies $\phi(a_1 \cdot a_2) = \phi(a_1) \cdot \phi(a_2)$.

With that in mind, we can see a functor as a homomorphism between categories, for it preserves the structures. We define a functor as follows.

Definition I.69 (Functor)**.** Let $\mathcal{C}$ and $\mathcal{D}$ be categories. A functor F from $\mathcal{C}$ to $\mathcal{D}$ is a mapping satisfying the following.

- Each object A in $\mathcal{C}$ is associated with an object $F(A)$ in $\mathcal{D}$.
- Each morphism $f : A \to B$ in $\mathcal{C}$ is associated with a morphism $F(f) : F(A) \to F(B)$ in $\mathcal{D}$ such that the identity morphisms and the compositions are preserved:

$$\forall A \in \mathsf{ob}(\mathcal{C}), F(id_A) = id_{F(A)}$$
$$\forall f : A \to B \in \mathsf{mor}(\mathcal{C}), g : B \to C \in \mathsf{mor}(\mathcal{C}), F(g \circ g) = F(g) \circ F(f)$$

We can now see the tensor product $\otimes$ as a functor from the product category $\mathcal{C} \times \mathcal{C}$ to $\mathcal{C}$, i.e. $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$. Note that defining $\otimes$ as a functor implies Equation (I.16), i.e. $(id \otimes g) \circ (f \otimes id) = (f \otimes g) = (f \otimes id) \circ (id \otimes g)$.

We now consider a category featuring that tensor product. In algebra, one of the simplest structure of interest equipped with a binary operation is the monoid — a set S with a binary operation $\cdot : S \times S \to$

S which is associative, i.e. $(a \cdot b) \cdot c = a \cdot (b \cdot c)$, and with an identity element $e \in S$, i.e. $e \cdot a = a \cdot e = a$ for all $a \in S$. A *categorification* of a monoid is a monoidal category, where the equalities are replaced by isomorphisms satisfying *coherence* conditions. In this thesis, we use a simpler version where we only have equalities: a *strict* monoidal category.

Definition I.70 (Strict monoidal category)**.** A strict monoidal category is a category $\mathcal{C}$ equipped with a functor (tensor product) $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ and such that:

- The tensor product is associative. For all objects A, B, C and morphisms f, g, h in $\mathcal{C}$:

$$A \otimes (B \otimes C) = (A \otimes B) \otimes C$$

$$f \otimes (g \otimes h) = (f \otimes g) \otimes h$$

- There is a unit object I such that for every object A and morphism f:

$$I \otimes A = A \otimes I = A$$

$$id_I \circ f = f \circ id_I = f$$

Definition I.71 (Strict braided monoidal category)**.** A category $\mathcal{C}$ is a strict braided monoidal category if, for any objects A and B, there exists an isomorphism $\sigma_{A,B} : A \otimes B \to B \otimes A$, called the *braiding*, such that:

- For all morphisms $f : A \to B$ and $g : C \to D$,

$$(g \otimes f) \circ \sigma_{A,C} = \sigma_{B,D} \circ (f \otimes g).$$

- $\sigma_{A \otimes B,C} = (\sigma_{A,C} \otimes \mathrm{id}_B) \circ (\mathrm{id}_A \otimes \sigma_{B,C})$.
- $\sigma_{A,B \otimes C} = (\mathrm{id}_B \otimes \sigma_{A,C}) \circ (\sigma_{A,B} \otimes \mathrm{id}_C)$.

Definition I.72 (Strict symmetric monoidal category)**.** A category $\mathcal{C}$ is called a strict symmetric monoidal category if, in addition to being a strict braided monoidal category:

- For all objects $A, B \in \mathrm{Ob}(\mathcal{C})$,

$$\sigma_{B,A} \circ \sigma_{A,B} = \mathrm{id}_{A \otimes B}\,.$$

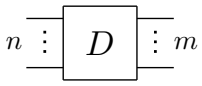
We're now able to properly define the categories of interest, that only use a single object as a generator. The first we introduce is a PRO (PROduct category).

Definition I.73 (PRO)**.** A PRO is a strict monoidal category whose objects are freely generated by a single object and $\otimes$.

Note that the PRO does not contain the simplifications of the SWAPs, e.g. Equation (I.18). To take into account these properties, we need to use the strict symmetric monoidal category. This is the PROP (Product and Permutation category).

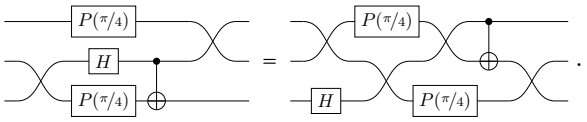
Definition I.74 (PROP)**.** A PROP is a strict symmetric monoidal category whose objects are freely generated by a single object and $\otimes$.

In both cases, as every object is of the form $X \otimes \cdots \otimes X = X^n$, it is usual to represent these objects by integers. A process $D : n \to m$ takes the object X^n as input and outputs X^m. Graphically, it simply corresponds to the number of input and output wires:



Example I.75 (Quantum circuits)**.** We can now formally define quantum circuits as a PROP, where the single object X represents a qubit. The morphisms, i.e. the boxes, correspond to elementary quantum gates, such as those listed in Table I.1. This framework provides a rigorous mathematical definition for such diagrams while capturing all natural deformations of wires depicted in Section I.4.2.

For instance, the PROP naturally ensures the following equality:



I.4.4 Manipulating diagrams with rewriting theory and logic

Now that we have mathematically formalized our diagrams, the goal is to reason directly with them. More precisely, we want to perform calculations and proofs through rewriting, similar to algebraic manipulation in traditional algebra. This requires introducing the notion of syntax and semantics, to distinguish the diagrams themselves and the process they represent (Section I.4.4.A.). Then we introduce properties of rewriting theory for transforming the diagrams (Section I.4.4.B.). Building on these concepts, we define what an equational theory is, together with prominent properties such as soundness and completeness (Section I.4.4.C.).

A. Syntax and semantics

Each diagram is a term that symbolizes a process. We distinguish between the graphical representation of a diagram and the rules used to construct it, on one hand, and its mathematical interpretation, on the other. These are captured by syntax and semantics: terminology is fundamental and common to every language, justifying the term graphical languages.

> **Definition I.76** (Syntax)**.** The syntax of a language consists of a set of rules for constructing valid terms. These rules define how basic elements (generators) can be combined.

Remark I.77 (Syntax of a graphical language). In the context of graphical languages, syntactic rules include horizontal and vertical composition, as described in Section I.4.2. The structures introduced in Section I.4.3 model sets of well-formed diagrams induced by these syntactic rules.

> **Definition I.78** (Semantics)**.** The semantics of a language assigns mathematical meaning to its terms. It provides an interpretation function that maps syntactic expressions to objects in a mathematical structure. The semantics is noted with denotation brackets $[\![\cdot]\!]$.

Remark I.79. In the case of graphical languages, the semantics is typically given by a functor $[\![\cdot]\!] : \mathcal{C} \to \mathcal{C}'$ where $\mathcal{C}$ is the syntactic category of diagrams, and $\mathcal{C}'$ is a target category that provides their mathematical meaning.

Example I.80 (Semantics of a quantum circuit)**.** For instance in the context of quantum circuits with qubits, the usual semantics maps diagrams to the category of finite-dimensional Hilbert spaces and linear maps. In particular, a circuit C acting on n qubits is an element of $U(\mathbb{C}^{2^n})$. The usual qubit semantics of the generators correspond to the matrices of Table I.1. For instance, we have:

$$\left[\!\left[-\boxed{H}- \right]\!\right]_{\mathsf{QC}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

B. Rewriting rules

To enable reasoning solely on diagrams, we need a mechanism to transform one diagram into another while preserving structural consistency. One approach is to use rewriting rules [DJ90; BN98], which specify how a diagram can be systematically modified. A general framework for studying such rewritings of objects is an abstract rewriting system.

Definition I.81 (Abstract Rewriting System)**.** An abstract rewriting system (ARS) is a pair $(A, \rightsquigarrow)$, where:

- A is a set of objects (terms, diagrams, etc.),
- $\rightsquigarrow \subseteq A \times A$ is a binary relation defining the rewriting steps.

If $a \rightsquigarrow b$, we say a rewrites to b.

Definition I.82 (Strongly normalizing)**.** An ARS is strongly normalizing (or terminating) if there is no infinite sequence:

$$a_1 \rightsquigarrow a_2 \rightsquigarrow a_3 \rightsquigarrow \dots$$

for any $a_1 \in A$.

Remark I.83. To prove an ARS is strongly normalizing, it is common to introduce a rank function $f : A \to \mathbb{N}$ that strictly decreases with each rewriting step. If such a function exists, no infinite descending chain can exist, ensuring termination.

Definition I.84 (Normal Form)**.** A term a is in normal form if there is no b such that $a \rightsquigarrow b$.

Together with termination, it is important to have a rewriting system where the result does not depend on the order of the rewriting steps. This property is known as confluence.

Definition I.85 (Confluence)**.** An ARS is confluent if, whenever a term rewrites in two different ways:

$$a \rightsquigarrow^* b_1, \quad a \rightsquigarrow^* b_2,$$

there exists c such that $b_1 \rightsquigarrow^* c$ and $b_2 \rightsquigarrow^* c$, where $\rightsquigarrow^*$ denotes the reflexive transitive closure of $\rightsquigarrow$.

The confluence property is highly useful but can be complex to prove. Remarkably, for a terminating ARS, it suffices to prove a weaker property known as local confluence. This result is known as Newman's Lemma [New42].

Definition I.86 (Local Confluence)**.** An ARS is locally confluent if for all a, b_1, b_2 such that:

$$a \rightsquigarrow b_1, \quad a \rightsquigarrow b_2,$$

there exists c such that $b_1 \rightsquigarrow^* c$ and $b_2 \rightsquigarrow^* c$.

Lemma I.87 (Newman's Lemma)**.** *A terminating ARS is confluent if and only if it is locally confluent.*

C. Equational Theory: Soundness and Completeness

In rewriting systems, rules are oriented, specifying a direction of transformation. In contrast, we seek unoriented rules where transformations can be applied in both directions, such as equations in traditional algebraic manipulations. To extend this principle to diagrammatic reasoning, we define the notion of a calculus.

Definition I.88 (Calculus)**.** A calculus $\mathcal{C}$ consists of:

- A set of terms (or diagrams) $\mathcal{D}$.
- A set of equations (or axioms) of the form $D_1 = D_2$, defining an equational theory.
- Inference rules allowing the derivation of new equations using substitution (replacing subterms with equivalent ones) and transitivity ($D_1 = D_2$, $D_2 = D_3$ implies $D_1 = D_3$).

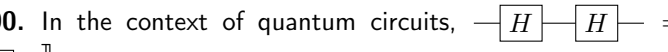
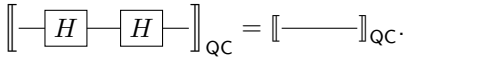
We write $\mathcal{C} \vdash D_1 = D_2$ if $D_1 = D_2$ is derivable from the axioms of $\mathcal{C}$.

Furthermore, for a calculus to be meaningful, we need the derived equations to correspond to actual equivalences in the intended interpretation. This leads to the concept of soundness.

Definition I.89 (Soundness)**.** A calculus $\mathcal{C}$ is sound with respect to a semantics $[\![\cdot]\!]$if for any terms D_1 and D_2, we have:

$$\mathcal{C} \vdash D_1 = D_2 \Rightarrow [\![D_1]\!] = [\![D_2]\!] .$$

That is, if an equation is derivable in the calculus, then we are ensured the two terms have the same interpretation.

Example I.90. In the context of quantum circuits,  is sound because  .

While soundness ensures that all derivable equations are valid in the interpretation, it does not guarantee that all valid equations are derivable. This leads to the notion of completeness.

Definition I.91 (Completeness)**.** A calculus $\mathcal{C}$ is *complete* if for any terms D_1, D_2, we have:

$$[\![D_1]\!] = [\![D_2]\!] \Rightarrow \mathcal{C} \vdash D_1 = D_2.$$

That is, if two terms have the same interpretation, then they must be derivable as equal using the axioms of $\mathcal{C}$.

The concepts of soundness and completeness originate in logic [Tar94], where equations serve as axioms, equality as statements, and derivations using the axioms as proofs. While having a complete equational theory is crucial for the soundness and reliability of a graphical language, achieving completeness is not always feasible.

In mathematics, the axioms commonly used are those of Zermelo-Fraenkel (ZF) set theory [Zer08; Fra22; Zer30][17]. Interestingly, Gödel's incompleteness theorem [Göd31] shows that such a theory is

17. Or ZFC if the axiom of choice is included.

inherently incomplete: there exist statements that cannot be derived from the axioms. In contrast, the ZX-calculus [CD08; CD11] — a prominent graphical language in quantum computing [CK17; Wet20] — has a complete equational theory [JPV18; HNW18]. This enables the transformation of any two equivalent ZX-diagrams using the set of axioms, thereby representing all transformations of quantum linear maps.

Chapter II
Simulation of linear optical circuits

To characterize physical hardware, assess its capabilities, and emulate its behavior in near-term architectures, it is crucial to develop efficient simulation algorithms that address a wide range of needs. Such algorithms become even more valuable when integrated into accessible and optimized software.

However, despite the need for simulation, the existing methods in the literature were limited. Most approaches focused on the computation of a single output probability [Sch04; GIQ19; Shc20], often under restrictive assumptions such as specific input states or idealized components. Thus, there was no comprehensive method for efficiently handling the many-output case. Furthermore, no algorithm in the literature efficiently supported feedforward — an essential feature for scalable and fault-tolerant photonic architectures [KLM01; Bar+23; Gli+24].

In this chapter, we introduce several state-of-the-art computational frameworks, covering both strong and weak simulation of linear optical circuits for various types of inputs and outputs. A key focus is on incorporating feedforward in all these different settings. The chapter is structured as follows.

- Section II.1. We present the simulation problems covered in this chapter, providing examples of applications. We also introduce key definitions and notations.

- Section II.2. We introduce SLOS (Strong Linear Optical Simulator), a generic algorithm for strong simulation that significantly reduces computation time at the cost of high memory consumption. We demonstrate how SLOS can simulate circuits with feedforward. This work led to [Heu+23b].

- Section II.3. We introduce LO-SLAP (Linear Optical Simulator through LAttice of Polynomials), a generic framework for simulation that requires significantly less memory than SLOS, with a trade-off in computational speed. We analyze the preferred regimes for SLOS and LO-SLAP, including the context of simulating feedforward. This work led to [GH25].

- Section II.4. Inspired by a gate-by-gate sampling algorithm for quantum circuits, we present a new Boson Sampling algorithm that supports both feedforward and generalized input states. We detail two use-cases, analyzing the complexity in both settings. The first involves input states that are superpositions of single-photon and vacuum states. The second considers an entangled linear cluster state, for which we additionally introduce a generalized Ryser's formula to compute sums of permanents derived from the same matrix, enabling faster computation than the state of the art.

II.1 Motivations and notation

II.1.1 Simulation problems and use cases

We introduce common simulation problems while describing when they can be useful. These problems include strong (Problem II.1 and II.3), weak (Problem II.4) simulation, and the feedforward case (Problem II.5 and II.6).

A. Computation of all output amplitudes

Problem II.1 (Full output amplitude simulation)**.** *Given an n-photon Fock state $|s\rangle$, and an m-mode linear optical circuit $\hat{U}$, output the state $\hat{U}|s\rangle$. Or equivalently, output all the amplitudes* $\left\{\langle t|\hat{U}|s\rangle \middle| |t\rangle \in \mathcal{F}_m^n\right\}$.

Example II.2. For instance, we can consider a linear optical circuit of three modes implementing a 3×3 unitary U, with an input of two photons $|1,1,0\rangle$. In that problem, we would like to compute every output state of $\mathcal{F}_3^2$, meaning we would like to know the output:

$$\hat{U}|1,1,0\rangle = \alpha_1|2,0,0\rangle + \alpha_2|0,2,0\rangle + \alpha_3|0,0,2\rangle + \alpha_4|1,1,0\rangle + \alpha_5|1,0,1\rangle + \alpha_6|0,1,1\rangle.$$

Notice that we can compute each amplitude α_i of an output state $|t\rangle$ by computing the permanent of $U_{|1,1,0\rangle,|t\rangle}$, as described in Proposition I.50. To compute α_3, we would need to compute $\mathrm{Perm}(U_{|1,1,0\rangle,|0,0,2\rangle})$. By taking the first and second column of U, $\begin{pmatrix} u_{1,1} & u_{1,2}\\ u_{2,1} & u_{2,2}\\ u_{3,1} & u_{3,2}\end{pmatrix}$ and repeating two times the third row to form a $U_{|1,1,0\rangle,|0,0,2\rangle}$, we have:

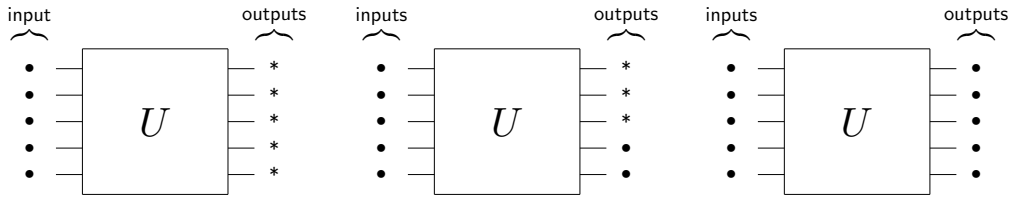
$$\alpha_3 = \frac{\mathrm{Perm}(U_{|1,1,0\rangle,|0,0,2\rangle})}{\sqrt{2!}} = \frac{\mathrm{Perm}\begin{pmatrix} u_{3,1} & u_{3,2}\\ u_{3,1} & u_{3,2}\end{pmatrix}}{\sqrt{2}} = \sqrt{2}\times u_{3,1}\times u_{3,2}$$

Typical use cases. As the full amplitude list is a complete description of a linear optical circuit and the input can be directly the state right after the photon sources, applications for this problem can be both practical and theoretical. One can think of:

- Simulate or look for circuits preparing a specific output distribution like entangled high-dimensional states [Gub+20] or Bell Measurements [OG18].
- Check the correctness and noise of experimental circuits with the theoretical distribution. One can certify the correctness of Boson sampling by computing metrics (such as total variational distance, bunching probabilities,...) from the statistics obtained [Vil+22; Wal+16; Shc21a; Shc22].
- Machine Learning algorithms as the approximation of differential equations of [GLA22], application detailed in Section II.2.4.A..

B. Computation over generic input and output sets

Problem II.3 (Generic Strong Simulation)**.** *Given a set of input Fock states $\mathcal{I}$, a set of output Fock states $\mathcal{O}$, and an m-mode linear optical circuit $\hat{U}$, output the amplitudes* $\left\{\langle t|\hat{U}|s\rangle, |s\rangle\in\mathcal{I}, |t\rangle\in\mathcal{O}\right\}$.



(a) Full distribution - one input (b) Heralded Scheme (c) Postselected Scheme

Figure II.1 – Illustrations of Problem II.1 and Problem II.3. • is a chosen number of photons, while $*$ means we consider every possible number.

Typical use cases. Often, only specific outcomes are of interest, so the full output distribution of a linear optical circuit is not needed. Also, for every circuit encoding logic gates or functions in general, we need to know the effect of the LO circuit on each different possible input, so several inputs are needed. Therefore, that problem is very general and tackles various applications, in which we can highlight two general classes of schemes.

- Post-selected scheme. Presented in Figure II.1c, it consists of only considering —or selecting— some inputs and some outputs according to some criteria. A canonical example is the implementation of the 2-qubit CNOT gate of [Ral+02]. Working on 6 modes with 2 photons, the considered input and output states are $X = \{|0,1,0,1,0,0\rangle, |0,1,0,0,1,0\rangle, |0,0,1,1,0,0\rangle, |0,0,1,0,1,0\rangle\}$. In other words, the circuit is regarded as a map restricted on the 4-dimensional subspace $\mathbb{C}[X]$ generated by X, and not the full 21-dimensional $\mathcal{H}_6^2$ space. The term "postselected" refers to the fact that in general, the output state might have a component orthogonal to $\mathbb{C}[X]$ (for instance, maybe the state $|0,2,0,0,0,0\rangle$ has a non-zero amplitude). Such orthogonal components are considered bogus. They are ruled out at the end of the computation, when measuring the system.
- Heralded scheme. Presented in Figure II.1b, it consists in designing a LO circuit that can fail, but for which the failure can be decided upon the measurement result of some output modes. For instance, in [Kni02] a scheme is proposed to implement the 2-qubit CZ gate. The 4 input modes can accommodate 2, 3 or 4 photons, and the input state is generated with $\{|1,1,1,1\rangle, |1,0,1,1\rangle, |0,1,1,1\rangle, |0,0,1,1\rangle\}$. For the output, we only consider the cases where the two last modes contain a photon: $|*,*,1,1\rangle$. This scheme is "heralded" in the sense that the computation only happens on the two first modes: the last two modes are just witnesses that the computation went well. Unlike the post-selected scheme, measuring these two last modes does not destruct possible entanglement on the two first modes.

C. Boson sampling with generalized input state

Problem II.4 (Generalized Boson Sampling)**.** *Given an arbitrary finite-dimensional quantum state $|\psi\rangle$ and an m-mode linear optical circuit $\hat{U}$, output a random sample $|t\rangle$ with probability $|\langle t|\hat{U}|\psi\rangle|^2$.*

Typical use case. Sampling from generic relevant to encompass all the operations that are physically possible. For instance, one can think of:

- Quantum linear optical computing schemes. Indeed, when performing computation, we may have generic qubit quantum states entering linear optical circuits. Such generalized Boson Sampling algorithm could therefore be useful for emulating such circuits.

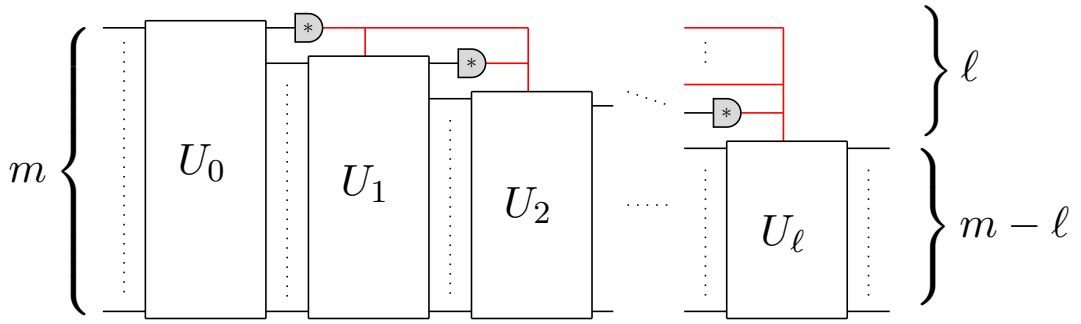


Figure II.2 – Illustration of a generic linear optical circuit with feedforward. This is the most generic scheme containing ℓ adaptive modes. Each U_i is parametrized by the previous i measurements.

- Sampling experiments where sources are generating other states than single photons or squeezed states. For instance, it is possible to generate a coherent superposition of single-photon state and vacuum directly from sources [Lor+19]. This use case is detailed in Section II.4.2.

D. Weak and strong simulation of feedforward

Without loss of generality when considering linear optical circuits with adaptive measurement, we can assume that the modes to be measured are always the first ones. Otherwise, we can rearrange the order by swapping the rows of U_i. The scheme of a linear optical circuit with feedforward is illustrated in Figure II.2. We consider the two following problems.

Problem II.5 (Strong simulation of feedforward)**.** *Given an input Fock state $|s\rangle$ and a circuit $\hat{C}$ constituted of ℓ linear optical circuits $\hat{U}_i$ conditioned on the i previous measurements, output the full output amplitude distribution* $\left\{\langle t|\hat{C}|s\rangle \middle| |t\rangle \in \mathcal{F}_m^n\right\}$.

Problem II.6 (Weak simulation of feedforward)**.** *Given an arbitrary finite-dimensional quantum state $|\psi\rangle$ and a circuit $\hat{C}$ constituted of ℓ linear optical circuits $\hat{U}_i$ conditioned on the i previous measurements, output a random sample $|t\rangle$ with probability $|\langle t|\hat{C}|\psi\rangle|^2$.*

Typical use cases. Feedforward is an adaptive scheme which is known to make linear optics universal for quantum computing [KLM01]. Simulating LO circuits with feedforward can therefore help to simulate:

- Every quantum linear optical scheme, e.g. [Bar+23] and [Gli+24]. Thus, it can directly characterize and emulate such systems at the hardware level rather than the qubit one.
- Non-universal schemes that make use of the expressivity and complexity of feedforward. For instance, we can use an adaptive linear optical circuit to define an observable, that can be used as a machine learning kernel [CMS21] hard to approximate classically.

II.1.2 Definitions and notation

By convention, an input state is often labeled by s, and an output state by t. Additionally, we define the following notations that will improve readability.

Notation II.7 (Vacuum state)**.** Regardless of the number of modes, we will note the state $|0\dots0\rangle$ as $|vac\rangle$.

Notation II.8 (Number of output states)**.** For conciseness, the number of configurations for distributing n indistinguishable photons among m modes (Proposition I.47) may be noted M_n instead of $\binom{m+n-1}{n}$.

Notation II.9 (Creation operators)**.** For simplicity, we now write x_i instead of $a_i^\dagger$ and we write $P(x)$ the m-variable n-degree homogeneous multivariate polynomial such that

$$|\psi\rangle = P(x)\,|00\dots0\rangle\,.$$

Given a state $|\psi\rangle = P(x)\,|00\dots0\rangle$, applying a unitary transformation U gives the new state $|\psi'\rangle = P(U^T x)\,|00\dots0\rangle$ with $U^T x = \begin{pmatrix} \sum_i u_{i1}x_i \\ \vdots \\ \sum_i u_{im}x_i \end{pmatrix}$.

In order to efficiently restrict the computational space (e.g. in Section II.2.1.B.), we introduce the notion of a mask, i.e. a state which will filter out states depending on the following relation.

Definition II.10 (Partial order on Fock basis states: Mask)**.** Let $\leq$ be a partial order defined on the basis states of the Fock space. Specifically, for any two basis states,

$$|t_1,\dots,t_m\rangle \leq |\mathcal{M}_1,\dots,\mathcal{M}_m\rangle \quad\iff\quad \bigwedge_{i=1}^{m}(t_i \leq \mathcal{M}_i).$$

Given a basis state $|\mathcal{M}\rangle$, we define the corresponding subset of the Fock space as

$$\mathcal{F}^k_{\leq\mathcal{M}} := \left\{|t\rangle \in \mathcal{F}^k_m \mid |t\rangle \leq |\mathcal{M}\rangle\right\}.$$

We refer to $\mathcal{M}$ as a mask. Furthermore, for a given set of masks $\mathcal{S}_\mathcal{M}$, we define the corresponding constrained Fock space as

$$\mathcal{F}^k_{\mathcal{S}_\mathcal{M}} := \bigcup_{|\mathcal{M}\rangle\in\mathcal{S}_\mathcal{M}} \mathcal{F}^k_{\leq\mathcal{M}}.$$

II.2 SLOS: Strong linear optical simulator

In this section, we present SLOS (Strong Linear Optical Simulator). We begin with a detailed description of how SLOS works, including detailed pseudocodes (Section II.2.1). Next, we analyze its time and memory complexity (Section II.2.2). We then describe its optimized implementation (Section II.2.3), and provide benchmarks comparing SLOS to state-of-the-art methods (Section II.2.4). Finally, we explore extensions of SLOS to feedforward and more general simulation settings (Section II.2.5).

Contributions

Most of this section is based on the paper [Heu+23b], published in Computer Physics Communications (CPC), coauthored with Shane Mansfield, Jean Senellart and Benoît Valiron. The optimized implementation (see Section II.2.3) and the benchmarking work were primarily carried out by Jean Senellart. The feedforward extension (Section II.2.5) was developed in collaboration with Timothée Goubault de Brugière [GH25]. The remainder of the work presented in this section is the result of my own contribution.

II.2.1 Algorithm

We first introduce an algorithm computing the full distribution of a given input (Section II.2.1.A.). This preliminary version helps to introduce a more generalized algorithm for various set of inputs and outputs: SLOS (Section II.2.1.B.).

A. Full output distribution

Given an n-photon input state $|s\rangle = |s_1, \dots, s_m\rangle$ and a linear optical circuit of m modes implementing a unitary matrix $U : m \times m$, up to normalization factors we have:

$$\hat{U}\,|s\rangle = \hat{U}\left(\prod_{j=1}^{m} \frac{1}{\sqrt{s_j!}} x_j^{s_j}\,|vac\rangle\right) = \prod_{j=1}^{m} \frac{1}{\sqrt{s_j!}} \left(\sum_{i=1}^{m} u_{i,j} x_i\right)^{s_j} |vac\rangle$$

As $\sum_{j=1}^{m} s_j = n$, the product only contains n non-trivial terms, that can be arbitrary labeled as $p_1, p_2, \dots, p_n$, corresponding to the position of each photon. The key idea, presented in Algorithm 1, is to decompose the product as follows:

$$\sum_{i_n=1}^{m} u_{i_n,p_n} x_{p_n} \left(\sum_{i_{n-1}=1}^{m} u_{i_{n-1},p_{n-1}} x_{p_{n-1}} \left(\dots \left(\sum_{i_1=1}^{m} u_{i_1,p_1} x_{p_1}\,|0\dots0\rangle\right)\dots\right)\right) \tag{II.1}$$

With this chosen order, we iteratively obtain the output of a state with $k+1$ photons from the output of a state with k photons. We obtain the following inductive formula for $k = 0$ to $n-1$:

$$\hat{U}|s^{(k+1)}\rangle = \sum_{i=1}^{m} u_{i,p_{k+1}} x_i \left(\hat{U}|s^{(k)}\rangle\right) \tag{II.2}$$

where we remind the notation: $|s^{(k)}\rangle = \prod_{j=1}^{k} x_{p_j}\,|vac\rangle$, where $|s^{(k)}\rangle$ is the Fock state $|s^{(k-1)}\rangle$ with one more photon in the mode p_k. We chose to store these intermediate states $|s^{(k)}\rangle$ in a list $\mathcal{I}_S$ of tuples: $[(|0^{(0)}\rangle, p_1), \dots, (|s^{(n-1)}\rangle, p_n)]$.

In practice, obtaining the full distribution $\hat{U}|s^{(k)}\rangle = \sum_{|t\rangle\in\mathcal{F}_m^k} \langle t|\,\hat{U}|s^{(k)}\rangle\,|t\rangle$ means computing and storing every coefficient $\langle t|\,\hat{U}|s^{(k)}\rangle$ for $|t\rangle \in \mathcal{F}_m^k$. We store each of them in an array $\hat{U}$, where each coefficient of $\hat{U}[|s^{(k)}\rangle][|t\rangle]$ for $|t\rangle \in \mathcal{F}_m^k$ is used for the computation of $\hat{U}[|s^{(k+1)}\rangle][|t\rangle]$ for $|t\rangle \in \mathcal{F}_m^{k+1}$. Note that the normalization factor of $1/\Pi_{j=1}^{m}\sqrt{s_j!}$ can be computed at the end or by renormalizing each step k such as in Algorithm 1, and that $x_i\,|t\rangle$ involves a normalization factor of $\sqrt{t_i+1}$.

Algorithm 1 Computation of the full distribution $\hat{U}|s^{(n)}\rangle$, given one input $|s^{(n)}\rangle$.
Each coefficient $\langle t|\,\hat{U}\,|s\rangle$ is stored in an array $\hat{U}$ and is accessed by $\hat{U}[|s\rangle][|t\rangle]$. The access of the coefficient $u_{i,p}$ is labelled as $U[i][p]$. Every coefficient of $\hat{U}$ is initialized to 0. For simplicity, the array $\hat{U}$ is of size $O\left(\left(\frac{n}{m}+1\right)M_n\right)$, even though it could be optimised to $O\left(M_{n-1}+M_n\right)$, by reallocating memory and erasing the intermediary states, as shown below in the comment.

procedure $\mathrm{SLOS_FULL}(|s^{(n)}\rangle, U)$
  $\hat{U}[|0\rangle][|0\rangle] \leftarrow 1$
  $\mathcal{I}_S \leftarrow \left[\left(|vac\rangle, p_1\right), \ldots, \left(|s^{(n-1)}\rangle, p_n\right)\right]$ ▷ Chosen arbitrarily, see Equation (II.1)
  for $k = 0$ to $n-1$ **do**
    $\left(|s\rangle, p\right) \leftarrow \mathcal{I}_S[k]$
    for $|t\rangle \in \mathcal{F}^k$ **do**
      for $i \in [m]$ **do**
        $\hat{U}[|s_1, \ldots, s_p+1, \ldots, s_m\rangle][|t_1, \ldots, t_i+1, \ldots, t_m\rangle]$
          $+= \sqrt{\frac{t_i+1}{s_p+1}} \times U[i][l] \times \hat{U}[|s\rangle][|t\rangle]$
      end for
          ▷ Possible memory optimization here: we don't need $\hat{U}[|s\rangle][|t\rangle]$ anymore
    end for
  end for
end procedure

Remark II.11. This way of developing the product has been widely used and known as folklore. The contribution of this work is to both perform a formal complexity analysis and efficient implementation, as well as extend it to generic set of inputs and outputs, as developed in the next sections.

B. SLOS: Generic input and output sets

Algorithm 1 always computes the full output distribution, and does not offer any granularity by restricting the set of input or outputs states.

Generic set of output states. In order to restrict the computation to a given output set, we will use the notion of masks (see Definition II.10).

Proposition II.12 (Masked output set for SLOS)**.** *Given a set of output basis states $\mathcal{O}$ and an input state $|s\rangle$, we can compute the set*

$$\left\{\langle t|\,\hat{U}\,|s\rangle \mid |t\rangle \in \mathcal{O}\right\}$$

by setting $\mathcal{S}_{\mathcal{M}} = \mathcal{O}$ and computing, for $k = 0$ to n, the values

$$\left\{\langle t|\,\hat{U}|s^{(k)}\rangle \mid |t\rangle \in \mathcal{F}^k_{\mathcal{S}_{\mathcal{M}}}\right\},$$

where

$$\mathcal{F}^k_{\mathcal{S}_{\mathcal{M}}} = \bigcup_{|\mathcal{M}\rangle \in \mathcal{S}_{\mathcal{M}}} \left\{|t\rangle \in \mathcal{F}^k_m \mid |t\rangle \leq |\mathcal{M}\rangle\right\}$$

(see Definition II.10).

Proof. For now, at the step k of the computation, each coefficient $\langle t|\,\hat{U}|s^{(k)}\rangle$ is computed as follows:

$$\langle t|\,\hat{U}|s^{(k)}\rangle = \frac{1}{\sqrt{s_{p_k}}} \sum_{i,t_i\neq 0} \sqrt{t_i} u_{i,p_k} \langle t_1,\ldots,t_i-1,\ldots,t_m|\,\hat{U}|s^{(k-1)}\rangle \tag{II.3}$$

Note that

$$|t^{(k)}\rangle \in \mathcal{F}^k_{\leq \mathcal{M}} \Rightarrow \{|t_1,\ldots,t_i-1,\ldots,t_m\rangle | t_i \neq 0\} \subset \mathcal{F}^{k-1}_{\leq \mathcal{M}}.$$

Therefore, for any mask $\mathcal{M}$, the set $\left\{\langle t|\,\hat{U}|s^{(k)}\rangle \middle| |t\rangle \in \mathcal{F}^k_{\leq \mathcal{M}}\right\}$ can be computed with $\left\{\langle t|\,\hat{U}|s^{(k-1)}\rangle \middle| |t\rangle \in \mathcal{F}^{k-1}_{\leq \mathcal{M}}\right\}$. For each k, it is therefore sufficient to compute the states $\mathcal{F}^k_{\mathcal{S}_\mathcal{M}} = \left\{\mathcal{F}^k_{\leq \mathcal{M}} \middle| \mathcal{M} \in \mathcal{S}_\mathcal{M}\right\}$. Setting $\mathcal{S}_\mathcal{M} = \mathcal{O}$ assures we obtain the right output set at the end of the computation:

$$\mathcal{F}^n_\mathcal{O} = \bigcup_{|o\rangle\in\mathcal{O}} \{|t\rangle \in \mathcal{F}^n_m \mid |t\rangle \leq |o\rangle\} = \mathcal{O}.$$

□

Several Inputs. Let us consider two inputs $|s\rangle$ and $|s'\rangle$ that have j photons in common modes. Thus, we can write them as $|d_1+c_1,\ldots,d_m+c_m\rangle$ and $|d'_1+c_1,\ldots,d'_m+c_m\rangle$, with $d_i, d'_i, c_i \geq 0$ and $\sum_{i=1}^m c_i = j$. The idea is to first compute the common term of both $\hat{U}|s^{(n)}\rangle$ and $\hat{U}|s'^{(n)}\rangle$:

$$\hat{U}\,|c_1,\ldots,c_m\rangle = \prod_{p=1}^m \left(\sum_{i=1}^n u_{i,p}x_i\right)^{c_p} |vac\rangle\,.$$

Note that $|c\rangle = |\min(s_1,s'_1),\ldots,\min(s_m,s'_m)\rangle$ and $j = \sum_{i=1}^m \min(s_i,s'_i)$.

More generally with a set of inputs $\mathcal{I}$, we can always factorize the common terms, and use it in the computation to reduce the number of intermediary states. It is especially efficient when the common term is the same for all the inputs, as it is the case when we add auxiliary photons for all different inputs, as in $|*,*,\ldots,*,1,1,1\rangle$.

In practice, we will need to implement $\mathcal{I}_S$ as a list of sets, where $\mathcal{I}_S[k]$ is the set of several tuples $\left(|s^{(k)}\rangle, p_k\right)$ to compute at the step k. The challenge is to build such $\mathcal{I}_S$ that optimizes the total number of operations. The heuristic we propose works as follows. Let us denote $\mathcal{I}^2$ as the set of all distinct pairs of $\mathcal{I}\times\mathcal{I}$. If we have $p \geq 2$ inputs, then we would choose one of the largest common term among all pairs of $\mathcal{I}_2$. Therefore, we would take one element of:

$$\underset{(|s\rangle,|s'\rangle)\in\mathcal{I}^2}{\operatorname{argmax}} \sum_{i=1}^m \min(s_i,s'_i) \tag{II.4}$$

We then remove the chosen pair $(|s\rangle, |s'\rangle)$ from $\mathcal{I}$, and add $|\min(s_1,t_1),\ldots,\min(s_m,t_m)\rangle$ both to $\mathcal{I}$ and to $\mathcal{I}_S$, with arbitrary paths from that state to $|s\rangle$ and $|s'\rangle$. The procedure will continue until we have only one input state remaining in $\mathcal{I}$. At that step, we add this state in I_S with an arbitrary path, as in Equation (II.1). The pseudocode of the whole procedure is the Algorithm 2.

Example II.13. To illustrate the correspondence between $\mathcal{I}$ and $\mathcal{I}_\mathcal{S}$, we give the steps for the computing the coefficients associated to $\{|1,0,0,1\rangle, |1,1,1,2\rangle, |1,1,2,1\rangle\}$.

1. This is the initial input to the procedure, and $\mathcal{I}_S$ is empty: $\mathcal{I} = \{|1,0,0,1\rangle, |1,1,1,2\rangle, |1,1,2,1\rangle\}$, $\mathcal{I}_S = [\{\},\{\},\{\},\{\},\{\}]$
2. We look for the best common factor between the three inputs (the largest common term of Equation (II.4)): it is $|1,1,1,1\rangle$, factor of $|1,1,1,2\rangle$ and $|1,1,2,1\rangle$. So now $\mathcal{I} = \{|1,0,0,1\rangle, |1,1,1,1\rangle\}$, and $\mathcal{I}_S$ is populated with the operations needed to recover the two elements being factored: $\mathcal{I}_S = [\{\},\{\},\{\},\{\},\{(|1,1,1,1\rangle, 4), (|1,1,1,1\rangle, 3)\}]$

Algorithm 2 Construction of $\mathcal{I}_S$. The function **path**$(|t\rangle \to |s\rangle)$ returns an arbitrary path from $|t\rangle$ to $|s\rangle$ as in Equation (II.1).

$\mathcal{I}_S \leftarrow [\{\}$ **for** $i \in [m]]$
while $\#\mathcal{I} \geq 2$ **do**
    argmax, max $\leftarrow$ **None**, 0
    for $(|s\rangle, |s'\rangle) \in \left\{(|s\rangle, |s'\rangle) \in \mathcal{I}^2, |s\rangle \neq |s'\rangle\right\}$ **do**
        $k \leftarrow \sum_{i=1}^{m} \min(s_i, s'_i)$
        if $k >$ max **then**
            argmax, max $\leftarrow (|s\rangle, |s'\rangle)$, k
        end if
    end for
    if max $= 0$ **then** ▷ No factorization found: we add the paths of all remaining elements in $\mathcal{I}_S$and empty $\mathcal{I}$
        for $|s\rangle \in \mathcal{I}$ **do**
            for $(|t^{(i)}, l) \in$ **path**$(|0\rangle_m \to |s\rangle)$ **do**
                $\mathcal{I}_S[i]$.add$((|t^{(i)}\rangle, l))$
            end for
        end for
        $\mathcal{I}$.clear()
    else ▷ Factorization for a pair found: we add the paths of the elements of the pair in $\mathcal{I}_S$, remove them from $\mathcal{I}$, and add the factor in $\mathcal{I}$
        $(|s\rangle, |s'\rangle) \leftarrow$ argmax
        $\left|\min_{s,s'}\right\rangle \leftarrow |\min(s_1, s'_1), \ldots, \min(s_m, s'_m)\rangle$
        for $|s^\star\rangle \in (|s\rangle, |s'\rangle)$ **do**
            for $(|t^{(i)}\rangle, p) \in$ **path**$(\left|\min_{s,s'}\right\rangle \to |s^\star\rangle)$ **do**
                $\mathcal{I}_S[i]$.add$((|t^{(i)}\rangle, p))$
            end for
            $\mathcal{I}$.remove$(|s^\star\rangle)$
        end for
        $\mathcal{I}$.add$(\left|\min_{s,s'}\right\rangle)$
    end if
end while
if $\#\mathcal{I} == 1$ **then** ▷ One element remaining: we add its path in $\mathcal{I}_S$
    $|s\rangle \leftarrow \mathcal{I}$.pop()
    for $(|t^{(i)}\rangle, p) \in$ **path**$(|0\rangle_m \to |s\rangle)$ **do**
        $\mathcal{I}_S[i]$.add$((|t^{(i)}\rangle, l))$
    end for
end if

Algorithm 3 SLOS with a set of q inputs $\mathcal{I}$ and a set of r outputs $\mathcal{O}$.
Each coefficient $\langle t|\hat{U}|s\rangle$ is stored in a dictionary or an array $\hat{U}$ and will be accessed by $\hat{U}[|s\rangle][|t\rangle]$. The access of the coefficient $u_{i,p}$ is labelled as $U[i][p]$. Every coefficient of $\hat{U}$ is initialized to 0. For simplicity, the array $\hat{U}$ is of size $O\left(q(\frac{n}{m}+1)M_n\right)$, even though it could be optimized by reallocating memory and erasing the intermediary states.

Global $\hat{U}$
procedure $\mathrm{SLOS_REC}(k, \mathcal{F}^{k+1}_{\mathcal{S}_\mathcal{M}}, U)$
  if $k > 0$ **then**
    $\mathrm{SLOS_REC}(k-1, \mathcal{F}^{k}_{\mathcal{S}_\mathcal{M}}, U)$ ▷ Build with Equation (II.5)
  end if
  for $(|s\rangle, p) \in \mathcal{I}_S[k]$ **do**
    for $|t\rangle \in \mathcal{F}^{k+1}_{\mathcal{S}_\mathcal{M}}$ **do**
      for $i \in [m]$ **when** $t_i \neq 0$ **do**
        $\hat{U}[|s_1, \dots, s_p+1, \dots, s_m\rangle][|t\rangle]$
          $+= \sqrt{\frac{t_i}{s_p}} \times U[i][p] \times \hat{U}[|s\rangle][|t_1, \dots, t_i-1, \dots, t_m\rangle]$
      end for
    end for
  end for
end procedure
procedure $\mathrm{SLOS}(\mathcal{I}, \mathcal{O}, U)$
  $\hat{U}[|0\rangle][|0\rangle] \leftarrow 1$
  $\mathcal{I}_S \leftarrow \left[\{(|vac\rangle, p_1), \dots\}, \dots, \left\{\left(|s^{(n-1)}\rangle, p_n\right), \dots\right\}\right]$ ▷ As in Algorithm 2
  $\mathrm{SLOS_REC}(n-1, \mathcal{O}, U)$
end procedure

3. The next step is similar, with the difference that the common factor is a term to be computed (so we keep it). $\mathcal{I} = \{|1,0,0,1\rangle\}$, and $\mathcal{I}_S = [\{\}, \{\}, \{(|1,0,0,1\rangle, 2)\}, \{(|1,1,0,1\rangle, 3)\}, \{(|1,1,1,1\rangle, 4), (|1,1,1,1\rangle, 3)\}]$
4. Finally, we can empty $\mathcal{I}$: $\mathcal{I} = \{\}$, and $\mathcal{I}_S = [\{(|vac\rangle, 1)\}, \{(|1,0,0,0\rangle, 4)\}, \{(|1,0,0,1\rangle, 2)\}, \{(|1,1,0,1\rangle, 3)\}, \{(|1,1,1,1\rangle, 4), (|1,1,1,1\rangle, 3)\}]$

SLOS: Recursive procedure

Unlike the iterative presentation of Algorithm 1, it is here more natural to have a recursive procedure, as we can directly build the intermediary states as follows:

$$\mathcal{F}^{k-1}_{\mathcal{S}_\mathcal{M}} = \left\{|t_1, \dots, t_i - 1, \dots, t_m\rangle, t_i \neq 0, |t^{(k)}\rangle \in \mathcal{F}^{k}_{\mathcal{S}_\mathcal{M}}\right\} \tag{II.5}$$

This makes it possible to build $\mathcal{F}^{k-1}_{\mathcal{S}_\mathcal{M}}$ from $\mathcal{F}^{k}_{\mathcal{S}_\mathcal{M}}$ on the fly without precompiling anything.

The pseudocode of SLOS is given in Algorithm 3. Note that SLOS is initialized at $\mathcal{F}^{n}_{\mathcal{S}_\mathcal{M}} = \mathcal{O}$.

II.2.2 Complexity analysis

To assess the time and memory complexity, we both study the full output distribution case (Section II.2.2.A.) and the computation of one single output (Section II.2.2.B.), that we compare to the permanent-based method. The results are summarized in Table II.1.

	Time Complexity		Memory Complexity	
#Outputs	One	All	One	All
Permanent-Based [Shc21b]	$O\left(n\frac{\prod_{i=1}^{m}(t_i+1)}{\min_{t_l\neq 0}(t_l+1)}\right)$	$\Omega\binom{2m+n-1}{n}$	$O(n)$	$O(M_n)$
SLOS	$\boldsymbol{O\left(\sum_{i=1}^{m} t_i \prod_{j\neq i}(t_j+1)\right)}$	$\boldsymbol{O(nM_n)}$	$O\binom{n}{n/2}$	$\boldsymbol{O(M_n)}$

Table II.1 – Time and memory complexity analysis of SLOS and [Shc21b] for a generic output $|t\rangle = |t_1, \dots, t_m\rangle \in \mathcal{F}_m^n$, where $M_n = \#\mathcal{F}_m^n = \binom{m+n-1}{n}$. We highlighted the SLOS values in bold when they were similar or better than the permanent-based ones. For the memory complexity of the one output case, we consider the worst case, as detailed in the proof of Proposition II.22. For the memory complexity of the full distribution case, we consider the problem of storing every value, therefore needing at least $O(M_n)$ of memory. If we consider the enumerating problem instead, the permanent-based method would have a memory complexity of $O(n)$.

A. Full distribution

Proposition II.14 (Time complexity of SLOS). *SLOS computes the full distribution of n-photon input state $|s_1, \dots, s_m\rangle$ in $O\left(n\binom{n+m-1}{m-1}\right)$ operations.*

Proof. The complexity can be directly deduced from the three "for loops" of Algorithm 1 or in the recursive procedure in Algorithm 3, knowing that $\#\mathcal{F}_m^k = \binom{k+m-1}{m-1}$ for the full output distribution. The total number of operations is:

$$\sum_{k=0}^{n-1} m\binom{k+m-1}{m-1} = m\frac{n}{m}\binom{n+m-1}{m-1} = n\binom{n+m-1}{m-1}.$$

□

Remark II.15 (Quasi-optimality in time). Note that there are $\binom{n+m-1}{m-1}$ coefficients to compute for having the full output distribution. The algorithm is therefore almost, if not already, optimal in the number of operations.

Remark II.16 (Ratio between SLOS and permanent-based method). We can compare this time with the permanent-based method, consisting of computing every single output amplitude independently with the matrix permanent. For computing the full distribution, the permanent-based method requires $\Omega\left(\binom{2m+n-1}{n}\right)$ complex operations, as described in Proposition I.60. To compare their time complexities, we can study the ratio $\frac{\binom{2m+n-1}{n}}{nM_n} = \frac{1}{n}\frac{\binom{2m+n-1}{n}}{\binom{n+m-1}{n}}$. By assuming $m = \theta n$ and for a fixed value θ, (cf Corollary 2 of [CC24]), we can apply Stirling's formula to show the ratio is $\Omega(\frac{1}{n}\rho_\theta^n)$ with $\rho_\theta = \frac{(2\theta+1)^{2\theta+1}}{(4\theta)^\theta(\theta+1)^{\theta+1}}$. Note that $\rho_1 \approx 1.69$, $\rho_2 \approx 1.80$, and that $\lim_{\theta\to\infty}\rho_\theta = 2$.

Proposition II.17 (Memory complexity of Algorithm 1). *SLOS needs to store $O\left(\binom{n+m-1}{m-1}\right)$ complex coefficients for computing the full output distribution.*

Proof. At the step k, we need to use a memory of at most $O(M_k + M_{k+1})$, as we use all the coefficients of $\hat{U}|s^{(k)}\rangle$ to compute and store the new coefficients of $\hat{U}|s^{(k+1)}\rangle$. At the end of the step k, we can erase all the coefficients of the step $k-1$.

Therefore, at the last step, we have stored at most $\binom{(n-1)+m-1}{m-1}+\binom{n+m-1}{m-1} \leq 2\binom{n+m-1}{m-1} = O(M_n)$ coefficients. Thus, if we allow reallocation, the complexity in memory is in $O\left(\binom{n+m-1}{m-1}\right)$. □

Remark II.18 (Optimal memory). If we consider the problem of storing all the output coefficients, we already need of memory of $O\left(\binom{n+m-1}{m-1}\right)$, making Algorithm 1 optimal for this kind of problem [1].

It is important to highlight for a simple allocating, we need $M_{n-1} + M_n$ space which can still be inconvenient for big values of n and m. We can reduce the overhead by only allocating the memory of $\hat{U}[|s^{(k+1)}\rangle][|t_1, \ldots, t_i + 1, \ldots, t_m\rangle]$ when needed, and by erasing coefficients as soon they have been used as in Algorithm 1. We can optimize even more by ordering $\mathcal{F}_m^k$ so that for consecutive $|t\rangle$, the coefficients $|t_1, \ldots, t_i + 1, \ldots, t_m\rangle$ overlap and less memory is added at each iteration of a new $|t\rangle$. The memory optimization can also only be done for the last steps as they are the most costly.

For faster overhead in time and easier implementation, we would rather store all the coefficients of the intermediary states without erasing them. This would require storing $\sum_{k=0}^{n}\binom{k+m-1}{m-1} = (\frac{n}{m}+1)\binom{n+m-1}{m-1}$ states, which is still feasible for reasonable values of m and n.

B. One single output

In this section, we discuss the complexity of the procedure in this simple one-input, one-output case [2], and compare it with the best formula for the permanent of a matrix with repeated rows. Interestingly, SLOS is more efficient in time.

Proposition II.19 (Time complexity of one-output SLOS)**.** *Given one input $|s\rangle$, a generic $m \times m$ unitary U and an output $|t\rangle = |t_1, \ldots, t_m\rangle$, SLOS computes $\langle t|\hat{U}|s\rangle$ in $O\left(\sum_{i=1}^{m} t_i \prod_{j\neq i}(t_j+1)\right)$ operations.*

Proof. For each state $|t\rangle$, let us define $\alpha_t = \#\{t_i \neq 0, 1 \leq i \leq m\}$. From Equation (II.3), we can see that every $\langle t|\hat{U}|s^{(k)}\rangle$ needs α_t computations, using the already computed coefficients. Given an output state $|t\rangle$, we will compute every $|t'\rangle \leq |t\rangle$ where $\leq$ is defined in Definition II.10. Therefore, the number of computations is $S(|t\rangle) = \sum_{|t'\rangle\leq|t\rangle} \alpha_{t'}$. For the proof, we will use the notation $|t\rangle_k = |t_1, \ldots, t_k\rangle$, restriction of $|t\rangle_m = |t\rangle$. To give a generic formula of S, we proceed by induction by defining $S_k(|t\rangle) = \sum_{|t'\rangle\leq|t\rangle_k} \alpha_{t'}$, with $S_m(|t\rangle) = S(|t\rangle)$ and $S_1(|t\rangle) = t_1$. We can show that

$$\begin{aligned}
S_m(|t\rangle) &= \sum_{|t'\rangle\leq|t\rangle, t'_m=0} \alpha_{t'} + \sum_{k=1}^{t_m} \sum_{|t'\rangle\leq|t\rangle, t'_m=k} \alpha_{t'} \\
&= S_{m-1}(|t\rangle) + \sum_{k=1}^{t_m} \sum_{|t'\rangle\leq|t\rangle_{m-1}} (\alpha_{t'}+1) \\
&= S_{m-1}(|t\rangle) + t_m S_{m-1}(|t\rangle) + t_m \sum_{|t'\rangle\leq|t\rangle_{m-1}} 1 \\
&= (t_m+1)S_{m-1}(|t\rangle) + t_m \prod_{i=1}^{m-1}(t_i+1)
\end{aligned}$$

1. In some contexts, we only need to be able to iterate through these output coefficients. This will lead to other methods that are more memory-efficient, leading to SLAP (Section II.3).

2. While it is usually better motivated physically to have one photon at most per mode in the input $|s\rangle$, if $|s\rangle$ has more repetition than the output $|t\rangle$, it is faster and equivalent to compute $\langle s|\hat{U}^\dagger|t\rangle^\dagger$ than $\langle t|\hat{U}|s\rangle$. That trick of taking the conjugate transpose allows to transform the repetitions of the columns into the repetition of the rows if it's more efficient. In the following, we will consider a generic input $|s\rangle$ and only consider the repetitions in the output $|t\rangle$.

Leading to the following formula:

$$S(|t\rangle) = \sum_{i=1}^{m} t_i \prod_{j\neq i}(t_j+1)$$

□

SLOS provides a tighter bound on the number of operations required to compute the permanent of a matrix in which the i^{th} row is repeated t_i times, outperforming the state-of-the-art methods.

Proposition II.20 (Faster matrix permanent formula)**.** *Whenever there exist two indices i and j such that $t_i > 0$, $t_j > 0$, and $t_i \neq t_j$, SLOS is strictly faster. In the worst case, the time complexity remains $O(n2^n)$.*

Proof.

$$\begin{aligned}
\sum_{i=1}^{m} t_i \prod_{j\neq i}(t_j+1) &= \sum_{t_i\neq 0} t_i \prod_{j\neq i}(t_j+1) \\
&\leq \sum_{t_i\neq 0} t_i \frac{1}{\min_{t_l\neq 0}(t_l+1)} \prod_{j=1}^{m}(t_j+1) \\
&= \left(\frac{\prod_{j=1}^{m}(t_j+1)}{\min_{t_l\neq 0}(t_l+1)}\right) \sum_{t_i\neq 0} t_i \\
&= n\left(\frac{\prod_{j=1}^{m}(t_j+1)}{\min_{t_l\neq 0}(t_l+1)}\right)
\end{aligned}$$

That bound is strict whenever there are two $t_i > 0, t_j > 0$ such that $t_i \neq t_j$. The worst case is for $|t\rangle = |1,\ldots,1,0,\ldots,0\rangle$, where $\sum_{i=1}^{m} t_i \prod_{j\neq i}(t_j+1) = \sum_{i=1}^{n}\prod_{j=1}^{n-1} 2 = n2^{n-1}$.

□

To give a more refined comparison, we can compare both permanent-based methods and one-output SLOS for computing the full distribution.

Proposition II.21 (Full output computation with permanent-based method and one-output SLOS)**.** *For computing every single amplitude output independently with n-photon and m-mode states, SLOS is faster than permanent-based method by a ratio of $\frac{m}{2m+n-1}$.*

Proof. For each state $|t^{(k)}\rangle \in \mathcal{F}_m^k$, for each $u_{i,*}$ with $i \in [m]$, the term $u_{i,*}\langle t|\hat{U}|s\rangle^k$ will be used as many times as $|t_1,\ldots,t_i+1,\ldots,t_m\rangle^{k+1} \leq |\mathcal{M}^{(n)}\rangle$ with $|\mathcal{M}^{(n)}\rangle \in \mathcal{F}_m^n$. We can show that the number of terms is $\binom{(n-k-1)+m-1}{m-1}$, which is the number of putting $n-k-1$ photons into m modes, as $|\mathcal{M}\rangle$ is necessarily of the form $|t_1+q_1,\ldots,t_i+1+q_i,\ldots,t_m+q_m\rangle$ with $q_i \geq 0$ and $\sum_i q_i = n-k-1$.

Thus, we can show the total number of operations for computing all outputs is:

$$\sum_{k=0}^{n-1} m\binom{k+m-1}{m-1}\binom{(n-k-1)+m-1}{m-1} = m\binom{2m+n-2}{n-1} = n\binom{2m+n-1}{n}\frac{m}{2m+n-1}$$

□

Even if SLOS is better in time than the permanent-based method, it is important to highlight it is far more costly in memory.

n	$m = n$	$m = 2n$	$m = 3n$
12	1.35×10^6	8.34×10^8	5.23×10^{10}
14	$2,01 \times 10^7$	3.52×10^{10}	4.35×10^{12}
16	3.01×10^8	1.5×10^{12}	3.66×10^{14}
17	1.17×10^9	9.85×10^{12}	3.37×10^{15}
18	4.54×10^9	6.46×10^{13}	3.11×10^{16}
20	6.89×10^{10}	2.79×10^{15}	2.65×10^{18}

Table II.2 – Value of $\binom{n+m-1}{m-1}$ for practical combinations of m and n.

	time necessary for single instructions processed on a 1GHz computer	equivalent storage size (any unit)	number of bytes necessary to store a pointer
10^6	1 milliseconds	0.95 Mega	3
10^8	0.1 seconds	95 Mega	4
10^{10}	10 seconds	9.3 Giga	5
10^{12}	17 minutes	903 Giga	5
10^{14}	27 hours	90 Tera	6
10^{16}	115d	8.8 Peta	7
10^{18}	31 years	888 Peta	8

Table II.3 – Equivalency of 10^k with concrete time and storage references

Proposition II.22 (Memory cost of one-output SLOS)**.** *Given one n-photon input $|s\rangle$, a generic $m \times m$ unitary U and an output $|t\rangle$, SLOS computes $\langle t|\hat{U}|s\rangle$ by storing $O\left(\binom{n}{\lfloor n/2 \rfloor}\right)$ complex coefficients in the worst case.*

Proof. Given an output state $|t\rangle$, at the step k, we need to store all the coefficients $\langle t^{(k-1)}|\hat{U}|s^{(k-1)}\rangle$ and $\langle t^{(k)}|\hat{U}|s^{(k)}\rangle$, with $|t^(k-1)\rangle \leq |t^{(k)}\rangle \leq |t\rangle$. The worst case is when $|t\rangle$ has at most one photon per mode, e.g. $|t\rangle = |1, 1, \ldots, 1, 0, \ldots, 0\rangle$. In that case, the space needed at each step is $O\left(\binom{n}{k}\right)$. As we only need to store the values of two steps, and not all the n steps, we can only consider the costliest step which is when $k = n/2$. The worst case space complexity is therefore $O\left(\binom{n}{\lfloor n/2 \rfloor}\right)$. □

II.2.3 Implementation

In this section, we discuss the concrete implementation of SLOS, that is coded in Perceval software [Heu+23a] (Section IV.2).

To build intuition, we provide orders of magnitude for the quantities involved in Table II.2. Additionally, Table II.3 offers a quick comparison of these quantities in terms of time and storage. Given the exponential growth of these values, our challenge is to optimize memory structures and implementation details as much as possible, gaining practical efficiency improvements that extend strong simulation to a few more photons.

In this section we assume that we have a fixed number of modes m. We will first describe the memory structure we have been introducing for the implementation of the algorithms. We then discuss the implementation challenges and the optimization tricks, and finally add a note on masking.

A. Memory Structures

The two main memory structures will be labeled as `fsarray` (Fock State Array) and `fsimap` (Fock State Inverse Map). `fsarray` is mainly used to get indices used for building `fsimap`. Both structures depend only on m and n and are independent of the unitary matrix. During the actual simulation they will be used as read-only structures containing the "*calculation path*". It is thus possible to pre-compute these structures and serialize them to files. For the largest ones, it is also possible to use the files as a memory-map.

Indexes of the Fock states - `fsarray`**.** Each `fsarray`(k) is an array in charge of representing all the Fock states $|s^{(k)}\rangle$ and assigning a unique ID to each of them. We want this structure to be searchable, and to have the least memory footprint. For this we use sequences $S(|s^{(k)}\rangle) = (p_1, ..., p_k)$ where p_i is the position of the i^{th} photon in the Fock state $|s^{(k)}\rangle$. For instance: $S(|1,0,0,2\rangle) = (1,4,4)$, which we denote `ADD`, mapping each number to an uppercase letter for the sake of readability.

This representation of Fock states, which we call *photon to mode* encoding, has the following nice properties:

1. It is ordered using an intuitive lexicographical order on the $(p_1, ..., p_k)$: e.g.

   $$S(|1,0,0,2\rangle) = \texttt{ADD} > S(|1,1,1,0\rangle) = \texttt{ABC}.$$

   Thus it is searchable.
2. The sequence $S(|s\rangle)$ is iterable: $(p_1, ..., p_k) \xrightarrow{+1} (p'_1, ..., p'_k)$, where we can simply increment as follows: (a) $p'_k = p_k + 1$ if $p_k < m$, otherwise (b) we calculate p_{k-1} the same way (possibly going to $p_{k-2}, \ldots, p_1$) and $p_k = p_{k-1}$, e.g. $(1,4,4)/\texttt{ADD} \xrightarrow{+1} (2,2,2)/\texttt{BBB}$.
3. The sequence $S(|s\rangle)$ can be represented in memory with a fixed $\lceil \log_2(m) \rceil$-bit buffer.
4. It is more compact than *mode to photon* encoding for $n < m$. This case is the most common use case with circuits with single photon sources, as each input mode would have at most one photon.

Properties 1 and 3 imply that, by binary search, it is possible to find a specific Fock state in a `fsarray` in $O(\log_2 M_k)$. Property 2 implies that it is even possible to not store the `fsarray` if we are interested in all the Fock states since we can directly iterate through them. Thus, we will only store `fsarray`(n), as the computation of the states `fsarray`$(k+1)$ will only need the iteration of `fsarray`(k) and the structure `fsimap` as detailed below.

Mapping between Fock States - `fsimap`**.** Our algorithms require that we store a map between the "child" Fock states of $k+1$ photons $|s^{(k+1)}\rangle$ to the "parent" Fock states of k photons $|s^{(k)}\rangle$ (Figure II.3). We call this structure `fsimap`$(k+1)$ and we need to build a list of n `fsimap`, `fsimap`(n), `fsimap`$(n-1)$, ..., `fsimap`(1).

For a Fock state to find its parent, we just need to remove one photon. For instance for `ADD` we just remove a `A` or a `D` to reach `AD` or `DD`. So more generally for each M_{k+1} Fock state in $\mathcal{F}_m^{k+1}$ we need to store up to $k+1$ indexes to the parent. Most of the time we could store less, but we prefer keeping direct access to each entry in the `fsmap`, so we are keeping for each state exactly $k+1$ pointers. The size of a layer is then: $S_{\texttt{fsimap}(k)} = \texttt{fsimap}(k)$ is $\lceil \log_{256}(M_{k-1}) \rceil \times k$ bytes.

To summarize, starting from all the states in $\mathcal{F}_m^{k+1}$, we can find their index $(index)$ thanks to `fsa`$(k+1)$. With the index, we can find the location of the index to the parent states in `fsa`(k) by direct memory access in `fsimap`$(k+1)$ to the position: $index \times (k+1) \times S_{\texttt{fsimap}(k)}$ [3].

The construction of `fsimap` follows directly from the structure of the two involved `fsarray`.

3. Note that we can get a more compact version of the `fsimap`(k) for $k < n$ when the size of the index to each state is 4-bytes long. Indeed, in that case, we can then get rid of the non-used cells for the parents

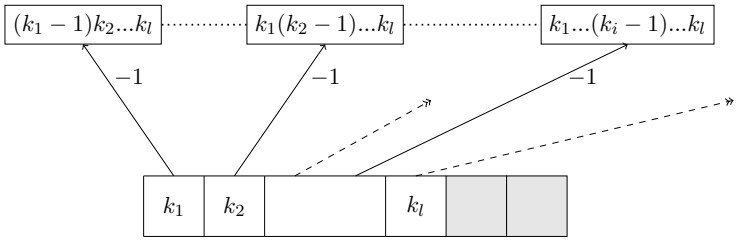


Figure II.3 – Representation of a cell of a `fsimap`(n): here the n photons are occupying l mode, so we have $n-l$ pointers not used

B. Implementation Optimization

All the implementation is done in C++ optimized with SIMD (Single Instruction, Multiple Data) vectorization. We detail the following points.

1. Vectorization: Cross-platform SIMD vectorization is done through MIPP [Cas+18] library [4] supporting SSE, AVX, AVX-512 and ARM NEON (32-bit and 64-bit) instructions. The main operation that we are interested in is "horizontal vector sum of complex number multiplication". This operation is not a primitive and is decomposed into the 3 MIPP primitives: `interleave`, `cmul`, `hadd`.
2. Iterative: Algorithm 1 is fully iterative. For $k=1$ to n the coefficients corresponding to $|s^{(k+1)}\rangle$ are computed in one single loop by iterating on the `fsimap`(n) structure. Each new coefficient is a sum of products of unitary matrix coefficient times coefficient of $|s^{(k)}\rangle$. There is no overhead in the implementation.
3. Recursion: SLOS (Algorithm 3) is recursive. Starting from the index retrieved by binary search on `fsaarray` on layer n, the expected coefficient (the probability amplitude) is calculated by recursively retrieving the coefficients for all the parents at level $k-1$. To reduce overhead caused by recursion, a bit vector is used to check if the parent coefficient is not yet calculated before going recursive. Also, all the structures necessary for Vectorization are pre-allocated globally to reduce overhead created by dynamic allocation on the stack.
4. Memory access: One major challenge for SLOS resides in memory allocation. When pushed to the maximum (say, 17 photons for 34 modes) - memory structures will use several hundreds/thousands of gigabytes in memory. Each coefficient calculation needs parent coefficient scattered on this memory range limiting possibility of processor to benefit from hardware L_x memory caching. Compared to local permanent calculation algorithm, this memory access time creates overhead depending strongly on hardware configuration as seen in Section II.2.4.
5. Multithreading:
   - Algorithm 1 can be fully run on multiple threads without additional overhead: for each k, we simply divide the coefficients list by the number of threads.
   - For Algorithm 3,multi-threading is more limited: it is only possible to distribute top branches of the calculation.

   In both cases, impact of the multi-threading is actually limited by memory access: adding more threads on very large memory structure do not significantly increase the performance.

pointers. We then lose the ability the direct access through the `fsa`(k) index, but this index is only used when referred to by child structure, so we can instead replace the index by the direct pointer to the position of the state in memory.

4. https://github.com/aff3ct/MIPP

	Number of photons n									
	2	3	4	5	6	7	8	9	10	11
M_n	3	10	35	126	462	1716	6435	24310	92378	352716
Permanent-Based	36s	3m	16m	1h30	11h	3d	*not possible*			
SLOS	14s	15s	29s	73s	2m	5m	22m	1h45m	10h	2d8h

Table II.4 – In this table[6], we compare time of QML algorithm of [GLA22] to perform 150 optimization iterations with both Permanent-Based and SLOS algorithms. This application is the ideal use case of SLOS since we are interested into the full output states exact distribution.

C. Masking

Let us note that the implementation proposed is fully compliant with the notion of "masking" introduced in Definition II.10 and Proposition II.12: technically masking is a transversal optimization allowing to reduce (potentially massively) the complete Fock state space and therefore reducing proportionally the time and storage.

The only location in the code impacted by `masking` is on the `fsaarray` construction. To find all the masked items, we are keeping the same global iterations on all possible Fock states, but are skipping the ones that are not compliant with the mask. It is likely that we could find a faster iteration method on masked Fock states, however, since this only impacts pre-computing, we don't need a special optimization.

Once the pre-computed memory structures detailed below are built, there is no difference in implementation when simulating on a masked or non-masked system.

II.2.4 Benchmark

In this section, we discuss practical aspects of SLOS. We first show a typical use case demonstrating the need of strong simulations and effectiveness of our implementation (Section II.2.4.A.). We then compare the performance of SLOS for a single output compared to the state-of-the-art matrix permanent computation (Section II.2.4.B.). Finally, we analyze the space limitation of SLOS (Section II.2.4.C.).

A. Quantum machine learning application requiring strong simulation

Using the framework Perceval [Heu+23a], we have implemented[5] the simulation of [GLA22], using a generic $m \times m$ interferometer and its full output probability distribution to train a model solving differential equations. For an evolving configuration of the circuit, the algorithm has to iterate on all n-photon output states $|t\rangle$ corresponding to the n-photon input state $|1, 1, \ldots, 1\rangle$. Based on Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimiser [Fle13], the algorithm converges in 200-400 iterations, each of them needing thousands of full distribution computations. Table II.4 shows the evolution of M_n for different values of n and the time necessary for 150 iterations of the algorithm comparatively with the direct calculation of M_n permanent with Glynn's algorithm and SLOS algorithm. Use of Algorithm 1 practically allows pushing of simulation from 6 photons (processed in 2min with Algorithm 1 and 11h without) to 10 photons.

5. Details of the algorithm can be found at https://perceval.quandela.net/docs/notebooks/Differential%20equation%20solving.html.

6. Note that this simulation has been performed in early 2022. We expect a bigger gap between the two methods with the current implementations.

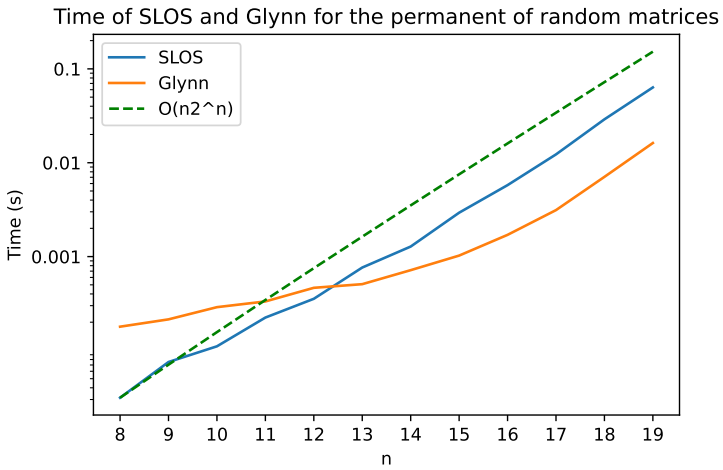


Figure II.4 – Comparative performance of Glynn and SLOS for calculation of a single $n \times n$ permanent 128-bit complex numbers. The green curve is the $n2^n$ curve renormalised with the first point of the SLOS curve. Benchmark computes 100 permanents in a row and outputs the mean time. The benchmark is ran on a Intel Core i7-10510U with 16GB of memory.

B. Benchmarking SLOS for One Output

We compare in Figure II.4 the speed of our algorithm to compute a single permanent in the worst possible situation, i.e. when the output is $|1, \ldots, 1\rangle$, with a traditional permanent calculation algorithm. We select Glynn's algorithm [Gly10] because we show later (cf Figure IV.7 in Section IV.2) that up to 19 modes, this algorithm is practically the most efficient.

The curves do not show the precompilation time needed for SLOS nor the allocation of the Fock states[7]. We can see that the practical time for SLOS is better for small cases, and that, as predicted by the complexity analysis, the growth is very close to the $n2^n$ curve. This is even true for small instances.

The code for Glynn's algorithm developed in Perceval makes heavy use of hardware optimizations, and shows a speedup compared to SLOS when $n > 12$. The speedup has a multiplicative factor between 2 and 4 depending on the machine. On Figure II.4, the factor is around 4 for large values of n. Indeed, as the data in Glynn's algorithm is local, one can rely on efficient libraries to sumprod the terms, improving the time of the computations. Due to the structure of the algorithm (because of the heavy use of pre-computed data), SLOS cannot make use of these optimizations. One has however to note that without these hardware-specific optimizations, there is no particular speedup for Glynn's algorithm compared to SLOS.

C. Memory Usage

Finally, since SLOS is making intensive use of memory, we have computed in Figure II.5 the practical limitations of strong simulation. The graphics vertical axis is the number of modes m, and the horizontal axis the number of photons n. We are only interested in $m \geq n$, and we have drawn boundaries of three typical workstations. In blue, computer with up to 8Gb of memory — typically any modern laptop. In orange, the limit of 256 Gb memory — typically a large compute node. Any (m, n) point below the orange curve will fit in 256 Gb memory. The red curve represents memory need up-to 4 Terabytes

7. This takes a significant amount of time but can only be done once and does not have to be computed again.

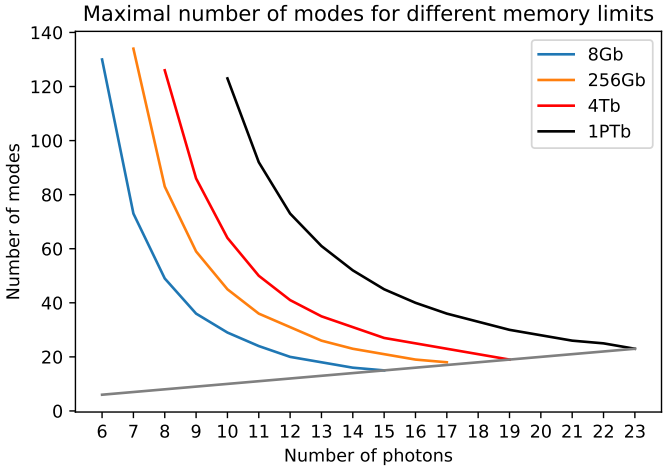


Figure II.5 – This graph shows four regions for (m, n) combinations. In blue - $m \geq n$ with configurations that can run on a personal laptop (up to 8Gb memory). Curves in orange/red/-black are the respective limits of 256Gb/4Tb/1Pb memory. For instance, strong simulation with SLOS of 24 photons on a 24 modes circuit would require $1500Tb$ of processing memory!

memory representing a very large HPC node. The black curve represents a potential super-computer with up to 1 Petabyte of memory. As of today technology this boundary can be considered as an area unreachable for strong simulation, and so for a full description of linear optical processes.

II.2.5 Extension to feedforward and hybrid simulation

A. Feedforward

In this section, we first explain how we can perform the simulation of adaptive schemes pictured in Figure II.2. Then we detail both its time and memory complexity.

Definition II.23 (ℓ-mode mask)**.** Given a ℓ-mode state $|\mathcal{M}\rangle$ and a m-mode state $|t\rangle$, we define the following relation:

$$|t_1, \ldots, t_m\rangle \leq_\ell |\mathcal{M}_1, \ldots, \mathcal{M}_\ell\rangle \iff \bigwedge_{i=1}^{\ell} (t_i \leq \mathcal{M}_i)$$

We say that $|\mathcal{M}\rangle$ is a ℓ-mode mask.

Proposition II.24 (Simulation of ℓ-adaptive measurements with SLOS)**.** *The full output amplitude distribution of a linear optical circuit with generic ℓ-adaptive measurements can be computed by performing SLOS on all possible ℓ-mode masks.*

Proof. Let $\hat{C}$ the circuit constituted of ℓ-adaptive measurements as pictured in Figure II.2. For any $|t\rangle \in \mathcal{F}_m^n$ we have:

$$\langle t| \hat{C} |s\rangle = \langle t_1, \ldots, t_\ell, t_{\ell+1}, \ldots, t_m| \hat{U}(t_1, \ldots, t_\ell) |s\rangle$$

where $\hat{U}(t_1, \ldots, t_\ell)$ is a usual linear optical circuit without adaptivity. In other words, we can compute all coefficients by enumerating all different configurations for the first ℓ modes:

$$\left\{ \langle t| \hat{C} |s\rangle \middle| |t\rangle \in \mathcal{F}_m^n \right\} = \bigsqcup_{d=0}^{n} \bigsqcup_{|\mathcal{M}\rangle \in \mathcal{F}_\ell^d} \left\{ \langle \mathcal{M}, t_{\ell+1}, \ldots, t_m| \hat{U}(\mathcal{M}) |s\rangle \middle| |t_{\ell+1}, \ldots, t_m\rangle \in \mathcal{F}_{m-\ell}^{n-d} \right\}$$

where d represents the number of photon of the mask $|\mathcal{M}\rangle$. It remains to notice that computing:

$$\left\{ \langle \mathcal{M}, t_{\ell+1}, \ldots, t_m| \hat{U}(\mathcal{M}) |s\rangle \middle| |t_{\ell+1}, \ldots, t_m\rangle \in \mathcal{F}_{m-\ell}^{n-d} \right\}$$

is the same as performing SLOS with the ℓ-mode mask $\mathcal{M}$ and fixing $\hat{U}$ as $\hat{U}(\mathcal{M})$. □

Proposition II.25 (Time complexity for simulating feedforward with SLOS)**.** *The full output amplitude distribution of a m-mode linear optical circuit with generic ℓ-adaptive measurements and an n-photon input state can be computed with:*

$$\sum_{s=1}^{n} \sum_{d=s}^{n} 2s \times \binom{n-d+\ell-\delta_{\ell m}}{n-d} \times \binom{m}{s} \times \binom{d-1}{d-s} \tag{II.6}$$

complex operations, where $\delta_{\ell m} = 1$ if $\ell = m$ and 0 otherwise.

Proof. For a given ℓ-mode mask $|\mathcal{M}\rangle$, let us first note that all the coefficients of interest are the states which both satisfy the mask condition and that can *reach* that mask at the end of the computation. For instance, let $(m, n) = (3, 4)$ and $|\mathcal{M}\rangle = |1, 1\rangle$. Even though $|0, 0, 3\rangle \leq_\ell |1, 1\rangle$, there is no need to compute $\langle 0, 0, 3| \hat{U} |s\rangle$, since it cannot be involved in the computation of $\langle 1, 1, t| \hat{U} |s\rangle$ for $0 \leq t \leq 2$. Mathematically, we compute only coefficients satisfying $|t\rangle \leq_\ell |\mathcal{M}\rangle$ and $\sum_{i=1}^{\ell} (\mathcal{M}_i - t_i) \leq n - \#|t\rangle$. For the following, we only refer to states that can reach the mask.

As detailed in the proof of Proposition II.19, each coefficient $\langle t| \hat{U} |s\rangle$ needs $2\alpha_t$ complex operations to be computed, where we remind that $\alpha_t = \#\{t_i \neq 0, 1 \leq i \leq m\}$. The total number of operations is therefore:

$$\sum_{\substack{\text{all mask } |\mathcal{M}\rangle \\ \text{on } \ell \text{ modes}}} \sum_{\substack{\text{states } |t\rangle \text{ that} \\ \text{can reach } |\mathcal{M}\rangle}} 2\alpha_t$$

By summing on the $|t\rangle$ instead, we have the following equality:

$$\sum_{\substack{\text{all mask } |\mathcal{M}\rangle \\ \text{on } \ell \text{ modes}}} \sum_{\substack{\text{states } |t\rangle \text{ that} \\ \text{can reach } |\mathcal{M}\rangle}} 2\alpha_t = \sum_{\text{all states } |t\rangle} 2\alpha_t \left(\sum_{\substack{\text{all } \ell\text{-mode masks} \\ \text{reachable by } |t\rangle}} \right)$$

The number of masks reachable by a state $|t\rangle$ essentially depends on $d = \sum_{i=1}^{m} t_i$. Indeed, any way of distributing between 0 and $n - d$ photons over the ℓ modes leads to a different mask. Thus, there are

$$\sum_{k=0}^{n-d} \binom{d+\ell-1}{\ell-1} = \binom{n-d+\ell}{n-d}$$

reachable masks. The only exception is the case $\ell = m$ where only masks with n photons are accepted as there is no mode left. In this case we need to distribute exactly $n - d$ photons over the ℓ modes and there are $\binom{n-d+\ell-1}{n-d}$ ways of doing it. It leads to the following time complexity:

$$\sum_{\text{all states } |t\rangle} 2\alpha_t \binom{n-d+\ell-\delta_{\ell m}}{\ell}$$

where $\delta_{\ell m} = 1$ if $\ell = m$ and 0 otherwise.

To obtain an explicit formula, we chose to iterate over the different values s of α_t, that range from 1 to n. Following the definition of $s = \#\{t_i \neq 0, 1 \leq i \leq m\}$, and given number of photons d, there are necessarily at least one photon in each of the s modes, and zero photon elsewhere. There are $\binom{(d-s)+s-1}{d-s} = \binom{d-1}{d-s}$ ways to distribute the remaining $d-s$ photons into the s modes. For m modes, there are $\binom{m}{s}$ different states $|t\rangle$ such that $\alpha_t = s$. The total number of operations is therefore:

$$\sum_{s=1}^{n}\sum_{d=s}^{n} 2s \times \binom{n-d+\ell-\delta_{\ell m}}{n-d} \times \binom{m}{s} \times \binom{d-1}{d-s}.$$

□

Proposition II.26 (Memory complexity for simulating feedforward with SLOS)**.** *The full output amplitude distribution of a m-mode linear optical circuit with generic ℓ-adaptive measurements and an n-photon input state needs to store at most*

$$\max_{p=1\ldots n} \left(1+\frac{p}{k}\right)^k \times \binom{m-k+n-p-1}{n-p}$$

complex coefficients.

Proof. As we iterate over the measurement outcomes, and each computation is independent of the others, the memory complexity is given by the maximum memory needed for one call to SLOS with mask. In other words, the memory complexity is

$$\max_{\substack{\text{mask } |\mathcal{M}\rangle \\ \text{on } \ell \text{ modes}}} \#\{\text{all states } |t\rangle | |t\rangle \leq_\ell |\mathcal{M}\rangle\}$$

For a given mask $|\mathcal{M}\rangle$, the cardinality is given by

$$\prod_{i=1}^{k}(\mathcal{M}_i+1) \times \binom{n-p+m-\ell-1}{n-p}$$

where $p = \sum_{i=1}^{k} \mathcal{M}_i$. Indeed, there are $\prod_{i=1}^{k}(\mathcal{M}_i+1)$ states satisfying $|t_1,\ldots,t_\ell\rangle \leq |\mathcal{M}_1,\ldots,\mathcal{M}_\ell\rangle$, and then there are $\binom{m-\ell+n-p-1}{n-p}$ ways to redistribute the $n-p$ remaining photons in the $m-\ell$ remaining modes.

To optimize over the mask $|\mathcal{M}\rangle$, we first rewrite the memory complexity as:

$$\max_{p=1\ldots n} \max_{\substack{\text{mask } |\mathcal{M}\rangle \\ \text{on } \ell \text{ modes} \\ \text{with } p \text{ photons}}} \prod_{i=1}^{\ell}(\mathcal{M}_i+1) \times \binom{n-p+m-\ell-1}{n-p}.$$

By using the inequality of arithmetic and geometry means, we have:

$$\frac{p+\ell}{\ell} = \frac{\sum_{i=1}^{\ell}(\mathcal{M}_i+1)}{\ell} \geq \left(\prod_{i=1}^{\ell}\mathcal{M}_i+1\right)^{\frac{1}{\ell}}$$

giving therefore an upper bound of the following quantity:

$$\max_{\substack{\text{mask } |\mathcal{M}\rangle \\ \text{on } \ell \text{ modes} \\ \text{with } p \text{ photons}}} \prod_{i=1}^{\ell} (\mathcal{M}_i + 1) \leq \left(1 + \frac{p}{\ell}\right)^k.$$

Therefore, the memory complexity is bounded by:

$$\max_{p=1\ldots n} \left(1 + \frac{p}{k}\right)^k \times \binom{m-\ell+n-p-1}{n-p}.$$

□

B. Generalized procedure for both weak and strong simulations

Algorithm 4 `SLOS_hyb` with one n-photon input $|s\rangle$ and `Select` function.
Each coefficient $\langle t|\hat{U}|s\rangle$ is stored in a dictionary or an array $\hat{U}$ and will be accessed by $\hat{U}[|s\rangle][|t\rangle]$. The access of the coefficient $u_{i,p}$ is labelled as $U[i][p]$. Every coefficient of $\hat{U}$ is initialized to 0. For simplicity, the array $\hat{U}$ is of size $O\left((\frac{n}{m}+1)M_n\right)$. The `Select` function enables strong simulation without or with masks, weak simulation, or a mix of them. $\mathcal{B}$ is an array or dictionary initialized to False and stores the coefficients already computed, so we don't need to recompute $\hat{U}[|s\rangle][|t\rangle]$ if

Global $\hat{U}$, $\mathcal{B}$
procedure $\mathrm{SLOS_Rec}(k, \mathcal{I}_S, \mathcal{F}_S^k, U)$
  $(|s\rangle, p) \leftarrow \mathcal{I}_S[k]$
  if $k > 1$ **then**
    $\mathrm{SLOS_Rec}(k-1, \mathcal{I}_S, \mathcal{F}_S^{k-1} \cap \overline{\mathcal{B}}_s, U)$ ▷ Build with Equation (II.5)
  end if
  for $|t\rangle \in \mathcal{F}_S^k$ **do**
    for $i \in [m]$ **when** $t_i \neq 0$ **do**
      $\hat{U}[|s_1, \ldots, s_p + 1, \ldots, s_m\rangle][|t\rangle]$
        $+= \sqrt{\frac{t_i}{s_p}} \times U[i][p] \times \hat{U}[|s\rangle][|t_1, \ldots, t_i - 1, \ldots, t_m\rangle]$
    end for
    $\mathcal{B}[|s\rangle][|t\rangle] \leftarrow$ True
  end for
end procedure
procedure $\mathrm{SLOS_HYB}(|s^{(n)}\rangle, U, \texttt{Select})$
  $\hat{U}[|0\rangle][|0\rangle] \leftarrow 1$
  $\mathcal{I}_S \leftarrow \left[(|vac\rangle, p_1), \ldots, (|s^{(n-1)}\rangle, p_n)\right]$ ▷ Chosen randomly from Equation (II.1)
  $\mathcal{F}_S^k \leftarrow \{|vac\rangle\}$
  for $k = 1$ **to** n **do**
    $\mathcal{F}_S^k \leftarrow \mathrm{SELECT}(\mathcal{I}_S, \mathcal{F}_S^k, k)$
    $\mathrm{SLOS_Rec}(k, \mathcal{I}_S, \mathcal{F}_S^k, U)$
  end for
  return $\mathcal{F}_S^k$
end procedure

This section presents an extension of SLOS with both an iterative procedure and a recursive subroutine, enabling the simulation of various problems, including both weak and strong simulation.

Let us denote $\mathcal{F}_S^k$ the set of intermediary states computed at the step k with Equation (II.3). In Section II.2.1.A. we were interested in computing every states, so $\mathcal{F}_S^k = \mathcal{F}_m^k$, while in Section II.2.1.B. we had masks to compute specific outputs, so $\mathcal{F}_S^k = \mathcal{F}_{S_\mathcal{M}}^k$. For the hybrid version in Algorithm 4, we use a `Select` function to select the next intermediary states we would like to compute, so $\mathcal{F}_S^{k+1}$ is defined as $\texttt{Select}(\mathcal{F}_S^k)$. In order to compute all coefficients in $\mathcal{F}_S^{k+1}$, we use `SLOS_Rec` with the particularity to only compute intermediary states that have not been computed so far. We can code that information in a Boolean array (or a hash table) $\mathcal{B}$ such that $\mathcal{B}[|s\rangle][|t\rangle]$ is true if $\hat{U}[|s\rangle][|t\rangle]$ has been computed, false otherwise.

For strong simulations, we are aiming at computing the probabilities of a given set of outputs. However, sometimes we would like to compute the states with the largest probabilities. More generally, we would like to compute outputs giving conditions on the distribution even if we don't know it yet. For instance, Algorithm 4 can solve the following problem.

Problem II.27 (Hybrid simulation)**.** *Given an input $|s^{(n)}\rangle$, a threshold η and a m-mode linear optical circuit $\hat{U}$, output a set of state*

$$\mathcal{O} = \{|t\rangle, |t\rangle \in \mathcal{F}_m^n\}$$

such that

$$\sum_{|t\rangle \in \mathcal{O}} \left|\langle t| \hat{U} |s\rangle\right|^2 > \eta.$$

Even though this problem can be solved by computing the full distribution and then taking a correct subset, we can have a more efficient procedure by using the Algorithm 4 and the `Select` function of Algorithm 5. Note the "while loop" adds several states in $\mathcal{F}$.

Interestingly, by taking a different `Select` function, Algorithm 4 can perform weak simulation, inspired by the photon-by-photon sampling algorithm of [CC18] described in the proof of Proposition I.57. To be clearer and slightly more efficient, the commands in the for loop are inversed: we first compute the amplitudes with `SLOS_Rec`, and then we would use the following `Select` function to sample:

1. Given an input $|s\rangle$, the random order $s^(1) \leq s^(2) \leq \cdots \leq s^(n)$ is done by the random choice of $\mathcal{I}_S$. We initialise $\mathcal{F}_S^k = \{|1, 0, \ldots, 0\rangle, \ldots, |0, \ldots, 0, 1\rangle\}$.

2. The `Select` function at the step $1 \leq k \leq n$ and given a set of states $\mathcal{F}_S^k$, will sample $\left|q^(k)\right\rangle$ from $\left\{\left|\langle t| \hat{U} |s\rangle\right|^2 \middle| |t\rangle \in \mathcal{F}_S^k\right\}$. Then if $k < n$, it will return the next values to compute, i.e. the set $\{|q_1, \ldots, q_i + 1, \ldots, q_m\rangle, i \in [m]\}$. If $k = n$, then we are at the last step, so we just can return the last sample $\{\left|q^(n)\right\rangle\}$.

Algorithm 5 `Select` function which answers to Problem II.27.

procedure $\textsc{Select}(\mathcal{I}_S, \mathcal{F}_S^k, k)$
  $S \leftarrow 0$
  $(|s\rangle, \star) \leftarrow \mathcal{I}_S[k]$
  $\mathcal{F} \leftarrow \{\}$
  while $S \leq \eta$ **do**
    $|t\rangle \leftarrow \mathcal{F}_S^k$.pop()
    $S \mathrel{+}= |\langle t|\hat{U}|s\rangle|^2$
    $\mathcal{F}.add(|t\rangle)$
  end while
  return $\{|t_1, \ldots, t_i + 1, \ldots, t_m\rangle \mid |t\rangle \in \mathcal{F}, i \in [m]\}$
end procedure

II.3 LO-SLAP: Linear optical simulator through lattice of polynomials

In this section, we present LO-SLAP (Strong Linear Optical Simulator through LAttice of Polynomials). We begin with a detailed description of how LO-SLAP works (Section II.2.1). We further optimize the run time of LO-SLAP by reducing the size of the lattice with Steiner trees (Section II.3.2). Next, we analyze LO-SLAP's time and memory complexity (Section II.2.2) and describe an optimized implementation (Section II.3.4). We then benchmark LO-SLAP with SLOS for linear optical circuits, both with and without feedforward (Section II.3.5). Finally, we explore the application of LO-SLAP for noisy linear optical circuits (Section II.3.6).

Contributions

This section is based on the paper [GH25], coauthored with Timothée Goubault de Brugière. The benchmarks were primarily carried out by T. Goubault de Brugière. The remainder of the work presented in this section is the result of equal collaboration.

II.3.1 Algorithm

In this section, we restrict ourselves to the case where the input is fixed to the state with one photon in the first n modes:

$$|1, \ldots, 1, 0, \ldots, 0\rangle = (x_1 \ldots x_n)\,|vac\rangle$$

Therefore, the output state is always of the form

$$P(x) = \prod_{j=1}^{n} \left(\sum_{i=1}^{m} u_{ij} x_j \right) = \prod_{j=1}^{n} P_j(x) \tag{II.7}$$

where $P_j \coloneqq \sum_{i=1}^{m} u_{ij} x_j$.

Definition II.28 (Polynomial semi-lattice)**.** We consider a semi-lattice structure, as shown in Figure II.6. Each node is labeled with a monomial $x^s = x_1^{s_1} \ldots x_m^{s_m}$ and stores the partial derivative

$$\frac{\partial P}{\partial x^s}.$$

For simplicity, we will refer to this structure simply as a lattice.

The root is the factorized version of P, while the leaves, i.e. nodes of the last layer of the semi-lattice, correspond to the amplitudes we want to compute up to some known normalization coefficient. Each node has m children, given from it by differentiating [8] the associated polynomial by an $x_i, i = 1 \dots m$. A level k of the lattice corresponds to all nodes of partial derivatives of same degree k. We will interchangeably label one node either by its monomial x^s or the associated Fock state $|s\rangle$.

We can now describe how LO-SLAP works.

Definition II.29 (LO-SLAP: A depth-first search traversing algorithm)**.** LO-SLAP is an algorithm that performs a depth-first search on the lattice, as illustrated in II.7. The full output amplitude distribution is computed by extracting the coefficients of the leaves.

LO-SLAP is faster than directly computing the output coefficients independently, as the coefficients stored at each node (see Proposition II.30) help compute those of its children (see Proposition II.31), as we explain in detail here.

Proposition II.30 (Set C_s of a node x^s)**.** *A node x^s at level k of the lattice is fully characterized by the set of $\binom{n}{k}$ coefficients:*

$$C_s = \left\{ \mathrm{Per}\left(U_{s,[n]\setminus J}\right) \middle|\ J \in \mathcal{C}^n_{n-k} \right\}.$$

Proof. Each node x^s represents the partial derivative of P with respect to x^s. The proof follows directly from

$$\frac{\partial P}{\partial x^s}(x) = \sum_{J \in \mathcal{C}^n_{n-k}} \mathrm{Per}\left(U_{s,[n]\setminus J}\right) \prod_{j \in J} P_j(x) \tag{II.8}$$

where $\mathcal{C}^n_{n-k}$ is the set of combinations of $n-k$ elements in $[n]$.

□

Proposition II.31 (Computation of a child node from a parent node)**.** *For $k < n$, given a node $x^s, |s| = k$ and $x^{s'}, |s'| = k+1$ one of its children. The set $C_{s'}$ of the node $x^{s'}$ can be computed from C_s with*

$$(n-k) \times \binom{n}{n-k} \tag{II.9}$$

complex additions and multiplications. The procedure is illustrated in Algorithm 6.

Algorithm 6 Update coefficient after differentiating by x_i at level k

```
procedure UPDATE_COEFFICIENTS(U, x_i, k, C_s, C_{s'})
    for J ∈ C^n_{n-k} do
        c ← C_s[J]
        for l ∈ J do
            prod ← c × U[i, l]
            C_{s'}[J\l] ← C_{s'}[J\l] + prod
        end for
    end for
end procedure
```

8. Note that formalization with partial derivatives can also be used for continuous variables, see [YMQ24].

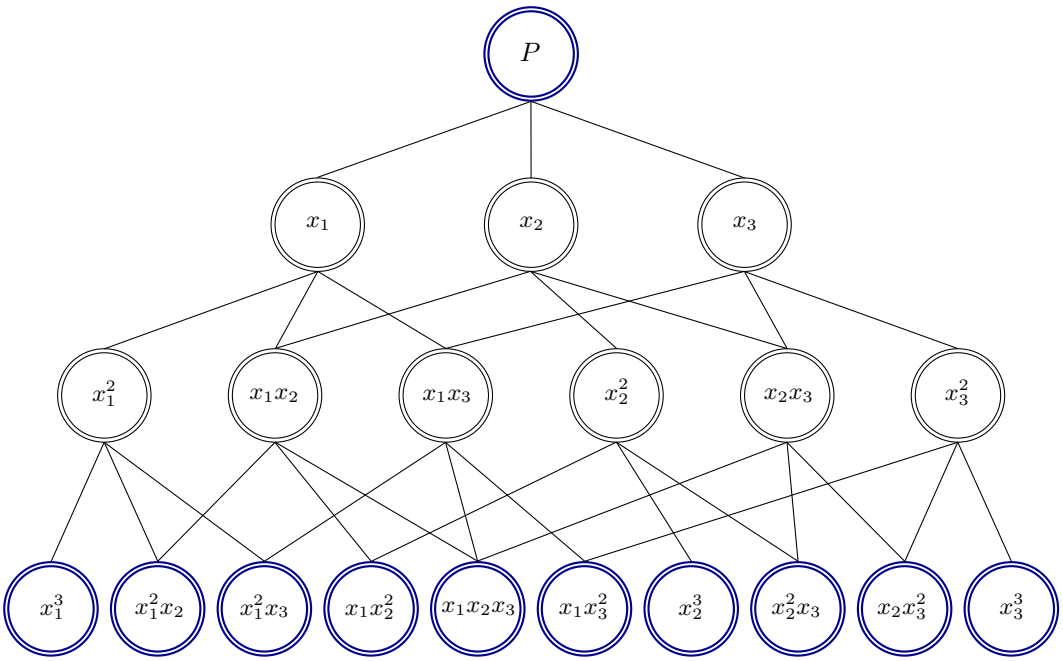


Figure II.6 – Example of the semi-lattice structure considered in our algorithm for 3 photons and 3 modes.

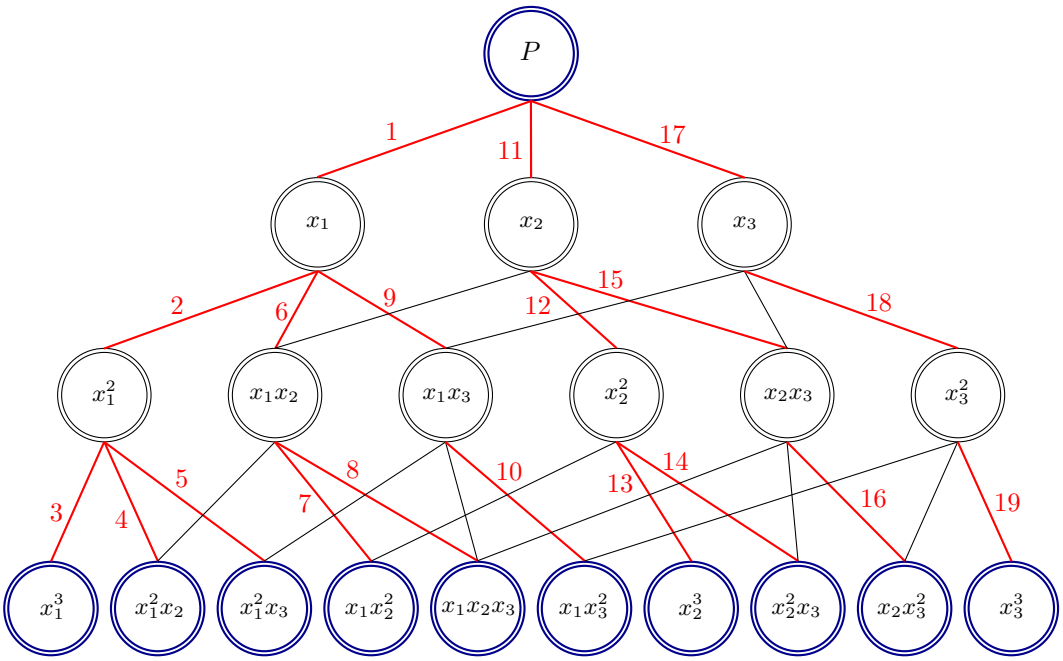


Figure II.7 – Illustration of the depth-first search traversal. Each red edge is labeled by its order in the traversal. Non-red edges are not used during the computation.

Proof. Let i be the entry where $|s\rangle$ and $|s'\rangle$ differ. We have

$$\frac{\partial P}{\partial x^{s'}} = \frac{\partial P}{\partial x_i}\frac{\partial P}{\partial x^s} = \sum_{J\in\mathcal{C}^n_{n-k}} \mathrm{Per}\left(U_{s,[n]\setminus J}\right)\frac{\partial}{\partial x_i}\prod_{j\in J} P_j(x).$$

Differentiating the product gives

$$\frac{\partial P}{\partial x^{s'}} = \sum_{J\in\mathcal{C}^n_{n-k}} \mathrm{Per}\left(U_{s,[n]\setminus J}\right)\sum_{\ell\in J}\frac{\partial P_\ell}{\partial x_i}\prod_{j\in J, j\neq\ell} P_j(x)$$

As $\frac{\partial P_\ell}{\partial x_i} = u_{i\ell}$ (cf Equation II.7), we have

$$\frac{\partial P}{\partial x^{s'}} = \sum_{J\in\mathcal{C}^n_{n-k}}\sum_{\ell\in J} \mathrm{Per}\left(U_{s,[n]\setminus J}\right)\cdot u_{i\ell}\prod_{j\in J, j\neq\ell} P_j(x). \tag{II.10}$$

Note that by equivalently summing over $(J\in\mathcal{C}^n_{n-k-1}, \ell\in[n]\setminus J)$ instead of $J\in\mathcal{C}^n_{n-k}$, we can prove Eq. II.8 by induction as follows.

$$\begin{aligned}\frac{\partial P}{\partial x^{s'}} &= \sum_{J\in\mathcal{C}^n_{n-k-1}}\left(\sum_{\ell\in[n]\setminus J}\mathrm{Per}\left(U_{s,[n]\setminus J\setminus\ell}\right)\cdot u_{i\ell}\right)\prod_{j\in J} P_j(x)\\ &= \sum_{J\in\mathcal{C}^n_{n-k-1}}\mathrm{Per}\left(U_{s',[n]\setminus J}\right)\prod_{j\in J} P_j(x)\end{aligned}$$

where the last equality holds by the Laplace expansion along the i-th row of U with columns $[n]\setminus J$ (see [MM65] for example).

Following Eq. II.10, as the P_j don't intervene, we only need to compute all the products

$$\mathrm{Per}\left(U_{s,[n]\setminus J}\right)\cdot u_{i\ell}$$

for $J\in\mathcal{C}^n_{n-k}$ and $\ell\in J$. Then we accumulate the products in the corresponding element of $C_{s'}$, as summarized in Algorithm 6. The index of the sets C_s and $C_{s'}$ are detailed in Section II.3.4. Therefore, we need

$$|J|\times|\mathcal{C}^n_{n-k}| = (n-k)\times\binom{n}{n-k}$$

complex multiplications and additions to compute $C_{s'}$.

□

II.3.2 Traversal time optimization with Steiner trees

The naive depth-first search we could use for the traversal through the lattice would not exploit the high connectivity of the lattice. If the goal is only to reach the leaves, more efficient traversals can be designed to avoid redundant nodes and save computational operations. For instance, still in our example of Fig II.6, we can skip the nodes x_1x_2 and x_2x_3 but still visit all the leaves, as described in Fig II.8. Finding a more efficient traversal would be very useful to reduce even further the total computational time.

The problem of finding the optimal traversal is closely related to the Steiner tree problem [HR92] for directed graph.

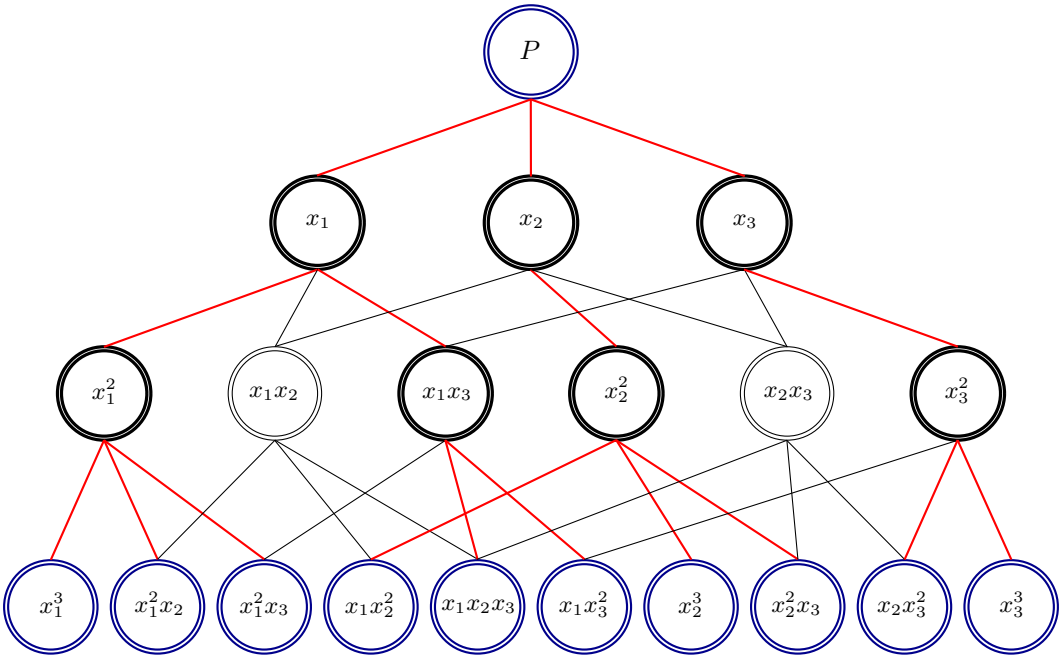


Figure II.8 – Optimized traversal of the graph. The nodes x_1x_2 and x_2x_3 are not visited anymore which will reduce the computational complexity.

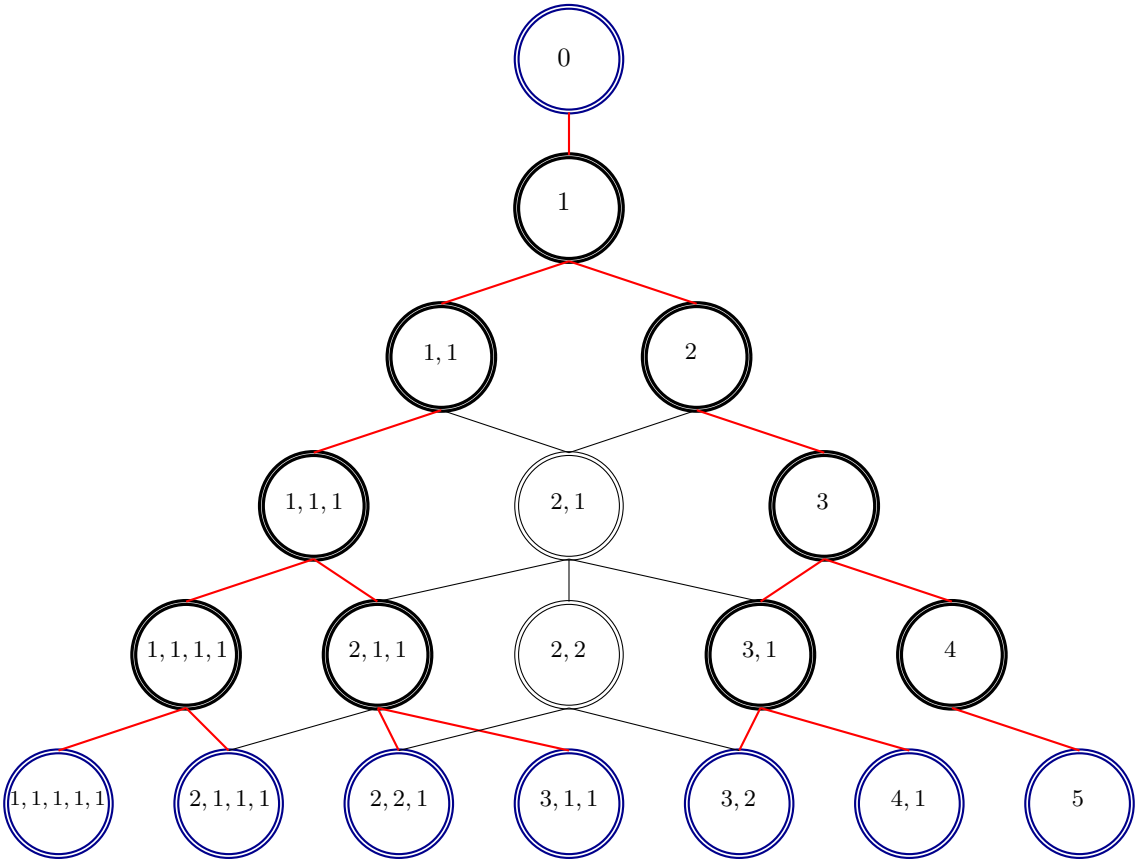


Figure II.9 – Reduced version of a lattice with 5 photons. Each node is now an equivalence class. The two nodes $(2,2)$ and $(2,1)$ not in bold are two nodes that will not be visited in an optimal traversal of the lattice with 5 photons. Note however that for 4 photons we need to visit these two nodes.

Definition II.32 (Steiner Tree Problem)**.** Given an undirected graph $G = (V, E)$ with weighted edges and a set of terminal nodes $S \subseteq V$, the Steiner tree problem consists in finding a minimum-weight tree in G that spans S. The weight of the tree is given by the sum of the weights of its edges.

A variant is the directed Steiner tree problem, where the graph G is oriented.

The lattice we consider has a natural interpretation as a directed graph, where each node is connected to its parents by a directed edge from the parent to the node. The weight of an edge is the number of complex operations required to compute the coefficients of the tail of the edge. Setting the root to the top node of the lattice and the terminals to the leaves, the solution of the associated directed Steiner tree problem would give the optimal traversal of the lattice.

Remark II.33 (Intractability of the Steiner tree problem for the full lattice). Solving the directed Steiner tree for the full lattice is both intractable and inefficient.

Proof. First, it is important to note that the generic Steiner tree problem is NP-hard [GGJ77]. Even if in the best case, we could hope for polynomial solutions for the specific kind of lattice, the main problem resides in the size of the full lattice.

Indeed, with n photons and m modes the total number of nodes in the lattice is

$$\sum_{k=0}^{n} \binom{k+m-1}{k} = \binom{n+m}{n}.$$

This is larger than the memory complexity of SLOS, that we know is almost optimal (Remark II.15) in time for the problem we are interested in. Not only we lose the memory advantage we are looking for with LO-SLAP, but most likely the computation of the Steiner tree will be very costly as well. It seems unlikely that computing a Steiner tree would provide any advantage for the full lattice. □

Instead of applying the Steiner tree problem on the full lattice, we propose to apply it on a reduced version.

Definition II.34 (Reduced Lattice)**.** At level k, each node represents a distinct partition of the integer k and encompasses all monomials that are equivalent up to a change of variables. The number of nodes at level k is therefore given by the number of partitions $p(k)$ of the integer k. The total number of nodes in this reduced lattice is

$$\sum_{k=0}^{n} p(k).$$

The weight of an edge is given by the cost of a node multiplied by the number of nodes in its equivalence class.

Example II.35. We illustrate a reduced lattice in Fig II.9 with $n = 5$ photons. In this reduced lattice, x_1^3, x_2^3 and x_3^3 are all merged into one node labeled (3). Similarly, x_1x_2, x_1x_3 and x_2x_3 are merged into a node of label $(1, 1)$. Therefore, each node is a canonical representative of an equivalence class of nodes in the original lattice. The new graph preserves the structure of the lattice: two nodes in the original lattice are connected only if their canonical representatives are connected in the new graph.

Remark II.36. Even for $n = 40$, which is way beyond what we could achieve, the total number of nodes is 215308. For comparison, the original lattice with $n = m = 40$ contains approximately 5×10^{22} nodes. Note also that the size of our new graph no longer depends on the number of modes.

	Memory	Time		
		gen.	$n = m$	$m \gg n$
Permanent-based	m^2	$n\binom{2m+n-1}{n}$	$n\frac{6.76^n}{\sqrt{\pi n}}$	$n2^n m^n$
SLOS	$\binom{n+m-1}{n}$	$n\binom{n+m-1}{n}$	$n\frac{4^n}{\sqrt{\pi n}}$	nm^{n-1}
LO-SLAP	2^n	See Eq. II.11	$n\frac{5.83^n}{\sqrt{\pi n}}$	nm^{n-1}

Table II.5 – Summary of the time and memory complexities of all methods. All complexities are $O()$ although not displayed for clarity. Best results are in blue, second-best results are in purple.

II.3.3 Complexity Analysis

All the complexities are summarized in Table II.5.

A. Full output

Proposition II.37 (Time complexity of LO-SLAP)**.** *LO-SLAP computes the full distribution of n-photon input state $|s_1, \ldots, s_m\rangle$ in*

$$2n \times \sum_{k=1}^{n} \binom{n-1}{k-1} \times \binom{m+k-1}{m-1} \tag{II.11}$$

operations.

Proof. Let us consider the full lattice as illustrated in Fig. II.6. The root is at level 0 of the lattice while the leaves are at level n. As described with Eq. II.9 (Proposition II.31), a node at level k of the lattice needs

$$2(n-k+1) \times \binom{n}{n-k+1} = 2n \times \binom{n-1}{k-1}$$

complex operations to be computed. Level k contains all k-photon m-mode Fock states. Therefore, the total number of complex operations of one level is

$$2n \times \binom{n-1}{k-1} \times \binom{m+k-1}{k}.$$

Summing over all possible levels give a total cost of

$$2n \times \sum_{k=1}^{n} \binom{n-1}{k-1} \times \binom{m+k-1}{m-1}$$

complex operations. □

To our knowledge, no known analytical formula exists for Equation (II.11), but in the case where $n = m$ we have

$$2n \times \sum_{k=1}^{n} \binom{n-1}{k-1} \times \binom{n+k-1}{n-1} \sim 2n \frac{(3+2\sqrt{2})^n}{2^{5/4}\sqrt{\pi n}} \approx O\left(n \frac{5.83^n}{\sqrt{\pi n}}\right).$$

When $m \gg n$, which happens in some boson sampling settings, we can approximate the complexity by

$$2n \times \sum_{k=1}^{n} \binom{n-1}{k-1} \times m^k \approx O(nm^{n-1}).$$

Comparison with the state of the art. By way of comparison, when $n = m$ the asymptotic complexity of SLOS is

$$O\left(n \frac{4^n}{\sqrt{\pi n}}\right)$$

and when $m \gg n$ the cost is

$$O(nm^{n-1}).$$

The permanent-based approach, when $n = m$ [CC24] has a complexity of

$$1.69^n \times \binom{n+m-1}{n} \sim 1.69^n \times \frac{4^n}{\sqrt{\pi n}} \approx O\left(\frac{6.76^n}{\sqrt{\pi n}}\right)$$

and when $m \gg n$ the cost is

$$n2^n m^{n-1}.$$

Proposition II.38 (Memory complexity of LO-SLAP)**.** *LO-SLAP needs to store $O(2^n)$ complex coefficients.*

Proof. In terms of memory, we need to store at most one node per level, for a total of

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n$$

complex numbers to store. □

B. Reduction in time complexity with Steiner traversal

To find a Steiner tree and provide numerical evidence that it could indeed improve traversal time, we use a heuristic in the SCIP-Jack package [RK23] which achieved top rankings at PACE 2018 [BS18].

We show in Table II.6 the theoretical gains we obtain after running the optimizations. We managed to get results up to 14 photons. Overall, we get significant savings in the total number of FLOPs, up to 90% savings, and the gains increase with the number of photons for similar ratios m/n.

Table II.6 – Summary of the theoretical gains obtained with the Steiner arborescence optimization. n is the number of photons, m the number of modes. In bold are shown the cases where more than 50% of FLOPs gain is achieved.

	Nodes in partition graph				Nodes in original graph			FLOPs		
n	Original	Optim.	Gain	m	Original	Optim.	Gain	Original	Optim.	Gain
				2	6	6	0%	20	20	0%
2	4	4	0%	4	15	15	0%	56	56	0%
				20	231	231	0%	920	920	0%
				3	20	20	0%	150	150	0%
3	7	7	0%	6	84	84	0%	624	624	0%
				30	39711	39711	0%	249240	249240	0%
				4	70	70	0%	1032	1032	0%
4	12	12	0%	8	495	495	0%	6448	6448	0%
				40	135751	135751	0%	1282800	1282800	0%
				5	252	222	12%	6810	5210	23%
5	19	17	10%	10	3003	2868	4%	64120	56920	11%
				50	3478761	3475086	0%	44715600	44519600	0%
				6	924	774	16%	43836	29436	33%
6	30	27	10%	12	18564	17112	8%	622896	488256	22%
				60	90858768	90649908	0%	1524786000	1505882400	1%
				7	3432	2585	25%	277550	141428	49%
7	45	36	20%	14	116280	99445	14%	5956664	3627792	39%
				70	2404808340	2391626465	1%	51221188760	49597362080	3%
				8	**12870**	**9034**	**30%**	**1736720**	**711248**	**59%**
8	67	53	21%	16	735471	626391	15%	56321120	33916640	40%
				80	64276915527	63987570127	0%	1702033764320	1662858005600	2%
				9	**48620**	**33518**	**31%**	**10771506**	**3557970**	**67%**
9	97	73	25%	**18**	**4686825**	**3868275**	**17%**	**527992920**	**256217184**	**51%**
				90	1731030945644	1719603126704	1%	56092374315180	53969620537620	4%
				10	**184756**	**123781**	**33%**	**66348900**	**17957700**	**73%**
10	139	97	30%	20	30045015	25232885	16%	4916825680	2498064880	49%
				100	46897636623981	45351094447131	3%	1836655872773600	1544330460713600	16%
				11	**705432**	**463531**	**34%**	**406441926**	**86382406**	**79%**
11	195	128	34%	**22**	**193536720**	**158971574**	**18%**	**45542059904**	**18400415836**	**60%**
				55	1074082795968	998028708078	7%	87690142188220	63328604101510	28%
				12	**2704156**	**1728918**	**36%**	**2478591000**	**429441480**	**83%**
12	272	176	35%	**24**	**1251677700**	**1025249324**	**18%**	**419980534176**	**143902408512**	**66%**
				60	15363284301456	14884985559986	3%	1583783064511080	1212435786358680	23%
				13	**10400600**	**6813874**	**34%**	**15058389450**	**2168746606**	**86%**
13	373	225	40%	**26**	**8122425444**	**6616339954**	**19%**	**3858768103216**	**1391546563836**	**64%**
				39	635013559600	558078517806	12%	161828613214632	85970855413170	47%
				14	**40116600**	**25664344**	**36%**	**91194804876**	**10195561460**	**89%**
14	508	293	42%	**20**	**1391975640**	**1028365800**	**26%**	**1690514536480**	**300732164320**	**82%**
				28	**52860229080**	**43509128034**	**18%**	**35343854172032**	**10686628719536**	**70%**

II.3.4 Implementation

In the depth-first search of LO-SLAP, the current node is represented by a stack, where each entry refers to one variable x_i, with possible repetition. Therefore, the size of the stack gives the degree of the current monomial. The global structure of the algorithm is given in Algorithm 7.

Algorithm 7 Core algorithm: LO-SLAP main loop and subroutines

```
1: procedure MAIN(U)
2:     flag ← 1
3:     stack ← []
4:     {C_s}_s ← []
5:     next_element_stack ← 1
6:     while flag > 0 do
7:         flag ← compute_next_node(U, stack, next_element_stack, {C_s}_s)
8:         if flag = 2 then
9:             s ← get_state_from_stack(stack)
10:            C_s[0] ← C_s[0]/√(s_1! ... s_m!)
11:            yield C_s[0]
12:        end if
13:    end while
14: end procedure
15:
16: procedure GET_STATE_FROM_STACK(stack)
17:    s ← [0]*m
18:    for i = 1 ... m do
19:        s[stack[i]] += 1
20:    end for
21: return s
22: end procedure
23:
24: procedure COMPUTE_NEXT_NODE(U, stack, next_element_stack, {C_s}_s)
25:    if next_element_stack > m or |stack| = n then
26:        if |stack| = 0 then return 0
27:        else
28:            a ← stack.pop()
29:            next_element_stack ← a+1  /* modified in place */
30:        end if
31:    else
32:        s ← get_state_from_stack(stack)
33:        stack.add(next_element_stack)
34:        s' ← get_state_from_stack(stack)
35:        update_coefficients(U, x_{stack[-1]}, |stack| − 1, C_s, C_{s'})
36:        if feed_forward then U ← update_U(stack)
37:        end if
38:    end if
39:    if |stack| = n then return 2
40:    else return 1
41:    end if
42: end procedure
```

A. Traversal through the full lattice: Implementation of Algorithm 6

The main bottleneck of the algorithm is the update of the coefficients, that was informally described in Section II.3.1. We recall that the update part consists in:

1. iterating over all coefficients of the parent node,
2. iterate over some entries of the j-th line of U,
3. multiply the current coefficient and the current entry of U,
4. add the result to one coefficient of the child node.

Fortunately we can efficiently store all the 2^n coefficients required in one vector of size 2^n. There is a natural mapping between the binary representation of the indices of the vector and the coefficients. Any Per $\left(U_{s,,[n]\setminus J}\right)$ can be associated to the n-bit integer b such that the i-th bit of b is 1 if i is in J. In other words, given v our vector of coefficients, $v[-1] = 1$, $v[0]$ contains the coefficients of the leaves and for instance[9] $v[13]$ is the coefficient in front of the product $P_1P_3P_4$.

Given a parent node at level k and x_ℓ the variable we want to differentiate the node with, the update part can be rephrased as:

1. iterate over all integers i of Hamming weight $n-k$,
2. iterate over the indices j of the nonzero bits of i,
3. set i_2 equal to i with the j-th bit set to 0,
4. update $v[i_2] \mathrel{+}= v[i] \times U[\ell, j]$.

This method can be compactly and efficiently implemented using bit twiddling hacks [And05]. We give the pseudocode in Algorithm 8. Most operations can be implemented in many programming languages, except maybe the built-in function ctz which is available in GCC and gives the number of trailing zeros in an integer starting from the least significant bit.

Algorithm 8 Implementation of Algorithm 6

```
1: procedure UPDATE_COEFFICIENTS(U, x_i, k, v)
2:     j ← 2^{n−k} − 1
3:     lim ← 2^n − 1
4:     while −1 < j < lim + 1 do
5:         v[j] ← 0
6:         j_2 ← j ⊕ lim   /* ⊕ stands for the bitwise XOR */
7:         while j_2 do
8:             p ← __builtin_ctz(j_2)
9:             j_2 ← j_2 ⊕ 2^p
10:            v[j] ← U[i, p] × v[j ⊕ 2^p]
11:        end while
12:        t ← j | (j − 1)   /* | stands for the bitwise OR */
13:        j ← (t + 1) | (((∼ t & − ∼ t) − 1) >> (__builtin_ctz(j) + 1))
14:    end while
15: end procedure
```

9. The binary representation of 13 is 1101.

B. Traversal through a reduced lattice

Given a solution to the Steiner tree problem on this reduced lattice, we need to design an efficient way to perform the traversal. Indeed, when visiting an equivalence class of the reduced lattice, we want to avoid visiting the same node multiple times. Furthermore, checking if a node has been visited could lead to significant time or memory overhead.

Instead of storing a stack to represent the current state, we will use a vector v of size $n+1$ where each entry $v[i]$ contains the set of variables $(x_j)_j$ whose degree is i in the monomial. For instance, for the 8-photon Fock state $|s\rangle = |2,0,1,2,0,3\rangle$ we have

$$v = [\{x_2, x_5\}, \{x_3\}, \{x_1, x_4\}, \{x_6\}, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset].$$

Each v is associated to one node in the reduced graph with the partition of n

$$n = \sum_{i=0}^{n} |v[i]| \times i.$$

Computing the children of a node in the original lattice now consists in choosing one variable from a set $v[i]$ of v and moving it to the set $v[i+1]$. To respect the traversal given by the Steiner tree, we look at the children of the equivalent node in the reduced graph. This will give us which entries i of v have to be modified. For each such entry i we iterate over the set $v[i]$, move one variable, and we perform our computations just like in the regular case. At this point this is strictly equivalent to adding a variable in our stack.

With this approach, there remains the problem of visiting the same node multiple times within an equivalent class. This can occur when an entry $v[i]$ contains more than one element. In such cases, nodes might be visited as many times as there are distinct ways to "fill" the set $v[i]$. To avoid this, we need to impose an order ensuring that the set $v[i]$ is constructed only once. Specifically, we require that a variable x_j cannot be added to $v[i]$ if there is already a variable x_k with $k > j$ present.

In practice, we use hash tables to represent an entry $v[i]$, with insert and delete doable in $O(1)$. We need to store the maximum element of a set $v[i]$, which may require up to $O(n)$ to be computed when removing an element from $v[i]$. This is the most costly operation that we introduce with this approach.

II.3.5 Benchmark with SLOS: memory trade-off and feedforward simulation

A. Memory complexity

Overall, our method offers a clear computational advantage over permanent-based formula and a clear memory advantage over SLOS. Furthermore, as the number of modes increases for a fixed number of photons, our method becomes more and more competitive in terms of computational time compared to SLOS while not increasing the memory cost.

We illustrate the computational advantage of LO-SLAP with the following experiment: given a computer with 8 Gb of memory and a clock rate at 1 GHz, we estimate the maximum number of modes our computer can strongly simulate in a day (86400 seconds) as a function of the number of photons. The results are given in Fig. II.10 for a comparison against SLOS and in Fig. II.11 for a comparison against SLOS with mask and the permanent-based method, consisting of computing every probability independently.

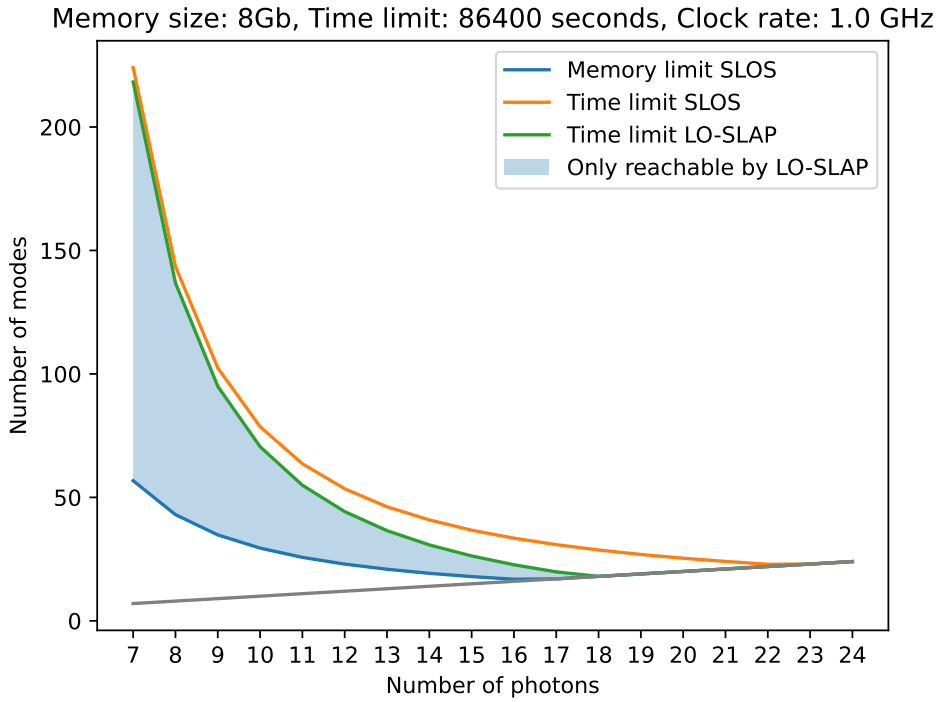


Figure II.10 – For a given number of photons, we give the maximum number of modes we can strongly simulate with a classical computer of 8 Gb of memory and a clock rate of 1 GHz. The memory limit of LO-SLAP is not shown as it is the vertical line at 29 photons.

B. Feedforward

In this section, we first explain how we can perform the simulation of adaptive schemes pictured in Figure II.2. Then we compare the performance of LO-SLAP with that of SLOS (Section II.2.5.A.).

Simulating feedforward with LO-SLAP.

> **Proposition II.39** (Simulation of ℓ-adaptive measurements with LO-SLAP)**.** *The full output amplitude distribution of a linear optical circuit with generic ℓ-adaptive measurements can be computed by performing one run of LO-SLAP, with the only additional cost of updating the matrix U during the process.*

Proof. A nice feature of the depth-first search traversal of the lattice is that it computes the output amplitudes *mode by mode* in order. Furthermore, note that as long as a set of variables $(x_i)_{i\in J\subset[n]}$ has not yet appeared in the nodes during the traversal, we can apply some unitary operations on the associated modes $i \in J$, modifying only the rows $i \in J$ of U. This will alter the output results related to these modes without affecting the previous computations, ensuring that all the previously computed nodes still valid and useful.

In practice, when a new node is computed, we need to check whether the added variable is a measured mode, and if so, we must update U accordingly. This corresponds to line 36 in Algorithm 7. Note that the update of U can be done simultaneously with the update of the coefficients, as the rows of U that are modified do not affect the update. Overall, LO-SLAP can handle feedforward with the

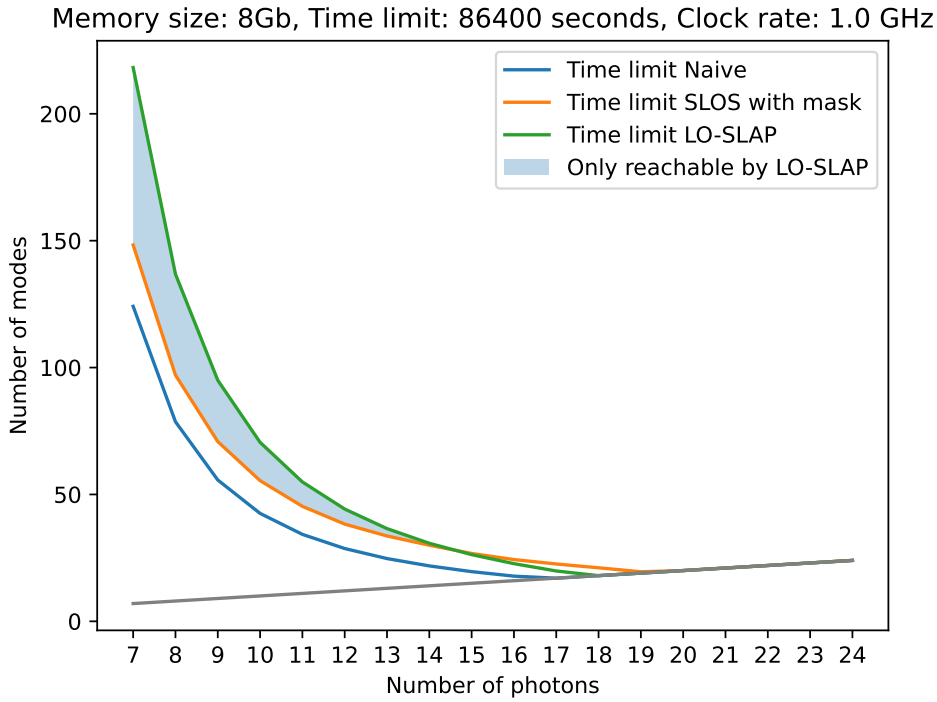


Figure II.11 – Similarly to Fig. II.10, we give the maximum number of modes we can strongly simulate with a classical computer of 8 Gb of memory and a clock rate of 1 GHz. We compare LO-SLAP against SLOS with mask and the permanent-based method. For SLOS, we took the best mask optimizing the time given the memory limit. Therefore, in this case, the fact that LO-SLAP can reach larger problem sizes is because LO-SLAP is faster than the other methods (including SLOS with mask).

only cost of updating U, which may introduce a multiplicative $O(m^2)$ factor in the time complexity in the worst case. □

Comparison with SLOS.

We recall here (cf Proposition II.25) the time complexity of SLOS with a mask on ℓ modes:

$$\sum_{s=1}^{n}\sum_{d=s}^{n} 2s \times \binom{n-d+\ell-\delta_{\ell m}}{n-d} \times \binom{m}{s} \times \binom{d-1}{d-s}$$

where $\delta_{\ell m} = 1$ if $\ell = m$ and 0 otherwise.

Note that for $\ell = 0$, i.e. with no adaptive measurements, we recover the standard version of SLOS which is the fastest method. For $\ell = m$, we recover independent computations with one-output SLOS, which is slower than LO-SLAP. LO-SLAP stands between these two extreme cases and, for a given n and m, there exists a ℓ such that SLOS with mask and LO-SLAP have the same computational complexity. This is illustrated in Figure II.12, where the number of photons has been fixed to an arbitrary value of 15. The plane is divided into two regions where each method is faster. When the number of adaptive measurements is large enough, LO-SLAP prevails.

So far we made no assumption on the memory to produce Fig II.12. However on a modest laptop, even with a small ℓ, if m is large enough SLOS with a mask of size ℓ may not fit in the memory. If this

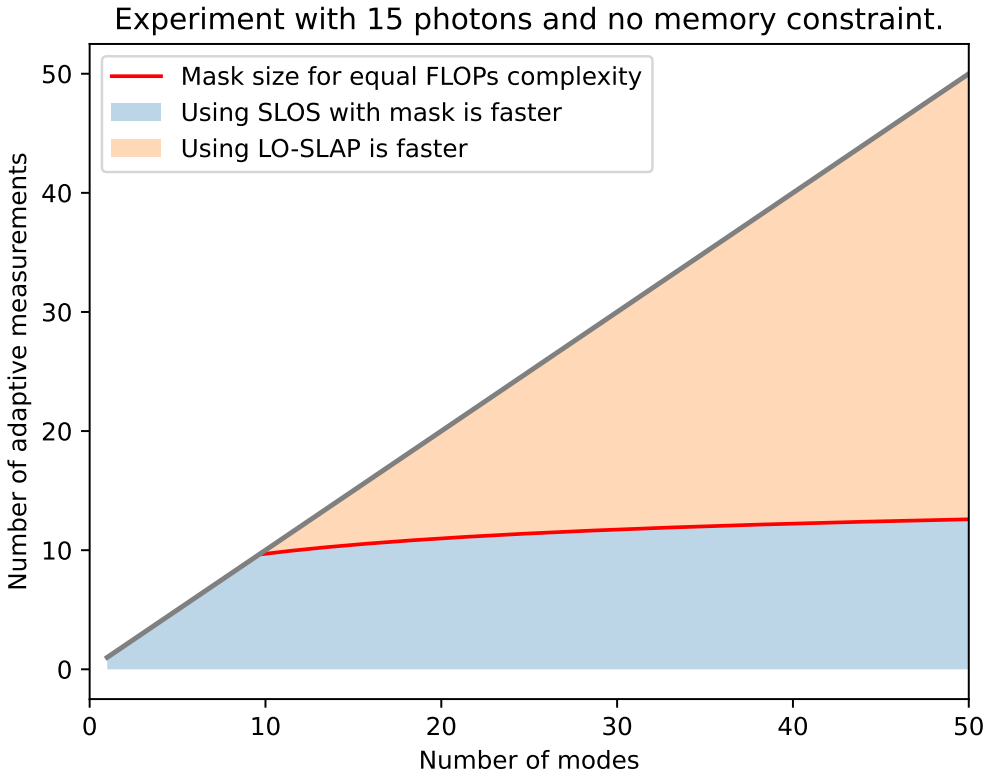


Figure II.12 – Regions indicating which of LO-SLAP or SLOS with mask is faster, based on the theoretical number of FLOPs, as a function of the number of adaptive measurements and the number of modes. The number of photons is fixed at 15.

is the case we would need a larger mask. Namely,

$$\text{mask size} = \max(\ell, \text{minimum mask size for SLOS to fit in memory})$$

and this will have an impact on which method is faster, especially for small ℓ and large m. We adapted Figure II.12 with a memory constraint of 8 Gb. The results are given in Figure II.13. The number of adaptive measurements and the actual mask size used in SLOS are now two different quantities. For a given point in the plane ($m = \#$ modes, $\ell = \#$ adaptive measurements) we now need to check if the memory constraint can be satisfied with a mask size ℓ or if we need a larger mask. If our point (m, ℓ) is below the minimum mask size required by the memory constraint then the actual mask size needs to be adjusted, giving a new point (m, ℓ'). This is illustrated with the points B and C which correspond to unfeasible configurations, and are updated to B' and C'. If the new point is below the curve giving the mask size for equal complexity between SLOS and LO-SLAP then SLOS with mask is faster, otherwise LO-SLAP is faster.

The main point we want to emphasize is that once the minimum mask size for SLOS to fit in the memory is larger than the mask size for SLOS and LO-SLAP to be of same computational complexity, then in any case LO-SLAP will be faster. Because the memory constraint imposes that the mask size will always be larger than the mask size required for SLOS and SLAP to be of similar efficiency, whatever the number of adaptive measurements.

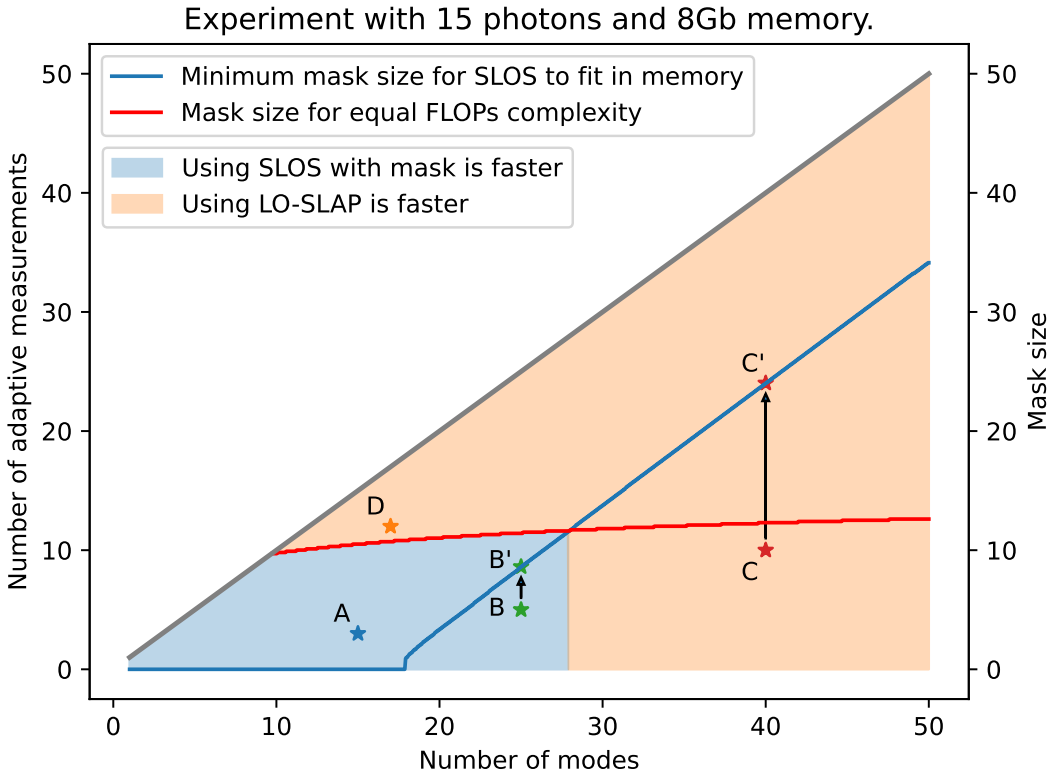


Figure II.13 – Same experiment than Fig II.12 with an additional memory constraint. The red curve gives the mask size for SLOS to have the same complexity as LO-SLAP. The blue curve represents the smallest possible mask for SLOS to run with 8Gb of memory — thus, any point below the blue curve exceeds the memory limit. For instance, points A and D are above the blue curve so setting the mask size to the number of adaptive measurements is possible. Note that for point A, the mask size is smaller than the red curve so SLOS with mask is faster, whereas for point D, LO-SLAP is better. Regarding B and C, both are under the blue curve which means that the configuration is unfeasible if we keep the mask size equal to the number of adaptive measurements. We need to set the mask size to larger values, giving points B' and C'. Note that point B' is still under the red curve so that SLOS with mask remains faster. However, point C' lies above the red curve, hence LO-SLAP is faster.

II.3.6 Extension to noisy simulation

We review the three kinds of noise described in Section I.3.5.D..

A. Uniform photon loss

Photons can be lost anywhere during the computation. Different models for loss exist, here we assume that each photon has the same probability $\eta \in [0, 1]$ of being lost in any mode. With such a uniform loss, we can use commutation rules for loss [OB18] to assume that, without loss of generality, the loss only happens at the beginning of the circuit, before applying the unitary U.

With this settings, the input of our system is no longer the Fock state $a_1^\dagger a_2^\dagger \dots a_n^\dagger |00\dots0\rangle$ but a

probabilistic mixture of all loss scenarios

$$\sum_{i=0}^{n} \sum_{J \in \mathcal{C}_{n-i}^{n}} \eta^{n-i}(1-\eta)^{i} \prod_{j \in [n]\setminus J} a_j^{\dagger} \left|00\ldots0\right\rangle \left\langle 00\ldots0\right| \prod_{j \in [n]\setminus J} a_j$$

where the sum on i is on the number of non-lost photons and the second sum is over all possible input Fock states with i photons among the initial n ones.

Given a state $|s\rangle$ on $k \leq n$ photons, the probability of measuring $|s\rangle$ is given by linearity over all canonical input states:

$$\eta^{n-k}(1-\eta)^{k} \times \sum_{J \in \mathcal{C}_{n-k}^{n}} \frac{\left|\mathsf{Per}\left(U_{s,[n]\setminus J}\right)\right|^2}{s_1!s_2!\ldots s_m!}.$$

Fortunately, all the coefficients $\left\{\mathsf{Per}\left(U_{s,[n]\setminus J}\right) \mid J \in \mathcal{C}_{n-k}^{n}\right\}$ are computed in the node associated to state $|s\rangle$ during the traversal of the LO-SLAP algorithm. In other words, all the coefficients for any loss scenario are naturally computed in one call of the lossless LO-SLAP traversal.

B. Distinguishability

For photons to interfere, they must be indistinguishable. A commonly used simplification is to treat partial distinguishability as a probabilistic mixture of scenarios, each involving subgroups of indistinguishable photons. In such a model, photons outside each subgroup can be treated as lost, and simulations can be run independently on each subgroup. This implies that any such scenario corresponds to an equivalent photon-loss configuration, up to a linear recombination of mixtures. Under this assumption, similar to the photon-loss case, LO-SLAP provides all the data necessary to compute outputs under imperfect indistinguishability.

C. Multi-photon emission

At source, there is a nonzero probability that more than one photon is emitted. Therefore, we should also consider, with some probability, inputs of $n+1, n+2, \ldots,$ photons. Given that the probability that one more photon is emitted is quite small, the chances of having more than 2 photons in each mode are quite unlikely. We can approximate the total possible number of photons to $2n$ and, based on the fact that all other scenarios are covered by our method, a single run of LO-SLAP with the input $x_1^2 x_2^2 \ldots x_n^2 \left|vac\right\rangle$ will deal with all possible smaller number of photons as well, including both uniform photon loss and imperfect indistinguishability.

II.4 Boson sampling for generic input and with feedforward

In this section, we present a classical algorithm to sample from generalized photonic input states and that is compatible with adaptive circuits. We begin with a description of how the sampling algorithm works (Section II.4.1). We then show a use-case with inputs states in superposition with the vacuum state (Section II.4.2). Next, we present a generalized version of Ryser's algorithm to compute a sum of permanents derived from the same matrix (Section II.4.3). We conclude by showing how this new generalized version enables to sample from linear cluster input state (Section II.4.4).

Contributions

This section contains unpublished work. The gate-by-gate section (Section II.4.1) has been a collaborative work with Timothée Goubault de Brugière. The use-case experiments (Definitions II.45,

II.48 and II.53) were defined and introduced by Rawad Mezher. All the remainder of this section is of my own contribution.

II.4.1 Gate-by-gate simulation

A. Description of the algorithm

This algorithm is inspired by the gate-by-gate sampling from [BGL22]. Let us first summarize the gate-by-gate algorithm of [BGL22].

Let $\hat{U}$ be a m-mode circuit we want to sample from. Let us consider that circuit is composed of M gates, i.e. $\hat{U} = \hat{U}_M \dots \hat{U}_1$. Let $|\psi\rangle$ be the input state.

Informal description of the gate-by-gate algorithm. The idea is to show we can sample from the final distribution $\hat{U}|\psi\rangle$ by sequentially sampling from $\hat{U}_i \dots \hat{U}_1 |\psi\rangle$, conditioned on the sample obtained from $\hat{U}_{i-1} \dots \hat{U}_1 |\psi\rangle$. The key advantage is that the conditioned previous sample can greatly restrict the number of outcomes. Indeed, we only need to consider the states that are affected by U_i, which is in practice a few states compared to the whole space. All the proofs and detailed are in [BGL22].

Algorithmic description of the gate-by-gate algorithm. The algorithm works as follows.

- Get a first sample t from $\{|\langle t|\psi\rangle|^2\}$.

Then for each step $i \in [1, M]$:

- We consider all the modes that are not affected by $\hat{U}_i$. Let us note that set A. It is defined as

$$A \coloneqq \{1, 2, \dots, m\} \setminus \mathrm{supp}(\hat{U}_i)$$

  where supp is the support of $\hat{U}_i$.

- We consider all the states s that coincide with the previous sample t on the modes A. We note that set

$$S \coloneqq \{|s_1, \dots, s_m\rangle \ : \ s_A = t_A\}.$$

- We get a new sample t among the distribution:

$$\frac{\left|\langle t_1, \dots, t_m| \hat{U}_i \hat{U}_{i-1} \dots \hat{U}_1 |\psi\rangle\right|^2}{\sum_{s \in S} |\langle s_1, \dots, s_m| U |\psi\rangle|^2}$$

Note that any linear optical circuit $\hat{U}$ on m modes can be decomposed in at most $M = \frac{m(m-1)}{2}$ beam splitters and $\frac{m(m+1)}{2}$ phase shifters (Theorem I.42), leading to a decomposition of $\hat{U}$ into M 2-mode components of matrix U_i. The adaptation of [BGL22] to linear optical circuit is straightforward and is described in Algorithm 9.

Algorithm 9 Boson sampling form generalized input state $|\psi\rangle$

Input: $|\psi\rangle \in \mathcal{F}_m$ and a m-mode linear optical circuit $\hat{U} = \hat{U}_M \cdots \hat{U}_2 \hat{U}_1$.
Output: $t = |t_1, \ldots, t_m\rangle$ with probability $|\langle t_1, \ldots, t_m| U |\psi\rangle|^2$.

$\hat{U} \leftarrow I$
Sample t from the probability distribution $|\langle t_1, \ldots, t_m|\psi\rangle|^2$
for $i = 1$ to M **do**
    $\hat{U} \leftarrow \hat{U}_i \times \hat{U}$
    $A \leftarrow \{1, 2, \ldots, m\} \setminus \mathrm{supp}(\hat{U}_i)$
    $S \leftarrow \{|s_1, \ldots, s_m\rangle : s_A = t_A\}$
    Sample $t \in S$ from $|\langle t_1, \ldots, t_m| \hat{U} |\psi\rangle|^2 / \sum_{s \in S} |\langle s_1, \ldots, s_m| \hat{U} |\psi\rangle|^2$
end for
return t

B. Complexity analysis

Interestingly and as precised in [BGL22], this gate-by-gate simulation is directly compatible for circuits with feedforward (Figure II.2). Indeed, we can first sample all the gates that are involved in the next measurement pattern. Then we can update the following gates accordingly to the result obtained. Thus, the gate-by-gate algorithm answers to both Problem II.4 and II.6. We can therefore give the following time complexity for performing such simulation.

Theorem II.40 (Classical simulation of Boson Sampling with generalized input and feedforward)**.** *Let $|\psi\rangle$ be an n-photon input state, $\hat{C}$ an m-mode adaptive linear optical circuit of depth d, and f_ψ the worst time complexity for computing a single probability $|\langle t| \hat{U} |\psi\rangle|^2$, where $\hat{U}$ is an arbitrary m-mode linear optical circuit. Then sampling from $\langle t| \hat{C} |\psi\rangle$ can be done in*

$$O(n \times d \times f_\psi)$$

complex operations.

Proof. Let us consider a circuit $\hat{C}$ with d layers of gates. Without loss of generality, let us assume we are sampling layer by layer.

Let us consider a beam splitter on the modes k, k'. Given our previous sample $|t_1, \ldots, t_m\rangle$, the size of S, i.e. the set of states of interest, is $t_k + t_{k'} + 1$. Indeed, it is the number of ways to distribute $t_k + t_{k'}$ photons over two modes [10]. As there are in total n photons, the number of pairs of modes that contain at least one photon is n. Given a layer of n beam splitters of interest that act on modes i, j, the number of probabilities to compute is therefore:

$$\sum_{i,j} (t_i + t_j + 1) \leq 2n$$

Each probability takes at most f_ψ operations, and there are d layers, which concludes the proof. □

Remark II.41 (Improvements of f_ψ). Note that we give a worst case complexity for f_ψ, that can be better in practice, as sometimes

10. For instance, if $|t\rangle = |1, 2, 0, 4\rangle$ and we are considering a beam splitter on the first two modes, we are considering the 4 following states, $|3, 0, 0, 4\rangle$, $|2, 1, 0, 4\rangle$, $|1, 2, 0, 4\rangle$, $|0, 3, 0, 4\rangle$.

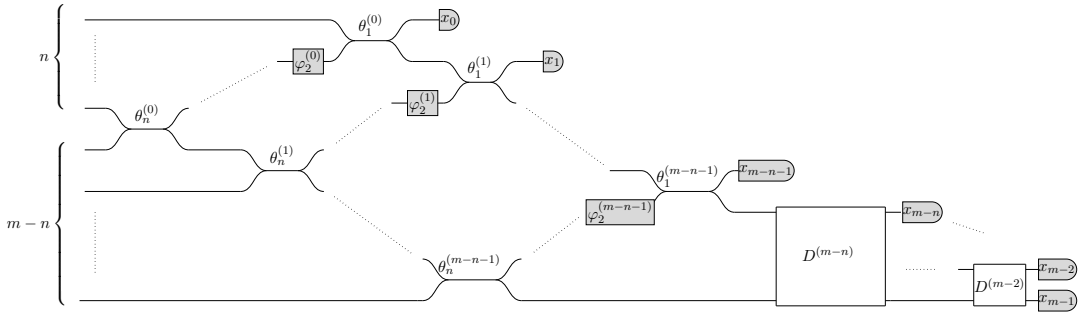


Figure II.14 – Generic form of a linear optical circuit with m adaptive measurements and an input state on the first n modes. Each $D^{(i)}$ represents a diagonal layer composed of beam splitters and phase shifters. The parameters $\theta^{(i)}$ and $\varphi^{(i)}$ on the i-th diagonal are functions of the measurement outcomes $x_1, \ldots, x_{i-1}$.

Proposition II.42 (Depth of a generic circuit with feedforward)**.** *Assuming the input state is prepared on the first n modes, the depth of an m-mode linear optical circuit with m adaptive measurements is*

$$\log(n) \times m.$$

In the following complexities, as we assume $m = O(n)$, we consider the depth of such a circuit to be:

$$O(n \log(n)).$$

Proof. Let us consider a generic linear optical circuit with feedforward, as shown in Figure II.14. The circuit consists of m layers, each corresponding to a diagonal matrix $D^{(i)}$, and each layer spans a depth of n modes. Our goal is to show that the depth of each layer can be reduced from n to $\log(n)$ without loss of generality.

To begin, consider the measurement on the first mode. The relevant information comes from the action of the first diagonal layer $D^{(0)}$ on that mode. In other words, our task reduces to efficiently synthesizing the first row of $D^{(0)}$, which involves n non-zero entries.

Using an approach similar to that in the proof of Theorem 45 from [AA11], we can construct this row using a layered optical circuit of depth $\log(n)$. The idea is as follows:

- A first layer with $n/2$ beam splitters can prepare half of the required coefficients.
- A second layer with $n/4$ beam splitters adds another quarter.
- Continuing this halving process, we require $\log(n)$ layers in total to construct all n coefficients.

Indeed, the number of synthesized coefficients is:

$$\sum_{i=1}^{\log(n)} \frac{n}{2^i} = n.$$

By repeating this $\log(n)$-depth synthesis for each of the m diagonals, we obtain a circuit with total depth $\log(n) \times m$.

Note that the beam splitters in this construction are not restricted to adjacent modes. While such non-local connections may be impractical for physical implementation, they are entirely permissible within the gate-by-gate model, which does not impose connectivity constraints.

□

Remark II.43 (Improved depth). Note that without feedforward, the depth of a m-mode generic circuit with a n-mode input state can be smaller than $\log n \times m$. It can be improved to $n + \log(m)$, cf. Theorem 4 of [Gou+25]. This means that considering a circuit with $\ell < m$ adaptive measurements, we could improve even further the complexity of Proposition II.42.

One straightforward corollary is the sampling from the input state $|1, \dots, 1\rangle$.

Corollary II.44 (Classical simulation of Boson Sampling with feedforward)**.** *Let $|\psi\rangle = |1, \dots, 1, 0, \dots, 0\rangle$ an n-photon input state, and $\hat{C}$ a m-mode adaptive linear optical circuit. Then we can sample from the output distribution $\hat{C}|\psi\rangle$ in*

$$O\left(n^3 \log(n) 2^n\right)$$

complex operations.

Proof. Following Proposition II.42, the depth of $\hat{C}$ is $O(n \log(n))$. The complexity of computing one probability is $n2^n$ in the worst case, as it is the computation of a $n \times n$ matrix permanent. Therefore, we have $n \times (n \log(n)) \times n2^n = n^3 \log(n) 2^n$. □

We now plan to introduce other corollaries for other input states. In Section II.4.2 we explicit the sampling from a single photons superposed with the vacuum state and from Dicke states. In Section II.4.4, we consider the linear cluster state. Note that for each of them, we need to compute the complexity of computing one single output probability.

II.4.2 SBS: Superposition Boson Sampling

In this section, we consider Superposition Boson Sampling (**SBS**), that we define as follows.

Definition II.45 (**SBS**)**. SBS** (Superposition Boson Sampling) consists of sampling from the following input state of n superpositions

$$|\psi_{\textbf{SBS}}\rangle := \bigotimes_{j=1}^{n} (\alpha_j |0\rangle + \beta_j |1\rangle) |0\rangle^{\otimes m-n}, \tag{II.12}$$

where $|0\rangle$, $|1\rangle$ are the vacuum and single photon states respectively, and $|\alpha_j|^2 + |\beta_j|^2 = 1$.

Remark II.46 (Special case: Boson Sampling). Note that when the $\alpha_j = 0$ and $\beta_j = 1$, this reduces to the case of Boson Sampling.

SBS is particularly relevant to consider as the input state is something we can directly produce with quantum dot sources [Lor+19]. Let us first show how to compute output probabilities with $|\psi_{\textbf{SBS}}\rangle$ as input state.

Proposition II.47 (**SBS** single output probability)**.** *Let $|t\rangle$ be a k-photon state and $\hat{U}$ a m-mode linear optical circuit. Then we have:*

$$|\langle t| \hat{U} |\psi_{\textbf{SBS}}\rangle|^2 = \left| \frac{\text{Per}\left(B_{(n-k, t_1, \dots, t_m)}\right)}{(n-k)! \sqrt{t_1! \dots t_m!}} \right|^2$$

where we consider the following $(m+1) \times n$ *matrix*

$$B \coloneqq \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \beta_1 u_{1,1} & \beta_2 u_{1,2} & \dots & \beta_n u_{1,n} \\ \dots & & & \\ \beta_1 u_{m,1} & \beta_2 u_{m,2} & \dots & \beta_n u_{m,n} \end{pmatrix}.$$

with $B_{(n-k,t_1,\dots,t_m)}$ *being formed by taking* $n-k$ *times the first row of* B*, and* t_i *times the* $i+1^{\text{th}}$ *row of* B *for* $i \geq 1$*. Such output probability can be computed in:*

$$O\left(n(n-k)2^{n-k}\right)$$

complex operations.

Proof. Note that after an m-mode interferometer $\hat{U}$, we have the following evolution:

$$|\psi_{\textsf{SBS}}\rangle = \prod_{j=1}^{n} (\alpha_j + \beta_j x_j) \, |0\rangle^{\otimes m} \mapsto P(x_1, \dots, x_m) \, |0\rangle^{\otimes m}$$

where:

$$P(x_1, \dots, x_m) \coloneqq \prod_{j=1}^{n} \left(\alpha_j + \beta_j \sum_{i=1}^{m} u_{i,j} x_i \right)$$

The idea is to relate these coefficients with the Boson Sampling setting, so that we can express these coefficients with matrix permanents. Informally, we do the link by considering a virtual mode that is associated with the vacuum state. Formally, let us define:

$$Q(x_0, x_1, \dots, x_m) \coloneqq \prod_{j=1}^{n} \left(\alpha_j x_0 + \beta_j \sum_{i=1}^{m} u_{i,j} x_i \right)$$

By developing the products, one can show that

$$Q[{x_0}^{n-k} {x_1}^{t_1} \dots {x_m}^{t_m}] = P[{x_1}^{t_1} \dots {x_m}^{t_m}],$$

where $Q[{x_0}^{n-k} {x_1}^{t_1} \dots {x_m}^{t_m}]$ (resp. $P[{x_1}^{t_1} \dots {x_m}^{t_m}]$) is the coefficient of the monomial ${x_0}^{n-k} {x_1}^{t_1} \dots {x_m}^{t_m}$ (resp. ${x_1}^{t_1} \dots {x_m}^{t_m}$) in Q (resp. P). Note that .

Let $\hat{B}$ be the following operator

$$\hat{B} : \hat{a}_j^\dagger \mapsto \alpha_j \, \hat{a}_0^\dagger + \sum_{i=1}^{m} \beta_j u_{i,j} \, \hat{a}_i^\dagger$$

and let consider an output $|t_1, \dots, t_m\rangle$ with:

$$\sum_i t_i = k$$

Then we have:

$$\begin{array}{rl} & \langle t_1, \dots, t_m | \, \hat{U} \, |\psi_{\textsf{SBS}}\rangle \\ = & \sqrt{t_1! \dots t_m!} P[{x_1}^{t_1}, \dots, {x_m}^{t_m}] \\ = & \sqrt{t_1! \dots t_m!} Q[{x_0}^{n-k}, {x_1}^{t_1}, \dots, {x_m}^{t_m}] \\ & \frac{1}{\sqrt{(n-k)!}} \langle n-k, t_1, \dots, t_m | \, \hat{B} \, |0\rangle \, |1\rangle^{\otimes n} \, |0\rangle^{\otimes m-n} \end{array}$$

where we conclude by expressing the last term as the matrix permanent in the usual Boson Sampling setting (Proposition I.50).

As the first row is repeated $n-k$, the complexity for computing such permanent directly follows from Proposition I.59. Note that $\sum_i t_i + (n-k) = n$, so the matrix $M_{n-k,t_1,\dots,t_m}$ is always of size $n \times n$. □

In **SBS**, the number of photons detected at the output can vary the complexity of computing the involved probabilities. In practice, it is important to highlight the two following points.

- Linear optical circuits conserve the number of photons (Proposition I.28). Classically, sampling from $|\psi_{\textbf{SBS}}\rangle$ is equivalent to first sample the number of photons k, and then sample the k-photon element of $|\psi_{\textbf{SBS}}\rangle$.
- The case of the uniform superposition, i.e. $\alpha_j = \beta_j = 1/\sqrt{2}$, is likely to be the most practical to perform experimentally.

Remarkably, the second point makes appear a particular state, i.e. the Dicke states [BE19]:

$$|\psi_{\textbf{SBS}}\rangle = \frac{1}{2^{n/2}} \left(|0\rangle + |1\rangle\right) |0\rangle^{\otimes m-n} = \frac{\sqrt{\binom{n}{n/2}}}{2^{n/2}} \sum_{k=0}^{n} D(m,n,k)$$

where the normalized Dicke state is defined as follows:

$$D(m,n,k) \coloneqq \frac{1}{\sqrt{\binom{n}{k}}} \sum_{\substack{s_1+\cdots+s_n=k \\ s_i\in\{0,1\}}} |s_1,\ldots,s_n\rangle\, |0\rangle^{\otimes m-n}$$

Note that $|\langle D(m,n,n/2)|\psi_{\textbf{SBS}}\rangle|^2$ is not negligible when $k = n/2$. This particular value[11] makes possible to efficiently sample from $D(m,n,n/2)$ by:

- Sampling from **SBS**
- Postselect on detecting $n/2$ photons at the output, i.e. discard all outcomes when the detected number of photons is not $n/2$ and resample again.

For this reason, we define **DBS** and we prove that it is indeed possible to perform **DBS** with **SBS**.

Definition II.48 (DBS). DBS (Dicke-state boson sampling) is a Boson Sampling task with the following Dicke-state input:

$$|D(m,n)\rangle = \frac{1}{\sqrt{\binom{n}{n/2}}} \sum_{\substack{s_1+\cdots+s_n=n/2 \\ s_i\in\{0,1\}}} |s_1,\ldots,s_n\rangle\, |0\rangle^{\otimes m-n},$$

Proposition II.49 (Sampling from **DBS** with **SBS**). *On average, one sample of* **DBS** *can be obtained after* $O(\sqrt{n})$ *runs of* **SBS**.

Proof. From a uniform **SBS** input state, we have:

$$|\psi_{\textbf{SBS}}\rangle = \frac{1}{2^{n/2}} \left(|0\rangle + |1\rangle\right) |0\rangle^{\otimes m-n} = \frac{\sqrt{\binom{n}{n/2}}}{2^{n/2}} \sum_{k=0}^{n} D(m,n,k)$$

The event of detecting $n/2$ photons appears with the probability:

$$|\langle D(m,n,n/2)|\psi_{\textbf{SBS}}\rangle|^2 = \frac{\binom{n}{n/2}}{2^n} \approx \sqrt{2/\pi n}$$

where we use Stirling's approximation. □

This leads to the following corollary.

11. For simplicity, we consider even n, so that $\lfloor n/2 \rfloor = n/2$.

Corollary II.50 (Classical Dicke-state Boson Sampling with feedforward)**.** *Let $|\psi\rangle = D(m, n, n/2)$ the $n/2$-photon Dicke state, and $\hat{C}$ an m-mode adaptive linear optical circuit. Then we can sample from the output distribution of $\hat{C}|\psi\rangle$ in*

$$O\left(n^4 \log(n) 2^{\frac{n}{2}}\right)$$

complex operations.

Proof. From Proposition II.42, the depth is $n\log(n)$. By Proposition II.47 the permanent involved in **DBS** is a $n \times n$ matrix with the first row repeated $n/2$ times. The complexity of computing such permanent is $n^2 2^{\frac{n}{2}}$, concluding the proof. □

II.4.3 Generalized Ryser's formula

A. Context and motivation

As described in Section I.3.5.B., we can compute the single output amplitude of a linear optical circuit $\hat{U}$ with a matrix permanent. We remind the formula as follows.

$$\langle t_1, \dots, t_m | \hat{U} | s_1, \dots, s_m \rangle = \frac{\text{Per}\left(U_{s,t}\right)}{\sqrt{s_1! \dots s_m! t_1! \dots t_m!}} \tag{II.13}$$

where $U_{s,t}$ is obtained from U by repeating s_j times its j^{th} column and t_i times its i^{th} row.

In practice, there exist different formulae for efficiently computing matrix permanents (see Section I.3.6). One of our interest is due to Ryser [Rys63]:

$$\text{Per}\left(A\right) = (-1)^n \sum_{S \subset \{1,\dots,n\}^n} (-1)^{|S|} \prod_{i=1}^{n} \sum_{j \in S}^{n} a_{ij}$$

where for each row in the product, we sum only a subset S of the columns. To illustrate the motivation of this section, let us consider the following input state:

$$|\psi\rangle = \frac{1}{\sqrt{2}} \left(|s_1, \dots, s_m\rangle + |s'_1, \dots, s'_m\rangle \right)$$

For simplicity and as it will be the case for Section II.4.4, we consider $s_i, s'_i \leq 1$. From Equation (II.13), we know we can compute the output amplitude with two permanents:

$$\langle t_1, \dots, t_m | \hat{U} | \psi \rangle = \frac{1}{\sqrt{2 t_1! \dots t_m!}} \left(\text{Per}\left(U_{s,t}\right) + \text{Per}\left(U_{s',t}\right) \right)$$

For computing such amplitude, we could compute the two permanents independently, leading to $2 \times n 2^n$ complex operations. However, if $\boldsymbol{s}$ and $\boldsymbol{s'}$ have similar components, we can see with Ryser's formula that some terms are computed twice.

Example II.51. Let us consider the two input states

$$|1, 0, s_1, \dots, s_m\rangle \qquad |0, 1, s_1, \dots, s_m\rangle$$

where $\sum_i s_i = n - 1$. Note they only differ in the first two modes, and coincide on all the other modes that contain $n-1$ photons. If we plan to independently compute $\text{Per}\left(U_{s,t}\right)$ and $\text{Per}\left(U_{s',t}\right)$ with

Ryser's formula, then the terms involved with $n-1$ common columns are computed twice. Note that it concerns 2^{n-1} terms. So instead of computing the permanent independently, leading [12] to

$$2 \times n2^n$$

operations, we could compute these 2^{n-1} operations once, leading to

$$n2^{n-1} + 2 \times n(2^{n-1}) = \frac{3}{2}2^n$$

Note the ratio goes from 2 to $\frac{3}{2}$. This is not changing the value of the exponent, but this is because there are only two input states. When there is an exponential number of input states, this can lead to improve the exponent, as described in Theorem II.62.

B. Explicit formula

The goal of the following is therefore to formalize and generalize the optimization described in Example II.51. We propose here a generalized version of Ryser's formula, that computes a sum of permanents of submatrices formed from a single matrix.

Definition II.52 (Generalization of Ryser's formula)**.** Let us consider an output state $|t_1, \ldots, t_m\rangle$ and an input state $|\psi\rangle$ as follows:

$$|\psi\rangle = \sum_{\boldsymbol{s} \in \mathcal{I}} \alpha_{\boldsymbol{s}} |s_1, \ldots, s_m\rangle$$

where $\mathcal{I}$ is a finite set states, and we assume $0 \leq s_j \leq 1$ and $0 \leq t_i \leq 1$. Let us define:

$$S \coloneqq \{j \mid s_j \neq 0\} \qquad T \coloneqq \{i \mid t_i \neq 0\}.$$

Then the following sum of permanents

$$\langle t| \hat{U} |\psi\rangle = \sum_{s \in \mathcal{I}} \alpha_s \mathsf{Per}\left(U_{s,t}\right).$$

is equal to

$$(-1)^n \sum_{\mathcal{R} \subseteq \bigcup\limits_{s \in \mathcal{I}} \mathcal{P}(S)} (-1)^{|\mathcal{R}|} \left(\sum_{s \in \mathcal{I} | \mathcal{R} \subseteq S} \alpha_s \right) \prod_{i \in T} \sum_{j \in \mathcal{R}} u_{ij}$$

where $\mathcal{P}(S)$ is the set of all the subsets of S.

Proof. The proof follows by a direct factorization of the involved terms.

$$\begin{aligned}
\langle t| \hat{U} |\psi\rangle &= \sum_{s \in \mathcal{I}} \alpha_s \mathsf{Per}\left(U_{s,t}\right) \\
&= \sum_{s \in \mathcal{I}} \alpha_s \left[(-1)^n \sum_{\mathcal{R} \subseteq S} (-1)^{|\mathcal{R}|} \prod_{i \in T} \sum_{j \in \mathcal{R}} u_{ij} \right] \\
&= (-1)^n \sum_{\mathcal{R} \subseteq \bigcup\limits_{s \in \mathcal{I}} \mathcal{P}(S)} (-1)^{|\mathcal{R}|} \left(\sum_{s \in \mathcal{I} | \mathcal{R} \subseteq S} \alpha_s \right) \prod_{i \in T} \sum_{j \in \mathcal{R}} u_{ij}
\end{aligned}$$

□

12. Note the number of operations in the permanent is $n2^{n-1}$ and not $n2^n$, but we keep the n for illustration purposes.

In practice, the implementation of the generalized Ryser's formula possesses two major technicalities. Indeed, we need to be able to:

- Iterate through all the subsets $\mathcal{R} \subseteq \bigcup_{s\in\mathcal{I}} \mathcal{P}(S)$, and
- Efficiently sum over all coefficients $\sum_{s\in\mathcal{I}|\mathcal{R}\subseteq S} \alpha_s$.

In the following section, we demonstrate a concrete case where such optimization is possible.

II.4.4 EBS: Entangled Boson Sampling

A. Qubit-based linear cluster state: definition and properties

To introduce the linear cluster state, we use qubits. These will be noted $|0\rangle_{\mathsf{L}}$ and $|1\rangle_{\mathsf{L}}$, where we use a subscript to distinguish them from Fock states. We remind the Hadamard basis states:

$$|+\rangle_{\mathsf{L}} \coloneqq \frac{|0\rangle_{\mathsf{L}} + |1\rangle_{\mathsf{L}}}{2} \qquad |-\rangle_{\mathsf{L}} \coloneqq \frac{|0\rangle_{\mathsf{L}} - |1\rangle_{\mathsf{L}}}{2}$$

This section focuses on the linear cluster state, defined as follows.

Definition II.53 (Linear cluster state)**.** Given $N > 1$ qubits in the $|+\rangle_{\mathsf{L}}$ state arranged in a line, we perform a CZ gate between the i^{th} and the $i+1^{th}$ qubit. This state will be called $|\psi_N\rangle_{\mathsf{L}}$ and is mathematically defined as follows:

$$|\psi_N\rangle_{\mathsf{L}} = \prod_{i=1}^{N-1} CZ_{i,i+1} \, |+\rangle_{\mathsf{L}}^N$$

Equivalently, we can write the linear cluster state as:

$$|\psi_N\rangle_{\mathsf{L}} = (|0\rangle + |1\rangle \, \sigma_z)^{N-1} \, |+\rangle_{\mathsf{L}}$$

where $\sigma_z \, |b\rangle_{\mathsf{L}} = (-1)^b \, |b\rangle_{\mathsf{L}}$ for $b \in \{0, 1\}$.

Naively, the sum involved in Definition II.53 has 2^N terms. However, we can drastically reduce the number of terms by allowing different bases for the qubits, e.g. some qubits can be in the $\{|+\rangle_{\mathsf{L}}, |-\rangle_{\mathsf{L}}\}$ basis instead of the $\{|0\rangle_{\mathsf{L}}, |1\rangle_{\mathsf{L}}\}$ basis. Note this change of base can be simply done by local operations on the corresponding qubits. This compact decomposition is the Schmidt decomposition that we describe below.

Proposition II.54 (Schmidt decomposition)**.** *We can efficiently write $|\psi_N\rangle_{\mathsf{L}}$ as a sum of $\lfloor (N+1)/2 \rfloor$ states [BR01], where for $N \geq 1$:*

$$\begin{aligned} |\psi_{2N}\rangle_{\mathsf{L}} &= \tfrac{1}{\sqrt{2}}(|0\rangle_{\mathsf{L}} + |1\rangle_{\mathsf{L}} \, \sigma_z) \, |\psi_{2N-1}\rangle_{\mathsf{L}} \\ |\psi_{2N+1}\rangle_{\mathsf{L}} &= \tfrac{1}{\sqrt{2}}(|+\rangle_{\mathsf{L}} \, |0\rangle_{\mathsf{L}} + |-\rangle_{\mathsf{L}} \, |1\rangle_{\mathsf{L}} \, \sigma_z) \, |\psi_{2N-1}\rangle_{\mathsf{L}} \end{aligned}$$

where $|\psi_1\rangle = |+\rangle$.

Proof. The first inductive formula directly follows from the inductive formula of Definition II.53. For the second one, we have:

$$\begin{aligned} \sqrt{2} \, |\psi_{2N+1}\rangle_{\mathsf{L}} &= \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle \, \sigma_z)(|0\rangle_{\mathsf{L}} + |1\rangle_{\mathsf{L}} \, \sigma_z) \, |\psi_{2N-1}\rangle_{\mathsf{L}} \\ &= \tfrac{1}{\sqrt{2}} \left[|0\rangle_{\mathsf{L}} \, |0\rangle_{\mathsf{L}} + |0\rangle_{\mathsf{L}} \, |1\rangle_{\mathsf{L}} \, \sigma_z + |1\rangle_{\mathsf{L}} \, |0\rangle_{\mathsf{L}} - |1\rangle_{\mathsf{L}} \, |1\rangle_{\mathsf{L}} \, \sigma_z \right] |\psi_{2N-1}\rangle_{\mathsf{L}} \\ &= \tfrac{1}{\sqrt{2}} \left[(|0\rangle_{\mathsf{L}} + |1\rangle_{\mathsf{L}}) \, |0\rangle_{\mathsf{L}} + (|0\rangle_{\mathsf{L}} - |1\rangle_{\mathsf{L}}) \, |1\rangle_{\mathsf{L}} \, \sigma_z \right] |\psi_{2N-1}\rangle_{\mathsf{L}} \\ &= (|+\rangle_{\mathsf{L}} \, |0\rangle_{\mathsf{L}} + |-\rangle_{\mathsf{L}} \, |1\rangle_{\mathsf{L}} \, \sigma_z) \, |\psi_{2N-1}\rangle_{\mathsf{L}} \end{aligned}$$

□

Example II.55 (Expressions for some values of N)**.** For giving an idea of the Schmidt decomposition, we put below the decomposition for $N = 1$ to 7, up to the normalization factor of $2^{-\frac{1}{2}\lfloor (N+1)/2 \rfloor}$

$$
\begin{aligned}
N = 1 \quad & |+\rangle_{\mathsf{L}} \\
N = 2 \quad & |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} \\
& + |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} \\
N = 3 \quad & |+\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} \\
& + |-\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} \\
N = 4 \quad & |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} \\
& + |0\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} \\
& + |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} \\
& + |1\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} \\
N = 5 \quad & |+\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} \\
& + |+\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} \\
& + |-\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} \\
& + |-\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}}
\end{aligned}
$$

$$
\begin{aligned}
N = 6 \quad & |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} \\
& + |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} \\
& + |0\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} \\
& + |0\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} \\
& + |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} \\
& + |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} \\
& + |1\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} \\
& + |1\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} \\
N = 7 \quad & |+\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} \\
& + |+\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} \\
& + |+\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} \\
& + |+\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} \\
& + |-\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} \\
& + |-\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} \\
& + |-\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}} |0\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} \\
& + |-\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |+\rangle_{\mathsf{L}} |1\rangle_{\mathsf{L}} |-\rangle_{\mathsf{L}}
\end{aligned}
$$

Conventions. For simplicity and without loss of generalitiy, for all the following proofs and formulae of the Schmidt decomposition, we consider only even N from now on.

From Example II.55, we can notice some patterns in the elements of $|\psi_N\rangle_{\mathsf{L}}$.

Proposition II.56. *For any even N, let us note the Schmidt decomposition of $|\psi_N\rangle_{\mathsf{L}}$ as follows:*

$$\frac{1}{\sqrt{2^{N/2}}} \sum_{i=1}^{2^{N/2}} |b_1^{(i)}\rangle_{\mathsf{L}} |\mu_1^{(i)}\rangle_{\mathsf{L}} \dots |b_{n/2}^{(i)}\rangle_{\mathsf{L}} |\mu_{n/2}^{(i)}\rangle_{\mathsf{L}}$$

where $b_j^{(i)} \in \{0,1\}$ and $\mu_j^{(i)} \in \{+,-\}$. Furthermore, we have the following properties.

- *All the $(N/2)$-bit strings appear, and they appear exactly once, i.e.*

$$\left\{ b_1^{(i)}, \dots, b_{N/2}^{(i)} | 1 \le i \le 2^{N/2} \right\} = \{0,1\}^{N/2}$$

- *The $\mu_j^{(i)}$ are uniquely determined as follows:*

$$\left| \mu_j^{(i)} \right\rangle_{\mathsf{L}} = \begin{cases} |+\rangle_{\mathsf{L}} \text{ when } b_j^{(i)} \oplus b_{j+1}^{(i)} = 0 \\ |-\rangle_{\mathsf{L}} \text{ when } b_j^{(i)} \oplus b_{j+1}^{(i)} = 1 \end{cases}$$

  where we choose the convention $b_{N/2+1}^{(i)} = 0$.

Without loss of generality, we can apply Hadamard gates on the $\mu_j^{(i)}$ qubits, so that we can

consider the following state instead:

$$\frac{1}{\sqrt{2^{N/2}}}\sum_{i=1}^{2^{N/2}} |b_1^{(i)}\rangle_{\mathsf{L}}|\nu_1^{(i)}\rangle_{\mathsf{L}}\dots|b_{n/2}^{(i)}\rangle_{\mathsf{L}}|\nu_{n/2}^{(i)}\rangle_{\mathsf{L}}$$

where $\nu_j^{(i)} \in \{1,0\}$. *As* $\nu_j^{(i)} = H\mu_j^{(i)}$, *they satisfy:*

$$\nu_j^{(i)} = b_j^{(i)} \oplus b_{j+1}^{(i)}$$

Proof. The properties directly follow from the recurrence formulae of Proposition II.54. For the even case, we can indeed note both values $|0\rangle_{\mathsf{L}}$ and $|1\rangle_{\mathsf{L}}$ are present on the even positions:

$$|\psi_{2N}\rangle_{\mathsf{L}} = \frac{1}{\sqrt{2}}(|0\rangle_{\mathsf{L}} + |1\rangle_{\mathsf{L}}\,\sigma_z)\,|\psi_{2N-1}\rangle_{\mathsf{L}}$$

by noting that σ_z cannot change the values of the bit strings.

For the odd case, we can show this property by strong induction, where the inductive step is for $N \geq 2$:

$$\begin{aligned}
&|\psi_{2N+1}\rangle_{\mathsf{L}}\\
=&\tfrac{1}{\sqrt{2}}(|+\rangle_{\mathsf{L}}\,|0\rangle_{\mathsf{L}} + |-\rangle_{\mathsf{L}}\,|1\rangle_{\mathsf{L}}\,\sigma_z)\,|\psi_{2N-1}\rangle_{\mathsf{L}}\\
=&\tfrac{1}{2}(|+\rangle_{\mathsf{L}}\,|0\rangle_{\mathsf{L}} + |-\rangle_{\mathsf{L}}\,|1\rangle_{\mathsf{L}}\,\sigma_z)(|+\rangle_{\mathsf{L}}\,|0\rangle_{\mathsf{L}} + |-\rangle_{\mathsf{L}}\,|1\rangle_{\mathsf{L}}\,\sigma_z)\,|\psi_{2N-3}\rangle_{\mathsf{L}}\\
=&\tfrac{1}{2}(|+\rangle_{\mathsf{L}}\,|0\rangle_{\mathsf{L}}\,|+\rangle_{\mathsf{L}}\,|0\rangle_{\mathsf{L}} + |+\rangle_{\mathsf{L}}\,|0\rangle_{\mathsf{L}}\,|-\rangle_{\mathsf{L}}\,|1\rangle_{\mathsf{L}}\sigma_z + |-\rangle_{\mathsf{L}}\,|1\rangle_{\mathsf{L}}\,|-\rangle_{\mathsf{L}}\,|0\rangle_{\mathsf{L}} + |-\rangle_{\mathsf{L}}\,|1\rangle_{\mathsf{L}}\,|+\rangle_{\mathsf{L}}\,|1\rangle_{\mathsf{L}}\sigma_z)\,|\psi_{2N-3}\rangle_{\mathsf{L}}
\end{aligned}$$

where we put in red the triples of interest. □

Remark II.57 (Enumerate the state with a Gray code). Note that the properties of Proposition II.56 allow to efficiently enumerate all the states. Indeed, we can enumerate the $b_1^{(i)},\dots,b_{N/2}^{(i)}$ and then uniquely assign the values of the $\mu_j^{(i)}$. Note that we can enumerate the bit strings $b_1^{(i)},\dots,b_{N/2}^{(i)}$ in the order of a Gray code, where only one value of b_j is changing at each step, thus also efficiently enumerating the μ_j, as it is at most changing two values.

B. Boson Sampling with dual-rail linear cluster state

To encode the qubit linear cluster state introduced in Section II.4.4.A., we introduce the dual-rail encoding.

Definition II.58 (Dual-rail linear cluster state)**.** We encode each qubit logical qubit $|b\rangle_{\mathsf{L}}$ with two spatial modes as follows:

$$|0\rangle_{\mathsf{L}} \equiv |1,0\rangle \qquad |1\rangle_{\mathsf{L}} \equiv |0,1\rangle$$

Let consider an even number of photons n. We define the dual-rail linear cluster state $|\psi_n\rangle$ as follows:

$$\frac{1}{\sqrt{2^{N/2}}}\sum_{i=1}^{2^{N/2}} |b_1^{(i)}\rangle|\nu_1^{(i)}\rangle\dots|b_{n/2}^{(i)}\rangle|\nu_{n/2}^{(i)}\rangle$$

where $|b_j^{(i)}\rangle \in \{|1,0\rangle, |0,1\rangle\}$ and $|b_j^{(i)}\rangle \in \{|1,0\rangle, |0,1\rangle\}$. Note that following from Proposition II.56, the strings $b_1^{(i)}\dots b_{n/2}^{(i)}$ take all the possible values, and

$$|\nu_j^{(i)}\rangle = \begin{cases} |1,0\rangle \text{ if } \mathbf{L}(|b_j^{(i)}\rangle) \oplus \mathbf{L}(|b_{j+1}^{(i)}\rangle) = 0 \\ |0,1\rangle \text{ if } \mathbf{L}(|b_j^{(i)}\rangle) \oplus \mathbf{L}(|b_{j+1}^{(i)}\rangle) = 1 \end{cases}$$

where $\mathbf{L}(|1,0\rangle) = 0$ and $\mathbf{L}(|0,1\rangle) = 1$.

Remark II.59. Note the n-photon dual-rail linear cluster state is encoded on $2n$ spatial modes.

We can now define the optical experiment we are interested in.

Definition II.60 (EBS). EBS (Entangled Boson Sampling) is the Boson Sampling task on an m-mode interferometer with the n-photon dual-rail linear cluster state as input on the first $2n$ modes.

Before stating and proving the main theorem on how to efficiently compute a single output amplitude of **EBS**, let us prove the following lemma.

Lemma II.61. *Let $\mathcal{I}$ the set of all terms involved in the dual-rail linear cluster state:*

$$\mathcal{I}_{n/2} = \left\{ |b_1^{(i)}\rangle|\nu_1^{(i)}\rangle \dots |b_{n/2}^{(i)}\rangle|\nu_{n/2}^{(i)}\rangle \middle| 1 \leq i \leq 2^{n/2} \right\}$$

where each state $s \in \mathcal{I}$ satisfied the properties of Definition II.58.

Let F be the following number of subsets:

$$F \coloneqq \left| \bigcup_{s \in \mathcal{I}_{n/2}} \mathcal{P}(\{j \mid s_j \neq 0\}) \right|$$

Then:

$$F = O\left(7.773^{n/2}\right)$$

Proof. To compute F, we exhibit a recurrence relation. For $1 \leq \ell \leq n/2$, let us define

$$\mathcal{I}_\ell \coloneqq \left\{ |b_{\frac{n}{2}-\ell+1}^{(i)}\rangle|\nu_{\frac{n}{2}-\ell+1}^{(i)}\rangle \dots |b_{n/2}^{(i)}\rangle|\nu_{n/2}^{(i)}\rangle \middle| 1 \leq i \leq 2^{n/2} \right\}$$

and

$$F_\ell = \left| \bigcup_{s \in \mathcal{I}_\ell} \mathcal{P}(\{j \mid s_j \neq 0\}) \right|$$

Note we want to compute $F = F_{n/2}$.

For the following, recall that each $\left|b_k^{(i)}\right\rangle$ corresponds to a 2-mode state, which can be either $|1,0\rangle$ or $|0,1\rangle$. As a result, the subsets associated with $\left|b_k^{(i)}\right\rangle$ in F can take three possible forms: the empty set, the index corresponding to $|1,0\rangle$, or the index corresponding to $|0,1\rangle$. For simplicity—and to avoid introducing multiple indices—we label these subsets as $(0,0)$, $(1,0)$, and $(0,1)$.

To compute F, let us consider the following disjoint cases:

$$\begin{array}{rcl} F_\ell(0) & = & F_\ell \text{ when the subset for } b_{n/2-\ell+1} \text{ is } (0,0) \\ F_\ell(\overline{0}) & = & F_\ell \text{ when the subset for } b_{n/2-\ell+1} \text{ is either } (1,0) \text{ or } (0,1) \end{array}$$

Note these two cases cover all possibilities, thus:

$$F_\ell = F_\ell(0) + F_\ell(\overline{0}) \tag{II.14}$$

Then, we consider the three disjoint cases.

- If the subset chosen for the first $b_{n/2-\ell+1}$ is $(0,0)$, then there is no constrain for the subsets of $\nu_{n/2-\ell+1}$. There are therefore 3 choices, and then we are free to decide for $b_{n/2-\ell+2}$. Therefore:

$$F_\ell(0) = 3F_{\ell-1}$$

- If the subset chosen for the first $b_{n/2-\ell+1}$ is $(1,0)$ or $(0,1)$, but the subset chosen for $b_{n/2-\ell+2}$ is $(0,0)$, then there is no constrain for $\nu_{n/2-\ell+1}$ and can take three values: $(0,0)$, $(1,0)$ or $(0,1)$. Note there are three choice for each of the two $b_{n/2-\ell+1}$, so 6 in total.
- If the subset chosen for the first $b_{n/2-\ell+1}$ is $(1,0)$ or $(0,1)$, and the subset chosen for $b_{n/2-\ell+2}$ is $(1,0)$ or $(0,1)$, then there are constrains for $\nu_{n/2-\ell+1}$, and for each possibility there are two choices:$(0,0)$ or the value determined by Proposition II.56. Note there are two choice for each of the two $b_{n/2-\ell+1}$, so 4 in total.

The previous two last cases lead to:

$$F_\ell(\overline{0}) = 6F_{\ell-1}(0) + 4F_{\ell-1}(\overline{0}) = 4\left(F_{\ell-1}(0) + F_{\ell-1}(\overline{0})\right) + 2F_{\ell-1}(0) = 4F_{\ell-1} + 2F_{\ell-1}(0)$$

By noting that $F_{\ell-1}(0) = 3F_{\ell-2}$ (cf. first case) and following Equation (II.14), we have the following recurrence relation:

$$F_\ell = 7F_{\ell-1} + 6F_{\ell-2}$$

This is a linear homogeneous recurrence relation with constant coefficients of order 2.

$$r^2 - 7r - 6 = 0 \to r = \frac{7 \pm \sqrt{73}}{2}$$

. As $F_0 = 0$ and $F_1 = 5$, we have:

$$F = \frac{5}{7}\left[\left(\frac{7+\sqrt{73}}{2}\right)^{n/2} + A\left(\frac{7-\sqrt{73}}{2}\right)^{n/2}\right]$$

For simplification, we consider only the largest value in absolute value, which is around:

$$7.772 \leq \frac{7+\sqrt{73}}{2} \leq 7.773$$

□

Theorem II.62 (**EBS** single output ampltitude)**.** *Let $|t\rangle$ be a n-photon state and $\hat{U}$ a m-mode linear optical circuit, with $m \geq 2n$. Then the single output amplitude of* **EBS**

$$\langle t| \hat{U} |\psi_n\rangle |0\rangle^{\otimes m-2n}$$

can be computed in

$$O\left(n^2 7.773^{\frac{n}{2}}\right)$$

complex operations, and in

$$O\left(n^2 7.773^{\frac{n}{2}}\right)$$

when the subsets are enumerated in a Gray code order.

Proof. First, let us note that the state shown in Proposition II.56 is a sum of $2^{n/2}$ Fock basis states. We can therefore apply Ryser's formula for each of those terms, leading to a time complexity of:

$$2^{n/2} \times n2^n = O\left(n8^{\frac{n}{2}}\right)$$

To aim for a more efficient computation, we can use the generalized Ryser's formula of Definition II.52. We therefore need to iterate through all subsets of columns that are involved in all the terms. We proceed to enumerate all the subsets as follows:

- We first iterate on all the subsets involed in the $|b\rangle_1^{(i)} \dots |b_{n/2}^{(i)}\rangle$. Note for each pair of modes $|b_j^{(i)}\rangle$, we have three possible subsets: either chose the left colunm ($|1,0\rangle$), the right one ($|0,1\rangle$), or neither of the two ($|0,0\rangle$). This can be done in a Gray code order.
- Once the $b_j^{(i)}$ are fixed, we enumerate through all subsets of the $\nu_j^{(i)}$. Note there are two cases.
  - When the surrounding of $\nu_j^{(i)}$ is fixed with either $(1,0)$ or $(0,1)$. Therefore, following Proposition II.56, $\nu_j^{(i)}$ is fixed. Then the subset of $\nu_j^{(i)}$ can only take two values, i.e. the fixed value or $(0,0)$.
  - When one of the surrounding of $\nu_j^{(i)}$ is not fixed, then the subsets of $\nu_j^{(i)}$ can take all the three values, i.e. the left column $(1,0)$, the right one $(0,1)$, or neither of the two $(0,0)$.

It remains to note that:

- We can indeed compute the sum $A = \sum \alpha_s$ efficiently. In our case, is the same as counting the number of states that are in adequacy with the chosen subset. Note that number of states is always of power of 2. For instance, when all subsets are $(0,0)$, we have $A = 2^{n/2}$, and when all the nonzero subsets of b_j are fixed we have $A = 1$. While enumerating, we can have the following changes:
  - When we change the subset of b_j or ν_j that has no constrains (i.e. the value is not fixed because it is not surrounded by fixed b_j)
    - from $(1,0)$ or $(0,1)$ to $(0,0)$: $A \mapsto A \times 2$.
    - from $(0,0)$ to $(1,0)$ or $(0,1)$ $(1,0)$: $A \mapsto A/2$.
  - When we change the subset of ν_j that is surrounded by fixed values, then we do not change A.

  By storing all the values of 2^k for $k = 1$ to $n/2$ and knowing the small change of ν in the enumeration, this value can be computed in $O(1)$ complex operations.
- The number of subsets to iterate is $O\left(7.773^{\frac{n}{2}}\right)$ (Lemma II.61).
- The sum $\prod_{i \in T} \sum_{j \in \mathcal{R}} u_{ij}$ takes $O\left(n^2\right)$ complex operations.

By assuming we can indeed iterate the ν_j in a Gray code order and similarly to Ryser's algorithm, the complexity of updating the product of the sum is only $O(n)$ instead of n^2, concluding the proof. □

We can conclude on the time complexity for sampling from **EBS**.

Corollary II.63 (Classical Entangled Boson Sampling with feedforward)**.** *Let $|\psi\rangle$ the n-photon dual rail linear cluster state, and $\hat{C}$ an m-mode adaptive linear optical circuit. Then we can sample from the output distribution of $\hat{C}|\psi\rangle$ in*

$$O\left(n^3 \log(n) 7.773^{\frac{n}{2}}\right)$$

complex operations.

Chapter III
Graphical language for linear optical circuits

To characterize physical systems and assess their behavior for long-term architectures, it is crucial to develop formal frameworks that address fundamental and high-level properties. Such frameworks are even more valuable when they are both powerful and easy to use.

Among the formal frameworks to study systems, one that has recently been particularly prominent for representing quantum processes is graphical languages (Section I.4). Based on category theory, this formalism aims to represent the state of the systems with wires, connected to gadgets representing the operations. Those diagrams can be visually and still formally manipulated, giving a new way to understand and reason on quantum operations.

This high-level representation has led to many new results in the field. The prime example is ZX-diagrams [CD08; CD11; HNW18; JPV18] with applications in compilation [KW20; Bac+21; Wet+24], simulation [KWV22; KW22; KYW23] and verification [DL14; GD18]. Inspired by these advances for qubit models, this work aims to provide a similar representation for linear optical circuits: both formal and graphical.

In this chapter, we introduce several graphical languages, covering various kinds of linear optical setups prominently used in the literature. Our work is the first to provide a complete description in all these settings, bringing new state-of-the-art theoretical frameworks for describing linear optical circuits. The chapter is structured as follows.

- Section III.1. We introduce the $\mathbf{LO}_{\mathbf{PP}}$-calculus (Linear Optical Polarizing-Preserving circuits), a graphical language for modeling circuits made of beam splitters and phase shifters. We present a minimal complete equational theory, alongside crucial unique forms and a deterministic rewriting system. This work has contributed to both [Clé+22] and [Heu25].

- Section III.2. We introduce the $\mathbf{LO}_{\mathbf{v}}$-calculus (Linear Optical circuits with vacuum generation and detection), an extension of $\mathbf{LO}_{\mathbf{PP}}$ with components that act on polarization, along with vacuum sources and postselection. We provide a complete equational theory and compact forms for such circuits. This work led to [Clé+22].

- Section III.3. We introduce the $\mathbf{LO}_{fi}$-calculus (Linear Optical circuits with finite-photon-number sources and detectors), a graphical language that extends $\mathbf{LO}_{\mathbf{PP}}$ with sources and detectors allowing arbitrary finite-dimensional Fock states, for which we present a complete equational theory. We introduce a deterministic rewriting procedure and a new basis of operators that help exhibit a unique form for such circuits. This work is described in [Heu25].

Related work

For capturing bosonic linear optics with pictures, some methods that have been proposed [KGZ17; Ata18] use graph theory. Closer to our goal, and using category theory, an essay exploring the links between bosonic linear optics and category theory has been presented [McC22].

However, these works are still different from what we aim to achieve, i.e., providing a complete and formal graphical framework for manipulating linear optical circuits. In this regard, two prominent works have emerged.

One prior work to this thesis is the **PBS**-calculus [CP20], which provides a complete equational theory for circuits made of polarizing beam splitters and polarization flips. While insightful and promising for studying quantum control, this framework does not capture the spatial modes of linear optics, e.g., it does not include phase shifters and beam splitters.

The other is **QPath** [FC22], published shortly after our first work [Clé+22]. Its generators differ greatly from physical components, using more abstract string diagrams. The advantage is that it brings the framework closer to languages like the ZX-calculus [Fel+23], and thus to qubit representations [Fel+24]. The drawbacks are that it does not naturally capture linear optical circuits, requiring an extraction procedure, and its semantics — based on unbounded operators instead of unitary ones — cannot be extended to infinite-dimensional Hilbert spaces such as the Fock space.

In this thesis, we propose formal graphical frameworks that capture the core components of linear optical circuits, while remaining compatible with infinite-dimensional semantics. These frameworks come with complete equational theories, normal forms, and new methods that offer both theoretical insights and practical tools for researchers and experimental physicists in the field.

III.1 $\mathbf{LO_{PP}}$: Linear optical polarizing-preserving circuit

In this section, we present the $\mathbf{LO_{PP}}$-calculus (Linear Optical Polarizing-Preserving circuit). We begin by introducing its syntax and semantics (Section III.1.1). Next, we provide a complete equational theory that we prove to be minimal (Section III.1.2). We then depict key forms of circuits and their properties (Section III.1.3) and provide a deterministic rewriting system converging to one of these forms (Section III.1.4). Finally, we give the full proof of the completeness of the equational theory previously introduced (Section III.1.5).

Contributions

The $\mathbf{LO_{PP}}$-calculus, the rewriting system, and the proof of completeness originate from [Clé+22], published in MFCS'22, and coauthored with Alexandre Clément, Shane Mansfield, Simon Perdrix, and Benoît Valiron.

The minimal equational theory greatly benefited from discussions with A. Clément.

The remainder of the section, including simplified proofs and new triangular forms, comes from a personal contribution [Heu25], published in CSL'25.

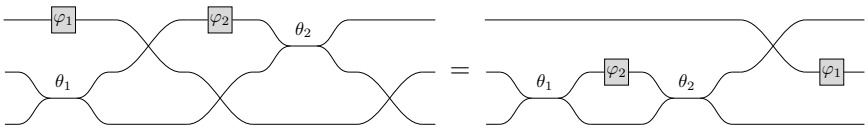
III.1.1 Syntax and single-photon semantics

In order to formalize linear optical circuits with diagrams, we use the formalism of PROPs (Definition I.74). We introduce the following PROP, constituted of beam splitters and phase shifters, which are linear optical components that preserve the polarization of the photons.

Definition III.1. **LO$_{PP}$** is the PROP [1] of **LO$_{PP}$**-circuits generated by:

The convention is to go through from left to right, meaning the inputs (resp. outputs) are on the left (resp. right), and from top to bottom, meaning the first (resp. last) input is the top (resp. last) wire. The identity, the swap and the empty diagrams are noted as follows: —, ⨉, ⬚.

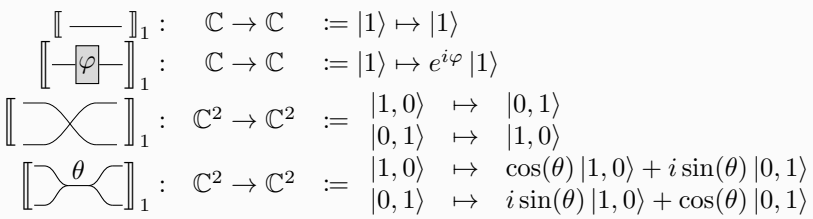
Example III.2. Here are two examples of **LO$_{PP}$**-circuits, that are equivalent up to deformation with the rules of PROPs:



When we are only interested in linear optical components, the semantics of such circuits can be fully described by their behavior when there is one single photon passing through those components (Section I.3.3). This is what we refer as single-photon semantics.

The phase shifters and beam splitters are components only acting on the spatial modes, i.e. the positions of the photon. Given a circuit of m wires, the single photon can be in a superposition of the m different positions, so its state is a vector in $\mathbb{C}^m$. We consider the standard orthonormal basis $\{|e_i\rangle, i \in [1, m]\}$ where $e_i = |0, \dots, 0, 1, 0, \dots, 0\rangle$ with a 1 at the i^{th} component. Formally, the object of our PROP is therefore $X = \mathbb{C}$, and the vertical composition is interpreted as the direct sum. The semantics is defined as follows.

Definition III.3 (Single-photon semantics of **LO$_{PP}$**)**.** The single-photon semantics of **LO$_{PP}$** is inductively defined as follows: $[\![C_1 \otimes C_2]\!]_1 = [\![C_1]\!]_1 \oplus [\![C_2]\!]_1$, $[\![C_2 \circ C_1]\!]_1 = [\![C_2]\!]_1 \circ [\![C_1]\!]_1$ and:



It is also usual to represent those linear operators as matrices, with

$$[\![C_1]\!]_1 \oplus [\![C_2]\!]_1 = \left(\begin{array}{c|c} [\![C_1]\!]_1 & 0 \\ \hline 0 & [\![C_2]\!]_1 \end{array}\right)$$

and for instance

$$[\![\ \theta\ ]\!]_1 = \begin{pmatrix} \cos(\theta) & i\sin(\theta) \\ i\sin(\theta) & \cos(\theta) \end{pmatrix}$$

.

Remark III.4 (Reflexivity and transmitivity). If a photon arrives from one wire, the amplitude to stay on the same (resp. other) wire is $\cos(\theta)$ (resp. $\sin(\theta)$). The reflexivity (resp. transmitivity), i.e. the probability of being reflected (resp. transmitted) is therefore $R(\theta) = \cos(\theta)^2$ (resp. $T(\theta) = \sin(\theta)^2$).

1. When first introduced in [Clé+22], **LO$_{PP}$** was defined as a PRO (Definition I.73), and the syntax of **LOPP** was used in [Clé+23] to define the PROP. To unify the conventions of non-capitalized letters after **LO**, we now choose to use **LO$_{PP}$** for the PROP and **LO$_{PP}^{PRO}$** for the PRO, respectively (Definition III.29).

Remark III.5 (Conventions for the beam splitter). Interestingly, one could have chosen a different convention for the semantics of the parameterized beam splitter. The chosen convention is convenient for the matrix is symmetric and is a rotation. See Remark I.33 for further details.

III.1.2 Minimal complete equational theory of LO$_{PP}$

Two distinct **LO$_{PP}$**-circuits may represent the same quantum evolution. For instance, shifting the phase of a photon by two phase shifters of phase φ_1 and φ_2 is the same as shifting it with one phase $\varphi_1 + \varphi_2$. In order to graphically characterize those equivalences, we introduce an equational theory of **LO$_{PP}$** in Figure III.1 (Section III.1.2.A.). We then prove the soundess of this equational theory (Section III.1.2.B.) and its minimality (Section III.1.2.C.).

A. Equational theory of LO$_{PP}$

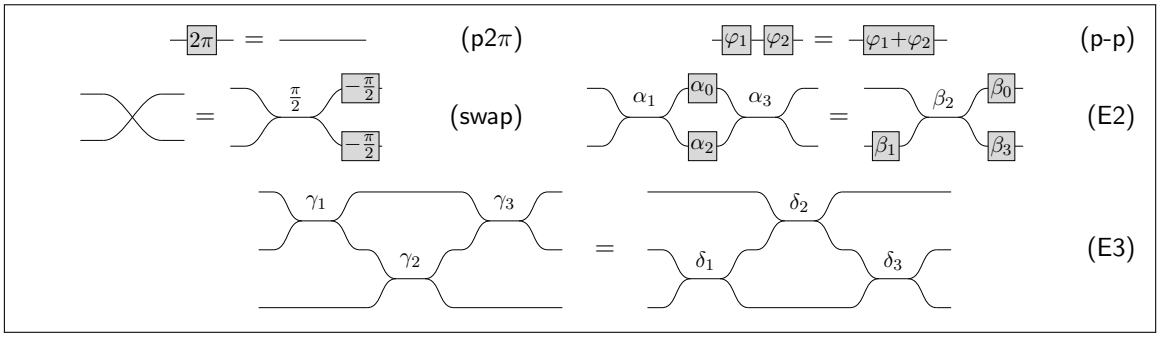


Figure III.1 – Minimal equational theory of the **LO$_{PP}$**-calculus. For any angle of the LHS (resp. RHS) of the Equation (E2) and (E3), there exist angles for the RHS (resp. LHS) such that the equations are sound. The angles are unique if we restrict $\alpha_0, \alpha_2, \beta_0, \beta_1, \beta_3 \in [0, 2\pi)$, $\alpha_1 \in [0, \frac{\pi}{2}), \alpha_3 \in [0, \pi)$, $\beta_2 \in [0, \frac{\pi}{2}]$, and by taking $\alpha_1 = 0$ if $\alpha_0 - \alpha_2 = 0 \mod \pi$ and $\beta_1 = 0$ if $\beta_2 \in \{0, \frac{\pi}{2}\}$. The rotations associated with Equations (E2) and (E2) and the explicit values of the angles are detailed in Proposition III.10 and III.11.

Definition III.6 (**LO$_{PP}$**-calculus)**.** Two **LO$_{PP}$**-circuits D, D' are equivalent according to the rules of the **LO$_{PP}$**-calculus, denoted **LO$_{PP}$** $\vdash D = D'$, if one can transform D into D' using the equations given in Figure III.1. More precisely, **LO$_{PP}$** $\vdash \cdot = \cdot$ is defined as the smallest congruence which satisfies the equations of Figure III.1 and the axioms of PROP.

Remarkably, the equational theory of Figure III.1 is sound, minimal and complete, that we formalize with the three following theorems.

Theorem III.7 (Soundness)**.** *For any two* **LO$_{PP}$***-circuits D_1 and D_2, if* **LO$_{PP}$** $\vdash D_1 = D_2$ *then* $[\![D_1]\!] = [\![D_2]\!]$.

Theorem III.8 (Minimality)**.** *The equational theory of Figure III.1 is minimal for* **LO$_{PP}$***-circuits, i.e. none of its equations can be derived from the others.*

Theorem III.9 (Completeness of **LO$_{\mathsf{PP}}$**). *For any two* **LO$_{\mathsf{PP}}$***-circuits D and D', if $[\![D]\!]_1 = [\![D']\!]_1$ then* **LO$_{\mathsf{PP}}$** $\vdash D = D'$.

We respectively prove the soundness and the minimality in the following sections (Section III.1.2.B. and III.1.2.C.). The proof for the completeness needs the tools introduced in Section III.1.3 and III.1.4, and is therefore provided in Section III.1.5.

B. Soundess of LO$_{\mathsf{PP}}$

Since semantic equality is a congruence, it suffices to check the soundness for every equation of Figure III.1. The soundness of (swap), p-p and p2π are direct consequences of Definition III.3. By definition, Euler equations (E2) and (E3) are necessarily sound as we only consider sets of angles that ensure their correctness. The difficulty resides in making these angles explicit, which we do in the following propositions. The proof of Proposition III.10 relies on the unitary representation of 2×2 matrix, where we show that any unitary can be decomposed with such circuits. The proof of Proposition III.11 makes the links with the real Euler angles for 3D rotations [GPS02], where we take the explicit formulae for the angles from [Ebe20].

Proposition III.10 (Angles of (E2)). *For any $\alpha_0, \alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}^4$, there exist $\beta_0, \beta_1, \beta_2, \beta_3 \in \mathbb{R}^4$ such that (E2) is sound. Reciprocally, for any $\beta_0, \beta_1, \beta_2, \beta_3 \in \mathbb{R}^4$, there exist $\alpha_0, \alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}^4$ such that (E2) is sound.*

Proof. First, let us notice that

$$\left[\!\left[\begin{array}{c} \\ \theta \\ \\ \end{array} \right]\!\right]_1 = R_X(-2\theta) \qquad \left[\!\left[\begin{array}{c} \overline{} \\ \boxed{\varphi} \\ \end{array} \right]\!\right]_1 = e^{i\frac{\varphi}{2}} R_Z(\varphi)$$

where R_X (resp. R_Z) is the rotation operator about the $\hat{x}$ axis (resp. $\hat{z}$ axis) of the Bloch sphere [NC10]. Any unitary 2×2 unitary U can be decomposed [Bar+95] into

$$e^{i\cdot} R_X(\cdot) R_Z(\cdot) R_X(\cdot) = U = e^{i\cdot} R_Z(\cdot) R_X(\cdot) R_Z(\cdot)$$

giving the LHS and RHS angles of (E2).

Let us note

$$U_{E_2} \coloneqq \begin{pmatrix} u_{1,1} & u_{1,2} \\ u_{2,1} & u_{2,2} \end{pmatrix} = [\![LHS]\!]_1 = [\![RHS]\!]_1 .$$

It is straightforward to compute the elements of U_{E_2} with either the angles of the LHS or the RHS from Definition III.3. Therefore, it is sufficient to give the angles from the elements of U_{E_2}. In the following paragraphs, we first give one solution for the angles, then specify all the angles that satisfy the equations.

From U_{E_2} to RHS.

We have:

$$[\![RHS]\!]_1 = \begin{pmatrix} e^{i\beta_0}\cos(\beta_2) & ie^{i(\beta_0+\beta_1)}\sin(\beta_2) \\ ie^{i\beta_3}\sin(\beta_2) & e^{i(\beta_1+\beta_3)}\cos(\beta_2) \end{pmatrix} = \begin{pmatrix} u_{1,1} & u_{1,2} \\ u_{2,1} & u_{2,2} \end{pmatrix}$$

The following is a solution:

$$\begin{cases} \beta_0 = \arg(u_{1,1}), \\ \beta_1 = \arg(u_{2,2}) - \arg(u_{2,1}) + \dfrac{\pi}{2}, \\ \beta_2 = \cos^{-1}\left(|u_{1,1}|\right), \\ \beta_3 = \arg(u_{2,1}) - \dfrac{\pi}{2}, \end{cases} \tag{III.1}$$

We can note that:

- if $\beta_2 \mapsto \beta_2 + \pi$ then the RHS is the same with $\beta_0 \mapsto \beta_0 + \pi$ and $\beta_3 \mapsto \beta_3 + \pi$,
- if $\beta_2 \mapsto \pi - \beta_2$ then the RHS is the same with $\beta_0 \mapsto \beta_0 + \pi$ and $\beta_1 \mapsto \beta_1 + \pi$,
- all the angles can be taken modulo 2π without changing the RHS.

The complete set of solutions for $\beta_0, \beta_1, \beta_2$ and β_3 includes all possible configurations that can be derived from the values of (III.1) and the three transformations outlined above. Note there is a unique set of angles such that $\beta_2 \in [0, \frac{\pi}{2}]$, $\beta_0, \beta_1, \beta_3 \in [0, 2\pi)$ satisfying $\beta_1 = 0$ if $\beta_2 \in \{0, \frac{\pi}{2}\}$. In particular, if $|u_{1,1}| = 1$, then $\beta_2 = 0$, $\beta_1 = 0$, and if $|u_{1,1}| = 0$, then $\beta_2 = \frac{\pi}{2}$ and $\beta_1 = 0$.

From U_{E_2} to LHS.

Let us first notice that:

$$HP(\varphi)H = e^{i\frac{\varphi}{2}} R_X(\varphi) = e^{i\frac{\varphi}{2}} B(-\frac{\varphi}{2})$$

where H is the Hadamard matrix (Table I.1). We can therefore come back to a form similar to the RHS:

$$\begin{aligned} H \left[\!\left[LHS \right]\!\right]_1 H &= e^{i\alpha_0} H \left(B(\alpha_1) P(\alpha_2 - \alpha_0) B(\alpha_3) \right) H \\ &= e^{i\alpha_0} \left(HB(\alpha_1)H \right) \left(HP(\alpha_2 - \alpha_0)H \right) \left(HB(\alpha_3)H \right) \\ &= e^{i\alpha_0} \left(e^{i\alpha_1} P(-2\alpha_1) \right) \left(e^{i\frac{\alpha_2 - \alpha_0}{2}} B\left(-\tfrac{\alpha_2 - \alpha_0}{2}\right) \right) \left(e^{i\alpha_3} P(-2\alpha_3) \right) \\ &= e^{i\left(\frac{\alpha_0 + 2\alpha_1 + \alpha_2 + 2\alpha_3}{2}\right)} P(-2\alpha_1) B\left(\tfrac{\alpha_0 - \alpha_2}{2}\right) P(-2\alpha_3) \end{aligned}$$

By noting $\begin{pmatrix} u'_{1,1} & u'_{1,2} \\ u'_{2,1} & u'_{2,2} \end{pmatrix} = HUH$, and from the previous paragraph we can infer the following solution:

$$\begin{cases} \dfrac{\alpha_0 + 2\alpha_1 + \alpha_2 + 2\alpha_3}{2} = \arg(u'_{1,1}), \\ \dfrac{\alpha_0 - \alpha_2}{2} = \cos^{-1}\left(|u'_{1,1}|\right), \\ -2\alpha_3 + \dfrac{\alpha_0 + 2\alpha_1 + \alpha_2 + 2\alpha_3}{2} = \arg(u'_{2,1}) - \dfrac{\pi}{2}, \\ -2\alpha_1 = \arg(u'_{2,2}) - \arg(u'_{2,1}) + \dfrac{\pi}{2} \end{cases}$$

Thus we have the following solution:

$$\begin{cases} \alpha_0 = \arg(u'_{1,1}) + \arg(u'_{2,2}), \\ \alpha_1 = \dfrac{\arg(u'_{2,1}) - \arg(u'_{2,2})}{2} - \dfrac{\pi}{4} \\ \alpha_2 = \arg(u'_{1,1}) + \arg(u'_{2,2}) - 2\cos^{-1}\left(|u'_{1,1}|\right), \\ \alpha_3 = \cos^{-1}\left(|u'_{1,1}|\right) - \dfrac{\arg(u'_{2,1}) + \arg(u'_{2,2})}{2} + \dfrac{\pi}{4}, \end{cases} \tag{III.2}$$

We can note that:

- if $\alpha_1 \mapsto \alpha_1 + \pi$ then the LHS is the same with $\alpha_0 \mapsto \alpha_0 + \pi$ and $\alpha_2 \mapsto \alpha_2 + \pi$,
- if $\alpha_3 \mapsto \alpha_3 + \pi$ then the LHS is the same with $\alpha_0 \mapsto \alpha_0 + \pi$ and $\alpha_2 \mapsto \alpha_2 + \pi$,
- all the angles can be taken modulo 2π without changing the LHS.

The complete set of solutions for $\alpha_0, \alpha_1, \alpha_2$ and α_3 includes all possible configurations that can be derived from the values of (III.2) and the three transformations outlined above. Note there is a unique set of angles such that $\alpha_1 \in [0, \frac{\pi}{2}), \alpha_3 \in [0, \pi)$, $\alpha_0, \alpha_2 \in [0, 2\pi)$ satisfying $\alpha_1 = 0$ if $\alpha_0 - \alpha_2 = 0 \mod \pi$. In particular, if $|u_{1,2}| = 0$, then $\alpha_0 - \alpha_2 = 0 \mod \pi$ and $\alpha_1 = \alpha_3 = 0$, and if $|u_{1,1}| = 0$, then $\alpha_0 - \alpha_2 = 0 \mod \pi$, $\alpha_1 = 0$ and $\alpha_3 = \frac{\pi}{2}$.

□

Proposition III.11 (Angles of (E3)). *For any $\gamma_1, \gamma_2, \gamma_3 \in \mathbb{R}^3$, there exist $\delta_1, \delta_2, \delta_3 \in \mathbb{R}^3$ such that (E3) is sound. Reciprocally, for any $\delta_1, \delta_2, \delta_3 \in \mathbb{R}^3$, there exist $\gamma_1, \gamma_2, \gamma_3 \in \mathbb{R}^3$ such that (E3) is sound.*

Proof. By transforming the iY-axis into the Y-axis, the three rotations of the LHS (resp. RHS) of (E3) can be seen as three real rotations along the $z - x - z$ real axes (resp. $x - z - x$). The angles will be therefore given by the Euler angles [GPS02]. Mathematically, we have

$$P_{iy}^\dagger B_{1,2}(\theta) P_{iy} = R_z(\theta) \qquad P_{iy}^\dagger B_{2,3}(\theta) P_{iy} = R_x(\theta)$$

with $P_{iy}^\dagger P_{iy} = I_3$. Thus,

$$\begin{aligned} P_{iy}^\dagger \llbracket LHS \rrbracket_1 P_{iy} &= P_{iy}^\dagger B_{1,2}(\gamma_1) B_{2,3}(\gamma_2) B_{1,2}(\gamma_3) P_{iy} \\ &= \left(P_{iy}^\dagger B_{1,2}(\gamma_1) P_{iy}\right) \left(P_{iy}^\dagger B_{2,3}(\gamma_2) P_{iy}\right) \left(P_{iy}^\dagger B_{1,2}(\gamma_3) P_{iy}\right) \\ &= R_z(\gamma_1) R_x(\gamma_2) R_z(\gamma_3) \end{aligned}$$

and

$$\begin{aligned} P_{iy}^\dagger \llbracket RHS \rrbracket_1 P_{iy} &= P_{iy}^\dagger B_{2,3}(\delta_1) B_{1,2}(\delta_2) B_{2,3}(\delta_3) P_{iy} \\ &= \left(P_{iy}^\dagger B_{2,3}(\delta_1) P_{iy}\right) \left(P_{iy}^\dagger B_{1,2}(\delta_2) P_{iy}\right) \left(P_{iy}^\dagger B_{2,3}(\delta_3) P_{iy}\right) \\ &= R_x(\delta_1) R_z(\delta_2) R_x(\delta_3) \end{aligned} .$$

The angles of (E3) are exactly the Euler angles for rotations in $\mathbb{R}^3$.

$$\begin{aligned} & \llbracket LHS \rrbracket_1 &=& \llbracket RHS \rrbracket \\ \Leftrightarrow \quad & P_{iy}^\dagger \llbracket LHS \rrbracket_1 P_{iy} &=& P_{iy}^\dagger \llbracket RHS \rrbracket P_{iy} \\ \Leftrightarrow \quad & R_z(\gamma_3) R_x(\gamma_2) R_z(\gamma_1) &=& R_x(\delta_3) R_z(\delta_2) R_x(\delta_1) \end{aligned}$$

In order to have the explicit formulae, let us define the 3×3 real matrix R_{E_3}:

$$R_{E_3} \coloneqq \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} \\ r_{2,1} & r_{2,2} & r_{2,3} \\ r_{3,1} & r_{3,2} & r_{3,3} \end{pmatrix} = P_{iy}^\dagger \llbracket LHS \rrbracket_1 P_{iy} = P_{iy}^\dagger \llbracket RHS \rrbracket_1 P_{iy}.$$

It is straightforward to compute the elements of R_{E_3} with either the angles of the LHS or the RHS with Definition III.3. Therefore, it is sufficient to give the angles from the elements of R_{E_3}. Here, we give the formulae directly taken from [Ebe20].

From R_{E_3} to RHS.

- If $-1 < r_{1,1} < 1$ then $\delta_1 = \operatorname{atan2}(r_{1,3}, -r_{1,2})$, $\delta_2 = \cos^{-1}(r_{1,1})$ and $\delta_3 = \operatorname{atan2}(r_{3,1}, r_{2,1})$.
- If $r_{1,1} = 1$, then necessarily $\delta_2 = 0$, and the solutions are all the angles such that $\delta_1 + \delta_3 = \operatorname{atan2}(r_{3,2}, r_{3,3})$.
- If $r_{1,1} = -1$, then necessarily $\delta_2 = \pi$, and the solutions are all the angles such that $\delta_1 - \delta_3 = \operatorname{atan2}(r_{3,2}, r_{3,3})$.

All those angles can be taken modulo 2π.

From R_{E_3} to LHS.

- If $-1 < r_{3,3} < 1$ then $\gamma_2 = \cos^{-1}(r_{3,3})$, $\gamma_3 = \mathrm{atan2}(r_{1,3}, -r_{2,3})$ and $\gamma_1 = \mathrm{atan2}(r_{3,0}, r_{3,2})$.
- If $r_{3,3} = 1$, then necessarily $\gamma_2 = 0$, and the solutions are all the angles such that $\gamma_1 + \gamma_3 = \mathrm{atan2}(-r_{1,2}, r_{1,1})$.
- If $r_{3,3} = -1$, then necessarily $\gamma_2 = \pi$, and the solutions are all the angles such that $\gamma_1 - \gamma_3 = \mathrm{atan2}(-r_{1,2}, r_{1,1})$.

All those angles can be taken modulo 2π.

□

C. Minimal equational theory

For proving the minimality, we proceed in two steps. First we prove that all the rules aside from (E3) are necessary. Then we prove that (E3) is necessary.

Proposition III.12. *The equations (p2π), (p-p), (swap) and (E2) cannot be derived from the others.*

Proof. We define an alternative interpretation which satisfies all the equations aside from the one we prove to be necessary. In particular:

- (p2π) is the only rule on one wire that changes the sum of the phases.
- (p-p) is the only rule on one wire that can reduce the number of phases different from 2π.
- (swap) is the only rule changing the parity of the number of SWAPs.
- (E2) is the only rule changing the parity of (number of beam splitter + number of SWAPs).

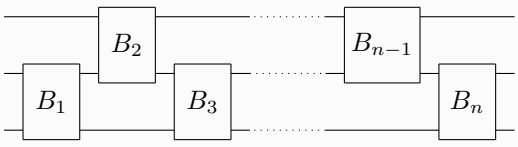
□

Proposition III.13. *The equation (E3) cannot be derived from the other equations of Figure III.1.*

Proof. For proving the necessity of (E3), we proceed as follows.

- We define an equivalence relation $\sim_\varphi$ on three-wire **LO$_{\mathsf{PP}}$**-circuits (Definition III.15).
- We introduce a confluent rewriting procedure (Definition III.16) that is conserving the relation $\sim_\varphi$, and that is converging to normal forms.
- All rules of PROPs, and all other equations of **LO$_{\mathsf{PP}}$**conserve the relation $\sim_\varphi$.
- We show that the LHS and RHS of (E3) are different normal forms.
- We conclude that (E3) is necessary, because we cannot transform the LHS into the RHS with any other rule than (E3).

Definition III.14 (Minimal block decomposition)**.** Let D be a **LO$_{\mathsf{PP}}$**-circuit on three wires. We define the minimal block decomposition as the minimal partition of D into sub-circuits $B_1, B_2, \ldots, B_n$ such that D, up to deformation, is of the form:



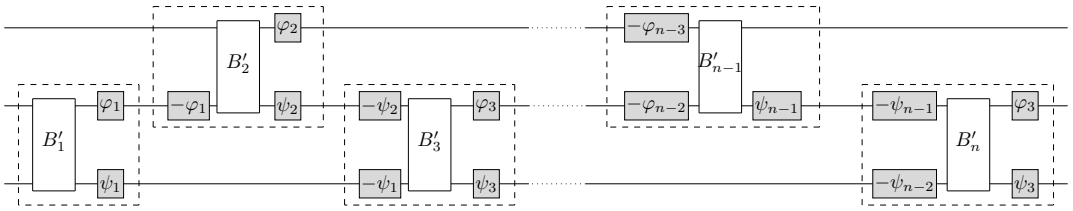
where for each i, B_i contains as many as phase shifters as possible that are not part of the

$B_1, \ldots, B_{i-1}$, and n is minimal.

Note that B_1 and B_n can be identities or swap, and that B_{n-1} can be a SWAP (but not the identity).

Definition III.15 (Equivalence relation)**.** Let D and D' be two three-wire **LO$_{\mathsf{PP}}$**-circuits. We have $D \sim_\varphi D'$ if and only if their minimal block decomposition $B_1, \ldots, B_n$ and $B'_1, \ldots, B'_n$ have the same number of blocks, and are semantically equal up to phases that can be exchanged between the blocks.

It means that $D \sim_\varphi D'$ if there exists $\varphi_1, \ldots, \varphi_n, \psi_1, \ldots, \psi_n$ such that the semantics of the i^{th} dash box of the following figure:

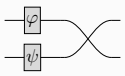


is equal to $\left[\!\left[B_i \right]\!\right]_1$.

Definition III.16 (Rewriting system)**.** We consider the rewriting system that is rewriting each block B_i containing at least one beam splitter by

if $\quad \left[\!\left[B_i \right]\!\right]_1 = \begin{pmatrix} e^{i\varphi} & 0 \\ 0 & e^{i\psi} \end{pmatrix}$

and by



if $\quad \left[\!\left[B_i \right]\!\right]_1 = \begin{pmatrix} 0 & e^{i\psi} \\ e^{i\varphi} & 0 \end{pmatrix}$.

We can notice that:

- The system strictly decreases the number of beam splitters, hence is strongly normalizing.
- It follows from the definition of $\sim_\varphi$ that the systems preserves $\sim_\varphi$.
- The system is locally confluent. Starting from D, if we rewrite B_i giving D_1 or B_j giving D_2, we can independently rewrite B_i in D_2 and B_j in D_1, giving the same circuit D'.

Finally, we can notice that the rules (p2π), (swap), (p-p), (E2) and all the rules of the PROP also conserve the relation $\approx_1$.

As the LHS and the RHS are distinct normal forms of our system, we know they can't be derived from the rules conserving $\sim_\varphi$. Therefore, (E3) is necessary.

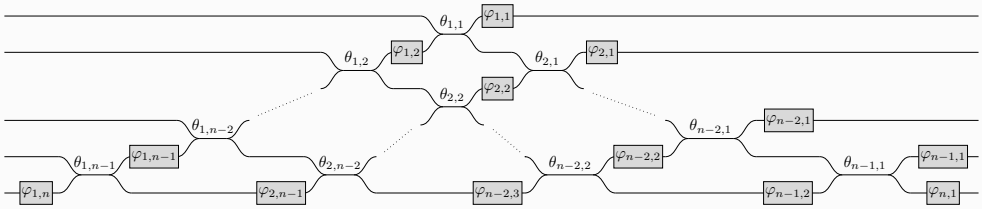
□

III.1.3 Universality and triangular LO$_{\mathsf{PP}}$-circuits

We introduce three classes of **LO$_{\mathsf{PP}}$**-circuits, with the following inclusions: ${}_{\tilde{n}}\Diamond_n \subset {}_{\tilde{n}}\triangle_{\tilde{m}} \subset \triangle$. They are of particular interest as $\triangle$-circuits are the normal forms of the **LO$_{\mathsf{PP}}$**-calculus (Lemma III.31), ${}_{\tilde{n}}\triangle_{\tilde{m}}$-circuits will be used in the normal forms of both the **LO$_{\mathsf{v}}$**-calculus (Definition III.59), and the **LO$_{\boldsymbol{fi}}$**-calculus (Definition III.91) where their uniqueness will be proved thanks to ${}_{\tilde{n}}\Diamond_n$-circuits (Section III.3.4). Their shape and properties are illustrated and summarized in Table III.1.

A. Triangular LO$_{PP}$-circuits: $\triangle$-circuits

Definition III.17 ($\triangle$-circuits)**.** A $\triangle$-circuit is a **LO$_{PP}$**-circuit with the following shape:



with $\varphi_{i,j} \in [0, 2\pi), \theta_{i,j} \in [0, \frac{\pi}{2}]$ and the following conditions: $\theta_{i,j} = 0 \Rightarrow (\forall j' > j, \varphi_{i,j'} = \theta_{i,j'} = 0)$ and $\theta_{i,j} = \frac{\pi}{2} \Rightarrow \varphi_{i,j} = 0$. $\theta_{i,j}$ is on the i^{th} right (resp. j^{th} left) diagonal, and on the $(i+j-1)^{th}$ row of beam splitters.

Definition III.18 ($\triangle^*$-circuits)**.** A $\triangle^*$-circuit is a $\triangle$-circuit where all the 0-angle phase shifters and beam splitters are identities instead.

Remark III.19 (Coefficients of $[\![\triangle]\!]_1$). The coefficient $t_{i,j}$ of the $n \times n$ matrix $[\![\triangle]\!]_1$ is determined by the sum of all the paths from the j^{th} input wire to the i^{th} output wire, where for each path we multiply by a $\cos$ (resp. $i \sin$) term when the photon is reflected on (resp. transmitted through) a beam splitter, and by a phase when the path crosses a phase shifter. For instance:

$$\begin{aligned} t_{1,2} &= \cos(\theta_{1,2}) e^{i\varphi_{1,2}} i \sin(\theta_{1,1}) e^{i\varphi_{1,1}} \text{ and} \\ t_{2,2} &= i \sin(\theta_{1,2}) \cos(\theta_{2,2}) e^{\varphi_{2,2}} i \sin(\theta_{2,1}) e^{i\varphi_{2,1}} + \cos(\theta_{1,2}) e^{\varphi_{1,2}} i \sin(\theta_{1,1}) \cos(\theta_{2,1}) e^{\varphi_{1,2}}. \end{aligned}$$

. More generally, we have

$$t_{i,j} = e^{i\varphi_{i,j}} \cos(\theta_{i,j}) \times q_{i,j} + r_{i,j} \tag{III.3}$$

where $q_{i,j}, r_{i,j}$ are terms depending uniquely on the angles with lower indexes.

Proposition III.20 (Universality and uniqueness of T)**.** *For any unitary matrix $U : n \times n$, there exists a unique circuit $T : n \to n$ in triangular form of Definition III.17 such that $U = [\![T]\!]_1$.*

Proof. Note that the existence is a direct consequence of [Rec+94], that describes an algorithm to find such T given a unitary U. We can prove the uniqueness by induction as follows.

First, we prove the base case for $n = 2$. For any z with $0 < |z| \leq 1$, there are unique $\varphi, \theta \in [0, 2\pi) \times [0, \frac{\pi}{2})$ such that $e^{i\varphi} \cos(\theta) = z$, with $\varphi, \theta = (0, 0)$ for the special case of $z = 0$. The case for $n = 1$ is straightforward.

For the generic case, following from Equation (III.3), let us notice that $t_{i,j} = e^{i\varphi_{i,j}} \cos(\theta_{i,j}) \times q_{i,j} + r_{i,j}$ where $q_{i,j}, r_{i,j}$ are terms depending uniquely on the angles with lower indexes. We aim to show that $(\varphi_{i,j}, \theta_{i,j})$ are indeed uniquely determined by $t_{i,j}$ and coefficients of lower indexes, concluding the induction proof.

Note that $q_{i,j} \neq 0$ if and only if all $\theta_{k<i,j}$ and $\theta_{i,\ell<j}$ are nonzero. If one $\theta_{i,\ell<j}$ is zero, then we have $\varphi_{i,j} = \theta_{i,j} = 0$ by the properties of the $\triangle$-circuits. If there are K values of $\theta_{k<i,j}$ which are zero, then all the K diagonals $\theta_{k,\ell' \geq j}$ are zero. By now considering the path from the $(j+K)^{th}$ input wire to the i^{th} output wire, we recover the same type of Equation (III.3) with $q_{i,j} \neq 0$. Now, we can subtract $r_{i,j}$ and dividing by $q_{i,j}$, so that we have $e^{i\varphi_{i,j}} \cos(\theta_{i,j}) = z_{i,j}$ with $z_{i,j} = (t_{i,j} - r_{i,j})/q_{i,j}$. If $z_{i,j} \neq 0$ then $\theta_{i,j} \in [0, \frac{\pi}{2})$ and $\varphi \in [0, 2\pi)$ are uniquely determined. If $z_{i,j} = 0$, then $\theta_{i,j} = \frac{\pi}{2}$ and by the properties of $\triangle$-circuits, we have $\varphi_{i,j} = 0$. □

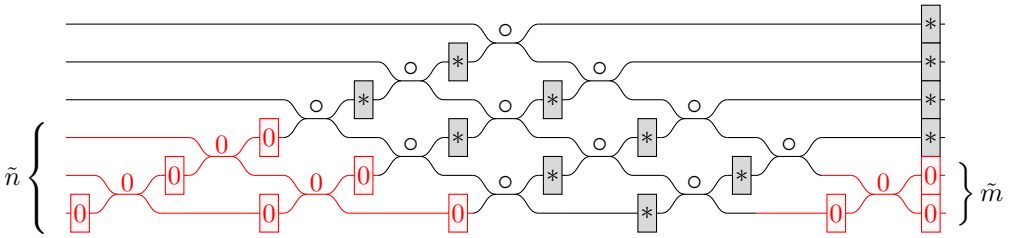
Remark III.21. A generic $\triangle$-circuit $T : n \to n$ has $\frac{n(n-1)}{2}$ beam splitters and $\frac{n(n+1)}{2}$ phase shifters, having a total of n^2 different angles, that is in adequacy with the dimension of the unitary group $U(n)$.

B. Subtriangular LO$_{PP}$-circuits: $_{\tilde{n}}\triangle_{\tilde{m}}$-circuits

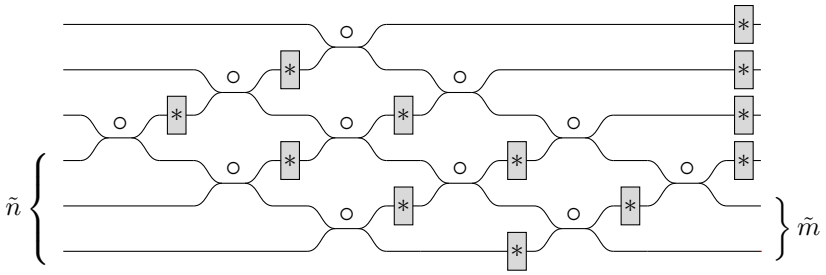
Definition III.22 ($_{\tilde{n}}\triangle_{\tilde{m}}$-circuits)**.** A **LO$_{PP}$**-circuit $\tilde{\triangle} : n + \tilde{n} \to m + \tilde{m}$ is a $_{\tilde{n}}\triangle_{\tilde{m}}$-circuit if:

1. $\tilde{\triangle}$ is a $\triangle$-circuit as defined in Definition III.17,
2. there is no beam splitter or phase shifter directly connected to the $\tilde{n}$ last input wires, ie. $\varphi_{i,j} = \theta_{i,j} = 0$ if $row_{i,j} = i + j - 1 > n$ and there doesn't exist (k, ℓ) such that $k + \ell - 1 = row_{i,j} - 1$, $k < i$ and $\theta_{k,\ell} \neq 0$,
3. there is no beam splitter or phase shifter directly connected to the $\tilde{m}$ last output wires, i.e. $\varphi_{i,j} = \theta_{i,j} = 0$ if $row_{i,j} = i + j - 1 > m$ and there doesn't exist (k, ℓ) such that $k + \ell - 1 = row_{i,j} - 1$, $k \geq i$ and $\theta_{k,\ell} \neq 0$, and
4. there exists one nonzero $\theta_{i,j}$ for each of the last $\max(\tilde{n}, \tilde{m})$ rows[2].

Example III.23. For instance, for $\tilde{n} = 3$ and $\tilde{m} = 2$, a $_{\tilde{n}}\triangle_{\tilde{m}}$-circuit is



where the 0-angle are identities:



where $*$ indicates any angle in $[0, 2\pi)$ and $\circ$ any angle in $[0, \frac{\pi}{2}]$.

Note that the second and third conditions described in Definition III.22 imply that the only nonzero angles have indices $(i \leq m, j \leq n)$. This leads to the following proposition, direct consequence of Remark III.19 and the proof of Proposition III.20.

Proposition III.24 (Uniqueness of $_{\tilde{n}}\triangle_{\tilde{m}}$-circuits on their $m \times n$ submatrix)**.** *For any $_{\tilde{n}}\triangle_{\tilde{m}}$-circuits $\triangle, \triangle' : n + \tilde{n} \to m + \tilde{m}$,*

$$[\![\triangle]\!]_1 (1 : m, 1 : n) = [\![\triangle']\!]_1 (1 : m, 1 : n) \Rightarrow \triangle = \triangle'$$

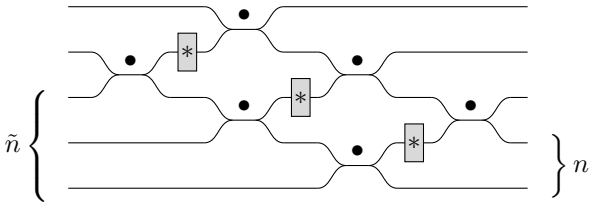
where $M(1 : k, 1 : \ell)$ is the $k \times \ell$ matrix composed of the first k rows and ℓ columns of M.

2. This property is an additional constraint that appears in the normal forms defined in Definition III.91.

C. Diamond LO$_{\mathsf{PP}}$-circuits: ${}_{\tilde{n}}\Diamond_{n}$-circuits

Definition III.25 (${}_{\tilde{n}}\Diamond_{n}$-circuits)**.** A ${}_{\tilde{n}}\triangle_{\tilde{m}}$-circuit $\tilde{\triangle} : n+\tilde{n} \to m+\tilde{m}$ is a ${}_{\tilde{n}}\Diamond_{n}$-circuit if $\tilde{m} = n$ and if there is no phase shifter on the m output wires.

Example III.26. For instance, for $\tilde{n} = 3$ and $\tilde{m} = 2$, a ${}_{\tilde{n}}\Diamond_{n}$-circuit is

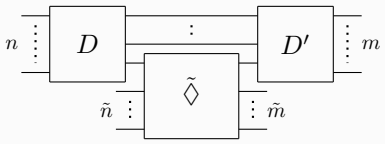


where $*$ indicates any angle in $[0, 2\pi)$ and $\bullet$ any angle in $(0, \frac{\pi}{2}]$.

Remark III.27. As $\tilde{m} = n$, ${}_{\tilde{n}}\Diamond_{n}$-circuits have $\tilde{n} \times n$ exactly beam splitters shaped like in Table III.1. Furthermore, their angles are necessarily nonzero, as one zero would imply the rest of the right-diagonal to be zero with Definition III.17, contradicting the Property 4. That particular shape and those nonzero properties will be useful in the proofs of Section III.3.4.

A key property that will be used in the proof of Proposition III.102 is that we can always extract a ${}_{\tilde{n}}\Diamond_{n}$-circuit from a ${}_{\tilde{n}}\triangle_{\tilde{m}}$-circuit as follows.

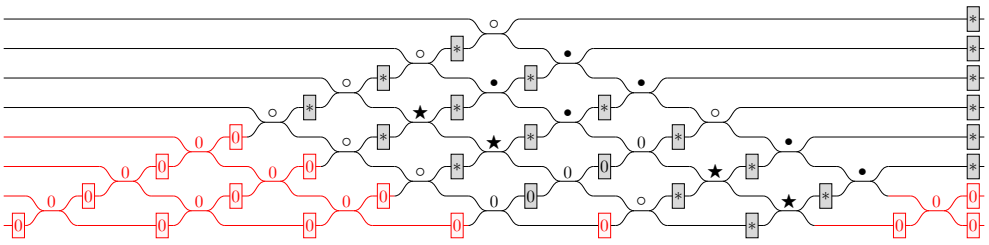
Lemma III.28 (${}_{\tilde{n}}\Diamond_{n}$-circuit extraction)**.** *Any ${}_{\tilde{n}}\triangle_{\tilde{m}}$-circuit can be rewritten to:*



where $D : n \to n$ and $D' : m \to m$ are two **LO$_{\mathsf{PP}}$***-circuits and $\tilde{\Diamond}$ is a ${}_{\tilde{n}}\Diamond_{\tilde{m}}$-circuit.*

Proof. Let us consider a generic subtriangle ${}_{\tilde{n}}\triangle_{\tilde{m}} : n+\tilde{n} \to m+\tilde{m}$. With the 4^{th} property of Definition III.22, we know there exists one beam splitter with a nonzero angle on the $(n+\tilde{n}-1, n+\tilde{n})$ wires. We note $\bigstar$ the one on the most right. As this beam splitter cannot be connected to the $\tilde{n}$ last input wires, there is necessarily one beam splitter on the $(n+\tilde{n}-1, n+\tilde{n})$ on its top left, and we also denote the most right of them with $\bigstar$. Up to the $(\tilde{n}-1, \tilde{n})$, we can show there is at least one nonzero beam splitter on the top left of the previous considered nonzero beam splitters, and we note with $\bigstar$ the most right of them, meaning all the other beam splitters on its right and before the have zero angles 0. We denote $\circ$ (resp. $\bullet$) an arbitrary (resp. arbitrary but nonzero) beam splitter angle. With the property of Definition III.17, if there is a nonzero angle $\bigstar$ (resp. zero angle 0), then we necessarily have nonzero (resp. zero) angles for all the beam splitters above (resp. under) in the right diagonal. The angles which are necessarily zero for the property 3 and 4 of Definition III.22 are in red.

Therefore, a ${}_{\tilde{n}}\triangle_{\tilde{m}}$-circuit is always of the following form:



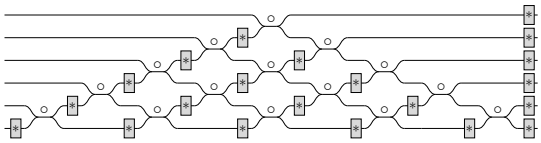
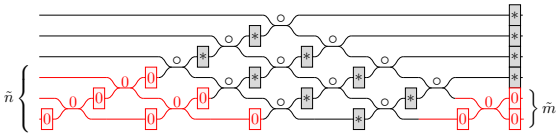
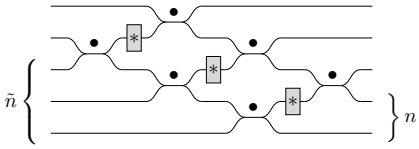
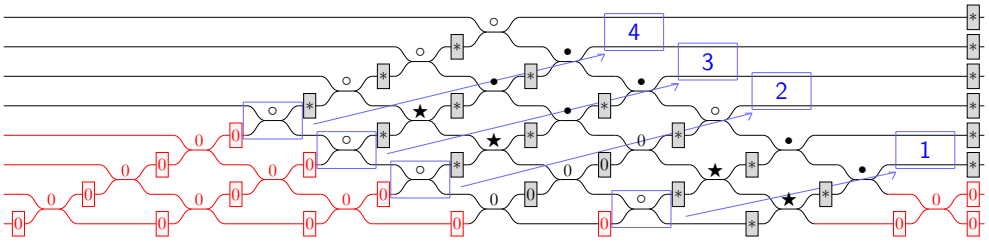
Shape	Properties
	$\triangle$-circuits (Definition III.17) Uniquely determined by $[\![\cdot]\!]_1$ (Proposition III.20).
	${}_{\tilde{n}}\triangle_{\tilde{m}}$-circuits (Definition III.22) Uniquely determined by the submatrix $[\![\cdot]\!]_1(1:m,1:n)$ (Proposition III.24).
	${}_{\tilde{n}}\tilde{\Diamond}_{n}$-circuits (Definition III.25) They have exactly $\tilde{n}\times\tilde{m}$ nonzero beam splitters, with no identity wire.

Table III.1 – Shapes and properties of classes of triangle **LO$_{\text{PP}}$**-circuits: $n+\tilde{n}\to m+\tilde{m}$. $(*,\circ)$ are angles in $[0,2\pi)\times[0,\frac{\pi}{2}]$ that satisfy the properties of Defitions III.17 and III.22. We emphasize the nonzero angles of $\tilde{\Diamond}$ by noting $\bullet$ an arbitrary angle in $(0,\frac{\pi}{2}]$. The angles which are necessarily zero for the property 3 and 4 of Definition III.22 are in red. We have $n=\tilde{n}=3, m=4$ and $\tilde{m}=2$ for the first two figures, and $\tilde{m}=n=2$ and $\tilde{n}=m=3$ for the third.

With the completeness of **LO$_{\text{PP}}$** [3], we can change the shape of the circuit, so the apex of the triangle is now at the bottom. More precisely, we can use a symmetric version of (R10) to move all the beam splitters under the diagonal of $\bigstar$ as follows:



We conclude by noticing we have the nonzero beam splitters as expected.

□

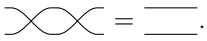
III.1.4 Rewriting system

We introduce a rewriting system that we prove to be terminating and confluent, along with describing its normal forms.

3. Note this proof assumes the completeness of **LO$_{\text{PP}}$**, which is proved later in Section III.1.5. This does not compromise the validity of the overall argument, as the lemma is only used in the proof of Proposition III.102.

A. Rewriting system: PPRS

Definition III.29 (**PPRS**: Rewriting system of $\mathbf{LO}_{\mathbf{PP}}^{\mathrm{PRO}}$)**.** We define **PPRS**, a rewriting system on the PRO of $\mathbf{LO}_{\mathbf{PP}}$-circuits according to the rules of Figure III.2.

Remark III.30 (The reason for defining a PRO). **PPRS** is defined on a PRO to prevent equivalences between diagrams, such as  = . Indeed, such equivalence could prevent us from having a system that terminates.

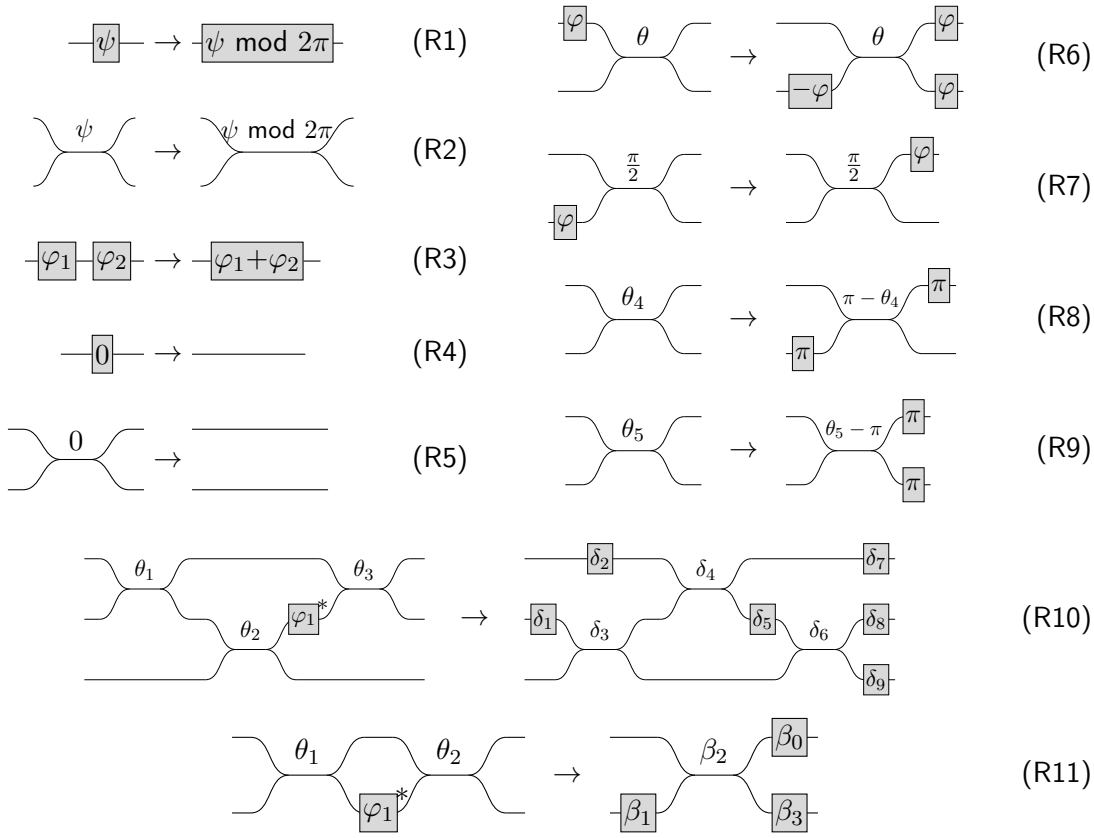
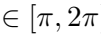


Figure III.2 – **PPRS**: Rewriting system[4] of the $\mathbf{LO}_{\mathbf{PP}}^{\mathrm{PRO}}$-calculus $\psi \in \mathbb{R} \setminus [0, 2\pi)$, $\varphi, \varphi_1, \varphi_2 \in (0, 2\pi)$, $\theta_4 \in (\frac{\pi}{2}, \pi)$, $\theta_5 \in [\pi, 2\pi)$, $\theta, \theta_1, \theta_2, \theta_3 \in (0, \pi)$. Note that  denotes either or .

This system is both strongly normalizing and confluent, as proven in the two following sections (Section III.1.4.B. and III.1.4.C.). Remarkably in the last section (Section III.1.4.D.) we also prove the normal forms of our systems are exactly the $\triangle$-circuits (Definition III.17).

4. Compared to [Clé+22], there are slight differences in the rules (R6), (R7), (R8) and (R11). This is because in the normal forms of $\mathbf{LO}_{\mathbf{PP}}$ and $\mathbf{LO}_{fi}$ of this thesis, phase shifters before the beam splitters are on the bottom wire instead of the top one.

Lemma III.31 (Normal forms of **PPRS**). *Any* $\mathbf{LO}_{\mathbf{PP}}^{PRO}$*-circuit, with the rules of* **PPRS**, *converges to a unique* $\triangle^*$*-circuit.*

Remark III.32. Note that with (swap), all the swaps can be rewritten to circuits with only beam splitters and phase shifters. Therefore, all results in $\mathbf{LO}_{\mathbf{PP}}^{\mathrm{PRO}}$can be used in $\mathbf{LO}_{\mathbf{PP}}$.

This lemma awd remark will be particularly important for the proof of completeness (Section III.1.5).

Remark III.33.

B. Strongly normalizing

Proposition III.34 (**PPRS** is strongly normalizing). *The rewriting system* **PPRS** *is strongly normalizing, i.e. every sequence of rewriting rules eventually terminates.*

Proof. We define a ranking function $(x_1, x_2, x_3) \in \mathbb{N}^3$, that we show to be strictly decreasing with respect to lexicographic order. Let C be the optical circuit. We define the variables as follows.

- $x_1 = \sum_{i=0}^{n-2} (n-i)c(i)$ where $c(i)$ is the number of beam splitters in C between the position i and $i+1$.
- x_2 is the sum of the number of beam splitters with angles not in $[0, \frac{\pi}{2}]$, with those not in $[0, \pi)$ and those not in $[0, 2\pi)$. For instance, a beam splitter with an angle of 3π will be counted three times, and those with $\frac{3}{2}\pi$ will be counted twice.
- To define x_3, let us define the depth of a phase shifter p of C, denoted $d(p)$, as the maximal number of beam splitters that a photon starting from p and going to the right would be able to traverse before reaching an output port, if it were allowed to choose each time whether to be reflected or transmitted. Then

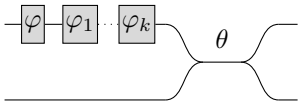
$$x_3 \coloneqq \sum_{\text{p phase shifter in } C} (v(p) + w(p))9^{d(p)}$$

where, given a phase shifter $p = \boxed{\varphi}$ of C,

$$v(p) \coloneqq \begin{cases} 1 \text{ if } \varphi \notin [0, 2\pi) \\ 0 \text{ otherwise.} \end{cases}$$

and

$$w(p) \coloneqq \begin{cases} 3 \text{ if } p \text{ belongs to a pattern of the form} \\ 1 \text{ otherwise.} \end{cases}$$



Note that x_3 strictly decreases when a phase on the top left of beam splitter is removed, even though we add one phase one the bottom left and two phases with lower depth. For such scenario, in the worst case, we have $3 \times 9^{d(p)} = 27 \times 9^{d(p)}$ becoming $2 \times 9^{d(p)} + 2 \times 4 \times 9^{d(p)-1} = 26 \times 9^{d(p)}$. Thus, Rule R6 is indeed strictly decreasing x_3. Note that removing phases, strictly decreasing the depth of one phase, or changing the angle of a phase to make it into $[0, 2\pi)$ also strictly decrease x_3. Thus, Rules R1, R3, R4 and R7 strictly decrease x_3. All the previous rules do not involve beam splitters, therefore they decrease x_3 without changing x_1 and x_2. For the other rules, we can notice that Rules R10 and R11 strictly decrease x_1, and Rules R2, R5, R8 and R9 strictly decrease x_2 without changing x_1. □

C. Confluence

For proving the confluence, as **PPRS**is strongly normalizing and as a consequence of Newman's lemma (Lemma I.87), it is sufficient to show the local confluence.

Proposition III.35 (**PPRS** is local confluent)**.** *The rewriting system* **PPRS** *is local confluent.*

Proof. To prove local confluence, it suffices to show that we can always recover the same term after applying any pair of rules. First, let us note that any critical pair, i.e., the non-trivial pairs that we need to consider, necessarily involves at most three wires. Indeed, if two rules apply on disjoint parts of the circuit, they are spatially independent and can therefore be closed independently.

All the critical pairs involving at most three wires are listed in Figure III.3. For each of them, we can show that we recover the same term. □

Remark III.36. An alternative proof is to show that any circuit on $n \in \{1, 2, 3\}$ wires converges to a $\triangle^*$-circuit of size n. This can be done with some elements of the proof of Proposition III.37. Since the $\triangle^*$-circuits are unique, this implies that any sequence of rule applications on circuits of size $n \in \{1, 2, 3\}$ can be resolved to the same term, thereby also proving local confluence.

D. Normal forms are $\triangle^*$-circuits

As **PPRS** is strongly normalizing and confluent, it is converging to unique normal forms. To prove Lemma III.31, it remains to prove that indeed, the irreducible forms of **PPRS** are indeed $\triangle^*$-circuits. This is the following proposition.

Proposition III.37 (Irreducible forms of **PPRS**)**.** *All irreducible forms of* **PPRS** *are $\triangle^*$-circuits.*

Proof. We do a proof by induction. The key idea is to notice a $\triangle^*$-circuit of size n has a nice inductive shape, where there is an upper top-left diagonal of beam splitters (and phase shifters), and then a $\triangle^*$-circuit of size $n - 1$ on the bottom right of this diagonal.

For $n \leq 3$, we can directly show the irreducible forms are necessarily $\triangle^*$-circuits. For $n = 1$, it is either of phase with $\varphi \in (0, 2\pi)$ or an identity wire. For $n = 2$, there is at most one beam splitter due to Rule R11, and no top left phase due to Rule R6. Similarly, for $n = 3$, Rules R10 and R11 assure there is at most three beam splitters in the shape of a triangle with the apex at the top. The other constrains follow from the other rules.

Let consider a circuit of size $n > 3$, by considering that any irreducible circuit of size $k < n$ is a $\triangle^*$-circuit. First, let us notice that necessarily, there is at most one beam splitter on the first two wires. Indeed, otherwise, there is necessarily two consecutive beam splitters, thus Rule R11 applies, or there is necessarily a pattern that corresponds to the LHS of Rule R10. If there is no beam splitter connected to the first wire, then we only have one identity tensored with a circuit C of size $n - 1$. By induction hypothesis, C is a $\triangle^*$-circuit. We conclude this case by noticing that $\text{——} \otimes C$ is indeed a $\triangle^*$-circuit. If there is a beam splitter, then its angle is necessarily not zero, otherwise Rule R5 applies. Then we look at everything that is on the bottom left. For the same reasons as before, there is at most one beam splitter on the wire $(2, 3)$ on the bottom left of the apex beam splitter. Otherwise, we can show that Rule R11 or R10 can be applied. With the same reasoning, that everything on the bottom left of the apex beam splitter is a diagonal that satisfies all the conditions of $\triangle^*$-circuits, which is concluding the proof. □

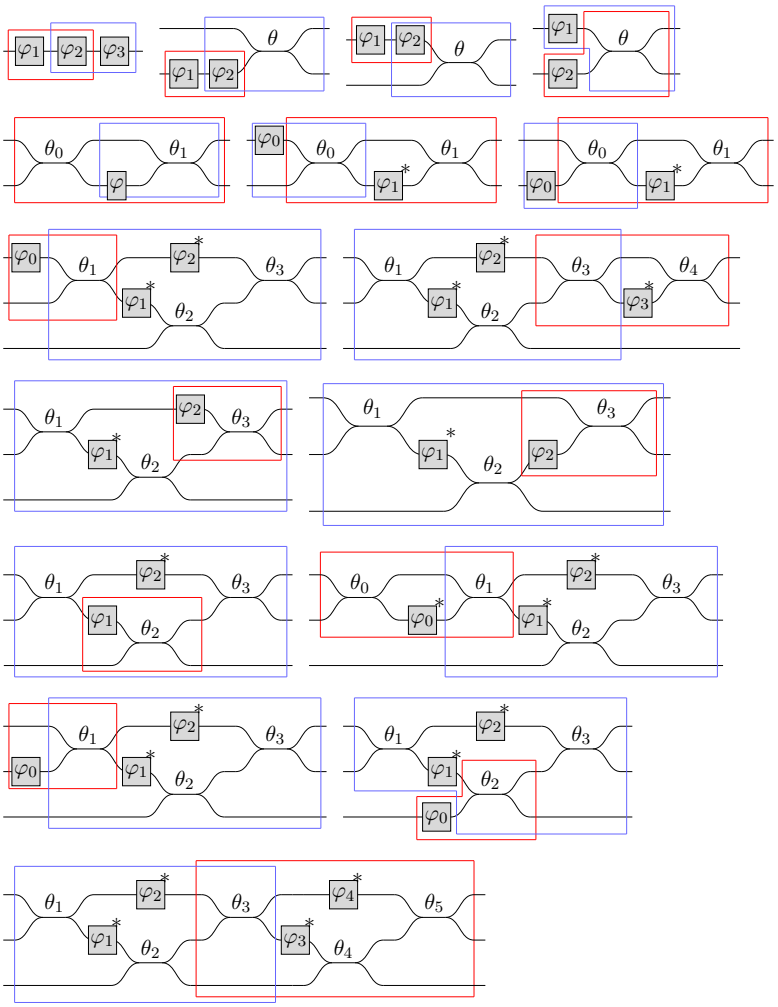


Figure III.3 – Non-trivial critical pairs. All angles are in $(0, 2\pi)$ except when we consider Rules (R1), (R2), (R4) and (R5). Note that additional constraints on the angles may occur for some rewrite rules to apply.

III.1.5 Completeness of LO$_{PP}$

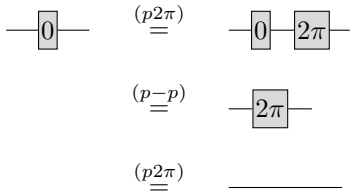
The goal of this section is to prove Theorem III.9, i.e. that **LO$_{PP}$** is complete. Following Lemma III.31 and Remark III.32, we know that any **LO$_{PP}$**-circuit converges to a unique $\triangle^*$-circuit. Therefore, it is sufficient to show that we can derive all equations of the rewriting systems (Figure III.2) with the axioms of **LO$_{PP}$**(Figure III.1). This is the purpose of the following propositions.

Proposition III.38 (Identities of 0-angle)**.**

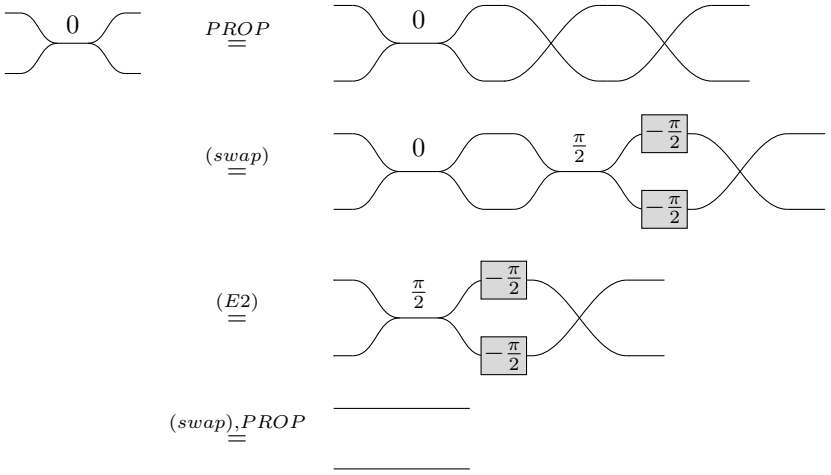
$$\textbf{LO}_{PP} \vdash \quad -\boxed{0}- \quad = \quad ---- \qquad \text{(p0)}$$

$$\textbf{LO}_{PP} \vdash \quad \overset{\diagdown\,0\,\diagup}{\diagup\quad\diagdown} \quad = \quad \begin{matrix}---\\---\end{matrix} \qquad \text{(b0)}$$

Proof. To prove (p0), we have:

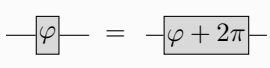


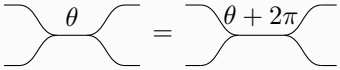
To prove (b0), we have:



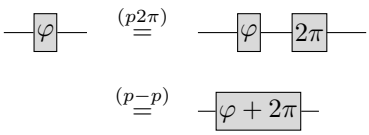
□

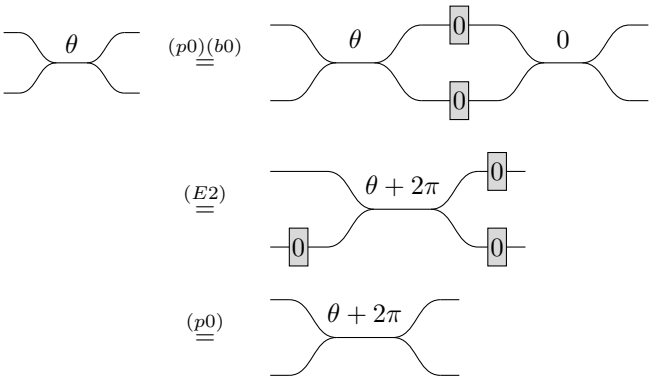
Proposition III.39 (2π-periodicity)**.** *The axioms of the* **LO$_{\mathsf{PP}}$***-calculus imply that the parameters are 2π-periodic, i.e. for any $\theta, \varphi \in \mathbb{R}$:*

$\mathbf{LO_{PP}} \vdash$ 

$\mathbf{LO_{PP}} \vdash$ 

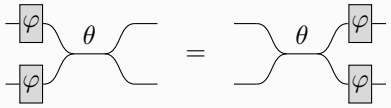
Proof.





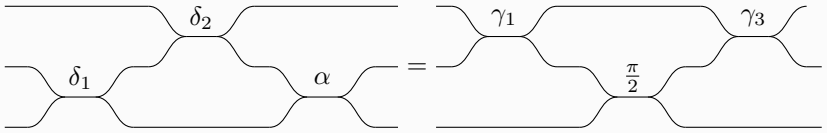
□

Proposition III.40 (Useful equations)**.** *We can derive the following equation with the axioms of* **LO$_{PP}$**.



(III.4)

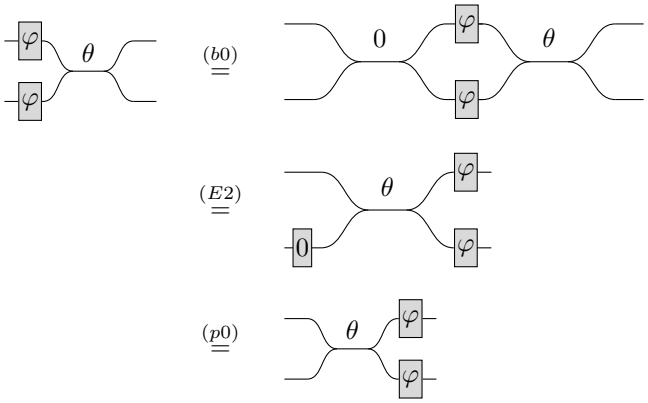
If $\sin(\delta_1) \neq 0$*, then the following equation is sound:*



(III.5)

with $\alpha = \tan^{-1}(\frac{c_{\delta_1}}{s_{\delta_1} c_{\delta_2}})$.

Proof. To prove (III.4), we have:



For (III.5), by looking at the matrices of both sides of the equation we have:

$$[\![LHS]\!]_1 = \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & -s_\alpha s_{\delta_1} c_{\delta_2} + c_\alpha c_{\delta_1} \end{pmatrix} = \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & 0 \end{pmatrix} = [\![RHS]\!]_1$$

Therefore, for $\delta_1 \neq 0$, we have:

$$-s_\alpha s_{\delta_1} c_{\delta_2} + c_\alpha c_{\delta_1} = 0 \Leftrightarrow \tan(\alpha) = \frac{c_{\delta_1}}{s_{\delta_1} c_{\delta_2}} \Leftrightarrow \alpha = \tan^{-1}(\frac{c_{\delta_1}}{s_{\delta_1} c_{\delta_2}}).$$

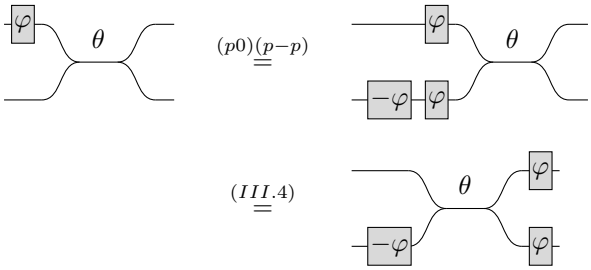
□

Proposition III.41 (Useful equations of **LO$_{PP}$**)**.** *The Rules (R6), (R7), (R8), (R9), and (R11) can be derived from the axioms of* **LO$_{PP}$**.

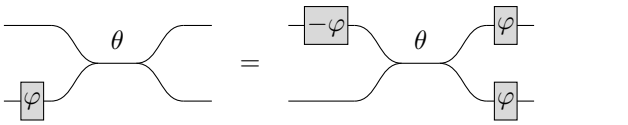
The soundness of **PPRS** and the completeness of **LO$_{PP}$** are direct consequences of the following lemma.

Lemma III.42 (Derivation of **PPRS**)**.** *All the rules of* **PPRS** *can be derived from the axioms of* **LO$_{PP}$**.

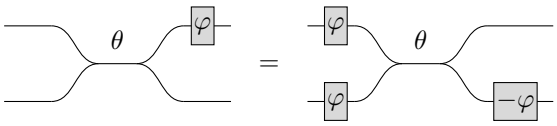
Proof. Following from Proposition III.38 and III.39, and by noticing that Rule (R3) is directly (p-p), we have derived the first five rules of Figure III.2. We can now prove all other rules.



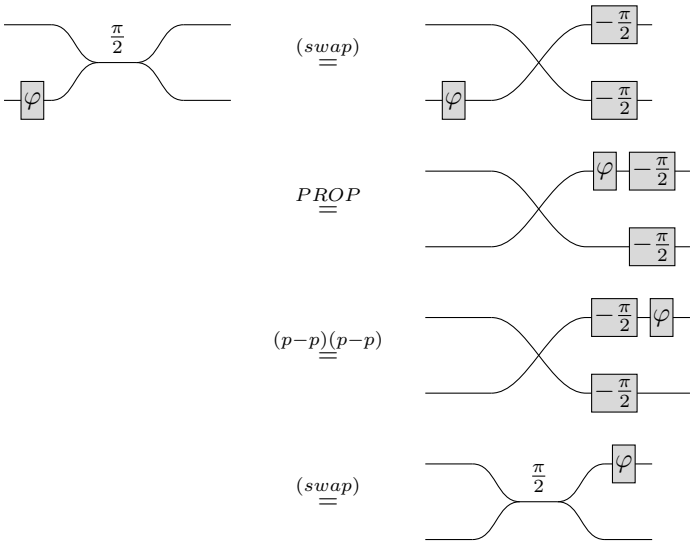
Similarly, we can prove the following two equations.



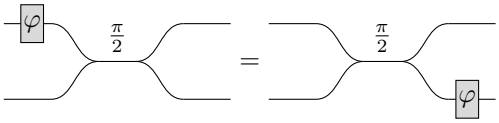
(III.6)



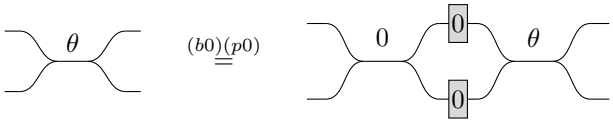
(III.7)

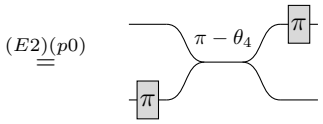


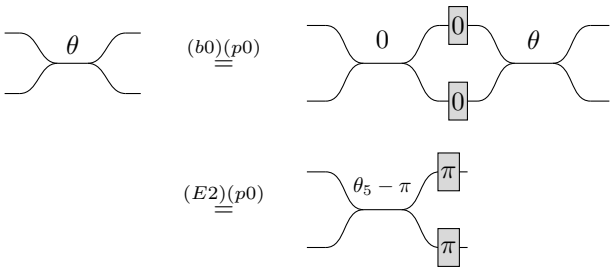
Similarly, we can prove the following equation.



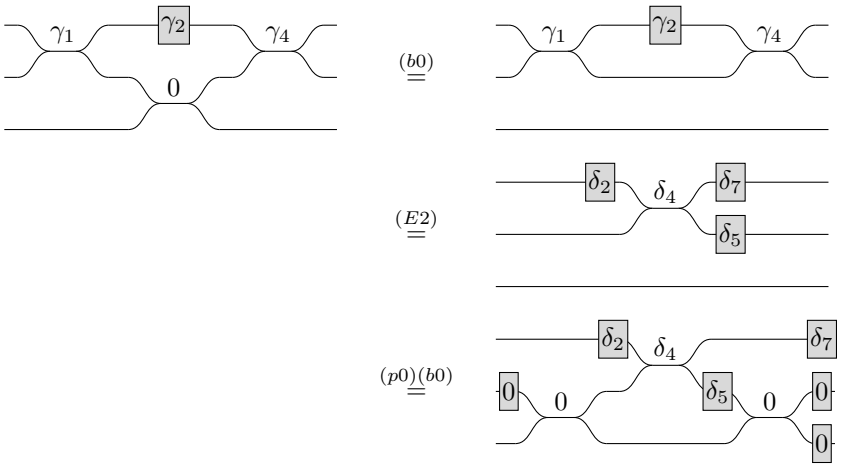
(III.8)



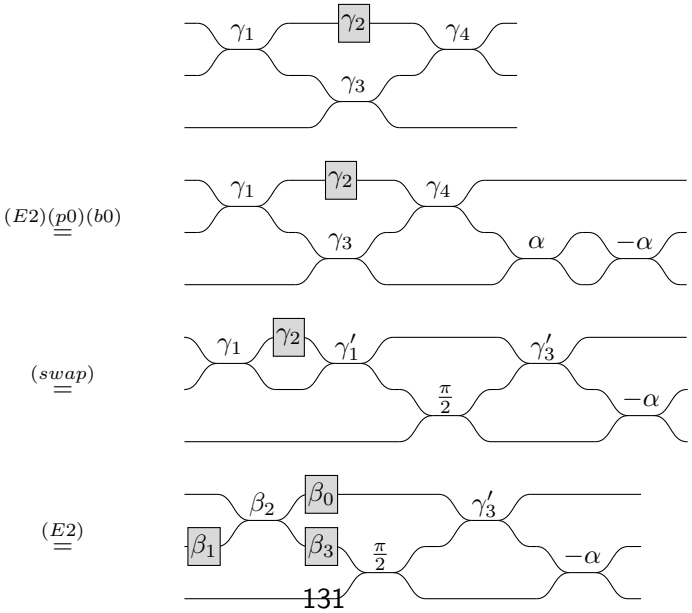


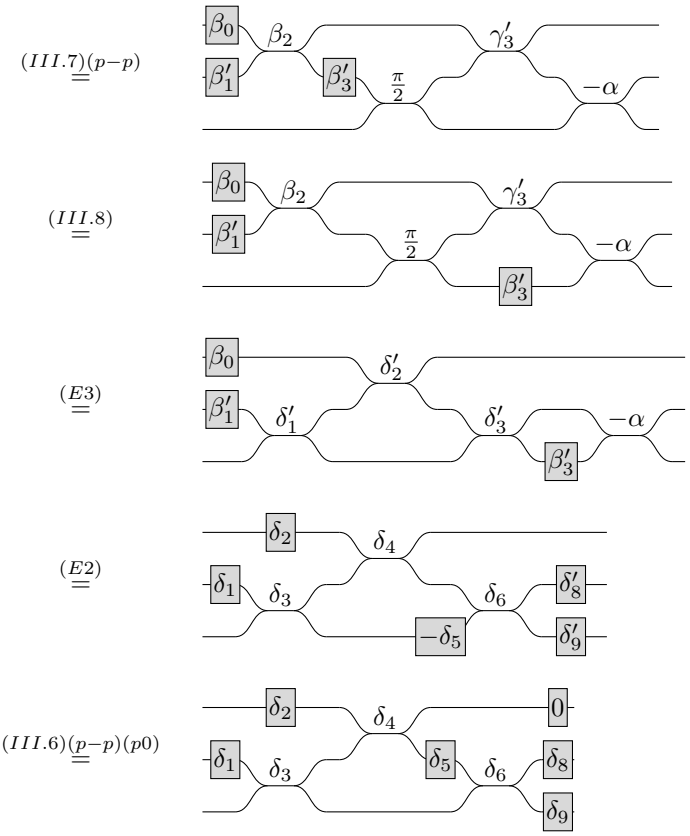


To derive (R10) if $\gamma_3 = 0$, we have:



To derive (R10) if $\gamma_3 \neq 0$, we have:





□

Following Lemma III.42 and III.31, any **LO$_{PP}$**-circuit can be rewritten to a unique normal form, concluding the proof of Theorem III.82 and proving the completeness of the **LO$_{PP}$**-calculus.

III.2 LO$_v$: Linear Optical circuits with vacuum sources and postselection

In this section, we present the **LO$_v$**-calculus (Linear Optical circuits with vacuum sources and postselection). We begin by introducing its syntax and semantics (Section III.2.1). Next, we provide a complete equational theory (Section III.1.2). We then make explicit a procedure to convert polarization modes into purely spatial modes (Section III.2.3), enabling us to use properties of **LO$_{PP}$**-circuits and leading to the proof of the completeness of the **LO$_v$**-calculus (Section III.2.4). Finally, we provide compact and universal new forms for **LO$_v$**-circuits (Section III.2.5).

Contributions

The **LO$_v$**-calculus and the proof of completeness originate from [Clé+22], published in MFCS'22, and coauthored with Alexandre Clément, Shane Mansfield, Simon Perdrix, and Benoît Valiron.

This section presents simplified personal proofs and a compact formulation for **LO$_v$**-circuits (Section III.2.5), which are the results of my own contribution.

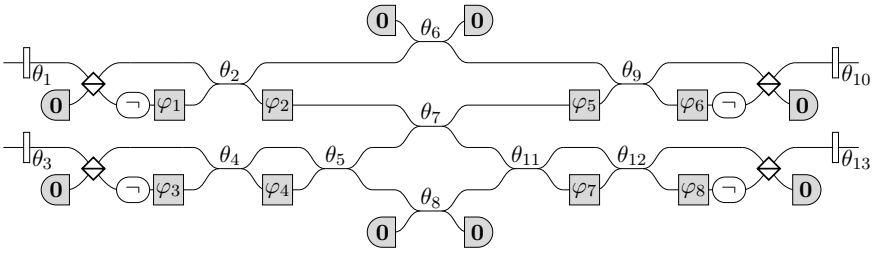
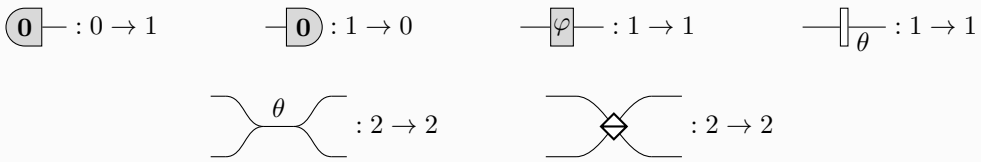


Figure III.4 – **LO$_v$**-circuit implementing a variational quantum eigensolver [Per+14], an algorithm with applications including calculation of ground-state energies in quantum chemistry.

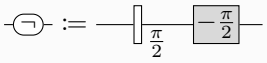
III.2.1 Syntax and semantics

In order to formalize linear optical circuits with diagrams, we use the formalism of PROPs (Definition I.74).

Definition III.43. **LO$_v$** is the PROP of **LO$_v$**-circuits generated by



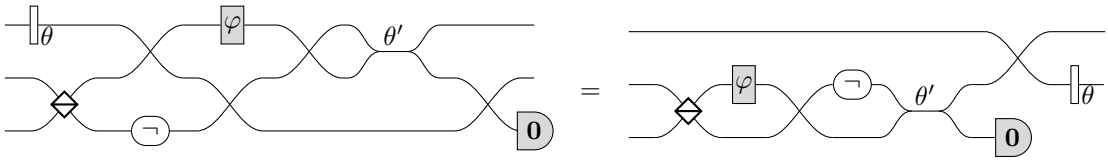
where $\theta, \varphi \in \mathbb{R}$. When the parameters θ and φ are omitted we take them to be equal to $\pi/4$. We use the following shortcut notation.



The tensor of the monoidal structure is denoted with $\oplus$, and the identity, swap and empty circuit (unit of $\oplus$) are denoted as follows: $: 1 \to 1$, $: 2 \to 2$, $: 0 \to 0$.

Example III.44. An example of a linear optical quantum circuit using all components presented in Definition III.43 is shown in Figure III.4.

Remark III.45. The axioms of PROPs guarantee that linear optical quantum circuits are defined up to deformations. The two following circuits are therefore the same.



We characterize photons by their spatial and polarization modes. Spatial modes refer to position, and polarization can be vertical (V) or horizontal (H). Note that the quantum formalism of linear optics (see Section I.3) admits superpositions of both spatial and polarization modes.

For any $n \in \mathbb{N}$, let $M_n = \{V, H\} \times [n]$, where[5] $[n] = \{0, \dots n-1\}$, be the set of states (spatial and polarization modes). The elements of M_n are denoted c_p with $c \in \{V, H\}$ and $p \in [n]$. The state space of a single photon is $\mathbb{C}^{M_n} = span(|V_i\rangle, |H_i\rangle \mid i \in [n])$. Notice that $\mathbb{C}^{M_0} = \mathbb{C}^{\emptyset} = \{0\}$ is the Hilbert space of dimension 0.

5. To simplify the formulae of the semantics, we choose to index the wires from 0 rather than from 1 as in Chapter II.

Example III.46. On 2 spatial modes (i.e. 2 wires), there are four possible basis states: $|V_0\rangle, |V_1\rangle, |H_0\rangle, |H_1\rangle$. Indeed, a photon can be on one of the two wires, while in the vertical or horizontal polarization. The state space is then a 4-dimensional Hilbert space.

The semantics of a **LO$_\mathsf{v}$**-circuit is defined as follows.

Definition III.47 (Single-photon semantics of **LO$_\mathsf{v}$**)**.** For any **LO$_\mathsf{v}$**-circuit $D : n \to m$, let $[\![D]\!]_\mathsf{v} : \mathbb{C}^{M_n} \to \mathbb{C}^{M_m}$ be the linear map inductively defined by $[\![D_2 \circ D_1]\!]_\mathsf{v} = [\![D_2]\!]_\mathsf{v} \circ [\![D_1]\!]_\mathsf{v}$, $[\![D_1 \oplus D_2]\!]_\mathsf{v} = [\![D_1]\!]_\mathsf{v} \oplus [\![D_2]\!]_\mathsf{v}$, with:

$$[\![\boxed{}]\!]_\mathsf{v} = 0 \qquad [\![\text{———}]\!]_\mathsf{v} = |c_0\rangle \mapsto |c_0\rangle \qquad [\![\text{⨯}]\!]_\mathsf{v} = |c_p\rangle \mapsto |c_{1-p}\rangle$$

$$[\![\boxed{0}\text{—}]\!]_\mathsf{v} = 0 \qquad [\![\text{—}\boxed{0}]\!]_\mathsf{v} = 0 \qquad [\![\text{—}\boxed{\varphi}\text{—}]\!]_\mathsf{v} = |c_0\rangle \mapsto e^{i\varphi} |c_0\rangle$$

$$[\![\overset{\theta}{\text{⨯}}]\!]_\mathsf{v} = |c_p\rangle \mapsto \cos(\theta) |c_p\rangle + i \sin(\theta) |c_{1-p}\rangle$$

$$[\![\text{⨯}]\!]_\mathsf{v} = \begin{cases} |V_p\rangle \mapsto |V_p\rangle \\ |H_p\rangle \mapsto |H_{1-p}\rangle \end{cases}$$

$$[\![\text{—}\overset{\theta}{\boxed{}}\text{—}]\!]_\mathsf{v} = \begin{cases} |V_0\rangle \mapsto \cos(\theta) |V_0\rangle + i \sin(\theta) |H_0\rangle \\ |H_0\rangle \mapsto \cos(\theta) |H_0\rangle + i \sin(\theta) |V_0\rangle \end{cases}$$

where for all $f \in \mathbb{C}^{M_n} \to \mathbb{C}^{M_m}$ and $g \in \mathbb{C}^{M_{n'}} \to \mathbb{C}^{M_{m'}}$,

$$(f \oplus g)(|c_k\rangle) = \begin{cases} f(|c_k\rangle) & \text{if } k < n \\ S_{m,m'}(g(|c_{k-n}\rangle)) & \text{if } k \geq n \end{cases}$$

where $S_{m,m'} : \mathbb{C}^{M_{m'}} \to \mathbb{C}^{M_{m+m'}} = |c_k\rangle \mapsto |c_{k+m}\rangle$ a shift of the positions by m.

Example III.48. The negation inverts polarization: $[\![\text{-⊃-}]\!]_\mathsf{v} : |V_0\rangle \mapsto |H_0\rangle$ and $|H_0\rangle \mapsto |V_0\rangle$.

Remark III.49. The semantics of the circuits is sound with respect to the axioms of PROPs. In other words two circuits that are equal up to deformation have the same semantics.

Remark III.50. All the generators of the **LO$_\mathsf{v}$**-circuits are photon preserving, even the vacuum state sources (0—) and detectors (—0). Indeed, the vacuum state source produces no photons, whereas the semantics of the detector corresponds to a postselection on the case where no photons are detected.

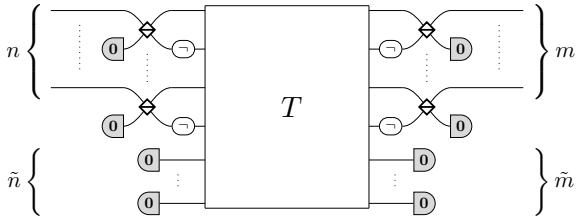
LO$_\mathsf{v}$-circuits are more expressive than **LO$_{\mathsf{PP}}$**-circuits (Section III.1), they not only act on the polarization but the use of detectors and sources allow the representation of non-unitary evolutions: For any **LO$_\mathsf{v}$**-circuit $D : n \to m$, $[\![D]\!]_\mathsf{v}$ is sub-unitary[6]. **LO$_\mathsf{v}$**-circuits are actually universal for sub-unitary transformations:

Theorem III.51 (Universality of **LO$_\mathsf{v}$**)**.** *For every sub-unitary map $V : \mathbb{C}^{M_n} \to \mathbb{C}^{M_m}$, there exists a* **LO$_\mathsf{v}$***-circuit $D : n \to m$ such that $[\![D]\!]_\mathsf{v} = V$.*

Proof. Let $V : \mathbb{C}^{M_n} \to \mathbb{C}^{M_m}$ be a sub-unitary of a unitary map $U : (n + \tilde{n}) \times (m + \tilde{m})$. Let T a

6. U is sub-unitary (see for instance [Sel04]) if and only if $U^\dagger U \sqsubseteq I$, where $\sqsubseteq$ is the Löwner partial order, i.e. $I - U^\dagger U$ is a positive semi-definite.

LO$_{PP}$-circuit such that $[\![T]\!]_1 = U$. The following circuit D



is such that $[\![D]\!]_{\mathsf{v}} = V$. □

III.2.2 Complete equational theory of LO$_{\mathsf{v}}$

Two distinct **LO$_{\mathsf{v}}$**-circuits may represent the same quantum evolution: for instance, composing two negations is equivalent to the identity. In order to characterize equivalences of **LO$_{\mathsf{v}}$**-circuits, we introduce a set of equations, shown in figure III.5. They capture basic properties of **LO$_{\mathsf{v}}$**-circuits, such as vacuum detectors and sources essentially absorbing the other generators (Equations (A.9) to (A.12)), or commutation properties (Equations (A.15), (A.16)). Note that there are two equations acting on 3 modes:(A.6) and (A.19). Equation (A.6) is a variant of the Yang-Baxter Equation [Jim89], whereas (A.19) is the same as (E3) in the **LO$_{PP}$**-calculus Section III.1.2.

Definition III.52 (**LO$_{\mathsf{v}}$**-calculus)**.** Two **LO$_{\mathsf{v}}$**-circuits D_1, D_2 are equivalent according to the rules of the **LO$_{\mathsf{v}}$**-*calculus*, denoted **LO$_{\mathsf{v}}$** $\vdash D_1 = D_2$, if one can transform D_1 into D_2 using the equations given in Figure III.5. More precisely, **LO$_{\mathsf{v}}$** $\vdash \cdot = \cdot$ is defined as the smallest congruence which satisfies the equations of Figure III.5 in addition to the axioms of PROP.

Proposition III.53 (Soundness)**.** *For any two* **LO$_{\mathsf{v}}$**-*circuits* D_1 *and* D_2*, if* **LO$_{\mathsf{v}}$** $\vdash D_1 = D_2$ *then* $[\![D_1]\!]_{\mathsf{v}} = [\![D_2]\!]_{\mathsf{v}}$.

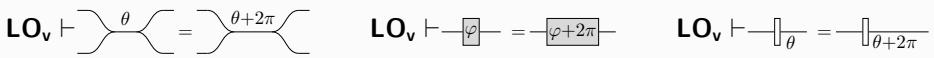
Proof. Since semantic equality is a congruence it suffices to check that for every equation of figure III.5 both sides have the same semantics. Note the soundness of (A.18) and (A.19) directly follows from Proposition III.10 and III.11. □

Let us note that, as every axiom of the **LO$_{PP}$**-calculus is present in the axioms of the **LO$_{\mathsf{v}}$**-calculus, we have the following proposition.

Proposition III.54. *For any* **LO$_{PP}$**-*circuits* D_1 *and* D_2*, if* **LO$_{PP}$** $\vdash D_1 = D_2$ *then* **LO$_{\mathsf{v}}$** $\vdash D_1 = D_2$.

Another particular useful property is the periodicity of all parameterized generators, that we prove here.

Proposition III.55 (2π-periodicity)**.** *The rules of the* **LO$_{\mathsf{v}}$**-*calculus imply that the parameters are* 2π-*periodic, i.e. for any* $\theta, \varphi \in \mathbb{R}$*:*



Proof. The proof for the beam splitter and phase shifter are in Proposition III.39. The 2π-periodicity for the wave plate follows from that for the beam splitter as follows.

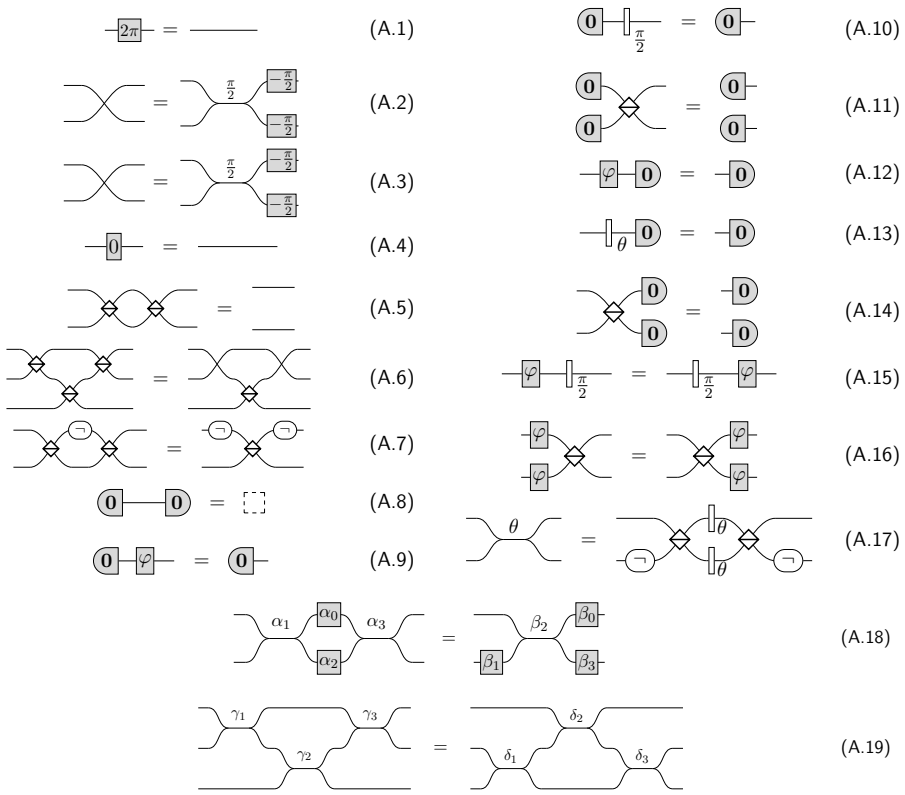
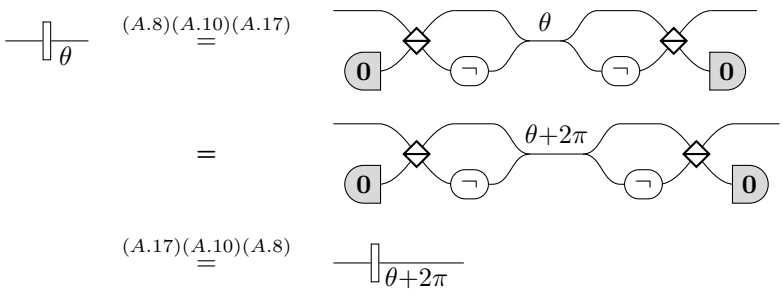


Figure III.5 – Axioms of the $\mathbf{LO_v}$-calculus. The equations are valid for arbitrary parameters $\varphi, \varphi_i, \theta, \theta_i \in \mathbb{R}$. The angles of (A.18) and (A.19) are the same than (E2) and (E3), detailed in Proposition III.10 and III.11.



□

The major contribution is to prove the equational theory of Figure III.5 is complete.

Theorem III.56 (Completeness). *For any two* $\mathbf{LO_v}$*-circuits* D_1 *and* D_2*, if* $[\![D_1]\!]_{\mathsf{v}} = [\![D_2]\!]_{\mathsf{v}}$ *then* $\mathbf{LO_v} \vdash D_1 = D_2$.

The proof is provided in Section III.2.4, where we explicit a normal form that uses $\mathbf{LO_{PP}}$-circuits, and where we do the link in Section III.2.3. The following sections use the equations of Figure III.6 that we can show to be consequences of the axioms of the $\mathbf{LO_v}$-calculus.

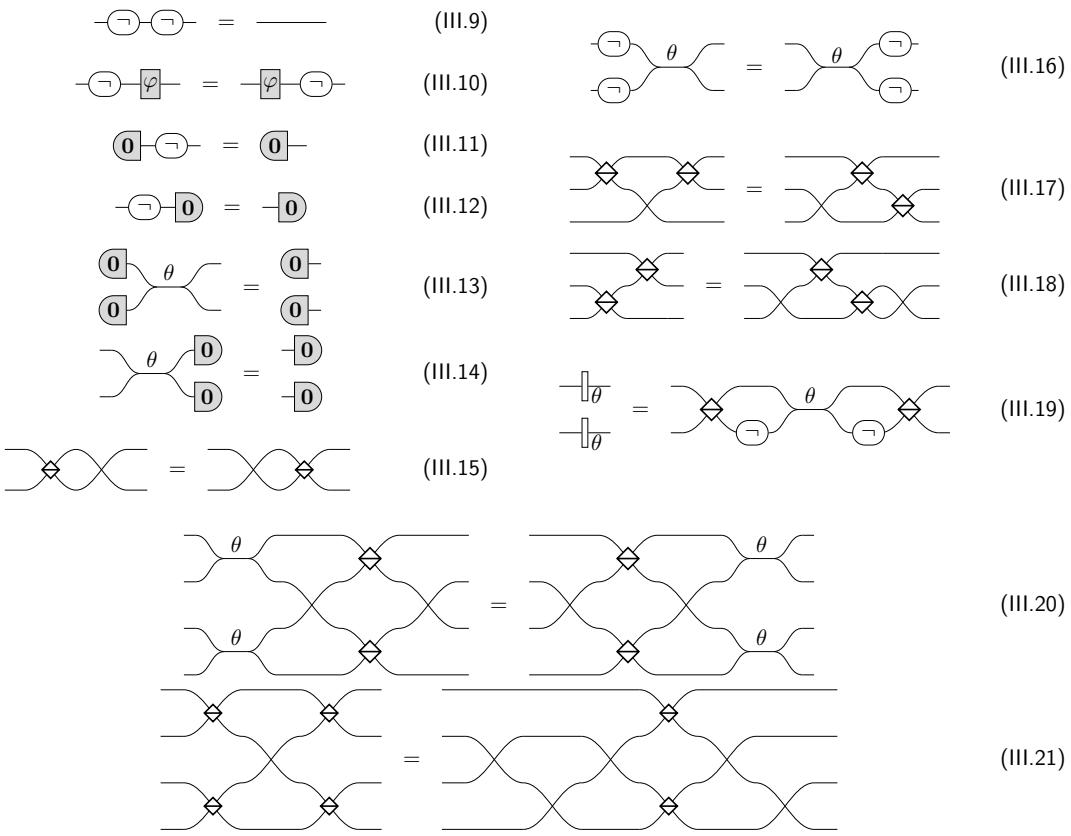
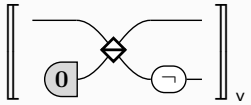


Figure III.6 – Useful consequences of the axioms of the $\mathbf{LO_v}$-calculus. Note that the derivation of these rules requires additional intermediate equations, leading to numerous diagrammatic steps for a complete proof. As they offer no further insight and are already fully detailed in both the original paper [Clé+22] and Alexandre Clément's PhD thesis [Clé23], we do not include the proofs in this thesis.

III.2.3 From polarization to spatial modes

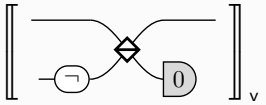
We introduce two circuits that allow to transform two polarization modes on one wire into the two spatial modes of two wires, where the polarization is arbitrary fixed to $|V\rangle$.

Definition III.57 (Splitter and merger)**.** The following circuit is splitting the two polarization modes into two spatial modes:



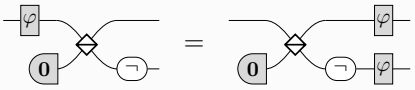
$$\begin{array}{lcccl} |V\rangle & \mapsto & |V,0\rangle & \approx & |1,0\rangle \\ |H\rangle & \mapsto & |0,V\rangle & \approx & |0,1\rangle \end{array}$$

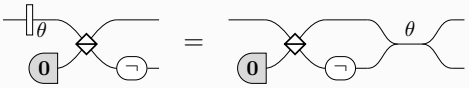
while the following is merging two spatial modes into two polarization modes on one wire:

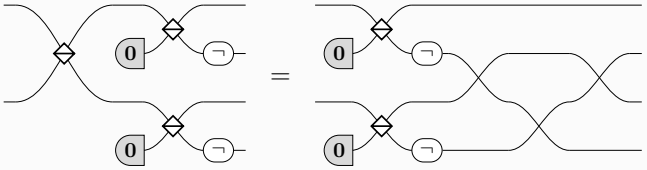


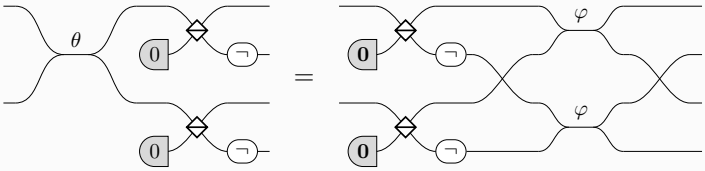
$$\begin{array}{lcccl} |1,0\rangle & \approx & |V,0\rangle & \mapsto & |V\rangle \\ |0,1\rangle & \approx & |0,V\rangle & \mapsto & |H\rangle \end{array}.$$

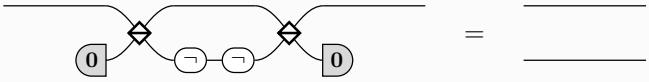
Proposition III.58 (Polarization modes into spatial modes)**.** *We can derive all the following equations with the axioms of the* $\mathbf{LO_v}$*-calculus.*


(III.22)


(III.23)

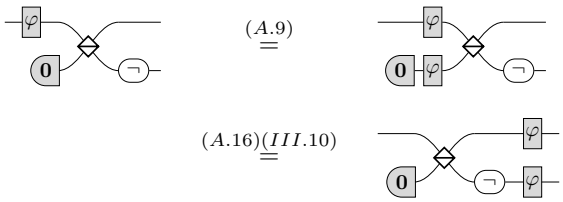

(III.24)


(III.25)

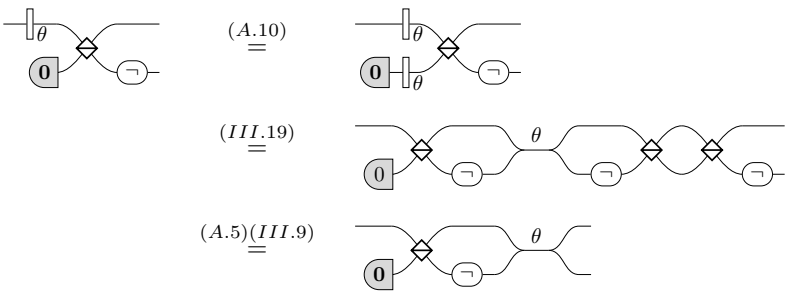

(III.26)

Proof. The equation (III.26) directly follows from (III.9) and (A.5). For the others, we derive them as follows.

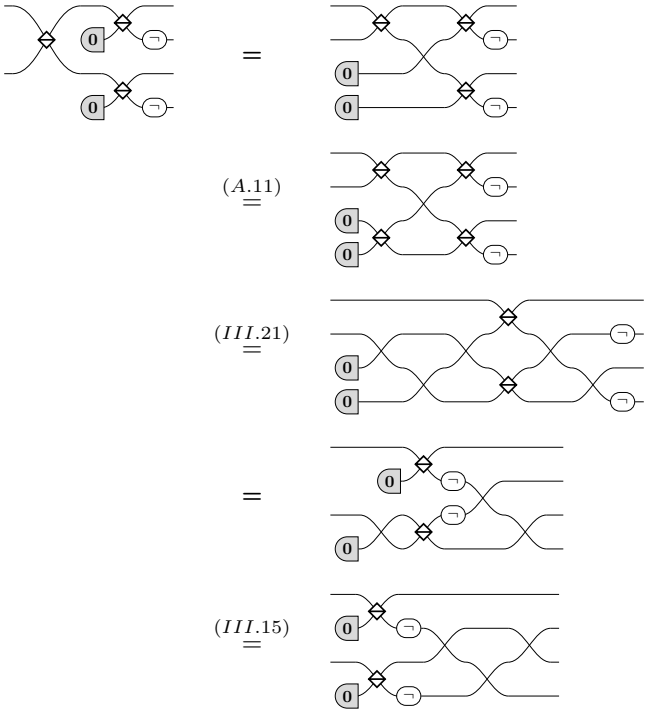
To prove (III.22), we have:



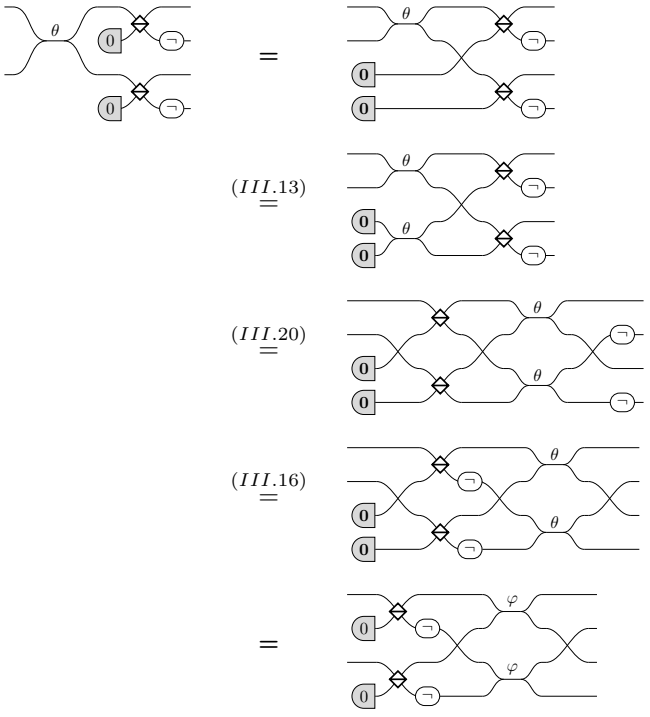
To prove (III.23), we have:



To prove (III.24), we have:



To prove (III.25), we have:



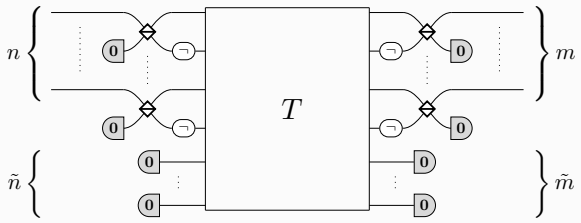
□

III.2.4 Completeness of $\mathbf{LO_v}$

We can now prove Theorem III.82. The proof is in two steps. First, we introduce normal forms for $\mathbf{LO_v}$-circuits that are unique. Then, we show that any $\mathbf{LO_v}$-circuit can be rewritten to such forms with

the rules of the **LO$_\mathsf{v}$**-calculus.

Definition III.59 (Normal form of **LO$_\mathsf{v}$**-circuits)**.** The normal forms of **LO$_\mathsf{v}$**-circuits $N : n \to m$ are of the form :



where $T : 2n + \tilde{n} \to 2n + \tilde{n}$ is a ${}_{\tilde{n}}\triangle_{\tilde{m}}$-circuit (Definition III.22).

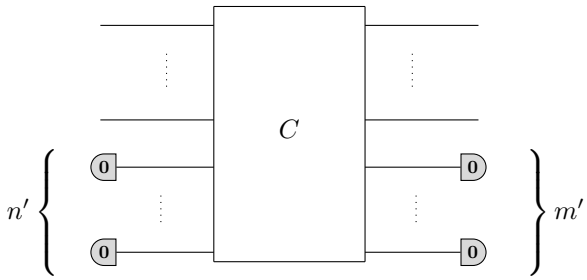
We now prove the two following lemmas.

Lemma III.60 (Uniqueness of the normal forms of **LO$_\mathsf{v}$**)**.** *Up to the rule (A.8), the normal forms of the* **LO$_\mathsf{v}$***-calculus are unique.*

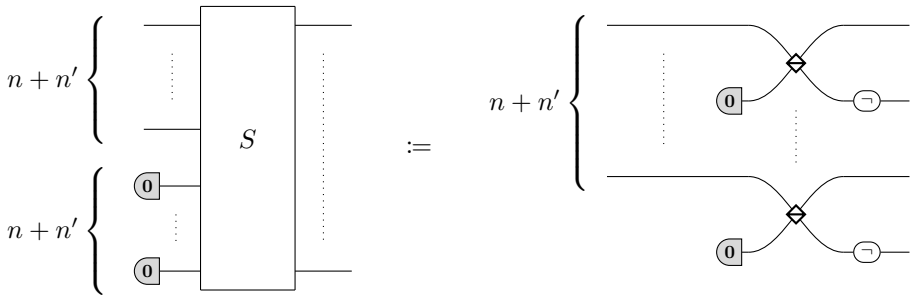
Proof. Note that due to the $\tilde{n}$ vacuum sources and $\tilde{m}$ vacuum detectors, all coefficients outside of the $2m \times 2n$ submatrix of $[\![T]\!]_1$ are zero. Therefore, $[\![N]\!]_\mathsf{v}$ is fully characterized by the $2m \times 2n$ submatrix of $[\![T]\!]_1$. The uniqueness of such normal forms is therefore a direct consequence of Proposition III.24, that states that such ${}_{\tilde{n}}\triangle_{\tilde{m}}$-circuit are indeed unique on such submatrix. Note that the rule (A.8) does not change the submatrix and can be applied many times, hence the need to define uniqueness up to this rule. □

Lemma III.61 (Universality of normal forms)**.** *Any* **LO$_\mathsf{v}$***-circuit $C : n \to m$ can be rewritten to a normal form with the rules of the* **LO$_\mathsf{v}$***-calculus.*

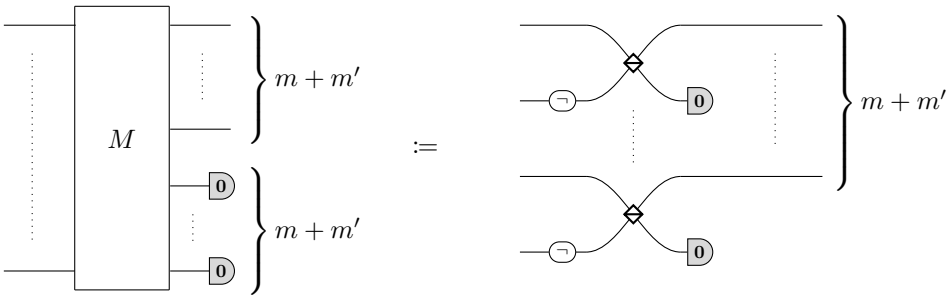
Proof. Given a generic **LO$_\mathsf{v}$**-circuit $C : n \to m$, we can bend the wires so it is rewritten as follows:



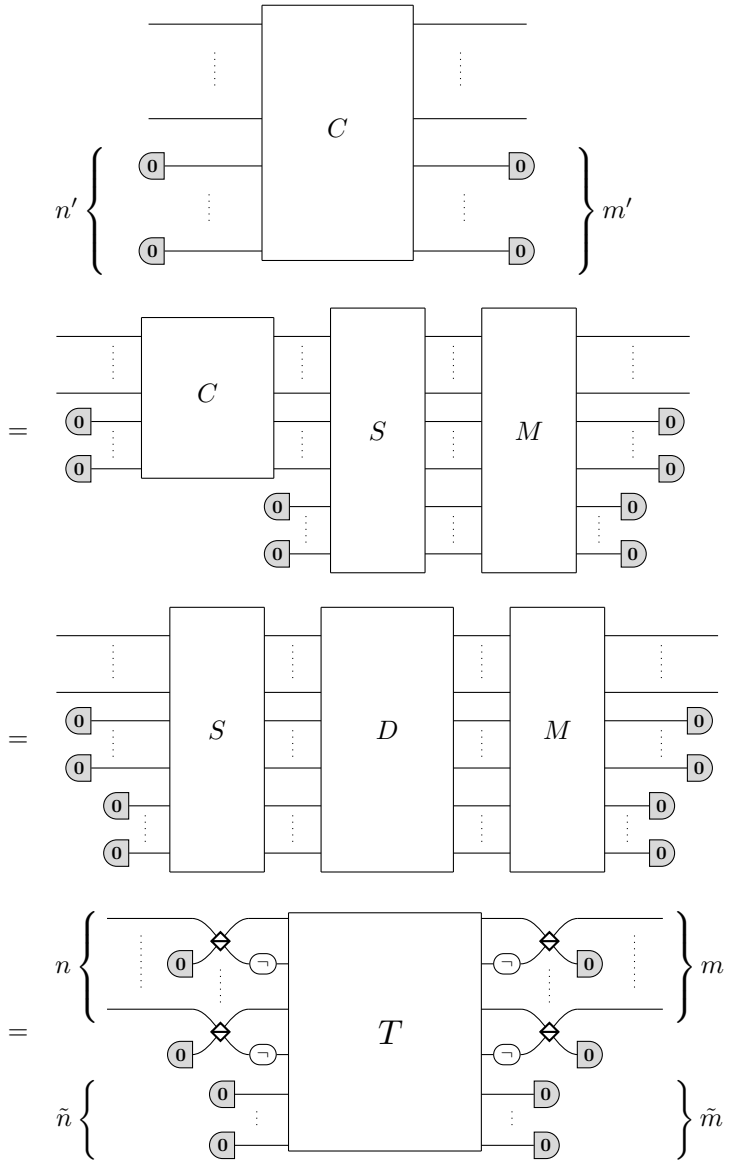
where C' is **LO$_\mathsf{v}$**-circuit without any source or detector. Let consider a layer of $n + n'$ splitters:



and $m + m' = n + n'$ mergers:



By propagating each component through the splits, we can convert any linear optical component into only phase shifters and beam splitters, providing us the desired normal form.



where $\tilde{n} = n + 2n'$ and $\tilde{m} = m + 2m'$. We obtained the last equality by absorbing all polarizing beam splitters and wave plates of the splitters and mergers into the vacuum sources and detectors with

(A.11), (A.14), (III.11), and (III.12). Also, as D is a $\mathbf{LO}_{\mathbf{PP}}$-circuit, it can be rewritten to its unique normal form T (Proposition III.54).

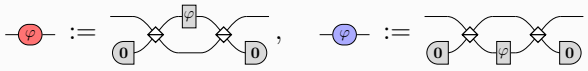
□

III.2.5 Compact universal forms of $\mathbf{LO}_{\mathbf{v}}$-circuits

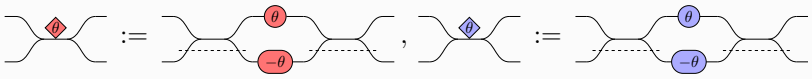
Up to this point, we have used auxiliary vacuum states (with and) in our normal forms (Definition III.59). In such circuits, we encode each wire with two polarization modes into two wires with spatial modes, potentially doubling the number of wires. This section shows that we can aim for more compact and universal forms for $\mathbf{LO}_{\mathbf{v}}$-circuits by trading this increase in width for an increase in depth.

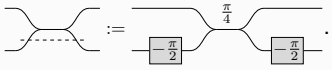
A. Red and blue phase shifters

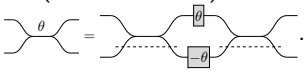
Definition III.62. We define two new macros, labeled as red and blue phase shifters:



We also define red and blue beam splitters :



where  .

Remark III.63. The red and blue phases can be physically implemented with birefringent materials [CB21]. The red and blue beam splitters are the polarization analogy of Mach-Zehnder interferometers (Definition I.35), which would be the usual physical implementation of a parametrised beam splitter  .

Remark III.64. We can show that red and blue beam splitters have the following semantics:

$$[\![-\varphi-]\!]_{\mathsf{v}} = \begin{cases} |V_0\rangle \mapsto e^{i\varphi}|V_0\rangle \\ |H_0\rangle \mapsto |H_0\rangle \end{cases} \qquad \left[\!\!\left[\theta \right]\!\!\right]_{\mathsf{v}} = \begin{cases} |V_0\rangle \mapsto \cos(\theta)|V_0\rangle + i\sin(\theta)|V_1\rangle \\ |H_0\rangle \mapsto |H_0\rangle \\ |V_1\rangle \mapsto i\sin(\theta)|V_0\rangle + \cos(\theta)|V_1\rangle \\ |H_1\rangle \mapsto |H_1\rangle \end{cases}$$

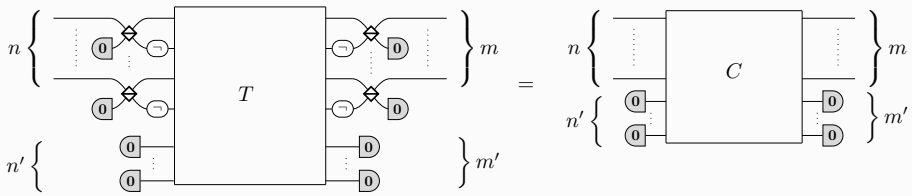
$$[\![-\varphi-]\!]_{\mathsf{v}} = \begin{cases} |V_0\rangle \mapsto |V_0\rangle \\ |H_0\rangle \mapsto e^{i\varphi}|H_0\rangle \end{cases} \qquad \left[\!\!\left[\theta \right]\!\!\right]_{\mathsf{v}} = \begin{cases} |V_0\rangle \mapsto |V_0\rangle \\ |H_0\rangle \mapsto \cos(\theta)|H_0\rangle + i\sin(\theta)|H_1\rangle \\ |V_1\rangle \mapsto |V_1\rangle \\ |H_1\rangle \mapsto i\sin(\theta)|H_0\rangle + \cos(\theta)|H_1\rangle \end{cases}$$

B. Compact Normal Form

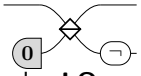
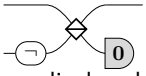
Considering the macros defined in Definition III.62, we can define a notion of universal *compact normal form* that does not use any auxiliary vacuum state.

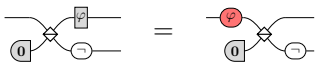
Theorem III.65 (Compact normal form)**.** *For any* $\mathbf{LO}_{\mathbf{v}}$*-circuit* $N : n \to m$ *in normal form, there exists a compact normal form* $C : n \to m$ *with the same semantics, that is only containing wave plates,* *, and colored phase shifters and beam splitters (Definition III.62), and does not have any additional auxiliary mode. In the following, the left-hand side is the original normal form of*

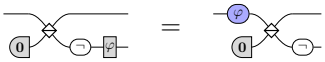
Definition III.59, while the right-hand side is the desired compact form.

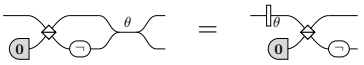


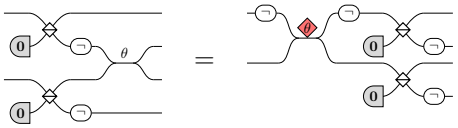
Proof. Instead of directly showing the compact normal form and checking that the semantics is correct, we prefer to give a rewrite procedure to transform one into the other.

For a $\mathbf{LO_v}$-circuit $N : n \to n$ in normal form as shown in Definition III.59, the auxiliary vacuum states are introduced by two operators  and  : we only need to get rid of them. Note we can start from a rectangular $\mathbf{LO_{PP}}$-circuit, as displayed in Figure I.4. Then we have four cases, depicted by the four following equations:

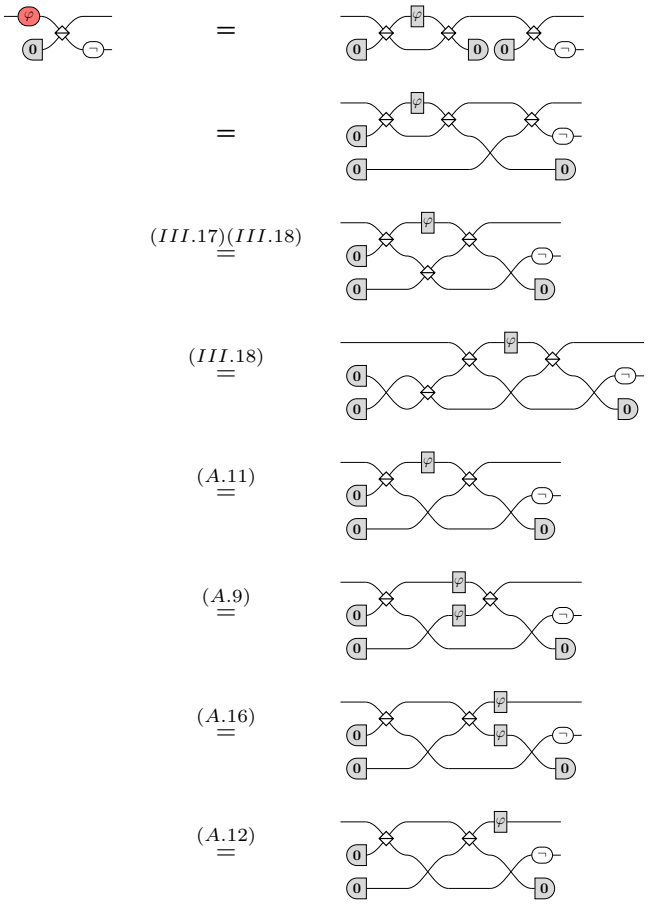
 (III.27)

 (III.28)

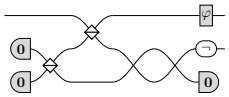
 (III.29)

 (III.30)

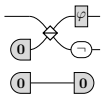
To derive (III.27):



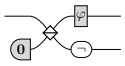
$(III.17)(III.18)$ $=$



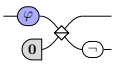
$(A.11)$ $=$



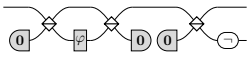
$(A.8)$ $=$



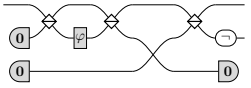
To derive (III.28):



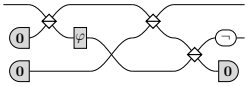
$=$



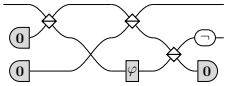
$=$



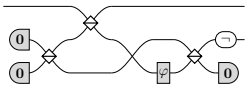
$(III.17)$ $=$



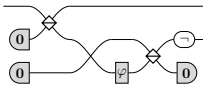
$=$



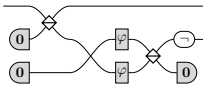
$(III.17)(III.18)$ $=$



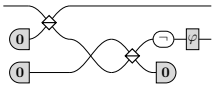
$(A.11)$ $=$



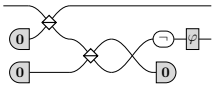
$(A.9)$ $=$



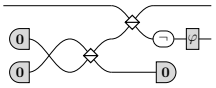
$(A.16)(A.12)(III.10)$ $=$



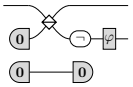
$(III.15)$ $=$



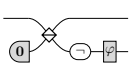
$(III.18)$ $=$



$(A.11)$ $=$

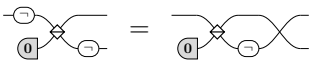


$(A.8)$ $=$

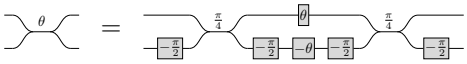


Equation (III.29) is the same as (III.23).

To derive (III.30), we use the following auxiliary equations:

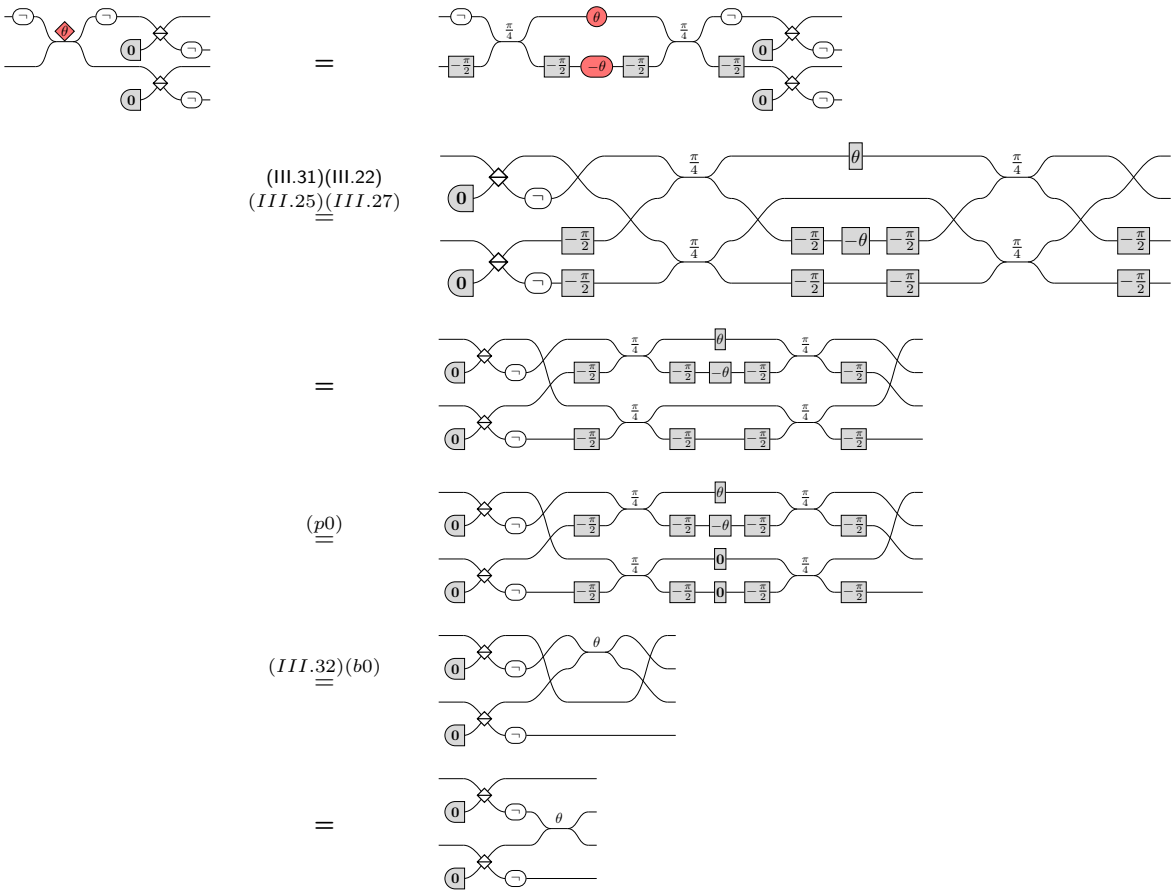


(III.31)



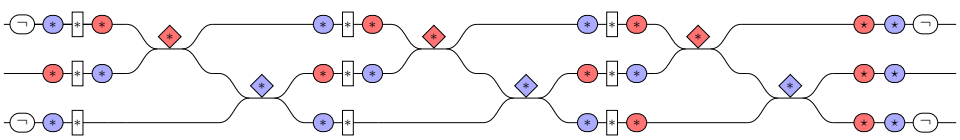
(III.32)

Now, the derivation of (III.30) is the following:

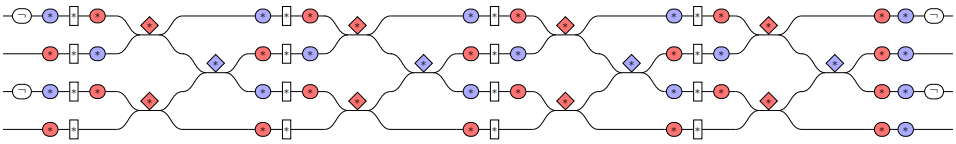


□

Example III.66 (3-mode compact form)**.** Here is the compact normal form of **LO$_v$**-circuits on 3 modes.

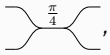


Example III.67 (4-mode compact form)**.** Here is the compact normal form of **LO$_v$**-circuits on 4 modes.

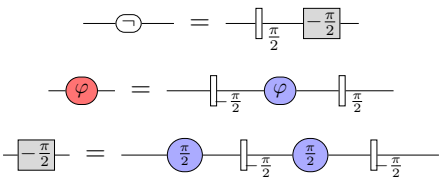


C. Discussion and Choice of Generators

Although one can take as primitives every single component introduced in the **LO$_{\mathbf{v}}$**-calculus, or all the components together with the blue and red phase shifters, it is also possible to only consider a subset of them. This can be useful depending on the availability and difficulty of working with different components in an experimental setting.

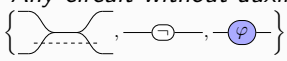
Theorem III.68. *Any circuit without auxiliary modes can be decomposed with*  , , *and* .

Proof. Aside from the already existing elements of the set of generators, a compact normal form is composed of: , and . It is therefore sufficient to show how to decompose these three elements with our generators, which can be done as follows.

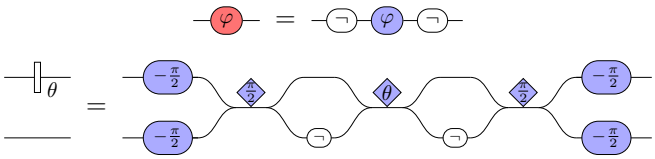


□

Remark III.69. For **LO$_{\mathbf{PP}}$**-circuits, many fixed angles of beam splitters can be used for universal sets [BA14].

Theorem III.70. *Any circuit without auxiliary modes and of at least two spatial modes can be written with only*  .

Proof. Aside from the generators, the compact normal form only uses two elements: and , so it suffices to show their decompositions:



Note the necessity of the additional mode, thus two spatial modes are needed. □

III.3 LO$_{\mathbf{fi}}$: Linear optical circuit with finite-number-photon sources and detectors

In this section, we present the **LO$_{fi}$**-calculus (Linear Optical circuits with finite-number-photon sources and detectors). We begin by introducing its syntax and semantics (Section III.3.1). Next, we provide a complete equational theory (Section III.3.2). We then present a rewriting system that we prove to be strongly normalizing to normal forms (Section III.3.3). We conclude by proving the uniqueness of such normal forms, introducing key new morphisms, and thereby proving the completeness of the **LO$_{fi}$**-calculus (Section III.3.4).

Contributions

This section presents [Heu25], a personal contribution published in CSL'25.

III.3.1 Syntax and many-photon semantics

A. Fock space

As described in Section III.1.1, the state space of one photon with m spatial modes is $\mathbb{C}^m$, as it can be on a superposition of all the different positions. Photons are particles that can bunch and share the same state, so each mode can be occupied by many photons. Furthermore, to observe quantum effects like interferences, we need the photons to be indistinguishable, meaning we cannot know which photon is in which mode.

For those two reasons, the usual way to represent a state with indistinguishable photons is by using the occupation number representation, where we indicate how many photons are there in each state. We consider the basis states $\mathcal{F}_m \coloneqq \{|n_1, n_2, \dots, n_m\rangle, (n_1, n_2, \dots, n_m) \in \mathbb{N}^m\}$, denoted as Fock states (Section I.2.3). The state $|n_1, n_2, \dots, n_m\rangle$ represents a configuration where n_i photons occupy the i^{th} mode. The space of possible many-photon states over m modes, the bosonic (symmetrical) Fock space and denoted as $\mathcal{F}_m$ (Definition I.3), where we recall here its definition.

Definition III.71 (Fock space)**.** We define the Hilbert space $\mathcal{F}_m$ as the Hilbert completion $\ell^2(\mathcal{F}_m)$ equipped with the bra-ket inner product $\langle\cdot|\cdot\rangle$, with the convention $\mathcal{F}_0 = \mathbb{C}$.

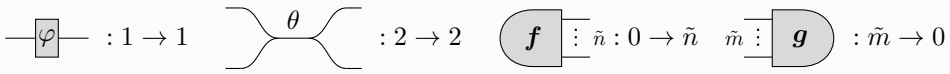
Remark III.72. $\mathcal{F}_1$ contains states that are an *infinite* superposition of basis states, like the coherent states $|\alpha\rangle = e^{-\frac{|\alpha|^2}{2}} \sum_{k=0}^{\infty} \frac{\alpha^k}{k!} |k\rangle$. We can note that $\mathcal{F}_1$ is isomorphic to $\ell^2(\mathbb{N})$.

To describe the space of the auxiliary sources, we consider a sub vector space of $\mathcal{F}_m$.

Definition III.73 (Subspace of the Fock space: $\mathcal{F}_m^{\text{pre}}$)**.** We define the pre-Hilbert space $\mathcal{F}_m^{\text{pre}}$ as the linear span of $\mathcal{F}_m$ equipped with the bra-ket inner product $\langle\cdot|\cdot\rangle$, with the convention $\mathcal{F}_0^{\text{pre}} = \mathbb{C}$. We define the pre-Hilbert space $\mathcal{B}_{\tilde{m}}^{\text{pre}*}$ as the linear span of $\left\{\langle n_1, \dots, n_{\tilde{m}}|, (n_1, n_2, \dots, n_{\tilde{m}}) \in \mathbb{N}^{\tilde{m}}\right\}$.

Remark III.74. $\mathcal{F}_m^{\text{pre}}$ only contain states that are *finite* linear combination of the Fock basis states. In particular, the coherent states are not included. We can note that $\mathcal{F}_1^{\text{pre}}$ is isomorphic to c_{00}, i.e. the space of eventually zero sequences.

Definition III.75 (**LO**$_{fi}$-calculus)**.** **LO**$_{fi}$ is the PROP of **LO**$_{fi}$-circuits generated by



where $\varphi, \theta \in \mathbb{R}$, and $\boldsymbol{f}$ (resp. $\boldsymbol{g}$) is a state in $\mathcal{F}_{\tilde{n}}^{\text{pre}}$ (resp. in $\mathcal{B}_{\tilde{m}}^{\text{pre}*}$) with $\tilde{n}, \tilde{m} \in \mathbb{N}^+$.

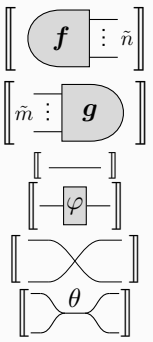
Definition III.76 (Conventions for the notations)**.** For readability and make the notions more accessible, we introduce the following conventions.

- The terms $\boldsymbol{f}, \boldsymbol{f}_k$ (resp. $\boldsymbol{g}, \boldsymbol{g}_\ell$) always represent kets (resp. bras).
- Bold integers $\boldsymbol{k}$ (resp. $\boldsymbol{\ell}$) will represent $|k\rangle$ (resp. $\langle\ell|$) in the sources (resp. detectors).
- The summation term $\sum$ is often omitted for readability, the index being implicitly the sum index. For instance $\boldsymbol{f} = \sum_{k\in\mathcal{K}} |\boldsymbol{f}_k\rangle\, |k\rangle$ may be simply noted as $\boldsymbol{f}_k \otimes \boldsymbol{k}$.
- $|.\rangle$ (resp. $\langle.|$) represents an arbitrary ket (resp. bra) on one mode. $|\dots\rangle$ (resp. $\langle\dots|$) represents

an arbitrary ket (resp. bra) for an arbitrary number of modes, representing an arbitrary scalar when the number of modes is zero. Those are used to omit terms when the specific value of those terms are not of interest, as in some equations of Figure III.7.

- For the zero vector $\boldsymbol{f} = 0$ (resp. $\boldsymbol{g} = 0$), as there is no term in the sum, we chose to represent it with (resp. an empty detector). Note it is different from (resp.) representing the nonzero vector $|0\rangle$ (resp. $\langle 0|$).

Definition III.77. Let $C\colon n \to m$ a **LO$_{fi}$**-circuit, let $[\![C]\!] : \mathcal{F}_n \to \mathcal{F}_m$ be the linear map inductively defined as $[\![C_2 \circ C_1]\!] = [\![C_2]\!] \circ [\![C_1]\!]$, $[\![C_1 \otimes C_2]\!] = [\![C_1]\!] \otimes [\![C_2]\!]$ and:



where $B_\theta(|k_1, k_2\rangle) \coloneqq \sum_{\ell_1+\ell_2=k_1+k_2} \sqrt{\frac{\ell_1!\ell_2!}{k_1!k_2!}} \sum_{\substack{p+q=\ell_1 \\ \delta=p-q}} \binom{k_1}{p}\binom{k_2}{q} c_\theta^{k_2+\delta} (is_\theta)^{k_1-\delta} |\ell_1, \ell_2\rangle$ and $P_\varphi(|k\rangle) \coloneqq e^{ik\varphi}|k\rangle$, with the convention $\binom{k}{k'} = 0$ for $k < k'$.

We can check that P_φ and B_θ are unitary operators [AA11] and are therefore well-defined on the Hilbert space by continuity and linearity. One can also look at the Examples III.103 and III.104 to link this formula with the usual single-photon definition as in Section I.3.3.

Remark III.78 (Hermitian conjugate). We have $P_\varphi^\dagger = P_{-\varphi}$ and $B_\theta^\dagger = B_{-\theta}$, where $\dagger$ is the Hermitian conjugate. Therefore, $\langle \ell | P_\varphi = (P_{-\varphi}|\ell\rangle)^\dagger$ and $\langle \ell_1, \ell_2 | B_\theta = (B_{-\theta}|\ell_1, \ell_2\rangle)^\dagger$.

B. Choice of the generators

The sources and detectors of the **LO$_{fi}$**-calculus allow any and arbitrary finite support state on many modes, which may seem to be too powerful or far from the physical implementation. In that regard, we would like to highlight that:

- Some sources can directly generate more generic states such as a coherent superposition with the vacuum of the 2-photon state [Lor+19], or even directly create entangled states [Cos+23].
- Linear optical circuits are very modular, and each building block is usually used many times. It would therefore be more convenient to sometimes represent those building blocks directly by specifying what they do, instead of how they are implemented.
- Optical interactions are very combinatorial, thus being unlikely to have a complete equational theory with only single mode sources [7].
- This formalism still allows finding new results for linear optics, like the unique normal forms presented in Section III.3.4.

7. We can note that [FC22] bypasses that problem by allowing sums of diagrams

III.3.2 Complete equational theory of $\mathbf{LO}_{\mathbf{fi}}$

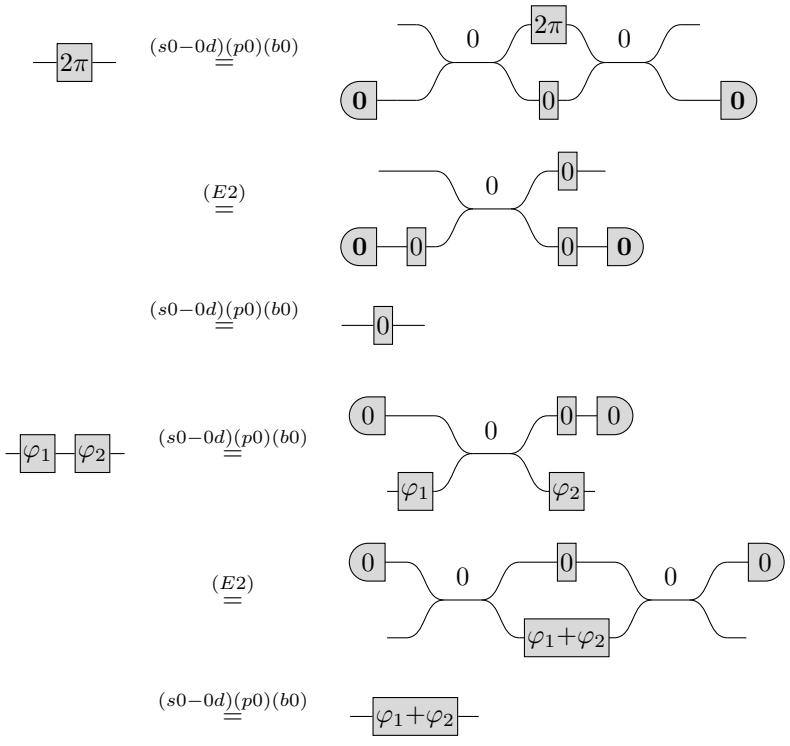
Definition III.79 (**LO**$_{fi}$-calculus)**.** Two **LO**$_{fi}$-circuits C, C' are equivalent according to the rules of the **LO**$_{fi}$-calculus, denoted **LO**$_{fi} \vdash C = C'$, if one can transform C into C' using the equations given in Figure III.7.

Proposition III.80 (Soundness)**.** *For any two* **LO**$_{fi}$*-circuits* C *and* C'*, if* **LO**$_{fi} \vdash C = C'$ *then* $[\![C]\!] = [\![C']\!]$.

Proof. Since semantic equality is a congruence, it suffices to check the soundness for every equation of Figure III.7, which follows from Theorem III.7 and Definition III.77. Informally, Axiom (zero) means that if there is the scalar[8] 0, then the semantics of all the circuit ($X \otimes 0 = 0$) is the null function. We can therefore nullify the other wires with the zeros $\boxed{\emptyset}$— and —$\boxed{\emptyset}$. This rule is specifically used for Proposition III.93. Axiom (s-0d) means we can either (from LHS to RHS) project on $|0\rangle$ on the last mode or (from right to left) add any states $\boldsymbol{f}_k \otimes |k\rangle$ with $k \neq 0$ as they are trivially orthogonal and cancelling. Axiom (h2) means we can *shift* any function $h : \mathcal{F}_2^{\mathsf{pre}} \to \mathcal{F}_2^{\mathsf{pre}}$ from left to right where there are identity wires, direct consequence of the associativity: $\langle \ell_1, \ell_2 |\, (h\, |k_1, k_2\rangle) = (\langle \ell_1, \ell_2 |\, h)\, |k_1, k_2\rangle$. The rules (dd), (b-d), (p-d) and (s0-d) are respectively the duals of (ss), (s-b), (s-p) and (s-0d). □

Lemma III.81 (**LO**$_{fi}$ is complete for **LO**$_{\mathsf{PP}}$)**.** *We can derive the equations of Figure III.1 with the rules of the* **LO**$_{fi}$*-calculus.*

Proof. All the rules are present aside from (p2π) and (p-p). We derive them as follows:



□

8. $\boxed{\mathbf{0}}$—$\boxed{\mathbf{1}}$ is an impossible event and is one way to represent the scalar $0 = \langle 1|0\rangle = [\![\boxed{\mathbf{0}}—\boxed{\mathbf{1}}]\!]$.

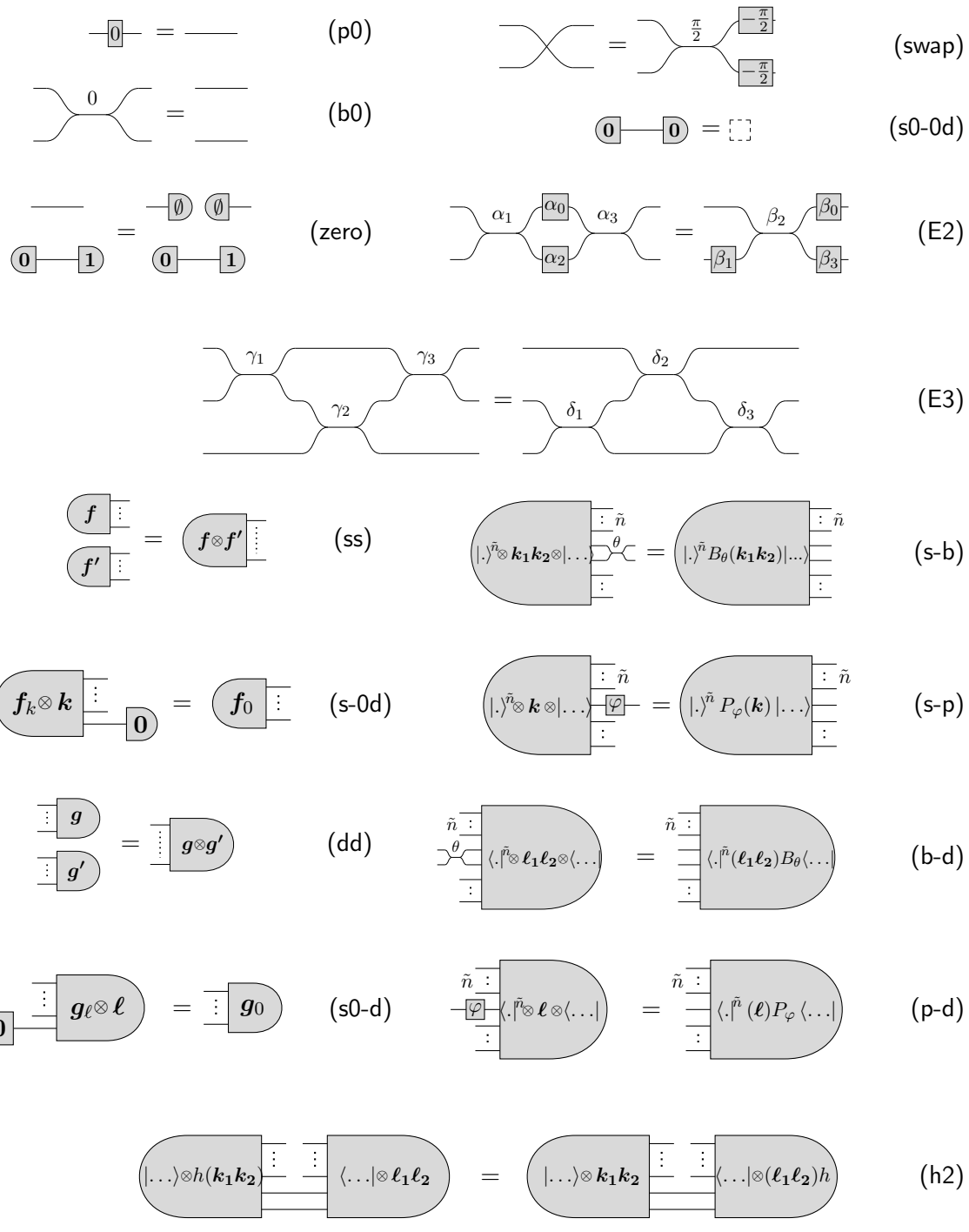


Figure III.7 – Axioms of the $\mathbf{LO}_{fi}$-calculus. The angles of (E2) and (E3) are the same as in the axioms of the $\mathbf{LO}_{\mathbf{PP}}$-calculus (Figure III.1). h is any linear function $\mathcal{F}_2^{\mathsf{pre}} \to \mathcal{F}_2^{\mathsf{pre}}$. The conventions for $\{\emptyset, |.\rangle, |\ldots\rangle, \langle.|, \langle\ldots|\}$, and the omitted sums are detailed in Definition III.76. The interpretations of the axioms are given in Proposition III.80.

Theorem III.82 (Completeness)**.** *For any two* $\mathbf{LO}_{fi}$*-circuits* C *and* C'*, if* $[\![C]\!] = [\![C']\!]$ *then* $\mathbf{LO}_{fi} \vdash C = C'$*.*

Proof. In Section III.3.3, we introduce a rewriting system (Definition III.83). We show that we can derive the rewriting rules with the axioms of the $\mathbf{LO}_{fi}$-calculus (Lemma III.87). The Section III.3.4 shows that such normal forms are unique (Lemma III.95), concluding the proof. □

III.3.3 Strongly normalizing rewriting system

A. Rewriting system

A strongly normalizing rewriting system, i.e. terminating to normal forms, has been introduced in Definition III.29 for $\mathbf{LO}_{\mathsf{PP}}$-circuits. We mainly reuse all the rules, alongside additional rules to now take into account the sources and the detectors.

Definition III.83 ($\mathbf{LO}_{fi}$ rewriting system)**.** Our rewriting system is defined on the PRO of $\mathbf{LO}_{fi}$-circuits with the rules of Figure III.8.

Definition III.84 (Inputs of the rewriting system)**.** For convenience, the inputs of the rewriting procedures are $\mathbf{LO}_{fi}$-circuits with at least one identity wire connecting sources and generators, and where all the sources (resp. detectors) are on the bottom left (resp. right).

Remark III.85. Note that choice is not restrictive, as the identity wire can always be added with Axiom (s0-0d), and the sources and detectors can be placed at the desired position, without changing the semantics, with the rules of PROP and by adding SWAPs.

Proposition III.86 (Soundness of $\mathbf{LO}_{fi}$ rewriting system)**.** *All the rules of Figure III.8 are sound.*

Proof. The soundness is a direct consequence of Definition III.77. Note they have the following meaning.

- The rules on the $\mathbf{LO}_{\mathsf{PP}}$-circuits are the same as the rules in Figure III.2. Therefore, any $\mathbf{LO}_{\mathsf{PP}}$-circuit converges to a $\triangle^*$-circuit (Definition III.17).
- The rule III.44 removes any vector $|f_{k'}\rangle \otimes |k'\rangle$ in the sources that is trivially cancelling with the detector on the connected last wire, meaning that $\langle g_{k'}| = 0$, i.e. that $k' \notin \mathcal{L}$.
- The rule III.45 removes any $\langle g_{\ell'}| \otimes \langle \ell'|$ in the detectors that is trivially cancelling with the source on the connected last wire, meaning that $|f_{\ell'}\rangle = 0$, i.e. that $\ell' \notin \mathcal{K}$.
- The rule III.46 allows, without changing the semantics, to transfer the generic coefficients from the detectors to the sources. Specifically, any $\langle \mathcal{N}_{\tilde{m}}(L)| \langle xi|$ where $\langle \xi|$ is generic, will be rewritten to $\langle \mathcal{N}_{\tilde{m}}(L)| \langle L|$. At the end, there won't be any *degree of freedom* in the detectors, and $\boldsymbol{g} = \sum_{\ell \in \mathcal{L}} \langle \mathcal{N}_{\tilde{m}}(\ell)| \langle \ell|$.
- The rule III.47 uses the bijection $\mathcal{N}_2 : \mathbb{N} \to \mathbb{N}^2$ to remove one identity wire, by just relabelling the indexes in the sources and detectors. Note that one identity wire will always remain at the end.
- The oriented rule (from left to right) coming from the axioms ss and dd merge all the sources and detectors together.

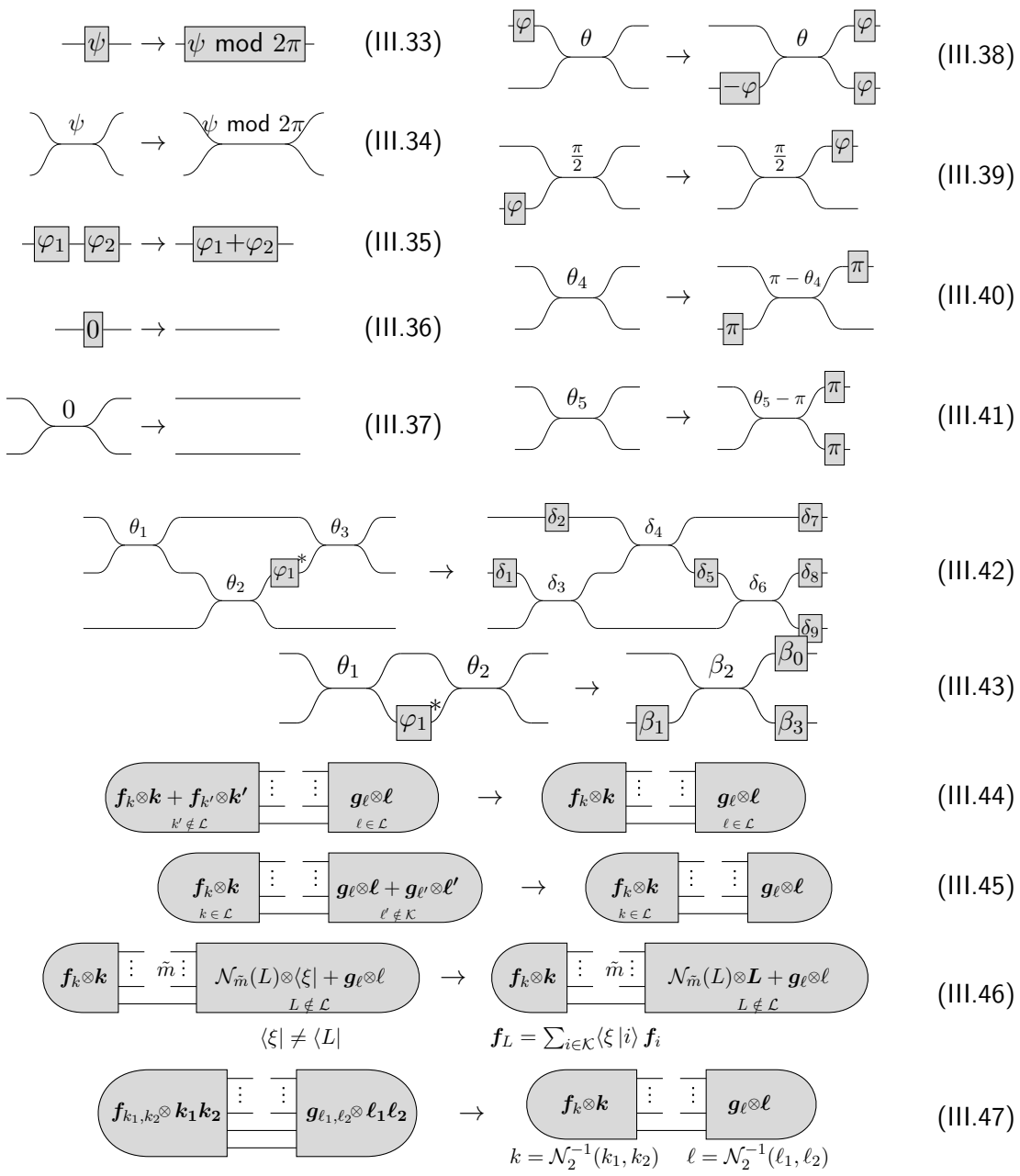


Figure III.8 – Rewriting system of the **LO**$_{fi}$-calculus, alongside with the oriented version, from the LHS to the RHS, of the axioms (ss), (s-b), (s-p), (dd), (b-d), and (p-d). $\psi \in \mathbb{R} \setminus [0, 2\pi)$, $\varphi, \varphi_1, \varphi_2 \in (0, 2\pi)$, $\theta_4 \in (\frac{\pi}{2}, \pi)$, $\theta_5 \in [\pi, 2\pi)$, $\theta, \theta_1, \theta_2, \theta_3 \in (0, \pi)$. Note that $\overset{*}{\boxed{\varphi}}$ denotes either $\boxed{\varphi}$ or ——.

- The oriented rule (from left to right) coming from the axioms s-b, s-p, b-d and p-d reduce the number of phase shifters and beam splitters as much as possible, by making them be *absorbed* into the sources and detectors.

□

Lemma III.87. *If C_1 rewrites to C_2 using the rules of Figure III.8, then* $\mathbf{LO}_{fi} \vdash C_1 = C_2$.

Proof. The rules are derived in Section III.3.3.C.. □

B. Strongly normalizing

Proposition III.88 (Strongly normalizing)**.** *The rewriting system of Figure III.8 is strongly normalizing, i.e. every sequence of rewriting rules eventually terminates.*

Proof. We define a ranking function $(x_1, x_2, x_3, x_4, x_5, x_6) \in \mathbb{N}^3$, that we show to be strictly decreasing with respect to lexicographic order. We define x_1, x_2, x_3 exactly as in the proof of Proposition III.34. We define the other variables as follows.

- x_4 is the number of identity wire connecting the sources and detectors.
- x_5 is the number of sources and detectors in C,
- to define x_6 let write in the Fock basis the generic terms in the sources as

$$\boldsymbol{f} = \sum_{k_1,\dots,k_{\tilde{n}+1}} \alpha_{k_1,\dots,k_{\tilde{n}+1}} |k_1, \dots, k_{\tilde{n}+1}\rangle$$

and in the detectors as

$$\boldsymbol{g} = \sum_{\ell_1,\dots,\ell_{\tilde{m}+1}} \beta_{\ell_1,\dots,\ell_{\tilde{m}+1}} |\ell_1, \dots, \ell_{\tilde{m}+1}\rangle$$

. Then

$$x_6 \coloneqq \sum_{\boldsymbol{f} \in \text{sources}} C_1(\boldsymbol{f}) + \sum_{\boldsymbol{g} \in \text{detectors}} (2C_2(\boldsymbol{g}) - C_3(\boldsymbol{g}))$$

, with $C_1(\boldsymbol{f}) \coloneqq \#\{\alpha_{k_1,\dots,k_{\tilde{n}+1}} \neq 0\}$, $C_2(\boldsymbol{g}) \coloneqq \#\{\beta_{\ell_1,\dots,\ell_{\tilde{m}+1}} \neq 0\}$, and $C_3(\boldsymbol{g}) \coloneqq \#\{\beta_{\mathcal{N}_{\tilde{m}}(L),L} = 1, L \in \mathbb{N}\}$.

We highlight that (x_1, x_2, x_3) are decreasing for the reasons described in Proposition III.34. We conclude the proof by noticing that:

- Rule III.44 strictly decreases C_1 without changing anything else, therefore strictly decreasing x_6.
- Rule III.45 strictly decreases $2C_2 - C_3$, without changing anything else. Indeed, if we remove a coefficient where $\beta_{\mathcal{N}_{\tilde{m}}(L),L} = 1$, then $-C_3$ increases by 1 but $2C_2$ decreases by two. Therefore, $2C_2 - C_3$ strictly decreases and x_6 is strictly decreasing.
- Rule III.46 can change the three values of C_1, C_2 and C_3, but still strictly decreases x_6 in any cases. Let us consider the two cases: $(\xi_L \neq 1) \wedge (\forall \ell \neq L, \xi_\ell = 0)$ and $(\xi_L = 1) \wedge (\exists \ell \neq L, \xi_\ell = 0)$. The first case doesn't change C_1 and C_2, but the term $-C_3$ strictly decreases by 1. The second case doesn't change C_3, and the increase of C_1, i.e. the amount of new terms in $\boldsymbol{f}$, is bounded by $\#\{\xi_i \neq 0, i \neq L\}$, the number of terms removed in $\boldsymbol{g}$, which is the exact decrease of C_2. Therefore, $C_1 + 2C_2$ decreases by at least $\#\{\xi_i \neq 0, i \neq L\} > 0$. We can check that the rule III.45 is not changing any other component of the tuple, and can conclude it only strictly decreases x_6.

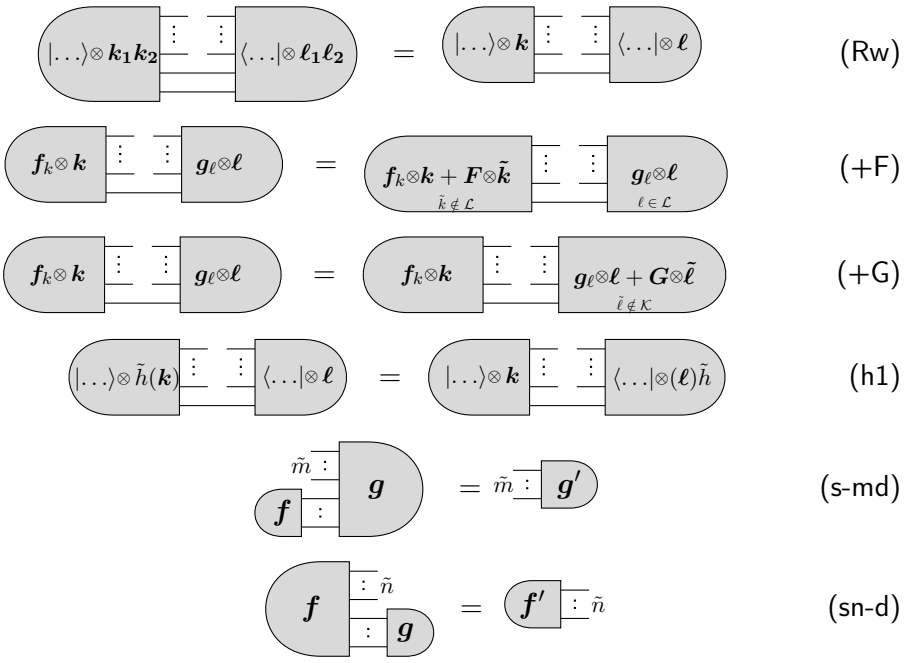


Figure III.9 – Useful equations derived by the **LO**$_{fi}$-calculus as show in Lemma III.89. Equation (Rw) is removing one identity wire when there are two, Equation (+F) (resp. (+G)) add any term in the source (resp. detector) trivially orthogonal on the last mode, Equation (h1) is the same as Axiom (h2) on one mode, Equations (s-md) and (sn-d) are projections.

- Rule III.47 strictly decreases x_4 without changing previous components of the tuple.
- The rules coming from the axioms ss and dd strictly decrease x_5 without changing the previous components.
- The rules coming from the axioms s-p and p-d strictly decrease x_3.
- The rules coming from the axioms s-b and b-d strictly decrease x_1.

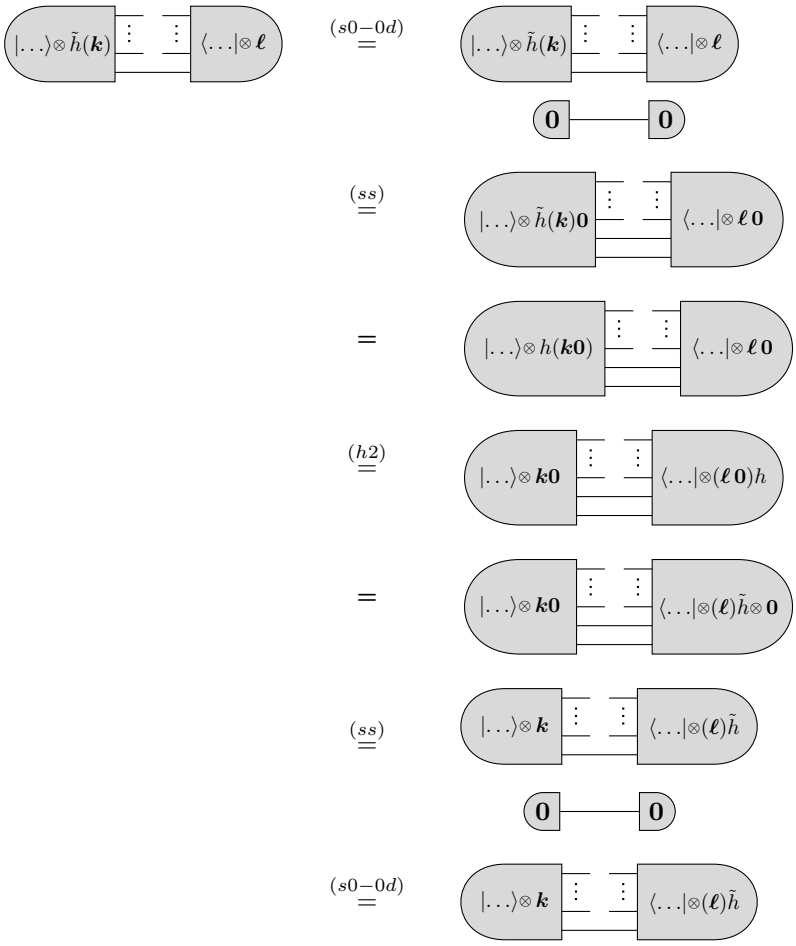
□

C. Derivation of the rewriting system

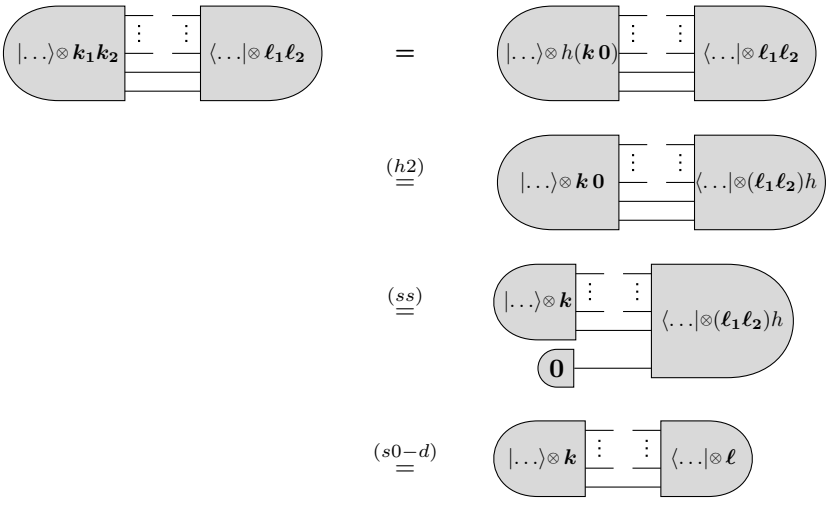
Note that all equations including **LO**$_{\mathsf{PP}}$-circuits of Figure III.8 can be derived, as a direct collorary of Lemma III.81. For the other equations, we show that we can derive all equations of Figure III.9, including the rules (III.44), (III.45) and (III.47). The rule (III.46) is a special case proven in Lemma III.90.

Lemma III.89 (Useful **LO**$_{fi}$-equations)**.** *We can derive the equations of Figure III.9 with the rules of the* **LO**$_{fi}$*-calculus.*

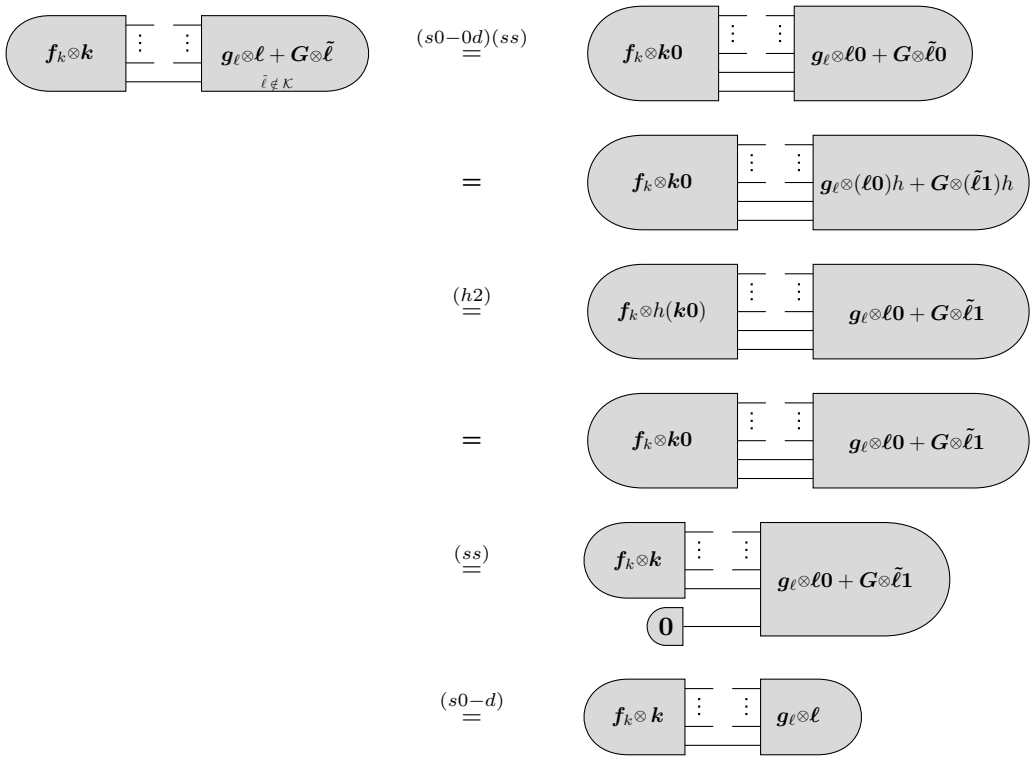
Proof. We consider a linear function $h : \mathcal{F}_2^{\mathsf{pre}} \to \mathcal{F}_2^{\mathsf{pre}}$ such that for every $k \in \mathbb{N}$, $h(|k,0\rangle) = \tilde{h}(|k\rangle)$ and $(\langle k,0|)h = (\langle k|)\tilde{h}$.



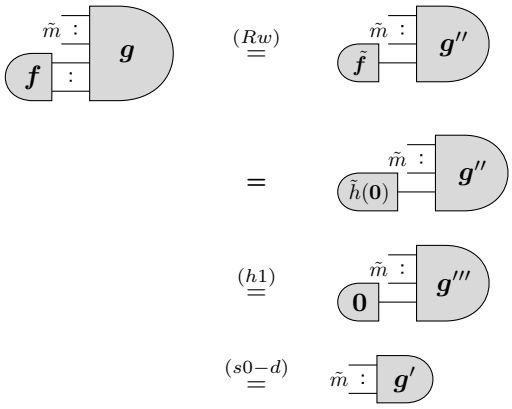
We consider a linear function $h : \mathcal{F}_2^{\mathsf{pre}} \to \mathcal{F}_2^{\mathsf{pre}}$ and a finite set $\mathcal{K} \subset \mathbb{N}$ such that for every $k \in \mathcal{K}$, $h(|k, 0\rangle) = |k_1, k_2\rangle$.



We consider a linear function $h : \mathcal{F}_2^{\mathsf{pre}} \to \mathcal{F}_2^{\mathsf{pre}}$ such that $h = h^\dagger$, h is swapping the states $\left|\tilde{\ell}, 0\right\rangle$ and $\left|\tilde{\ell}, 1\right\rangle$ and is the identity everywhere else. We have:



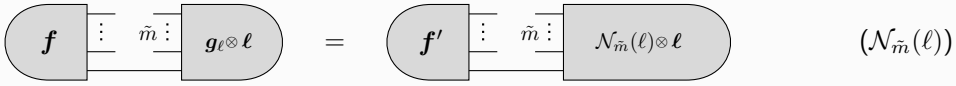
Similarly and symmetrically, we can derive Equation (+F).
We consider a linear map $\tilde{h} : \mathcal{F}_1^{\mathsf{pre}} \to \mathcal{F}_1^{\mathsf{pre}}$ such that $\tilde{h}(|0\rangle) = \tilde{f}$. We have:



Similarly and symmetrically, we can derive Equation (sn-d).

□

Lemma III.90. *We can derive the following equation in the* $\mathbf{LO}_{fi}$*-calculus:*



Proof. Without loss of generality, just a different way of decomposing, we can write $g = \sum_\ell g_\ell \otimes \langle \ell | = \sum_\ell \langle \mathcal{N}_m(\ell) | \langle \psi_\ell |$ where $|\psi_\ell\rangle$ is a generic state of $\mathcal{F}_1^{\mathsf{pre}}$. We will show we can derive the following

transformation:

$$\boldsymbol{g}^{L} = \sum_{\ell<L} \langle \mathcal{N}_m(\ell)| \langle \ell| + \sum_{\ell\geq L} \langle \mathcal{N}_m(\ell)| \langle \psi_\ell| \quad \text{with } v < L \Rightarrow \langle \psi_\ell|v\rangle = 0$$
$$\rightarrow \quad \boldsymbol{g}^{L+1} = \sum_{\ell\leq L} \langle \mathcal{N}_m(\ell)| \langle \ell| + \sum_{\ell>L} \langle \mathcal{N}_m(\ell)| \langle \psi'''_\ell| \quad \text{with } v \leq L \Rightarrow \langle \psi''_\ell|v\rangle = 0$$

We take a linear function $\tilde{h} : \mathcal{F}_1^{\text{pre}} \rightarrow \mathcal{F}_1^{\text{pre}}$ which is the identity for every $|k\rangle$ with $k < L$ and such that $\langle L|\,\tilde{h} = \langle \psi_L|$ and $\langle \psi'_\ell|\,\tilde{h} = \langle \psi_\ell|$ with $\langle \psi'_\ell|k\rangle$ for $k < L$. We have:

$$\boldsymbol{g}^{L} = \sum_{\ell<L} \langle \mathcal{N}_m(\ell)| \langle \ell|\,\tilde{h} + \langle \mathcal{N}_m(L)| \langle L|\,\tilde{h} + \sum_{\ell>L} \langle \mathcal{N}_m(\ell)| \langle \psi'_\ell|\,\tilde{h}$$

Note that $\langle \psi'_\ell|L\rangle$ can be nonzero. We remove $\tilde{h}$ with Equation (h1):

$$\sum_{\ell<L} \langle \mathcal{N}_m(\ell)| \langle \ell| + \langle \mathcal{N}_m(L)| \langle L| + \sum_{\ell>L} \langle \mathcal{N}_m(\ell)| \langle \psi'_\ell|$$

We extract $\langle \Psi| = \sum_{\ell>L} \langle \mathcal{N}_m(\ell)| \langle \psi'_\ell|L\rangle$:

$$\sum_{\ell<L} \langle \mathcal{N}_m(\ell)| \langle \ell| + (\langle \mathcal{N}_m(L)| + \langle \Psi|) \langle L| + \sum_{\ell>L} \langle \mathcal{N}_m(\ell)| \langle \psi''_\ell|$$

with $\langle \psi''_\ell|k\rangle = 0$ for $k \leq L$. We use Equation (+G) to add $\langle G|\left\langle \tilde{\ell}\right|$ with $\langle G| = \langle \mathcal{N}_m(L)| - \langle \Psi|$ and $\tilde{\ell} \notin \mathcal{K} \cup \mathcal{L}$. We have:

$$\sum_{\ell<L} \langle \mathcal{N}_m(\ell)| \langle \ell| + \langle \mathcal{N}_m(L)| \left(\langle L| + \left\langle \tilde{\ell}\right|\right) + \langle \Psi| \left(\langle L| - \left\langle \tilde{\ell}\right|\right) + \sum_{\ell>L} \langle \mathcal{N}_m(\ell)| \langle \psi''_\ell|$$

By taking a linear function $\tilde{h} : \mathcal{F}_1^{\text{pre}} \rightarrow \mathcal{F}_1^{\text{pre}}$ such that $\langle L|\,h = \langle L| + \left\langle \tilde{\ell}\right|$ and $\left\langle \tilde{\ell}\right|\,h = \langle L| - \left\langle \tilde{\ell}\right|$ and which is the identity everywhere else, we have:

$$\sum_{\ell<L} \langle \mathcal{N}_m(\ell)| \langle \ell|\,\tilde{h} + \langle \mathcal{N}_m(L)| \langle L|\,\tilde{h} + \langle \Psi| \left\langle \tilde{\ell}\right|\,\tilde{h} + \sum_{\ell>L} \langle \mathcal{N}_m(\ell)| \langle \psi''_\ell|\,\tilde{h}$$

By removing the $\tilde{h}$ with Equation (h1) and taking $\langle \psi'''_\ell| = \langle \psi''_\ell| + \langle \Psi|\mathcal{N}_m(\ell)\rangle \left\langle \tilde{\ell}\right|$, we obtain the desired $\boldsymbol{g}^{L+1}$.

We conclude the proof by starting at the generic state $\boldsymbol{g} = \boldsymbol{g}^{L=0}$ and apply that transformation for $L = 0$ to $L = \max(\mathcal{L})$.

□

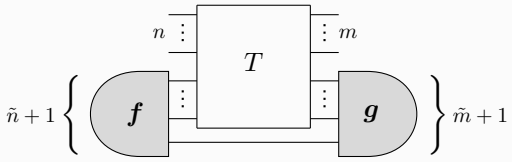
III.3.4 Unique normal forms

First, we present the normal forms of the **LO$_{fi}$**-calculus (Section III.3.4.A.), where we show the uniqueness of the **LO$_{\text{PP}}$**-circuit part (Section III.3.4.B.). Then, we introduce Ω-morphisms, and show that any normal form can be canonically decomposed with them (Section III.3.4.C.). Next, we recall the properties of the evolution of creation operators through linear optical components (Section III.3.4.D.), which enable us to introduce Δ-morphisms, that possess key threshold properties proving their unique decomposition (Section III.3.4.E.). We prove that these two sets of morphisms are isomorphic, therefore proving the uniqueness of the Ω-decomposition, thereby proving the uniqueness of the normal forms (Section III.3.4.F.).

A. Normal forms

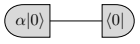
Formally with the rules of Figure III.8 and informally with their meaning described in Section III.1.4.A., we can show that an irreducible form is a $\mathbf{LO}_{fi}$-circuit defined as follows:

Definition III.91 (Normal form of $\mathbf{LO}_{fi}$-circuits)**.** The normal forms of any nonzero $\mathbf{LO}_{fi}$-circuits are denoted $N(T, \boldsymbol{f}) : n \to m$ and are of the form:



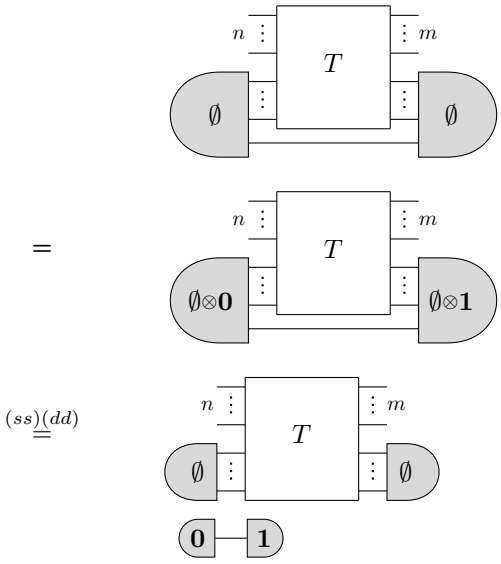
where:

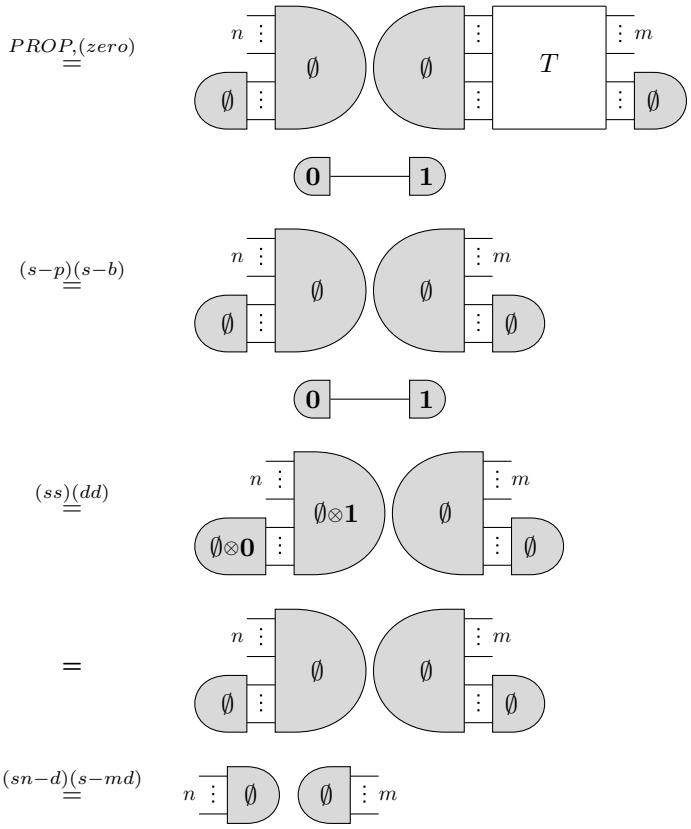
- $\boldsymbol{f}$ is a nonzero generic state of $\mathcal{F}^{\text{pre}}_{\tilde{n}+1}$.
- $\boldsymbol{g} = \sum_{\ell \in \mathcal{K}} \langle \mathcal{N}_m(\ell)| \otimes \langle \ell|$, where $\mathcal{N}_m : \mathbb{N} \to \mathbb{N}^m$ is a bijection defined in Figure III.8 and $\mathcal{K}$ is the nonempty finite set $\{k \in \mathbb{N} \mid \boldsymbol{f}_k \neq 0\}$ of $\boldsymbol{f} = \sum_{k \in \mathcal{K}} \boldsymbol{f}_k \otimes |k\rangle$, with the convention $\mathcal{K} = \{0\}$ if $\tilde{n} = 0$ or $\tilde{m} = 0$.
- $T : n + \tilde{n} \to m + \tilde{m}$ is a ${}_{\tilde{n}}\triangle_{\tilde{m}}$-circuit as defined in Definition III.22.

Remark III.92. If $\tilde{n} = \tilde{m} = 0$, then the normal form is a normal form of $\mathbf{LO}_{\mathbf{PP}}$ (can be $\boxed{\ }$ for $n = m = 0$) tensored with the scalar  which has the semantics of a global scalar $\alpha \in \mathbb{C}$.

Proposition III.93 (Normal form of the zero function)**.** *Let $N : n \to m$ be a normal form such that $\boldsymbol{f} = 0$, i.e. $\mathcal{K} = \emptyset$. Therefore $[\![N]\!]$ is the null function and N can be rewritten to $(\ \boxed{\emptyset}\!- \ )^{\otimes m} \circ (\ -\!\boxed{\emptyset}\ )^{\otimes n}$ with the rules of the $\mathbf{LO}_{fi}$-calculus.*

Proof. Note the zero vector on $\mathcal{F}^{\text{pre}}_2$ can be written as $0 = 0 \otimes |\mathbf{0}\rangle$ and the zero vector on $\mathcal{B}^{\text{pre}*}_2$ as $0 = 0 \otimes \langle \mathbf{1}|$. We have:





□

Note the identity wire allows, in a sense, to capture a sum of diagrams.

Proposition III.94 (Sum of diagrams)**.** *A finite sum of diagrams with the same* **LO$_{\mathsf{PP}}$***-circuit can be factorized with an identity wire connecting entangled sources and detectors.*

Proof.

$$\left[\!\!\left[\quad \right]\!\!\right] = \sum_{k\in\mathcal{K}} \sum_{\ell\in\mathcal{L}} \langle \ell \mid k \rangle \left[\!\!\left[\quad \right]\!\!\right] = \sum_{j\in\mathcal{L}\cap\mathcal{K}} \left[\!\!\left[\quad \right]\!\!\right]$$

□

The main lemma that proves the completeness of the **LO$_{fi}$**-calculus is the following. We give a sketch of the proof here.

Lemma III.95 (Uniqueness of the Normal Forms)**.** *If two* $\mathbf{LO}_{fi}$*-circuits* N *and* N' *in normal forms are such that* $[\![N]\!] = [\![N']\!]$*, then* $N = N'$*.*

Proof. Let $N(T, \boldsymbol{f})$ be a normal form. In order to prove the uniqueness of T and $\boldsymbol{f}$, we proceed with the following steps.

1. We first show that T is unique.
2. We introduce a set of operators $\Omega^{\boldsymbol{i},\boldsymbol{j}}(T)$, such that $[\![N]\!] = \sum_{\boldsymbol{i},\boldsymbol{j}} \omega_{\boldsymbol{i},\boldsymbol{j}} \Omega^{\boldsymbol{i},\boldsymbol{j}}(T)$. We show the $\omega_{\boldsymbol{i},\boldsymbol{j}}$ to be canonically and uniquely associated with the coefficients of $\boldsymbol{f}$.
3. We introduce a set of operators $\Delta^{\boldsymbol{u},\boldsymbol{v}}(T)$, that have very convenient properties and that we show to be linearly independent.
4. We give an isomorphism between the Ω and Δ operators, therefore proving the linear independence of the $\Omega^{\boldsymbol{i},\boldsymbol{j}}(D)$, and proving the uniqueness of the coefficients of $\boldsymbol{f}$.

□

B. Uniqueness of the $\mathbf{LO}_{\mathsf{PP}}$-circuit part

Proposition III.96. *Let consider a beam splitter followed by a phase shifter:*

$$\left[\!\!\left[\quad \theta \quad \boxed{\varphi} \quad \right]\!\!\right]_1 = \begin{pmatrix} ae^{i\varphi} & be^{i\varphi} \\ b & a \end{pmatrix}$$

where $a = \cos(\theta)$ *and* $b = i\sin(\theta)$*. Let* $k, \ell, n \in \mathbb{N}$*, with* $n >> k, \ell$ *and* $N = n + k - \ell$*. For* $a \neq 0$*, we have:*

$$\langle N, \ell | \left[\!\!\left[\quad \theta \quad \boxed{\varphi} \quad \right]\!\!\right] |n, k\rangle \quad = \quad e^{iN\varphi} a^N \sqrt{\tfrac{N!}{n!}} Q_k^\ell(n)$$

where Q_k^ℓ *is a real polynom of degree* ℓ*.*

Proof. It directly follows from Definition III.77, by noting that $n - N = \ell - k$:

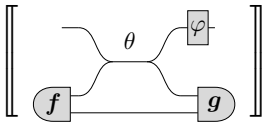
$$\langle N, \ell | \left[\!\!\left[\quad \theta \quad \boxed{\varphi} \quad \right]\!\!\right] |n, k\rangle$$

$$\begin{aligned}
&= e^{iN\varphi} \sqrt{\tfrac{N!\ell!}{n!k!}} \sum_{\substack{p+q=N \\ \delta=p-q}} \tbinom{n}{p} \tbinom{k}{q} a^{k+\delta} b^{n-\delta} \, |\ell_1, \ell_2\rangle \\
&= e^{iN\varphi} \sqrt{\tfrac{N!\ell!}{n!k!}} \sum_{q=0}^{k} \tbinom{n}{N-q} \tbinom{k}{q} a^{k+(N-q)-q} b^{n-((N-q)-q)} \\
&= e^{iN\varphi} \sqrt{\tfrac{N!\ell!}{n!k!}} \sum_{q=0}^{k} \tbinom{n}{n-N+q} \tbinom{k}{q} a^{N+k-2q} b^{n-N+2q} \\
&= e^{iN\varphi} \sqrt{\tfrac{N!\ell!}{n!k!}} \sum_{q=0}^{k} \tbinom{n}{\ell-k+q} \tbinom{k}{q} a^{N+k-2q} b^{(\ell-k)+2q} \\
&= e^{iN\varphi} a^N \sqrt{\tfrac{N!}{n!}} \left(\sqrt{\tfrac{\ell!}{k!}} \sum_{q=0}^{k} \tfrac{n(n-1)\ldots(n-\ell+k-q)}{(\ell-k+q)!} \tbinom{k}{q} a^{k-2q} b^{(\ell-k)+2q} \right) \\
&= e^{iN\varphi} a^N \sqrt{\tfrac{N!}{n!}} Q_k^\ell(n)
\end{aligned}$$

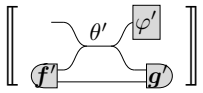
□

Lemma III.97 (Uniqueness of T). *For any two normal forms $N(T, \boldsymbol{f})$ and $N'(T', \boldsymbol{f'})$, if $[\![N]\!] = [\![N']\!]$ then $T = T'$.*

Proof. For any nonzero

$W =$ 

and

$W' =$ 

, we first show that:

$$(\theta, \varphi) \neq (\theta', \varphi') \Rightarrow \exists k \in \mathbb{N}, \lim_{n\to\infty} \frac{\langle n+k| W |n\rangle}{\langle n+k| W' |n\rangle} \neq 1.$$

Meaning that if $W = W'$, then the limit of the ratio is necessarily equal to 1, therefore implying the parameters are equal. We then prove the uniqueness of D by induction on $\min$ (number of inputs, number of outputs).

Base case. Let

$$\left[\!\left[\tilde{\triangle}\right]\!\right]_1 = \left[\!\left[\theta \;\; \varphi \right]\!\right]_1 = \begin{pmatrix} ae^{i\varphi} & be^{i\varphi} \\ b & a \end{pmatrix}$$

and

$$\left[\!\left[\tilde{\triangle}'\right]\!\right]_1 = \left[\!\left[\theta' \;\; \varphi' \right]\!\right]_1 = \begin{pmatrix} a'e^{i\varphi'} & b'e^{i\varphi'} \\ b' & a' \end{pmatrix}$$

with $0 < a \leq 1$ and $0 < a' \leq 1$ as[9] $\theta, \theta' \in (0, \frac{\pi}{2}]$. We have

$$W = \sum_{k,\ell} \omega_{k,\ell} \Omega^{k,\ell}(\tilde{\triangle}) = \sum_{k,\ell} \omega'_{k,\ell} \Omega^{k,\ell}(\tilde{\triangle}') = W'.$$

Without loss of generality, let us consider that $0 < a \leq a' \leq 1$. By hypothesis, there exists one $\omega'_{k,\ell} \neq 0$. Let us consider the maximum ℓ_{max} such that there exists k_m with $\omega_{k_m,\ell_{max}} \neq 0$. We have that for every $n >> \ell_{max}, k$ with $N = k_m - \ell_{max} + n$:

$$\begin{aligned} & & \langle N| W |n\rangle &= \langle N| W' |n\rangle \\ \Leftrightarrow \quad & e^{iN\varphi} a^N \sqrt{\tfrac{N!}{n!}} \left(\sum_{k-\ell=k_m-\ell_{max}} \omega_{k,\ell} Q_k^\ell(n) \right) &= e^{iN\varphi'} a'^N \sqrt{\tfrac{N!}{n!}} \left(\sum_{k-\ell=k_m-\ell_{max}} \omega'_{k,\ell} Q_k'^\ell(n) \right) \end{aligned}$$

where Q_k^ℓ and $Q_k'^\ell$ are real polynomials of degree ℓ (Proposition III.96). Therefore for every such n, N we have:

$$e^{iN(\varphi-\varphi')} \left(\tfrac{a}{a'}\right)^N \left(\sum_{k-\ell=k_m-\ell_{max}} \omega_{k,\ell} Q_k^\ell(n) \right) = \left(\sum_{k-\ell=k_m-\ell_{max}} \omega'_{k,\ell} Q_k'^\ell(n) \right)$$

As $\omega_{k_m,\ell_{max}} \neq 0$, the RHS is a nonzero complex polynomial of degree ℓ_{max} evaluated in n. Note the limit of the absolute value of the RHS is ∞ for $n \to \infty$. Then the LHS has necessarily a nonzero $\omega_{k,\ell}$ and $a' \leq a$, otherwise the limit of the absolute value of the LHS is 0. Therefore, $a = a'$, thus $\cos(\theta) = \cos(\theta')$ and $\theta = \theta'$ as they are in $[0, \frac{\pi}{2}]$.

9. Note the case $a = 0$ is not considered as $\tilde{\triangle}$ or $\tilde{\triangle}'$ would be a phase tensored with the identity, and therefore wouldn't be a $_{\tilde{n}}\triangle_{\tilde{m}}$-circuit.

As $\theta = \theta'$, we have $a = a'$ and $b = b'$, thus $Q_k^\ell = Q_k'^\ell$ and for every $N >> k_m, \ell_{max}$ with $n = \ell_{max} - k_m + N$:

$$\sum_{k-\ell=k_m-\ell_{max}} \left(e^{iN(\varphi-\varphi')}\omega_{k,\ell} - \omega'_{k,\ell}\right) Q_k^\ell(n) \quad = \quad 0$$

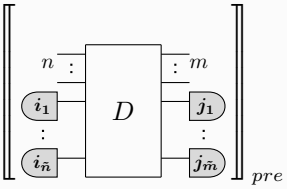
Necessarily, there exists N_0 such that for every $N \geq N_0$, we have $e^{iN(\varphi-\varphi')}\omega_{k_m,\ell_{max}} = \omega'_{k_m,\ell_{max}} \neq 0$. Otherwise, the absolute value of the LHS doesn't converge to 0 and therefore can't be equal to 0 for every N. As it is true for every $N \geq N_0$, is it in particular true for $N = N_0$ and $N = N_0 + 1$, thus $e^{i(\varphi-\varphi')} = 1$, thus $\varphi = \varphi'$.

Induction. For the induction step, we add arbitrary nonzero inputs on the first k (resp. last k') wires and arbitrary nonzero ℓ (resp. last ℓ') outputs on $\tilde{\triangle}$ and $\tilde{\triangle}'$. That projects both sides of the equation on a strictly smaller space. By the induction hypothesis, we can deduce an equality between smaller submatrices, i.e. $\left[\!\left[\tilde{\triangle}\right]\!\right]_1 (k : k', \ell : \ell') = \left[\!\left[\tilde{\triangle}'\right]\!\right]_1 (k : k', \ell : \ell')$. By taking complementary subspaces, we recover that $\left[\!\left[\tilde{\triangle}\right]\!\right]_1 (1 : m, 1 : n) = \left[\!\left[\tilde{\triangle}'\right]\!\right]_1 (1 : m, 1 : n)$. We conclude that $\tilde{\triangle} = \tilde{\triangle}'$ with Proposition III.24. □

C. Ω-morphisms: Canonical basis

Definition III.98 (Ω-morphisms)**.** For any **LO$_{\mathsf{PP}}$**-circuit D, $(\boldsymbol{i}, \boldsymbol{j}) \in (\mathbb{N}^{\tilde{n}}, \mathbb{N}^{\tilde{m}})$ we define $\Omega^{\boldsymbol{i},\boldsymbol{j}}(D) : n \to m$:

$$\Omega^{\boldsymbol{i},\boldsymbol{j}}(D) \coloneqq$$

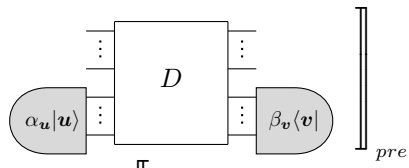


where $[\![\cdot]\!]_{pre} \coloneqq [\![\cdot]\!]\,|_{pre}$ is the restriction of $[\![\cdot]\!]$ to $\mathcal{F}^{\mathsf{pre}}$.

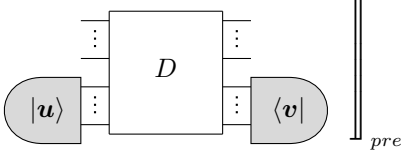
Proposition III.99 (Canonical decomposition)**.** *For every* **LO$_{\boldsymbol{fi}}$***-circuit C, $[\![C]\!]_{pre}$ can be expressed as a linear combination of $\Omega^{\boldsymbol{u},\boldsymbol{v}}(D)$, with D a* **LO$_{\mathsf{PP}}$***-circuit.*

Proof. This is a direct consequence of the linearity of $[\![\cdot]\!]_{pre}$. With $\boldsymbol{f} = \sum_{\boldsymbol{u}\in\mathcal{U}} \alpha_{\boldsymbol{u}} |\boldsymbol{u}\rangle$, $\mathcal{U} \subset_f \mathbb{N}^{\tilde{n}}$ and $\boldsymbol{g} = \sum_{\boldsymbol{v}\in\mathcal{V}} \beta_{\boldsymbol{v}} \langle\boldsymbol{v}|$, $\mathcal{V} \subset_f \mathbb{N}^{\tilde{m}}$, we have:

$$[\![C]\!]_{pre} \quad =$$



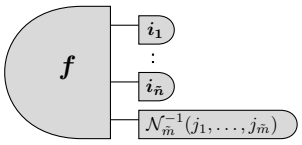
$$= \sum_{\boldsymbol{u}\in\mathcal{U}} \sum_{\boldsymbol{v}\in\mathcal{V}} \alpha_{\boldsymbol{u}}\beta_{\boldsymbol{v}}$$



$$= \sum_{\boldsymbol{u},\boldsymbol{v}} \alpha_{\boldsymbol{u}}\beta_{\boldsymbol{v}}\Omega^{\boldsymbol{u},\boldsymbol{v}}(D)$$
$$= \sum_{\boldsymbol{u},\boldsymbol{v}} \omega_{\boldsymbol{u},\boldsymbol{v}}\Omega^{\boldsymbol{u},\boldsymbol{v}}(D)$$

□

Proposition III.100 (Unique Ω-decomposition of the normal forms)**.** *For any ${}_{\tilde{n}}\triangle_{\tilde{m}}$-circuit $T : n+\tilde{n} \to m+\tilde{m}$ and finite set $\{\omega_{\boldsymbol{u},\boldsymbol{v}}, (\boldsymbol{i},\boldsymbol{j}) \in (\mathbb{N}^{\tilde{n}}, \mathbb{N}^{\tilde{m}})\}$, there exists an unique normal form $N(T,\boldsymbol{f}) : n \to m$, such that $[\![N]\!]_{pre} = \sum_{\boldsymbol{u},\boldsymbol{v}\in(\mathcal{U},\mathcal{V})} \omega_{\boldsymbol{i},\boldsymbol{j}} \Omega^{\boldsymbol{i},\boldsymbol{j}}(T)$.*

Proof. It follows from the linearity of $[\![.]\!]_{pre}$ and that $\omega_{\boldsymbol{i},\boldsymbol{j}} =$ 

. □

Lemma III.101 (Linear independence of $\Omega(\tilde{\triangle})$)**.** *For any ${}_{\tilde{n}}\triangle_{\tilde{m}}$-circuit $\tilde{\triangle} : n+\tilde{n} \to m+\tilde{m}$, if $\sum_{\boldsymbol{u},\boldsymbol{v}} \omega_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\triangle}) = \sum_{\boldsymbol{u},\boldsymbol{v}} \omega'_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\triangle})$ then and $\omega_{\boldsymbol{u},\boldsymbol{v}} = \omega'_{\boldsymbol{u},\boldsymbol{v}}$ for every $\boldsymbol{u},\boldsymbol{v} \in \mathcal{U},\mathcal{V}$.*

The proof of Lemma III.101 is the focus of Section III.3.4.F.. In order to simplify the proofs, we show that the linear independence of $\Omega(\tilde{\triangle})$ with ${}_{\tilde{n}}\triangle_{\tilde{m}}$-circuits is equivalent to a linear independence on ${}_{\tilde{n}}\Diamond_{n}$-circuits introduced in Definition III.25. That is the following proposition.

Proposition III.102 (Linear independence of $\Omega(\tilde{\Diamond})$)**.** *For any ${}_{\tilde{n}}\triangle_{\tilde{m}}$-circuit $\tilde{\triangle} : n'+\tilde{n} \to m'+\tilde{m}$, there exist $n \leq n'$, $m \leq m'$ and a ${}_{\tilde{n}}\Diamond_{n}$-circuit $\tilde{\Diamond} : n+\tilde{n} \to m+\tilde{m}$ such that $\sum_{\boldsymbol{u},\boldsymbol{v}} \omega_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\triangle}) = \sum_{\boldsymbol{u},\boldsymbol{v}} \omega'_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\triangle})$ if and only if $\sum_{\boldsymbol{u},\boldsymbol{v}} \omega_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\Diamond}) = \sum_{\boldsymbol{u},\boldsymbol{v}} \omega'_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\Diamond})$.*

Proof. Following Lemma III.28 by rewriting the shape of $\tilde{\triangle}$ with the rules of **LO$_{\mathbf{PP}}$**, we can show that

$$\left[\!\!\left[\tilde{\triangle}\right]\!\!\right]_{pre} = [\![D']\!]_{pre} \left[\!\!\left[(id^{\otimes n-\tilde{m}}) \otimes \tilde{\Diamond}\right]\!\!\right]_{pre} [\![D]\!]_{pre}$$

. Let $U = D \otimes id^{\otimes \tilde{n}}$, $U' = D' \otimes id^{\otimes \tilde{m}}$ and $I_k = id^{\otimes n-\tilde{m}}$. We have:

$$\begin{array}{rrcl}
 & \sum_{\boldsymbol{u},\boldsymbol{v}} \omega_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\triangle}) & = & \sum_{\boldsymbol{u},\boldsymbol{v}} \omega'_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\triangle}) \\
\Leftrightarrow & \sum_{\boldsymbol{u},\boldsymbol{v}} \omega_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(U' \circ (I_k \otimes \tilde{\Diamond}) \circ U) & = & \sum_{\boldsymbol{u},\boldsymbol{v}} \omega'_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(U' \circ (I_k \otimes \tilde{\Diamond}) \circ U) \\
\Leftrightarrow & \sum_{\boldsymbol{u},\boldsymbol{v}} \omega_{\boldsymbol{u},\boldsymbol{v}} \left([\![D']\!]_{pre} \Omega^{\boldsymbol{u},\boldsymbol{v}}(I_k \otimes \tilde{\Diamond}) [\![D]\!]_{pre} \right) & = & \sum_{\boldsymbol{u},\boldsymbol{v}} \omega'_{\boldsymbol{u},\boldsymbol{v}} \left([\![D']\!]_{pre} \Omega^{\boldsymbol{u},\boldsymbol{v}}(I_k \otimes \tilde{\Diamond}) [\![D]\!]_{pre} \right) \\
\Leftrightarrow & [\![D']\!]_{pre} \left(\sum_{\boldsymbol{u},\boldsymbol{v}} \omega_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(I_k \otimes \tilde{\Diamond}) \right) [\![D]\!]_{pre} & = & [\![D']\!]_{pre} \left(\sum_{\boldsymbol{u},\boldsymbol{v}} \omega'_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(I_k \otimes \tilde{\Diamond}) \right) [\![D]\!]_{pre} \\
\Leftrightarrow & \sum_{\boldsymbol{u},\boldsymbol{v}} \omega_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(I_k \otimes \tilde{\Diamond}) & = & \sum_{\boldsymbol{u},\boldsymbol{v}} \omega'_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(I_k \otimes \tilde{\Diamond}) \\
\Leftrightarrow & I_k \otimes \left(\sum_{\boldsymbol{u},\boldsymbol{v}} \omega_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\Diamond}) \right) & = & I_k \otimes \left(\sum_{\boldsymbol{u},\boldsymbol{v}} \omega'_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\Diamond}) \right) \\
\Leftrightarrow & \sum_{\boldsymbol{u},\boldsymbol{v}} \omega_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\Diamond}) & = & \sum_{\boldsymbol{u},\boldsymbol{v}} \omega'_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\Diamond})
\end{array}$$

□

D. Creation operators

It is standard in the second quantification formalism of quantum physics (Section I.2.3) to look at the evolution of the operators acting on the system rather than directly the evolution of the state. In particular, we associate for each mode i a creation operator $\hat{a}_i^\dagger$ such that $\hat{a}_j^\dagger |x_1, \ldots, x_m\rangle = \sqrt{x_j+1}\, |x_1, \ldots, x_j+1, \ldots, x_m\rangle$. This formalism has shown to be simpler to work with as given a

linear optical circuit D. Indeed, while the state evolutions are given by $[\![D]\!]_{pre}$, the evolution of the creation operators are linear and directly given by $[\![D]\!]_1$ (Theorem I.30). We give here the examples of the evolution of $\hat{a}_j^\dagger$ through a phase shifter and a beam splitter, while the case for a generic circuit D is described in Proposition III.105.

Example III.103 ($\hat{a}_j^\dagger$ through a phase shifter)**.** Let $D = \boxed{\varphi}$ with $[\![D]\!]_1 = e^{i\varphi}$. Through $[\![D]\!]_{pre}$, the creation operator has the following transformation: $\hat{a}_1^\dagger \mapsto e^{i\varphi}\hat{a}_1^\dagger$. We can check it is sound with $P_\varphi^{pre} \coloneqq \left[\!\!\left[\boxed{\varphi} \right]\!\!\right]_{pre}$. By first noticing $|k\rangle = \frac{(\hat{a}_1^\dagger)^k}{\sqrt{k!}}|0\rangle$, we have:

$$P_\varphi^{pre}|k\rangle \;=\; P_\varphi^{pre}\tfrac{(\hat{a}_1^\dagger)^k}{\sqrt{k!}}|0\rangle \;=\; \tfrac{(e^{i\varphi}\hat{a}_1^\dagger)^k}{\sqrt{k!}}P_\varphi^{pre}|0\rangle \;=\; \tfrac{(e^{i\varphi}\hat{a}_1^\dagger)^k}{\sqrt{k!}}|0\rangle \;=\; e^{ik\varphi}|k\rangle$$

Example III.104 ($\hat{a}_j^\dagger$ through a beam splitter)**.** Let $D = \theta$ with $[\![D]\!]_1 = \begin{pmatrix} c_\theta & is_\theta \\ is_\theta & c_\theta \end{pmatrix}$. Through $[\![D]\!]$, the creation operators have the following transformations: $\hat{a}_1^\dagger \mapsto c_\theta\hat{a}_1^\dagger + is_\theta\hat{a}_2^\dagger$ and $\hat{a}_2^\dagger \mapsto is_\theta\hat{a}_1^\dagger + c_\theta\hat{a}_2^\dagger$. We can check it is sound with $\left[\!\!\left[\theta \right]\!\!\right]$ defined in Definition III.77. Indeed, we have:

$$\begin{aligned}
&\left[\!\!\left[\theta \right]\!\!\right]|k_1,k_2\rangle \\
=\;& \left[\!\!\left[\theta \right]\!\!\right]\tfrac{(\hat{a}_1^\dagger)^{k_1}(\hat{a}_2^\dagger)^{k_2}}{\sqrt{k_1!k_2!}}|0,0\rangle \\
=\;& \tfrac{1}{\sqrt{k_1!k_2!}}(c_\theta\hat{a}_1^\dagger + is_\theta\hat{a}_2^\dagger)^{k_1}(is_\theta\hat{a}_1^\dagger + c_\theta\hat{a}_2^\dagger)^{k_2}\left[\!\!\left[\theta \right]\!\!\right]|0,0\rangle \\
=\;& \tfrac{1}{\sqrt{k_1!k_2!}}(c_\theta\hat{a}_1^\dagger + is_\theta\hat{a}_2^\dagger)^{k_1}(is_\theta\hat{a}_1^\dagger + c_\theta\hat{a}_2^\dagger)^{k_2}|0,0\rangle \\
=\;& \tfrac{1}{\sqrt{k_1!k_2!}}\left(\sum_{p=0}^{k_1}\tbinom{k_1}{p}(c_\theta\hat{a}_1^\dagger)^p(is_\theta\hat{a}_2^\dagger)^{k_1-p}\right)\left(\sum_{q=0}^{k_2}\tbinom{k_2}{q}(is_\theta\hat{a}_1^\dagger)^q(c_\theta\hat{a}_2^\dagger)^{k_2-q}\right)|0,0\rangle \\
=\;& \tfrac{1}{\sqrt{k_1!k_2!}}\left(\sum_{\ell_1=0}^{k_1+k_2}\sum_{\substack{p+q=\ell_1\\ \delta=p-q}}\tbinom{k_1}{p}\tbinom{k_2}{q}c_\theta^{k_2+\delta}(is_\theta)^{k_1-\delta}(\hat{a}_1^\dagger)^{\ell_1}(\hat{a}_2^\dagger)^{k_1+k_2-\ell_1}\right)|0,0\rangle \\
=\;& \sum_{\ell_1+\ell_2=k_1+k_2}\sqrt{\tfrac{\ell_1!\ell_2!}{k_1!k_2!}}\sum_{\substack{p+q=\ell_1\\ \delta=p-q}}\tbinom{k_1}{p}\tbinom{k_2}{q}c_\theta^{k_2+\delta}(is_\theta)^{k_1-\delta}|\ell_1,\ell_2\rangle
\end{aligned}$$

Proposition III.105 (General commutation rule with $[\![\cdot]\!]_1$)**.** *For any* **LO$_{\mathbf{PP}}$***-circuit* $D : n+\tilde{n} \to m+\tilde{m}$ *with* $(d_{i,j}) = [\![D]\!]_1$*, we have:*

$$\Lambda^{\boldsymbol{u}}[\![D]\!]_{pre} = \prod_{j=1}^{\tilde{n}}\left(\sum_{i=1}^{n+\tilde{n}} d_{i,j}\,\hat{a}_i^\dagger\right)^{u_j}[\![D]\!]_{pre} = \sum_{\#\boldsymbol{s}+\#\boldsymbol{s'}=\#\boldsymbol{u}}\delta'_{\boldsymbol{s},\boldsymbol{s'}}\left(\Lambda^{\boldsymbol{s}}\otimes\Lambda^{\boldsymbol{s'}}\right)[\![D]\!]_{pre}$$

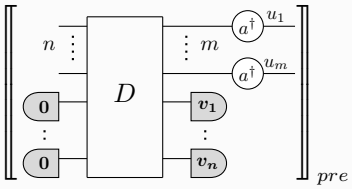
where $\#\boldsymbol{x} \coloneqq \sum_k x_k$ *and* $\Lambda^{\boldsymbol{u}} \coloneqq \prod_{j=1}^{\tilde{n}}(a_j^\dagger)^{u_j}$.

Proof. We can prove $[\![D]\!]_{pre}\,\hat{a}_j^\dagger = (\sum_i d_{i,j}\hat{a}_i^\dagger)\,[\![D]\!]_{pre}$ by commuting each $\hat{a}_i^\dagger$ with all the phase shifters and beam splitters of D with the formulas of Example III.103 and III.104. The first equality follows as the $\hat{a}_j^\dagger$ commute with each other. The second equality is shown by developing and noticing that $(\hat{a}_1^\dagger)^{s_1}\dots(\hat{a}_{\tilde{n}}^\dagger)^{s_{\tilde{n}}}(\hat{a}_{\tilde{n}+1}^\dagger)^{s'_1}\dots(\hat{a}_{\tilde{n}+n}^\dagger)^{s'_n} = \Lambda^{\boldsymbol{s}}\otimes\Lambda^{\boldsymbol{s'}}$. □

E. Δ-morphisms: Ordered basis

Definition III.106 (Δ-morphisms)**.** For any $\mathbf{LO_{PP}}$-circuit D, $(\boldsymbol{u},\boldsymbol{v}) \in (N^m, \mathbb{N}^{\tilde{m}})$ we define $\Delta^{\boldsymbol{u},\boldsymbol{v}}(D) : n \to m$ as:

$$\Delta^{\boldsymbol{u},\boldsymbol{v}}(D) \coloneqq$$



where $\hat{a}^\dagger : |k\rangle \mapsto \sqrt{k+1}\,|k+1\rangle$ is the creation operator and $[\![\cdot]\!]_{pre} \coloneqq [\![\cdot]\!]\,|_{pre}$ is the restriction of $[\![\cdot]\!]$ to $\mathcal{F}^{\mathsf{pre}}$.

Remark III.107. The definitions of the Ω- and Δ-morphisms use the semantics on the pre-Hilbert space $[\![.]\!]_{pre}$. That ensures the soundness of the proofs even though we use unbounded operators $\hat{a}^\dagger$, as all sums are finite.

Let us consider the scalar $z = \langle \boldsymbol{y}|\, \Delta^{\boldsymbol{s},\boldsymbol{t}}(\Diamond)\, |\boldsymbol{x}\rangle$. $\Lambda^{\boldsymbol{s}}$ adds $\boldsymbol{s_j}$ photons on the j^{th} output wire, so we need to detect at least y_j on the j^{th} output mode, otherwise z is zero. Furthermore, $\Omega^{\boldsymbol{0},\boldsymbol{t}}(\Diamond)$ has no ancilla photon in the input, so to detect t_n in the last detector, we need at least $t_{\tilde{n}}$ in the last input mode. If we have exactly t_n in the last input mode, then we need at least t_{n-1} photon in the second to last input mode, to be able to detect t_{n-1} in the second last detector, otherwise $z = 0$. We can therefore prove the following proposition.

Proposition III.108 (Threshold properties of the Δ-morphisms)**.** *For any ${}_{\tilde{n}}\Diamond_n$-circuit $\tilde{\Diamond} : n+m \to n+m$ and $(\boldsymbol{u},\boldsymbol{v}) \in (\mathbb{N}^m, \mathbb{N}^{\tilde{m}})$,*

$$\begin{aligned} &\langle \boldsymbol{y}|\, \Delta^{\boldsymbol{u},\boldsymbol{v}}(\Diamond)\, |\boldsymbol{x}\rangle \neq 0 \quad \textit{for } (\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{u},\boldsymbol{v}) \\ &\langle \boldsymbol{y}|\, \Delta^{\boldsymbol{u},\boldsymbol{v}}(\Diamond)\, |\boldsymbol{x}\rangle = 0 \quad \textit{for } (\boldsymbol{x} \prec_r \boldsymbol{v}) \vee (\boldsymbol{y} \prec_r \boldsymbol{u}) \end{aligned}$$

where $\prec_r$ is the reverse lexicographical order, i.e. $\boldsymbol{y} \prec_r \boldsymbol{v}$ if there exists k such that $y_n = v_n, \ldots, y_{k+1} = v_{k+1}$ and $y_k < v_k$.

Proof. It is a consequence of the shape of the ${}_{\tilde{n}}\Diamond_n$-circuits (Definition III.25), and the properties of $\Delta^{\boldsymbol{u},\boldsymbol{v}}$. As there is no photon in the auxiliary sources, the input needs a certain number of photons for them to be detected in the auxiliary detectors. Similarly, as we create photons at the output with the creation operators $\hat{a}^\dagger$, the output needs a certain number of photons. □

Lemma III.109 (Linear independence of $\Delta^{\boldsymbol{s},\boldsymbol{t}}(\tilde{\Diamond})$)**.** *For any ${}_{\tilde{n}}\Diamond_n$-circuit $\tilde{\Diamond} : n+m \to n+m$, if $\sum_{\boldsymbol{s},\boldsymbol{t}} \delta_{\boldsymbol{s},\boldsymbol{t}} \Delta^{\boldsymbol{s},\boldsymbol{t}}(\tilde{\Diamond}) = \sum_{\boldsymbol{s},\boldsymbol{t}} \delta'_{\boldsymbol{s},\boldsymbol{t}} \Delta^{\boldsymbol{s},\boldsymbol{t}}(\tilde{\Diamond})$ then $\delta_{\boldsymbol{s},\boldsymbol{t}} = \delta'_{\boldsymbol{s},\boldsymbol{t}}$ for every $\boldsymbol{s},\boldsymbol{t} \in (\mathcal{S},\mathcal{T})$.*

Proof. Let $\Delta = \sum_{\boldsymbol{s},\boldsymbol{t}} \delta_{\boldsymbol{s},\boldsymbol{t}} \Delta^{\boldsymbol{s},\boldsymbol{t}}(\tilde{\Diamond})$ and $\Delta' = \sum_{\boldsymbol{s},\boldsymbol{t}} \delta'_{\boldsymbol{s},\boldsymbol{t}} \Delta^{\boldsymbol{s},\boldsymbol{t}}(\tilde{\Diamond})$. By Proposition III.108, we have $\langle \boldsymbol{y}|\, \Delta^{\boldsymbol{x},\boldsymbol{y}}(\tilde{\Diamond})\, |\boldsymbol{x}\rangle \neq 0$. Let us consider the equation $\langle \boldsymbol{y}|\, \Delta\, |\boldsymbol{x}\rangle = \langle \boldsymbol{y}|\, \Delta'\, |\boldsymbol{x}\rangle$. By starting with $|\boldsymbol{x}\rangle = |0\rangle^{\otimes n}$ and $\langle \boldsymbol{y}| = \langle 0|^{\otimes m}$, we deduce with Proposition III.108 that $\delta_{\boldsymbol{0},\boldsymbol{0}} = \delta'_{\boldsymbol{0},\boldsymbol{0}}$. By instancing $\boldsymbol{y}$ with the successive values of $\mathcal{S}$ ordered by $\prec_r$, we deduce that for every $\boldsymbol{s} \in \mathcal{S}$, $\delta_{\boldsymbol{s},\boldsymbol{0}} = \delta'_{\boldsymbol{s},\boldsymbol{0}}$. Then, for each $\boldsymbol{s}$ and by instancing $\boldsymbol{x}$ with the successive values of $\mathcal{T}$ ordered by $\prec_r$, we deduce that for every $\boldsymbol{s},\boldsymbol{t}$, $\delta_{\boldsymbol{s},\boldsymbol{t}} = \delta'_{\boldsymbol{s},\boldsymbol{t}}$. □

F. Uniqueness of the normal forms

We can now make the links between Ω- and Δ-morphisms.

Lemma III.110 (Δ-decomposition of $\Omega^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\Diamond})$)**.** *For any ${}_{\tilde{n}}\Diamond_n$-circuit $\tilde{\Diamond}$, $\boldsymbol{u} \in \mathbb{N}^{\tilde{n}}, \boldsymbol{v} \in \mathbb{N}^n$:*

$$\Omega^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\Diamond}) = \sum_{\substack{\boldsymbol{s} \preceq \boldsymbol{u},\ \boldsymbol{t} \leq \boldsymbol{v} \\ \#\boldsymbol{s} - \#\boldsymbol{t} = \#\boldsymbol{u} - \#\boldsymbol{v}}} \delta_{\boldsymbol{s},\boldsymbol{t}} \Delta^{\boldsymbol{s},\boldsymbol{t}}(\tilde{\Diamond})$$

where $\Delta^{\boldsymbol{s},\boldsymbol{t}}(\tilde{\Diamond}) \coloneqq (\Lambda^{\boldsymbol{s}} \otimes id^n) \circ \Omega^{\boldsymbol{0},\boldsymbol{t}}(\tilde{\Diamond})$, $\delta_{\boldsymbol{u},\boldsymbol{v}} \neq 0$, $\prec$ is the lexicographical order, ie. $\boldsymbol{x} \prec \boldsymbol{u}$ if there exists k such that $x_1 = u_1, \dots, x_{k-1} = u_{k-1}$ and $x_k < u_k$, and $<$ is the weak order on vectors such that $\boldsymbol{y} < \boldsymbol{v}$ if there exists ℓ with $y_\ell < v_\ell$ and $y_j \leq v_j$ for $j \neq \ell$.

Proof. Let us define $\lambda(\boldsymbol{u}) \coloneqq \prod_{j=1}^{\tilde{n}} \sqrt{u_j!}$. First, we commute $\left[\!\left[\tilde{\Diamond}\right]\!\right]_{pre}$ and $\Lambda^{\boldsymbol{u}}$ as follows:

$$\begin{aligned}
\Omega^{\boldsymbol{u},\boldsymbol{v}}(D) &\overset{\text{Def. } III.98\ (\Omega)}{=} (id^{\otimes m} \otimes \langle \boldsymbol{v}|) \left[\!\left[\tilde{\Diamond}\right]\!\right]_{pre} (id^{\otimes n} \otimes |u\rangle) \\
&\overset{\text{Def. } III.105\ (\Lambda)}{=} (id^{\otimes m} \otimes \langle \boldsymbol{v}|) \left[\!\left[\tilde{\Diamond}\right]\!\right]_{pre} (id^{\otimes n} \otimes \lambda(\boldsymbol{u})^{-1} \Lambda^{\boldsymbol{u}} |\boldsymbol{0}\rangle^{\tilde{n}}) \\
&\overset{\text{Prop. } III.105}{=} \sum_{\substack{(\boldsymbol{s},\boldsymbol{s'}) \in (\mathcal{S},\mathcal{S}') \\ \#\boldsymbol{s} + \#\boldsymbol{s'} = \#\boldsymbol{u}}} \lambda(\boldsymbol{u})^{-1} \delta'_{\boldsymbol{s},\boldsymbol{s'}} \left(\Lambda^{\boldsymbol{s}} \otimes \langle \boldsymbol{v}| \Lambda^{\boldsymbol{s'}}\right) \left[\!\left[\tilde{\Diamond}\right]\!\right]_{pre} (id^{\otimes n} \otimes |\boldsymbol{0}\rangle^{\tilde{n}})
\end{aligned}$$

- If $s'_j > v_j$ then $\langle \boldsymbol{v}| \Lambda^{\boldsymbol{s'}} = 0$. Otherwise, we have $\langle \boldsymbol{v}| \Lambda^{\boldsymbol{s'}} \propto \langle \boldsymbol{v} - \boldsymbol{s'}|$. By taking $\boldsymbol{t} = \boldsymbol{v} - \boldsymbol{s'}$, the sum is over the set $\mathcal{T} = \{\boldsymbol{t}, 0 \leq t_1 \leq v_1 \wedge \dots \wedge 0 \leq t_n \leq v_n\} = \{\boldsymbol{t}, \boldsymbol{t} \leq \boldsymbol{v}\}$.
- The elements of $\mathcal{S}$ are determined by the formula in Proposition III.105. As $\tilde{\Diamond}$ is a ${}_{\tilde{n}}\Diamond_n$-circuit, $d_{i,j} = 0$ for $i < j - n$. We can show by developing the product of sums that necessarily $\delta'_{\boldsymbol{s},\boldsymbol{s'}} = 0$ if $s \succ u$ and $\delta'_{\boldsymbol{u},\boldsymbol{v}} \neq 0$. That is illustrated in Example III.111.
- $\#\boldsymbol{s} + \#\boldsymbol{s'} = \#\boldsymbol{u}$. As $\boldsymbol{t} = \boldsymbol{v} - \boldsymbol{s'}$, we have $\#\boldsymbol{t} = \#\boldsymbol{v} - \#\boldsymbol{s'}$ so $\#\boldsymbol{s} - \#\boldsymbol{t} = \#\boldsymbol{u} - \#\boldsymbol{v}$.

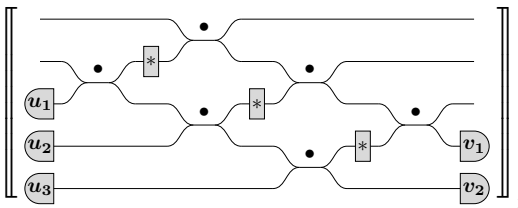
By taking $\delta_{\boldsymbol{s},\boldsymbol{t}} = \lambda(\boldsymbol{u})^{-1} \delta'_{\boldsymbol{s},\boldsymbol{t}}$, we conclude that $\Omega^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\Diamond})$ is equal to:

$$\begin{aligned}
&\sum_{\substack{\boldsymbol{s} \preceq \boldsymbol{u},\ \boldsymbol{t} \leq \boldsymbol{v} \\ \#\boldsymbol{s} - \#\boldsymbol{t} = \#\boldsymbol{u} - \#\boldsymbol{v}}} \delta_{\boldsymbol{s},\boldsymbol{t}} \left(\Lambda^{\boldsymbol{s}} \otimes \langle \boldsymbol{t}|\right) \left[\!\left[\tilde{\Diamond}\right]\!\right]_{pre} (id^{\otimes n} \otimes |\boldsymbol{0}\rangle^{\tilde{n}}) \\
= &\sum_{\substack{\boldsymbol{s} \preceq \boldsymbol{u},\ \boldsymbol{t} \leq \boldsymbol{v} \\ \#\boldsymbol{s} - \#\boldsymbol{t} = \#\boldsymbol{u} - \#\boldsymbol{v}}} \delta_{\boldsymbol{s},\boldsymbol{t}} (\Lambda^{\boldsymbol{u}} \otimes id^n) \circ \Omega^{\boldsymbol{0},\boldsymbol{t}}(\tilde{\Diamond}) \qquad = \sum_{\substack{\boldsymbol{s} \preceq \boldsymbol{u},\ \boldsymbol{t} \leq \boldsymbol{v} \\ \#\boldsymbol{s} - \#\boldsymbol{t} = \#\boldsymbol{u} - \#\boldsymbol{v}}} \delta_{\boldsymbol{s},\boldsymbol{t}} \Delta^{\boldsymbol{s},\boldsymbol{t}}(\tilde{\Diamond})
\end{aligned}$$

□

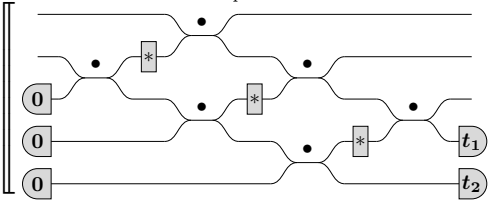
Example III.111 (Illustration)**.** We want to illustrate that $\mathbf{s} \preceq \mathbf{t}$ in the Δ-decomposition of $\Omega^{u,v}(\tilde{\Diamond})$. Let us consider the following example with $n = 2$ and $\tilde{n} = 3$:

$$\Omega^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\Diamond}) = \left[\!\left[\right.\right.$$



$$\left.\left.\right]\!\right]_{pre}$$

$$= \sum_{\boldsymbol{s},\boldsymbol{t}} \delta_{\boldsymbol{s},\boldsymbol{t}} (\hat{a}_1^\dagger)^{s_1} (\hat{a}_2^\dagger)^{s_2} (\hat{a}_3^\dagger)^{s_3} \left[\!\left[\right.\right.$$



$$\left.\left.\right]\!\right]_{pre}$$

$$= \sum_{\boldsymbol{s},\boldsymbol{t}} \delta_{\boldsymbol{s},\boldsymbol{t}} \Delta^{\boldsymbol{s},\boldsymbol{t}}(\tilde{\Diamond})$$

At first, we have $\left[\!\left[\tilde{\Diamond}\right]\!\right]_{pre}(\hat{a}_3^\dagger)^{u_1}(\hat{a}_4^\dagger)^{u_2}(\hat{a}_5^\dagger)^{u_3}$. The only path for the photons to be in the first output wire is by the third input wire. Therefore, we necessarily have $s_1 \leq u_1$. If all the photons went into the first output wire, ie. $s_1 = u_1$, then necessarily we have $s_2 \leq u_2$, as the only paths that could add photons to the second output are the third and fourth input modes. Similarly, if $s_2 = u_2$ then $s_3 \leq u_3$. Therefore, we have $\boldsymbol{s} \preceq \boldsymbol{u}$.

We can now prove the uniqueness of the Ω-decomposition, thereby showing the uniqueness of the coefficients ω and therefore the uniqueness of the normal forms.

Proof of Lemma III.101. Let us consider

$$W = \sum_{\boldsymbol{u},\boldsymbol{v}} \omega_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\Diamond}) \overset{\text{Lem. } III.110}{=} \sum_{\boldsymbol{u},\boldsymbol{v}} \omega_{\boldsymbol{u},\boldsymbol{v}} \left(\sum_{\boldsymbol{s},\boldsymbol{t}} \delta_{\boldsymbol{s},\boldsymbol{t}}^{\boldsymbol{u},\boldsymbol{v}} \Delta^{\boldsymbol{s},\boldsymbol{t}}(\tilde{\Diamond}) \right) = \sum_{\boldsymbol{s},\boldsymbol{t}} \delta_{\boldsymbol{s},\boldsymbol{t}} \Delta^{\boldsymbol{s},\boldsymbol{t}}(\tilde{\Diamond})$$

and

$$W' = \sum_{\boldsymbol{u},\boldsymbol{v}} \omega'_{\boldsymbol{u},\boldsymbol{v}} \Omega^{\boldsymbol{u},\boldsymbol{v}}(\tilde{\Diamond}) \overset{\text{Lem. } III.110}{=} \sum_{\boldsymbol{u},\boldsymbol{v}} \omega'_{\boldsymbol{u},\boldsymbol{v}} \left(\sum_{\boldsymbol{s},\boldsymbol{t}} \delta_{\boldsymbol{s},\boldsymbol{t}}^{\boldsymbol{u},\boldsymbol{v}} \Delta^{\boldsymbol{s},\boldsymbol{t}}(\tilde{\Diamond}) \right) = \sum_{\boldsymbol{s},\boldsymbol{t}} \delta'_{\boldsymbol{s},\boldsymbol{t}} \Delta^{\boldsymbol{s},\boldsymbol{t}}(\tilde{\Diamond})$$

with $W' = W$. We have $\delta_{\boldsymbol{u},\boldsymbol{v}}^{\boldsymbol{u},\boldsymbol{v}} \neq 0$ (Lemma III.110) and $\delta_{\boldsymbol{s},\boldsymbol{t}} = \delta'_{\boldsymbol{s},\boldsymbol{t}}$ (Lemma III.109). By taking $\boldsymbol{u}_M$ the maximum of $\mathcal{U}$ with the total order $\prec$, and $\boldsymbol{v}_m$ a maximal element of $\{\boldsymbol{v}, \omega_{\boldsymbol{u}_M,\boldsymbol{v}} \neq 0\}$ with the partial order $<$, we have

$$\omega_{\boldsymbol{u}_M,\boldsymbol{v}_m} \delta_{\boldsymbol{u}_M,\boldsymbol{v}_m}^{\boldsymbol{u}_M,\boldsymbol{v}_m} = \delta_{\boldsymbol{u}_M,\boldsymbol{v}_m} = \omega'_{\boldsymbol{u}_M,\boldsymbol{v}_m} \delta_{\boldsymbol{u}_M,\boldsymbol{v}_m}^{\boldsymbol{u}_M,\boldsymbol{v}_m}$$

, therefore

$$\omega_{\boldsymbol{u}_M,\boldsymbol{v}_m} = \omega'_{\boldsymbol{u}_M,\boldsymbol{v}_m}$$

. By repeating the same procedure with $W - \omega_{\boldsymbol{u}_M,\boldsymbol{v}_m} \Omega^{\boldsymbol{u}_M,\boldsymbol{v}_m}(\tilde{\Diamond})$ and $W' - \omega'_{\boldsymbol{u}_M,\boldsymbol{v}_m} \Omega^{\boldsymbol{u}_M,\boldsymbol{v}_m}(\tilde{\Diamond})$, we deduce that $\omega_{\boldsymbol{u},\boldsymbol{v}} = \omega'_{\boldsymbol{u},\boldsymbol{v}}$ for every $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{U}, \mathcal{V}$, thus proving Proposition III.102 and Lemma III.101. □

III.3.5 Conclusion

From the rewriting system defined in Definition III.83, we obtain the normal forms explicited in Definition III.91. As all the rules of the rewriting system can be derived with the axioms of the $\mathbf{LO}_{fi}$-calculus (Lemma III.87), it is sufficient to prove the normal forms are unique to prove the completeness of $\mathbf{LO}_{fi}$-calculus.

In order to prove the uniqueness of these normal forms, we first show their semantics are uniquely and directly related to a linear combination with Ω-morphisms (Proposition III.100). The uniqueness of the normal forms is therefore tantamount to the uniqueness of the decomposition with Ω-morphisms.

To show the uniqueness of the Ω-morphism decomposition, we introduce a new family of morphisms: Δ-morphisms (Definition III.106). We show they are linearly independant (Lemma III.109). We conclude at the end of Section III.3.4.F. by showing Δ-morphisms are uniquely related to Ω-morphisms. This shows the uniqueness of the decomposition with Ω-morphisms, thus the uniqueness of the normal forms and the completeness of the $\mathbf{LO}_{fi}$-calculus.

Chapter IV
Applications of the theoretical contributions

While formal graphical frameworks for linear optical circuits are interesting on their own, we still aim for them to have applications in quantum optics and even outreach beyond this field. Similarly, we aim for the simulation algorithms to not only be theoretical, but also accessible through optimized, open-source, and user-friendly software. More broadly, we aim our work to be a key interplay between computer science and physics.

In this chapter, we present how the results introduced throughout this thesis have led to several contributions — with theoretical result in quantum circuits, a computational framework through accessible and optimized software, and practical applications in physics, particularly for mitigating experimental errors. The chapter is structured as follows:

- Section IV.1. We present how the completeness of $\mathbf{LO}_{\mathbf{PP}}$-circuit (Section III.1) allows us to solve a longstanding open problem: the completeness of quantum circuits. This work is detailed in [Clé+23].
- Section IV.2. We introduce Perceval, a software platform developed by Quandela, dedicated to discrete-variable linear optics. It integrates both SLOS (Section II.2) and LO-SLAP (Section II.3) as simulation backends. This work is presented in [Heu+23a].
- Section IV.3. We introduce the problem of mitigating crosstalk, a prominent source of error in photonic integrated circuits. To address this, we define an extended crosstalk model that captures induced phases. We then present an efficient algorithm inspired by the graphical rules of the $\mathbf{LO}_{\mathbf{PP}}$-calculus (Section III.1). We conclude by providing a graph-theoretic criterion to efficiently determine when crosstalk can be fully mitigated. This work is presented in [Fyr+25].

IV.1 Completeness of quantum circuits

First, we present the context of this work and describe what the focus of this thesis is (Section IV.1.1). Then, we introduce the syntax, semantics and multi-controlled gate formalism (Section IV.1.2). Next, we introduce the first complete equational theory of quantum circuits (Section IV.1.3). We conclude by providing a sketch of the proof using a back-and-forth translation with the $\mathbf{LO}_{\mathbf{PP}}$-calculus for proving the completeness of quantum circuits (Section IV.1.4).

Contributions

This section presents [Clé+23], published in LICS'23, and coauthored with Alexandre Clément, Shane Mansfield, Simon Perdrix, and Benoît Valiron.

The original idea of the proof and the most technical parts are due to A. Clément. More details are provided in Section IV.1.1.B..

IV.1.1 Context and scope of this thesis

A. Open problem

Quantum circuit is the ubiquitous formalism to represent logical instructions on quantum devices (Section I.2.5). The wires represent the quantum bits (qubits) and boxes are the quantum gates, i.e. elementary operations performed on the hardware. Mathematically, qubits correspond to vectors in a 2-dimensional Hilbert space, gates to unitary maps and parallel composition to the tensor product (Kronecker product).

However, as crucial quantum circuits are to quantum computing, the graphical language of quantum circuits has never been fully formalized. In particular, a complete equational theory has been a long-standing open problem for decades. It would make it possible to directly prove properties such as circuit equivalence without having to rely on ad-hoc set of equations. Before this work, complete equational theories were only known for non-universal fragments, such as circuits acting on at most two qubits [BS23; CW18], the stabilizer fragment [MRS21; RC14], the CNot-dihedral fragment [ACR18], or fragments of reversible circuits [IKY02; CCS18; CC19].

Interestingly enough, other diagrammatic languages for quantum computation have been developed on sound foundations: it is reasonable to think that this could help in developing a complete equational theory for circuits. Arguably the strongest candidate has been the ZX-calculus [CD08; CD11], [1] equipped with complete equational theories [JPV18; HNW18]. The ZX-calculus shares the same underlying mathematical representation for states: wires corresponds to Hilbert spaces and parallel composition to the tensor operation. Nonetheless, the completeness of the ZX-calculus does not lead a priori to a complete equational theory for quantum circuits. The reason lies in the expressiveness of the ZX-calculus that can represent any linear maps, not only unitaries. If any quantum circuit can be straightforwardly seen as a ZX-diagram, a ZX-diagram does not necessarily represent a unitary map. Remarkably, even extracting a corresponding quantum circuit from a unitary ZX-diagram is known to be a hard task in general [Dun+20; BKW22].

B. In this thesis

Following the collaboration with Alexandre Clément, Shane Mansfield, Simon Perdrix and Benoît Valiron that led to [Clé+22], we made the breakthrough of introducing the first complete equational theory for quantum circuits in [Clé+23]. The proof relies on translating the rules of the $\mathbf{LO}_{\mathbf{PP}}$-calculus (Section III.1.5), a complete graphical language with a semantics on unitaries, to quantum circuits. To understand the content presented in this thesis, we highlight the following points.

First, the initial proof of this work involves a lot of technical lemmas. Indeed, it requires a translation between $\mathbf{LO}_{\mathbf{PP}}$-circuits, that are vertically composed with a $\oplus$, and the quantum circuits, that are vertically composed with the tensor product $\otimes$. Therefore, there is no simple translation, and the back-and-forth encoding between the two languages needs a lot of intermediary steps and derived equations to be fully proven.

Second, even though the idea behind the proofs and the historical first version of the equational theory are still of interest, the presented equational theory is now obsolete. Since the initial result, the equational theory has been simplified in [Clé+24] and even further simplified in [CDP24], where the authors propose both a simple and minimal equational theory.

1. or its variants like ZH [BK19] and ZW [VV24], sharing several similar properties.

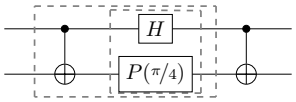
Last but not least, although I contributed to the discussions and the writing of the paper, the main technical developments and original ideas are primarily due to A. Clément.

For all these reasons, we do not detail the complete proof of this work in this thesis. For all the technical details, please see the original paper [Clé+23] or A. Clément's PhD thesis [Clé+23]. Instead, we provide a complete overview of the main ideas and steps that have led to such result.

IV.1.2 Quantum circuits: syntax, semantics and multi-controlled gates

A. Raw circuits

Before providing a formal definition of circuits with the formalism of PROPs, we first use the notion of raw circuits.[2] Given a set of generators, one can generate a raw circuit by means of iterative sequential ($\circ$) and parallel ($\otimes$) compositions. For instance, given the elementary gates $-\boxed{H}-$ and $-\boxed{P(\pi/4)}-$ (with one input and one output) and $\bullet\!\!\oplus$ (with two inputs and two outputs), one can construct the raw circuit $\bullet\!\!\oplus \circ ((-\boxed{H}- \otimes -\boxed{P(\pi/4)}-) \circ \bullet\!\!\oplus)$. Notice that a sequential composition $C' \circ C$ requires that the number of outputs of C matches the number of inputs of C'. This raw circuit can be depicted by gluing the generators together and using boxes to witness how the generators have been composed:



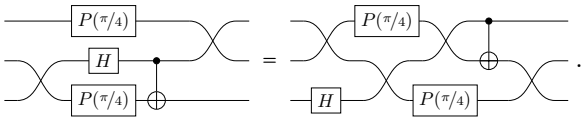
Note that we can recover the notion of PROPs by consists raw circuits up to the rules given in Figure IV.1. More precisely, a PROP generated by a set $\mathcal{G}$ of elementary gates is the collection of raw circuits generated by $\mathcal{G} \cup \{—, \times, \boxed{\ }\}$[3] quotiented by the equations of Figure IV.1.

B. Syntax and semantics

We consider quantum circuits defined on one standard set of generators: Hadamard, Control-Not, and Phase-gates together with global phases.

Definition IV.1. Let **QC** be the PROP generated by $-\boxed{H}-$, $\bullet\!\!\oplus$, and for any $\varphi \in \mathbb{R}$, $-\boxed{P(\varphi)}-$ and $\textcircled{\varphi}$.

The formalism of PROPs (Definition I.74) guarantees that circuits can be depicted graphically without ambiguity and be freely deformed, such as for instance:



The gates $-\boxed{H}-$ and $-\boxed{P(\varphi)}-$ have one input and one output, while $\bullet\!\!\oplus$ has two and $\textcircled{\varphi}$ zero. A quantum circuit C with n inputs and n outputs is called a n-qubit circuit. Given an n-qubit circuit C, the corresponding unitary map $[\![C]\!]_{\mathsf{QC}}$ is acting on the Hilbert space $\mathbb{C}^{\{0,1\}^n} = \mathsf{span}(|x\rangle, x \in \{0,1\}^n)$:[4]

2. Raws terms are for instance similarly used [PS20] as an intermediate step in the definition of PROP.
3. —— denotes the identity, $\times$ the swap and $\boxed{\ }$ the empty circuit.
4. We use the standard Dirac notations.

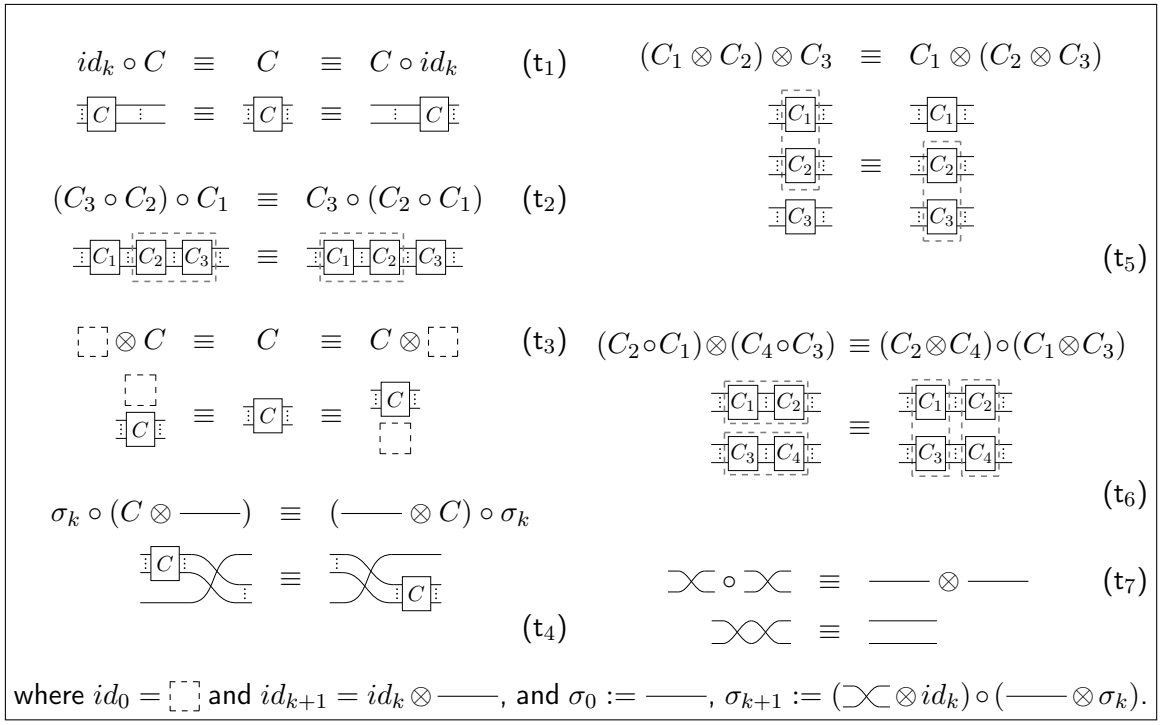


Figure IV.1 – Definition of $\equiv$ for raw circuits (either raw quantum circuits or raw optical circuits).

Definition IV.2 (Semantics)**.** For any n-qubit quantum circuit C, let $[\![C]\!]_{\mathsf{QC}} : \mathbb{C}^{\{0,1\}^n} \to \mathbb{C}^{\{0,1\}^n}$ be the linear map inductively defined as follows: $[\![C_2 \circ C_1]\!]_{\mathsf{QC}} = [\![C_2]\!]_{\mathsf{QC}} \circ [\![C_1]\!]_{\mathsf{QC}}$, $[\![C_1 \otimes C_3]\!]_{\mathsf{QC}} =$

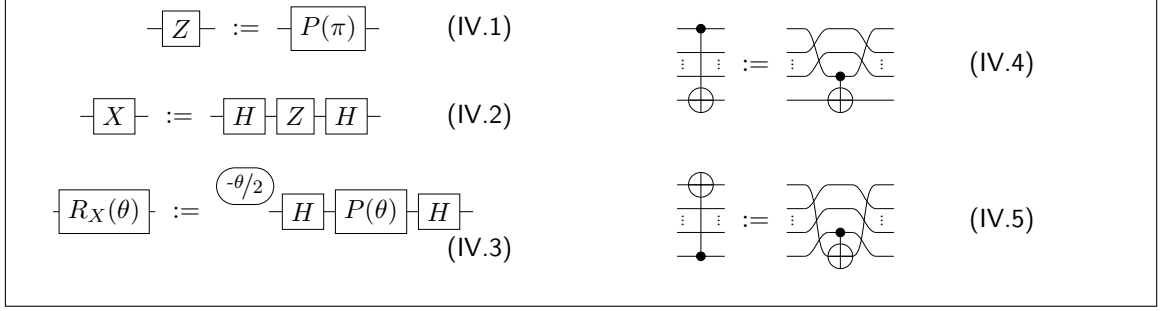


Figure IV.2 – Usual abbreviations of quantum circuits.

$[\![C_1]\!]_{\mathsf{QC}} \otimes [\![C_3]\!]_{\mathsf{QC}}$, and $\forall x, y \in \{0,1\}$, $\forall \varphi \in \mathbb{R}$,

$$
\begin{aligned}
[\![-\boxed{H}-]\!]_{\mathsf{QC}} &= |x\rangle \mapsto \tfrac{1}{\sqrt{2}}(|0\rangle + (-1)^x |1\rangle), \\
[\![-\boxed{P(\varphi)}-]\!]_{\mathsf{QC}} &= |x\rangle \mapsto e^{ix\varphi} |x\rangle, \\
[\![\text{—}]\!]_{\mathsf{QC}} &= |x\rangle \mapsto |x\rangle, \\
[\![\text{●⊕}]\!]_{\mathsf{QC}} &= |x, y\rangle \mapsto |x, x \oplus y\rangle, \\
[\![\text{××}]\!]_{\mathsf{QC}} &= |x, y\rangle \mapsto |y, x\rangle, \\
[\![\text{Ⓟ}]\!]_{\mathsf{QC}} &= 1 \mapsto e^{i\varphi}, \\
[\![\text{▭}]\!]_{\mathsf{QC}} &= 1 \mapsto 1.
\end{aligned}
$$

Proposition IV.3 (Universality [Bar+95]). *For any n-qubit unitary map U acting on $\mathbb{C}^{\{0,1\}^n}$, there exists an n-qubit circuit C such that $[\![C]\!]_{\mathsf{QC}} = U$.* □

We use standard shortcuts in the description of quantum circuits, given in figure IV.2. In textual description, we sometimes use CNot, $s(\varphi)$, X, $P(\varphi)$, *etc* to denote respectively ●⊕, Ⓟ, $-\boxed{X}-$, $-\boxed{P(\varphi)}-$, *etc*. Moreover, when the parameters (e.g. φ) are not specific values they can take arbitrary ones. We write $R_X(\theta)$ for the so-called X-rotation, whereas the standard phase gate $P(\varphi)$ is a Z-rotation only up to a global phase.

C. Multi-controlled gates

Multi-controlled gates are useful to describe more elaborate quantum circuits, and will be essential in the following sections. For controls, we use the notations "λ" and "Λ".

Given a 1-qubit gate G, $\lambda^1 G$ is a 2-qubit positively controlled gate: if the control qubit (the top one) is in state $|1\rangle$ (resp. $|0\rangle$) then G (resp. the identity) is applied on the target qubit (the bottom one). $\lambda^2 G$ is a 3-qubit positively controlled gate, where the two upper qubits are controls: they both need to be in state $|1\rangle$ for the gate G to fire on the bottom qubit. We also consider more general multi-controlled gates $\Lambda^{x_1 \cdots x_k} G$ with positive (when $x_i = 1$) and negative (when $x_i = 0$) controls: if the first qubit is in the state $|x_1\rangle$ (resp. $|\bar{x}_1\rangle$) then $\Lambda^{x_2 \cdots x_k} G$ (resp. the identity) is applied on the remaining qubits. Finally, $\Lambda_y^x G$ denotes a multi-controlled gate with control qubits on both sides – above and below – of the target qubit.

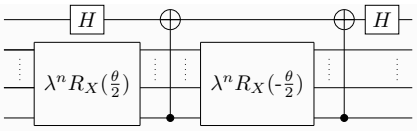
We will follow a standard construction for multi-controls using a decomposition into elementary 1- and 2-qubit gates (see for instance [Bar+95]). Note that we do not aim here at defining all controlled operators. We only focus on the operations $s(\varphi)$, X, $R_X(\theta)$ and $P(\varphi)$. Other controlled operations can then be derived if needed.

We give the following definitions.

Definition IV.4 (Positively multi-controlled gates). For all $n \in \mathbb{N}$ and $G \in \{s(\varphi), X, R_X(\theta), P(\varphi)\}$, we define a quantum circuit $\lambda^n G$. This circuit acts on n wires when $G = s(\varphi)$ and $n+1$ otherwise. We define each circuit $\lambda^n G$ as follows.

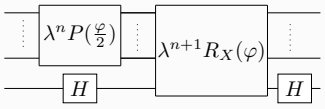
- $\lambda^n R_X(\theta)$ is defined by induction:

$$\lambda^0 R_X(\theta) \coloneqq R_X(\theta),$$

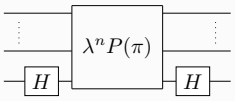
$\lambda^{n+1}R_X(\theta) \coloneqq$ 

- $\lambda^n P(\varphi)$ is defined by induction using $\lambda^n R_X(\varphi)$:

$$\lambda^0 P(\varphi) \coloneqq P(\varphi),$$

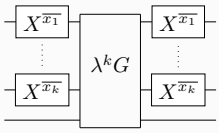
$\lambda^{n+1}P(\varphi) \coloneqq$ 

- $\lambda^n X$ is a simple macro:

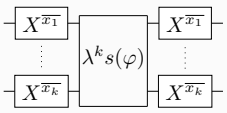
$\lambda^n X \coloneqq$ 

- Finally, $\lambda^0 s(\psi) \coloneqq s(\psi)$ and $\lambda^{n+1} s(\psi) \coloneqq \lambda^n P(\psi)$.

Definition IV.5 (Multi-controlled gates)**.** For any k-length list of booleans $x = x_1, \ldots, x_k$ ($x_i \in \{0,1\}$), for any $G \in \{s(\varphi), X, R_X(\theta), P(\varphi)\}$ we define the quantum circuit $\Lambda^x G$ as

$\Lambda^x G \coloneqq$ 

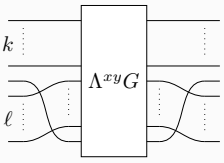
when $G \in \{X, R_X(\theta), P(\varphi)\}$, and

$\Lambda^x s(\varphi) \coloneqq$ 

where $\overline{x} = 1 - x$, $\boxed{X^1} = \boxed{X}$ and $\boxed{X^0} =$ —— .

Definition IV.6 (General multi-controlled gates)**.** Given two lists of booleans $x \in \{0,1\}^k$ and $y \in \{0,1\}^\ell$, if xy is the concatenation of x and y we define the two quantum circuits

- for any $G \in \{X, R_X(\theta), P(\varphi)\}$

$\Lambda^x_y G \coloneqq$ 

- $\Lambda^x_y s(\varphi) \coloneqq \Lambda^{xy} s(\varphi)$.

One can double-check using the semantics that $\Lambda^x_y G$ is actually a multi-controlled gate:

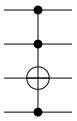
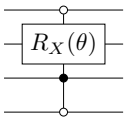
Proposition IV.7. *For any $x, u \in \{0,1\}^k$, $y, v \in \{0,1\}^\ell$, $a \in \{0,1\}$ and $G \in \{X, R_X(\theta), P(\varphi)\}$,*

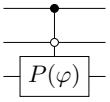
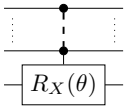
$$[\![\Lambda_y^x G]\!]_{\mathsf{QC}} |u, a, v\rangle = \begin{cases} |u\rangle \otimes ([\![G]\!]_{\mathsf{QC}} |a\rangle) \otimes |v\rangle & \text{if } uv = xy, \\ |u, a, v\rangle & \text{otherwise,} \end{cases}$$

and

$$[\![\Lambda_y^x s(\varphi)]\!]_{\mathsf{QC}} |u, v\rangle = \begin{cases} e^{i\varphi} |u, v\rangle & \text{if } uv = xy, \\ |u, v\rangle & \text{otherwise.} \end{cases}$$

We use the standard bullet-based graphical notation for multi-controlled gates: the i^{th} control is black (resp. white) when $x_i = 1$ (resp. $x_i = 0$), and the j^{th} from the end control is black (resp. white) when $y_{\ell-j+1} = 1$ (resp. $= 0$), e.g.:

$\Lambda_1^{11}X$:  , $\Lambda_{10}^{0}R_X(\theta)$:  ,

$\Lambda^{10}P(\varphi)$:  , $\Lambda^{1\dots1}R_X(\theta)$:  .

To avoid ambiguity with CNot we will not use this notation in the particular case of $\Lambda^1 X$ and $\Lambda_1 X$. Note that $\Lambda^1 X$ and CNot are technically different circuits but are equivalent with the equational theory introduced in Figure IV.3.

IV.1.3 First complete equational theory

A. Structural equation

We introduce in Figure IV.3 a set QC_0 of structural equations on quantum circuits. These equations are structural in the sense that the transformations on the parameters are only based on the fact that $\mathbb{R}$ is an additive group.

Definition IV.8 (QC_0)**.** Let $\mathsf{QC}_0 \vdash \cdot = \cdot$ be the smallest congruence which satisfies equations of Figure IV.1 and IV.3. We write $\mathsf{QC}_0 \vdash C_1 = C_2$ when C_1 can be transformed into C_2 using the equations of figure IV.3 (up to deformations).

Proposition IV.9 (Soundness of QC_0)**.** *For any quantum circuits C_1 and C_2, if $\mathsf{QC}_0 \vdash C_1 = C_2$ then $[\![C_1]\!]_{\mathsf{QC}} = [\![C_2]\!]_{\mathsf{QC}}$.*

Remark IV.10. The equations from (a) to (l) are fairly standard in quantum computing. Equation (m), which is used for instance in [ASD14], describes two equivalent ways to define a controlled-Z gate. Note that this equation cannot be derived from the other axioms as it is the only equation on 2 qubits which does not preserve the parity of the number of CNots plus the number of swaps. Equations (n) and (o) are more involved and account for some specific commutation properties of controlled gates [Clé+23].

Example IV.11 (Derivations from structural equations)**.** The axioms of QC_0 are sufficient to derive standard elementary circuit identities like the ones given in figure IV.4.

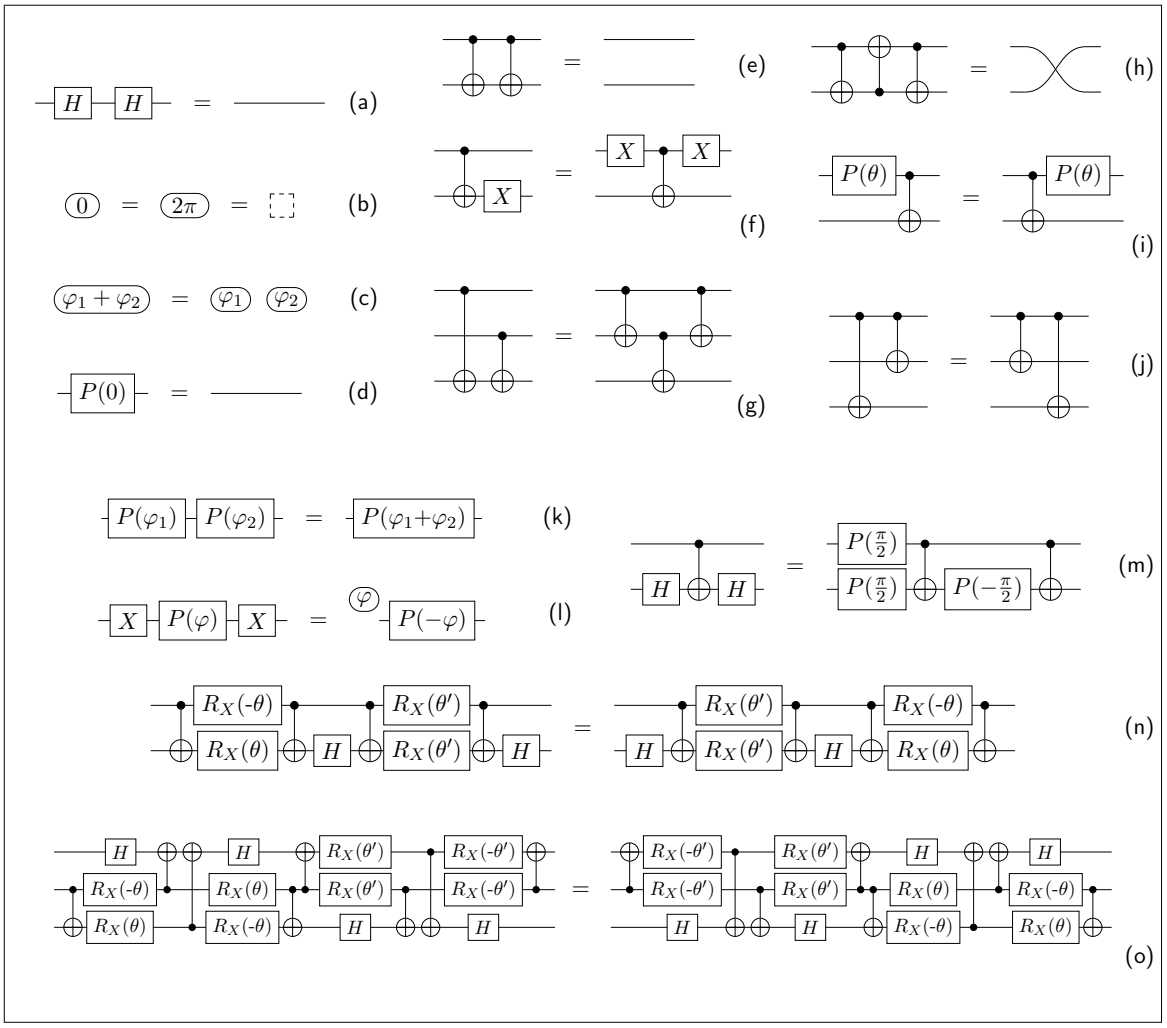


Figure IV.3 – **Axioms of QC$_0$:** Structural equations on quantum circuits. The equations are defined for any $\varphi, \varphi_1, \varphi_2, \theta, \theta' \in \mathbb{R}$.

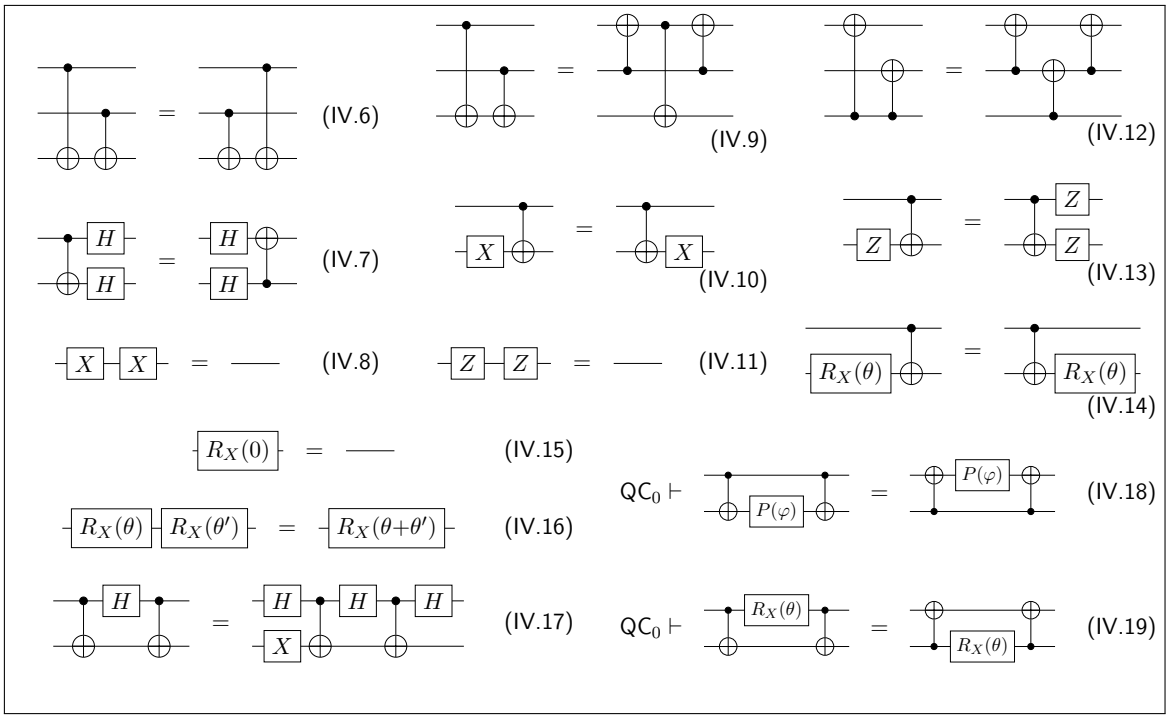
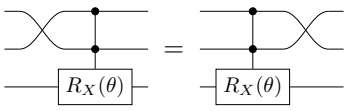


Figure IV.4 – Standard circuit identities that can be derived from the axioms of QC$_0$, given in figure IV.3 .

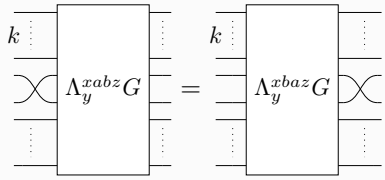
More importantly, we can derive a lot of equations on the multi-controlled gates. For instance, we can derive the following:

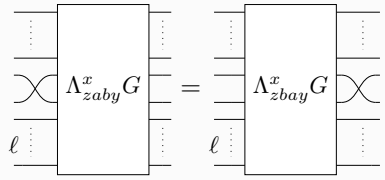


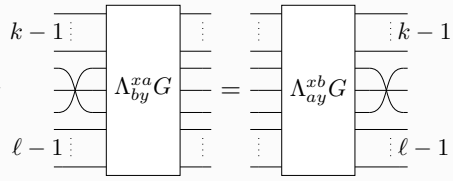
which is indeed sound because control qubits play a similar role.
This can be expressed as the following commuting property:

Proposition IV.12. *For any* $x \in \{0,1\}^k, y \in \{0,1\}^\ell, z \in \{0,1\}^m$, $a, b \in \{0,1\}$ *and any* $G \in$

$\{s(\psi), X, R_X(\theta), P(\varphi)\}$,

$\mathsf{QC}_0 \vdash$  (IV.20)

$\mathsf{QC}_0 \vdash$  (IV.21)

$\mathsf{QC}_0 \vdash$  (IV.22)

Note we can also derive the additive properties of the parametrized elementary quantum gates.

Proposition IV.13. *For any $x \in \{0,1\}^k$, $y \in \{0,1\}^\ell$,*

$$\begin{aligned}
&\mathsf{QC}_0 \vdash \Lambda_y^x R_X(\theta') \circ \Lambda_y^x R_X(\theta) = \Lambda_y^x R_X(\theta + \theta'),\\
&\mathsf{QC}_0 \vdash \Lambda_y^x P(\varphi') \circ \Lambda_y^x P(\varphi) = \Lambda_y^x P(\varphi + \varphi'),\\
&\mathsf{QC}_0 \vdash \Lambda_y^x s(\varphi') \circ \Lambda_y^x s(\varphi) = \Lambda_y^x s(\varphi + \varphi'),\\
&\mathsf{QC}_0 \vdash \Lambda_y^x R_X(0) = id_{k+\ell+1},\\
&\mathsf{QC}_0 \vdash \Lambda_y^x P(0) = id_{k+\ell+1},\\
&\mathsf{QC}_0 \vdash \Lambda_y^x s(0) = id_{k+\ell},
\end{aligned}$$

where id_k is defined as in figure IV.1.

Remark IV.14. Note that the rules of Proposition IV.13 do not imply the periodicity of controlled gates. The latter can only be proven with the help of the rules of Figure IV.5.

Note that more equations can be derived from QC_0 (see [Clé+23] or [Clé23]).

B. Non-structural equations

The equational theory of QC_0 is not complete. Indeed, for instance, equations based on Euler angles, which require non-trivial calculations on the angles, cannot be derived. As a consequence, we add to the equational theory the three rules shown in Figure IV.5, leading to the equational theory QC.

Definition IV.15 (QC)**.** Let $\mathsf{QC} \vdash \cdot = \cdot$ be the smallest congruence which satisfies equations of Figure IV.1, IV.3 and IV.5. We write $\mathsf{QC} \vdash C_1 = C_2$ when C_1 can be rewritten into C_2 using equations of figure IV.3 and figure IV.5 (up to deformations).

Similarly to Euler equations in **LO**$_{\mathsf{PP}}$ (Section III.1.2.B.), we can show that Equations (q) and (r) are sound, leading to the soundness of QC.

5. This is similar to Euler equations in **LO**$_{\mathsf{PP}}$, as described in Section III.1.2.B.. Thus, for any $\alpha_i \in \mathbb{R}$, there exist $\beta_j \in [0, 2\pi)$ such that équation (q) is sound. We make the angles β_j unique by assuming that $\beta_1 \in [0, \pi)$, $\beta_2 \in [0, 2\pi)$ and if $\beta_2 \in \{0, \pi\}$ then $\beta_1 = 0$. équation (p) is the particular Euler decomposition of H. Similarly

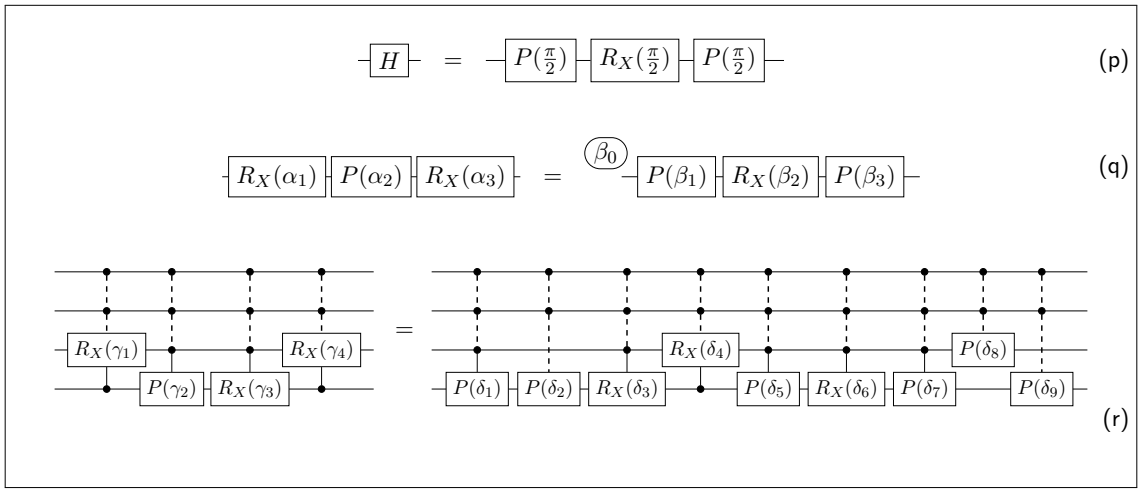
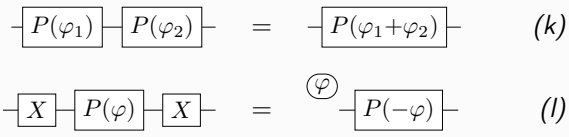


Figure IV.5 – **Non-structural equations.** In Equations (q) and (r) the LHS circuit has arbitrary parameters which uniquely determine the parameters of the RHS circuit [5]. Equation (q) is the Euler-decomposition rule for 2×2 matrix, which states that any unitary can be decomposed, up to a global phase, into basic X- and Z-rotations (Proposition III.10). Equation (r) reads as follows: the equation is defined for any $n \geq 2$ input qubits, in such a way that all gates are controlled by the first $n-2$ qubits. It can be seen as a generalization of the Euler rule, using multi-controlled gates.

Proposition IV.16 (Soundness of QC). *For any quantum circuits C_1 and C_2, if* $\mathsf{QC} \vdash C_1 = C_2$ *then* $[\![C_1]\!]_{\mathsf{QC}} = [\![C_2]\!]_{\mathsf{QC}}$.

The introduction of the additional equations of Figure IV.5 allows us to prove some extra properties about multi-controlled gates, like the periodicity of the parametrized elementary gates and that the multi-controlled X gate is self-inverse.

Proposition IV.17. *The following two equations of* QC,



can be derived from the other axioms of QC.

to équation (q), for any $\gamma_i \in \mathbb{R}$, there exist $\delta_j \in [0, 2\pi)$ such that équation (r) is sound. We can ensure that the angles δ_j are uniquely determined by assuming that $\delta_1, \delta_2, \delta_5, \in [0, \pi)$, $\delta_3, \delta_4, \delta_6 \in [0, 2\pi)$, if $\delta_3 = 0$ then $\delta_2 = 0$, if $\delta_3 = \pi$ then $\delta_1 = 0$, if $\delta_4 = 0$ then $\delta_1 = \delta_3\ (= \delta_2) = 0$, if $\delta_4 = \pi$ then $\delta_2 = 0$, if $\delta_4 = \pi$ and $\delta_3 = 0$ then $\delta_1 = 0$, and if $\delta_6 \in \{0, \pi\}$ then $\delta_5 = 0$.

Proposition IV.18. *For any $x \in \{0,1\}^k$, $y \in \{0,1\}^\ell$, $\theta \in \mathbb{R}$,*

$$\begin{aligned}
&\mathsf{QC} \vdash \Lambda_y^x X \circ \Lambda_y^x X = id_{k+\ell+1} \\
&\mathsf{QC} \vdash \Lambda_y^x R_X(\theta + 4\pi) = \Lambda_y^x R_X(\theta) \\
&\mathsf{QC} \vdash \Lambda_y^x P(\theta + 2\pi) = \Lambda_y^x P(\theta) \\
&\mathsf{QC} \vdash \Lambda_y^x s(\theta + 2\pi) = \Lambda_y^x s(\theta)
\end{aligned}$$

IV.1.4 Completeness: Sketch of the proof

In this section, we develop the key idea for proving the completeness of QC. First, we introduce **LO$_{\mathsf{PP}}$**-circuits, their semantics, and a complete equational theory (Section IV.1.4.A.). Then, we describe the difference in the semantics of **LO$_{\mathsf{PP}}$**-circuits ($\otimes$), with the one of QC ($\otimes$), thus illustrating why the translation needs to be on raw circuits instead (Section IV.1.4.B.). Then we present the encoding, i.e. **LO$_{\mathsf{PP}}$**-circuits to quantum circuits (Section IV.1.4.C.), and the decoding (Section IV.1.4.D.). We conclude by formalizing the core theorems of the proof (Section IV.1.4.E.).

A. LO$_{\mathsf{PP}}$-circuits

LO$_{\mathsf{PP}}$-circuits are linear optical circuits made of beam splitters ( ) and phase shifters ( ). See Section III.1 for a complete description of **LO$_{\mathsf{PP}}$**-circuits. We remind here what is needed to know.

Definition IV.19. Let **LO$_{\mathsf{PP}}$** be the PROP generated by , with $\varphi, \theta \in \mathbb{R}$.

Definition IV.20 (Single-photon semantics of **LO$_{\mathsf{PP}}$**)**.** The single-photon semantics of **LO$_{\mathsf{PP}}$** is inductively defined as follows: $[\![C_1 \otimes C_2]\!]_1 = [\![C_1]\!]_1 \oplus [\![C_2]\!]_1$, $[\![C_2 \circ C_1]\!]_1 = [\![C_2]\!]_1 \circ [\![C_1]\!]_1$ and:

$$\begin{aligned}
[\![\text{—}]\!]_1 &: \mathbb{C} \to \mathbb{C} &&:= |1\rangle \mapsto |1\rangle \\
[\![\text{—}\boxed{\varphi}\text{—}]\!]_1 &: \mathbb{C} \to \mathbb{C} &&:= |1\rangle \mapsto e^{i\varphi}|1\rangle \\
[\![\text{⨉}]\!]_1 &: \mathbb{C}^2 \to \mathbb{C}^2 &&:= \begin{array}{lcl} |1,0\rangle & \mapsto & |0,1\rangle \\ |0,1\rangle & \mapsto & |1,0\rangle \end{array} \\
[\![\theta]\!]_1 &: \mathbb{C}^2 \to \mathbb{C}^2 &&:= \begin{array}{lcl} |1,0\rangle & \mapsto & \cos(\theta)|1,0\rangle + i\sin(\theta)|0,1\rangle \\ |0,1\rangle & \mapsto & i\sin(\theta)|1,0\rangle + \cos(\theta)|0,1\rangle \end{array}
\end{aligned}$$

Remark IV.21. Similarly to quantum circuits, the definition of $[\![\cdot]\!]_1$ relies on the inductive structure of raw LOPP-circuits. It is however well-defined on LOPP-circuits as for any raw LOPP-circuits C, C', $C \equiv C'$ implies $[\![C]\!]_1 = [\![C']\!]_1$.

Definition IV.22 (**LO$_{\mathsf{PP}}$**)**.** Let **LO$_{\mathsf{PP}}$** $\vdash \cdot = \cdot$ be the smallest congruence which satisfies equations of Figure IV.6. We write **LO$_{\mathsf{PP}}$** $\vdash C_1 = C_2$ when C_1 can be transformed into C_2 using the equations of figure IV.6 (up to deformations).

6. This work [Clé+23] uses the axioms from [Clé+22], thereby using the initially introduced axioms of the **LO$_{\mathsf{PP}}$**-calculus. Note that we provided a simplified and minimal complete theory in [Heu25], as described in Section III.3 and depicted in Figure III.1.

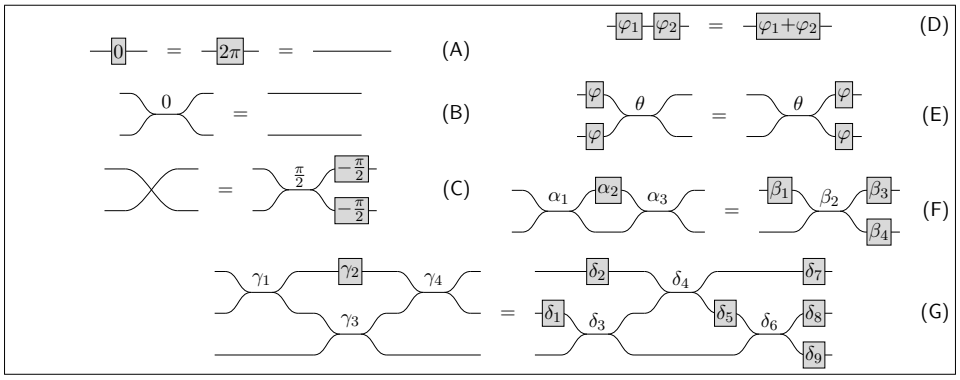


Figure IV.6 – Old axioms of the **LO$_{PP}$**-calculus[6]. In (F) and (G), the LHS circuit has arbitrary parameters which uniquely determine the parameters of the RHS circuit. For any $\alpha_i \in \mathbb{R}$, there exist $\beta_j \in [0, 2\pi)$ such that (F) is sound, and for any $\gamma_i \in \mathbb{R}$, there exist $\delta_j \in [0, 2\pi)$ such that (G) is sound. We can ensure that the angles β_j are unique by assuming that $\beta_1, \beta_2 \in [0, \pi)$ and if $\beta_2 \in \{0, \frac{\pi}{2}\}$ then $\beta_1 = 0$, and we can ensure that the angles δ_j are unique by assuming that $\delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6 \in [0, \pi)$. If $\delta_3 \in \{0, \frac{\pi}{2}\}$ then $\delta_1 = 0$, if $\delta_4 \in \{0, \frac{\pi}{2}\}$ then $\delta_2 = 0$, if $\delta_4 = 0$ then $\delta_3 = 0$, and if $\delta_6 \in \{0, \frac{\pi}{2}\}$ then $\delta_5 = 0$.

Theorem IV.23. *The equational theory given by figure IV.6 is sound and complete: for any* LOPP*-circuits* C_1, C_2, $\mathsf{LOPP} \vdash C_1 = C_2$ *if and only if* $[\![C_1]\!]_{\mathsf{QC}} = [\![C_2]\!]_{\mathsf{QC}}$.

B. Difference of the structures: need for raw circuits

The proof of completeness for quantum circuits is based on a back and forth translation from linear optical circuits. While both kinds of circuits form a prop, so both have a monoidal structure, these monoidal structures do not coincide. The monoidal structure of quantum circuits corresponds to the tensor product, whereas that of linear optical circuits is a direct sum. Hence, the translations do not preserve the monoidal structure.

As a consequence there is a technical issue around defining the translation directly on circuits. We instead define the transformations on raw circuits (cf. section IV.1.2.A.). The collection of raw quantum (resp. **LO$_{PP}$**) circuits is denoted **QC$_{raw}$** (resp. **LOPP$_{raw}$**). Notice that we recover the standard circuits by considering the raw circuits up to the equivalence relation $\equiv$ given in figure IV.1: **QC** = **QC$_{raw}$**$/\equiv$ and **LOPP** = **LOPP$_{raw}$**$/\equiv$.

Note that there exists a derivation between two circuits if and only if there exists a derivation between two of their representative raw circuits. Indeed, intuitively the only difference is that the derivation on raw circuits is more fine-grained as the equivalence relation $\equiv$ is made explicit.

C. Encoding quantum circuits into optical ones

We can now define the encoding of raw quantum circuits into raw linear optical circuits. For being consistent in the dimensions of the involved spaces, an n-qubit system is encoded into 2^n modes. One can naturally choose to encode $|x\rangle$, with $x \in \{0,1\}^n$, into the mode $|\underline{x}\rangle$ where $\underline{x} = \sum_{i=1}^{n} x_i 2^{n-i}$ is the usual binary encoding. Alternatively, we use Gray codes to produce circuits with a simpler connectivity, in particular two adjacent modes encode basis qubit states which differ on exactly one qubit.

Definition IV.24 (Gray code)**.** Let $\mathfrak{G}_n : \mathbb{C}^{2^n} \to \mathbb{C}^{\{0,1\}^n}$ be the map $|k\rangle \mapsto |G_n(k)\rangle$ where $G_n(k)$ is the Gray code of k, inductively defined by $G_0(0) = \epsilon$ and

$$G_n(k) = \begin{cases} 0G_{n-1}(k) & \text{if } k < 2^{n-1}, \\ 1G_{n-1}(2^n - 1 - k) & \text{if } k \geq 2^{n-1}. \end{cases}$$

For instance G_3 is defined as follows:

$$\begin{array}{llll} 0 & \mapsto & 000 & 4 \mapsto 110 \\ 1 & \mapsto & 001 & 5 \mapsto 111 \\ 2 & \mapsto & 011 & 6 \mapsto 101 \\ 3 & \mapsto & 010 & 7 \mapsto 100 \end{array}$$

In order to encode an n-qubit circuit, we proceed by sequentializing it: an n-qubit circuit is seen as a sequential composition of layers, each layer being an n-qubit circuit made of an elementary gate g acting on at most two qubits in parallel with the identity on all other qubits, e.g. $id_k \otimes g \otimes id_l$. The encoding of such a layer, denoted $E_{k,l}(g)$, is a 2^n-mode optical circuit acting non-trivially on potentially all the modes.

Example IV.25. Let us consider a 3-qubit layer which consists in applying $P(\varphi)$ on the second qubit. Its semantics is $|x, y, z\rangle \mapsto e^{i\varphi y} |x, y, z\rangle$. Such a circuit is encoded into an 8-mode optical circuit $E_{1,1}(P(\varphi))$ made of 4 phase shifters acting on the modes $p \in [2, 5]$ (those s.t. $G_3(p) = x1z$). Indeed, the semantics of $E_{1,1}(P(\varphi))$ is $|p\rangle \mapsto \begin{cases} e^{i\varphi} |p\rangle & \text{if } p \in [2, 5] \\ |p\rangle & \text{otherwise} \end{cases}$.

The encoding map is formally defined as follows.

Definition IV.26 (Encoding)**.** Let $E : \mathbf{QC_{raw}} \to \mathbf{LOPP_{raw}}$ be defined as follows: for any n-qubit circuit C, $E(C) = E_{0,0}(C)$ where $E_{k,\ell}$ is inductively defined as:

- $E_{k,\ell}(C_1 \otimes C_2) = E_{k+n_1,\ell}(C_2) \circ E_{k,\ell+n_2}(C_1)$, where C_1 (resp. C_2) is acting on n_1 (resp. n_2) qubits;
- $E_{k,\ell}(C_2 \circ C_1) = E_{k,\ell}(C_2) \circ E_{k,\ell}(C_1)$;

Let us define $\sigma_{k,n,\ell}$ as a $2^{k+n+\ell}$-mode linear optical circuit made only of swaps (that is, without any $-\boxed{\varphi}-$ or $\overset{\theta}{\times}$) such that $\mathfrak{G}_n \circ [\![\sigma_{k,n,\ell}]\!]_{\mathsf{QC}} \circ \mathfrak{G}_n^{-1}(|x, y, z\rangle) = |x, z, y\rangle$ for any $x \in \{0,1\}^k$, $y \in \{0,1\}^n$ and $z \in \{0,1\}^\ell$. We then define

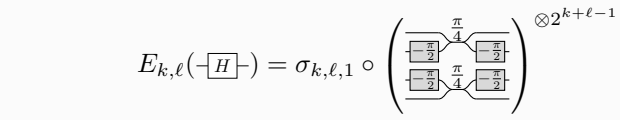
$$\begin{aligned} E_{k,\ell}(\times) &= \sigma_{k,\ell,2} \circ \sigma_{k+\ell,1,1} \circ \sigma_{k,2,\ell}, \\ E_{k,\ell}(\,\vdots\,) &= (\text{——})^{\otimes 2^{k+\ell}}, \\ E_{k,\ell}(\text{——}) &= (\text{——})^{\otimes 2^{k+\ell+1}}, \\ E_{k,\ell}(s(\varphi)) &= (-\boxed{\varphi}-)^{\otimes 2^{k+\ell}}. \end{aligned}$$

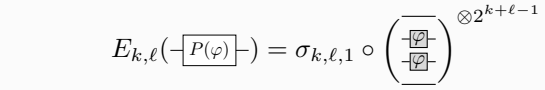
where $C^{\otimes n}$ means C n times in parallel: $C^{\otimes 0} =$ [ ] and $C^{\otimes n+1} = C \otimes C^{\otimes n}$.

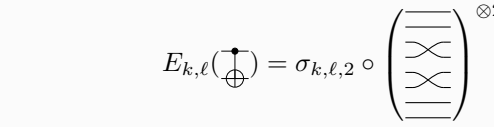
For the remaining generators, we have:

$$\begin{aligned} E_{0,0}(-\boxed{H}-) &= \overset{\frac{\pi}{4}}{\overline{\boxed{-\frac{\pi}{2}}\quad\boxed{-\frac{\pi}{2}}}}, \\ E_{0,0}(-\boxed{P(\varphi)}-) &= \overline{-\boxed{\varphi}-}, \\ E_{0,0}(\bullet\!\!\oplus) &= \overline{\times}, \end{aligned}$$

and whenever $(k,\ell) \neq (0,0)$:

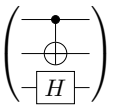
$$E_{k,\ell}(-\boxed{H}-) = \sigma_{k,\ell,1} \circ$$  $$\circ\, \sigma_{k,1,\ell},$$

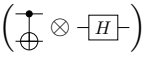
$$E_{k,\ell}(-\boxed{P(\varphi)}-) = \sigma_{k,\ell,1} \circ$$  $$\circ\, \sigma_{k,1,\ell},$$

$$E_{k,\ell}(\text{⊕}) = \sigma_{k,\ell,2} \circ$$  $$\circ\, \sigma_{k,2,\ell}.$$

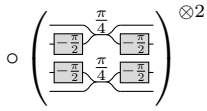
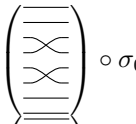
Remark IV.27. Note that for any n-qubit circuit C, $E_{k,\ell}(C)$ is a $2^{k+n+\ell}$-mode optical circuit. Also note that $\sigma_{k,n,\ell}$ is nothing but a permutation of wires. By Lemma IV.37 — which is independent of the definition of E — any actual circuit satisfying the above property ($\mathfrak{G}_n \circ [\![\sigma_{k,n,\ell}]\!]_{\mathsf{QC}} \circ \mathfrak{G}_n^{-1}(|x,y,z\rangle) = |x,z,y\rangle$) is convenient for our purposes. A formal definition of $\sigma_{k,n,\ell}$ is however given in [Clé+23].

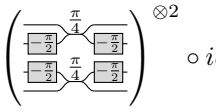
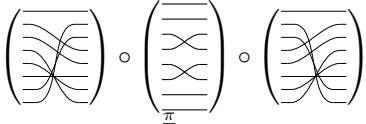
Example IV.28. Using the topological rules, one can simplify the following.

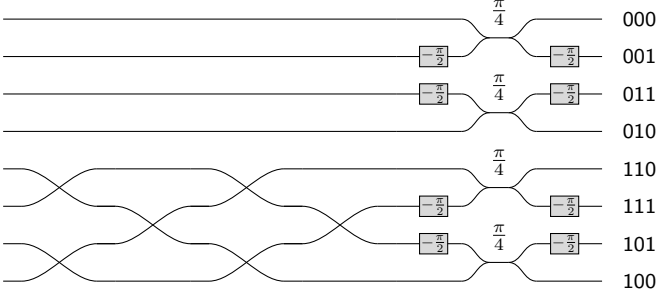
E 

$= E_{0,0}$ 

$= E_{2,0}(-\boxed{H}-) \circ E_{0,1}$

$= \sigma_{2,0,1} \circ$  $\circ\, \sigma_{2,1,0} \circ \sigma_{0,1,2} \circ$  $\circ\, \sigma_{0,2,1}$

$= id_8 \circ$  $\circ\, id_8 \circ$ 

$=$ 

Note the Hadamard gate on the third qubit is indeed performing the transformation.

$$[\![\text{—} \\ \text{—} \\ -\boxed{H}-]\!]_{\mathsf{QC}} = \begin{array}{l} |00\rangle\,|q_0\rangle \mapsto |00\rangle\, H\,|q_0\rangle \\ |01\rangle\,|q_1\rangle \mapsto |01\rangle\, H\,|q_1\rangle \\ |10\rangle\,|q_2\rangle \mapsto |10\rangle\, H\,|q_2\rangle \\ |11\rangle\,|q_3\rangle \mapsto |11\rangle\, H\,|q_3\rangle \end{array}$$

Remark IV.29. Note the Gray code allows a simpler encoding, as the beam splitters are indeed on two neighboring wires.

The encoding of quantum circuits into linear optical circuits preserves the semantics, up to Gray codes.

Proposition IV.30. *For any n-qubit quantum circuit C,*

$$\mathfrak{G}_n \circ [\![E(C)]\!]_{\mathsf{QC}} = [\![C]\!]_{\mathsf{QC}} \circ \mathfrak{G}_n$$

D. Decoding

Regarding the decoding, i.e. the translation back from linear optical circuits to quantum circuits, we use the same sequentialization approach. Note that such a decoding is defined only for optical circuits with a power of two number of modes. Indeed, a m-mode linear optical circuit does not represent a quantum circuit if $m \neq 2^n$. The decoding of a 2^n-mode layer $id_k \otimes g \otimes id_l$ is a n-qubit circuit denoted $D_{k,n}(g)$.

Example IV.31. Let us consider a 16-mode layer which consists in applying $-[\varphi]-$ on the fourth mode. Its semantics is $|p\rangle \mapsto \begin{cases} e^{i\varphi}|p\rangle & \text{if } p = 3 \\ |p\rangle & \text{otherwise} \end{cases}$. Such a circuit is decoded into a 4-qubit circuit $D_{3,4}(-[\varphi]-)$ implementing the multi-controlled phase $\Lambda^{G_4(3)}s(\varphi)$, whose semantics is $|x,y,z,t\rangle \mapsto \begin{cases} e^{i\varphi}|x,y,z,t\rangle & \text{if } xyzt = G_4(3) \\ |x,y,z,t\rangle & \text{otherwise} \end{cases}$.

The decoding map is formally defined as follows.

Definition IV.32 (Decoding)**.** Let $D : \mathbf{LOPP}_{\mathbf{raw}} \to \mathbf{QC}_{\mathbf{raw}}$ be defined as follows: for any 2^n-mode circuit C, $D(C) = D_{0,n}(C)$ where for any n, k, ℓ with $k + \ell \leq 2^n$ and $C : \ell \to \ell$, $D_{k,n}(C)$ is inductively defined as follows.

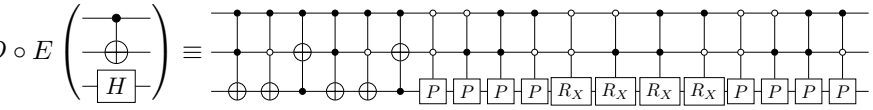
- $D_{k,n}(C_1 \otimes C_2) = D_{k+\ell_1,n}(C_2) \circ D_{k,n}(C_1)$, where C_1 is acting on ℓ_1 modes;
- $D_{k,n}(C_2 \circ C_1) = D_{k,n}(C_2) \circ D_{k,n}(C_1)$;
- $D_{k,n}(——) = id_n$.

The remaining generators are treated as follows.

$$D_{k,n}(\text{⊓}) = id_n, \qquad D_{k,n}(-[\varphi]-) = \Lambda^{G_n(k)}s(\varphi),$$
$$D_{k,n}(\text{⌒×⌒}) = \Lambda^{x_{k,n}}_{y_{k,n}}X, \qquad D_{k,n}(\text{⌒}\theta\text{⌒}) = \Lambda^{x_{k,n}}_{y_{k,n}}R_X(-2\theta),$$

where $x_{2k,n} \coloneqq G_{n-1}(k)$, $y_{2k,n} \coloneqq \epsilon$, $x_{2k+1,n} \coloneqq w$ and $y_{2k+1,n} \coloneqq 1.0^q$, where $q \in \{0, ..., n-2\}$ and $w \in \{0,1\}^{n-q-2}$ are such that $G_n(2k+1) = wa1.0^q$ for some $a \in \{0,1\}$.

Example IV.33. Let us consider the optical circuit obtained in Example IV.28. With all the gates P and R_X parametrized with $\frac{-\pi}{2}$, we can show that:

$D \circ E$  $\equiv$

Similarly to the encoding function, the decoding function preserves the semantics up to Gray codes.

Proposition IV.34. *For any 2^n-mode optical circuit C,*

$$[\![D(C)]\!]_{\mathsf{QC}} \circ \mathfrak{G}_n = \mathfrak{G}_n \circ [\![C]\!]_{\mathsf{QC}}\,.$$

Remark IV.35. We can see that Example IV.28 and IV.33 do not directly lead to the same circuit, and that the decoded circuit is being made of multi-controlled gates. This issue is resolved with Lemma IV.36, showing that we can still recover the original circuit.

E. Completeness

The proof of completeness is based on the encoding and decoding of quantum circuits into optical circuits. Intuitively, given two quantum circuits representing the same unitary map, one can encode them as linear optical circuits. Since the encoding preserves the semantics and **LO$_{\mathbf{PP}}$** is complete, there exists a derivation proving the equivalence of the encoded circuits. In order to lift this proof to quantum circuits, it remains to prove that the decoding of an encoded quantum circuit is provably equivalent to the original quantum circuit, and that each axiom of **LO$_{\mathbf{PP}}$** can be mimicked in QC. Notice that since the encoding/decoding is defined on raw circuits, an extra step in the proof consists in showing that the axioms of $\equiv$ can also be mimicked in QC.

Following the issue raised in Remark IV.35, we show that the equivalence with the initial circuit can always be derived in QC:

Lemma IV.36. *For any n-qubit raw quantum circuit C,*

$$\mathsf{QC} \vdash D(E(C)) = C.$$

Proof. We prove by structural induction on C that

$$\forall k, \ell, \ \mathsf{QC} \vdash D(E_{k,\ell}(C)) = id_k \otimes C \otimes id_\ell.$$

For any two n-qubit raw circuits C_1, C_2, one has

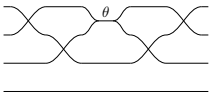
$$D(E_{k,\ell}(C_2 \circ C_1)) = D(E_{k,\ell}(C_2)) \circ D(E_{k,\ell}(C_1))$$

and for any m-qubit raw circuit C_3,

$$D(E_{k,\ell}(C_1 \otimes C_3)) = D(E_{k+n,\ell}(C_3)) \circ D(E_{k,\ell+m}(C_1)).$$

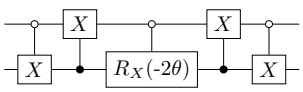
Hence, it remains the basis cases we do not detail here (see [Clé+23]). □

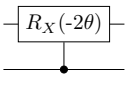
Note that in general, the decoding function does not preserve the topological equivalence. For instance, with the raw circuits

$C_1 =$ 

$C_2 =$

, we have $C_1 \equiv C_2$ but

$D(C_1) =$ 

$D(C_2) =$ 

therefore $D(C_1) \not\equiv D(C_2)$. Thus, the topological rules also have to be mimicked in QC:

Lemma IV.37. *For any 2^n-mode raw optical circuits C_1, C_2, if $C_1 \equiv C_2$ then* $\mathsf{QC} \vdash D(C_1) = D(C_2)$.

Proof. The proof consists intuitively in verifying that the decoding of every equation of figure IV.1 is provable in QC. □

Lemma IV.38. *For any 2^n-mode raw optical circuits C_1, C_2, if* LOPP $\vdash C_1 = C_2$ *then* QC $\vdash D(C_1) = D(C_2)$.

Proof. The proof consists intuitively in verifying that the decoding of every equation of figure IV.6 is provable in QC. □

Theorem IV.39 (Quantum circuit completeness)**.** QC *is a complete equational theory for quantum circuits: for any quantum circuits C_1, C_2, if $[\![C_1]\!]_{\mathsf{QC}} = [\![C_2]\!]_{\mathsf{QC}}$ then* QC $\vdash C_1 = C_2$.

Proof. Given two quantum circuits C_1, C_2 such that $[\![C_1]\!]_{\mathsf{QC}} = [\![C_2]\!]_{\mathsf{QC}}$, let C_1' (resp. C_2') be a raw quantum circuit, representative of C_1 (resp. C_2). Thanks to Proposition IV.30 we have

$$[\![E(C_1')]\!]_{\mathsf{QC}} = [\![E(C_2')]\!]_{\mathsf{QC}}\,.$$

The completeness of $\mathbf{LO}_{\mathsf{PP}}$ implies

$$\mathbf{LO}_{\mathsf{PP}} \vdash E(C_1') = E(C_2').$$

By Lemma IV.38, we have

$$\mathsf{QC} \vdash D(E(C_1')) = D(E(C_2')).$$

Moreover, Lemma IV.36 implies

$$\mathsf{QC} \vdash C_1' = C_2'.$$

From this derivation and up to the equivalence relation $\equiv$ applied on the raw circuits C_1' and C_2', we conclude:

$$\mathsf{QC} \vdash C_1 = C_2.$$

□

IV.2 Perceval: Software platform for discrete-variable linear optical quantum computing

While simulation algorithms are crucial for characterizing physical linear optical circuits, they are not always easy to use for the diverse range of users who may need them. Our goal is to enable users to simulate various experiments and schemes directly from a more physical description rather than relying solely on a mathematical framework. Even more importantly, previous algorithms were specifically designed for optics, lacking a connection to quantum computation, which should be the broader context for such linear optical simulations.

In this section, we introduce Perceval, a comprehensive and efficient software platform for the discrete-variable model of photonic quantum computing. We begin with an overview of what Perceval aims to do and how I personally contributed (Section IV.2.1). Next, we present the main classes and functions that Perceval possesses (Section IV.2.2). Finally, we give a walkthrough tutorial covering most of what Perceval can do (Section IV.2.3).

Contributions

This section presents Perceval, a software described in [Heu+23a], published in Quantum, and coauthored with Andreas Fyrillas, Grégoire de Gliniasty, Raphaë Le Bihan, Sébastien Malherbe, Marceau Pailhas, Eric Bertasi, Boris Bourdoncle, Pierre-Emmanuel Emeriau, Rawad Mezher, Luka Music, Nadia Belabas, Benoît Valiron, Pascale Senellart, Shane Mansfield and Jean Senellart.

My personal contributions are detailed in Section IV.2.1.A. and IV.2.1.C..

IV.2.1 Overview

First, we briefly present how Perceval arose as a continuation of SLOS (Section IV.2.1.A.). Then, we do an overview of the features and core ideas of Perceval (Section IV.2.1.B.). We finish by describing the scope of my contributions on Perceval, leading to introduce the content of Section IV.2.2 and Section IV.2.3 (Section IV.2.1.C.).

A. A few words on the birth of Perceval

In 2021, there was no efficient software for computing all output probabilities at once. Strawberry Fields [Kil+19] was not well-suited for this type of simulation, as it was primarily focused on continuous variables. This limitation led to the development of SLOS (Section II.2), for which I initially coded a Python version. Notably, even in its unoptimized Python implementation, SLOS was still faster than permanent-based methods and Strawberry Fields.

During the second half of 2021, Jean Senellart visited Quandela and took a strong interest in the challenge of efficiently simulating linear optical circuits. Through discussions on the principles of linear optics and inspired by the ideas behind SLOS, Jean, with his deep expertise and software skills, gave birth to what is now the software of Quandela: Perceval.

Since its first public release in March 2022, Perceval has undergone significant changes and improvements, largely thanks to contributions from Quandela's software team, including Aurélien Baert, Eric Bertasi, Marion Fabre, Benoît Fanchon, Raksha Singla, and Mario Valdivia.

B. Global features

Perceval is a modular object-oriented Python code, with optimised functions written in C or C++. Perceval aims to create a linear optical circuit development framework and is based on the following core ideas:

- It is simple to use, both for experimentalists wishing to design photonic experiments, and for computer scientists and theoreticians seeking to develop algorithms and applications for photonic quantum computers.
- It is intended for community development via GitHub [7], the project forum website [8] and an updated documentation [9].
- It does not constrain the user to any framework-specific conventions that are theoretically equivalent (including for instance the flexibility of the conventions for beam splitters, as described in Remark I.33).
- It provides state-of-the-art optimized algorithms with SIMD (Single Instruction, Multiple Data) implementations.
- It provides — when possible and appropriate — access to symbolic calculations for finding analytical solutions.
- It provides companion tools, such as a unitary matrix toolkit, and LaTeX or HTML rendering of circuits.
- It incorporates realistic, parameterizable error and noise modelling.
- It aims to provide a seamless transition from simulators to actual photonic processors (QPU). Since version 0.7, Perceval is able to run samples from real photonic chips through the Cloud [10],

7. https://github.com/Quandela/Perceval
8. https://perceval.quandela.net/forum
9. https://perceval.quandela.net/docs
10. https://cloud.quandela.com/

giving the user access in real-time to the exact hardware characteristics of the photonic source (brightness, purity, indistinguishability), the chip, and the detectors.

C. The focus of this thesis

Perceval was first publicly released in March 2022 with version 0.3.3. The latest version available at the time of writing this thesis is 0.13.0, and the following sections are based on this version. This chapter aims to present Perceval's core functionalities where I have contributed directly (SLOS, LOSLAP, etc) or indirectly (Conventions for components, etc). In particular, I developed a detailed tutorial for LOQCathon 2022 [11], the first hackathon organized by Quandela. The latest version of this tutorial, adapted for Perceval 0.12.2 with the support of Quandela's software team, is presented in Section IV.2.3.

Note that we do not aim to cover all of Perceval's capabilities. For instance, while Perceval supports various quantum experiments and algorithms, I do not include them here as I did not contribute to their implementations. These aspects are detailed in [Heu+23a] and the online documentation.

IV.2.2 Main Classes

A. States

The photonic states are implemented in Perceval by the following two classes.

- `BasicState` is used to describe Fock basis states of n photons over m modes. By default, photons are indistinguishable, but each photon can be annotated, controlling its distinguishability. An annotation is a way to associate additional information to each photon and can thus represent additional degrees of freedom. We can for instance represent different polarizations with different annotation.
- `StateVector` extends `BasicState` to represent superpositions of Fock basis states.

B. Circuits

The `Circuit` class provides a practical way to assemble a linear optical circuit from predefined elementary components, other circuits, or unitary matrices. It can also compute the unitary matrix associated to a given circuit, or conversely decompose a unitary into a linear optical circuit consisting of user-defined components. A library of predefined elementary components is provided (see Table IV.1). Those components can be added on the desired mode(s) on the right of a `Circuit` with the `Circuit.add(modes, component)` method. An example of code using this class can be found in Section IV.2.3.

C. Back-ends

In order to accomplish different tasks efficiently, Perceval possesses different backends. Below, we list all the backends available in version 0.12.2.

- `Naive` (since v0.3.3) is the permanent-based method (see Section I.3.6.A.), suited for single output probabilities with small memory cost. Both Ryser's [Rys63] and Glynn's [Gly10] algorithms have been implemented. Extra care has been taken on the implementation of these algorithms,

11. https://www.quandela.com/about-us/newsroom/loqcathon-2022-challenge-unloqced/
12. Note the matrix for the polarazing beam splitter is different from the one described in Section I.3.3.C.. Indeed, Perceval orders the polarization as $|H\rangle, |V\rangle$ instead of $|V\rangle, |H\rangle$.

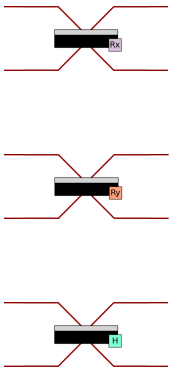
Name	Unitary Matrix	Representation
Beam splitter	R_X convention: $\begin{bmatrix} e^{i(\phi_{tl}+\phi_{tr})}\cos\left(\frac{\theta}{2}\right) & ie^{i(\phi_{bl}+\phi_{tr})}\sin\left(\frac{\theta}{2}\right) \\ ie^{i(\phi_{tl}+\phi_{br})}\sin\left(\frac{\theta}{2}\right) & e^{i(\phi_{bl}+\phi_{br})}\cos\left(\frac{\theta}{2}\right) \end{bmatrix}$ R_Y convention: $\begin{bmatrix} e^{i(\phi_{tl}+\phi_{tr})}\cos\left(\frac{\theta}{2}\right) & -e^{i(\phi_{bl}+\phi_{tr})}\sin\left(\frac{\theta}{2}\right) \\ e^{i(\phi_{tl}+\phi_{br})}\sin\left(\frac{\theta}{2}\right) & e^{i(\phi_{bl}+\phi_{br})}\cos\left(\frac{\theta}{2}\right) \end{bmatrix}$ H convention: $\begin{bmatrix} e^{i(\phi_{tl}+\phi_{tr})}\cos\left(\frac{\theta}{2}\right) & e^{i(\phi_{bl}+\phi_{tr})}\sin\left(\frac{\theta}{2}\right) \\ e^{i(\phi_{tl}+\phi_{br})}\sin\left(\frac{\theta}{2}\right) & -e^{i(\phi_{bl}+\phi_{br})}\cos\left(\frac{\theta}{2}\right) \end{bmatrix}$	
Phase shifter	$\begin{bmatrix} e^{i\phi} \end{bmatrix}$	
SWAP	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	
Wave plate	$\begin{bmatrix} i\sin(\delta)\cos(2\xi)+\cos(\delta) & i\sin(\delta)\sin(2\xi) \\ i\sin(\delta)\sin(2\xi) & -i\sin(\delta)\cos(2\xi)+\cos(\delta) \end{bmatrix}$	
Polarizing beam splitter [12]	$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	
Polarizing rotator	$\begin{bmatrix} \cos(\delta) & \sin(\delta) \\ -\sin(\delta) & \cos(\delta) \end{bmatrix}$	
Time delay		

Table IV.1 – Components available in the `components` library. The extra parameters of the beam splitters allow the user to fix their own preferred convention. Perceval includes a library which allows the user to define their own personal set of components, each with its own visual representation and unitary matrix. By default, all the phases ϕ of the beam splitters are set to zero.

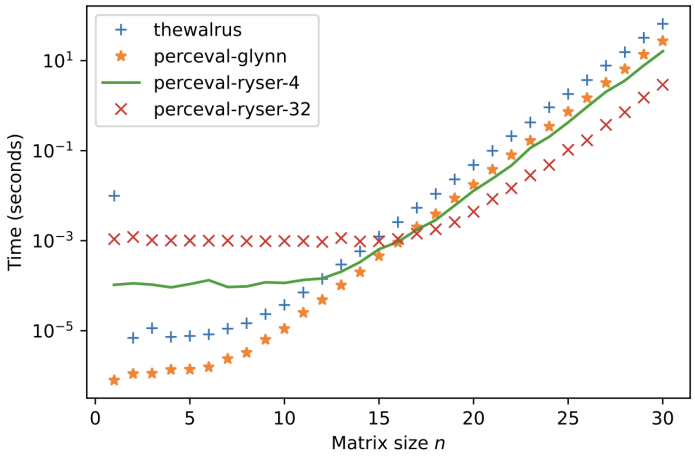


Figure IV.7 – Comparison of the average time to calculate a permanent [13] of an $n \times n$ Haar random matrix. The processor is a 32 core, 3.1GHz Intel Haswell. For *The Walrus*, version 0.19 is used and installed from `pypi`. The Ryser implementation is run on 4 or 32 threads. The Glynn implementation is run on a single thread. Note that all implementations converge to the theoretical performance, but practical factors can create noticeable time differences for the end user. Perceval automatically selects the best algorithm based on the problem size and multi-threading options.

with usage of different optimization techniques including native multithreading and SIMD vectorization primitives. A benchmark of these algorithms against the implementation present in *The Walrus* library [GIQ19] is provided in Figure IV.7.

- `CliffordClifford2017` (since v0.3.3) is the implementation of [CC18], suited for sampling one single output. See the proof of Proposition I.57 for further details on how its works.
- `SLOS` (since v0.3.3) is the algorithm presented in Section II.2. It is particularly suited for exactly computing many output probabilities.
- `Stepper` (since v0.3.3) aims at strongly simulating optical circuits component by component. The complexity is therefore proportional to the number of components. This granularity allows to be more flexible, including losses, delays or nonlinear components.
- `MPS` (since v0.7.0) stands for Matrix Product State. `MPS` is based on a type of tensor network simulation, which gives an approximation of the output states [Sch11; Oh+21]. As `Stepper`, `MPS` backend does the computation on each component of the circuits one-by-one, but this time performs an approximate state evolution. The states are represented by tensors, which are then updated at each component. These tensors can be seen as a big set of matrices, and the approximation is done by choosing the dimension of these matrices, called the bond dimension.
- `NaiveApprox` (since v0.11.0) approximates a single output probabilities by using Gurvit's [Gur05] algorithm to additively approximate a permanent, as described in Section I.3.6.B..
- `SLAP` (since v0.12.0) is the algorithm presented in Section II.3. It is particularly suited for exactly computing many output probabilities and offer different memory trade-off than SLOS, as described in Section II.3.5.

13. Following the methodology presented at `https://the-walrus.readthedocs.io/en/latest/gallery/permanent_tutorial.html`.

D. Processors

The `Processor` class allows the user to emulate a real photonic quantum processor, taking into account the single photon source, the circuit and the detectors. The source model of Perceval allows tuning of the transmittance, multiple photon emission probability, and indistinguishability. By default, the source is perfect. The circuit can be composed of unitary or non-unitary components. Detector imperfections can be emulated — one can switch between number resolving detectors and threshold detectors. Post-selection features can also be added to a `Processor` via heralded modes and/or a final post-selection function.

E. Algorithms

The `algorithm` library provides a set of tasks which can be performed on a `Processor`. These tasks can be as simple as obtaining a sample result, or slightly more complex. For instance `Analyzer` can output a probability table of expected input-output correlations.

F. Bridges to other quantum computing toolkits

In order to facilitate work across multiple platforms, Perceval provides both a catalog of predefined components and specific converters:

- The predefined component catalog, available in `perceval.components.catalog`, allows to access a wide range of quantum logic gates and specific interferometers like MZIs. Each component can be easily accessed with a key, from which we can obtain both a circuit with `build_circuit()` or a processor with `build_processor()`.
- Converters to Perceval from other frameworks are available in `perceval.converters`. Their task is to provide a photonic Perceval equivalent for circuits and algorithms defined in other open-source frameworks. For example, a converter from `qiskit` [Ale+19] is available.

IV.2.3 Walkthrough tutorial with Perceval

In this tutorial, we will try to cover some basic code to get our hands on Perceval.

A. Introduction

1. Perceval's import

```
import perceval as pcvl
import numpy as np

## Use the symbolic skin for display
from perceval.rendering import DisplayConfig, SymbSkin
DisplayConfig.select_skin(SymbSkin)
```

2. BasicStates

In linear optical circuits, photons can have many discrete degrees of freedom, called modes. It can be the frequency, the polarization, the position, or all of them. We represent these degrees of freedom with Fock states. If we have n photons over m modes, the Fock state $|s_1, s_2, \ldots, s_m\rangle$ means we have s_i photons in the i^{th} mode. Note that $\sum_{i=1}^{m} s_i = n$. In Perceval, we will use the module `pcvl.BasicState`.

```
    ## Syntax of different BasicState (list, string, etc)
    bs1 = pcvl.BasicState('|0,2,0,1>')
    bs2 = pcvl.BasicState([0, 2, 0, 1])

    if bs1==bs2:
        print("Those are the same states")

    ## You can iterate on modes
    for i, photon_count in enumerate(bs2):
        print(f"There is {photon_count} photon in mode {i}")
```

Output:

Those are the same states
There is 0 photon in mode 0
There are 2 photons in mode 1
There is 0 photon in mode 2
There is 1 photon in mode 3

3. Linear optical components

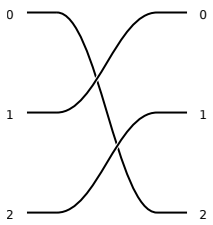
The linear optical components are the elementary blocks which will act on our Fock states. It's important to know all the possible components that can be found in Perceval and understand their effects.

```
    from perceval.components import PS, BS, PERM

    ## Permutation
    perm = PERM([2, 0, 1])

    print(perm.name)
    print(perm.describe())
    pcvl.pdisplay(perm.definition())
    pcvl.pdisplay(perm)
```

Output:

PERM
PERM([2, 0, 1])

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

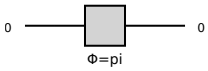


```
## Phase shifter
ps = PS(phi=np.pi)

print(ps.name)
print(ps.describe())
pcvl.pdisplay(ps.definition())
pcvl.pdisplay(ps)  # A pdisplay call on a circuit/processor needs to
    be the last line of a cell
```

Output:

PS
PS(phi=pi)

$$\left[e^{i\phi}\right]$$




```
## Beam splitters
bs_rx = BS.Rx()  # By default, a beam splitter follows the Rx gate
    convention, so bs=BS() has the same matrix

# But other conventions exist too:
bs_h = BS.H()
bs_ry = BS.Ry()

## Check the difference in the unitary definition:
print("BS.Rx() unitary matrix")
pcvl.pdisplay(bs_rx.definition())
print("BS.H() unitary matrix")
pcvl.pdisplay(bs_h.definition())
print("BS.Ry() unitary matrix")
pcvl.pdisplay(bs_ry.definition())
print("BS displays its convention with a small label")
pcvl.pdisplay(bs_ry)
```

Output:

BS.Rx() unitary matrix

$$\begin{bmatrix} e^{i(\phi_{tl}+\phi_{tr})}\cos\left(\frac{\theta}{2}\right) & ie^{i(\phi_{bl}+\phi_{tr})}\sin\left(\frac{\theta}{2}\right) \\ ie^{i(\phi_{br}+\phi_{tl})}\sin\left(\frac{\theta}{2}\right) & e^{i(\phi_{bl}+\phi_{br})}\cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$

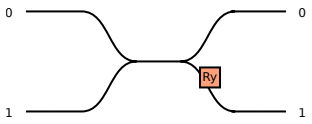
BS.H() unitary matrix

$$\begin{bmatrix} e^{i(\phi_{tl}+\phi_{tr})}\cos\left(\frac{\theta}{2}\right) & e^{i(\phi_{bl}+\phi_{tr})}\sin\left(\frac{\theta}{2}\right) \\ e^{i(\phi_{br}+\phi_{tl})}\sin\left(\frac{\theta}{2}\right) & -e^{i(\phi_{bl}+\phi_{br})}\cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$

BS.Ry() unitary matrix

$$\begin{bmatrix} e^{i(\phi_{tl}+\phi_{tr})}\cos\left(\frac{\theta}{2}\right) & -e^{i(\phi_{bl}+\phi_{tr})}\sin\left(\frac{\theta}{2}\right) \\ e^{i(\phi_{br}+\phi_{tl})}\sin\left(\frac{\theta}{2}\right) & e^{i(\phi_{bl}+\phi_{br})}\cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$

BS displays its convention with a small label




```
# You can ask for the symbolic matrix value of your component with the
    attribute U
my_ps = PS(phi=np.pi/8)
pcvl.pdisplay(my_ps.U)
# And for the numerical value with the method compute_unitary
pcvl.pdisplay(my_ps.compute_unitary())

#  - by using the syntax pcvl.P to create a symbolic variable
#    (note that you cannot compute the numerical value of your
    component anymore)
print("A Phase Shifter with a symbolic value for phi:")
ps = PS(phi=pcvl.P('\psi'))
pcvl.pdisplay(ps.U)
```

Output:

$$\begin{bmatrix} e^{0.3926990816987241i} \end{bmatrix}$$

$$\begin{bmatrix} 0.92388 + 0.382683i \end{bmatrix}$$

A Phase Shifter with a symbolic value for phi:

$$\begin{bmatrix} e^{i\psi} \end{bmatrix}$$

```
# If you do it for a Beam-Splitter, you can see that by default
    theta=pi/2, and the phi's are 0
print("A default beam-splitter:")
pcvl.pdisplay(BS().compute_unitary())  # this is a balanced
    Beamsplitter
```

```
    print("")

    # To control the value of the parameters of a component, several
        choices are possible:
    #  - by setting a numerical value during the creation of the component
    print("A Beam-Splitter with a numerical value for theta:")
    bs_rx = BS.Rx(theta=10)
    pcvl.pdisplay(bs_rx.U)
    pcvl.pdisplay(bs_rx.compute_unitary())
    print("")

    #  - by using the syntax pcvl.P to create a symbolic variable
    #    (note that you cannot compute the numerical value of your
        component anymore)
    print("A Phase Shifter with a symbolic value for phi:")
    ps = PS(phi=pcvl.P('\psi'))
    pcvl.pdisplay(ps.U)
    print("")

    #  - you can still modify the value of a symbolic variable after its
        creation
    #    This is not true for a numerical variable!
    print("A beam-splitter with a symbolic variable...")
    bs_rx = BS(theta=pcvl.P('toto'))
    pcvl.pdisplay(bs_rx.U)
    bs_rx.assign({'toto':5})
    bs_rx.assign({'toto':10})
    print("... set to a numerical value")
    pcvl.pdisplay(bs_rx.compute_unitary())
```

Output:

A default beam-splitter:

$$\begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}i}{2} \\ \frac{\sqrt{2}i}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$$

A Beam-Splitter with a numerical value for theta:

$$\begin{bmatrix} \cos(5) & i\sin(5) \\ i\sin(5) & \cos(5) \end{bmatrix}$$

$$\begin{bmatrix} 0.283662 & -0.958924i \\ -0.958924i & 0.283662 \end{bmatrix}$$

A Phase Shifter with a symbolic value for phi:

$$\begin{bmatrix} e^{i\psi} \end{bmatrix}$$

A beam-splitter with a symbolic variable...

$$\begin{bmatrix} \cos\left(\frac{toto}{2}\right) & i\sin\left(\frac{toto}{2}\right) \\ i\sin\left(\frac{toto}{2}\right) & \cos\left(\frac{toto}{2}\right) \end{bmatrix}$$

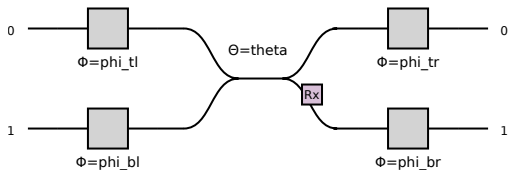
... set to a numerical value

$$\begin{bmatrix} 0.283662 & -0.958924i \\ -0.958924i & 0.283662 \end{bmatrix}$$

```
## to understand the conventions, you can note that a BS.Rx with the 4
    phases phi (top left/right and bottom left/right) can be
    represented like that

bs_rx_circuit=pcvl.Circuit(2) // (0, PS(phi=pcvl.P("phi_tl"))) // (1,
    PS(phi=pcvl.P("phi_bl"))) // BS(theta=pcvl.P('theta')) // (0,
    PS(phi=pcvl.P("phi_tr"))) // (1, PS(phi=pcvl.P("phi_br")))

pcvl.pdisplay(bs_rx_circuit.U)

# we can check it's the same as bs_rx.definition()
pcvl.pdisplay(bs_rx_circuit)

## For this cell, we needed the syntax to builds circuits... Good
    transition !
```

Output:

$$\begin{bmatrix} e^{i(\phi_{tl}+\phi_{tr})}\cos\left(\frac{\theta}{2}\right) & ie^{i(\phi_{bl}+\phi_{tr})}\sin\left(\frac{\theta}{2}\right) \\ ie^{i(\phi_{br}+\phi_{tl})}\sin\left(\frac{\theta}{2}\right) & e^{i(\phi_{bl}+\phi_{br})}\cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$



B. Linear optical circuits

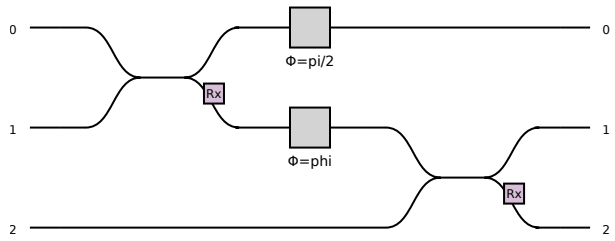
From the linear optical elementary components, we can build a linear optical circuit, i.e. a sequence of these components acting on different modes.

1. Syntax

```
circuit = pcvl.Circuit(3)  # Create an empty 3 mode circuit

circuit.add(0, BS())  # The beam splitter is added to the circuit on
    mode 0 and 1
                      # even though only the first mode is required in
                          'add' method
circuit.add(0, PS(phi=np.pi/2)).add(1, PS(phi=pcvl.P('phi'))).add(1,
    BS())

# Equivalent syntax:
# circuit // BS() // PS(phi=np.pi/2) // (1, PS(phi=pcvl.P('phi'))) //
    (1, BS())

pcvl.pdisplay(circuit.U)
pcvl.pdisplay(circuit)
```

Output:

$$\begin{bmatrix} \frac{\sqrt{2}e^{1.5707963267949i}}{2} & \frac{\sqrt{2}ie^{1.5707963267949i}}{2} & 0 \\ \frac{ie^{i\phi}}{2} & \frac{e^{i\phi}}{2} & \frac{\sqrt{2}i}{2} \\ -\frac{e^{i\phi}}{2} & \frac{ie^{i\phi}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$$

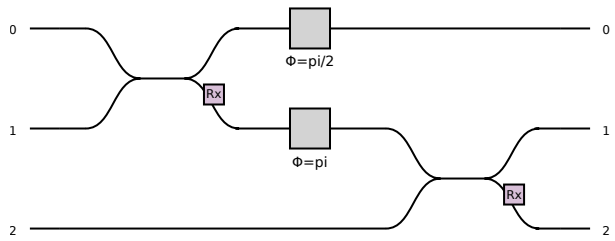


The syntax `pcvl.P('phi')` allows you to use parameters in the circuit, where you can assign a value or not. The behavior of the parameters of a circuit is similar to the case of the components. For instance, you can use:

```
params=circuit.get_parameters()
print(params) # list of the parameters

# the value is not set, but we can change that with:
params[0].set_value(np.pi)
pcvl.pdisplay(circuit)
```

Output:

[Parameter(name='phi', value=None, min_v=0.0, max_v=6.283185307179586)]



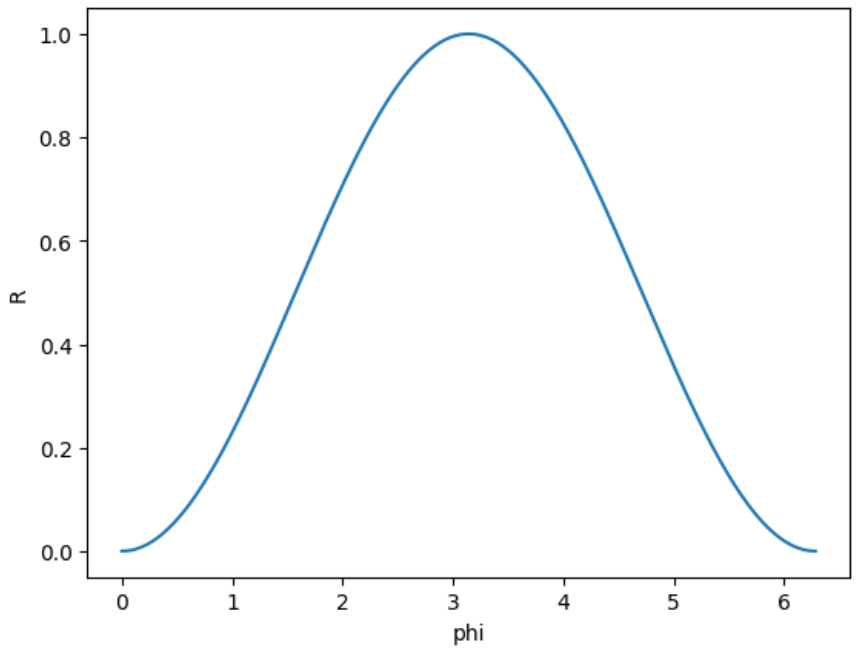
2. Mach-Zehnder interferometers

The beam splitter's angle θ can also theoretically be defined as a parameter. However, it is not practical to have such parameterization on the physical component. Therefore, in practice, we use a Mach-Zehnder interferometer (see Definition I.35). The beam splitter with a parameter θ is implemented with a parameterised phase shifter ϕ between two fixed balanced beam splitters.

```
## Exercise: build a circuit implementing the mzi

```

```
    ## Solution:
    mzi = pcvl.Circuit(2) // BS() // (1,PS(phi=pcvl.P("phi"))) // BS()
    pcvl.pdisplay(mzi)

    ## Exercise: Check that the parameterised phase allows you to change
        the reflexivity of your MZI


    ## Solution:
    import matplotlib.pyplot as plt

    X = np.linspace(0, 2*np.pi, 1000)  # We create a list of all different
        values for theta
    Y = []
    for theta in X:
        phase = mzi.get_parameters()[0]
        phase.set_value(theta)
        Y.append(abs(mzi.compute_unitary()[0,0])**2)  #compute_unitary is
            numerical, so far faster that mzi.U, which uses symbolic
            expressions.

    plt.plot(X, Y)
    plt.xlabel("phi")
    plt.ylabel("R")
    plt.show()

    ## Note: If you need to create a BS directly from the reflexivity
        value, please use:
    ## BS(BS.r_to_theta(reflectivity_value))
    ## However, be aware that only theta value is stored inside the BS
        object
```

Output:



3. Universal circuits

A linear optical circuit can be expressed as a unitary on the modes. For three modes, the unitary

$$U = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix}$$

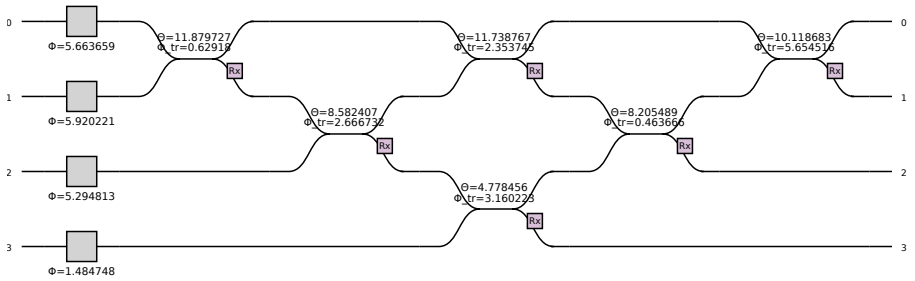
performs the following operation on the Fock state basis:

$$\begin{aligned} |1,0,0\rangle &\mapsto a_{1,1}|1,0,0\rangle + a_{1,2}|0,1,0\rangle + a_{1,3}|0,0,1\rangle \\ |0,1,0\rangle &\mapsto a_{2,1}|1,0,0\rangle + a_{2,2}|0,1,0\rangle + a_{2,3}|0,0,1\rangle \\ |0,0,1\rangle &\mapsto a_{3,1}|1,0,0\rangle + a_{3,2}|0,1,0\rangle + a_{3,3}|0,0,1\rangle \end{aligned}$$

Since 1994, we know that any U on the modes can be implemented with a linear optical circuit [Rec+94]. This decomposition can be done easily in Perceval using beam splitters and phase-shifters as follows.

```
## From any unitary
n = 4
U = pcvl.Matrix.random_unitary(n)

decomposed_circuit = pcvl.Circuit.decomposition(U,
    BS(theta=pcvl.P('theta'), phi_tr=pcvl.P('phi')),
    phase_shifter_fn=PS)
pcvl.pdisplay(decomposed_circuit)
```

Output:




```
print("The error between the two unitaries is", np.linalg.norm(U -
    decomposed_circuit.compute_unitary()))
```

Output:

The error between the two unitaries is 1.1611589802526527e-08

```
## Exercise: decompose your unitary with only phase shifters and
    balanced beamsplitters.

## Solution:
mzi = pcvl.Circuit(2) // BS() // PS(pcvl.P("phi1")) //  BS()  //
    PS(pcvl.P("phi2"))

circuit_u = pcvl.Circuit.decomposition(U,mzi,phase_shifter_fn=PS)

## Note: you can use a MZI. Be careful to put the phase on the right,
    as the full layer of phase_shifter_fn is on the left of the circuit
```

```
## Exercise: check the norm of the difference to be sure it has worked
    well

## Solution:
print("The error between the two unitaries is", np.linalg.norm(U -
    circuit_u.compute_unitary()))
```

Output:

The error between the two unitaries is 1.3800231171497524e-08

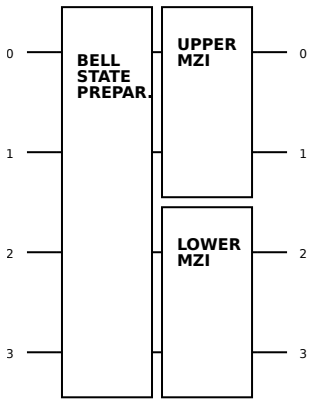
C. 4. Black Box

To improve readability, the circuit can be constructed in multiple steps which are then combined as black boxes. This will also help when we will need generic operations.

```
pre_MZI = (pcvl.Circuit(4, name="Bell State Prepar.")
           .add(0, BS())
           .add(2, BS())
           .add(1, PERM([1, 0])))
```

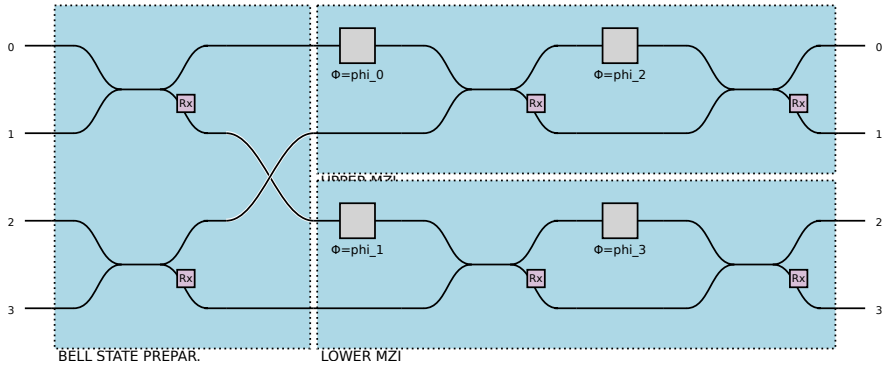
```
5
6     upper_MZI = (pcvl.Circuit(2, name="upper MZI")
7                  .add(0, PS(phi=pcvl.P('phi_0')))
8                  .add(0, BS())
9                  .add(0, PS(phi=pcvl.P('phi_2')))
10                 .add(0, BS()))
11
12    lower_MZI = (pcvl.Circuit(2, name="lower MZI")
13                 .add(0, PS(phi=pcvl.P('phi_1')))
14                 .add(0, BS())
15                 .add(0, PS(phi=pcvl.P('phi_3')))
16                 .add(0, BS()))
17
18    chip = (pcvl.Circuit(4)
19                  .add(0, pre_MZI)
20                  .add(0, upper_MZI, merge=False)
21                  .add(2, lower_MZI, merge=False))
22
23    pcvl.pdisplay(chip)
```

Output:




```
1     ## You can still display the inside of black boxes with:
2     pcvl.pdisplay(chip, recursive=True)
```

Output:



D. Simulation

Up to this point, we have focused on creating circuits. It's time to learn how to sample from them or describe their output distribution on many inputs.

1. Computing probabilities

For this part, we will take the Hong-Ou-Mandel (see Example I.34) experience as an example. It's one of the simplest experiments and yet it is very useful.

Making two indistinguishable photons, one in each mode, enter one balanced beam splitter

$$BS = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

we expect the outcome to be:

$$|1,1\rangle \mapsto \frac{|2,0\rangle - |0,2\rangle}{\sqrt{2}}$$

We will show how to verify this in the next steps using the `Naive` backend to recover the full probability distribution.

```
## Exercise: build the circuit using the convention above

## Solution:
circuit = pcvl.Circuit(2) // BS.H()
```

```
# Syntax to compute the amplitudes
backend = pcvl.BackendFactory.get_backend("SLOS")
backend.set_circuit(circuit)
backend.set_input_state(pcvl.BasicState([1, 1]))
print(backend.prob_amplitude(pcvl.BasicState([2, 0])))  # note that
    it's the amplitude !
print(backend.prob_amplitude(pcvl.BasicState([0, 2])))
print(backend.probability(pcvl.BasicState([0, 2])))


## We can also use the Analyser module to compute a table of
    probabilities
```

```
    ## The Analyser uses a Processor to work with. A Processor aims at
        simulating a photonic source plugged into a circuit
    ## with a given backend.
    ## The main syntax is :
    ## >>> p = pcvl.Processor(backend_name, circuit, noise=NoiseModel)
    from perceval.algorithm import Analyzer

    p = pcvl.Processor("SLOS", BS())
    analyzer = Analyzer(p, [pcvl.BasicState([1, 1])], '*')  # '*' means
        all possible output states
    pcvl.pdisplay(analyzer)
```

Output:

(0.7071067811865476+0j)
(-0.7071067811865476+0j)
0.5000000000000001

	$\|1,1\rangle$	$\|0,2\rangle$	$\|2,0\rangle$
$\|1,1\rangle$	0	$\frac{1}{2}$	$\frac{1}{2}$

```
    ## Exercise:  Choose a random unitary 3x3 U, and output the table
        probabilities when the input |1,1,0> passes through the LO-Circuit
        of unitary U.

    ## Solution:
    randU = pcvl.Unitary(pcvl.Matrix.random_unitary(3))
    input = pcvl.BasicState([1, 1, 0])

    p = pcvl.Processor("SLOS", randU)  #We can put in the processor a
        pcvl.Unitary instead of a circuit ! We don't need to use
        pcvl.decomposition
    analyzer = pcvl.algorithm.Analyzer(p, [input], '*')
    pcvl.pdisplay(analyzer)
```

Output:

	$\|0,2,0\rangle$	$\|0,1,1\rangle$	$\|2,0,0\rangle$	$\|1,1,0\rangle$	$\|1,0,1\rangle$	$\|0,0,2\rangle$
$\|1,1,0\rangle$	0.01784	0.160755	0.27189	0.266615	0.066101	0.216799

2. Sampling.

Although it's crucial to compute the output distribution, it is not what we obtain from a photonic chip. Indeed, in practice, we only can obtain a single sample from the distribution each time we run the circuit. This can be done using the backend `SLOS` [14].

```
    from perceval.algorithm import Sampler  # import the Sampler class

    p = pcvl.Processor("SLOS", BS())
    p.min_detected_photons_filter(0)  # Do not filter out any output state
    p.with_input(pcvl.BasicState([1,1]))

    # The sampler holds 'probs', 'sample_count' and 'samples' calls. You
        can use the one that fits your needs!
```

14. Natively, SLOS computes the probabilities. If we ask SLOS to sample, it will therefore compute all the probabilities and then get a sample. If we just want one sample, it is better to use `CliffordClifford2017`, as detailed in the next code.

```
    sampler = Sampler(p)

    # A sampler call will return a Python dictionary containing sampling
        results, and two performance scores
    # sample_count = sampler.sample_count(1000)
    # sample_count contains {'results': <actual count>, 'physical_perf':
        float [0.0 - 1.0], 'logical_perf': float [0.0 - 1.0]}
    sample_count = sampler.sample_count(1000)
    print(sample_count['results'])
```

Output:

{
|0,2>: 503
|2,0>: 497
}

```
    ## Exercise: Implement the code to sample from the 3x3 Unitary of
        earlier

    #Solution:
    p = pcvl.Processor("CliffordClifford2017", randU)
    p.min_detected_photons_filter(0)  # Do not filter out any output state
    p.with_input(pcvl.BasicState([1, 1, 0]))

    sampler = pcvl.algorithm.Sampler(p)
    sample_count = sampler.sample_count(1000)
    print(sample_count['results'])

    ## Question: How many states do we have for 3 modes and 2 photons?
    ## Answer: There are 6 different states


    ## Question: How many states do we have for m modes and n photons?
    ## Answer: There are m+n-1 choose n different states (cf Bar and Star
        problems).
```

Output:

{
|1,0,1>: 65
|0,2,0>: 24
|2,0,0>: 265
|0,1,1>: 157
|1,1,0>: 267
|0,0,2>: 222
}

Note that for approximating with decent precision a distribution over M different states, we would need M^2 samples. This can be shown by Hoeffding's inequality.

3. Performance and output state filtering

Perceval Processors have a built-in way of computing performance scores. There are two different performance scores:

- Physical performance
- Logical performance

These performance scores help measure the real duration of a data acquisition on a real QPU.

a. Physical performance.

This score is related to the number of detections (on a QPU, it is the number of clicks). It drops output states where photons have been lost, or finish in the same mode. For instance, an imperfect source makes this score drop.

However, you can choose not to filter any output state by lowering the expected clicks with

```
proc.min_detected_photons_filter(0)
```

.

Perceval aims at being an interface for the QPU and as such,

```
proc.min_detected_photons_filter(k)
```

post selects on having at least k photons detected (for threshold detection: photons on at least k different modes). By default, this value is set to n where n is the expected number of input photons. This is useful for retrieving a logical interpretation, making sure that no photon has been lost due to noise and coherent with the use of threshold detectors. However, for various applications such as machine learning where we use the full Fock space and resolve the number of photons, you will have to set it to 0 (and you may introduce you own post selection scheme if needed).

```
    # Create an empty circuit (each input mode is directly connected to a
        detector without interacting with any other)
    empty_circuit = pcvl.Circuit(4)

    perfect_proc = pcvl.Processor("SLOS", empty_circuit)
    imperfect_proc = pcvl.Processor("SLOS", empty_circuit,
        noise=pcvl.NoiseModel(brightness=0.3))

    # We need to specify how many photons we want
    perfect_proc.min_detected_photons_filter(2)
    imperfect_proc.min_detected_photons_filter(2)

    # Set the same input in both processors
    input_state = pcvl.BasicState([1, 0, 1, 0])
    perfect_proc.with_input(input_state)
    imperfect_proc.with_input(input_state)

    perfect_sampler = pcvl.algorithm.Sampler(perfect_proc)
    perfect_probs = perfect_sampler.probs()
    imperfect_sampler = pcvl.algorithm.Sampler(imperfect_proc)
    imperfect_probs = imperfect_sampler.probs()

    print('Physical perf of perfect processor =',
        perfect_probs['physical_perf'])
    print('Physical perf of imperfect processor =',
        imperfect_probs['physical_perf'])  # source emission probability**2

    # You can still disable output state filtering
    imperfect_proc.min_detected_photons_filter(0)
    imperfect_probs = imperfect_sampler.probs()
    print('Physical perf of imperfect processor (without selection) =',
        imperfect_probs['physical_perf'])
```

Output:
Physical perf of perfect processor = 1
Physical perf of imperfect processor = 0.08999999999999997
Physical perf of imperfect processor (without selection) = 1

b. Logical performance

This performance computation is set up by heralded modes and/or post-selection function set in a processor. Depending on the circuit used, on the post-selection function, you may observe that physical and logical performance score interact. If you are interested in a theoretical gate performance, you can compute the global performance as

```
global_perf = res["physical_perf"] * res["logical_perf"]
```

.

Here is a quick example of the heralding / post-selection syntax in Perceval. You will see the result later in this tutorial.

```
    circuit = pcvl.Circuit(3) // BS() // (1, BS()) // BS()
    p = pcvl.Processor("SLOS", circuit)
    p.add_herald(2, 0)  # Third mode is heralded (0 photon in, 0 photon
        expected out)

    p.min_detected_photons_filter(1)  # Only non-heralded modes must be
        taken into account in this filter

    # After a mode is heralded, you must not take it into account when
        setting an input to the processor
    p.with_input(pcvl.BasicState([1, 0]))
    sampler = pcvl.algorithm.Sampler(p)
    probs = sampler.probs()
    print("With herald only")
    print("Logical perf =", probs['logical_perf'])
    print(probs['results'])

    # A post-selection function can be created like this:
    postselect_func = pcvl.PostSelect("[1] == 1")  # meaning we required 1
        photon detection in mode #1

    p.set_postselection(postselect_func)  # Add post-selection
    probs = sampler.probs()
    print("With herald + post-selection function")
    print("Logical perf =", probs['logical_perf'])
    print(probs['results'])
```

Output:

With herald only
Logical perf = 0.75
{
|1,0>: 0.028595479208968374
|0,1>: 0.9714045207910317
}
With herald + post-selection function
Logical perf = 0.7285533905932737
{
|0,1>: 1.0
}

4. Variational algorithm

In variational algorithms, the samples from a quantum circuit allow us to approximate an expectation value, which is then used to determine the value of a loss function. This loss function is chosen such that minimizing it yields a solution to a given problem. By changing the values of the parameters in our quantum circuit, we can search for this minimum.

We won't go into the details of variational algorithms. However, it may be useful to see how to perform an optimization with Perceval. We will use the library scipy.optimise.

The following code solves the problem of finding a linear optical circuit which, given a Fock State $|1,1,1,1\rangle$, maximizes the probability of outputting $|4,0,0,0\rangle$. The solution below works for an arbitrary n.

```
import random
from scipy import optimize

# Data
n = 4
input = pcvl.BasicState([1]*n)
output_to_max = pcvl.BasicState([n]+[0]*(n-1))
backend = pcvl.SLOSBackend()

# TO-DO: implement a generic circuit of size n with parameters. Code
    the loss function to maximise the good output. Launch the
    optimisation procedure. Output the probability and circuit obtained

# We take a universal circuit
circuit = pcvl.GenericInterferometer(n,
    lambda i: BS(theta=pcvl.P(f"theta{i}"),
        phi_tr=pcvl.P(f"phi_tr{i}")),
    phase_shifter_fun_gen=lambda i: PS(phi=pcvl.P(f"phi{i}")))
param_circuit = circuit.get_parameters()
params_init = [random.random()*np.pi for _ in param_circuit]


def loss_function(params):
    for i, value in enumerate(params):
        param_circuit[i].set_value(value)
    backend.set_circuit(circuit)
    backend.set_input_state(input)
    return -backend.probability(output_to_max)  # we want to maximise
        the prob, so we want to minimise the -prob
```

```
26
27
28      # Run the optimisation
29      o = optimize.minimize(loss_function, params_init, method="Powell")
30
31      print(f"The maximum probability is {-loss_function(o.x)}")
32
33      # For n=4, the probability should be 3/32
34      # The maximum can also be obtained with the Hadamard matrix :
35
36      H4 = (1/2)*np.array([[1,1,1,1], [1,-1,1,-1], [1,1,-1,-1], [1,-1,-1,1]])
37      backend.set_circuit(pcvl.Unitary(pcvl.Matrix(H4)))
38      backend.set_input_state(input)
39      backend.probability(output_to_max)
```

Output:
The maximum probability is 0.09374999565153506

5. To go further: connect to a chip

Perceval is also connected to real physical chips. The syntax to directly sample from them is on https://perceval.quandela.net/docs/notebooks/Remote_computing.html.

E. Encoding Qubits

1. Dual-rail encoding

To perform quantum computations using photons, we need an encoding, i.e. a mapping between Fock states and qubit states. One natural way to encode qubits is the path encoding. A qubit will be encoded with two optical spatial modes: this is the dual-rail or path encoding.

In the dual-rail encoding, the logical qubit state $|0\rangle_L$ will correspond to a photon in the upper mode, as in the Fock state $|1,0\rangle$, while $|1\rangle_L$ will be encoded with $|0,1\rangle$. We can extend this to multiple qubits by having twice as many modes as there are qubits. For example the 3-qubit state $\frac{1}{\sqrt{2}}(|000\rangle_L + |111\rangle_L)$ can be encoded with 3 photons and $3 \times 2 = 6$ modes: $\frac{1}{\sqrt{2}}(|1,0,1,0,1,0\rangle + |0,1,0,1,0,1\rangle)$

2. Single-qubit gates

Using the dual-rail encoding, single-qubit gates only deal with one photon. Thus, their implementations are straightforward with elementary linear optical components. Exercise: Please provide the linear optical circuits for the gates below.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$R_X = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$

$$R_Y = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$

$$R_Z = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix}$$

```
    ## Exercise: find the LO-circuits for each gate

    ## Solution:
    circuit_x = PERM([1, 0])  # it's not the only way
    circuit_y = PERM([1, 0]) // (0, PS(-np.pi/2)) // (1, PS(np.pi/2))
    circuit_z = pcvl.Circuit(2) // (1, PS(np.pi))
    circuit_h = BS.H()

    circuit_rx = pcvl.Circuit(2) // (0, PS(np.pi)) //
        BS.Rx(theta=pcvl.P("theta")) // (0, PS(np.pi))
    circuit_ry = BS.Ry(theta=pcvl.P("theta"))
    circuit_rz = BS.H() // circuit_rx // BS.H()  # Indeed, Rz = H Rx H
```

3. Two-qubit gates

On the other hand, in dual-rail encoding, most of the two-qubit gates can not be deterministic, i.e. they have a probability to fail [Kok+07]. There are two ways to detect that failure:

- We can use additional photons called ancillas, which we can measure independently of the main circuit photons. Depending on the state obtained on the ancilla, we know whether the gate has succeeded or not on the main qubits. These gates are called heralded.
- We can also directly measure the main circuit qubits, and depending on the result, assess whether the gate has succeeded or not. These gates are postselected.

The CNOT gate acts on two qubits, a control and a target, and flips the value of the target if the control qubit is in state $|1\rangle_L$. In the following two exercises, we see the two types of CNOT gates:

- the postselected CNOT of Ralph et al. [Ral+02].
- the heralded CNOT of KLM [KLM01].

```
    ## We introduce the component catalog. It contains both CNOT gates.
    from perceval.components import catalog
    print(catalog.list())
```

Output:

['klm cnot', 'heralded cnot', 'postprocessed cnot', 'heralded cz', 'generic 2 mode circuit', 'mzi phase first', 'mzi phase last', 'symmetric mzi', 'postprocessed ccz', 'toffoli', 'postprocessed controlled gate', 'x', 'y', 'z', 'h', 'rx', 'ry', 'rz', 'ph', 's', 'sdag', 't', 'tdag']

```
    ## Ralph's et al. CNot

    print(catalog['postprocessed cnot'].doc)
    ralph_cnot = catalog['postprocessed cnot'].build_processor()
    ralph_cnot.min_detected_photons_filter(0)
    ## You can set its input state with a LogicalState
    ralph_cnot.with_input(pcvl.LogicalState([1, 0]))

    pcvl.pdisplay(ralph_cnot, recursive=True, render_size=1.25)
```

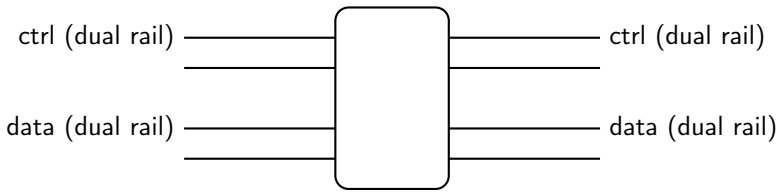
Output:

POSTPROCESSED CNOT DOCUMENTATION

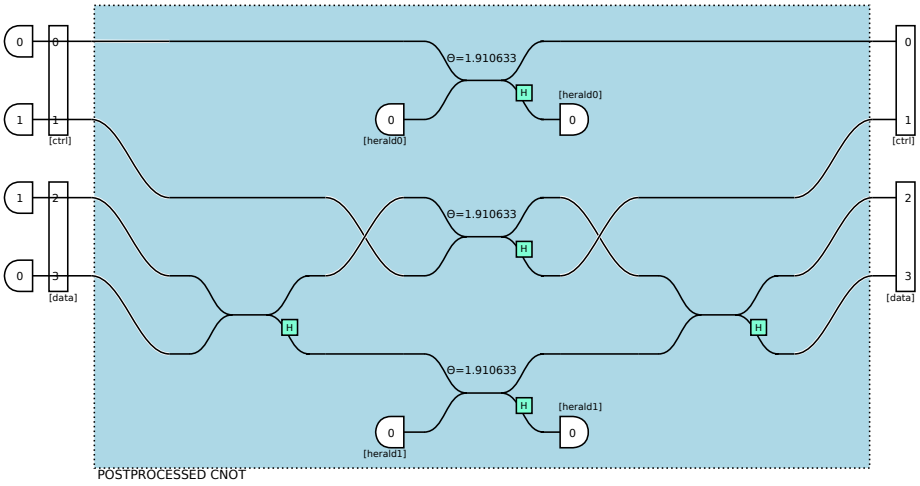
CNOT gate with 2 heralded modes and a post-selection function

Scientific article reference:
https://journals.aps.org/pra/abstract/10.1103/PhysRevA.65.062324

Schema:



See also: klm cnot and heralded cnot (using cz)




```
## Exercise: Check/convince yourself that the circuit above is
    performing a CNOT in the dual rail encoding
```

```
## You can sample some output states
cnot_sampler = pcvl.algorithm.Sampler(ralph_cnot)
samples = cnot_sampler.probs()
print(samples['results'])
print("Some output states were not selected because of heralds and
    post-processing => you can check the logical performance")
print("Logical performance = ", samples['logical_perf'])
```

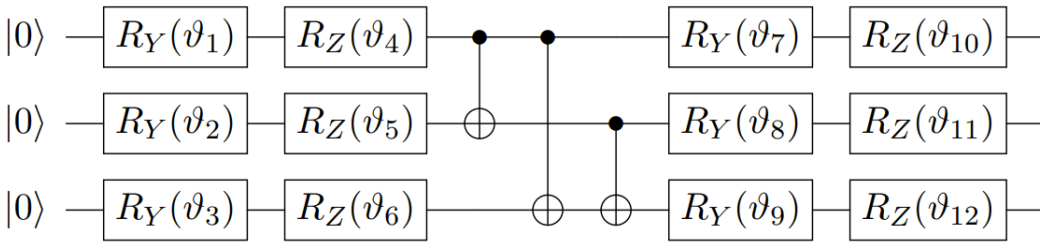
Output:

{
|0,1,0,1>: 1.0
}
Some output states were not selected because of heralds and post-processing => you can check the logical performance
Logical performance = 0.1111111111111111

```
        ## Exercise: Check it performs a CNOT, and explicit the difference
            between the two types of CNOT
```

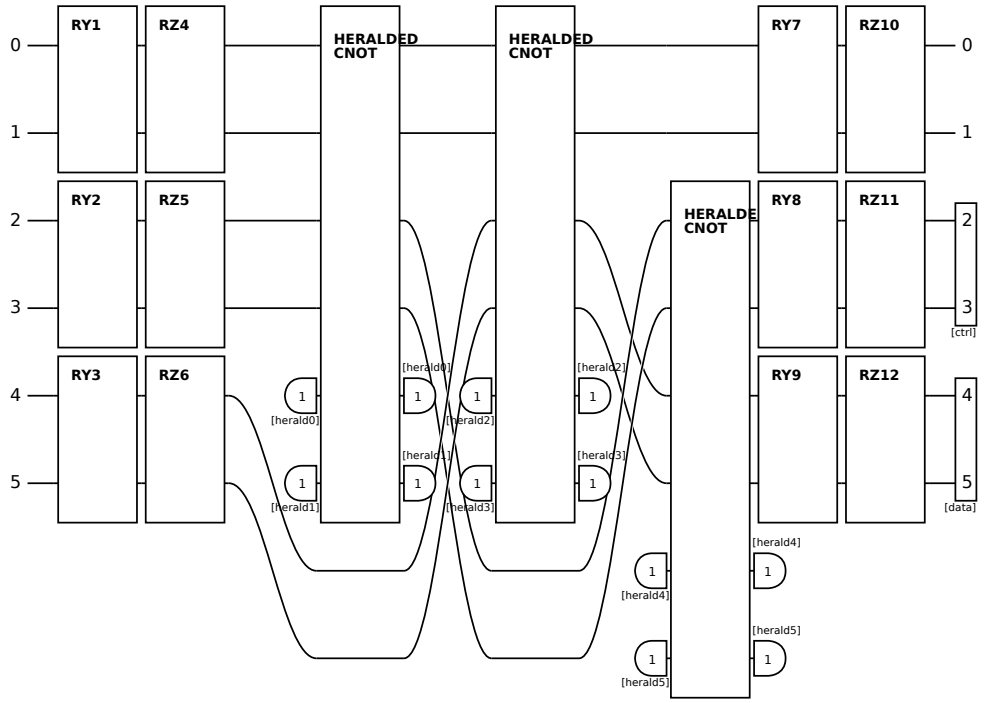
4. Exercise

The following circuit comes from [Fun+21].




```
    ## Exercise: give the corresponding linear optical circuit in the
        dual-rail encoding

    ## Solution:
    # Let's try to implement that circuit properly.
    # First, the quantum gates, as coded above:

    Rx = lambda i: pcvl.Circuit(2) // (0,PS(np.pi)) //
        BS.Rx(theta=pcvl.P(f"theta{i}")) // (0,PS(np.pi)) #Be careful for
        the minus! We use a convention
    Ry = lambda i: pcvl.Circuit(2,name=f"Ry{i}") //
        BS.Ry(theta=pcvl.P(f"theta{i}"))
    Rz = lambda i: pcvl.Circuit(2,name=f"Rz{i}") // BS.H() // circuit_rx
        // BS.H()
    cnot = catalog['heralded cnot'].build_processor()

    # Our qubits in the dual rail encoding
    q1, q2, q3 = [0,1], [2,3], [4,5]

    p = pcvl.Processor("SLOS",6)

    for i in range(3):
        p.add(2*i,Ry(i+1)).add(2*i,Rz(i+4))
    p.add(q1+q2, cnot)
    p.add(q1+q3, cnot)
    p.add(q2+q3, cnot)

    for i in range(3):
        p.add(2*i, Ry(i+7)).add(2*i, Rz(i+10))

    pcvl.pdisplay(p,recursive=False)
```

Output:



IV.3 Mitigating the crosstalk of linear optical circuits

Crosstalk refers to unwanted interactions between phase shifters, thus introducing errors in linear optical implementations. As such, it is crucial to have effective methods for mitigating crosstalk.

In this section, we first define the phase-voltage relation typically used to model crosstalk (Section IV.3.1). We then extend this model to include parasitic phases that appear on empty waveguides, introducing the concept of induced phase shifters (Section IV.3.2). Next, we introduce the notion of invariant transformations and demonstrate how they help to efficiently mitigate the extended crosstalk model (Section IV.3.3). We detail an efficient algorithm for finding these invariant phase transformations (Section IV.3.4). Finally, we present a graph-theory framework to prove the optimality of our algorithm and establish a simple crosstalk robustness criterion (Section IV.3.5).

Contributions

This section presents [Fyr+25], a manuscript in preparation and coauthored with Andreas Fyrillas, Simone Piacentini, Nicolas Maring, Jean Senellart, and Nadia Belabas.

The extended crosstalk model (Section IV.3.2 and IV.3.3) has been introduced by Andreas Fyrillas, and is also detailed in his PhD thesis [Fyr25].

The remainder of this section (Section IV.3.4 and IV.3.5), is of my own contribution.

IV.3.1 Crosstalk: definition and model

In practice, as described in Section I.3.3.A., phase shifters that change the refractive index of the waveguide can be implemented with thermo-optic [Har+14] or electro-optic effects [Li+20]. These effects are controlled by applying a specific value of voltage or electric current on the phase shifters. Our focus in this section is on voltage-controlled thermo-optic phase shifters, which is the most widely used technology.

Let us consider a PIC (photonic integrated circuit) with n_{CPS} controllable phase shifters (CPS). In the ideal case, the phase ϕ_i of the i^{th} CPS is quadratically linked to the voltage V_i applied on it as follows:

$$\phi_i = \alpha_i V_i^2 + \phi_i^{(0)} \tag{IV.23}$$

where $\phi_i^{(0)}$ is the inherent constant phase when there is no voltage applied.

However, this model does not account for the physical effect of applying heat to change the phase. In practice, the heat generated by a single CPS diffuses throughout the circuit, impacting other phase shifters. This phenomenon, known as crosstalk, is recognized as a significant source of imperfection in the integrated photonics literature [BNC23].

To model the crosstalk, it is usual [Fyr+24] to consider the following equation:

$$\Phi_{\text{CPS}} = C_{\text{CPS}} \times V^{\circ 2} + \Phi_{\text{CPS}}^{(0)} \tag{IV.24}$$

where Φ_{CPS} is a column vector of n_{CPS} entries, C_{CPS} is an $n_{\text{CPS}} \times n_{\text{CPS}}$ crosstalk matrix, V is a column vector of n_{CPS} applied voltages, $\circ$ denotes the Hadamard (element-wise) product, i.e. $V^{\circ 2}$ is the element-wise square of V, and all quantities are real-valued.

Remark IV.40 (Inversibility of the crosstalk matrix). The diagonal elements quantify how much the voltage applied on a CPS does indeed change the phase on which the CPS is located on. Very often in well-designed PICs, these coefficients are significant compared to the other of the same rows. In that case, the crosstalk matrix is diagonally dominant and is therefore invertible.

In practice, the goal is for a PIC to implement a unitary operation U_{target} on the optical modes. From this target matrix, we derive the corresponding target vector Φ_{CPS}, which specifies the phase values required to perform the desired transformation [15]. When the crosstalk matrix is invertible, the voltages to be applied can be obtained by inverting Equation (IV.24):

$$V^{\circ 2} = C_{\text{CPS}}^{-1}(\Phi_{\text{CPS}} - \Phi_{\text{CPS}}^{(0)}) \tag{IV.25}$$

Once the voltages are found, we can apply it and physically evaluate the fidelity [Pen+24] between the target U_{target} and the implemented circuit:

$$\frac{\mathrm{Tr}(|U_{\text{target}}|^{\top}|U_{\text{observed}}|)}{m} \tag{IV.26}$$

where Tr denotes the matrix trace, the absolute value is taken element-wise, and m is the number of modes of the PIC.

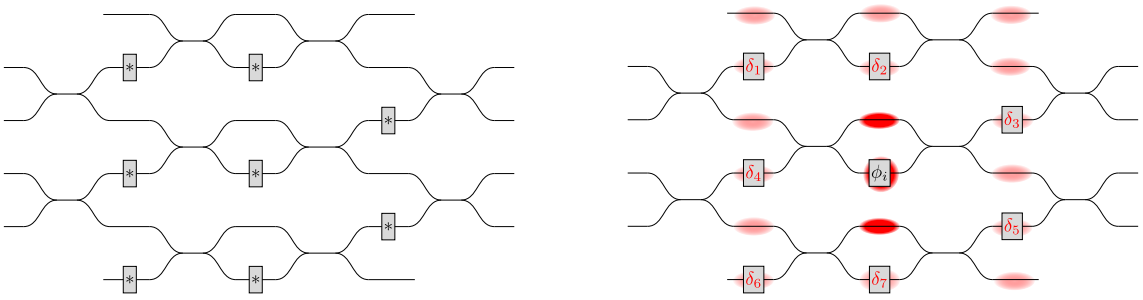
15. The required phases can be determined either algebraically — when exact solutions exist, such as for universal interferometers [Rec+94; Cle+16] — or via numerical optimization, typically using gradient descent methods [Fyr+24].

Until now, state-of-the-art fidelities [Fyr+24] were obtained using the usual crosstalk model (Equation (IV.24)), combined with machine learning and clear-box methods that have been shown to introduce artifacts under the form of non-local crosstalk effects.

The purpose of this work is to show that we can recover such fidelities with a more efficient, systematic, and local crosstalk model.

IV.3.2 Extended crosstalk model: induced phases

In practice, applying heat on one CPS is not only affecting the other CPS as in Equation (IV.24) but all other zones of the circuit, as depicted in Figure IV.8.



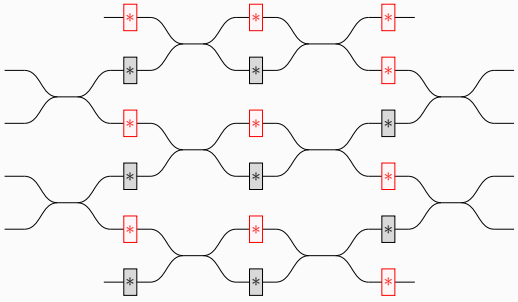
(a) PIC with no applied voltage.

(b) PIC with voltage applied on the middle phase shifter.

Figure IV.8 – On the left, a PIC with controllable phase shifters and no voltage applied. On the right, a voltage is applied to the middle phase shifter to induce a phase ϕ_i. Heat diffuses through the chip, affecting nearby waveguides and altering their phases. Note that in practice, the induced phase shift is stronger when closer to the heat source.

To capture this effect in the crosstalk model, we propose to consider these zones as phase shifters we also want to take into account. We label them as induced phase shifters.

Definition IV.41 (Induced phase shifter (IPS))**.** An induced phase shifter (IPS) is a phase shifter whose phase arises from the crosstalk of CPSs. We represent them as red phase shifters as follows.



We are now ready to extend our crosstalk model with both CPSs and IPSs.

Definition IV.42 (Extended crostallk model)**.** The extended phase-voltage relation is:

$$\Phi = C \times V^{\circ 2} + \Phi^{(0)}$$

where Φ is a column vector of n_{PS} phases, C_{CPS} is an $n_{\mathsf{PS}} \times n_{\mathsf{CPS}}$ crosstalk matrix, V is the

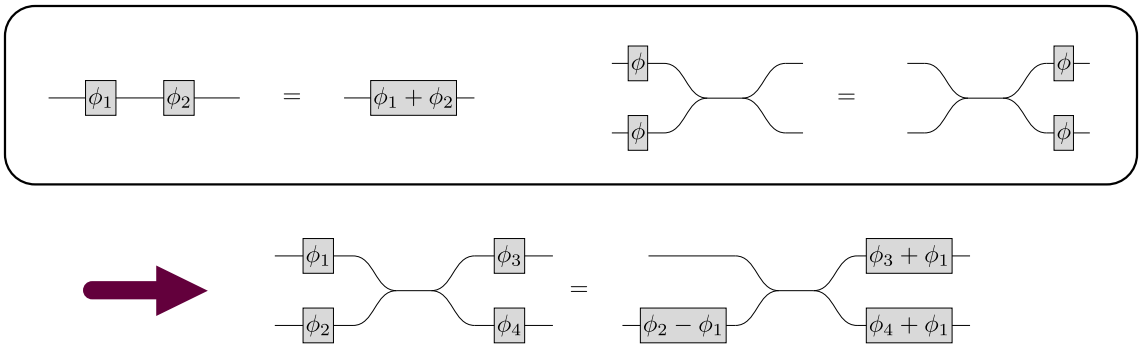


Figure IV.9 – We consider every rule that can be derived with the sum of phases (top left) and the propagation of equal phases (top right) through a beam splitter. In particular, we can derive the bottom rule, that allows to simplify one phase shifter.

column vector of n_{CPS} applied voltages and $n_{\text{PS}} = n_{\text{CPS}} + n_{\text{IPS}}$, i.e. including both controllable and induced phase shifters.

Note we cannot recover the voltage for the target phases anymore as in Equation (IV.25). Indeed, the crosstalk matrix C is now rectangular, and thus does not have an inverse. We could take the Moore-Penrose inverse of C, but this does not yield a high-fidelity implementation of the desired operation.

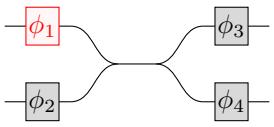
In the following, we propose to remove the rows of C associated with IPSs, by applying invariant transformations on the matrix. Such transformation can be done when we have linear relations between IPSs and CPSs, that do not change the measurable outcome of the circuit (Section IV.3.3). We assume here that such crosstalk matrix C can be found. Another contribution of [Fyr+24] is to provide an efficient machine learning method to characterize this matrix.

IV.3.3 Reducing the crosstalk matrix with invariant transformations

With our new extended crosstalk model (Definition IV.42), we have now a rectangular matrix C that we cannot invert to find the target phases. We propose to remove the additional rows of the crosstalk matrix. In order to perform that reduction, we need to characterize transformations that do not change the outcome of the system. We label them as invariant.

Definition IV.43 (Invariant transformation)**.** A transformation of the PIC is invariant when the measurable outcomes of the physical experiment do not change. Formally, a transformation is invariant when the semantics of the previous and new PIC are equal up to a global phase.

Example IV.44 (Invariant phase transformation)**.** To illustrate such transformations, let us consider the following small circuit, composed of three CPSs and one IPS on the top left:



Using the bottom rule of Figure IV.9, we know there exists an equivalent circuit. Formally, we say the following transformation

$$\begin{aligned}
\phi_1 &\longrightarrow \phi_1 - \phi_1 = 0 \\
\phi_2 &\longrightarrow \phi_2 - \phi_1 \\
\phi_3 &\longrightarrow \phi_3 + \phi_1 \\
\phi_4 &\longrightarrow \phi_4 + \phi_1
\end{aligned}$$

is invariant.

We can now show that such invariant transformations that link CPSs and IPSs enable us to remove rows of the crosstalk matrix without changing the outcome.

Theorem IV.45 (Crosstalk matrix invariant transformation)**.** *Let us consider a PIC with PSs labeled* $1, 2, ..., n$ *(including controlled and induced PSs) and a crosstalk matrix* C*. We denote* ϕ_i *the phase shift value implemented by the* i^{th} *PS.*

Let us assume there exists an invariant phase transformation of the form

$$\phi_i \longrightarrow \phi_i + \alpha_i \psi \tag{IV.27}$$

that holds for all real values of ψ *and where the coefficients* $\alpha_1, ..., \alpha_n$ *are real-valued.*

Then the following transformation on the i^{th} *row of* C*:*

$$R_i \longrightarrow R_i + \alpha_i R_k \tag{IV.28}$$

is an invariant transformation.

Proof. First, let us note that

$$R_i \longrightarrow R_i + \alpha_i R_k$$

is equivalent to

$$C_{ij} \longrightarrow C_{ij} + \alpha_i C_{kj}.$$

We remind the phase-voltage relation of our extended crosstalk model (Definition IV.42):

$$\phi_i = \sum_{j=1}^{n_{\text{CPS}}} C_{ij} V_j^2 + \phi_i^{(0)}$$

It remains to note that $C_{ij} \longrightarrow C_{ij} + \alpha_i C_{kj}$ is equivalent to:

$$\begin{aligned} \phi_i \longrightarrow & \sum_{j=1}^{n_{\text{CPS}}} (C_{ij} + \alpha_i C_{kj}) V_j^2 + \phi_i^{(0)} \\ = & \ \phi_i + \alpha_i \sum_{j=1}^{n_{\text{CPS}}} C_{kj} \\ = & \ \phi_i + \alpha_i \psi \end{aligned}$$

which is an invariant transformation by assumption. □

Reducing the crosstalk matrix C is a consequence of the following corollary.

Corollary IV.46 (Remove row of the crosstalk matrix)**.** *Let* $(\alpha_1, \ldots, \alpha_{n_{PS}})$ *be an invariant phase transformation with* $\alpha_k = -1$*. Then there is an invariant transformation that both removes the* k^{th} *row of the crosstalk matrix and the* k^{th} *PS from the circuit.*

Proof. Following Theorem IV.45, $\alpha_k = -1$ implies that the k^{th} row of the crosstalk matrix is zero. By applying the same transformation to ϕ_0, we can also set the k^{th} entry to zero. Then, due to the phase-voltage relation (Definition IV.42), the k^{th} phase remains zero for any applied voltage. We can therefore remove this component from the PIC. □

Thus, the goal is to use such invariant phase transformations to remove as many IPSs as possible from the PIC. If all IPSs are removed, then the crosstalk matrix C becomes square again, leading to significant improvements in fidelity compared to the Moore–Penrose pseudoinverse on the initial rectangular matrix.

IV.3.4 Finding invariant phase transformations

Following Collorary IV.46, we want to find invariant phase transformations to reduce the crosstalk matrix and remove IPSs from the PIC. In this section, we provide an algorithm to find such transformations. More precisely and similarly to Example IV.44, we aim to remove all the IPSs that can be propagated with the rules of Figure IV.9. The phases that can be simplified with such rules are the ones we define as follows.

Definition IV.47 (BS-adjacent)**.** Let a beam splitter be surrounded by four waveguides. We say that the waveguides, or the components on them, are BS-adjacentif they are directly connected to the beam splitter.

Definition IV.48 ((Non-) Removable phase shifters)**.** An induced phase shifter is removable if and only if it is BS-adjacent (see Definition IV.47) to three of the following:

- Controlled phase shifters,
- Phase-invariant input/output port [16], or
- Removable induced phase shifter.

Otherwise, we say an induced phase shifter is nonremovable.

Note that the definition of removable IPS is recursive. To find removable IPSs, we propose Algorithm 10. The idea is the following.

- We start with all trivially removable IPSs, i.e. those adjacent to three CPSs or input/output ports. We label them as removable and store the corresponding linear phase-invariant transformations in a stack.
- Then, we visit all IPSs that are neighbors of the previously visited ones and that have not yet been explored. If they are removable, we label them as such and push their associated invariant phase transformations to the stack.

Once the algorithms terminates, we can use all the found phase invariant transformations of the stack $\mathcal{L}_\alpha$ and apply them sequentially to remove all the rows associated with these removable IPS Collorary IV.46. The time complexity is detailed in the following proposition.

Proposition IV.49 (Full reduction of the crosstalk matrix)**.** *All the crosstalk matrix rows associated with removable IPSs can be removed in*

$$O\left(n_{\text{PS}} + K n_{\text{removable IPS}}\right)$$

operations, where K is the cost of updating one row of the crosstalk matrix, and $n_{\text{removable IPS}}$ is the number of removable IPSs. In the worst case, the complexity of the full reduction is:

$$O\left(n_{\text{IPS}} \times n_{\text{CPS}}\right)$$

Proof. Note that in Algorithm 10, each phase is at most visited once [17]. At each visited phase, we check its neighbors to determine if they are removable. Since there are at most 6 neighbors (3 from

16. Usually in near-term PICs, the input and output ports are connected to source or detectors. When we consider Fock basis states at the inputs as in the original paper [Fyr+25], then the phase on such zones do not need correction as they do not affect the outcome. These kinds of input/output ports are therefore phase invariant.

17. Note that we can implement the *is visited* property with a Boolean array of size n_{PS}

Algorithm 10 Finding invariant phase transformations of removable IPSs

procedure REMOVEIPS
  $\mathcal{R}_{\text{curr}} \leftarrow$ trivial removable IPS
  $\mathcal{R}_{\text{next}} \leftarrow \emptyset$
  $\mathcal{L}_\alpha \leftarrow$ empty stack
  while $\mathcal{R}_{\text{curr}} \neq \emptyset$ **do**
    for all unvisited IPS neighbors i of $\mathcal{R}_{\text{curr}}$ **do**
      if i is removable **then**
        Add i to $\mathcal{R}_{\text{next}}$
        Push (k, p, q, r) to $\mathcal{L}_\alpha$
          where $\alpha_k = \alpha_p = -1$, $\alpha_q = \alpha_r = 1$, others $= 0$
      end if
    end for
    $\mathcal{R}_{\text{curr}} \leftarrow \mathcal{R}_{\text{next}}$
    $\mathcal{R}_{\text{next}} \leftarrow \emptyset$
  end while
end procedure

the left and 3 from the right), each step takes $O(1)$ operations. Thus, the time complexity of the algorithm is, in the worst case:

$$O(n_{\text{PS}})$$

Once the algorithm finishes, we have a stack $\mathcal{L}_\alpha$ of $n_{\text{removable IPS}}$ elements. Each element involves 4 row operations on the crosstalk matrix. Hence, the reduction step from the stack costs:

$$K \times n_{\text{removable IPS}}$$

If there is no optimized implementation for the rows of C, and if the rows are not sparse, then $K = O(n_{\text{PS}})$, concluding the proof. [18] □

Remark IV.50 (Flexibility). The presented phase simplification algorithm is flexible as it can be extended with additional rules for removing IPSs. For instance, we could include other linear optical components, and we would change the definition of removable IPSs accordingly.

It is important to note that, at this stage, there is no a priori guarantee that Algorithm 10 removes all potentially removable IPSs. This property is formalized in the following theorem, which we prove in the next section.

Theorem IV.51 (Optimality of Algorithm 10)**.** *All phases that can be removed using the equations of Figure IV.9 are labeled as removable in Algorithm 10.*

IV.3.5 Graph-theory criteria for crosstalk robustness: acyclic graph and circuit rank

We propose to both provide simple criteria for characterizing the crosstalk robustness and prove Theorem IV.51, by introducing graph-theory tools. Let us first define the pruned graph of a given PIC.

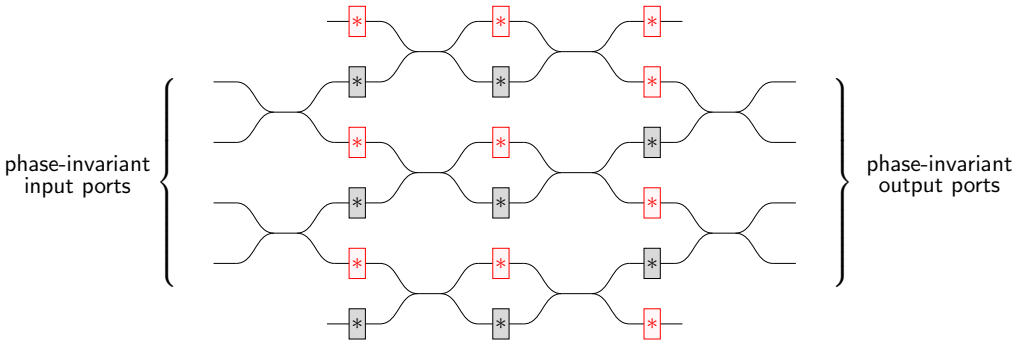
18. Experimentally, crosstalk is quite local, so some row operations can be optimized, and the upper bound is not reached in practice; see [Fyr+25], where we numerically show a scaling in $O(n_{\text{PS}}^{3/2})$.

Definition IV.52 (Pruned-graph representation of a PIC)**.** The graph representation of a PIC is a graph where:

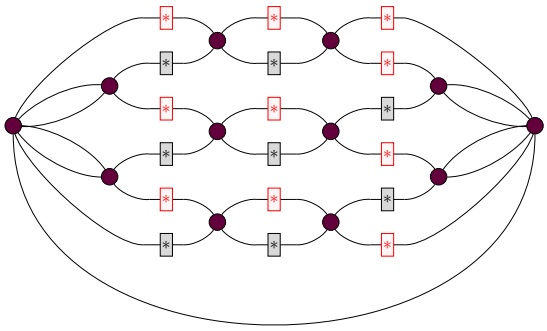
- The nodes represent the beam splitters,
- The edges correspond to the waveguides connecting the beam splitters,
- An input and output node are included, connected by an edge.

The pruned graph of a PIC is obtained by removing all edges that correspond to controllable phase shifters and phase invariant input/output ports.

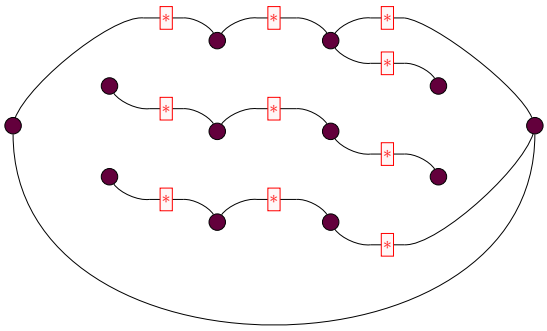
Example IV.53. We provide examples of the graph representation of two PICs and their associated pruned graph in Figure IV.10.



(a) PIC with four phase invariant input/output ports.



(b) Graph of the PIC.



(c) Pruned graph of the PIC.

Figure IV.10 – We present an example of a PIC, its graph, and its pruned graph (see Definition IV.52). Note that the pruned graph contains one cycle. Thus, one induced phase shifter cannot be removed.

A key property is the following lemma, providing a link between the removability of an IPS and cycles in a graph.

Lemma IV.54 (Cycle)**.** *An IPS is not removable with any combination of the rules of Figure IV.9 if and only if the associated edge in the PIC's pruned graph is in a cycle.*

Proof. Following the recursive definition of removable induced phase shifters (Definition IV.48), an IPS is removable to the left (resp. right) when all paths starting from the left (resp. right) of the IPS eventually reach a phase invariant input/output port or a controlled PS. An IPS is thus removable when it is removable to the left or to the right. Consequently, an IPS is non-removable when it is non-removable to the left and to the right. In other words, an IPS is non-removable when one of the paths to the left and to the right ends on a phase-dependent input/output port or performs a cycle.

Note that it is important to connect the input and output nodes in the pruned graph (Definition IV.52) if the input/output ports are phase-dependent and not invariant. Indeed, if an induced PS in the PIC has paths to the left and to the right leading to phase-dependent input/output ports, it should also be non-removable. If the input and output nodes are connected in the graph, this non-removability also manifests as the presence of a cycle.

□

Lemma IV.54 has two direct important consequences. First, it means that when there is no cycle in the pruned graph, there is no non-removable phases, giving a criterion for crosstalk robustness.

Definition IV.55 (Crosstalk robust)**.** A PIC is crosstalk robust if its pruned graph is acyclic.

Second consequence of Lemma IV.54 is that the minimal number of non-removable phase shifters is therefore the minimal number of edges we can remove from a graph to make it acyclic. Remarkably, this number is well known in graph theory and is known as the circuit rank [Ber01]. It leads to the following proposition.

Proposition IV.56 (Minimal number of non-removable PS)**.** *The minimal number of non-removable PS in a given PIC is the circuit rank of its associated pruned graph. As the pruned graph is undirected, the circuit rank r is straightforward to compute from*

$$r = e - v + c$$

where e is the number of edges, v is the number of vertices, and c is the number of connected components in the pruned graph.

Example IV.57. For instance in Figure IV.10c, we have $e = 11$, $v = 12$ and $c = 2$, yielding indeed $r = 1$, i.e. a single edge must be removed from graph to make it acyclic.

We can now conclude and prove that Algorithm 10 is optimal.

Proof of Theorem IV.51.

Remarkably, it is possible to find a minimum set of edges breaking all the pruned graph cycles using a greedy algorithm [Ber73]. At each step, the greedy algorithm chooses to cut an edge belonging to at least one cycle of the pruned graph, and continues until there is no cycle. A cut edge by the algorithm in the pruned graph corresponds, in the phase simplification algorithm, to label a non-removable induced PS.

More precisely, Algorithm 10 follows the greedy approach: when no additional induced PS can be identified as removable, the algorithm randomly selects one of the remaining unlabeled induced PS and designates it as non-removable (cutting a random edge in a cycle) and continues until every PS is removable (until the pruned graph is acyclic). The greedy algorithm being optimal for finding the

minimum set of edges breaking all the cycles, the phase simplification algorithm does indeed give the minimum number of non-removable IPSs. Thus, the algorithm and the reduction of the crosstalk matrix are optimal.

□

Conclusion and outlook

In this thesis, we have introduced a range of state-of-the-art algorithms — such as SLOS, LO-SLAP, and the gate-by-gate method with generalized Ryser's formula — as well as three complete graphical languages — $\mathbf{LO}_{\mathsf{PP}}$-, $\mathbf{LO}_{\mathsf{v}}$-, and $\mathbf{LO}_{fi}$-calculus — capturing both the computational and formal behaviour of various linear optical circuits.

Remarkably, these applications have proven to be multifold. More precisely, $\mathbf{LO}_{\mathsf{PP}}$-circuits enabled us to derive a major result on the completeness of quantum circuits. SLOS led to the development of Perceval, and both SLOS and LO-SLAP now serve as key backends in this software. Finally, from simple rewriting rules emerged graph-theoretic criteria that help assess — and ultimately guide — the design of crosstalk-robust, or more generally, fault-tolerant photonic integrated circuits.

While our contributions have already shown significant impact, they leave room to tackle even broader and more challenging questions.

On the computational side, all the algorithms introduced focus on exact computation or sampling. While exact methods are essential for theoretical understanding, characterization, and benchmarking, in practice, one can often trade precision for improved computational efficiency — targeting approximations of the outcome probabilities or sampling distributions. Interestingly, many promising approaches have recently emerged for continuous-variable optics, and building a bridge between our discrete-variable models and the continuous-variable literature remains an open and compelling direction.

Finally, for large-scale challenges, it remains to investigate the formalization of optical circuits involving auxiliary sources, postselection schemes, and feedforward — which can be seen as a natural extension of the $\mathbf{LO}_{fi}$-circuits. Such a formalism would capture all current fault-tolerant photonic architectures and is thus closely related to the efficient compilation of quantum logical operations into adaptive linear optical circuits. Many questions remain open regarding the optimality of such constructions, and their resolution could greatly impact not only this model and photonic architectures, but more broadly our understanding of what linear optical quantum computing can efficiently — or even naturally — achieve.

Index

Adjoint, Hermitian adjoint, 22
Annihilation operator, 27

Beam splitter, 37
Boson Sampling, 43
Bosons, 25
Bra, $\langle\cdot|$, 22

Category, 50
Classical eletromagnetic field, 32
Completeness, 55
Confluence, 54
CPS, 213
Creation operator, 28
Crosstalk, 213

DBS, 102
Diamond $\mathbf{LO_{PP}}$-circuits, ${}_{\tilde{n}}\Diamond_{n}$-circuits, 122
Dirar notation, 22
Discrete-variable linear optics, 41

EBS, 108

Feedforward, 60
Fermions, 25
Fock basis state, Fock state, 27
Fock space, 26

Generalized Ryser's formula, 104
Glynn's formula, 46

Hamiltonian, 25
Heisenberg picture, 26
Hilbert space, 21
Hong-Ou-Mandel effect, HOM, 38

IPS, 214

Ket, $|\cdot\rangle$, 22

Light, 32
Linear optics, 34, 35
LO-SLAP, 80
$\mathbf{LO}_{fi}$, $\mathbf{LO}_{fi}$-circuits, $\mathbf{LO}_{fi}$-calculus, 147
$\mathbf{LO_{PP}}$, $\mathbf{LO_{PP}}$-circuits, $\mathbf{LO_{PP}}$-calculus, 113
$\mathbf{LO_{v}}$, $\mathbf{LO_{v}}$-circuits, $\mathbf{LO_{v}}$-calculus, 133

Mask, 61
Measurement, 24
Mode, 27, 31, 34
MZI, Mach-Zehnder interferometer, 38

Normal form, 54
Number particle operator, 27

Observable, 24

Perceval, 186
Permanent, matrix permanent, 42
Phase shifter, 37
Photon, 33
PIC, 213
Polarization mode, 34
Polarizing beam splitter, 38
PRO, 52
PROP, 52

Quantized electromagnetic field, 33
Quantum circuit, 29, 171

Ryser's formula, 45

SBS, 100
SLOS, 61
Soundness, 55
Spatial mode, 34
Steiner tree, 85
Strong simulation, 42
Strongly normalizing, 54

Subtriangular **LO$_{\mathbf{PP}}$**-circuits, $_{\tilde{n}}\triangle_{\tilde{m}}$-circuits, 121

Tensor product, 22
Triangular **LO$_{\mathbf{PP}}$**-circuits, $\triangle$-circuits, 120

Wave, 30
Wave plate, 39
Weak simulation, 43

Bibliography

[AA11] Scott Aaronson and Alex Arkhipov. "The Computational Complexity of Linear Optics". In: *Proceedings of the 43th Annual ACM Symposium on Theory of Computing (STOC'11)*. San Jose, California, USA: Association for Computing Machinery, 2011, pp. 333–342. ISBN: 9781450306911. DOI: 10.1145/1993636.1993682 (cit. on pp. 10, 16, 42, 43, 47, 99, 148).

[AC04] Samson Abramsky and Bob Coecke. "A categorical semantics of quantum protocols". In: *19th Annual IEEE Symposium on Logic in Computer Science (LICS'04)*. 2004, pp. 415–425. DOI: 10.1109/LICS.2004.1319636 (cit. on p. 48).

[AC09] Samson Abramsky and Bob Coecke. "Categorical Quantum Mechanics". In: *Handbook of Quantum Logic and Quantum Structures*. Ed. by Kurt Engesser, Dov M. Gabbay and Daniel Lehmann. Amsterdam: Elsevier, 2009, pp. 261–323. ISBN: 978-0-444-52869-8. DOI: https://doi.org/10.1016/B978-0-444-52869-8.50010-4 (cit. on p. 48).

[Ach+24] Rajeev Acharya et al. "Quantum error correction below the surface code threshold". In: *Nature* 1476-4687 (2024). Accessed: 2024-12-09. DOI: 10.1038/s41586-024-08449-y (cit. on pp. 10, 16).

[ACR18] Matthew Amy, Jian Chen and Neil J. Ross. "A finite presentation of CNOT-dihedral operators". In: *Proceedings of the 15th International Workshop on Quantum Programming Languages (QPL'17)*. Vol. 266. Feb. 2018, pp. 84–97. DOI: 10.4204/EPTCS.266.5 (cit. on p. 170).

[AF19] François Arago and Augustin-Jean Fresnel. "Mémoire sur l'action que les rayons de lumière polarisée exercent les uns sur les autres". French. In: *Annales de Chimie et de Physique* 10 (1819), pp. 288–304 (cit. on p. 38).

[AH14] Scott Aaronson and Travis Hance. "Generalizing and derandomizing Gurvits's approximation algorithm for the permanent". In: *Quantum Info. Comput.* 14.7 & 8 (May 2014), pp. 541–559. ISSN: 1533-7146 (cit. on p. 47).

[AL18] Tameem Albash and Daniel A. Lidar. "Adiabatic quantum computation". In: *Rev. Mod. Phys.* 90 (1 Jan. 2018), p. 015002. DOI: 10.1103/RevModPhys.90.015002 (cit. on pp. 9, 15).

[Ale+19] Gadi Aleksandrowicz et al. *Qiskit: An Open-source Framework for Quantum Computing*. Version 0.7.2. Feb. 2019. DOI: 10.5281/zenodo.2562111 (cit. on p. 191).

[And05] Sean Eron Anderson. *Bit twiddling hacks*. 2005 (cit. on p. 90).

[Aru+19] Frank Arute et al. "Quantum supremacy using a programmable superconducting processor". In: *Nature* 574.7779 (2019), pp. 505–510. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1666-5 (cit. on pp. 10, 16).

[ASD14] Nabila Abdessaied, Mathias Soeken and Rolf Drechsler. "Quantum Circuit Optimization by Hadamard Gate Reduction". In: *Reversible Computation*. Ed. by Shigeru Yamashita and Shin-ichi Minato. Cham: Springer International Publishing, 2014, pp. 149–162. ISBN: 978-3-319-08494-7 (cit. on p. 175).

[Ata18] Stefan Ataman. "A graphical method in quantum optics". In: *Journal of Physics Communications* 2.3 (Mar. 2018), p. 035032. DOI: 10.1088/2399-6528/aab50f (cit. on p. 112).

[BA14] Adam Bouland and Scott Aaronson. "Generation of universal linear optics by any beam splitter". In: *Physical Review A* 89.6 (16th June 2014), p. 062316. ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.89.062316 (cit. on p. 146).

[Bac+21] Miriam Backens et al. "There and back again: A circuit extraction tale". In: *Quantum* 5 (Mar. 2021), p. 421. ISSN: 2521-327X. DOI: 10.22331/q-2021-03-25-421 (cit. on pp. 10, 16, 111).

[Bar+23] Sara Bartolucci et al. "Fusion-based quantum computation". In: *Nature Communications* 14.1 (2023), p. 912. DOI: 10.1038/s41467-023-36493-1. arXiv: 2101.09310 [quant-ph] (cit. on pp. 10, 16, 57, 60).

[Bar+95] Adriano Barenco et al. "Elementary gates for quantum computation". en. In: *Physical Review A* 52.5 (Nov. 1995), pp. 3457–3467. ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.52.3457 (cit. on pp. 29, 115, 173).

[BB14] Charles H. Bennett and Gilles Brassard. "Quantum cryptography: Public key distribution and coin tossing". In: *Theoretical Computer Science* 560 (2014), pp. 7–11. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2014.05.025 (cit. on pp. 10, 16).

[BB84] Charles H. Bennett and Gilles Brassard. "Quantum cryptography: Public key distribution and coin tossing". In: *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*. 1984, p. 175 (cit. on pp. 10, 16).

[BE19] Andreas Bärtschi and Stephan Eidenbenz. "Deterministic Preparation of Dicke States". In: *Fundamentals of Computation Theory*. Ed. by Leszek Antoni Gąsieniec, Jesper Jansson and Christos Levcopoulos. Cham: Springer International Publishing, 2019, pp. 126–139. ISBN: 978-3-030-25027-0 (cit. on p. 102).

[Bel64] John S. Bell. "On the Einstein Podolsky Rosen paradox". In: *Physics Physique Fizika* 1 (3 Nov. 1964), pp. 195–200. DOI: 10.1103/PhysicsPhysiqueFizika.1.195 (cit. on p. 23).

[Ber01] Claude Berge. *The Theory of Graphs*. Dover books on mathematics. Dover Publications, 2001. ISBN: 9780486419756 (cit. on p. 220).

[Ber73] Claude Berge. *Graphs and Hypergraphs*. English. Revised. Vol. 6. North-Holland Mathematical Library. Amsterdam: North-Holland Publishing Company, 1973. ISBN: 0720424534, 9780720424539 (cit. on p. 220).

[BGL22] Sergey Bravyi, David Gosset and Yinchen Liu. "How to Simulate Quantum Measurement without Computing Marginals". In: *Phys. Rev. Lett.* 128 (22 June 2022), p. 220503. DOI: 10.1103/PhysRevLett.128.220503 (cit. on pp. 97, 98).

[Bia+17] Jacob Biamonte et al. "Quantum machine learning". In: *Nature* 549.7671 (2017), pp. 195–202. DOI: 10.1038/nature23474 (cit. on pp. 9, 15).

[Bjö16] Andreas Björklund. "Below All Subsets for Some Permutational Counting Problems". In: *15th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2016)*. Ed. by Rasmus Pagh. Vol. 53. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016, 17:1–17:11. ISBN: 978-3-95977-011-8. DOI: 10.4230/LIPIcs.SWAT.2016.17 (cit. on p. 46).

[BK19] Miriam Backens and Aleks Kissinger. "ZH: A Complete Graphical Calculus for Quantum Computations Involving Classical Non-linearity". In: *Electronic Proceedings in Theoretical Computer Science* 287 (Jan. 2019), pp. 23–42. ISSN: 2075-2180. DOI: 10.4204/eptcs.287.2 (cit. on p. 170).

[BKW22] Niel de Beaudrap, Aleks Kissinger and John van de Wetering. "Circuit Extraction for ZX-Diagrams Can Be #P-Hard". In: *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*. Ed. by Mikołaj Bojańczyk, Emanuela Merelli and David P. Woodruff. Vol. 229. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 119:1–119:19. DOI: 10.4230/LIPIcs.ICALP.2022.119 (cit. on p. 170).

[BN98] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998 (cit. on p. 54).

[BNC23] Sanmitra Banerjee, Mahdi Nikdast and Krishnendu Chakrabarty. "Characterizing Coherent Integrated Photonic Neural Networks under Imperfections". In: *Journal of Lightwave Technology* 41.5 (Mar. 2023), pp. 1464–1479. ISSN: 0733-8724, 1558-2213. DOI: 10.1109/JLT.2022.3193658 (cit. on p. 213).

[BO20] Daniel Jost Brod and Michał Oszmaniec. "Classical simulation of linear optics subject to nonuniform losses". In: *Quantum* 4 (May 2020), p. 267. ISSN: 2521-327X. DOI: 10.22331/q-2020-05-14-267 (cit. on p. 45).

[Boh28] Niels Bohr. "The Quantum Postulate and the Recent Development of Atomic Theory". In: *Nature* 121.3050 (Apr. 1928), pp. 580–590. ISSN: 1476-4687. DOI: 10.1038/121580a0 (cit. on p. 24).

[Boh35] Niels Bohr. "Can Quantum-Mechanical Description of Physical Reality be Considered Complete?" In: *Phys. Rev.* 48 (8 Oct. 1935), pp. 696–702. DOI: 10.1103/PhysRev.48.696 (cit. on p. 23).

[Boo47] George Boole. *The Mathematical Analysis of Logic*. Cambridge University Press, 1847. DOI: 10.1017/CBO9780511701337 (cit. on p. 28).

[Boo54] George Boole. *An Investigation of the Laws of Thought*. Cambridge University Press, 1854. DOI: 10.1017/CBO9780511693090 (cit. on p. 28).

[Bor26] Max Born. "Zur Quantenmechanik der Stoßvorgänge". In: *Zeitschrift für Physik* 37.12 (Dec. 1926), pp. 863–867. ISSN: 0044-3328. DOI: 10.1007/BF01397477 (cit. on p. 24).

[Bos24] Satyendra Nath Bose. "Plancks Gesetz und Lichtquantenhypothese". In: *Zeitschrift für Physik* 26.1 (1924), pp. 178–181. DOI: 10.1007/BF01327326 (cit. on p. 25).

[Boy+00] P. Oscar Boykin et al. "A new universal and fault-tolerant quantum basis". In: *Information Processing Letters* 75.3 (2000), pp. 101–107. ISSN: 0020-0190. DOI: https://doi.org/10.1016/S0020-0190(00)00084-3 (cit. on p. 29).

[BR01] Hans J. Briegel and Robert Raussendorf. "Persistent Entanglement in Arrays of Interacting Particles". In: *Phys. Rev. Lett.* 86 (5 Jan. 2001), pp. 910–913. DOI: 10.1103/PhysRevLett.86.910 (cit. on p. 105).

[BR05] Daniel E. Browne and Terry Rudolph. "Resource-Efficient Linear Optical Quantum Computation". In: *Phys. Rev. Lett.* 95 (1 June 2005), p. 010501. DOI: 10.1103/PhysRevLett.95.010501 (cit. on pp. 10, 16).

[Bra+24] Sergey Bravyi et al. "High-threshold and low-overhead fault-tolerant quantum memory". In: *Nature* 627.8005 (2024), pp. 778–782. ISSN: 1476-4687. DOI: 10.1038/s41586-024-07107-7 (cit. on pp. 10, 16).

[Bro+19] Daniel J. Brod et al. "Photonic implementation of boson sampling: a review". In: *Advanced Photonics* 1.3 (2019), p. 034001. DOI: 10.1117/1.AP.1.3.034001 (cit. on pp. 10, 16).

[Bru+19] Colin D. Bruzewicz et al. "Trapped-ion quantum computing: Progress and challenges". In: *Applied Physics Reviews* 6.2 (May 2019). ISSN: 1931-9401. DOI: 10.1063/1.5088164 (cit. on pp. 10, 16).

[BS11] J. Baez and M. Stay. "Physics, Topology, Logic and Computation: A Rosetta Stone". In: *New Structures for Physics*. Ed. by Bob Coecke. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 95–172. ISBN: 978-3-642-12821-9. DOI: 10.1007/978-3-642-12821-9_2 (cit. on p. 48).

[BS18] Édouard Bonnet and Florian Sikora. "The PACE 2018 Parameterized Algorithms and Computational Experiments Challenge: The Third Iteration". In: *13th International Symposium on Parameterized and Exact Computation, IPEC 2018, August 20-24, 2018, Helsinki, Finland*. Ed. by Christophe Paul and Michal Pilipczuk. Vol. 115. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, 26:1–26:15. DOI: 10.4230/LIPICS.IPEC.2018.26 (cit. on p. 87).

[BS23] Xiaoning Bian and Peter Selinger. "Generators and Relations for 2-Qubit Clifford+T Operators". In: *Electronic Proceedings in Theoretical Computer Science* 394 (Nov. 2023), pp. 13–28. ISSN: 2075-2180. DOI: 10.4204/eptcs.394.2 (cit. on p. 170).

[Bul+24] Jacob F. F. Bulmer et al. *Simulating lossy and partially distinguishable quantum optical circuits: theory, algorithms and applications to experiment validation and state preparation*. 2024. arXiv: 2412.17742 [quant-ph] (cit. on p. 45).

[BW21] Bryn A. Bell and Ian A. Walmsley. "Further Compactifying Linear Optical Unitaries". In: *APL Photonics* 6.7 (1st July 2021), p. 070804. ISSN: 2378-0967. DOI: 10.1063/5.0053421. arXiv: 2104.07561 (cit. on p. 41).

[Cao+19] Yudong Cao et al. "Quantum chemistry in the age of quantum computing". In: *Chemical Reviews* 119.19 (Oct. 2019), pp. 10856–10915. DOI: 10.1021/acs.chemrev.8b00803 (cit. on pp. 9, 15).

[Cas+18] Adrien Cassagne et al. "MIPP: a Portable C++ SIMD Wrapper and its use for Error Correction Coding in 5G Standard". In: *Proceedings of the 2018 4th Workshop on Programming Models for SIMD/Vector Processing*. WPMVP'18. Vienna, Austria: Association for Computing Machinery, 2018. ISBN: 9781450356466. DOI: 10.1145/3178433.3178435 (cit. on p. 72).

[CB21] Maria Chekhova and Peter Banzer. *In Classical, Quantum, and Nonlinear Optics*. Berlin, Boston: De Gruyter, 2021. ISBN: 9783110668025. DOI: doi:10.1515/9783110668025 (cit. on pp. 39, 142).

[CC18] Peter Clifford and Raphaël Clifford. "The Classical Complexity of Boson Sampling". In: *Proceedings of the 2018 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2018, pp. 146–155. DOI: 10.1137/1.9781611975031.10. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611975031.10 (cit. on pp. 43, 44, 79, 190).

[CC19] Robin Cockett and Cole Comfort. "The Category TOF". In: *Electronic Proceedings in Theoretical Computer Science* 287 (Jan. 2019), pp. 67–84. ISSN: 2075-2180. DOI: 10.4204/eptcs.287.4 (cit. on p. 170).

[CC24] Peter Clifford and Raphaël Clifford. "Faster classical boson sampling". In: *Physica Scripta* 99.6 (May 2024), p. 065121. DOI: 10.1088/1402-4896/ad4688 (cit. on pp. 43, 44, 47, 67, 87).

[CCS18] Robin Cockett, Cole Comfort and Priyaa Srinivasan. "The Category CNOT". In: *Electronic Proceedings in Theoretical Computer Science* 266 (Feb. 2018), pp. 258–293. ISSN: 2075-2180. DOI: 10.4204/eptcs.266.18 (cit. on p. 170).

[CD08] Bob Coecke and Ross Duncan. "Interacting Quantum Observables". In: *Proceedings of the 35th International Colloquium on Automata, Languages, and Programming (ICALP'08)*. Ed. by Luca Aceto et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 298–310. ISBN: 978-3-540-70583-3 (cit. on pp. 10, 16, 56, 111, 170).

[CD11] Bob Coecke and Ross Duncan. "Interacting Quantum Observables: Categorical Algebra and Diagrammatics". In: *New Journal of Physics* 13.4 (Apr. 2011), p. 043016. ISSN: 1367-2630. DOI: 10.1088/1367-2630/13/4/043016 (cit. on pp. 10, 16, 56, 111, 170).

[CDL19a] Claude Cohen-Tannoudji, Bernard Diu and Franck Laloë. *Quantum Mechanics, Volume 1: Basic Concepts, Tools, and Applications*. 2nd. Wiley-VCH, 2019. ISBN: 978-3-527-34553-3 (cit. on p. 23).

[CDL19b] Claude Cohen-Tannoudji, Bernard Diu and Franck Laloë. *Quantum Mechanics, Volume 2: Angular Momentum, Spin, and Approximation Methods*. 2nd. Wiley-VCH, 2019. ISBN: 978-3-527-34554-0 (cit. on p. 23).

[CDL19c] Claude Cohen-Tannoudji, Bernard Diu and Franck Laloë. *Quantum Mechanics, Volume 3: Fermions, Bosons, Photons, Correlations, and Entanglement*. 1st. Wiley-VCH, 2019. ISBN: 978-3-527-34555-7 (cit. on p. 23).

[CDP24] Alexandre Clément, Noé Delorme and Simon Perdrix. "Minimal Equational Theories for Quantum Circuits". In: *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'24)*. Tallinn, Estonia: Association for Computing Machinery, 2024. ISBN: 9798400706608. DOI: 10.1145/3661814.3662088 (cit. on p. 170).

[Cha20] Ulysse Chabaud. "Continuous variable quantum advantages and applications in quantum optics". Theses. Sorbonne Université, July 2020 (cit. on p. 41).

[CK17] Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge: Cambridge University Press, 2017. ISBN: 978-1-107-10422-8. DOI: 10.1017/9781316219317 (cit. on p. 56).

[Cle+16] William R. Clements et al. "Optimal design for universal multiport interferometers". In: *Optica* 3.12 (Dec. 2016), pp. 1460–1465. DOI: 10.1364/OPTICA.3.001460 (cit. on pp. 40, 41, 213).

[Clé+22] Alexandre Clément et al. "LOv-Calculus: A Graphical Language for Linear Optical Quantum Circuits". In: *Proceedings of the 47th International Symposium on Mathematical Foundations of Computer Science (MFCS'22)*. Vol. 241. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 35:1–35:16. ISBN: 978-3-95977-256-3. DOI: 10.4230/LIPIcs.MFCS.2022.35 (cit. on pp. 12, 18, 111–113, 124, 132, 137, 170, 180).

[Clé+23] Alexandre Clément et al. "A Complete Equational Theory for Quantum Circuits". In: *Proceedings of the 38th Annual ACM IEEE Symposium on Logic in Computer Science (LICS'23)*. Boston, MA, USA, 2023. eprint: 2206.10577 (cit. on pp. 12, 18, 113, 169–171, 175, 178, 180, 183, 185).

[Clé+24] Alexandre Clément et al. "Quantum Circuit Completeness: Extensions and Simplifications". In: *32nd EACSL Annual Conference on Computer Science Logic (CSL 2024)*. Ed. by Aniello Murano and Alexandra Silva. Vol. 288. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, 20:1–20:23. ISBN: 978-3-95977-310-2. DOI: 10.4230/LIPIcs.CSL.2024.20 (cit. on p. 170).

[Clé23] Alexandre Clément. "Graphical Languages for Quantum Control and Linear Optics". Theses. Université de Lorraine, May 2023 (cit. on pp. 137, 178).

[CMA21] Gabriele Carcassi, Lorenzo Maccone and Christine A. Aidala. "Four Postulates of Quantum Mechanics Are Three". In: *Phys. Rev. Lett.* 126 (11 Mar. 2021), p. 110402. DOI: 10.1103/PhysRevLett.126.110402 (cit. on p. 26).

[CMS21] Ulysse Chabaud, Damian Markham and Adel Sohbi. "Quantum machine learning with adaptive linear optics". In: *Quantum* 5 (July 2021), p. 496. ISSN: 2521-327X. DOI: 10.22331/q-2021-07-05-496 (cit. on p. 60).

[Cos+23] N. Coste et al. "High-rate entanglement between a semiconductor spin and indistinguishable photons". In: *Nature Photonics* 17.7 (1st July 2023), pp. 582–587. ISSN: 1749-4893. DOI: 10.1038/s41566-023-01186-0 (cit. on p. 148).

[Cou85] Charles-Augustin Coulomb. "Premier mémoire sur l'électricité et le magnétisme". In: *Histoire de l'Académie Royale des Sciences* (1785), pp. 569–577 (cit. on p. 32).

[CP20] Alexandre Clément and Simon Perdrix. "PBS-Calculus: A Graphical Language for Coherent Control of Quantum Computations". In: *Proceedings of the 45th International Symposium on Mathematical Foundations of Computer Science (MFCS'20)*. Ed. by Javier Esparza and Daniel Kráľ. Vol. 170. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Aug. 2020, 24:1–24:14. ISBN: 978-3-95977-159-7. DOI: 10.4230/LIPIcs.MFCS.2020.24. eprint: 2002.09387 (quant-ph) (cit. on p. 112).

[CST89] Richard A. Campos, Bahaa E. A. Saleh and Malvin C. Teich. "Quantum-mechanical lossless beam splitter: SU(2) symmetry and photon statistics". In: *Phys. Rev. A* 40 (3 Aug. 1989), pp. 1371–1384. DOI: 10.1103/PhysRevA.40.1371 (cit. on p. 37).

[CW18] Bob Coecke and Quanlong Wang. "ZX-Rules for 2-Qubit Clifford+T Quantum Circuits". In: *Reversible Computation*. Ed. by Kazuo Iwama and Tiziana Margaria. Springer International Publishing, 2018, pp. 123–140. ISBN: 978-3-319-99497-0. DOI: 10.1007/978-3-319-99498-7_10 (cit. on p. 170).

[CZ95] J. I. Cirac and P. Zoller. "Quantum Computations with Cold Trapped Ions". In: *Phys. Rev. Lett.* 74 (20 May 1995), pp. 4091–4094. DOI: 10.1103/PhysRevLett.74.4091 (cit. on pp. 10, 16).

[DF25] Paul Adrien Maurice Dirac and Ralph Howard Fowler. "The fundamental equations of quantum mechanics". In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 109.752 (1925), pp. 642–653. DOI: 10.1098/rspa.1925.0150. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.1925.0150 (cit. on p. 25).

[DF28] Paul Adrien Maurice Dirac and Ralph Howard Fowler. "The quantum theory of the electron". In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 117.778 (1928), pp. 610–624. DOI: 10.1098/rspa.1928.0023. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.1928.0023 (cit. on p. 25).

[Dir27] Paul Adrien Maurice Dirac. "The quantum theory of the emission and absorption of radiation". In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 114.767 (1927), pp. 243–265. DOI: 10.1098/rspa.1927.0039 (cit. on p. 26).

[Dir30] Paul A. M. Dirac. *The Principles of Quantum Mechanics*. 1st. Oxford University Press, 1930 (cit. on pp. 23, 26).

[Dir39] Paul A. M. Dirac. "A new notation for quantum mechanics". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 35.3 (1939), pp. 416–418. DOI: 10.1017/S0305004100021162 (cit. on p. 22).

[DiV00] David P. DiVincenzo. "The Physical Implementation of Quantum Computation". In: *Fortschritte der Physik* 48.9 (Sept. 2000), pp. 771–783. ISSN: 00158208, 15213978. DOI: 10.1002/1521-3978(200009)48:9/11<771::AID-PROP771>3.0.CO;2-E. arXiv: quant-ph/0002077 (cit. on pp. 10, 15).

[DJ90] Nachum Dershowitz and Jean-Pierre Jouannaud. "Rewrite Systems". In: *Handbook of Theoretical Computer Science*. Elsevier, 1990, pp. 243–320 (cit. on p. 54).

[DL14] Ross Duncan and Maxime Lucas. "Verifying the Steane code with Quantomatic". In: *Electronic Proceedings in Theoretical Computer Science (EPTCS)* 171 (Dec. 2014), pp. 33–49. ISSN: 2075-2180. DOI: 10.4204/eptcs.171.4 (cit. on pp. 10, 16, 111).

[Don+22] Mark Dong et al. "High-speed programmable photonic circuits in a cryogenically compatible, visible–near-infrared 200 mm CMOS architecture". In: *Nature Photonics* 16.1 (2022), pp. 59–65. ISSN: 1749-4893. DOI: 10.1038/s41566-021-00903-x (cit. on p. 37).

[DP85] David Deutsch and Roger Penrose. "Quantum theory, the Church–Turing principle and the universal quantum computer". In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pp. 97–117. DOI: 10.1098/rspa.1985.0070 (cit. on pp. 9, 15).

[Dun+20] Ross Duncan et al. "Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus". In: *Quantum* 4 (June 2020), p. 279. ISSN: 2521-327X. DOI: 10.22331/q-2020-06-04-279 (cit. on p. 170).

[Ebe20] David Eberly. *Euler Angle Formulas*. Geometric Tools. 2020 (cit. on pp. 115, 117).

[Ein05] Albert Einstein. "On a heuristic point of view concerning the production and transformation of light". In: *Annals of Physics* 322.6 (1905), pp. 132–148. DOI: https://doi.org/10.1002/andp.19053220607 (cit. on pp. 23, 25, 33).

[Ein24] Albert Einstein. "Quantentheorie des einatomigen idealen Gases. Erste Abhandlung". In: *Sitzungsberichte der Preussischen Akademie der Wissenschaften, Physikalisch-Mathematische Klasse* (1924), pp. 261–267 (cit. on p. 25).

[Ein36] Albert Einstein. "Physik und Realität". In: *Journal of the Franklin Institute* 221.3 (1936), pp. 313–347. ISSN: 0016-0032. DOI: 10.1016/S0016-0032(36)91045-1 (cit. on p. 23).

[EPR35] Albert Einstein, Boris Podolsky and Nathan Rosen. "Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?" In: *Phys. Rev.* 47 (10 May 1935), pp. 777–780. DOI: 10.1103/PhysRev.47.777 (cit. on p. 23).

[FC22] Giovanni de Felice and Bob Coecke. "Quantum Linear Optics via String Diagrams". In: *Proceedings of the 19th International Workshop on Quantum Programming Languages (QPL'22)*. Vol. 394. Open Publishing Association, June 2022, pp. 83–100. DOI: 10.4204/eptcs.394.6 (cit. on pp. 112, 148).

[Fel+23] Giovanni de Felice et al. "Light-Matter Interaction in the ZXW Calculus". In: *Proceedings of the 20th International Workshop on Quantum Programming Languages (QPL'23)*. Vol. 384. Open Publishing Association, July 2023, pp. 20–46. DOI: 10.4204/eptcs.384.2 (cit. on p. 112).

[Fel+24] Giovanni de Felice et al. *Fusion and flow: formal protocols to reliably build photonic graph states*. 2024. arXiv: 2409.13541 [quant-ph] (cit. on p. 112).

[Fel68] William Feller. *An Introduction to Probability Theory and Its Applications*. Vol. 1. Wiley, 1968. ISBN: 0471257087 (cit. on p. 42).

[Fey+66] Richard P. Feynman et al. "The Feynman Lectures on Physics, Vol. 3: Quantum Mechanics". In: *Physics Today* 19.11 (Nov. 1966), pp. 80–83. ISSN: 0031-9228. DOI: 10.1063/1.3047826 (cit. on p. 23).

[Fey49] Richard P. Feynman. "Space-Time Approach to Quantum Electrodynamics". In: *Phys. Rev.* 76 (6 Sept. 1949), pp. 769–789. DOI: 10.1103/PhysRev.76.769 (cit. on p. 47).

[Fey82] Richard P. Feynman. "Simulating Physics with Computers". In: *International Journal of Theoretical Physics* 21.6 (June 1982), pp. 467–488. ISSN: 1572-9575. DOI: 10.1007/BF02650179 (cit. on pp. 9, 15).

[FGG14] Edward Farhi, Jeffrey Goldstone and Sam Gutmann. *A Quantum Approximate Optimization Algorithm*. 2014. arXiv: 1411.4028 [quant-ph] (cit. on pp. 9, 15).

[Fle13] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013 (cit. on p. 73).

[Foc32] Vladimir Fock. "Konfigurationsraum und zweite Quantelung". In: *Zeitschrift für Physik* 75.9 (1932), pp. 622–647. ISSN: 0044-3328. DOI: 10.1007/BF01344458 (cit. on p. 26).

[Fou22] J. Fourier. *Théorie analytique de la chaleur*. Paris: Chez Firmin Didot, père et fils, 1822 (cit. on p. 32).

[Fra22] Abraham Fraenkel. "Über die Grundlagen der Cantor-Zermeloschen Mengenlehre". In: *Mathematische Annalen* 86 (1922), pp. 230–237. DOI: 10.1007/BF01452247 (cit. on p. 55).

[FSK21] Suren A. Fldzhyan, Mikhail Yu. Saygin and Sergei P. Kulik. "Compact linear optical scheme for Bell state generation". In: *Physical Review Research* 3.4 (13th Oct. 2021), p. 043031. DOI: 10.1103/PhysRevResearch.3.043031 (cit. on p. 41).

[Fun+21] Lena Funcke et al. "Dimensional Expressivity Analysis of Parametric Quantum Circuits". In: *Quantum* 5 (Mar. 2021), p. 422. ISSN: 2521-327X. DOI: 10.22331/q-2021-03-29-422 (cit. on p. 211).

[Fyr+24] Andreas Fyrillas et al. "Scalable machine learning-assisted clear-box characterization for optimally controlled photonic circuits". en. In: *Optica* 11.3 (Mar. 2024), pp. 427–436. ISSN: 2334-2536. DOI: 10.1364/OPTICA.512148 (cit. on pp. 213–215).

[Fyr+25] Andreas Fyrillas et al. *Resource-efficient crosstalk mitigation for the high-fidelity operation of photonic integrated circuits with induced phase shifters*. 2025. arXiv: 2506.05988 [physics.optics] (cit. on pp. 13, 19, 169, 212, 217, 218).

[Fyr25] Andreas Fyrillas. "High-fidelity photonic quantum information: protocols and universal circuit control". PhD thesis. Université Paris-Saclay, Mar. 2025 (cit. on p. 213).

[GB17] Christian Gross and Immanuel Bloch. "Quantum simulations with ultracold atoms in optical lattices". In: *Science* 357.6355 (2017), pp. 995–1001. DOI: 10.1126/science.aal3837 (cit. on pp. 10, 16).

[GD18] Liam Garvie and Ross Duncan. "Verifying the Smallest Interesting Colour Code with Quantomatic". In: *Electronic Proceedings in Theoretical Computer Science (EPTCS)* 266 (Feb. 2018), pp. 147–163. ISSN: 2075-2180. DOI: 10.4204/eptcs.266.10 (cit. on pp. 10, 16, 111).

[GG46] Herman H. Goldstine and Adele Goldstine. "The electronic numerical integrator and computer (ENIAC)". In: *Mathematics of Computation* 2.15 (1946), pp. 97–110. DOI: 10.1090/S0025-5718-1946-0018977-0 (cit. on pp. 9, 15).

[GGJ77] M. R. Garey, R. L. Graham and D. S. Johnson. "The Complexity of Computing Steiner Minimal Trees". In: *SIAM Journal on Applied Mathematics* 32.4 (1977), pp. 835–859. DOI: 10.1137/0132072 (cit. on p. 85).

[GH25] Timothée Goubault de Brugière and Nicolas Heurtel. *Fast and memory efficient strong simulation of noisy adaptive linear optical circuits*. 2025. arXiv: 2503.05699 [quant-ph] (cit. on pp. 11, 17, 57, 61, 80).

[GIQ19] Brajesh Gupt, Josh Izaac and Nicolás Quesada. "The Walrus: a library for the calculation of hafnians, Hermite polynomials and gaussian boson sampling". In: *Journal of Open Source Software* 4.44 (2019), p. 1705 (cit. on pp. 10, 16, 57, 190).

[GLA22] Beng Yee Gan, Daniel Leykam and Dimitris G. Angelakis. "Fock state-enhanced expressivity of quantum machine learning models". In: *EPJ Quantum Technology* 9.1 (2022). DOI: 10.1140/epjqt/s40507-022-00135-0 (cit. on pp. 58, 73).

[Gli+24] Grégoire de Gliniasty et al. "A Spin-Optical Quantum Computing Architecture". In: *Quantum* 8 (July 2024), p. 1423. ISSN: 2521-327X. DOI: 10.22331/q-2024-07-24-1423 (cit. on pp. 10, 16, 57, 60).

[Gly10] David G. Glynn. "The permanent of a square matrix". In: *European Journal of Combinatorics* 31.7 (2010), pp. 1887–1891. ISSN: 0195-6698. DOI: 10.1016/j.ejc.2010.01.010 (cit. on pp. 45, 46, 74, 188).

[Gly13] David G. Glynn. "Permanent formulae from the Veronesean". In: *Designs, Codes and Cryptography* 68.1 (2013), pp. 39–47. ISSN: 1573-7586. DOI: 10.1007/s10623-012-9618-1 (cit. on p. 46).

[Göd31] Kurt Gödel. "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I". In: *Monatshefte für Mathematik und Physik* 38.1 (1931), pp. 173–198. ISSN: 1436-5081. DOI: 10.1007/BF01700692 (cit. on p. 55).

[Gou+25] Timothée Goubault de Brugière et al. *New designs of linear optical interferometers with minimal depth and component count*. 2025. arXiv: 2504.06059 [quant-ph] (cit. on p. 100).

[GPS02] Herbert Goldstein, Charles Poole and John Safko. *Classical Mechanics*. 3rd ed. Addison Wesley, 2002. ISBN: 0201657023 (cit. on pp. 115, 117).

[Gro96] Lov K. Grover. "A fast quantum mechanical algorithm for database search". In: *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC'96)*. New York, NY, USA: Association for Computing Machinery, July 1996, pp. 212–219. ISBN: 978-0-89791-785-8. DOI: 10.1145/237814.237866 (cit. on pp. 9, 15).

[GRS19] Raúl García-Patrón, Jelmer J. Renema and Valery Shchesnovich. "Simulating boson sampling in lossy architectures". In: *Quantum* 3 (Aug. 2019), p. 169. ISSN: 2521-327X. DOI: 10.22331/q-2019-08-05-169 (cit. on p. 45).

[Gub+20] F. V. Gubarev et al. "Improved heralded schemes to generate entangled states from single photons". In: *Physical Review A* 102.1 (2020), p. 012604. DOI: 10.1103/PhysRevA.102.012604 (cit. on p. 58).

[Gur05] Leonid Gurvits. "On the Complexity of Mixed Discriminants and Related Problems". In: *Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science (MFCS'05)*. Ed. by Joanna Jędrzejowicz and Andrzej Szepietowski. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 447–458. ISBN: 978-3-540-31867-5 (cit. on pp. 47, 190).

[Ham+17] Craig S. Hamilton et al. "Gaussian Boson Sampling". In: *Phys. Rev. Lett.* 119 (17 Oct. 2017), p. 170501. DOI: 10.1103/PhysRevLett.119.170501 (cit. on p. 43).

[Har+14] Nicholas C. Harris et al. "Efficient, compact and low loss thermo-optic phase shifter in silicon". In: *Opt. Express* 22.9 (May 2014), pp. 10487–10493. DOI: 10.1364/OE.22.010487 (cit. on pp. 37, 213).

[Har46] Douglas R. Hartree. "The ENIAC, An Electronic Calculating Machine". In: *Nature* 157.3990 (Apr. 1946), p. 527. ISSN: 1476-4687. DOI: 10.1038/157527a0 (cit. on pp. 9, 15).

[Hei27] Werner Heisenberg. "Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik". In: *Zeitschrift für Physik* 43.3 (Mar. 1927), pp. 172–198. ISSN: 0044-3328. DOI: 10.1007/BF01397280 (cit. on p. 24).

[Heu+23a] Nicolas Heurtel et al. "Perceval: A Software Platform for Discrete Variable Photonic Quantum Computing". In: *Quantum* 7 (Feb. 2023), p. 931. ISSN: 2521-327X. DOI: 10.22331/q-2023-02-21-931 (cit. on pp. 13, 18, 70, 73, 169, 186, 188).

[Heu+23b] Nicolas Heurtel et al. "Strong simulation of linear optical processes". In: *Computer Physics Communications* (2023), p. 108848. ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2023.108848 (cit. on pp. 11, 17, 57, 61).

[Heu25] Nicolas Heurtel. "A Complete Graphical Language for Linear Optical Circuits with Finite-Photon-Number Sources and Detectors". In: *33rd EACSL Annual Conference on Computer Science Logic (CSL 2025)*. Ed. by Jörg Endrullis and Sylvain Schmitz. Vol. 326. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025, 38:1–38:23. ISBN: 978-3-95977-362-1. DOI: 10.4230/LIPIcs.CSL.2025.38 (cit. on pp. 12, 18, 111, 112, 147, 180).

[HH15] Sarah Harris and David Harris. *Digital Design and Computer Architecture: ARM Edition*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2015. ISBN: 0128000562 (cit. on p. 28).

[HHL09] Aram W. Harrow, Avinatan Hassidim and Seth Lloyd. "Quantum Algorithm for Linear Systems of Equations". In: *Physical Review Letters* 103.15 (7th Oct. 2009), p. 150502. DOI: 10.1103/PhysRevLett.103.150502 (cit. on pp. 9, 15).

[HNW18] Amar Hadzihasanovic, Kang Feng Ng and Quanlong Wang. "Two complete axiomatisations of pure-state qubit quantum computing". In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'18)*. Oxford, United Kingdom: Association for Computing Machinery, 2018, pp. 502–511. ISBN: 9781450355834. DOI: 10.1145/3209108.3209128 (cit. on pp. 10, 16, 56, 111, 170).

[HOM87] Chung K. Hong, Zheyu Y. Ou and Leonard Mandel. "Measurement of subpicosecond time intervals between two photons by interference". In: *Phys. Rev. Lett.* 59 (18 Nov. 1987), pp. 2044–2046. DOI: 10.1103/PhysRevLett.59.2044 (cit. on p. 38).

[HR92] F. K. Hwang and Dana S. Richards. "Steiner tree problems". In: *Networks* 22.1 (1992), pp. 55–89. DOI: https://doi.org/10.1002/net.3230220105 (cit. on p. 83).

[HV19] Chris Heunen and Jamie Vicary. *Categories for Quantum Theory: An Introduction*. Oxford University Press, Nov. 2019. ISBN: 9780198739623. DOI: 10.1093/oso/9780198739623.001.0001. eprint: https://academic.oup.com/book/43710/book-pdf/50991591/9780191060069_web.pdf (cit. on p. 48).

[IKY02] Kazuo Iwama, Yasuaki Kambayashi and Shigeru Yamashita. "Transformation rules for designing CNOT-based quantum circuits". In: *Proceedings of the 39th Annual Design Automation Conference*. 2002, pp. 419–424 (cit. on p. 170).

[Jak+99] D. Jaksch et al. "Entanglement of Atoms via Cold Controlled Collisions". In: *Phys. Rev. Lett.* 82 (9 Mar. 1999), pp. 1975–1978. DOI: 10.1103/PhysRevLett.82.1975 (cit. on pp. 10, 16).

[Jim89] Michio Jimbo. "Introduction to the Yang-Baxter Equation". In: *International Journal of Modern Physics A* 04.15 (1989), pp. 3759–3777. DOI: 10.1142/S0217751X89001503 (cit. on p. 135).

[JPV18] Emmanuel Jeandel, Simon Perdrix and Renaud Vilmart. "A Complete Axiomatisation of the ZX-Calculus for Clifford+T Quantum Mechanics". In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'18)*. Oxford, United Kingdom: Association for Computing Machinery, 2018, pp. 559–568. ISBN: 9781450355834. DOI: 10.1145/3209108.3209131 (cit. on pp. 10, 16, 56, 111, 170).

[JS91] André Joyal and Ross Street. "The geometry of tensor calculus, I". In: *Advances in Mathematics* 88.1 (1991), pp. 55–112. ISSN: 0001-8708. DOI: https://doi.org/10.1016/0001-8708(91)90003-P (cit. on pp. 48, 50).

[JSV04] Mark Jerrum, Alistair Sinclair and Eric Vigoda. "A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries". In: *J. ACM* 51.4 (July 2004), pp. 671–697. ISSN: 0004-5411. DOI: 10.1145/1008731.1008738 (cit. on p. 47).

[JW28] P. Jordan and E. Wigner. "Über das Paulische Äquivalenzverbot". In: *Zeitschrift für Physik* 47.9 (1928), pp. 631–651. ISSN: 0044-3328. DOI: 10.1007/BF01331938 (cit. on p. 26).

[KGZ17] Mario Krenn, Xuemei Gu and Anton Zeilinger. "Quantum Experiments and Graphs: Multiparty States as Coherent Superpositions of Perfect Matchings". In: *Phys. Rev. Lett.* 119 (24 Dec. 2017), p. 240403. DOI: 10.1103/PhysRevLett.119.240403 (cit. on p. 112).

[Kil+19] Nathan Killoran et al. "Strawberry Fields: A Software Platform for Photonic Quantum Computing". In: *Quantum* 3 (Mar. 2019), p. 129. ISSN: 2521-327X. DOI: 10.22331/q-2019-03-11-129 (cit. on p. 187).

[Kim08] H. J. Kimble. "The quantum internet". In: *Nature* 453.7198 (June 2008), pp. 1023–1030. ISSN: 1476-4687. DOI: 10.1038/nature07127 (cit. on pp. 10, 16).

[Kja+20] Morten Kjaergaard et al. "Superconducting Qubits: Current State of Play". In: *Annual Review of Condensed Matter Physics* 11.1 (Mar. 2020), pp. 369–395. ISSN: 1947-5462. DOI: 10.1146/annurev-conmatphys-031119-050605 (cit. on pp. 10, 16).

[KLM01] E. Knill, R. Laflamme and G. J. Milburn. "A scheme for efficient quantum computation with linear optics". In: *Nature* 409.6816 (Jan. 2001), pp. 46–52. ISSN: 1476-4687. DOI: 10.1038/35051009 (cit. on pp. 10, 16, 41, 57, 60, 209).

[Kni02] E. Knill. "Quantum gates using linear optics and postselection". In: *Physical Review A* 66.5 (14th Nov. 2002), p. 052306. ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.66.052306. arXiv: quant-ph/0110144 (cit. on pp. 41, 59).

[Kok+07] Pieter Kok et al. "Review article: Linear optical quantum computing". In: *Reviews of Modern Physics* 79.1 (24th Jan. 2007), pp. 135–174. ISSN: 0034-6861, 1539-0756. DOI: 10.1103/RevModPhys.79.135. arXiv: quant-ph/0512071 (cit. on pp. 10, 16, 34, 209).

[KW20] Aleks Kissinger and John van de Wetering. "Reducing T-count with the ZX-calculus". In: *Physical Review A* 102 (2 Aug. 2020), p. 022406. DOI: 10.1103/PhysRevA.102.022406 (cit. on pp. 10, 16, 111).

[KW22] Aleks Kissinger and John van de Wetering. "Simulating quantum circuits with ZX-calculus reduced stabiliser decompositions". In: *Quantum Science and Technology* 7.4 (July 2022), p. 044001. ISSN: 2058-9565. DOI: 10.1088/2058-9565/ac5d20 (cit. on pp. 10, 16, 111).

[KWV22] Aleks Kissinger, John van de Wetering and Renaud Vilmart. "Classical Simulation of Quantum Circuits with Partial and Graphical Stabiliser Decompositions". In: *Proceedings of the 17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC'22)*. Ed. by François Le Gall and Tomoyuki Morimae. Vol. 232. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 5:1–5:13. ISBN: 978-3-95977-237-2. DOI: 10.4230/LIPIcs.TQC.2022.5 (cit. on pp. 10, 16, 111).

[KYW23] Mark Koch, Richie Yeung and Quanlong Wang. *Speedy Contraction of ZX Diagrams with Triangles via Stabiliser Decompositions*. 2023. arXiv: 2307.01803 [quant-ph] (cit. on pp. 10, 16, 111).

[Lan+10] B. P. Lanyon et al. "Towards quantum chemistry on a quantum computer". In: *Nature Chemistry* 2.2 (2010), pp. 106–111. DOI: 10.1038/nchem.483 (cit. on pp. 9, 15).

[Leo03] Ulf Leonhardt. "Quantum physics of simple optical instruments". In: *Reports on Progress in Physics* 66.7 (June 2003), p. 1207. DOI: 10.1088/0034-4885/66/7/203 (cit. on p. 37).

[Lew26] Gilbert N. Lewis. "The Conservation of Photons". In: *Nature* 118.2981 (Dec. 1926), pp. 874–875. ISSN: 1476-4687. DOI: 10.1038/118874a0 (cit. on p. 25).

[Li+20] Mingxiao Li et al. "Lithium niobate photonic-crystal electro-optic modulator". In: *Nature Communications* 11.1 (2020), p. 4123. ISSN: 2041-1723. DOI: 10.1038/s41467-020-17950-7 (cit. on pp. 37, 213).

[Lit20] Robert G. Littlejohn. *The Quantized Electromagnetic Field*. Physics 221B Lecture Notes, Notes 40. 2020 (cit. on p. 35).

[LL93] Arjen K. Lenstra and Hendrik W. Lenstra, eds. *The Development of the Number Field Sieve*. 1st ed. Lecture Notes in Mathematics. Springer Berlin, Heidelberg, 1993, pp. VIII, 140. ISBN: 978-3-540-57013-4. DOI: 10.1007/BFb0091534 (cit. on pp. 9, 15).

[Llo96] Seth Lloyd. "Universal Quantum Simulators". In: *Science* 273.5278 (1996), pp. 1073–1078. DOI: 10.1126/science.273.5278.1073 (cit. on pp. 9, 15).

[Lor+19] J. C. Loredo et al. "Generation of non-classical light in a photon-number superposition". In: *Nature Photonics* 13.11 (2019), pp. 803–808. DOI: 10.1038/s41566-019-0506-3 (cit. on pp. 60, 100, 148).

[Lou00] Rodney Loudon. *The Quantum Theory of Light*. Oxford University Press, Sept. 2000. ISBN: 9780198501770. DOI: 10.1093/oso/9780198501770.001.0001 (cit. on pp. 30, 37, 45).

[MA23] Jeffrey Marshall and Namit Anand. "Simulation of quantum optics by coherent state decomposition". In: *Optica Quantum* 1.2 (Dec. 2023), pp. 78–93. DOI: 10.1364/OPTICAQ.504311 (cit. on p. 42).

[Mac65] Saunders Mac Lane. "Categorical algebra". In: *Bulletin of the American Mathematical Society* 71 (1965), pp. 40–106. ISSN: 0002-9904. DOI: 10.1090/S0002-9904-1965-11234-4 (cit. on p. 48).

[Mac71] Saunders Mac Lane. *Categories for the Working Mathematician*. Springer New York, 1971. ISBN: 9781461298397. DOI: 10.1007/978-1-4612-9839-7 (cit. on p. 48).

[Mac92] Ernst Mach. "Ueber einen Interferenzrefraktor". In: *Zeitschrift für Instrumentenkunde* 12 (1892), pp. 89–93 (cit. on p. 38).

[Max65] James Clerk Maxwell. "A Dynamical Theory of the Electromagnetic Field". In: *Philosophical Transactions of the Royal Society of London* 155 (1865), pp. 459–512. DOI: 10.1098/rstl.1865.0008 (cit. on p. 32).

[McC22] Paul McCloud. *The Category of Linear Optical Quantum Computing*. 2022. arXiv: 2203.05958 [quant-ph] (cit. on p. 112).

[MG64] A. M. L. Messiah and O. W. Greenberg. "Symmetrization Postulate and Its Experimental Foundation". In: *Phys. Rev.* 136 (1B Oct. 1964), B248–B267. DOI: 10.1103/PhysRev.136.B248 (cit. on p. 25).

[MGM19] Lluís Masanes, Thomas D. Galley and Markus P. Müller. "The measurement postulates of quantum mechanics are operationally redundant". In: *Nature Communications* 10.1 (Mar. 2019), p. 1361. ISSN: 2041-1723. DOI: 10.1038/s41467-019-09348-x (cit. on p. 25).

[Mic81] A. A. Michelson. "The relative motion of the Earth and of the luminiferous ether". In: *American Journal of Science* s3-22.128 (Aug. 1881), pp. 120–129. DOI: 10.2475/ajs.s3-22.128.120 (cit. on p. 37).

[MM65] Marvin Marcus and Henryk Minc. "Permanents". In: *The American Mathematical Monthly* 72.6 (1965), pp. 577–591 (cit. on p. 83).

[MRS21] Jakub Makary, Neil J. Ross and Peter Selinger. "Generators and relations for real stabilizer operators". In: *Proceedings of the 18th International Conference on Quantum Physics and Logic (QPL'21)*. Ed. by Chris Heunen and Miriam Backens. Vol. 343. EPTCS. 2021, pp. 14–36 (cit. on p. 170).

[NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum computation and quantum information*. en. 10th anniversary edition. Cambridge, New York: Cambridge University Press, 2010. ISBN: 978-1-107-00217-3 (cit. on pp. 9, 10, 15, 21, 28, 29, 115).

[Neu32] John von Neumann. *Mathematische Grundlagen der Quantenmechanik*. ger. Translated by Robert T. Beyer as *Mathematical Foundations of Quantum Mechanics*, Princeton University Press, 1955. Springer, 1932 (cit. on p. 23).

[New42] M. H. A. Newman. "On Theories with a Combinatorial Definition of "Equivalence"". In: *Annals of Mathematics* 43.2 (1942), pp. 223–243 (cit. on p. 54).

[NG24] Samo Novák and Raúl García-Patrón. *Laplace expansions and tree decompositions: polytime algorithm for shallow nearest-neighbour Boson Sampling*. 2024. arXiv: 2412.18664 [quant-ph] (cit. on p. 44).

[Nie04] Michael A. Nielsen. "Optical Quantum Computation Using Cluster States". In: *Phys. Rev. Lett.* 93 (4 July 2004), p. 040503. DOI: 10.1103/PhysRevLett.93.040503 (cit. on pp. 10, 16).

[NPT99] Y. Nakamura, Yu. A. Pashkin and J. S. Tsai. "Coherent control of macroscopic quantum states in a single-Cooper-pair box". In: *Nature* 398.6730 (1999), pp. 786–788. ISSN: 1476-4687. DOI: 10.1038/19718 (cit. on pp. 10, 16).

[NW78] Albert Nijenhuis and Herbert S. Will. *Combinatorial Algorithms for Computers and Calculators*. Academic Press, 1978 (cit. on p. 45).

[OB18] Michał Oszmaniec and Daniel J Brod. "Classical simulation of photonic linear optics with lost particles". In: *New Journal of Physics* 20.9 (2018), p. 092002 (cit. on pp. 45, 95).

[OFV09] Jeremy L. O'Brien, Akira Furusawa and Jelena Vučković. "Photonic quantum technologies". In: *Nature Photonics* 3.12 (Dec. 2009), pp. 687–695. ISSN: 1749-4893. DOI: 10.1038/nphoton.2009.229 (cit. on pp. 10, 16).

[OG18] Andrea Olivo and Frédéric Grosshans. "Ancilla-assisted linear optical Bell measurements and their optimality". In: *Physical Review A* 98.4 (16th Oct. 2018), p. 042323. DOI: 10.1103/PhysRevA.98.042323 (cit. on p. 58).

[Oh+21] Changhun Oh et al. "Classical simulation of lossy boson sampling using matrix product operators". In: *Phys. Rev. A* 104 (2 Aug. 2021), p. 022407. DOI: 10.1103/PhysRevA.104.022407 (cit. on p. 190).

[Pae+24] A. Paetznick et al. *Demonstration of logical qubits and repeated error correction with better-than-physical error rates*. 2024. arXiv: 2404.02280 [quant-ph] (cit. on pp. 10, 16).

[Pau25] Wolfgang Pauli. "Über den Zusammenhang des Abschlusses der Elektronengruppen im Atom mit der Komplexstruktur der Spektren". In: *Zeitschrift für Physik* 31.1 (1925), pp. 765–783. DOI: 10.1007/BF02980631 (cit. on p. 25).

[Pau40] Wolfgang Pauli. "The Connection Between Spin and Statistics". In: *Phys. Rev.* 58 (8 Oct. 1940), pp. 716–722. DOI: 10.1103/PhysRev.58.716 (cit. on p. 25).

[Pen+24] Ciro Pentangelo et al. "High-fidelity and polarization-insensitive universal photonic processors fabricated by femtosecond laser writing". en. In: *Nanophotonics* (Jan. 2024). ISSN: 2192-8614. DOI: 10.1515/nanoph-2023-0636 (cit. on p. 213).

[Pen71a] Roger Penrose. "Angular momentum: an approach to combinatorial space-time". In: *Quantum Theory and Beyond*. Ed. by T. Bastin. Cambridge: Cambridge University Press, 1971, pp. 151–180 (cit. on p. 48).

[Pen71b] Roger Penrose. "Applications of Negative Dimensional Tensors". In: *Combinatorial Mathematics and its Applications*. Ed. by D. Welsh. New York: Academic Press, 1971, pp. 221–244 (cit. on p. 48).

[Per+14] Alberto Peruzzo et al. "A variational eigenvalue solver on a quantum processor". In: *Nature Communications* 5.1 (Sept. 2014), p. 4213. ISSN: 2041-1723. DOI: 10.1038/ncomms5213 (cit. on pp. 9, 15, 133).

[PH13] David A. Patterson and John L. Hennessy. *Computer Organization and Design, Fifth Edition: The Hardware/Software Interface*. 5th. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2013. ISBN: 0124077269 (cit. on pp. 9, 15, 28).

[Pla01] Max Planck. "Ueber das Gesetz der Energieverteilung im Normalspectrum". In: *Annalen der Physik* 309.3 (1901), pp. 553–563. DOI: https://doi.org/10.1002/andp.19013090310 (cit. on pp. 23, 33).

[Pre18] John Preskill. "Quantum Computing in the NISQ era and beyond". In: *Quantum* 2 (Aug. 2018). ISSN: 2521-327X. DOI: 10.22331/q-2018-08-06-79 (cit. on pp. 9, 15).

[Pre23] John Preskill. *Quantum computing 40 years later*. 2023. arXiv: 2106.10522 [quant-ph] (cit. on pp. 9, 15).

[PS20] João Paixão and Paweł Sobociński. "Calculational Proofs in Relational Graphical Linear Algebra". In: *Formal Methods: Foundations and Applications*. Ed. by Gustavo Carvalho and Volker Stolz. Cham: Springer International Publishing, 2020, pp. 83–100. ISBN: 978-3-030-63882-5 (cit. on p. 171).

[PSM87] Sudhakar Prasad, Marlan O. Scully and Werner Martienssen. "A quantum description of the beam splitter". In: *Optics Communications* 62.3 (1987), pp. 139–145. ISSN: 0030-4018. DOI: https://doi.org/10.1016/0030-4018(87)90015-0 (cit. on p. 37).

[Qi+20] Haoyu Qi et al. "Regimes of Classical Simulability for Noisy Gaussian Boson Sampling". In: *Phys. Rev. Lett.* 124 (10 Mar. 2020), p. 100502. DOI: 10.1103/PhysRevLett.124.100502 (cit. on p. 45).

[Ral+02] T. C. Ralph et al. "Linear Optical CNOT Gate in the Coincidence Basis". In: *Physical Review A* 65.6 (20th June 2002), p. 062324. ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.65.062324. arXiv: quant-ph/0112088 (cit. on pp. 41, 59, 209).

[RC14] Antoine Ranchin and Bob Coecke. "Complete set of circuit equations for stabilizer quantum mechanics". In: *Physical Review A* 90.1 (2014), p. 012109 (cit. on p. 170).

[Rec+94] Michael Reck et al. "Experimental realization of any discrete unitary operator". In: *Physical Review Letters* 73 (1st July 1994), pp. 58–61. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.73.58 (cit. on pp. 40, 41, 120, 199, 213).

[Ren+18] J. J. Renema et al. "Efficient Classical Algorithm for Boson Sampling with Partially Distinguishable Photons". In: *Phys. Rev. Lett.* 120 (22 May 2018), p. 220502. DOI: 10.1103/PhysRevLett.120.220502 (cit. on p. 45).

[RK23] Daniel Rehfeldt and Thorsten Koch. "Implications, Conflicts, and Reductions for Steiner Trees". In: *Mathematical Programming* 197 (2023), pp. 903–966. DOI: 10.1007/s10107-021-01757-5 (cit. on p. 87).

[RRC16] Saleh Rahimi-Keshari, Timothy C. Ralph and Carlton M. Caves. "Sufficient Conditions for Efficient Classical Simulation of Quantum Optics". In: *Phys. Rev. X* 6 (2 June 2016), p. 021039. DOI: 10.1103/PhysRevX.6.021039 (cit. on p. 45).

[RS72] Michael Reed and Barry Simon. *Methods of Modern Mathematical Physics, Volume I: Functional Analysis*. New York: Academic Press, 1972. ISBN: 978-0-12-585001-8. DOI: 10.1016/B978-0-12-585001-8.X5001-6 (cit. on p. 22).

[Rud17] Terry Rudolph. "Why I am optimistic about the silicon-photonic route to quantum computing". In: *APL Photonics* 2 (2017). DOI: 10.1063/1.4976737. eprint: 1607.08535 (cit. on pp. 10, 16).

[Rue11] Laura Ruetsche. *Interpreting Quantum Theories: The Art of the Possible*. Oxford University Press, 2011 (cit. on p. 23).

[Rys63] Herbert J. Ryser. *Combinatorial mathematics*. Vol. 14. The Carus Mathematical Monographs. American Mathematical Society, 1963. ISBN: 9781614440147 (cit. on pp. 45, 103, 188).

[Saf16] Mark Saffman. "Quantum computing with atomic qubits and Rydberg interactions: progress and challenges". In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 49.20 (Oct. 2016), p. 202001. DOI: 10.1088/0953-4075/49/20/202001 (cit. on pp. 10, 16).

[Sch04] Stefan Scheel. *Permanents in linear optical networks*. 2004. eprint: quant-ph/0406127 (cit. on pp. 10, 16, 42, 57).

[Sch11] Ulrich Schollwöck. "The density-matrix renormalization group in the age of matrix product states". In: *Annals of Physics* 326.1 (2011). January 2011 Special Issue, pp. 96–192. ISSN: 0003-4916. DOI: https://doi.org/10.1016/j.aop.2010.09.012 (cit. on p. 190).

[Sch35] Erwin Schrödinger. "Die gegenwärtige Situation in der Quantenmechanik". In: *Naturwissenschaften* 23.48 (Nov. 1935), pp. 807–812. ISSN: 1432-1904. DOI: 10.1007/BF01491891 (cit. on p. 23).

[Sch36] Erwin Schrödinger. "Probability relations between separated systems". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 32.3 (1936), pp. 446–452. DOI: 10.1017/S0305004100019137 (cit. on p. 23).

[Sch55] J. J. Schäffer. "On Unitary Dilations of Contractions". In: *Proceedings of the American Mathematical Society* 6.2 (1955). Accessed 25 Mar. 2025, pp. 322–322. DOI: 10.2307/2032368 (cit. on p. 43).

[Sel04] Peter Selinger. "Towards a quantum programming language". In: *Mathematical Structures in Computer Science* 14.4 (2004), pp. 527–586. DOI: 10.1017/S0960129504004256 (cit. on p. 134).

[Sel07] Peter Selinger. "Dagger Compact Closed Categories and Completely Positive Maps: (Extended Abstract)". In: *Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL'05)*. Vol. 170. Elsevier, 2007, pp. 139–163. DOI: 10.1016/j.entcs.2006.12.018 (cit. on p. 48).

[Sel11] Peter Selinger. "A Survey of Graphical Languages for Monoidal Categories". In: *New Structures for Physics*. Ed. by Bob Coecke. Vol. 813. Lecture Notes in Physics. Springer, 2011, pp. 289–355. ISBN: 978-3-642-12821-9. DOI: 10.1007/978-3-642-12821-9_4 (cit. on pp. 48, 50).

[SG21] Daniel Stilck França and Raul García-Patrón. "Limitations of optimization algorithms on noisy quantum devices". In: *Nature Physics* 17.11 (2021), pp. 1221–1227. DOI: 10.1038/s41567-021-01356-3 (cit. on p. 45).

[Sha+07] Wang Shao-Kai et al. "Realization of Arbitrary Inverse Unitary Transformation of Single Mode Fibre by Using Three Wave Plates". In: *Chinese Physics Letters* 24.9 (Sept. 2007), pp. 2471–2474. ISSN: 0256-307X, 1741-3540. DOI: 10.1088/0256-307X/24/9/003 (cit. on p. 39).

[Sha38] Claude E. Shannon. "A symbolic analysis of relay and switching circuits". In: *Transactions of the American Institute of Electrical Engineers* 57.12 (1938), pp. 713–723. DOI: 10.1109/T-AIEE.1938.5057767 (cit. on p. 28).

[Sha48] Claude E. Shannon. "A mathematical theory of communication". In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x (cit. on p. 28).

[Shc13] Valery Shchesnovich. "Asymptotic Evaluation of Bosonic Probability Amplitudes in Linear Unitary Networks in the Case of Large Number of Bosons". In: *International Journal of Quantum Information* 11.5, 1350045 (Sept. 2013), p. 1350045. DOI: 10.1142/S0219749913500457 (cit. on p. 46).

[Shc19] V. S. Shchesnovich. "Noise in boson sampling and the threshold of efficient classical simulatability". In: *Phys. Rev. A* 100 (1 July 2019), p. 012340. DOI: 10.1103/PhysRevA.100.012340 (cit. on p. 45).

[Shc20] Valery S. Shchesnovich. "On the classical complexity of sampling from quantum interference of indistinguishable bosons". In: *International Journal of Quantum Information* 18.07 (2020), p. 2050044. DOI: 10.1142/S0219749920500446. eprint: https://doi.org/10.1142/S0219749920500446 (cit. on pp. 10, 16, 46, 57).

[Shc21a] Valery Shchesnovich. "Distinguishing noisy boson sampling from classical simulations". In: *Quantum* 5 (2021), p. 423. DOI: 10.22331/q-2021-03-29-423 (cit. on p. 58).

[Shc21b] Valery Shchesnovich. *On the classical complexity of sampling from quantum interference of indistinguishable bosons*. 2021. arXiv: 1904.02013 [quant-ph] (cit. on p. 67).

[Shc22] Valery Shchesnovich. *Boson sampling cannot be faithfully simulated by only the lower-order multi-boson interferences*. 2022. arXiv: 2502.12882 [quant-ph] (cit. on p. 58).

[Sho94] Peter W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS)*. Nov. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700 (cit. on pp. 9, 15).

[Spa+13] Nicolò Spagnolo et al. "General Rules for Bosonic Bunching in Multimode Interferometers". In: *Phys. Rev. Lett.* 111 (13 Sept. 2013), p. 130503. DOI: 10.1103/PhysRevLett.111.130503 (cit. on p. 38).

[Spa+23] Nicolò Spagnolo et al. "Non-linear Boson Sampling". In: *npj Quantum Information* 9.1 (2023), p. 3. ISSN: 2056-6387. DOI: 10.1038/s41534-023-00676-x (cit. on p. 43).

[SS03] Elias M. Stein and Rami Shakarchi. *Fourier Analysis: An Introduction*. Vol. 1. Princeton Lectures in Analysis. Princeton, NJ: Princeton University Press, 2003. ISBN: 978-0-691-11384-5 (cit. on p. 32).

[Tar94] Alfred Tarski. *Introduction to Logic and to the Methodology of the Deductive Sciences*. 4th. Originally published in 1941. Oxford University Press, Jan. 1994. ISBN: 9780195044720. DOI: 10.1093/oso/9780195044720.001.0001 (cit. on p. 55).

[Tic11] Malte Christopher Tichy. "Entanglement and interference of identical particles". PhD thesis. Freiburg University, 2011 (cit. on p. 46).

[UG26] G. E. Uhlenbeck and S. Goudsmit. "Spinning Electrons and the Structure of Spectra". In: *Nature* 117.2938 (Feb. 1926), pp. 264–265. ISSN: 1476-4687. DOI: 10.1038/117264a0 (cit. on p. 25).

[Val24] Benoît Valiron. *Introduction to Quantum Algorithms and Quantum Programming*. Online Lecture Notes. 2024 (cit. on p. 21).

[Val79] Leslie G. Valiant. "The complexity of computing the permanent". en. In: *Theoretical Computer Science* 8.2 (1979), pp. 189–201. ISSN: 0304-3975. DOI: 10.1016/0304-3975(79)90044-6 (cit. on pp. 10, 16, 43, 45).

[Vil+22] Benjamin Villalonga et al. "Efficient approximation of experimental Gaussian boson sampling". In: *APS March Meeting Abstracts*. Vol. 2022. APS Meeting Abstracts. Mar. 2022, K40.002, K40.002 (cit. on p. 58).

[VV24] Marc de Visme and Renaud Vilmart. *Minimality in Finite-Dimensional ZW-Calculi*. 2024. arXiv: 2401.16225 [quant-ph] (cit. on p. 170).

[Wal+16] Mattia Walschaers et al. "Statistical benchmark for BosonSampling". In: *New Journal of Physics* 18.3 (2016), p. 032001. DOI: 10.1088/1367-2630/18/3/032001 (cit. on p. 58).

[Wan+20] Yuchen Wang et al. "Qudits and High-Dimensional Quantum Computing". In: *Frontiers in Physics* 8 (2020). ISSN: 2296-424X. DOI: 10.3389/fphy.2020.589504 (cit. on p. 28).

[WEH18] Stephanie Wehner, David Elkouss and Ronald Hanson. "Quantum internet: A vision for the road ahead". In: *Science* 362.6412 (2018), eaam9288. DOI: 10.1126/science.aam9288 (cit. on pp. 10, 16).

[Wet+24] John van de Wetering et al. *Optimal compilation of parametrised quantum circuits*. Jan. 2024. arXiv: 2401.12877 [quant-ph] (cit. on pp. 10, 16, 111).

[Wet20] John van de Wetering. *ZX-calculus for the working quantum computer scientist*. Dec. 2020. arXiv: 2012.13966 [quant-ph] (cit. on p. 56).

[Wig31] Eugene P. Wigner. *Gruppentheorie und ihre Anwendung auf die Quantenmechanik der Atomspektren*. F. Vieweg & Sohn Akt.-Ges., 1931 (cit. on p. 25).

[Wig39] E. Wigner. "On Unitary Representations of the Inhomogeneous Lorentz Group". In: *Annals of Mathematics* 40.1 (1939), pp. 149–204. ISSN: 0003486X, 19398980 (cit. on p. 25).

[YMQ24] Yuan Yao, Filippo Miatto and Nicolás Quesada. "Riemannian optimization of photonic quantum circuits in phase and Fock space". In: *SciPost Phys.* 17 (2024), p. 082. DOI: 10.21468/SciPostPhys.17.3.082 (cit. on p. 81).

[You04] Thomas Young. "Experiments and calculations relative to physical optics". In: *Philosophical Transactions of the Royal Society of London* 94 (1804), pp. 1–16. DOI: 10.1098/rstl.1804.0001 (cit. on pp. 32, 37).

[YR03] N. Yoran and B. Reznik. "Deterministic Linear Optics Quantum Computation with Single Photon Qubits". In: *Phys. Rev. Lett.* 91 (3 July 2003), p. 037903. DOI: 10.1103/PhysRevLett.91.037903 (cit. on pp. 10, 16).

[Zeh91] Ludwig Zehnder. "Ein neuer Interferenzrefraktor". In: *Zeitschrift für Instrumentenkunde* 11 (1891), pp. 275–285 (cit. on p. 38).

[Zei81] Anton Zeilinger. "General properties of lossless beam splitters in interferometry". In: *American Journal of Physics* 49.10 (1981), pp. 882–883. DOI: 10.1119/1.12387 (cit. on p. 37).

[Zer08] Ernst Zermelo. "Untersuchungen über die Grundlagen der Mengenlehre". In: *Mathematische Annalen* 65 (1908), pp. 261–281. DOI: 10.1007/BF01448712 (cit. on p. 55).

[Zer30] Ernst Zermelo. "Über Grenzzahlen und Bereiche von Mengen". In: *Fundamenta Mathematicae* 16 (1930), pp. 29–47 (cit. on p. 55).

[Zur03] Wojciech Hubert Zurek. "Decoherence, einselection, and the quantum origins of the classical". In: *Rev. Mod. Phys.* 75 (3 May 2003), pp. 715–775. DOI: 10.1103/RevModPhys.75.715 (cit. on p. 25).