PBS-Calculus: A Graphical Language for Coherent Control of Quantum Computations

Alexandre Clément

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
alexandre.clement@loria.fr

Simon Perdrix

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
simon.perdrix@loria.fr

Abstract

We introduce the PBS-calculus to represent and reason on quantum computations involving coherent control of quantum operations. Coherent control, and in particular indefinite causal order, is known to enable multiple computational and communication advantages over classically ordered models like quantum circuits. The PBS-calculus is inspired by quantum optics, in particular the polarising beam splitter (PBS for short). We formalise the syntax and the semantics of the PBS-diagrams, and we equip the language with an equational theory, which is proved to be sound and complete: two diagrams are representing the same quantum evolution if and only if one can be transformed into the other using the rules of the PBS-calculus. Moreover, we show that the equational theory is minimal. Finally, we consider applications like the implementation of controlled permutations and the unrolling of loops.

2012 ACM Subject Classification Theory of computation → Quantum computation theory; Theory of computation → Axiomatic semantics; Theory of computation → Categorical semantics; Hardware → Quantum computation; Hardware → Quantum communication and cryptography

Keywords and phrases Quantum Computing, Diagrammatic Language, Completeness, Quantum Control, Polarising Beam Splitter, Categorical Quantum Mechanics, Quantum Switch

Digital Object Identifier 10.4230/LIPIcs.MFCS.2020.24

Related Version A full version of the paper is available at `https://arxiv.org/abs/2002.09387` [9].

Funding This work is funded by ANR-17-CE25-0009 SoftQPro, ANR-17-CE24-0035 VanQuTe, PIA-GDN/Quantex, and LUE / UOQ.

Acknowledgements The authors want to thank Mehdi Mhalla, Emmanuel Jeandel and Titouan Carette for fruitful discussions. All diagrams were written with the help of TikZit.

1 Introduction

Quantum computers can solve problems which are out of reach of classical computers [28, 19]. One of the resources offered by quantum mechanics to speed up algorithms is the superposition phenomenon which allows a quantum memory to be in several possible classical states at the same time, in superposition. Less explored in quantum computing models, one can also consider a superposition of processes. Called *coherent control* or simply *quantum control*, it can be illustrated with the following example called quantum switch: the order in which two unitary evolutions U and V are applied is controlled by the state of a control qubit. In particular, if the control qubit is in superposition, then both UV and VU are applied, in superposition.

Coherent control is loosely represented in usual formalisms of quantum computing. For instance, in the quantum circuit model, the only available quantum control is the controlled gate mechanism: a gate U is applied or not depending on the state of a control qubit. The

45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020).
Editors: Javier Esparza and Daniel Král'; Article No. 24; pp. 24:1–24:14
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

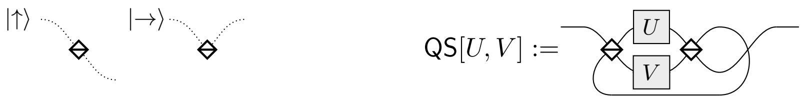


Figure 1 (a) Intuitive behaviour of a polarising beam splitter: vertical polarisation goes through, horizontal polarisation is reflected; (b) Quantum switch of two matrices U and V.

quantum switch cannot be implemented with a single copy of U and a single copy of V in the quantum circuit model, and more generally using any language with a fixed or classically controlled order of operations. Quantum switch has however been realised experimentally [24, 25]. Moreover, such a quantum control has been proved to enable various computational and communication advantages over classically ordered models [3, 15, 16, 17, 1], for instance for deciding whether two unitary transformations are commuting or anti-commuting [8] (see Example 12).

Notice that other models of quantum computations (e.g. Quantum Turing Machines) or programming languages (e.g. Lineal [13] or QML [2]), allow for arbitrary coherent control of quantum evolutions, the price to pay is, however, the presence of non-trivial well-formedness conditions to ensure that the represented evolution is valid. Indeed, the superposition (i.e. linear combination) of two unitary evolutions is not necessarily a unitary evolution.

We introduce a graphical language, the PBS-calculus, for representing coherent control of quantum computations, where arbitrary gates can be coherently controlled. Our goal is to provide the foundations of a formal framework which will be further developed to explore the power and limits of the coherent control of quantum evolutions. Contrary to the quantum circuit model, the PBS-calculus allows a representation of the quantum switch with a single copy of each gate to be controlled. Moreover, any PBS-diagram is valid by construction (no side nor well-formedness condition). The syntax of the PBS-diagrams is inspired by quantum optics and is actually already used in several papers dealing with coherent control of quantum evolutions [1, 3]. Our contribution is to provide formal syntax and semantics (both operational and denotational) for these diagrams, and also to introduce an equational theory which allows one to transform diagrams. Our main technical contribution is the proof that the equational theory is complete (if two diagrams have the same semantics then one can be transformed into the other using the equational theory) and minimal (in the sense that each of the equations is necessary for the completeness of the language).

The syntax of the PBS-calculus is inspired by linear optics, and in particular by the peculiar behaviour of the polarising beam splitter. A polarising beam splitter transforms a superposition of polarisations into a superposition of positions: if the polarisation is vertical the photon is transmitted whereas it is reflected when the polarisation is horizontal (see Figure 1.a). As a consequence a photon can be routed in different parts of a scheme, this routing being quantumly controlled by the polarisation of the photon. This is a unique behaviour which has no counterpart in the quantum circuit model for instance. Polarising beam splitters can be used to perform a quantum switch, as depicted as a PBS-diagram in Figure 1.b.

Related works. In the context of categorical quantum mechanics several graphical languages have already been introduced: ZX-calculus [10, 20], ZW-calculus [18], ZH-calculus [4] and their variants. Notice in particular a proposal for representing fermionic (non polarising) beam splitters in the ZW-calculus [12]. An apparent difference between the PBS-calculus and these languages, is that the category of PBS-diagrams is *traced* but not *compact closed*. This difference is probably not fundamental, as for any traced monoidal category there is a completion of it to a compact closed category [21]. The fundamental difference is the parallel composition: in the PBS-calculus two parallel wires correspond to two possible positions of a single particle (i.e. a direct sum in terms of semantics), whereas, in the other languages it corresponds to two particles (i.e. a tensor product).

The parallel composition makes the PBS-calculus closer to the *graphical linear algebra* approach [7, 6, 5], however the generators and the fundamental structures (e.g. Frobenius algebra, Hopf algebra) are *a priori* unrelated to those of the PBS-calculus.

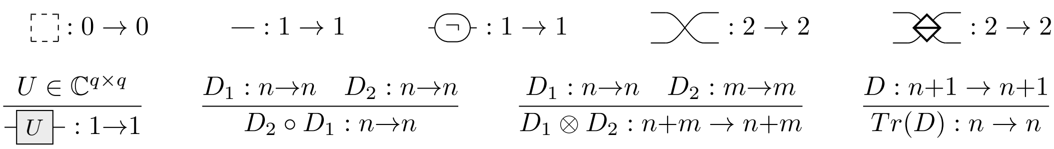
In the context of quantum programming languages, there are a few proposals for representing quantum control [13, 2, 29, 26]. Colnaghi et al. [11] have introduced a graphical language with *programmable connections*. The language uses the quantum switch as a generator, but does not aim to describe schemes with polarising beam splitters. Notice also that the inputs/outputs of the language are quantum channels.

Structure of the paper. In Section 2, the syntax of the PBS-diagrams is introduced. The PBS-diagrams are considered up to a structural congruence which allows one to deform the diagrams at will. Section 3 is dedicated to the semantics of the language: two semantics, a path semantics and a denotational semantics, are introduced. The denotational semantics is proved to be adequate with respect to the path semantics. In Section 4, the axiomatisation of the PBS-calculus is introduced, and our main result, the soundness and completeness of the language, is proved. The axiomatisation is also proved to be minimal in the sense that none of the axioms can be derived from the others. Finally, in Section 5, we consider the application of the PBS-calculus to the problem of loop unrolling. We show in particular that any PBS-diagram involving unitary matrices can be transformed into a trace-free diagram. The paper is written such that the reader does not need any particular knowledge in category theory. Basic definitions, in particular of Traced PROP, are however given for completeness in the full version of the paper [9], as well as all omitted proofs.

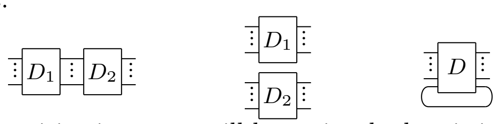
2 Syntax

A PBS-diagram is made of polarising beam splitters , polarisation flips , and gates U for any matrix $U \in \mathbb{C}^{q\times q}$, where q is a fixed positive integer. One can also use wires like the identity — or the swap . Diagrams can be combined by means of sequential composition $\circ$, parallel composition $\otimes$, and trace $Tr(\cdot)$. The trace consists in connecting the last output of a diagram to its last input, like a feedback loop. The symbol represents the empty diagram. Any diagram has a type $n \to n$ which corresponds to the numbers of input/output wires. The syntax of the language is the following:

▶ **Definition 1.** *Given $q \in \mathbb{N} \setminus \{0\}$, a PBS_q-diagram $D : n \to n$ is inductively defined as:*

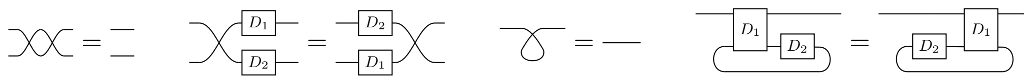


Sequential composition $D_2 \circ D_1$, parallel composition $D_1 \otimes D_2$, and trace $Tr(D)$ are respectively depicted as follows:



In the following, the positive integer q will be omitted when it is useless or clear form the context.

Notice that two distinct terms, like $\neg \circ (U \circ \neg)$ and $(\neg \circ U) \circ \neg$, can lead to the same graphical representation: $\neg\, U\, \neg$. To avoid ambiguity, we define diagrams modulo a structural congruence detailed in the full version of the paper [9]. Roughly speaking the structural congruence guarantees that (i) two terms leading to the same graphical representation are equivalent, and (ii) a diagram can be deformed at will, e.g.:



In the categorical framework of PROP [23, 30], PBS-diagrams modulo the structural congruence form a Traced PROP, i.e. they are morphisms of a traced strict symmetric monoidal category whose objects are natural numbers. It is known (Theorem 20[1] of [27]) that two diagrams are equivalent according to the axioms of a traced PROP if and only if they are isomorphic in a graph-theoretical sense, that is, if one can be obtained from the other by moving, stretching and reorganising the wires in any way, while keeping their two ends fixed.

3 Semantics

In this section, we introduce the semantics of the PBS-diagrams. First, we introduce an operational semantics for PBS-diagrams with a classical control. The operational semantics, called *path semantics* is based on the graphical intuition of a routed particle. Then we introduce a denotational semantics for the general case, with a quantum control. We show the adequacy between the two semantics, providing a graphical way to compute the denotational semantics of a PBS-diagram.

In this paper, we only consider the case where a *single* particle, say a photon, is present in the diagram. The particle is made of a polarisation and an additional data register. The particle has: an initial polarisation, which is an arbitrary superposition of the horizontal $(\rightarrow)$ and vertical $(\uparrow)$ polarisations (that we call *classical* polarisations in the following); an arbitrary position, which is a superposition of the possible input wires of the diagram; and an input data state, which is a vector $\varphi \in \mathbb{C}^q$.

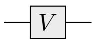
3.1 Classical control – Path semantics

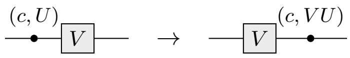
Classical control. We first consider input particles with a classical polarisation and a classical position. Roughly speaking, the particle is initially located on one of the input wires with a given polarisation in $\{\rightarrow, \uparrow\}$, and moves through the diagram depending on its polarisation. The action of a PBS-diagram can be *informally* described as follows using a token made of the current polarisation c of the particle and a matrix U representing the matrix applied so far to the data register:

- The particle is either reflected or transmitted by a beam splitter, depending on its polarisation:

[1] Notice that in [27], the author points out that this result relies on a result by Kelly and Laplaza (Theorem 8.2, [22]) which is only proven for simple signatures – which is not the case for the PBS-diagrams. The general case does not appear in the literature.

- The polarisation may vary but remains classical (that is, in $\{\rightarrow,\uparrow\}$) as the polarisation flip – the only one which acts on the polarisation – interchanges horizontal and vertical polarisations:

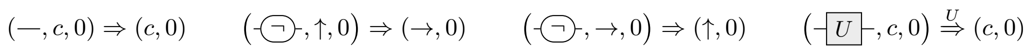
-  acts on the data register, transforming the state φ into $V\varphi$:

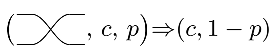


- The particle can freely move through wires, e.g.:

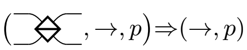
Thus the token follows a path from the input to the output and accumulates a matrix along the path. We formalise this intuitive behaviour as a big-step operational semantics that we call *path semantics* in this context. A *configuration* is a triplet (D, c, p), where $D : n \to n$ is a PBS-diagram, $c \in \{\rightarrow,\uparrow\}$ is the input polarisation of the particle, and $p \in [n] := \{0, \ldots, n-1\}$ its input position: 0 means that the particle is located on the first upper input wire, 1 on the second one and so on. The result is made of the final polarisation c' and position p', and of the matrix U representing the overall action of D on the data register.

▶ **Definition 2** (Path semantics). *Given a PBS-diagram $D : n \to n$, a polarisation $c \in \{\rightarrow,\uparrow\}$ and a position $p \in [n]$, let $(D, c, p) \overset{U}{\Rightarrow} (c', p')$ (or simply $(D, c, p) \Rightarrow (c', p')$ when U is the identity) be inductively defined as follows:*

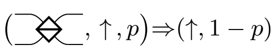




$$\frac{(D_1, c, p) \overset{U}{\Rightarrow} (c', p') \qquad (D_2, c', p') \overset{V}{\Rightarrow} (c'', p'')}{(D_2 \circ D_1, c, p) \overset{VU}{\Longrightarrow} (c'', p'')}(\circ)$$



$$\frac{D_1 : n \to n \qquad p < n \qquad (D_1, c, p) \overset{U}{\Rightarrow} (c', p')}{(D_1 \otimes D_2, c, p) \overset{U}{\Rightarrow} (c', p')}(\otimes 1)$$



$$\frac{D_1 : n \to n \qquad p \geq n \qquad (D_2, c, p - n) \overset{U}{\Rightarrow} (c', p')}{(D_1 \otimes D_2, c, p) \overset{U}{\Rightarrow} (c', p' + n)}(\otimes 2)$$

$$\frac{D : n \to n \qquad \forall i \in \{0, \ldots, k\}, (D, c_i, p_i) \overset{U_i}{\Longrightarrow} (c_{i+1}, p_{i+1}) \qquad (p_{i+1} = n) \Leftrightarrow (i < k)}{(Tr(D), c_0, p_0) \xrightarrow{U_k \cdots U_0} (c_{k+1}, p_{k+1})}(\mathsf{T}_k)$$

with $k \in \{0, 1, 2\}$.

▶ **Example 3.** As expected, the path semantics of the quantum switch $\mathsf{QS}[U,V] := Tr\,(\cdots \circ \cdots \circ (\text{-}\boxed{U}\text{-} \otimes \text{-}\boxed{V}\text{-}) \circ \cdots)$ (see Figure 1.b) is $(\mathsf{QS}[U,V], \uparrow, 0) \overset{UV}{\Longrightarrow} (\uparrow, 0)$ and $(\mathsf{QS}[U,V], \rightarrow, 0) \overset{VU}{\Longrightarrow} (\rightarrow, 0)$.

▶ **Example 4.** PBS-diagrams implementing a controlled permutation are given in Figure 2.

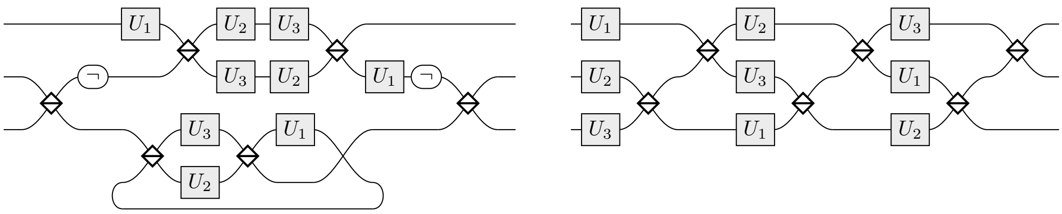


Figure 2 Two diagrams having the same semantics, that implement a controlled permutation of 3 unitary maps. Given a permutation (xyz) of (123), we have $(D,c,x) \xrightarrow{U_zU_yU_x} (c,x)$, where D is any of the two diagrams and $c = \rightarrow$ if the signature of (xyz) is 1, $c = \uparrow$ otherwise. A generalisation to the controlled permutation of n unitary maps is given in the full version of the paper [9].

Notice that the path semantics does not need to be defined for the empty diagram. Indeed, for any diagram $D : 0 \to 0$ there is no valid configuration (D, c, p) as p should be one of the input wires of D.

The (T_k)-rule is parametrised by an integer k. Intuitively, this parameter is the number of times the photon goes through the corresponding trace. We show in the following that roughly speaking, a particle can never go through a given trace more than twice. In other words, the path semantics which assumes $k \leq 2$, is well-defined for any valid configuration:

▶ **Proposition 5.** *For any diagram $D : n \to n$ and any $(c,p) \in \{\rightarrow,\uparrow\} \times [n]$, there exist unique $(c',p') \in \{\rightarrow,\uparrow\} \times [n]$ and $U \in \mathbb{C}^{q\times q}$ such that $(D,c,p) \overset{U}{\Rightarrow} (c',p')$.*

In the previous proposition, uniqueness means that the path semantics is deterministic: since diagrams are considered modulo structural congruence (i.e. up to deformation), it implies that these deformations preserve the path semantics.

Moreover, all PBS-diagrams are invertible in the following sense:

▶ **Proposition 6.** *For any diagram $D : n \to n$ and any $(c,p) \in \{\rightarrow,\uparrow\} \times [n]$, there exist unique $(c',p') \in \{\rightarrow,\uparrow\} \times [n]$ and $U \in \mathbb{C}^{q\times q}$ such that $(D,c',p') \overset{U}{\Rightarrow} (c,p)$.*

As a consequence, any diagram $D : n \to n$ essentially acts as a permutation on $\{\rightarrow,\uparrow\}\times[n]$, if one ignores its action on the data register. We introduce dedicated notations for representing the corresponding permutation, as well as the actions on the data register:

▶ **Definition 7.** *For any diagram $D : n \to n$, we call τ_D the permutation of $\{\rightarrow,\uparrow\} \times [n]$ and for any $c,p \in \{\rightarrow,\uparrow\}\times[n]$, we call $[D]_{c,p} \in \mathbb{C}^{q\times q}$ the matrix such that $(D,c,p) \xrightarrow{[D]_{c,p}} \tau_D(c,p)$.*

In a PBS-diagram, the particle can go through each wire at most twice, otherwise, roughly speaking, it would go back to the same position with the same polarisation and thus will come back again and again to this same configuration and thus enter an infinite loop – which is prevented by Proposition 5. In particular, each gate of the diagram is visited at most twice:

▶ **Proposition 8.** *Any gate U of a diagram D contributes to at most two paths $[D]_{c_0,p_0}$ and $[D]_{c_1,p_1}$, i.e. given D' the diagram D where one occurrence of U has been replaced by an arbitrary matrix V, $\forall(c,p) \notin \{(p_0,c_0),(p_1,c_1)\}$, $[D]_{c,p} = [D']_{c,p}$.*

Proof. The proof is straightforward by induction on D. ◀

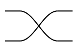
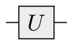
As a consequence the diagrams of Figure 2 are optimal in the number of uses of each U_i: since each of the 6 paths must depend on each U_i, at least three copies of each U_i are required in a diagram which solves the permutation problem of 3 unitaries.

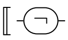
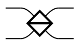
3.2 Quantum control – Denotational semantics

A crucial property of PBS-diagram is to offer the ability to have a quantum control, i.e. a particle whose input state is a superposition of polarisations, positions, or both. To encounter the quantum control, we introduce in this section a denotational semantics which associates with any diagram a map acting on the state space $\mathcal{H}_n := \mathbb{C}^{\{\rightarrow,\uparrow\}} \otimes \mathbb{C}^n \otimes \mathbb{C}^q$. Using Dirac notations, $\{|\rightarrow\rangle, |\uparrow\rangle\}$ (resp. $\{|x\rangle \mid x \in \{0 \dots k-1\}\}$) is an orthonormal basis of $\mathbb{C}^{\{\rightarrow,\uparrow\}}$ (resp. $\mathbb{C}^k$). Thus $\{|c,p,x\rangle \mid c \in \{\rightarrow,\uparrow\}, p \in [n], x \in [q]\}$ is an orthonormal basis of $\mathcal{H}_n$.

▶ **Definition 9.** *The denotational semantics of a* PBS*-diagram $D : n \to n$ is the linear map $[\![D]\!] : \mathcal{H}_n \to \mathcal{H}_n$ inductively defined as follows:*

$[\![$  $]\!] = 0$ $\qquad$ $[\![—]\!] = |c,0,x\rangle \mapsto |c,0,x\rangle$

$[\![$  $]\!] = |c,p,x\rangle \mapsto |c,1-p,x\rangle$ $\qquad$ $[\![$  $]\!] = |c,0,x\rangle \mapsto |c,0\rangle \otimes U\,|x\rangle$

$[\![$  $]\!] = \begin{cases} |\rightarrow,0,x\rangle \mapsto |\uparrow,0,x\rangle \\ |\uparrow,0,x\rangle \mapsto |\rightarrow,0,x\rangle \end{cases}$ $\qquad$ $[\![$  $]\!] = \begin{cases} |\rightarrow,p,x\rangle \mapsto |\rightarrow,p,x\rangle \\ |\uparrow,p,x\rangle \mapsto |\uparrow,1-p,x\rangle \end{cases}$

$$[\![D_2 \circ D_1]\!] = [\![D_2]\!] \circ [\![D_1]\!] \qquad [\![D_1 \otimes D_2]\!] = [\![D_1]\!] \boxplus [\![D_2]\!] \qquad [\![Tr(D)]\!] = \mathcal{T}([\![D]\!])$$

where:

- $f \boxplus g := \varphi \circ (f \oplus g) \circ \varphi^{-1}$ *with* $\varphi\colon \mathcal{H}_n \oplus \mathcal{H}_m \to \mathcal{H}_{n+m}$ *the isomorphism defined as* $(|c,p,x\rangle, |c',p',x'\rangle) \mapsto |c,p,x\rangle + |c',p'+n,x'\rangle$.
- $\mathcal{T}(f) := \sum_{k\in\mathbb{N}} \pi_1 \circ (f \circ \pi_0)^k \circ f \circ \iota$ *with* $\iota : \mathcal{H}_n \to \mathcal{H}_{n+1} :: |c,x,y\rangle \mapsto |c,x,y\rangle$, $\pi_0 : \mathcal{H}_{n+1} \to \mathcal{H}_{n+1} :: |c,x,y\rangle \mapsto \begin{cases} 0 & \text{if } x < n \\ |c,n,y\rangle & \text{if } x = n \end{cases}$, *and* $\pi_1 : \mathcal{H}_{n+1} \to \mathcal{H}_n :: |c,x,y\rangle \mapsto \begin{cases} |c,x,y\rangle & \text{if } x < n \\ 0 & \text{if } x = n. \end{cases}$

Notice that while the semantics of the trace is defined by means of an infinite sum, this sum is actually made of a finite number of nonzero elements, which guarantees that the denotational semantics is well-defined:

▶ **Proposition 10.** *For any diagram $D : n \to n$, $[\![D]\!] \in \mathcal{SLP}_n$, where $\mathcal{SLP}_n$ is the monoid of the linear maps $f\colon \mathcal{H}_n \to \mathcal{H}_n$ such that $f\,|c,p,x\rangle = |\tau(c,p)\rangle \otimes U_{c,p}\,|x\rangle$ for some permutation τ on $\{\rightarrow,\uparrow\} \times [n]$ and matrices $U_{c,p} \in \mathbb{C}^{q\times q}$.*

The denotational semantics is adequate with respect to the path semantics:

▶ **Theorem 11** (Adequacy). *For any $D : n \to n$, $[\![D]\!] = |c,p,x\rangle \mapsto |\tau_D(c,p)\rangle \otimes [D]_{c,p}\,|x\rangle$, where τ_D and $[D]_{c,p}$ are such that $(D,c,p) \xRightarrow{[D]_{c,p}} \tau_D(c,p)$*

The adequacy theorem implies that two diagrams have the same denotational semantics if and only if they have the same path semantics. As a consequence, it provides a graphical characterisation of the denotational semantics. Indeed, for any diagram $D : n \to n$, $[\![D]\!]$ is, by linearity, entirely defined by τ_D and $\{[D]_{c,p}\}_{c\in\{\to,\uparrow\},p\in[n]}$. Since τ_D and $[D]_{c,p}$ have a nice graphical interpretation as paths from the inputs to the outputs, the adequacy theorem provides a graphical way to compute the denotational semantics of any PBS-diagram.

▶ **Example 12.** The quantum switch (Figure 1.b and Example 3) can be used to decide whether U and V are commuting or anti-commuting [8]. The semantics of the quantum switch is $[\![\mathsf{QS}[U,V]]\!] = \begin{cases} |\to,0,x\rangle \mapsto |\to,0\rangle \otimes VU\,|x\rangle \\ |\uparrow,0,x\rangle \mapsto |\uparrow,0\rangle \otimes UV\,|x\rangle \end{cases}$. We assume that $UV = (-1)^k VU$ and call the quantum switch with a control qubit in a uniform superposition: $[\![\mathsf{QS}[U,V]]\!]\,\frac{|\to\rangle+|\uparrow\rangle}{\sqrt{2}} \otimes |0,x\rangle = \frac{|\to,0\rangle\otimes VU|x\rangle+|\uparrow,0\rangle\otimes UV|x\rangle}{\sqrt{2}} = \frac{|\to,0\rangle\otimes VU|x\rangle+(-1)^k|\uparrow,0\rangle\otimes VU|x\rangle}{\sqrt{2}} = \frac{|\to\rangle+(-1)^k|\uparrow\rangle}{\sqrt{2}} \otimes VU\,|0,x\rangle$. Thus, by measuring the control qubit in the $\{\frac{|\to\rangle+|\uparrow\rangle}{\sqrt{2}}, \frac{|\to\rangle-|\uparrow\rangle}{\sqrt{2}}\}$-basis, one can decide whether U and V are commuting or anti-commuting.

4 Equational theory – PBS-calculus

The representation of a quantum computation using PBS-diagrams is not unique, in the sense that two distinct PBS-diagrams may have the same semantics (e.g. diagrams of Figure 2). In this section, we introduce 10 equations on PBS-diagrams (see Figure 3) as the axioms of a language that we call the PBS-calculus. We prove that the PBS-calculus is sound (that is, consistent with the semantics), complete (that is, it captures entirely the semantic equivalence) and minimal (that is, all axioms are necessary to have completeness). Completeness is proved by means of a normal form.

4.1 Axiomatisation

▶ **Definition 13** (PBS-calculus). *Two* PBS-*diagrams* D_1, D_2 *are equivalent according to the rules of the* PBS-*calculus, denoted* PBS $\vdash D_1 = D_2$*, if one can transform* D_1 *into* D_2 *using the equations given in Figure 3. More precisely,* PBS $\vdash \cdot = \cdot$ *is defined as the smallest congruence*[2] *which satisfies the equations of Figure 3.*

Equations (1) and (6) in Figure 3 reflect the monoidal structure of the matrices, with the identity element (Equation (1)) and the associative binary operation (Equation (6)). Equations (2) and (3) mean that both the polarising beam splitter and the polarisation flip commute with a gate. Moreover, the polarising beam splitter is self inverse (Equation (8)). Notice that the negation is also self-inverse and that this is a consequence of the axioms (see Example 14). Equation (5) translates the fact that flipping the control state before and after performing a control of the position results in flipping the final position. To give a meaning to Equation (10), it is useful to flip it upside down, and to remark that in a two-wire diagram, polarising beam splitters and negations on the bottom wire each perform a CNOT on the qubits representing the polarisation and the position, in opposite ways, so that each side of the equation combines 3 CNOTs and thus performs a swap between these two qubits. In Equation (4), there are essentially two steps: first, the wire with the gate V is a dead code, as no photon can go to the wire, so it can be discarded; the second step consists in merging the two polarising beam splitters. Equation (9) is the only equation acting on three wires:

[2] see the full version of the paper [9] for a formal definition of congruence in this context. Notice that any congruence has to be consistent with the structural congruence in order to be well-defined.

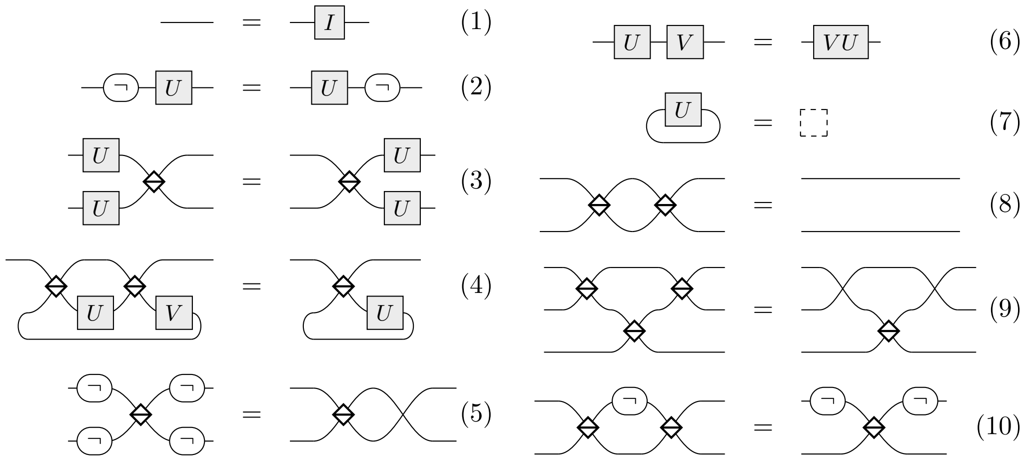


Figure 3 Axioms of the PBS-calculus. Given q a positive integer, $U, V \in \mathbb{C}^{q\times q}$ are arbitrary matrices, $I \in \mathbb{C}^{q\times q}$ is the identity.

in this particular configuration given by the left hand side of the equation, two polarising beam splitters can be replaced by swaps. Equation (7) reflects the fact that isolated parts of a diagram have no effect on the rest.

▶ **Example 14.** The fact that the negation is self inverse can be derived in the PBS-calculus: PBS ⊢ -(¬)-(¬)- = —. A more sophisticated example is the proof that the two diagrams of Figure 2 are equivalent. The derivations are given in the full version of the paper [9].

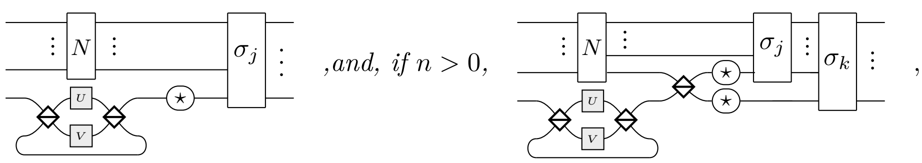
All these equations preserve the semantics of the PBS-diagrams:

▶ **Proposition 15** (Soundness). *For any two diagrams D_1 and D_2, if PBS ⊢ $D_1 = D_2$ then* $[\![D_1]\!] = [\![D_2]\!]$.

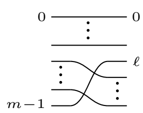
4.2 Normal forms

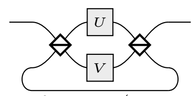
In this section, we introduce a notion of diagrams in normal form which is used in the next sections to prove both the universality and the completeness of the PBS-calculus. They are made of two parts: the first one corresponds to a superposition of linear maps, and the second one corresponds to a permutation of the polarisations and positions, written in a way that is convenient here.

▶ **Definition 16** (Normal Form). *Diagrams in normal form are inductively defined as:* ⬚ *is in normal form, and for any $N : n \to n$ in normal form,*



are in normal form, where -(⋆)- *denotes either* — *or* -(¬)- *, and $\sigma_\ell : m \to m =$*



▶ Remark 17. For any $U, V \in \mathbb{C}^{q\times q}$ let $\mathrm{E}(U,V) :=$  . A diagram in normal form can be written in the form $P \circ E$, where E is of the form $E(U_0, V_0) \otimes \cdots \otimes E(U_{n-1}, V_{n-1})$, and P is built using only — , , , , $\circ$ and $\otimes$.

In the following we show that any diagram is equivalent to a diagram in normal form.

▶ **Lemma 18.** *If N_1 and N_2 are in normal form then $N_1 \otimes N_2$ is in normal form.*

Proof. By definition of the normal forms. ◀

▶ **Lemma 19.** *If $N_1 : n \to n$ and $N_2 : n \to n$ are in normal form then there exists $N' : n \to n$ in normal form such that* $\mathrm{PBS} \vdash N_2 \circ N_1 = N'$.

Proof. Notice that using the axioms of PROP, $N_2 = g_\ell \circ \ldots \circ g_0$ where each g_k consists of either $E(U,V)$, —, , or acting on any one or two consecutive positions, in parallel with the identity on the other positions. We show that every g_k can be successively integrated to the normal form (see the full version of the paper [9] for details). ◀

▶ **Lemma 20.** *If $N : n+1 \to n+1$ is in normal form then there exists $N' : n \to n$ in normal form such that* $\mathrm{PBS} \vdash Tr(N) = N'$.

We are now ready to prove that any PBS-diagram can be put in normal form:

▶ **Proposition 21.** *For any $D : n \to n$, there exists a* PBS*-diagram $N : n \to n$ in normal form such that* $\mathrm{PBS} \vdash D = N$.

Proof. Combining the previous three lemmas, it remains to prove that any generator of the language can be put in normal form. We do so in the full version of the paper [9]. ◀

▶ Remark 22. By unfolding the proof of Proposition 21, one can obtain a deterministic procedure to transform any diagram into its normal form. Its complexity, defined as the number of transformations by one of Equations (1) to (10), is $\mathcal{O}(tm^2)$, where m is the number of generators (, , and), and t the number of traces in the diagram. Notice that this procedure has probably not the best possible complexity.

4.3 Completeness

The main application of the normal forms is the proof of completeness:

▶ **Theorem 23** (Completeness). *For any $D, D' : n \to n$, if $[\![D]\!] = [\![D']\!]$ then* $\mathrm{PBS} \vdash D = D'$.

Proof. There exist N, N' in normal form such that $\mathrm{PBS} \vdash D = N$ and $\mathrm{PBS} \vdash D' = N'$. Moreover, by soundness, $[\![N]\!] = [\![D]\!] = [\![D']\!] = [\![N']\!]$. Finally, one can show that $[\![N]\!] = [\![N']\!]$ implies that $N = N'$. In particular, one can show inductively that the normal form is entirely determined by its semantics by considering the path semantics for a particle located on the last input wire. ◀

4.4 Minimality of the set of axioms

In the following we show that each of the ten equations of Figure 3 is necessary for the completeness of the PBS-calculus:

▶ **Theorem 24** (Minimality). *None of Equations* (1) *to* (10) *is a consequence of the others.*

Notice that all equations involving matrices, except Equation (1), are schemes of equations i.e. one equation for each possible matrix (or matrices). In Theorem 24, we show that each of these equations, for most of the matrices, cannot be derived from the other axioms. More precisely, Equation (4) (resp. (7)) is not a consequence of the nine others for any U (resp. any U, V); Equation (2) (resp. (6)) is not a consequence of the others for any $U \neq I$ (resp. any $U, V \neq I$). Finally, if $\det(U) \neq 1$, then Equation (3) is not a consequence of the others. We conjecture that the condition $\det(U) \neq 1$ can be relaxed to $U \neq I$.

4.5 Universality

A PBS-diagram represents a superposition of linear maps together with a permutation of polarisations and positions. Indeed, Proposition 10 shows that for any diagram $D : n \to n$, $[\![D]\!] \in \mathcal{SLP}_n$, where $\mathcal{SLP}_n$ is the monoid of the linear maps $f\colon \mathcal{H}_n \to \mathcal{H}_n$ such that $f\,|c, p, x\rangle = |\tau(c, p)\rangle \otimes U_{c,p}\,|x\rangle$ for some permutation τ on $\{\to, \uparrow\} \times [n]$ and matrices $U_{c,p} \in \mathbb{C}^{q\times q}$. We show in the following that the PBS-calculus is universal, in the sense that any linear map in $\mathcal{SLP}_n$ can be represented by a PBS-diagram:

▶ **Theorem 25.** *The* PBS*-calculus is universal: for any $f \in \mathcal{SLP}_n$, $\exists D : n \to n$, $[\![D]\!] = f$.*

Proof. The proof relies on the normal forms: given a linear map $f \in \mathcal{SLP}_n$ one can inductively construct a diagram in normal form, by considering the image of f when the particle is located on the last position ($p = n - 1$). ◀

Notice that $\mathcal{SLP}_n$ is strictly included in the set of linear maps from $\mathcal{H}_n$ to $\mathcal{H}_n$. Thus while being universal for $\mathcal{SLP}_n$ the PBS-diagrams are not expressive enough to represent a (non-polarising) beam splitter for instance.

5 Removing the trace – Loop unrolling

We consider in this section an application of the PBS-calculus. The semantics of the language points out that each trace, or feedback loop, is used at most twice. As a consequence, a natural question is to decide whether all loops can be unrolled, in order to transform any PBS-diagram into a trace-free PBS-diagram. Such a transformation is possible when all matrices are invertible:

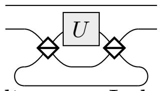
▶ **Proposition 26.** *Let $D : n \to n$ with $n \geq 2$ be a* PBS*-diagram such that all matrices appearing in some gate* $-\boxed{U}-$ *in D are invertible. Then there exists a trace-free* PBS*-diagram D' such that* PBS $\vdash D = D'$*.*

Notice that Proposition 26 is not true for PBS-diagrams with a single input/output. Indeed a trace-free diagram of type $1 \to 1$ is made of generators acting on 1 wire only, so in particular it has no polarising beam splitter and as a consequence cannot have a behaviour which depends on the polarisation. For instance, the diagram $E(U, V)$ used in the normal forms (see Remark 17) cannot be transformed into a trace-free diagram unless $U = V$.

On the other hand, PBS-diagrams involving at least one non-invertible matrix are not necessarily equivalent to a trace-free one. Indeed, we have the following property:

▶ **Lemma 27.** *For any trace-free PBS-diagram D, either all $[D]_{c,p}$ are invertible or at least two of them are not.*

This prevents the following diagram from being equivalent to a trace-free one:

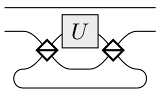
▶ **Example 28.** If U is not invertible, then the diagram $D_U : 2 \to 2 =$ 

is not equivalent, according to the rules of the PBS-calculus, to any trace-free diagram. Indeed, for any $(c, p) \neq (\to, 1)$ we have $[D_U]_{c,p} = I_q$, which is invertible, whereas $[D_U]_{\to,1} = U$.

Another interesting property is that loop unrolling, when it is possible, requires the use of matrices that were not present in the original diagram. This is a consequence of the following lemma:

▶ **Lemma 29.** *Given any diagram $D : n \to n$, let us define* $|D| := \prod_{c\in\{\to,\uparrow\},p\in[n]} \det\left([D]_{c,p}\right)$. *Then for any trace-free diagram D, we have* $|D| = \prod_{G \text{ gate in } D} \det\left(U(G)\right)^2$ *where $U(G)$ denotes the matrix with which G is labelled.*

Proof. Intuitively, due to the invertibility of the PBS-diagrams (Proposition 6), for each wire of a trace-free diagram D, there are exactly two initial configurations which are going through this particular wire. As a consequence each gate of D contributes twice to $|D|$ (see the full version of a paper [9] for a formal proof). ◀

▶ **Example 30.** Unless $\det(U)$ is a kth root of unity for some odd integer k, the following diagram D_U does not have the same semantics as any trace-free diagram in which all gates are labelled by U: 

. Indeed, we have $|D_U| = \det(U)$, and by Lemma 29, if D_U is equivalent through PBS to a trace-free diagram D'_U in which all gates are labelled by U, then we have $|D_U| = \det(U) = \det(U)^{2N}$, where N is the number of gates in D'_U. By Lemma 27, we have $\det(U) \neq 0$, so that $\det(U)^{2N-1} = 1$, that is, $\det(U)$ is a kth root of unity with $k = 2N - 1$ odd (if $N = 0$ then $\det(U) = 1$ so the result is still true).

6 Conclusion and Perspectives

In this paper, we have introduced a rigorous framework to reason on quantum computations involving coherent control, which are sometimes informally represented by schemes involving polarising beam splitters and black boxes. The main result is the introduction of an equational theory which makes the PBS-calculus sound and complete. We have also proved that the axiomatisation is minimal in the sense that each axiom is necessary for the completeness. Moreover, we have demonstrated for instance that the PBS-calculus can be used for loop unrolling.

So we have introduced the foundations of a formal framework, that we believe will be a useful tool to study the power and the limits of computations and protocols involving coherent control. We mention here three perspectives in the development of the PBS-calculus.

First, the expressivity of the language can be increased by adding, for instance, a (not polarising) beam splitter as a generator of the language, or by allowing more than one particle in the diagrams. Both are necessary for the representation of Boson sampling for instance.

Another perspective is to allow the gates to be arbitrary quantum channels. Indeed recent results [14, 1] point out interesting and unexpected behaviours of coherently controlled quantum channels. Our objective is to make the PBS-calculus a formal framework to explore and study such phenomena.

Finally, the calculus can be made more resource-sensitive, by allowing only the equations for which the number of occurrences of each gate (or black box) is preserved. For instance, we have seen examples in which loop unrolling requires to introduce new gates that were not present in the initial diagram. Transforming a diagram into its normal form is another example that does not, in general, preserve the number of occurrences of each gate.

References

1 Alastair A. Abbott, Julian Wechs, Dominic Horsman, Mehdi Mhalla, and Cyril Branciard. Communication through coherent control of quantum channels. *arXiv preprint*, October 2018. `arXiv:1810.09826`.

2 Thorsten Altenkirch and Jonathan Grattage. A functional quantum programming language. In *20th Annual IEEE Symposium on Logic in Computer Science (LICS'05)*, pages 249–258. IEEE, 2005. `doi:10.1109/LICS.2005.1`.

3 Mateus Araújo, Fabio Costa, and Časlav Brukner. Computational advantage from quantum-controlled ordering of gates. *Physical review letters*, 113(25):250402, 2014. `doi:10.1103/PhysRevLett.113.250402`.

4 Miriam Backens and Aleks Kissinger. ZH: A complete graphical calculus for quantum computations involving classical non-linearity. *Electronic Proceedings in Theoretical Computer Science*, 287:23–42, January 2019. `doi:10.4204/eptcs.287.2`.

5 Filippo Bonchi, Joshua Holland, Robin Piedeleu, Paweł Sobociński, and Fabio Zanasi. Diagrammatic algebra: from linear to concurrent systems. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–28, 2019. `doi:10.1145/3290338`.

6 Filippo Bonchi, Robin Piedeleu, Pawel Sobociński, and Fabio Zanasi. Graphical affine algebra. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE, 2019. `doi:10.1109/LICS.2019.8785877`.

7 Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. Interacting hopf algebras. *Journal of Pure and Applied Algebra*, 221(1):144–184, 2017. preprint available from `https://eprints.soton.ac.uk/406232`. `doi:10.1016/j.jpaa.2016.06.002`.

8 Giulio Chiribella. Perfect discrimination of no-signalling channels via quantum superposition of causal structures. *Physical Review A*, 86(4):040301, 2012. `doi:10.1103/PhysRevA.86.040301`.

9 Alexandre Clément and Simon Perdrix. PBS-calculus: A graphical language for coherent control of quantum computations. *arXiv preprint*, 2020. `arXiv:2002.09387`.

10 Bob Coecke and Ross Duncan. Interacting quantum observables: categorical algebra and diagrammatics. *New Journal of Physics*, 13(4):043016, 2011. `doi:10.1088/1367-2630/13/4/043016`.

11 Timoteo Colnaghi, Giacomo Mauro D'Ariano, Stefano Facchini, and Paolo Perinotti. Quantum computation with programmable connections between gates. *Physics Letters A*, 376(45):2940–2943, 2012. `doi:10.1016/j.physleta.2012.08.028`.

12 Giovanni de Felice, Amar Hadzihasanovic, and Kang Feng Ng. A diagrammatic calculus of fermionic quantum circuits. *Logical Methods in Computer Science*, 15(3), 2019. `doi:10.23638/LMCS-15(3:26)2019`.

13 Gilles Dowek and Pablo Arrighi. Lineal: A linear-algebraic lambda-calculus. *Logical Methods in Computer Science*, 13(1), 2017. `doi:10.23638/LMCS-13(1:8)2017`.

14 Daniel Ebler, Sina Salek, and Giulio Chiribella. Enhanced communication with the assistance of indefinite causal order. *Physical review letters*, 120(12):120502, March 2018. `arXiv:1711.10165`. `doi:10.1103/PhysRevLett.120.120502`.

15 Stefano Facchini and Simon Perdrix. Quantum circuits for the unitary permutation problem. In *International Conference on Theory and Applications of Models of Computation*, pages 324–331. Springer, 2015. `doi:10.1007/978-3-319-17142-5_28`.

16 Adrien Feix, Mateus Araújo, and Časlav Brukner. Quantum superposition of the order of parties as a communication resource. *Physical Review A*, 92(5):052326, 2015. `doi:10.1103/PhysRevA.92.052326`.

17 Philippe Allard Guérin, Adrien Feix, Mateus Araújo, and Časlav Brukner. Exponential communication complexity advantage from quantum superposition of the direction of communication. *Physical review letters*, 117(10):100502, 2016. doi:10.1103/PhysRevLett.117.100502.

18 Amar Hadzihasanovic. *The algebra of entanglement and the geometry of composition*. PhD thesis, University of Oxford, 2017. arXiv:1709.08086.

19 Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15), October 2009. doi:10.1103/physrevlett.103.150502.

20 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*, pages 559–568, 2018. doi:10.1145/3209108.3209131.

21 André Joyal, Ross Street, and Dominic Verity. Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society*, 119(3):447–468, 1996. doi:10.1017/S0305004100074338.

22 Gregory M Kelly and Miguel L Laplaza. Coherence for compact closed categories. *Journal of pure and applied algebra*, 19:193–213, 1980. doi:10.1016/0022-4049(80)90101-2.

23 Saunders Mac Lane. Categorical algebra. *Bull. Amer. Math. Soc.*, 71:40–106, 1965. doi:10.1090/S0002-9904-1965-11234-4.

24 Lorenzo M Procopio, Amir Moqanaki, Mateus Araújo, Fabio Costa, Irati Alonso Calafell, Emma G Dowd, Deny R Hamel, Lee A Rozema, Časlav Brukner, and Philip Walther. Experimental superposition of orders of quantum gates. *Nature communications*, 6:7913, 2015. doi:10.1038/ncomms8913.

25 Giulia Rubino, Lee A Rozema, Adrien Feix, Mateus Araújo, Jonas M Zeuner, Lorenzo M Procopio, Časlav Brukner, and Philip Walther. Experimental verification of an indefinite causal order. *Science advances*, 3(3):e1602589, 2017. doi:10.1126/sciadv.1602589.

26 Amr Sabry, Benoît Valiron, and Juliana Kaizer Vizzotto. From symmetric pattern-matching to quantum control. In *International Conference on Foundations of Software Science and Computation Structures*, pages 348–364. Springer, 2018. doi:10.1007/978-3-319-89366-2_19.

27 Peter Selinger. A survey of graphical languages for monoidal categories. In Bob Coecke, editor, *New Structures for Physics*, volume 813 of *Lecture Notes in Physics*, pages 289–355. Springer, 2011. Also available from arXiv:0908.3347. doi:10.1007/978-3-642-12821-9_4.

28 Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, October 1997. doi:10.1137/S0097539795293172.

29 Mingsheng Ying, Nengkun Yu, and Yuan Feng. Alternation in quantum programming: from superposition of data to superposition of programs. *arXiv preprint*, 2014. arXiv:1402.5172.

30 Fabio Zanasi. *Interacting Hopf Algebras- the Theory of Linear Systems*. PhD thesis, Ecole normale supérieure de lyon - ENS LYON, October 2015. URL: https://tel.archives-ouvertes.fr/tel-01218015.