



HAL
open science

The Many-Worlds Calculus: Representing Quantum Control

Kostia Chardonnet, Marc de Visme, Benoît Valiron, Renaud Vilmart

► **To cite this version:**

Kostia Chardonnet, Marc de Visme, Benoît Valiron, Renaud Vilmart. The Many-Worlds Calculus: Representing Quantum Control. 2022. hal-03654190v1

HAL Id: hal-03654190

<https://hal.science/hal-03654190v1>

Preprint submitted on 28 Apr 2022 (v1), last revised 3 Aug 2022 (v2)

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.

The Many-Worlds Calculus: Representing Quantum Control

Kostia Chardonnet  

Université Paris-Saclay, CNRS, ENS Paris-Saclay, LMF, 91190, Gif-sur-Yvette, France.

Université Paris Cité, CNRS, IRIF, F-75006, Paris, France

Marc de Visme  

Université Paris-Saclay, CNRS, ENS Paris-Saclay, Inria, LMF, 91190, Gif-sur-Yvette, France

Benoît Valiron   

Université Paris-Saclay, CNRS, CentraleSupélec, ENS Paris-Saclay, LMF, 91190, Gif-sur-Yvette, France

Renaud Vilmart   

Université Paris-Saclay, CNRS, ENS Paris-Saclay, Inria, LMF, 91190, Gif-sur-Yvette, France

Abstract

We propose a new typed graphical language for quantum computation, based on compact categories with biproducts. Our language generalizes existing approaches such as ZX-calculus and quantum circuits, while offering a natural framework to support quantum control: it natively supports “quantum tests”.

The language comes equipped with a denotational semantics based on linear applications, and an equational theory. Through the use of normal forms for the diagrams, we prove the language to be universal, and the equational theory to be complete with respect to the semantics.

2012 ACM Subject Classification Theory of computation → Quantum computation theory; Theory of computation → Linear logic; Theory of computation → Equational logic and rewriting

Keywords and phrases Quantum Computation, Linear Logic, Graphical Languages, Geometry of Interaction

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Introduction

Conventional wisdom has it that quantum computation is about *quantum data in superposition*. In the standard model, the memory holding quantum data is encapsulated inside a coprocessor accessed through a simple interface: The coprocessor holds individually addressable registers holding *quantum* bits, on which one can apply a fixed set of operations — *gates*— specified by the interface. If some of these gates can generate superposition of data, this is kept inside the coprocessor and opaque to the programmer. A typical interaction with the coprocessor is a purely classical sequence of elementary operations of the form “Apply gate X to register n ; apply gate Y to register m ; *etc*”. Such a sequence of instructions is usually represented as a *quantum circuit*. In this model, a quantum program is then a conventional program building a quantum circuit and sending it as a batch-job to the coprocessor.

From a semantical perspective, the *state* of a quantum memory consisting of n quantum bits is a vector in a 2^n -dimensional Hilbert space. A quantum circuit is a linear, sequential description of elementary operations describing a *linear, unitary map* on the state space. The quantum coprocessor should come with a *universal* set of such operations, that is, such that any unitary operation on the state space can be realized using the given elementary gates.

If unitary maps form the original representation for (pure) quantum operations, this extensional presentation makes a very rough semantics for the interaction with the quantum memory. It is akin to say that one could abstract a conventional program by the plot of



© Kostia Chardonnet and Marc de Visme and Benoît Valiron and Renaud Vilmart; licensed under Creative Commons License CC-BY 4.0

42nd Conference on Very Important Topics (CVIT 2016).

Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:31



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

45 the corresponding function: this hides useful information one might want to keep track of,
 46 such as topological constraints, composition of subroutines and execution flow, required
 47 resources or other costs, *etc* [33]. Most of the *structure* of the computation is lost in the
 48 matrix representation.

49 Coming all the way from Feynman's diagrams [22], graphical languages for representing
 50 quantum processes can be seen as an answer to the limitations of plain unitary matrices. In
 51 recent year, in the field of quantum computation has blossomed a wide variety of proposals
 52 to focus on specific aspects of the semantics of quantum programs [23, 14, 8].

53 Quantum circuits are an obvious candidate for a graphical language, and indeed, several
 54 lines of research took them as their main object of study [24, 16, 30, 7]. Quantum circuits in
 55 particular form a natural medium for describing the execution flow of a computation. Similar
 56 to the intuition one can build for classical boolean circuits, one can make an operational
 57 semantics for quantum circuits using a token-machine Geometry of Interaction. Each token
 58 in the circuit carries a quantum bit, and they flow in the circuit from inputs to outputs
 59 while getting modified as they pass through gates [16]. The main problem with the model of
 60 quantum circuits is the lack of a satisfactory equational presentation. If several attempts have
 61 been made for various subsets [12, 11, 26, 27], none of them provides a complete presentation.

62 A recent proposal answering the defects of quantum circuits as a (formal) language is the
 63 ZX-calculus [14]. Rooted in category theory [1], it comes with a small set of generators and
 64 a sound and complete equational theory [36]. Conservative extension of quantum circuits,
 65 yet versatile, the ZX-calculus has shown its use in a wide variety of applications, ranging
 66 from optimization [20, 3], verification [21, 19], and error-correction [17].

67 Despite its success, the ZX-calculus is nonetheless still tied to the quantum coprocessor
 68 model. Indeed, it fails to account for one peculiar feature of quantum computation: non-
 69 causal execution paths. Indeed, the Janus-faced quantum computational paradigm features
 70 two seemingly distinct notions of control structure. On the one hand, a quantum program
 71 follows *classical* control: it is hosted on the conventional computer governing the coprocessor,
 72 and can therefore only enjoy loops, tests and other regular causally ordered sequences of
 73 operations. On the other hand, the lab bench turns out to be more flexible than the rigid
 74 coprocessor model, permitting more elaborate *purely quantum* computational constructs than
 75 what quantum circuits or ZX-calculus allow.

76 The archetypal example of an quantum computational behavior hardly attainable within
 77 ZX-calculus (or quantum circuits) is the *Quantum Switch*. Consider two quantum bits x
 78 and y and two unitary operations U and V acting on y . The problem consists in generating
 79 the operation that performs UV on y if x is in state $|0\rangle$ and VU if it is in state $|1\rangle$. As
 80 x can be in superposition, in general the operation is then sending $(\alpha|0\rangle + \beta|1\rangle) \otimes |y\rangle$ to
 81 $\alpha|0\rangle \otimes (UV|y\rangle)_x + \beta|1\rangle \otimes (VU|y\rangle)$. It is a *purely quantum test*: not only can we have values
 82 in superpositions (here, x) but also *execution orders*. This is in sharp contrast with what
 83 happens within the standard quantum coprocessor model.

84 Computational models supporting superpositions of execution orders have been studied
 85 in the literature. One trend of research consists in proposing a suitable extension of
 86 quantum circuits [9, 31, 35, 37]. These approaches typically aim at discussing the notion
 87 of quantum channel from a quantum information theoretical standpoint. Another research
 88 line is concerned with exhibiting the execution flow internal to ZX-terms in a distributed
 89 manner [6]. Somewhat similarly to what has been proposed in [16], tokens are let to flow in
 90 a ZX-term, realizing the computation, the difference being that the tokens can themselves
 91 be in superposition. The limit is however that a ZX-term is inherently *causal*, therefore
 92 restricting the possibilities for superposition of orders.

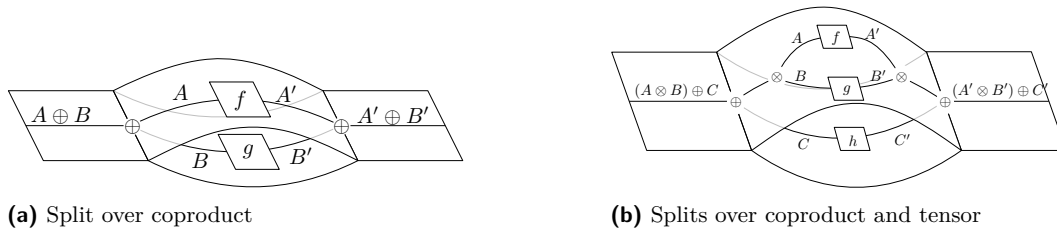


Figure 1 Examples of splits

93 One limitation of the existing extensions of quantum circuits or of graphical languages
 94 based on ZX-calculus is the limited support for *typed quantum data*. In the context of
 95 quantum circuits, the strategy have been to consider extended structures stemming from
 96 proofs-nets and resource-sensitive logics such as linear logic. A proof-net is a structured graph
 97 describing a proof of a linear logic formula. Through the Curry-Howard isomorphism, a proof
 98 can be regarded as a program and a formula as a type specifying how the program handles
 99 resources. This turns out to be particularly well-suited for quantum computation [18, 29, 16].

100 In the approaches merging quantum circuits with types based on linear logic, the *additive*
 101 *types* such as the sum-type are *purely classical*: $1 \oplus 1$ is always a (conventional) boolean. In
 102 quantum computation, the sum-type $1 \oplus 1$ can however be understood as a *sum of vector*
 103 *spaces*, giving an alternative interpretation to $1 \oplus 1$: it can be regarded as the type of a quantum
 104 bit, superposition of True and False. One should note that this appealing standpoint should
 105 be taken cautiously: (Pure) quantum information imposes strong constraints on the structure
 106 of the data in superposition: orthogonality and unit-norm have to be preserved [2, 32].

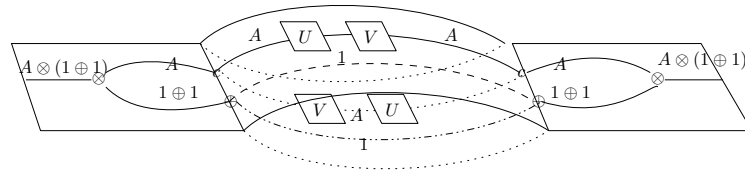
107 The standard categorical formalization underlying quantum computation, whether with
 108 the coprocessor or with the purely quantum viewpoint, is symmetric monoidal categories,
 109 and compact closed categories in particular. The monoidal structure stands for the tensor of
 110 quantum data, while coproducts are used to represent “tests”, whether classical [16, 18] or
 111 quantum [32].

112 Graphical languages for symmetric monoidal structure with coproducts usually rely on a
 113 notion of *sheet*, or *worlds*, to handle tests and coproducts in general [18, 28]. Figure 1a shows
 114 for instance how to represent the construction of the morphism $f \oplus g : A \oplus A' \rightarrow B \oplus B'$
 115 out of $f : A \rightarrow B$ and $g : A' \rightarrow B'$. The symbol “ \oplus ” stands for the “split” of worlds. Such a
 116 graphical language therefore comes with two distinct “splits”: one for the monoidal structure
 117 —leaving inside one specific world—, and one for the coproduct —splitting worlds—. They
 118 can be intertwined, as shown in Figure 1b. Another approach followed by [15] externalizes the
 119 two products (tensor product and coproduct) into the structure of the diagrams themselves,
 120 at the price of a less intuitive tensor product and a form of synchronization constraint.

121 However, in the state of the art this “splitting-world” understanding has only been carried
 122 for *classical* tests [16] and probabilistic branching stemming from measurements [18, 34],
 123 and *not* for quantum superposition. The Quantum Switch can for instance be naturally
 124 understood in this framework. Consider for instance Figure 2, read from left to right: as
 125 input, a pair of an element of type A and a quantum bits. Based on the value of the qubit
 126 (True or False), the wire A goes in the upper or the lower sheet, and is fed with U then V or
 127 V then U . Then everything is merged back together.

128 In this paper, we provide a rigorous interpretation for this type of (purely quantum)
 129 behavior.

130 **Contributions** In this paper, we introduce a new graphical language for quantum



■ Figure 2 Quantum Switch with Worlds

131 computation, based on compact category with biproduct [?]. This language allows us to
 132 express any quantum process, as we can encode the ZX-Calculus within it. We develop a
 133 denotational semantic and an equational theory, and prove the soundness and completeness
 134 of the semantics. As a case-study, we show how the Quantum Switch can naturally be
 135 encoded in the language.
 136 In the paper, the missing proofs can be found in Appendix.

2 The Many-Worlds Calculus

138 While the goal is to define a graphical language in which each wire can be enabled or disabled
 139 depending on the world in which the computation takes place, we first define the category
 140 \mathbf{C}_D of diagrams without any "worlds", and will then add the world annotations.

2.1 A First Graphical Language

142 We define our graphical language within the paradigm of colored PROP [4, 25], meaning
 143 that a diagram will be composed of nodes, or *generators*, linked to each others through *typed*
 144 *wires*, wired that are allowed to cross each others. Additionally, we assume that our colored
 145 PROP is compact closed and auto-dual, meaning that we allow to curve wires to obtain a
 146 Cup or a Cap.

147 The generators of our language are described in Figure 3 and are respectively the Identity,
 148 the Swap, the Cup, the Cap, the Plus, the Tensor, the Unit, the n -ary Contraction, and
 149 the Scalar indexed with s ranging over \mathbb{C} . Mirrored versions of those generators are defined
 150 as syntactic sugar through the compact closure, as shown for the mirrored Plus on the
 151 right-hand-side of Figure 3. Diagrams are read top-to-bottom: the top-most wires are the
 152 *input* wires and the bottom-most wires are the *output* wires.

153 The types of our wires are built from the syntax $A, B ::= 1 \mid A \oplus B \mid A \otimes B$. The objects
 154 of our category are the set of wires as generated by the grammar $\mathfrak{A}, \mathfrak{B} ::= \mathfrak{A} \square \mathfrak{B} \mid \bigvee \mid A$. The
 155 choice of the notation \square for wires in parallel is uncommon, we use it to put an emphasis on
 156 the fact that contrary to languages like the ZX-calculus, wires that are in parallel are not
 157 necessarily "in tensor with one another". In fact, $A \square B$ can be understood semantically as
 158 "either $A \otimes B$ or $A \oplus B$ ".

159 Diagrams are obtained from generators
 160 by composing them in parallel (written \square),
 161 or sequentially (written \circ) as follows. Se-
 162 quential composition requires the type (and
 163 number) of wires to match. We write \mathbf{C}_D for the category of diagrams we defined as such.

$$D_2 \circ D_1 := \begin{array}{c} \dots \\ \boxed{D_1} \\ \dots \\ \boxed{D_2} \\ \dots \end{array} \quad D_1 \square D_2 := \begin{array}{c} \dots \\ \boxed{D_1} \\ \dots \end{array} \square \begin{array}{c} \dots \\ \boxed{D_2} \\ \dots \end{array}$$

164 ► **Example 1.** While our language lacks the worlds labeling, we can already illustrate it by
 165 encoding some basic quantum primitive in it and show how they operate. In Figure 4 we
 166 show the encoding of a quantum bit $\alpha|0\rangle + \beta|1\rangle$ and the Hadamard unitary. In particular,

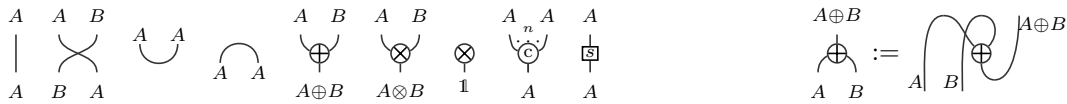


Figure 3 Generators of our First Graphical Language ($n \geq 0, s \in \mathbb{C}$)

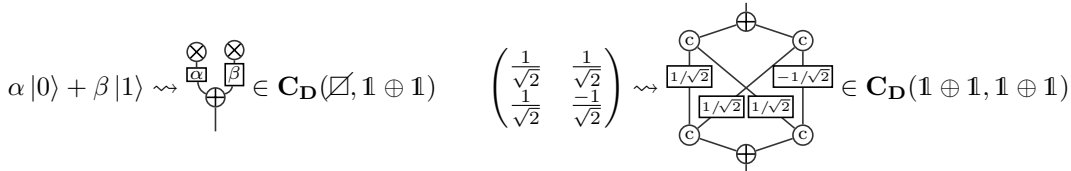
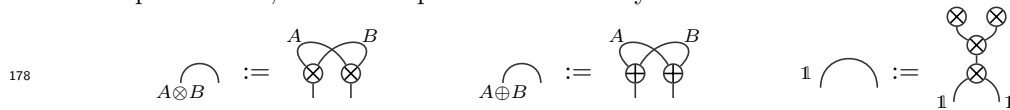


Figure 4 A Quantum Bit and the Hadamard Unitary

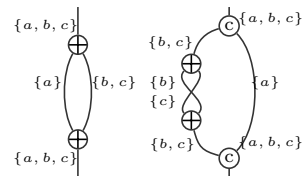
167 the Plus allows to "build" a new quantum bit from two scalars in parallel or to "open" a
 168 quantum bit to recover its corresponding scalars, the left branch corresponding to $|0\rangle$ and the
 169 right branch to $|1\rangle$. The meaning of the Contraction is better seen when applying Hadamard
 170 to a quantum bit as we show in Figure 5: it allows us to duplicate and sum scalars. The
 171 rewriting sequence of Figure 5 is made using the equational theory defined in Section 4,
 172 however to correctly define our equational theory, the worlds labeling are required. So while
 173 this specific worlds-free rewriting sequence is sound, many other similar worlds-free rewriting
 174 sequences are unsound.

175 ► Remark 2. Instead of having the Cup and the Cap as generators and defining the mirrored
 176 version of each generator through them, one could proceed the other way around by defining
 177 the Cap as follows, and the Cup in a mirrored way:



179 2.2 Adding Worlds Labeling

180 We now label wires of our diagram with worlds sets $w \subseteq W$
 181 for a given a set of worlds W . For each world $a \in W$,
 182 wires labeled by a set containing a are said to be "enabled
 183 in a ", and the others are said "disabled in a ". This allows
 184 us to correlate the enabling of wires. For example, the



185 "controlled not" can be represented by the diagram on the right, with worlds $\{a, b, c, \star\}$. The
 186 left-hand-side of the diagram forces the world a to correspond to the case where the control
 187 qubit is equal to $|0\rangle$, and the worlds b and c when it is equal to $|1\rangle$. The right-hand-side
 188 of the diagram applies the identity in the world a , and a negation in the worlds b and c .
 189 Lastly, the world \star appears nowhere in the labels, and corresponds to "we do not evaluate
 190 this circuit at all". While not strictly necessary, it is often practical to have a world absent
 191 from every wire.

192 ► Definition 3. Given a set of worlds W , we define the auto-dual compact closed colored
 193 PROP ($\mathbf{MW}_W, \square, \boxtimes$) of many-worlds calculus over W as follows:

194 Its colors are the pairs $(A : w)$ of colors A of \mathbf{C}_D and subsets $w \subseteq W$. We write $(\mathfrak{A}, \ell_{\mathfrak{A}})$ for
 195 the objects, where \mathfrak{A} is an object of \mathbf{C}_D and $\ell_{\mathfrak{A}}$ is a labeling function from the colors of \mathfrak{A} to
 196 the subsets of W .

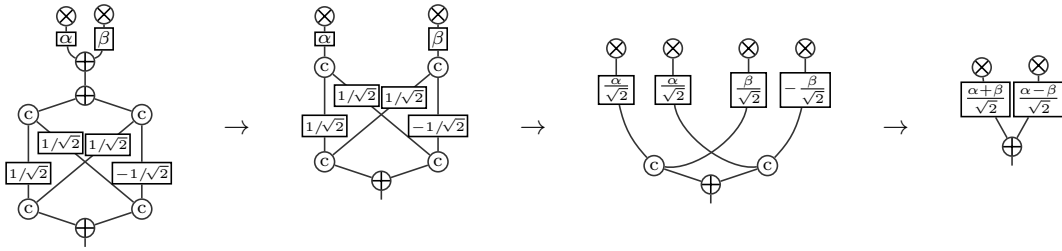


Figure 5 Applying the Hadamard unitary to a quantum bit

197 Its morphisms $f : (\mathfrak{A}, \ell_{\mathfrak{A}}) \rightarrow (\mathfrak{B}, \ell_{\mathfrak{B}})$ are pairs (\mathcal{D}_f, ℓ_f) of a morphism $\mathcal{D}_f \in \mathbf{C}_{\mathbf{D}}(\mathfrak{A}, \mathfrak{B})$ and
 198 a labeling function ℓ_f from the wires of \mathcal{D}_f to the subsets of W , satisfying the following
 199 constraints: The label on an input or output wire of color $(A : w)$ must be equal to w , and

200 $\begin{array}{c} |w \\ \cap \\ \cup \\ \oplus \\ \otimes \end{array} \begin{array}{c} w \\ \cup \\ \cap \\ \oplus \\ \otimes \end{array} \begin{array}{c} w \\ \cup \\ \cap \\ \oplus \\ \otimes \end{array} \begin{array}{c} w \\ \cup \\ \cap \\ \oplus \\ \otimes \end{array} \begin{array}{c} w \\ \cup \\ \cap \\ \oplus \\ \otimes \end{array} \forall n \geq 0, \begin{array}{c} w_1 \dots w_n \\ \oplus \\ w_1 \sqcup \dots \sqcup w_n \end{array}$

201 where \sqcup denotes a set union which is disjoint. The constraints for the mirrored versions are
 202 similar. The sequential composition \circ and the parallel composition \square preserve the labels.

203 Because of the first restriction on labels, given a morphism f , one can infer the labels on
 204 its objects from the label ℓ_f , hence we can write $f : \mathfrak{A} \rightarrow \mathfrak{B}$ instead of $f : (\mathfrak{A}, \ell_{\mathfrak{A}}) \rightarrow (\mathfrak{B}, \ell_{\mathfrak{B}})$
 205 unambiguously. It is discussed in Appendix B.1 how we can compose $f : \mathfrak{A} \rightarrow \mathfrak{B}$ and $g : \mathfrak{B} \rightarrow \mathfrak{C}$
 206 even when their induced labels do not match.

207 **Example 4 (The Quantum Switch).** Given two subdiagrams U and V , the Quantum
 208 Switch of $U; V$ and $V; U$ presented in Figure 2 can be encoded as in ???.The figure is split
 209 into two parts: the control part on the left-hand-side, and the computational part on the
 210 right-hand-side. The idea is that the control part, that uses \oplus , will behave as an *if-then-else*
 211 and will bind the world w to the case where the control quantum bit is $|0\rangle$, and the world
 212 v to the case where the control quantum bit is $|1\rangle$. On the computational part, $U \circ V$
 213 will be applied within the world v while $V \circ U$ will be applied within the world w . The “sheets”
 214 presented in Figure 2 are here modeled with world labels.

215 For the sake of simplicity, we represented the Quantum Switch here with two copies of U
 216 and V , one for each branch. It is actually possible within our language to share U and V
 217 between the two worlds in order to only have one copy of each (which is the whole purpose
 218 of the Quantum Switch). We describe in Example 8 this “true” Quantum Switch, and one
 219 can show using the equational theory given in Section 4 that the two are equivalent.

2.3 Comparison with Other Graphical Languages

221 We draw some comparison between our language and two other graphical languages: the
 222 ZX-Calculus [13], and Duncan’s Tensor-Sum Logic [18].

2.3.1 ZX-Calculus

224 The first difference is the restrictions of the ZX-calculus to computations between qubits,
 225 in other words linear map from $\mathbb{C}^{2^n} \mapsto \mathbb{C}^{2^m}$, while our language can encode any linear map
 226 from $\mathbb{C}^n \mapsto \mathbb{C}^m$. The Tensor generator allowing the decomposition of \mathbb{C}^{2^n} into instances
 227 of \mathbb{C}^2 was already present in the *scalable* extension of the ZX-calculus [5], but the main
 228 difference comes from the Plus (and the Contraction).

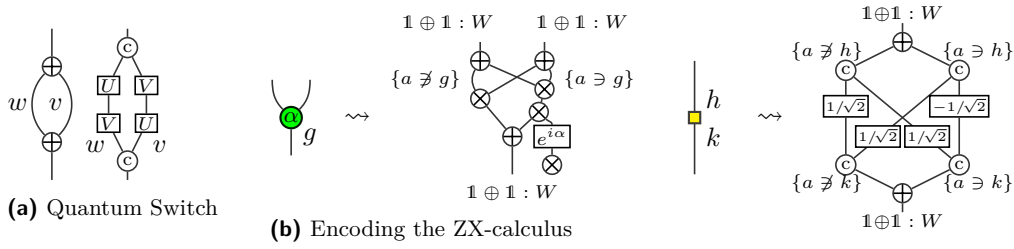
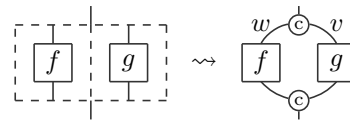


Figure 6

229 Additionally, every ZX-diagram can be encoded in our graphical language. The identity,
 230 swap, cup and cap of the ZX calculus are encoded by the similar generators over the type
 231 $\mathbb{1} \oplus \mathbb{1}$, the green spider and Hadamard gate are encoded as described in ??, and an encoding
 232 for the red spider can be deduced from those. The worlds set W is equal to $\mathcal{P}(\text{Names})$ where
 233 each green node of the ZX-diagram is given a name, and each input and each output of
 234 an Hadamard gate is given a name too. For a name g , we write $\{a \ni g\} \subseteq W$ for the set
 235 $\{a \subseteq \text{Names} \mid g \in a\}$ and $\{a \not\ni g\}$ for its complementary.

2.3.2 Tensor-Sum Logic

237 The core difference between their work and ours is the
 238 presence of the contraction in our graphical language. They
 239 instead rely on an enrichment of their category by a sum,
 240 which they represent graphically with boxes. We show on
 241 the right how the morphism $f + g$ would be encoded in both their and our language. More
 242 generally, their boxes correspond to uses of our contraction generator in a "well-bracketed"
 243 way. Another point of difference is their approach to quantum computation, as we do
 244 not assign the same semantics to those superpositions of morphisms. In their approach,
 245 the superposition is a classical construction and corresponds to the measurement and the
 246 classical control flow, while in our approach the superposition is a quantum construction and
 247 corresponds to the quantum control.



3 Semantics of the Many-Worlds Calculus

249 Our calculus aims at pure quantum computations. Following the ZX approach, we relax the
 250 condition on unitarity and define a semantics for our Many-World Calculus based on finite
 251 dimensional Hilbert spaces and (general) linear operators. More precisely, we will define two
 252 semantics, a world-dependent semantics $\llbracket - \rrbracket_a$ for every world a , which will be a monoidal
 253 functor from MW_W to FdHilb , and a worlds-agnostic semantics $\llbracket - \rrbracket$ which will not be
 254 functorial for the standard sequential composition and parallel composition, though we define
 255 in Appendix B.1 the worlds-agnostic sequential composition and parallel composition for
 256 which $\llbracket - \rrbracket$ will be functorial.

257 We start by defining those semantics on the objects. For every object \mathfrak{A} of C_D , we define
 258 its *enablings* \mathfrak{A}^\bullet as "replacing any number of wire type by \bullet ". For example $(A \square B)^\bullet =$
 259 $\{\bullet \square \bullet, \bullet \square B, A \square \bullet, A \square B\}$. To each enabling $\mathfrak{E} \in \mathfrak{A}^\bullet$ we associate a Hilbert space $\mathcal{H}_{\mathfrak{E}}$ as
 260 follows: $\mathcal{H}_{\mathfrak{E} \square \mathfrak{F}} := \mathcal{H}_{\mathfrak{E}} \otimes \mathcal{H}_{\mathfrak{F}}$, $\mathcal{H}_{\mathfrak{A}^\bullet} = \mathcal{H}_\bullet = \mathcal{H}_\mathbb{1} := \mathbb{C}$, $\mathcal{H}_{A \otimes B} := \mathcal{H}_A \otimes \mathcal{H}_B$, $\mathcal{H}_{A \oplus B} := \mathcal{H}_A \oplus \mathcal{H}_B$.
 261 Then, for any object $(\mathfrak{A}, \ell_{\mathfrak{A}})$ of MW_W and any world $a \in W$, we define $(\mathfrak{A}, \ell_{\mathfrak{A}}) \Downarrow a$ to be
 262 the enabling of \mathfrak{A} in which every $(A : w)$ with $a \in w$ is preserved and every $(A : w)$
 263 with $a \notin w$ is replaced by \bullet . For example $(A : \{a\} \square B : \{b\}) \Downarrow a = A \square \bullet$. We can then

$$\begin{aligned}
 \llbracket \text{Cup} \rrbracket_a &= \begin{cases} \text{Id} & \in \mathbf{FdHilb}(\mathcal{H}_A, \mathcal{H}_A) & \text{if } a \in w \setminus v \\ \text{Id} & \in \mathbf{FdHilb}(\mathcal{H}_B, \mathcal{H}_B) & \text{if } a \in v \setminus w \\ h \otimes h' \mapsto h' \otimes h & \in \mathbf{FdHilb}(\mathcal{H}_{A \otimes B}, \mathcal{H}_{B \otimes A}) & \text{if } a \in w \cap v \\ (1) & \in \mathbf{FdHilb}(\mathbb{C}, \mathbb{C}) & \text{otherwise} \end{cases} \\
 \llbracket \text{Cup}^\dagger \rrbracket_a &= \begin{cases} h \otimes h' \mapsto \langle h|h' \rangle & \in \mathbf{FdHilb}(\mathcal{H}_{A \otimes A}, \mathbb{C}) & \text{if } a \in w \\ (1) & \in \mathbf{FdHilb}(\mathbb{C}, \mathbb{C}) & \text{otherwise} \end{cases} \quad \begin{array}{l} \text{and the (conjugate)} \\ \text{transposed operator} \\ \text{for the Cup} \end{array} \\
 \llbracket \text{S} \rrbracket_a &= \begin{cases} s \cdot \text{Id} & \in \mathbf{FdHilb}(\mathcal{H}_A, \mathcal{H}_A) & \text{if } a \in w \\ (1) & \in \mathbf{FdHilb}(\mathbb{C}, \mathbb{C}) & \text{otherwise} \end{cases} \\
 \llbracket \text{Plus} \rrbracket_a &= \begin{cases} \begin{pmatrix} \text{Id} \\ 0 \end{pmatrix} & \in \mathbf{FdHilb}(\mathcal{H}_{A \oplus B}, \mathcal{H}_A) & \text{if } a \in w \\ \begin{pmatrix} 0 \\ \text{Id} \end{pmatrix} & \in \mathbf{FdHilb}(\mathcal{H}_{A \oplus B}, \mathcal{H}_B) & \text{if } a \in v \\ (1) & \in \mathbf{FdHilb}(\mathbb{C}, \mathbb{C}) & \text{otherwise} \end{cases} \quad \begin{array}{l} \text{and the (conjugate)} \\ \text{transposed operator} \\ \text{for the mirrored} \\ \text{Plus} \end{array}
 \end{aligned}$$

■ **Figure 7** Semantics of the Generators of \mathbf{MW}_W in a World $a \in W$.

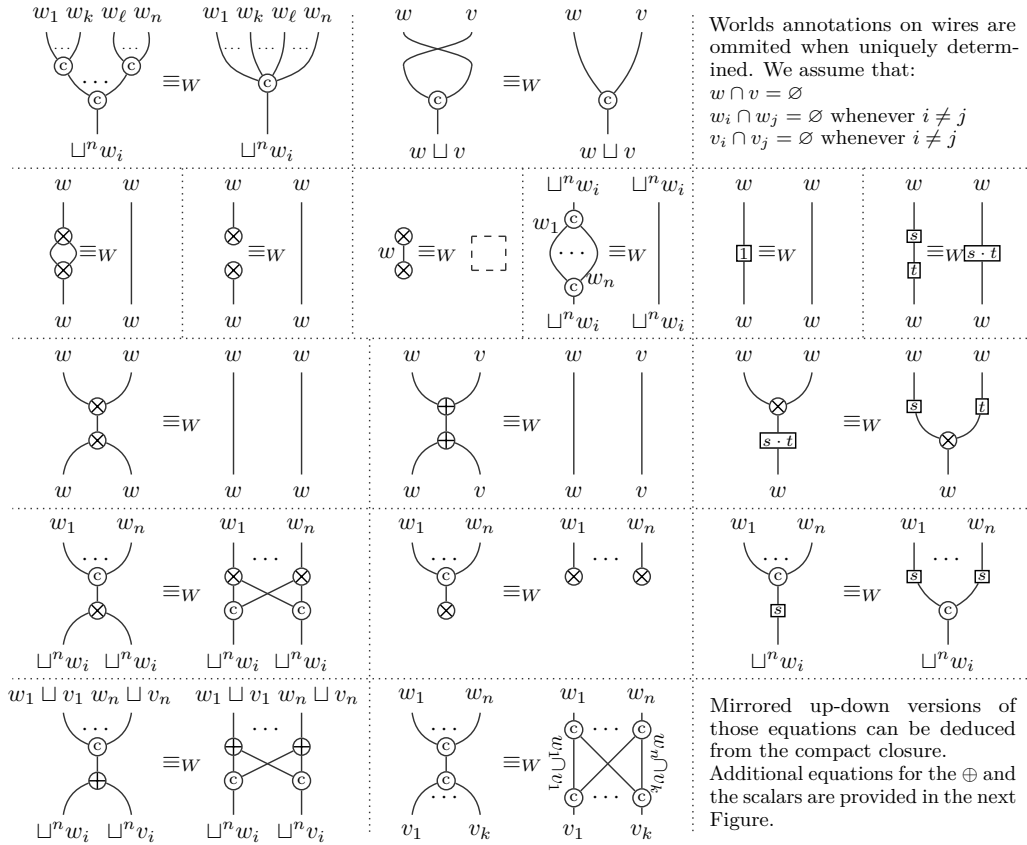
264 define the semantics $\llbracket - \rrbracket_a : \mathbf{MW}_W \rightarrow \mathbf{FdHilb}$ and $\llbracket - \rrbracket : \mathbf{MW}_W \rightarrow \mathbf{FdHilb}$ on objects
 265 as $\llbracket (\mathfrak{A}, \ell_{\mathfrak{A}}) \rrbracket_a := \mathcal{H}_{(\mathfrak{A}, \ell_{\mathfrak{A}}) \downarrow a}$ and $\llbracket (\mathfrak{A}, \ell_{\mathfrak{A}}) \rrbracket := \bigoplus_{\mathfrak{C} \in \mathfrak{A} \bullet} \mathcal{H}_{\mathfrak{C}}$. Then, for the morphisms, we
 266 proceed by compositionality for $\llbracket - \rrbracket_a$, meaning that we define $\llbracket - \rrbracket_a$ on every generator and
 267 compute the semantics of a diagram by decomposing it with $\llbracket g \circ f \rrbracket_a := \llbracket g \rrbracket_a \circ \llbracket f \rrbracket_a$ and
 268 $\llbracket f \square g \rrbracket_a := \llbracket f \rrbracket_a \otimes \llbracket g \rrbracket_a$. When looking at a generator $gen \in \mathbf{MW}_W((\mathfrak{A}, \ell_{\mathfrak{A}}), (\mathfrak{B}, \ell_{\mathfrak{B}}))$, in most
 269 cases $\llbracket (\mathfrak{A}, \ell_{\mathfrak{A}}) \rrbracket_a = \llbracket (\mathfrak{B}, \ell_{\mathfrak{B}}) \rrbracket_a = H$: we then define $\llbracket gen \rrbracket_a$ as the identity of $\mathbf{Hilb}(H, H)$.
 270 The semantics of the generators are given in Figure 6, for all the other generators their
 271 semantics is simply the identity.

272 The worlds-agnostic semantics is defined from the world-dependent semantics, as follows.
 273 Consider $f \in \mathbf{MW}_W((\mathfrak{A}, \ell_{\mathfrak{A}}), (\mathfrak{B}, \ell_{\mathfrak{B}}))$. Then $\llbracket f \rrbracket \in \mathbf{FdHilb}(\bigoplus_{\mathfrak{C} \in \mathfrak{A} \bullet} \mathcal{H}_{\mathfrak{C}}, \bigoplus_{\mathfrak{F} \in \mathfrak{B} \bullet} \mathcal{H}_{\mathfrak{F}})$ is
 274 defined as $\llbracket f \rrbracket := \left\{ \sum_{a \in W} \mathfrak{A} \mathfrak{B} \mathfrak{C} \mathfrak{F} \llbracket f \rrbracket_a \right\}_{\mathfrak{C} \in \mathfrak{A} \bullet, \mathfrak{F} \in \mathfrak{B} \bullet}$, where $W_{\mathfrak{A} \mathfrak{B} \mathfrak{C} \mathfrak{F}}$ is a shortcut notation
 275 for the set $\{a \in W \mid (\mathfrak{A}, \ell_{\mathfrak{A}}) \downarrow a = \mathfrak{C} \text{ and } (\mathfrak{B}, \ell_{\mathfrak{B}}) \downarrow a = \mathfrak{F}\}$.

276 For example, the worlds-agnostic semantics of the Tensor and the Plus are simply the
 277 collection of all their world-dependent semantics assembled into a single linear operator:

$$\begin{aligned}
 278 \quad \left[\begin{array}{c} \text{World set: } \{a, \star\} \\ A : \{a\} \quad B : \{a\} \\ \text{Cup} \\ A \otimes B : \{a\} \end{array} \right] &= A \otimes B \begin{pmatrix} A \square B & A \square \bullet & \bullet \square B & \bullet \square \bullet \\ \text{Id} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 279 \quad \left[\begin{array}{c} \text{World set: } \{a, b, \star\} \\ A : \{a\} \quad B : \{b\} \\ \text{Plus} \\ A \oplus B : \{a, b\} \end{array} \right] &= A \oplus B \begin{pmatrix} A \square B & A \square \bullet & \bullet \square B & \bullet \square \bullet \\ 0 & \text{Id} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
 \end{aligned}$$

281 The worlds-agnostic semantics is universal in the following sense:



■ **Figure 8** Equations with a Fixed Worlds Set W

282 ▶ **Theorem 5** (Universality). *For every $\mathfrak{A}, \mathfrak{B}$ objects of \mathbf{C}_D , and for every linear operator in*
 283 $U \in \mathbf{FdHilb} \left(\bigoplus_{\mathfrak{c} \in \mathfrak{A}} \mathcal{H}_{\mathfrak{c}}, \bigoplus_{\mathfrak{f} \in \mathfrak{B}} \mathcal{H}_{\mathfrak{f}} \right)$, *there exists a worlds set W , some labelings $\ell_{\mathfrak{A}}, \ell_{\mathfrak{B}}$,*
 284 *and a morphism $f \in \mathbf{MW}_W((\mathfrak{A}, \ell_{\mathfrak{A}}), (\mathfrak{B}, \ell_{\mathfrak{B}}))$ such that $\llbracket f \rrbracket = U$.* ◀

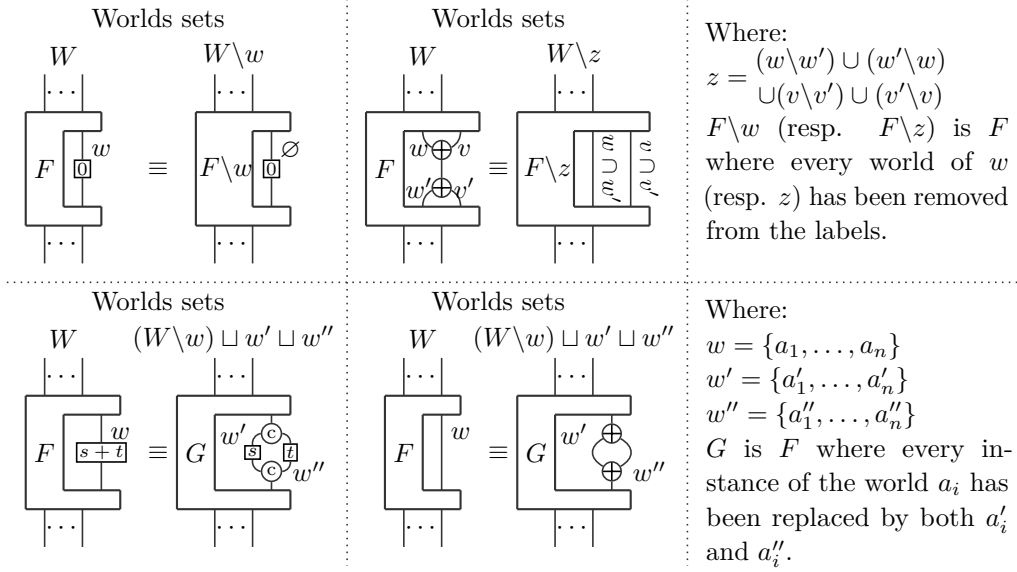
285 4 The Equational Theory

286 Similarly to how our semantics is defined in two steps, the equational theory is also defined
 287 in two steps:

288 (1) A set of equations within \mathbf{MW}_W for a fixed set of worlds W , which will not be complete,
 289 but will be sound for $\llbracket - \rrbracket_a$ for every $a \in W$, hence sound for $\llbracket - \rrbracket$ too. We write \equiv_W for the
 290 induced congruence¹ over \mathbf{MW}_W . We list those equations in Figure 7. Quite notably, the
 291 last two rows describe the fact that the contraction is a natural transformation.

292 (2) Five additional equations with side effects on the set of worlds, which will be sound and
 293 complete for $\llbracket - \rrbracket$, but not for the $\llbracket - \rrbracket_a$. We write \equiv for the induced equivalence relation,
 294 spanning over the morphisms of all the categories \mathbf{MW}_W for W any set of worlds. We have:
 295 One equation that allows us to rename the worlds: for every morphism (\mathcal{D}_f, ℓ_f) of \mathbf{MW}_W ,
 296 and for every bijection $i : W \rightarrow V$, we have $(\mathcal{D}_f, \ell_f) \equiv (\mathcal{D}_f, i \circ \ell_f) \in \mathbf{MW}_V$; Two equations

¹ In other words the smallest equivalence relation satisfying those equations and such that $f \equiv_W f' \implies \forall g, h, l, g \circ (f \square h) \circ k \equiv_W g \circ (f' \square h) \circ k$.



■ **Figure 9** Equations with Side-Effects on Worlds Sets

297 allowing the annihilation (or creation, when looking at them from right to left) of worlds
 298 due to coproducts or scalars (first row of Figure 8); Two equations allowing the splitting (or
 299 merging, when looking at them from right to left) of worlds due to coproducts or scalars
 300 (second row of Figure 8).

301 ► **Proposition 6** (Soundness). *For f a morphism of \mathbf{MW}_W and g of \mathbf{MW}_W , whenever $f \equiv g$
 302 we have $\llbracket f \rrbracket = \llbracket g \rrbracket$. Additionally if $W = V$, whenever $f \equiv_W g$ we have $\forall a \in W, \llbracket f \rrbracket_a = \llbracket g \rrbracket_a$.*

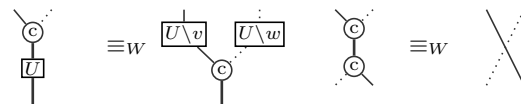
303 ► **Theorem 7** (Completeness). *For every $f : \mathfrak{A} \rightarrow \mathfrak{B}$ morphism of \mathbf{MW}_W and $g : \mathfrak{A} \rightarrow \mathfrak{B}$ one
 304 of \mathbf{MW}_V , $\llbracket f \rrbracket = \llbracket g \rrbracket \iff f \equiv g$ ◀*

305 The soundness can be proved by a case-by-case analysis on every equation. The complete-
 306 ness theorem follows from the existence of a normal form for \equiv , as described in Section 5.1.

307 ► **Example 8** (The Quantum Switch). As claimed in Example 4, it is possible to represent
 308 the Quantum Switch with only one copy of U and V , and one can rewrite it to the version
 309 with two copies of each using the equational theory as shown in Figure 9.

310 In those diagrams, the worlds set is $W = w \sqcup v$ and we rely on thick, thin and dotted wires
 311 to indicate respectively worlds labels $w \sqcup v$, w and v . Each figure has a control side which
 312 operates on a quantum bit (type $\mathbb{1} \oplus \mathbb{1}$) and binds the world w to $|0\rangle$ and the world v to
 313 $|1\rangle$; and a computational side which operates on some data of an arbitrary type A , on which
 314 could be applied U and/or V which stand for two morphisms of $\mathbf{MW}_W (A : W, A : W)$.

315 The first rewriting step relies on the two lemmas on the right, both of which being deducible
 from the equational theory (see Appendix A). The second rewriting step is simply using the
 properties of a compact closed category.



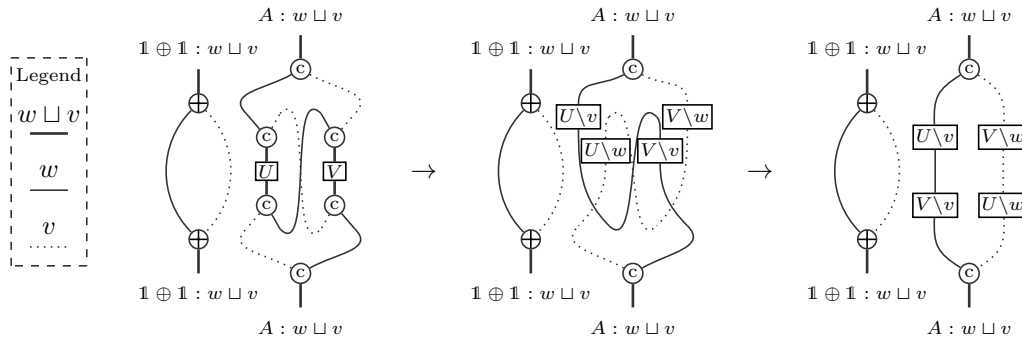


Figure 10 Rewriting the Quantum Switch

5 Normal Form and Completeness

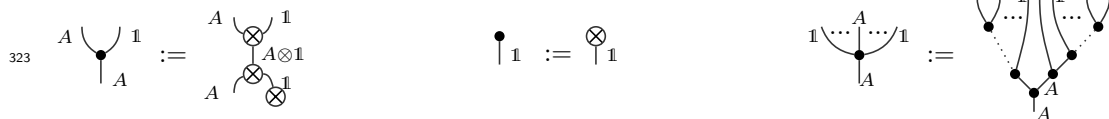
316

317 We can prove that the previous equational theory is complete, by defining a normal form on
 318 the morphisms, and by showing that all diagrams can be put in this normal form, which is
 319 unique.

5.1 Normal Form

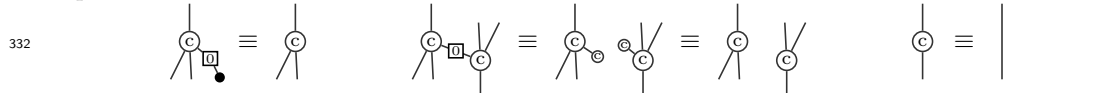
320

321 It will be practical for our normal forms to define the following syntactic sugar, which we
 322 call the *unitor*, its *unit* and its generalized form:



323

324 We define the short-hand on the right with the
 325 assumption that the worlds set is in bijection with the
 326 set of scalars, in other words the scalars λ_{ij} , λ'_i , λ''_j and
 327 λ_0 have for worlds label $\{a_{ij}\}$, $\{a'_i\}$, $\{a''_j\}$ and $\{a_0\}$
 328 respectively. In particular, all the input (resp. output)
 329 wires live in mutually exclusive worlds. An important observation is that any permutation of
 330 wires (all mutually exclusive, and of type $\mathbb{1}$) can easily be put in this form using the following
 331 equations:



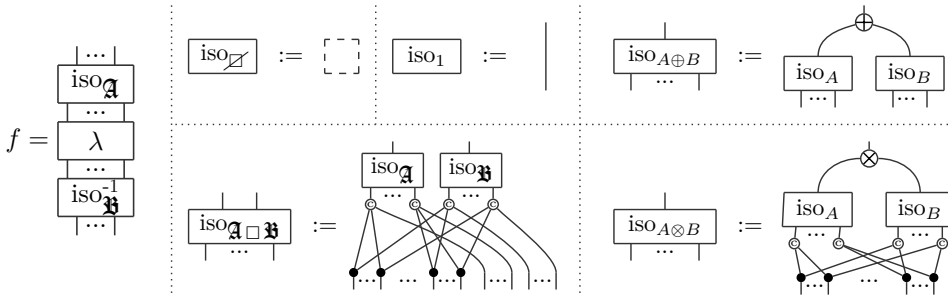
332

333 The normal form of a morphism $f : \mathfrak{A} \rightarrow \mathfrak{B}$ is defined as the form of the diagram on the left
 334 of Figure 10, where the morphism $\text{iso}_{\mathfrak{A}}$ is defined inductively right.

335 The output wires of $\text{iso}_{\mathfrak{A}}$ for any \mathfrak{A} live in mutually exclusive worlds, but once again, we
 336 don't overload the diagrams with unitors or world names encoding this information, although
 337 it will be used in the following.

338 Notice that graphically, there is no difference between $\mathfrak{A} \square (\mathfrak{B} \square \mathfrak{C})$ and $(\mathfrak{A} \square \mathfrak{B}) \square \mathfrak{C}$, in
 339 other words \square is strictly associative. However $\text{iso}_{\mathfrak{A} \square (\mathfrak{B} \square \mathfrak{C})}$ and $\text{iso}_{(\mathfrak{A} \square \mathfrak{B}) \square \mathfrak{C}}$ are different,
 340 but they are equivalent up to a rearranging of the output wires:

341 ► **Lemma 9.** *There exists a wire permutation σ such that $\text{iso}_{\mathfrak{A} \square (\mathfrak{B} \square \mathfrak{C})} = \sigma \circ \text{iso}_{(\mathfrak{A} \square \mathfrak{B}) \square \mathfrak{C}}$.* ◀



■ **Figure 11** The Normal Form (left) and the Inductive Definition of $\text{iso}_{\mathfrak{a}}$ (right)

342 We hence have a choice to make here for canonicity, and choose $\text{iso}_{A_0 \square A_1 \square A_2 \square \dots} :=$
 343 $\text{iso}(\dots((A_0 \square A_1) \square A_2) \square \dots)$.

344 We define $\text{iso}_{\mathfrak{a}}^{-1}$ inductively in the same way, but upside-down. We note that $\text{iso}_{\mathfrak{a}}^{-1} \circ \text{iso}_{\mathfrak{a}}$
 345 is the normal form of $\text{id}_{\mathfrak{a}}$.

346 ► **Proposition 10.** *The normal form is unique.*

347 5.2 Completeness

348 We can now use this normal form to show that our equational theory is complete for arbitrary
 349 morphisms. To do so, we need to show that all the generators can be put in normal form,
 350 and then that any composition of morphisms in normal form can be put in normal form. We
 351 do exactly so, and leave the details to the appendix.

352 ► **Proposition 11.** *The generators can be put in normal form.* ◀

353 ► **Proposition 12.** *Compositions of diagrams in normal form can be put in normal form.*

354 This allows us to prove the completeness theorem claimed above:

355 **Proof of Theorem 7.** The right-to-left direction of the equivalence can be directly checked
 356 by verifying that all the axioms preserve the semantics.

357 Let f_1 and f_2 be two morphisms such that $\llbracket f_1 \rrbracket = \llbracket f_2 \rrbracket$. Both morphisms can be put in
 358 normal form, resp. f_1^{NF} and f_2^{NF} , with $f_i \equiv f_i^{NF}$ and thus $\llbracket f_i^{NF} \rrbracket = \llbracket f_i \rrbracket$. By uniqueness of
 359 the normal form, and since $\llbracket f_1^{NF} \rrbracket = \llbracket f_2^{NF} \rrbracket$, we get $f_1^{NF} \equiv f_2^{NF}$, which ends the proof that
 360 $f_1 \equiv f_2$. ◀

361 6 Conclusion

362 We introduced a new sound and complete graphical language based on compact categories
 363 with biproducts, along with a equational theory and a worlds system, helping us build a
 364 denotational semantics of our language.

365 This language allows us to generalize already existing quantum graphical languages such
 366 as the ZX-calculus with the additions of richer types than just the usual qubits and tensors
 367 of qubits. In particular the biproduct allows us to reason about superposition of executions,
 368 as shown by the encoding of the Quantum Switch in Example 8.

369 While our language allows for quantum control, it does not capture another language
 370 that aims at formalizing quantum control, namely the PBS-Calculus [10]. How and in which
 371 context could we capture the PBS-Calculus is left for future work.

372 ——— **References** ———

- 373 1 Samson Abramsky and Bob Coecke. A categorical semantics of quantum protocols. In
374 *Proceedings of the 19th Symposium on Logic in Computer Science, LICS'04*, pages 415–425.
375 IEEE, IEEE Computer Society Press, July 2004. doi:10.1109/LICS.2004.1319636.
- 376 2 Thorsten Altenkirch and Jonathan Grattage. A functional quantum programming language.
377 In Prakash Panangaden, editor, *Proceedings of the 20th Symposium on Logic in Computer
378 Science, LICS'05*, pages 249–258. IEEE, IEEE Computer Society Press, 2005. doi:10.1109/
379 LICS.2005.1.
- 380 3 Miriam Backens, Hector Miller-Bakewell, Giovanni de Felice, Leo Lobski, and John van de
381 Wetering. There and back again: A circuit extraction tale, 2020. arXiv:2003.01664.
- 382 4 Titouan Carette. *Wielding the ZX-calculus, Flexsymmetry, Mixed States, and Scalable Notations.*
383 *(Manier le ZX-calcul, flexsymétrie, systèmes ouverts et limandes)*. PhD thesis, University
384 of Lorraine, Nancy, France, 2021. URL: <https://tel.archives-ouvertes.fr/tel-03468027>.
- 385 5 Titouan Carette, Dominic Horsman, and Simon Perdrix. SZX-Calculus: Scalable Graphical
386 Quantum Reasoning. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen,
387 editors, *44th International Symposium on Mathematical Foundations of Computer Science
388 (MFCS 2019)*, volume 138 of *Leibniz International Proceedings in Informatics (LIPIcs)*,
389 pages 55:1–55:15, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
390 URL: <http://drops.dagstuhl.de/opus/volltexte/2019/10999>, doi:10.4230/LIPIcs.
391 MFCS.2019.55.
- 392 6 Kostia Chardonnet, Benoît Valiron, and Renaud Vilmart. Geometry of interaction for ZX-
393 diagrams. In Filippo Bonchi and Simon J. Puglisi, editors, *Proceedings of the 46th International
394 Symposium on Mathematical Foundations of Computer Science, MFCS 2021*, volume 202 of
395 *LIPIcs*, pages 30:1–30:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2021. doi:
396 10.4230/LIPIcs.MFCS.2021.30.
- 397 7 Christophe Chareton, Sébastien Bardin, François Bobot, Valentin Perrelle, and Benoît
398 Valiron. A Deductive Verification Framework for Circuit-building Quantum Programs.
399 arXiv:2003.05841. To appear in *Proceedings of ESOP'21*.
- 400 8 Christophe Chareton, Sébastien Bardin, Dongho Lee, Benoît Valiron, Renaud Vilmart, and
401 Zhaowei Xu. Formal methods for quantum programs: A survey. arXiv:2109.06493v1, 2021.
- 402 9 G. Chiribella, G. M. D'Ariano, and P. Perinotti. Transforming quantum operations: Quantum
403 supermaps. *EPL (Europhysics Letters)*, 83(3):30004, 2008. doi:10.1209/0295-5075/83/
404 30004.
- 405 10 Alexandre Clément and Simon Perdrix. Pbs-calculus: A graphical language for coherent
406 control of quantum computations. arXiv preprint arXiv:2002.09387, 2020.
- 407 11 Robin Cockett and Cole Comfort. The category TOF. In Peter Selinger and Giulio Chiribella,
408 editors, *Proceedings 15th International Conference on Quantum Physics and Logic, QPL 2018*,
409 volume 287 of *EPTCS*, pages 67–84, 2019.
- 410 12 Robin Cockett, Cole Comfort, and Priyaa Srinivasan. The category CNOT. In Peter Selinger
411 and Giulio Chiribella, editors, *Proceedings 15th International Conference on Quantum Physics
412 and Logic, QPL 2018*, volume 287 of *EPTCS*, pages 258–293, 2019. doi:10.4204/EPTCS.266.
413 18.
- 414 13 Bob Coecke and Ross Duncan. Interacting quantum observables: categorical algebra and
415 diagrammatics. *New Journal of Physics*, 13(4):043016, 2011.
- 416 14 Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes: A First Course in Quantum
417 Theory and Diagrammatic Reasoning*. Cambridge University Press, 2017. doi:10.1017/
418 9781316219317.
- 419 15 Cole Comfort, Antonin Delpuech, and Jules Hedges. Sheet diagrams for bimonoidal categories,
420 2020. URL: <https://arxiv.org/abs/2010.13361>, doi:10.48550/ARXIV.2010.13361.
- 421 16 Dal Lago, Ugo and Faggian, Claudia and Valiron, Benoît and Yoshimizu, Akira. The geometry
422 of parallelism: Classical, probabilistic, and quantum effects. In *Proceedings of the 44th ACM
423 SIGPLAN Symposium on Principles of Programming Languages, POPL 2017*, page 833–845,

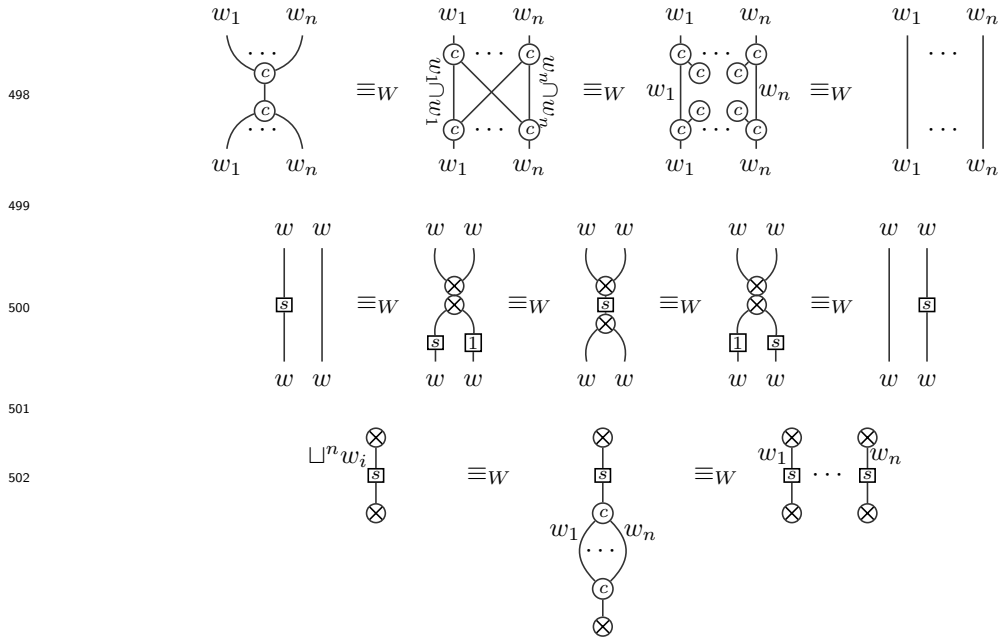
- 424 New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3009837.
425 3009859.
- 426 **17** Niel de Beaudrap and Dominic Horsman. The ZX calculus is a language for surface code
427 lattice surgery. *Quantum*, 4:218, January 2020. doi:10.22331/q-2020-01-09-218.
- 428 **18** Ross Duncan. Generalized proof-nets for compact categories with biproducts. In Gay and
429 Mackie [23], chapter 3, pages 70–134.
- 430 **19** Ross Duncan and Liam Garvie. Verifying the smallest interesting colour code with quantomatic.
431 In Bob Coecke and Aleks Kissinger, editors, *Proceedings 14th International Conference on*
432 *Quantum Physics and Logic, Nijmegen, The Netherlands, 3-7 July 2017*, volume 266 of
433 *Electronic Proceedings in Theoretical Computer Science*, pages 147–163, 2018. doi:10.4204/
434 EPTCS.266.10.
- 435 **20** Ross Duncan, Aleks Kissinger, Simon Perdrix, and John Van De Wetering. Graph-theoretic
436 Simplification of Quantum Circuits with the ZX-calculus. *Quantum*, 4:279, 2020.
- 437 **21** Ross Duncan and Maxime Lucas. Verifying the Steane code with Quantomatic. In Bob Coecke
438 and Matty Hoban, editors, *Proceedings of the 10th International Workshop on Quantum*
439 *Physics and Logic, Castelldefels (Barcelona), Spain, 17th to 19th July 2013*, volume 171
440 of *Electronic Proceedings in Theoretical Computer Science*, pages 33–49. Open Publishing
441 Association, 2014. doi:10.4204/EPTCS.171.4.
- 442 **22** Richard P. Feynman and A. R. Hibbs. *Quantum Mechanics and Path Integrals*. McGraw-Hill
443 Publishing Company, 1965.
- 444 **23** Simon Gay and Ian Mackie, editors. *Semantic Techniques in Quantum Computation*. Cambridge
445 University Press, 2009.
- 446 **24** Alexander S. Green, Peter LeFanu Lumsdaine, Neil J. Ross, Peter Selinger, and Benoît Valiron.
447 Quipper: A scalable quantum programming language. In Hans-Juergen Boehm and Cormac
448 Flanagan, editors, *Proceedings of the ACM SIGPLAN Conference on Programming Language*
449 *Design and Implementation, PLDI'13*, pages 333–342. ACM, 2013. doi:10.1145/2491956.
450 2462177.
- 451 **25** Philip Hackney and Marcy Robertson. On the category of props. *Appl. Categorical Struct.*,
452 23(4):543–573, 2015. doi:10.1007/s10485-014-9369-4.
- 453 **26** Christian Hutzlar, Jacques Carette, and Amr Sabry. A library of reversible circuit transforma-
454 tions (work in progress). In Jarkko Kari and Irek Ulidowski, editors, *Proceedings of the 10th*
455 *International Conference on Reversible Computation, RC 2018*, volume 11106 of *Lecture Notes*
456 *in Computer Science*, pages 339–345. Springer, 2018. doi:10.1007/978-3-319-99498-7_24.
- 457 **27** Justin Makary, Neil J. Ross, and Peter Selinger. Generators and relations for real stabilizer
458 operators. In Chris Heunen and Miriam Backens, editors, *Proceedings of the 18th International*
459 *Conference on Quantum Physics and Logic, QPL 2021*, volume 343 of *EPTCS*, pages 14–36,
460 2021. doi:10.4204/EPTCS.343.2.
- 461 **28** Paul-André Mellies. Local states in string diagrams. In *Rewriting and Typed Lambda Calculi*,
462 pages 334–348. Springer, 2014.
- 463 **29** Michele Pagani, Peter Selinger, and Benoît Valiron. Applying quantitative semantics to higher-
464 order quantum computing. In Suresh Jagannathan and Peter Sewell, editors, *Proceedings*
465 *of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages,*
466 *POPL'14*, pages 647–658. ACM, 2014. doi:10.1145/2535838.2535879.
- 467 **30** Jennifer Paykin, Robert Rand, and Steve Zdancewic. QWIRE: a core language for quantum
468 circuits. In Giuseppe Castagna and Andrew D. Gordon, editors, *Proceedings of the 44th ACM*
469 *SIGPLAN Symposium on Principles of Programming Languages, POPL'17*, pages 846–858.
470 ACM, 2017. doi:10.1145/3009837.3009894.
- 471 **31** Christopher Portmann, Christian Matt, Ueli Maurer, Renato Renner, and Björn Tackmann.
472 Causal boxes: Quantum information-processing systems closed under composition. *IEEE*
473 *Transactions on Information Theory*, 63(5):3277–3305, 2017. doi:10.1109/TIT.2017.2676805.
- 474 **32** Amr Sabry, Benoît Valiron, and Juliana Kaizer Vizzotto. From symmetric pattern-matching
475 to quantum control. In Christel Baier and Ugo Dal Lago, editors, *Proceedings of the 21st*

- 476 *International Conference on Foundations of Software Science and Computation Structures,*
477 *FoSSaCS 2018*, volume 10803 of *Lecture Notes in Computer Science*, pages 348–364. Springer,
478 2018. doi:10.1007/978-3-319-89366-2_19.
- 479 33 Seyon Sivarajah, Silas Dilkes, Alexander Cowtan, Will Simmons, Alec Edgington, and Ross
480 Duncan. t|ket): a retargetable compiler for NISQ devices. *Quantum Science and Technology*,
481 6(1):014003, 2020. doi:10.1088/2058-9565/ab8e92.
- 482 34 Sam Staton. Algebraic effects, linearity, and quantum programming languages. In Sriram K.
483 Rajamani and David Walker, editors, *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT*
484 *Symposium on Principles of Programming Languages, POPL'15*, pages 395–406. ACM, 2015.
485 doi:10.1145/2676726.2676999.
- 486 35 Augustin Vanrietvelde, Hlér Kristjánsson, and Jonathan Barrett. Routed quantum circuits.
487 *Quantum*, 5:503, 2021. doi:10.22331/q-2021-07-13-503.
- 488 36 Renaud Vilmart. *ZX-Calculi for Quantum Computing and their Completeness*. Theses,
489 Université de Lorraine, September 2019. URL: [https://hal.archives-ouvertes.fr/
490 tel-02395443](https://hal.archives-ouvertes.fr/tel-02395443).
- 491 37 Julian Wechs, Hippolyte Dourdent, Alastair A. Abbott, and Cyril Branciard. Quantum
492 circuits with classical versus quantum control of causal order. *PRX Quantum*, 2:030335, 2021.
493 doi:10.1103/PRXQuantum.2.030335.

494 **A** Induced Equations

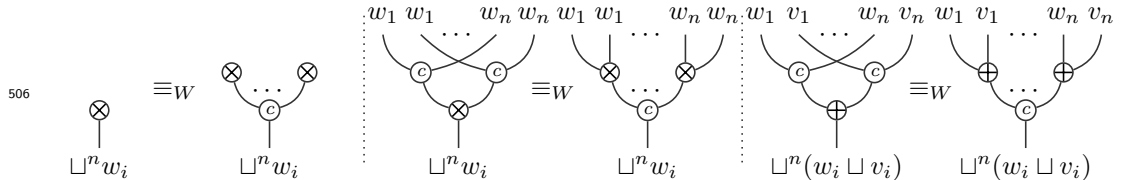
495 We include in this section a few practical lemmas that are provable from the equational
 496 theory \equiv_W .

497 **► Lemma 13.** *Whenever w_i are disjoint sets of worlds, we have the following:*

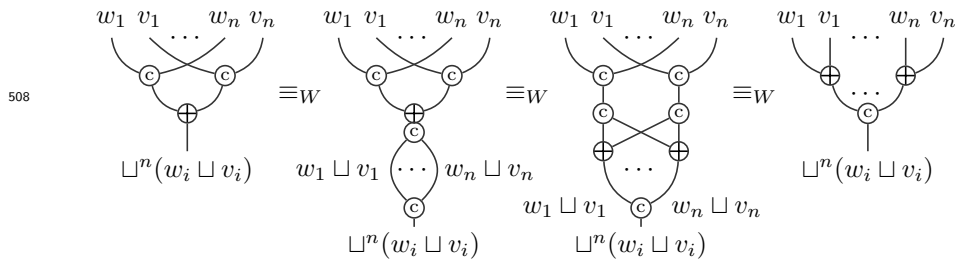


503 ◀

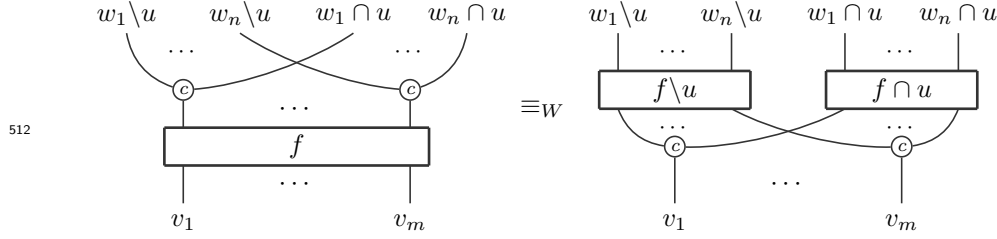
504 **► Lemma 14.** *Whenever w_i are disjoint sets of worlds, and that the v_i are disjoint set of
 505 worlds too, we have the following:*



507 **Proof.** We provide a proof for the third equation, the first two are proven similarly.



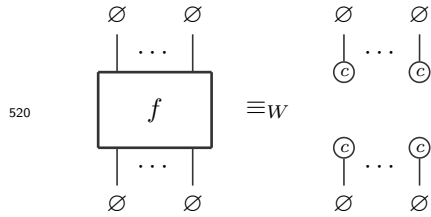
510 ► **Corollary 15.** For every $f : \square^n(A_i : w_i) \rightarrow \square^m(B_j : v_j)$ with worlds set W and every
 511 $u \subseteq W$, we have



513 where $f \setminus u : \square^n(A_i : w_i \setminus u) \rightarrow \square^m(B_j : v_j \setminus u)$ is equal to f where every worlds label w has
 514 been replaced by $w \setminus u$, and similarly for $f \cap u$.

515 This is simply proven by induction over f . All the generator cases (including Cup and
 516 Cap) follow directly from the equations given in Figure 7 and Lemma 14 together with the
 517 properties of a compact close category.

518 ► **Lemma 16.** For every $f : \square^n(A_i : \emptyset) \rightarrow \square^m(B_j : \emptyset)$ with worlds set W but such that
 519 every worlds label of f is \emptyset , we have



521 This is simply proven by replacing every wire by two contractions of arity zero (sixth
 522 axiom of Figure 7 with $n = 0$), and then using the naturality of the contraction of arity zero
 523 (last two lines of Figure 7 with $n = 0$) to consume every generator.

524 **B Dealing with Worlds Labeling**

525 The worlds labeling can appear to be a very strict structure making it unpractical to
 526 manipulate diagrams. In this appendix, we show a number of tools that allow to manipulate
 527 worlds labeling in a more flexible way.

528 **B.1 Worlds-Agnostic Category**

529 In this section, we define the category \mathbf{MW}_V which has the same objects as \mathbf{C}_D , so no worlds
 530 labeling on the objects, and has for morphisms all the morphisms of the categories \mathbf{MW}_W
 531 for every worlds set W . In other words, we explain how to make sequential composition and
 532 parallel composition of morphisms that have different worlds set.

533 We write $[f]_W : \mathfrak{A} \rightarrow \mathfrak{B}$ for a morphism $(\mathcal{D}_f, \ell_f) \in \mathbf{MW}_W((\mathfrak{A}, \ell_f^{\mathfrak{A}}), (\mathfrak{B}, \ell_f^{\mathfrak{B}}))$, where $\ell_f^{\mathfrak{A}}$ and
 534 $\ell_f^{\mathfrak{B}}$ are the labeling induced on \mathfrak{A} and \mathfrak{B} by ℓ_f .

535 The goal of this section is to allow to compose $[f]_W : \mathfrak{A} \rightarrow \mathfrak{B}$ and $[g]_V : \mathfrak{B} \rightarrow \mathfrak{C}$ even when
 536 the inferred labels on \mathfrak{B} do not match and when W and V are different.

$$\begin{array}{c}
 W = \{a, \star\} \quad V = \{b, \star\} \quad W \times V = \begin{array}{l} \{(a,b), (a,\star), \\ (\star,b), (\star,\star)\} \end{array} \\
 \left. \{a\} \right| \quad \square \quad \left. \{b\} \right| \quad = \quad \left. \begin{array}{l} \{(a,b), \\ (a,\star)\} \end{array} \right| \quad \left. \begin{array}{l} \{(a,b), \\ (\star,b)\} \end{array} \right|
 \end{array}$$

■ **Figure 12** Example of Worlds-Agnostic Parallel Composition

537 B.1.1 Extending the Worlds Set

538 Before tackling the sequential composition, we start by defining the parallel composition \square
 539 between morphisms with different world sets. For W, V two sets of worlds, and for $\ell : S \rightarrow W$,
 540 we define $\ell^{-\times V} : S \rightarrow W \times V$ as $\ell^{-\times V}(s) = \{(a, b) \mid a \in \ell(s), b \in V\}$. Similarly, for
 541 $[f]_W : \mathfrak{A} \rightarrow \mathfrak{B}$, we define $[f^{-\times V}]_{W \times V}$ as the morphism $(\mathcal{D}_f, \ell_f^{-\times V})$. We define $\ell^{V \times -}$ and
 542 $f^{V \times -}$ symmetrically.

543 ► **Definition 17** (Worlds-Agnostic Parallel Composition). For $[f]_W : \mathfrak{A} \rightarrow \mathfrak{B}$ and $[g]_V : \mathfrak{C} \rightarrow \mathfrak{D}$
 544 two morphisms, we define

$$545 \quad [f]_W \square [g]_V := [f^{-\times V} \square g^{W \times -}]_{W \times V} : (\mathfrak{A} \square \mathfrak{B}) \rightarrow (\mathfrak{C} \square \mathfrak{D})$$

546 In Figure 11, we show the result of this parallel composition when $[f]_{\{a, \star\}} = \mathbf{id}_{A: \{a\}}$ and
 547 $[g]_{\{b, \star\}} = \mathbf{id}_{A: \{b\}}$ for a color A .

548 B.1.2 Restricting the Worlds Set

549 We now tackle the sequential composition. For W a set of worlds, $w \subseteq W$, we define the
 550 restriction of $\ell : S \rightarrow W$ to $W \setminus w$ written $\ell^{\setminus w} : S \rightarrow W \setminus w$ as $\ell^{\setminus w}(s) = \{a \in \ell(s) \mid a \notin w\}$.
 551 For $[f]_W : \mathfrak{A} \rightarrow \mathfrak{B}$ we define $[f^{\setminus w}]_{W \setminus w} : \mathfrak{A} \rightarrow \mathfrak{B}$ as the morphism $(\mathcal{D}_f, \ell_f^{\setminus w})$.

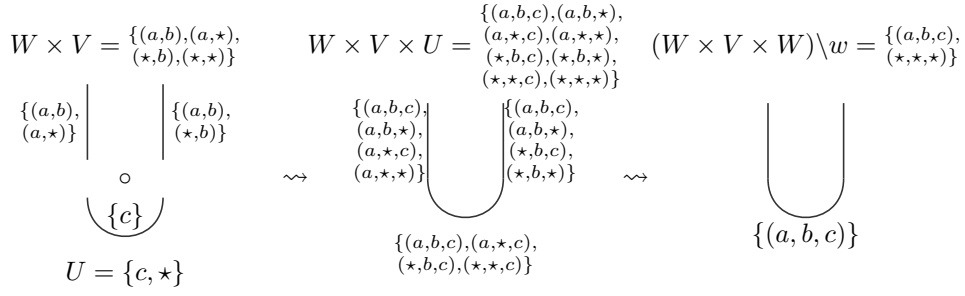
552 ► **Definition 18.** Given two morphisms $[f]_W : (\mathfrak{A}, \ell_{\mathfrak{A}}) \rightarrow (\mathfrak{B}, \ell_{\mathfrak{B}})$ and $[g]_V : (\mathfrak{B}, \ell_{\mathfrak{B}}) \rightarrow (\mathfrak{C}, \ell_{\mathfrak{C}})$,
 553 we write w for the smallest subset of $W \times V$ such that $(f^{-\times V})^{\setminus w}$ induces the same worlds
 554 labeling on \mathfrak{B} as $(g^{W \times -})^{\setminus w}$. We then define

$$555 \quad [g]_V \circ [f]_W := [(g^{W \times -})^{\setminus w} \circ (f^{-\times V})^{\setminus w}]_{(W \times V) \setminus w}$$

556 We continue the previous example in Figure 12 by composing the result with the Cup over
 557 $A : \{c, \star\}$. We proceed in two steps: first we handle the situation as if it was a parallel
 558 composition, leading to a diagram labeled over $W \times V \times U$, but with multiple contradictory
 559 labels on the wire. Then, we eliminate as few worlds as possible to make those labels
 560 compatible:

- 561 ■ We eliminate (a, b, \star) and (a, \star, \star) which are on the *left* label but not on the *bottom* one.
- 562 ■ We eliminate (\star, b, c) and (\star, \star, c) which are on the *bottom* label but not on the *left* one.
- 563 ■ We eliminate (\star, b, \star) which is on the *right* label but not on the *bottom* one. We would
 564 also eliminate (a, b, \star) if we had not done so already.
- 565 ■ We eliminate (a, \star, c) which is on the *bottom* label but not on the *right* one. We would
 566 also eliminate (\star, \star, c) if we had not done so already.

567 The eliminated worlds are $w = \{(a, b, \star), (a, \star, c), (\star, b, c), (a, \star, \star), (\star, b, \star), (\star, \star, c)\}$, and what
 568 remains is $\{(a, b, c), (\star, \star, \star)\}$.



■ **Figure 13** Example of Worlds-Agnostic Sequential Composition

B.1.3 The Worlds-Agnostic Category and its Semantics

569

570 \mathbf{MW}_\forall with the sequential and parallel composition as described above forms a category (up
571 to renaming of worlds), and is in fact an auto-dual compact closed colored PROP (up to
572 renaming of worlds).

573 ► **Proposition 19.** *The worlds-agnostic semantics $\llbracket - \rrbracket$ defined in Section 3 is a monoidal
574 functor from \mathbf{MW}_\forall to \mathbf{FdHilb} .*

575 **Proof.** We recall here the definition of the worlds-agnostic semantics of $[f]_W : \mathfrak{A} \rightarrow \mathfrak{B}$:

$$576 \quad \llbracket [f]_W \rrbracket := \left\{ \sum_{\substack{a \in W \\ (\mathfrak{A}, \ell_a^f) \Downarrow a = \mathfrak{A}' \\ (\mathfrak{B}, \ell_{\mathfrak{B}}^f) \Downarrow a = \mathfrak{B}'}} \llbracket [f]_a \rrbracket \right\}_{\mathfrak{A}' \in \mathfrak{A}^*, \mathfrak{B}' \in \mathfrak{B}^*}$$

577 From the definition of the worlds-agnostic compositions, we directly have:

$$578 \quad \llbracket [f]_W \square [g]_V \rrbracket_{(a,b)} = \llbracket [f]_W \rrbracket_a \otimes \llbracket [g]_V \rrbracket_b \quad \llbracket [g]_V \circ [f]_W \rrbracket_{(a,b)} = \llbracket [g]_V \rrbracket_b \circ \llbracket [f]_W \rrbracket_a$$

579 The functoriality with respect to the parallel composition is then immediate:

$$580 \quad \llbracket [f]_W \square [g]_V \rrbracket = \left\{ \sum \llbracket [f]_W \square [g]_V \rrbracket_{(a,b)} \right\} = \left\{ \sum \llbracket [f]_W \rrbracket_a \right\} \otimes \left\{ \sum \llbracket [g]_V \rrbracket_b \right\} = \llbracket [f]_W \rrbracket \otimes \llbracket [g]_V \rrbracket$$

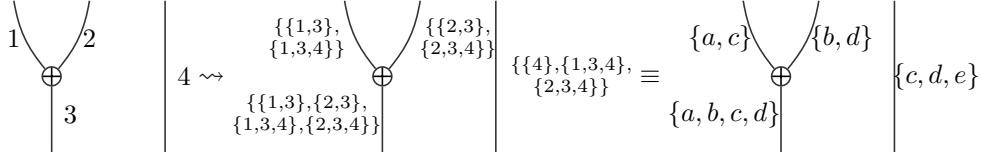
581 The functoriality with respect to the sequential composition is more subtle, as one must
582 carefully manipulate the indices of the sum and remark that the set of worlds w eliminated
583 by the worlds-agnostic composition satisfies the following:

$$584 \quad (a, b) \notin w \iff (\mathfrak{B}, \ell_{\mathfrak{B}}^f) \Downarrow a = (\mathfrak{B}, \ell_{\mathfrak{B}}^g) \Downarrow b \text{ where } \begin{array}{l} f : (\mathfrak{A}, \ell_{\mathfrak{A}}^f) \rightarrow (\mathfrak{B}, \ell_{\mathfrak{B}}^f) \\ g : (\mathfrak{B}, \ell_{\mathfrak{B}}^g) \rightarrow (\mathfrak{C}, \ell_{\mathfrak{C}}^g) \end{array}$$

585 Then, we have

$$586 \quad \llbracket [g]_V \circ [f]_W \rrbracket = \left\{ \sum \llbracket [g]_V \circ [f]_W \rrbracket_{(a,b)} \right\} = \left\{ \sum \llbracket [g]_V \rrbracket_b \right\} \circ \left\{ \sum \llbracket [f]_W \rrbracket_a \right\} = \llbracket [g]_V \rrbracket \circ \llbracket [f]_W \rrbracket$$

587



■ **Figure 14** A diagram, its canonical labeling, and its equivalent with worlds set $\{a, b, c, d, e, \star\}$

588 B.2 Keeping the Worlds Labeling Implicit

589 In Figure 5, we provide an example where one can use the equational theory without the
 590 worlds. In the previous section, we provided a worlds-agnostic semantics to our diagrams. It
 591 is reasonable to wonder how much can be done within \mathbf{C}_D , in other words without using the
 592 worlds labeling. In the following, we show how to provide a canonical labeling for diagrams
 593 of \mathbf{C}_D and deduce a denotational semantics for \mathbf{C}_D .

594 We consider a diagram \mathcal{D} of \mathbf{C}_D and we want to define a canonical worlds labeling over
 595 its set of wires \mathbb{W} . The core idea is that a world will be a set of wires that can cohabit with
 596 each others while satisfying the constraints of the nodes Plus, Tensor, etc. For example in
 597 Figure 13, there are four wires with wire 1 and 2 being incompatible due to the Plus, but
 598 also inseparable from wire 3, hence the canonical worlds set would be:

$$599 \quad \{\emptyset, \{1, 3\}, \{2, 3\}, \{4\}, \{1, 3, 4\}, \{2, 3, 4\}\}$$

600 More formally, a subset of wires $S \subseteq \mathbb{W}$ is said *valid* if the labeling $\ell_S : x \mapsto \{\star\}$ if $x \in S$
 601 and \emptyset otherwise is a valid labeling for \mathcal{D} , in other words if (\mathcal{D}, ℓ_S) is a morphism of $\mathbf{MW}_{\{\star\}}$.
 602 We will take for worlds set $W_{\mathcal{D}} \subseteq \mathcal{P}(\mathbb{W})$ the sets of valid subsets of wires. The labeling will
 603 simply be: $\ell_{\mathcal{D}} : x \mapsto \{S \in W_{\mathcal{D}} \mid x \in S\}$.

604 ► **Proposition 20.** *For every diagram \mathcal{D} of \mathbf{C}_D , $(\mathcal{D}, \ell_{\mathcal{D}})$ is a morphism of $\mathbf{MW}_{W_{\mathcal{D}}}$. Up to
 605 renaming of the worlds, this construction is a monoidal functor from \mathbf{C}_D to \mathbf{MW}_{\vee} .*

606 It follows that if we define $\llbracket \mathcal{D} \rrbracket$ as $\llbracket (\mathcal{D}, \ell_{\mathcal{D}}) \rrbracket$, this semantics is a monoidal functor from
 607 \mathbf{C}_D to \mathbf{FdHilb} . We note that it is enough to compute the semantics of the generators of
 608 \mathbf{C}_D (see FIGURE) to obtain the semantics for every diagram using $\llbracket \mathcal{D} \circ \mathcal{D}' \rrbracket = \llbracket \mathcal{D} \rrbracket \circ \llbracket \mathcal{D}' \rrbracket$
 609 and $\llbracket \mathcal{D} \square \mathcal{D}' \rrbracket = \llbracket \mathcal{D} \rrbracket \otimes \llbracket \mathcal{D}' \rrbracket$.

C Proofs of Universality and Uniqueness of Normal Form

Universality

In this section, we show that for every $\mathfrak{A}, \mathfrak{B}$ objects of \mathbf{C}_D , and for every linear operator

$$\Lambda = \begin{pmatrix} \lambda_{11} & \cdots & \lambda_{n1} & \lambda'_1 \\ \vdots & & \vdots & \vdots \\ \lambda_{1m} & \cdots & \lambda_{nm} & \lambda''_m \\ \lambda'_1 & \cdots & \lambda'_n & \lambda_0 \end{pmatrix} \in \mathbf{FdHilb} \left(\bigoplus_{\mathfrak{C} \in \mathfrak{A}^\bullet} \mathcal{H}_{\mathfrak{C}}, \bigoplus_{\mathfrak{F} \in \mathfrak{B}^\bullet} \mathcal{H}_{\mathfrak{F}} \right)$$

then the morphism $f = \text{iso}_{\mathfrak{B}}^{-1} \circ \lambda \circ \text{iso}_{\mathfrak{A}}$ defined in Figure 10 satisfies $\llbracket f \rrbracket = \Lambda$.

As stated in the corresponding section, there is one world for each scalar in λ , so we write a_{ij} the world associated to λ_{ij} and a'_i, a''_j, a_0 similarly, and W the set of all those worlds. Writing \bullet^k for $\square \dots \square \bullet$ (with k elements) and $\mathbb{1}_i^k$ for \bullet^k where the i -th \bullet has been replaced by $\mathbb{1}$, we have the following:

$$\begin{cases} \llbracket \lambda \rrbracket_{a_{ij}} : \mathbb{1}_i^n \rightarrow \mathbb{1}_j^m : x \mapsto \lambda_{ij} \cdot x \\ \llbracket \lambda \rrbracket_{a'_i} : \mathbb{1}_i^n \rightarrow \bullet^m : x \mapsto \lambda'_i \cdot x \\ \llbracket \lambda \rrbracket_{a''_j} : \bullet^n \rightarrow \mathbb{1}_j^m : x \mapsto \lambda''_j \cdot x \\ \llbracket \lambda \rrbracket_{a_0} : \bullet^n \rightarrow \bullet^m : x \mapsto \lambda_0 \cdot x \end{cases} \text{ where } \boxed{\lambda} := \begin{array}{c} \begin{array}{c} \dots \\ \lambda'_1 \\ \dots \end{array} \\ \begin{array}{c} \lambda_{11} \quad \lambda_{12} \\ \vdots \quad \vdots \\ \lambda_{nm} \end{array} \\ \begin{array}{c} \lambda''_1 \\ \dots \\ \lambda''_m \\ \lambda_0 \end{array} \end{array}$$

Additionally, one can show by induction that $\llbracket \text{iso}_{\mathfrak{A}} \rrbracket_{a_{ij}}$ and $\llbracket \text{iso}_{\mathfrak{A}} \rrbracket_{a'_i}$ (resp. $\llbracket \text{iso}_{\mathfrak{A}} \rrbracket_{a''_j}$ and $\llbracket \text{iso}_{\mathfrak{A}} \rrbracket_{a_0}$) are simply the projection on the i -th (resp. $(n+1)$ -th) element of the canonical basis, and $\llbracket \text{iso}_{\mathfrak{B}}^{-1} \rrbracket_{a_{ij}}$ and $\llbracket \text{iso}_{\mathfrak{B}}^{-1} \rrbracket_{a'_i}$ (resp. $\llbracket \text{iso}_{\mathfrak{B}}^{-1} \rrbracket_{a''_j}$ and $\llbracket \text{iso}_{\mathfrak{B}}^{-1} \rrbracket_{a_0}$) are simply the injection on the j -th (resp. $(m+1)$ -th) element of the canonical basis. Since we have $\llbracket f \rrbracket_a = \llbracket \text{iso}_{\mathfrak{B}}^{-1} \rrbracket_a \circ \llbracket \lambda \rrbracket_a \circ \llbracket \text{iso}_{\mathfrak{A}} \rrbracket_a$ for every $a \in W$, and $\llbracket f \rrbracket$ being the collection of all the $\llbracket f \rrbracket$, we obtain that $\llbracket f \rrbracket = \Lambda$.

Uniqueness of Normal Form

Let f and g be two diagrams in normal form (with respectively λ and μ as inner block), such that $\llbracket f \rrbracket = \llbracket g \rrbracket$ (the naming of the worlds is taken to be the same in both diagrams, and is the same as in the previous proof). By the definition of $\llbracket \cdot \rrbracket$, we have $\llbracket f \rrbracket_a = \llbracket g \rrbracket_a$ for every $a \in W$. We hence have $\llbracket \text{iso}_{\mathfrak{B}}^{-1} \rrbracket_a \circ \llbracket \lambda \rrbracket_a \circ \llbracket \text{iso}_{\mathfrak{A}} \rrbracket_a = \llbracket \text{iso}_{\mathfrak{B}}^{-1} \rrbracket_a \circ \llbracket \mu \rrbracket_a \circ \llbracket \text{iso}_{\mathfrak{A}} \rrbracket_a$.

Denoting $e_i^{\mathfrak{A}}$ (resp. $e_i^{\mathfrak{B}}$) the i -th element of the basis of \mathfrak{A} (resp. \mathfrak{B}), we have:

$$\begin{cases} \lambda_{ij} = e_j^{\mathfrak{B}^\dagger} \llbracket \text{iso}_{\mathfrak{B}}^{-1} \rrbracket_{a_{ij}} \circ \llbracket \lambda \rrbracket_{a_{ij}} \circ \llbracket \text{iso}_{\mathfrak{A}} \rrbracket_{a_{ij}} e_i^{\mathfrak{A}} = e_j^{\mathfrak{B}^\dagger} \llbracket \text{iso}_{\mathfrak{B}}^{-1} \rrbracket_{a_{ij}} \circ \llbracket \mu \rrbracket_{a_{ij}} \circ \llbracket \text{iso}_{\mathfrak{A}} \rrbracket_{a_{ij}} e_i^{\mathfrak{A}} = \mu_{ij} \\ \lambda'_i = \bullet^{\mathfrak{B}^\dagger} \llbracket \text{iso}_{\mathfrak{B}}^{-1} \rrbracket_{a'_i} \circ \llbracket \lambda \rrbracket_{a'_i} \circ \llbracket \text{iso}_{\mathfrak{A}} \rrbracket_{a'_i} e_i^{\mathfrak{A}} = \bullet^{\mathfrak{B}^\dagger} \llbracket \text{iso}_{\mathfrak{B}}^{-1} \rrbracket_{a'_i} \circ \llbracket \mu \rrbracket_{a'_i} \circ \llbracket \text{iso}_{\mathfrak{A}} \rrbracket_{a'_i} e_i^{\mathfrak{A}} = \mu'_i \\ \lambda''_j = e_j^{\mathfrak{B}^\dagger} \llbracket \text{iso}_{\mathfrak{B}}^{-1} \rrbracket_{a''_j} \circ \llbracket \lambda \rrbracket_{a''_j} \circ \llbracket \text{iso}_{\mathfrak{A}} \rrbracket_{a''_j} \bullet^{\mathfrak{A}} = e_j^{\mathfrak{B}^\dagger} \llbracket \text{iso}_{\mathfrak{B}}^{-1} \rrbracket_{a''_j} \circ \llbracket \mu \rrbracket_{a''_j} \circ \llbracket \text{iso}_{\mathfrak{A}} \rrbracket_{a''_j} \bullet^{\mathfrak{A}} = \mu''_j \\ \lambda_0 = \bullet^{\mathfrak{B}^\dagger} \llbracket \text{iso}_{\mathfrak{B}}^{-1} \rrbracket_{a_0} \circ \llbracket \lambda \rrbracket_{a_0} \circ \llbracket \text{iso}_{\mathfrak{A}} \rrbracket_{a_0} \bullet^{\mathfrak{A}} = \bullet^{\mathfrak{B}^\dagger} \llbracket \text{iso}_{\mathfrak{B}}^{-1} \rrbracket_{a_0} \circ \llbracket \mu \rrbracket_{a_0} \circ \llbracket \text{iso}_{\mathfrak{A}} \rrbracket_{a_0} \bullet^{\mathfrak{A}} = \mu_0 \end{cases}$$

where $|\mathfrak{B}|$ denotes the number of wires in \mathfrak{B} .

Hence, all coefficients in the scalars of f and g are the same. Since the structure is otherwise the same for f and g , they are the same diagram.

636 **D Proof of Soundness**

637 Given that most of the time, $\llbracket \text{gen} \rrbracket_a$ is the identity, the equations defining \equiv_W are quite
 638 straightforward to verify. We immediately have that \equiv_W is sound with respect to $\llbracket - \rrbracket_a$ for
 639 every $a \in W$. Since $\llbracket - \rrbracket$ is defined from $\llbracket - \rrbracket_a$, soundness with respect to $\llbracket - \rrbracket$ is also correct.
 640 We then handle the five additional equations of \equiv .

641 **Renaming** Applying a bijection to the worlds set W does not change the result computed
 642 by $\sum_{a \in W} \dots$, hence this equation is sound with respect to $\llbracket - \rrbracket$.

643 **Annihilation due to Scalars** This equation simply removes elements equal to zero from the
 644 sum $\sum_{a \in W} \dots$, hence it is sound with respect to $\llbracket - \rrbracket$.

645 **Annihilation due to Plus** Since \oplus is a biproduct in \mathbf{FdHilb} , we have $\text{proj}_H^{H \oplus K} \circ \text{inj}_H^{H \oplus K} =$
 646 \mathbf{id}_H , $\text{proj}_K^{H \oplus K} \circ \text{inj}_K^{H \oplus K} = \mathbf{id}_K$, $\text{proj}_K^{H \oplus K} \circ \text{inj}_H^{H \oplus K} = 0$ and $\text{proj}_H^{H \oplus K} \circ \text{inj}_K^{H \oplus K} = 0$. One
 647 can then simply remove from the $\sum_{a \in W} \dots$ the elements equal to zero, which proves
 648 that *Annihilation due to Plus* is sound with respect to $\llbracket - \rrbracket$.

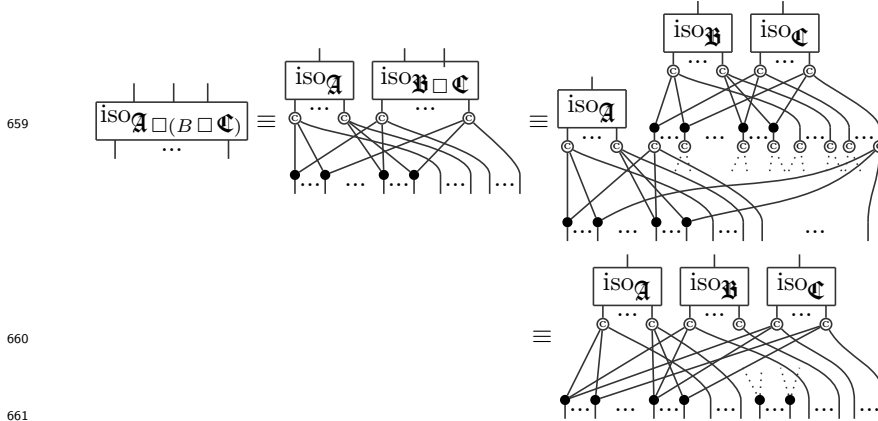
649 **Splitting due to Scalars** Since \mathbf{FdHilb} is a vector space, we have $(s + t) \cdot f = s \cdot f + t \cdot f$,
 650 which is exactly the property required for this equation to be sound for $\llbracket - \rrbracket$.

651 **Splitting due to Plus** Similarly, we have in \mathbf{FdHilb} the property that $\mathbf{id}_{H \oplus K} = \text{inj}_H^{H \oplus K} \circ$
 652 $\text{proj}_H^{H \oplus K} + \text{inj}_K^{H \oplus K} \circ \text{proj}_K^{H \oplus K}$, which is the property required for this equation to be
 653 sound for $\llbracket - \rrbracket$.

E Proofs for Completeness

E.1 The Normal Form

Proof of Lemma 9. First notice that in both $\text{iso}_{\mathcal{A} \square (\mathcal{B} \square \mathcal{C})}$ and $\text{iso}_{(\mathcal{A} \square \mathcal{B}) \square \mathcal{C}}$, we can use the bialgebra between contractions and unitors, followed by their respective fusions in the following way:



and similarly for $\text{iso}_{(A \square B) \square C}$. It then suffices to check which contractions the bottom unitors are linked to. Naming the i th contraction existing iso_A as a_i , and similarly for B and C , we can see that for each triple (a_i, b_j, c_k) there is exactly one unitor connected to precisely contraction a_i, b_j and c_k , in both diagrams. The same is true for every pair (a_i, b_j) , (a_i, c_k) and (b_j, c_k) , as well as for every 1-tuple (a_i, \cdot) , (b_j, \cdot) and (c_k, \cdot) . This shows that both diagrams are equal up to rearranging of the outputs. ◀

► **Lemma 21.** Notice that $\text{iso}_{A \oplus B} \equiv \text{iso}_{A \square B}$ and $\text{iso}_{A \otimes B} \equiv \text{iso}_{A \square B}$.

► **Lemma 22.** We have the following identities:

$\text{iso}_{\mathcal{A}} \equiv \mathcal{A}$ and $\text{iso}_{\mathcal{A}}^{-1} \equiv \dots$ where $\ell(s_i) \cap \ell(s_j) = \emptyset$ when $i \neq j$.

Proof of Lemma 22. We will use the following identities:

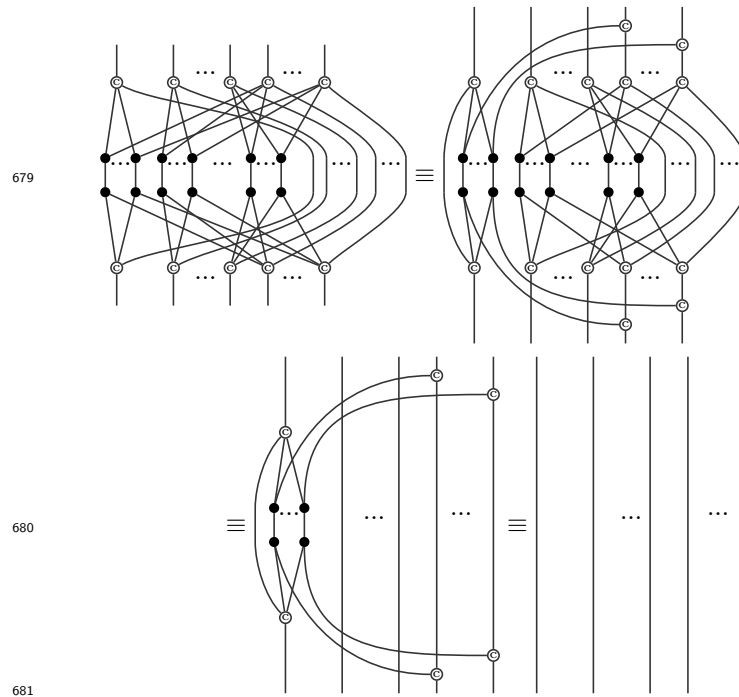
$\text{Diagram} \equiv \dots$ when all the wires a_{ij}, b_i, c_j are mutually exclusive (1)

We can show this result by induction on n and m . Case $(0, m)$ is obvious. Case $(1, 1)$ can be proven easily using worlds sets:

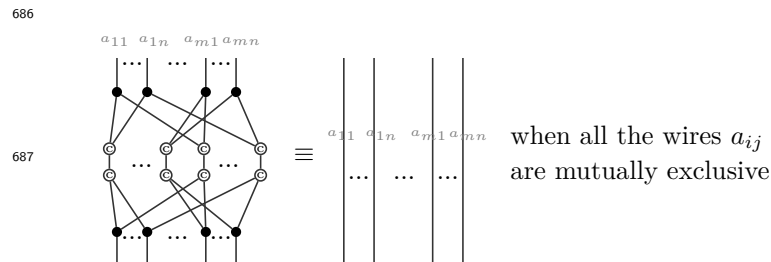
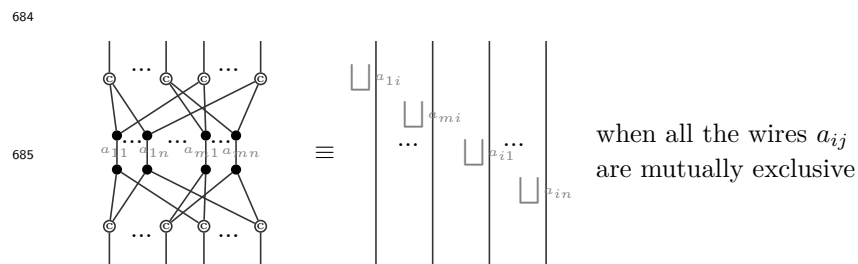
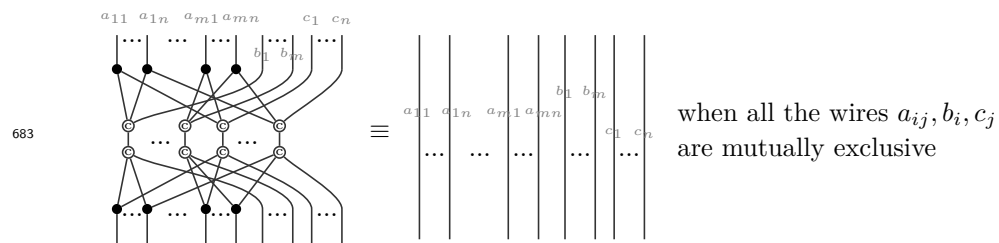
$\text{Diagram} \equiv \dots \equiv ab \quad ac$

23:24 The Many-Worlds Calculus: Representing Quantum Control

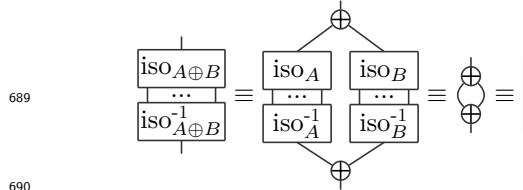
677 For any n and m , we can then prove the case $(n + 1, m)$ using the cases (n, m) and $(1, m)$
 678 (the case $(n, m + 1)$ is completely symmetric):



682 In a similar way, it is possible to show the following three identities:



688 ■ $[\text{iso}_A^{-1} \circ \text{iso}_A]$: The result is obvious in cases $\mathbb{1}$ and \mathbb{Z} . For $A \oplus B$:



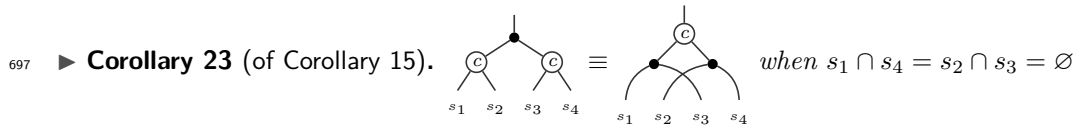
691 The proof is similar for \otimes and \square using the previous identities.

692 ■ $[\text{iso}_A \circ \text{iso}_A^{-1}]$: The result is again obvious for $\mathbb{1}$ and \mathbb{Z} . The general result is easy to prove
 693 by induction using the above identities.

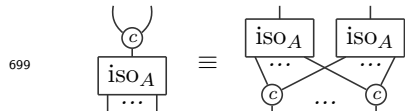
694

695 E.2 The Completeness

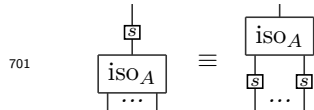
696 We want to show here the results of Section 5.2. To do so we will first derive a few lemmas:



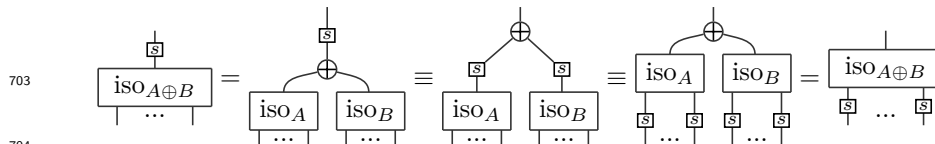
698 ► **Corollary 24** (of Corollary 15). *Single-colored isos distribute over the contraction:*



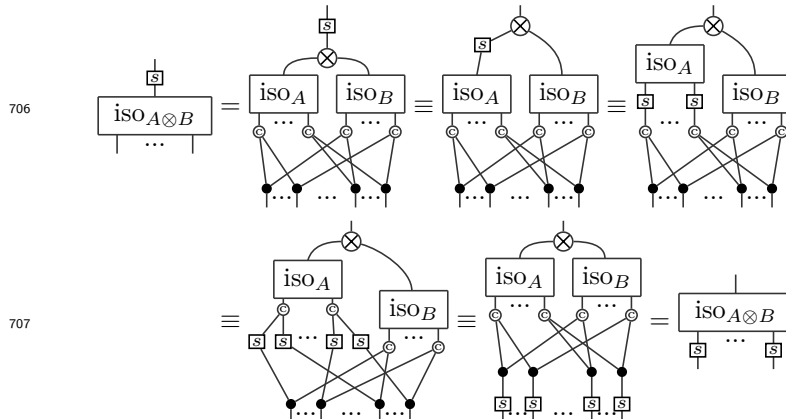
700 ► **Lemma 25.** *Scalars distribute over single-colored isos:*



702 **Proof.** The result is obvious for $\mathbb{1}$ and \mathbb{Z} . For $A \oplus B$:



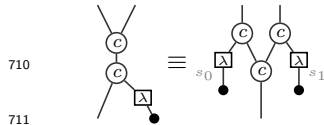
705 For $A \otimes B$:



23:26 The Many-Worlds Calculus: Representing Quantum Control

708
709

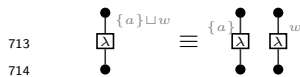
► **Corollary 26** (of Corollary 15).



710
711

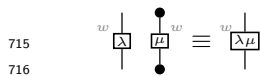
with $s_0 \cap s_1 = \emptyset$.

► **Corollary 27** (of Lemma 13).



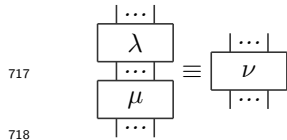
713
714

► **Corollary 28** (of Lemma 13).



715
716

► **Lemma 29.**

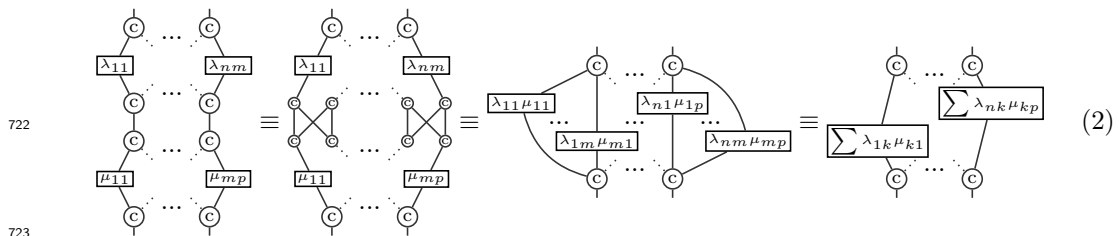


717
718

with $\nu_{ij} = \sum_k \lambda_{ik} \mu_{kj} + \lambda'_i \mu''_j$, $\nu'_i = \sum_k \lambda_{ik} \mu''_k + \mu_0 \lambda'_i$, $\nu''_i = \sum_k \lambda''_k \mu_{ki} + \lambda_0 \mu''_i$ and $\nu_0 = \lambda_0 \mu_0 + \sum_k \lambda''_k \mu'_k$.

720

Proof. Consider first the following simpler case:



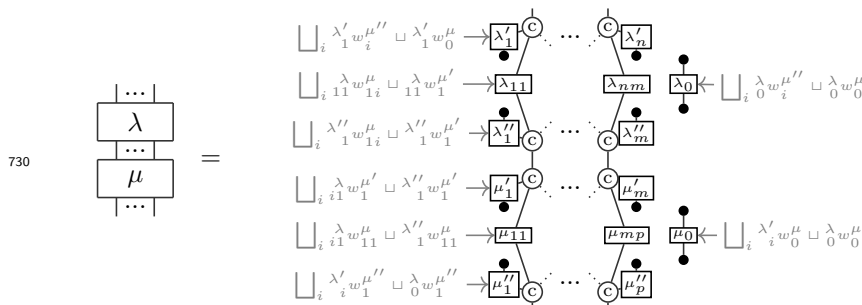
722
723

Importantly, in the last diagram, we have that each internal wire lives in a set of worlds that has empty intersection with any of the others. It is hence possible to rename this set of worlds to a singleton, and to do it for each internal wire.

724
725
726

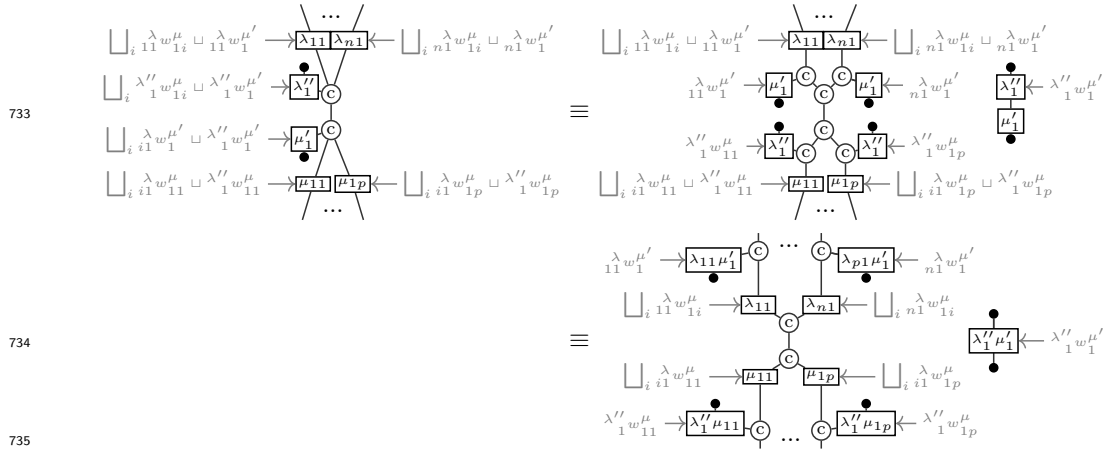
For the general case, it is important to properly look at the worlds. Denoting w_{ij}^λ the (singleton) world that bears λ_{ij} , $w_i^{\lambda'}$ the one that bears λ'_i , $w_i^{\lambda''}$ the one that bears λ''_i and w_0^λ the one that bears λ_0 in matrix block λ , and similarly in matrix block μ , we get:

729



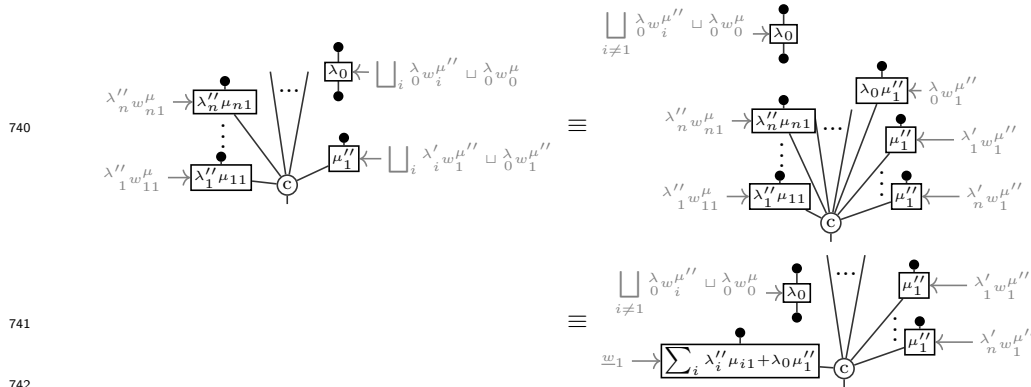
730

731 where $a_x w_y^b = \{(z_1, z_2) \mid z_1 \in w_x^a, z_2 \in w_y^b\}$. Let us now look at how to push the middle
 732 dangling wires through. At the first junction, we have:



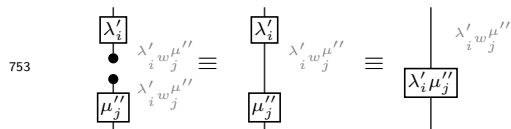
736 Doing this at every junction wire, we are able to push all the middle dangling wires to the
 737 boundaries, and where the middle is in the shape of the previous simple case (2).

738 Let us now look at the boundaries, for instance the first output. After pushing all dangling
 739 wires, and taking into account the scalar diagram λ_0 , we have:



743 where on the right hand side of the diagram, we decomposed the dangling wire by its different
 744 elementary worlds. One of them was shared with the λ_0 scalar, and using Lemmas 27 and
 745 28, we could bring them together. Then on the left hand side, we used the fact that none of
 746 the $\lambda'_i w_{i1}^{\mu''}$ nor the $\lambda_0 w_1^{\mu''}$ could be found elsewhere in the diagram, so we could sum all the
 747 scalars without altering the rest of the diagram.

748 Doing so for all boundaries, scalars λ_0 and μ_0 are only left on world $\lambda_0 w_0^{\mu''}$. We can then use
 749 Lemma 28 to bring them together. All the worlds on right hand side of the last diagram
 750 above $\lambda'_i w_1^{\mu''}$ can be found on the boundaries at the top (i.e. for each input i and output j
 751 there is world $\lambda'_i w_j^{\mu''}$ shared between the two). Using the small following derivation, we can
 752 bring them together:

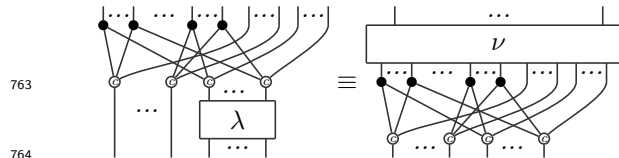


754 Now this world can be found nowhere else in the diagram, it is hence possible to sum the
 755 scalar with the other that has the same neighbourhood, obtained from (2).

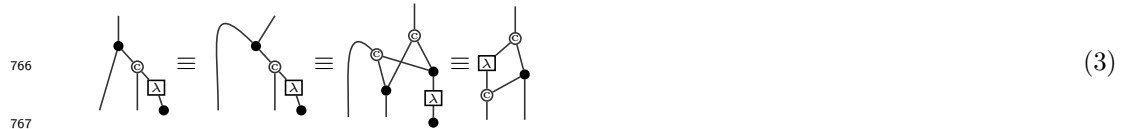
23:28 The Many-Worlds Calculus: Representing Quantum Control

756 It remains to deal with the scalars. We have $\lambda_0 \mu_0$ on world $\lambda_0 w_0^\mu$ from what we just
 757 discussed. We also have scalar $\lambda_i' \mu_i'$ on world $\lambda_i' w_i^{\mu'}$ from the dangling wires at the junction.
 758 All these worlds are again mutually exclusive and found nowhere else in the diagram. We
 759 can hence sum them and get $\lambda_0 \mu_0 + \sum_i \lambda_i' \mu_i'$ as the scalar on the side. After all this, the
 760 diagram is in the appropriate form (up to renaming of worlds), with all worlds in the internal
 761 wires being mutually exclusive. ◀

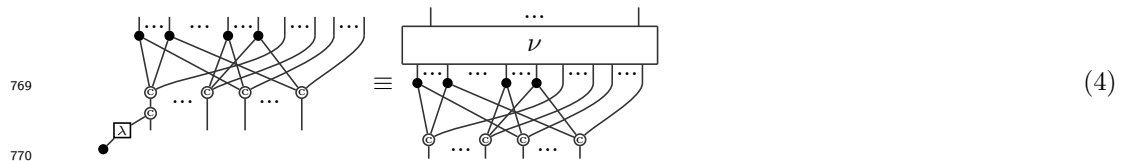
762 ▶ **Lemma 30.** For any "matrix block" λ , there exists a "matrix block" ν such that:



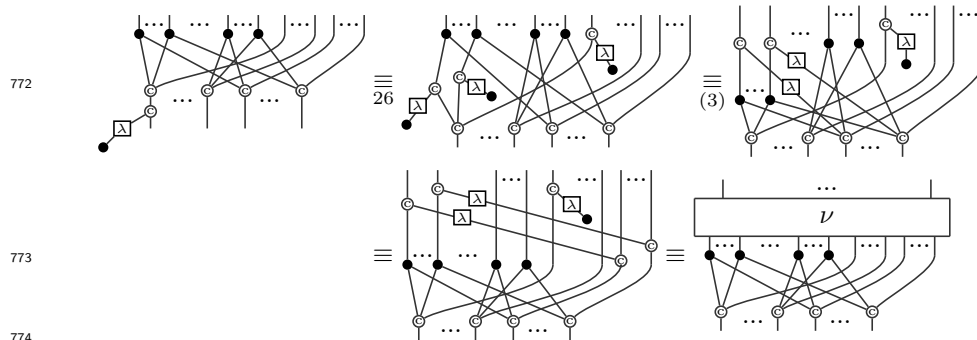
765 **Proof.** We show the result through several steps. First we show the following:



768 Then, we deal with the top dangling scalars and prove that:

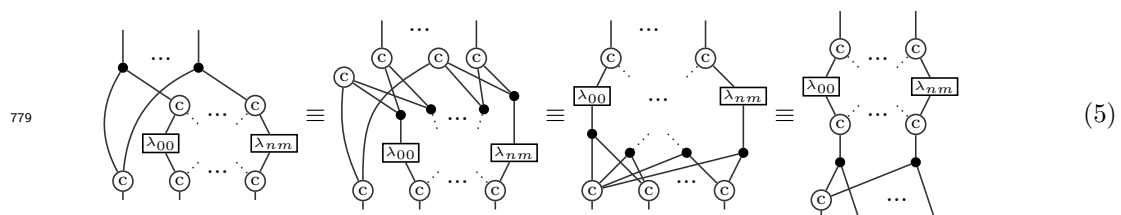


771 Indeed:



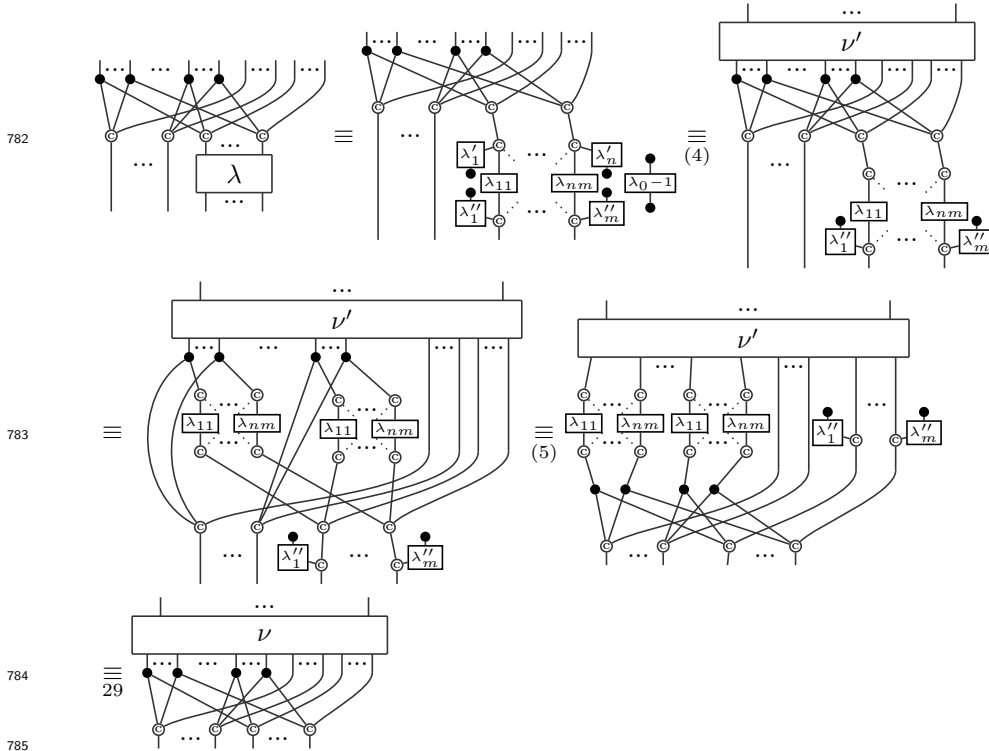
775 This case generalizes to any bottom wire the dangling scalar is applied on. When we have
 776 several of them, we can make them go through the top part in turns, then aggregate them
 777 under a single "matrix block" using Lemma 29.

778 We then derive the following equation:



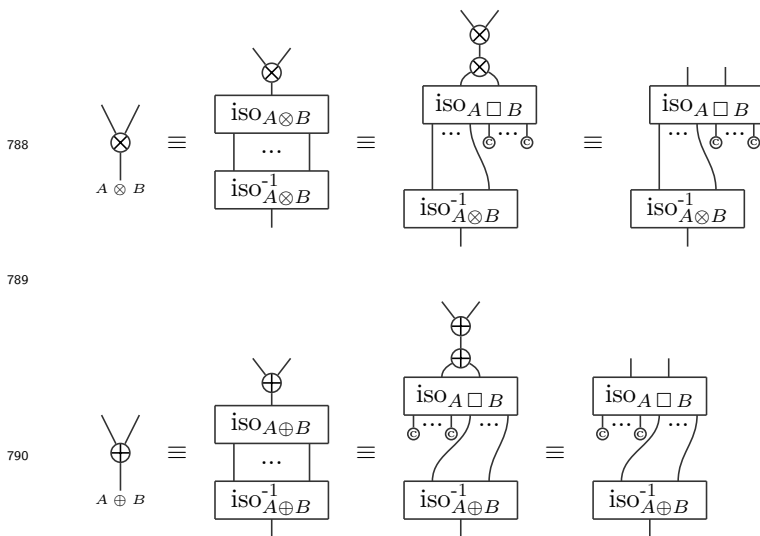
780

781 We can finally prove the lemma:



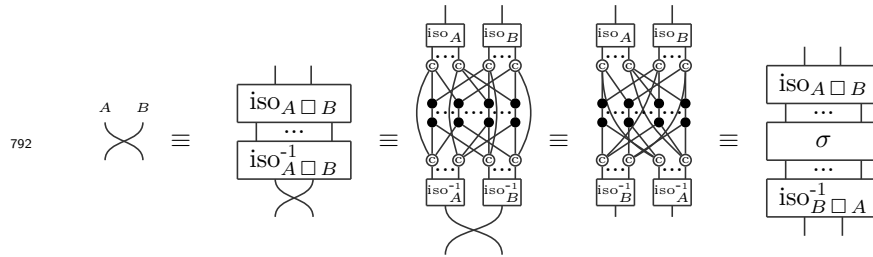
787 We can now move on to show that generators can be put in normal form:

Proof of Proposition 11.

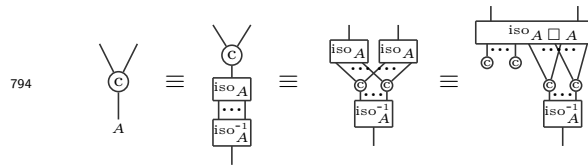


23:30 The Many-Worlds Calculus: Representing Quantum Control

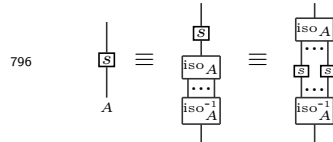
791



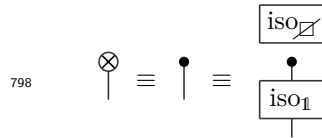
793 with σ a simple permutation of wires.



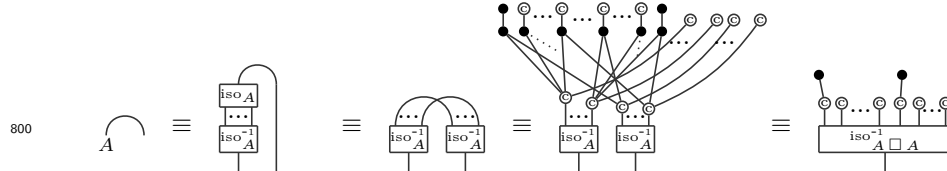
795



797



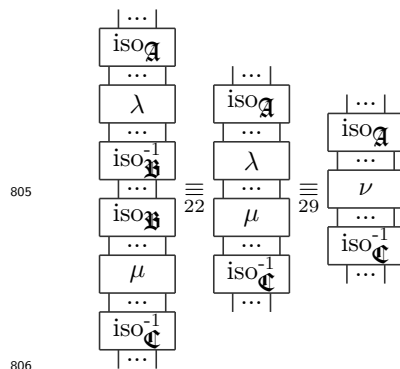
799



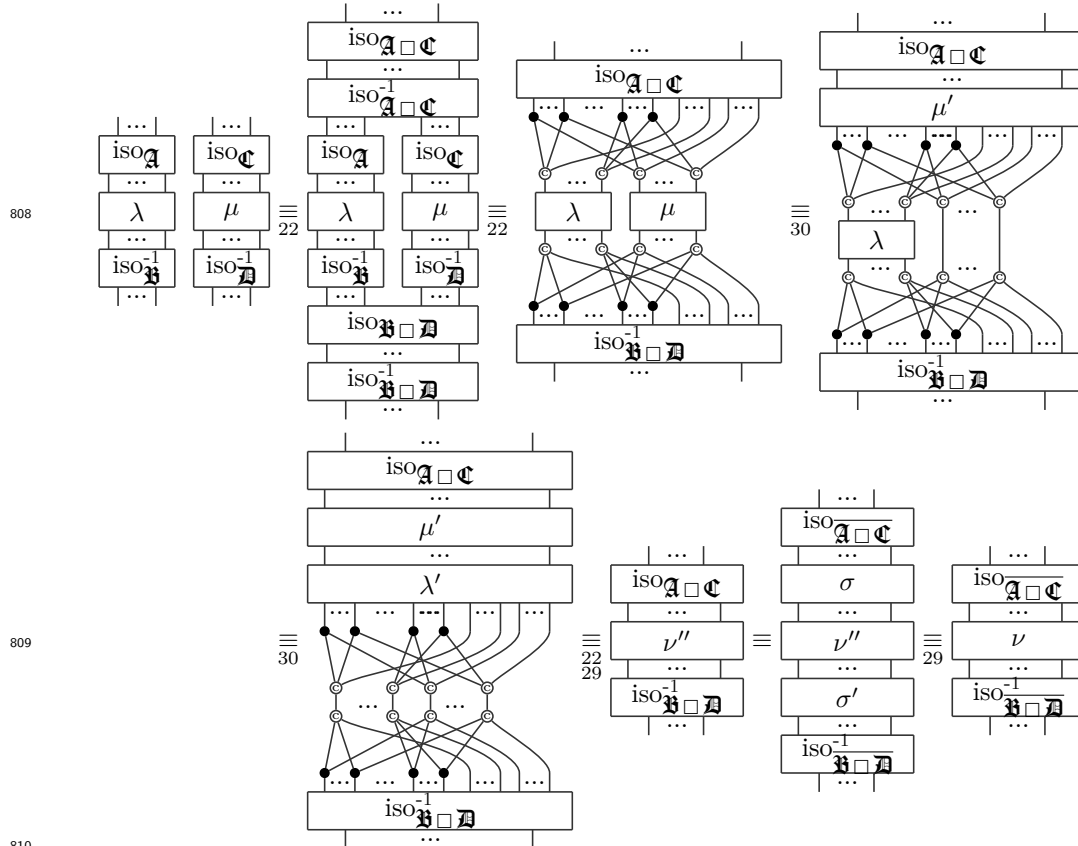
801 The upside-down versions of the generators are provided in exactly the same way (but
802 upside-down). ◀

803 And then that compositions of diagrams in normal form can be put in normal form:

804 **Proof of Proposition 12.** In the case of sequential composition:



807 In the case of parallel composition:



811 where $\overline{a \square c}$ represents the canonical choice of composition with \square (and similarly for $\overline{b \square d}$).

812