

Towards Term-based Verification of Diagrammatic Equivalence

Julie Cailler^{[0000-0002-6665-8089]¹}, Noé Delorme^{[0000-0002-4544-9691]¹},
Simon Perdrix^{[0000-0002-1808-2409]¹}, and Sophie Tourret^{[0000-0002-6070-796X]^{1,2}}

¹ Université de Lorraine, CNRS, INRIA, LORIA, Nancy, France

² Max Plank Institute for Informatics, Saarbrücken, Germany
`firstname.lastname@inria.fr`

Abstract. A string diagram is a two-dimensional graphical representation that can be described as a one-dimensional term generated from a set of primitives using sequential and parallel compositions. Since different syntactic terms may represent the same diagram, this syntax is quotiented by a collection of coherence equations expressing equivalence up to deformation. This work lays foundations for automated reasoning about diagrammatic equivalence, motivated primarily by the verification of quantum circuit equivalences. We consider two classes of diagrams, for which we introduce normalizing term rewriting systems that equate diagrammatically equivalent terms. In both cases, we prove termination and confluence with the help of the proof assistant Isabelle/HOL.

1 Introduction

Diagrammatic reasoning has proved to be a powerful and expressive framework across a wide range of domains, including quantum computing [1, 3, 9, 12, 16, 18], linear algebra [5, 22], boolean circuits [10], and natural language processing [11, 25]. By representing computations as compositional diagrams in two dimensions, these formalisms offer intuitive and structural insights that are difficult to witness otherwise. As diagrammatic methods continue to improve, the need for automated reasoning techniques expands. In particular, problems such as equivalence checking, verification, certification, and diagrammatic manipulation require algorithmic foundations in order to scale.

An important motivation for our work arises from quantum computing, a field in which diagrammatic reasoning has been extensively developed [7–9, 13]. In particular, quantum circuits admit natural diagrammatic representations, and reasoning about their equivalence is essential for tasks such as optimization, compilation, and verification. Our work is motivated by the goal of obtaining semi-automated verification techniques for quantum circuit equivalences. In this paper, we establish foundational results, in the form of normal forms and verified rewriting systems, that constitute essential building blocks towards a full-fledged proof certification pipeline for quantum circuits.

One way to represent diagrammatic reasoning problems suitably for automation is to adopt a syntactic term-based representation of diagrams. By encoding

diagrams as terms equipped with operations for sequential and parallel compositions, diagrammatic deformations can be translated into equations between terms. This enables the use of term rewriting to prove diagram equivalences, including standard techniques for proving termination and confluence, for which there exists automated tools [6, 15, 24, 26]. As a result, the term-based representation provides a natural framework for certifying diagrammatic equivalence in a rigorous and machine-checkable way.

The natural mathematical framework to encode diagrams as terms is monoidal categories, and more precisely PROs and PROPs [21]. Both structures are similar, with one distinction: a PROP additionally allows swapping wires. Formally, the terms are freely generated from a set of generators—corresponding to black box processes with a certain amount of input and output wires—and can be composed in two ways: in sequence with the operator $;$ and in parallel with the operator $\otimes$. Depending on the order in which the generators are syntactically composed, multiple terms may represent the same diagrammatic evolution. For instance, $(u_1 ; u_2) \otimes (v_1 ; v_2)$ and $(u_1 \otimes v_1) ; (u_2 \otimes v_2)$ can both be graphically depicted as follows.

$$\begin{array}{c} \boxed{u_1} \boxed{u_2} \\ \boxed{v_1} \boxed{v_2} \end{array}$$

Hence, the syntax of terms is quotiented by a collection of *coherence equations* capturing the intuitive notion of diagram equivalence. These equations can be turned into rewrite rules, allowing the rewrite of equivalent terms.

Contributions. We focus on two diagrammatic equivalence problems. In the first one, we consider arbitrary state-and-effect-free PROs, meaning that all generators have at least one input wire and one output wire. We define a normal form together with an associated normalizing rewriting system, and prove termination and confluence. Then, any pair of diagrammatically equivalent terms can be rewritten one into the other using the normalization procedure. In the second problem, we consider the PRO of permutations, which is precisely the PROP with no generators. Building upon our previous normal form, we define a canonical form and a canonizing rewriting system that we also prove to be terminating and confluent. This ensures the existence and uniqueness of the canonical form for any permutation. In both cases, the key steps of the termination and confluence proofs are verified using the proof assistant Isabelle/HOL.

Related Work. Diagrammatic equivalence has also been studied with a fundamentally different approach: instead of working with a syntactic term representation, diagrams are encoded as hypergraphs and rewriting is studied at the level of graph structures [4]. Then, diagrammatic equivalence is witnessed by hypergraph isomorphism. In contrast, our work remains entirely in the term-based setting and directly equips terms with rewriting systems yielding explicit normal forms that we plan to use for certification. Related questions have also been investigated in the context of the ZX-calculus, notably with the tool VyZX [20].

While sharing the same overall goal of certifying diagrammatic equivalences, their approach focuses on filling the gap between the use of inductive datatypes in proof assistants and the use of graph representation in ZX-calculus. Diagrammatic equivalence in the PRO of permutations has already been addressed, but from a diagrammatic perspective, where equivalence is handled implicitly via graphical transformation [19]. Although the resulting normal form is closely related in shape to ours, this work remains purely graphical and does not account for the syntactic order of sequential and parallel compositions.

Outline of the Paper. In Section 2, we fix the notations and give term-based definitions of PROs and PROPs. In Section 3, we introduce the normal form for an arbitrary state-and-effect-free PRO and prove the termination and confluence of its underlying term rewriting system. In Section 4, we introduce the canonical form for the PRO of permutations and prove the termination and confluence of its underlying term rewriting system. In Section 5, we provide our concluding remarks. The Isabelle companion files are available on Zenodo [14].

2 Background

The standard mathematical framework for diagrammatic reasoning is the one of *monoidal categories*. In particular, circuit-based graphical languages—such as quantum circuits—are defined as a particular kind of monoidal categories, namely PROs or PROPs (which stands for *PROduct and Permutation categories*). A PRO is a strict monoidal category generated by a single object, and a PROP is additionally symmetric (meaning that there are swap primitives allowing to exchange wires). For non-category theorists, we provide self-contained definitions of PROs and PROPs in this section. Roughly speaking, these structures are generated from a set of generators—corresponding to basic gates—via sequential and parallel composition. Graphically, this amounts to represent the generators as boxes and connect them with wires.

2.1 Terms and their Diagrammatic Representation

Let Σ be a set of generators equipped with two functions $\text{dom}, \text{cod} : \Sigma \rightarrow \mathbb{N}$. We inductively define the set of terms $\langle \Sigma \rangle$ induced by Σ , and extend $\text{dom}, \text{cod} : \langle \Sigma \rangle \rightarrow \mathbb{N}$ as follows, using the notation $t : n \rightarrow m \in \langle \Sigma \rangle$ (or simply $t : n \rightarrow m$ if Σ is clear from the context) when $t \in \langle \Sigma \rangle$, $\text{dom}(t) = n$ and $\text{cod}(t) = m$.

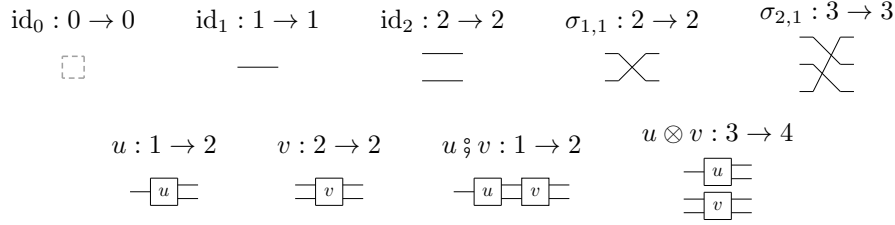
$$\frac{g \in \Sigma}{g : \text{dom}(g) \rightarrow \text{cod}(g) \in \langle \Sigma \rangle} \quad \frac{n \in \mathbb{N}}{\text{id}_n : n \rightarrow n \in \langle \Sigma \rangle}$$

$$\frac{u : n \rightarrow p \in \langle \Sigma \rangle \quad v : p \rightarrow m \in \langle \Sigma \rangle}{(u \circ v) : n \rightarrow m \in \langle \Sigma \rangle} \quad \frac{u : n \rightarrow m \in \langle \Sigma \rangle \quad v : p \rightarrow q \in \langle \Sigma \rangle}{(u \otimes v) : n + p \rightarrow m + q \in \langle \Sigma \rangle}$$

The *domain* and *codomain* of a term $t \in \langle \Sigma \rangle$ are given by $\text{dom}(t)$ and $\text{cod}(t)$, representing its number of inputs and outputs, respectively. We define the *defect*

of t by $\text{def}(t) := \text{dom}(t) - \text{cod}(t)$. A term t is called a *state* if $\text{dom}(t) = 0$, and an *effect* if $\text{cod}(t) = 0$. Moreover, for each $t \in \langle \Sigma \rangle$, let $\text{gen}(t) \subseteq \Sigma$ denote the set of generators appearing in t . This set is defined inductively by $\text{gen}(\text{id}_n) = \emptyset$, $\text{gen}(g) = \{g\}$ for $g \in \Sigma$, and $\text{gen}(u \circ v) := \text{gen}(u \otimes v) := \text{gen}(u) \cup \text{gen}(v)$.

A common set of generators is the set of swaps $\Sigma_0 := \{\sigma_{n,m} : n, m \in \mathbb{N}\}$ with $\text{dom}(\sigma_{n,m}) = \text{cod}(\sigma_{n,m}) = n + m$. Intuitively, $\sigma_{n,m}$ represents a diagram that exchanges two registers consisting of n and m wires, respectively. Terms admit a natural graphical representation as string diagrams. A term $t : n \rightarrow m$ is depicted as a box with n input wires and m output wires; the identity id_n is depicted as n parallel wires; and the swap $\sigma_{n,m}$ is depicted as a swap of the n and m wires. Here are some examples.



Moreover, given $r, s \in \mathbb{N}$, the set of contextualized terms $\langle \Sigma \rangle_{r,s}$ is defined as the terms containing a single occurrence of an additional generator $[\cdot] : r \rightarrow s$ called a *hole*. Then, given a contextualized term $K[\cdot] : n \rightarrow m \in \langle \Sigma \rangle_{r,s}$ and a term $t : r \rightarrow s \in \langle \Sigma \rangle$, the substitution $K[t] : n \rightarrow m \in \langle \Sigma \rangle$ is the term where the unique hole $[\cdot]$ in $K[\cdot]$ has been replaced by t .

2.2 Diagrammatic Equivalence

Up to here, terms are purely syntactic, and two terms are only equal if they are constructed exactly in the same way. However, several terms may be represented by the same diagram. Thus, we need the equivalence equations known as the *coherence equations*, depicted in Figure 1, to precisely capture the intuitive notion of diagram deformation, which yields our definition of PROs and PROPs.

The PRO freely generated by the set of generators Σ , denoted $\mathbf{PRO}_\Sigma$, is defined as the collection of equivalence classes of terms $\langle \Sigma \rangle$ under Equations (1) to (6). Furthermore, given a set of equations $\mathcal{E}$ between terms, let $\mathbf{PRO}_{\Sigma, \mathcal{E}}$ be the PRO whose equivalence classes are the ones of $\mathbf{PRO}_\Sigma$ additionally quotiented by the equations of $\mathcal{E}$. The equivalence relation is denoted $\mathbf{PRO}_{\Sigma, \mathcal{E}} \vdash t_1 = t_2$ when t_1 can be rewritten into t_2 using Equations (1) to (6) together with the equations in $\mathcal{E}$. The equivalence relation is closed under context, meaning that if $\mathbf{PRO}_{\Sigma, \mathcal{E}} \vdash t_1 = t_2$ where $t_i : r \rightarrow s$, then $\mathbf{PRO}_{\Sigma, \mathcal{E}} \vdash K[t_1] = K[t_2]$ for any contextualized term $K[\cdot] \in \langle \Sigma \rangle_{r,s}$.

The PROP freely generated by the set of generators Σ , denoted $\mathbf{PROP}_\Sigma$, is defined as $\mathbf{PROP}_\Sigma := \mathbf{PRO}_{\Sigma_0 \cup \Sigma, \mathcal{E}_0}$ where $\mathcal{E}_0$ is the set containing Equations (7) to (10). That is, the terms are induced by Σ as well as the set of swaps Σ_0 , and are quotiented by Equations (1) to (10) to form the equivalence class of $\mathbf{PROP}_\Sigma$.

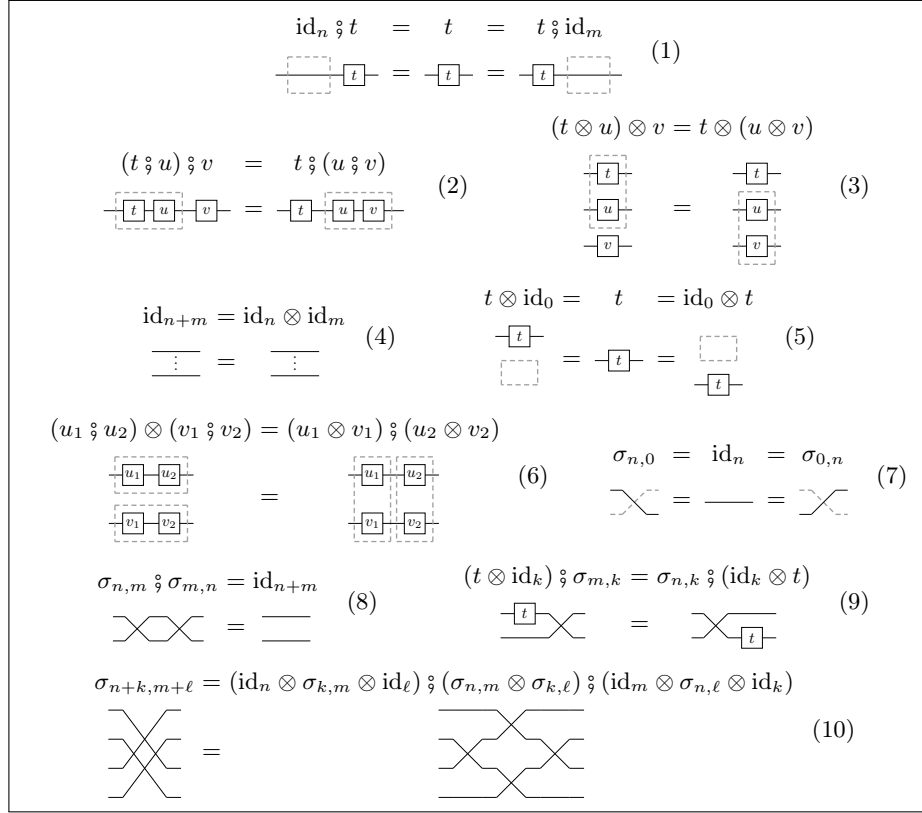


Fig. 1: Equations (1) to (6) are the coherence equations of PROs. Equations (1) to (10) are the coherence equations of PROPs. Dotted boxes and wires highlight the correspondence with the syntax. For convenience, we can draw a single wire to represent multiple wires when it is clear from the context.

Intuitively, compared to PROs, PROPs can bend the wires and gates can slide along those bends.

In this paper, we focus on two diagrammatic equivalence problems. For the first one (Section 3), we consider an arbitrary *state-and-effect-free* PRO, meaning that the set of generators is arbitrary but contains no states and no effects. In the second one (Section 4), we consider the PRO of permutations $\mathbf{PRO}_{\Sigma_0, \mathcal{E}_0}$, which is precisely the simplest PROP that one can think about, namely $\mathbf{PROP}_\emptyset$.

2.3 Notations

A list of p elements of A is denoted $ks = [ks_1, \dots, ks_p] \in A^p$. The list with 0 elements is denoted $[] \in A^0$. One can add $x \in A$ to the left of $ks \in A^p$ with the operator $\#$ defined as $x \# ks := [x, ks_1, \dots, ks_p] \in A^{p+1}$. More generally, one can define the concatenation of two lists $ks \in A^p$ and $ls \in A^q$ with the operator $@$ defined as $ks @ ls := [ks_1, \dots, ks_p, ls_1, \dots, ls_q] \in A^{p+q}$. If $(A, +)$ is a

group, given $ks \in A^p$ and $x \in A$, we define $ks + x := [ks_1 + x, \dots, ks_q + x]$ and $ks - x := ks + (-x)$.

Term rewriting systems (TRS) are defined as usual [2]. In particular, given two terms t_1 and t_2 , we write $t_1 \rightarrow_{\mathcal{R}} t_2$ when t_1 can be rewritten into t_2 using exactly one step of the rewriting system $\mathcal{R}$, and $t_1 \rightarrow_{\mathcal{R}}^* t_2$ when t_1 can be rewritten into t_2 using any finite number (including zero) of steps of $\mathcal{R}$. We write $t \not\rightarrow_{\mathcal{R}}$ when no rules of $\mathcal{R}$ can be applied from the term t .

3 Term Rewriting in a State-and-effect-free PRO

In this section we fix Σ a state-and-effect-free set of generators, meaning that each generator $g \in \Sigma$ satisfies $\text{dom}(g) > 0$ and $\text{cod}(g) > 0$. This assumption enables termination and confluence of our TRS while remaining sufficiently general for a wide range of practical applications. We consider $\mathbf{PRO}_{\Sigma}$ and solve its underlying diagrammatic equivalence problem: given two terms $t_1, t_2 \in \langle \Sigma \rangle$ such that $\mathbf{PRO}_{\Sigma} \vdash t_1 = t_2$, how to effectively rewrite t_1 into t_2 using the coherence equations? To do so, we introduce a normal form and an associated rewriting system that we prove to be terminating and confluent.

3.1 Normal Form and Preprocessing

We define a *pre-normal form* as a sequence of layers, a layer being a single generator g sandwiched between identities: $L_{\ell}^k(g) := \text{id}_k \otimes (g \otimes \text{id}_{\ell})$ where $k, \ell \in \mathbb{N}$ and $g \in \Sigma$. We say that g has *upper width* k , *lower width* ℓ , and *height* $k + \text{dom}(g)$.

A term $\mathbf{L}_{\ell \# \ell s}^{ks}(gs)$, defined as follows, is a non-empty sequence of $p > 0$ layers, characterized by the list $gs \in \Sigma^p$ of the associated generators and the lists $ks, \ell s \in \mathbb{N}^p$ of their corresponding upper and lower widths.

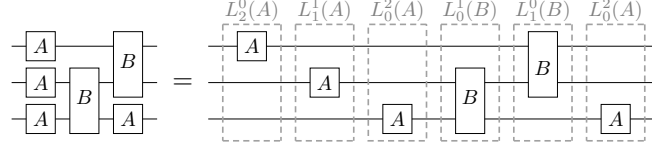
$$\mathbf{L}_{\ell \# \ell s}^{ks}(g \# gs) := \begin{cases} L_{\ell}^k(g) & \text{if } ks = \ell s = gs = [] \\ L_{\ell}^k(g) ; \mathbf{L}_{\ell s}^{ks}(gs) & \text{otherwise} \end{cases}$$

Notice that to guarantee that $\mathbf{L}_{\ell s}^{ks}(gs)$ is well-defined, we assume throughout this paper that the three lists ks , ℓs and gs have the same size p and that $\forall i \in \{1, \dots, p-1\}$, $ks_i + \text{cod}(gs_i) + \ell s_i = ks_{i+1} + \text{dom}(gs_{i+1}) + \ell s_{i+1}$, to ensure that the layers have appropriate domains and codomains.

Notice that, according to the coherence equations, two layers in sequence may commute when the outputs of the first generator are not connected to any inputs of the second generator. To obtain a *normal form*, we must fix an order for the commuting layers. To do so, we require the upper width of a generator to be smaller than the height of the next one, since the two layers can be commuted otherwise. This leads to the following definition of normal forms.

Definition 1 (normal form). *The set $\langle \Sigma \rangle^{\text{nf}}$ of terms in normal form is inductively defined as follows: id_n and $L_{\ell}^k(g)$ are in normal form ; and $\mathbf{L}_{\ell \# \ell s}^{ks}(g \# gs)$ is in normal form if $\mathbf{L}_{\ell s}^{ks}(gs)$ is in normal form while satisfying $k < ks_1 + \text{dom}(gs_1)$.*

Example 2. For instance, if $\Sigma = \{A : 1 \rightarrow 1, B : 2 \rightarrow 2\}$, the term $t = ((A \circledast \text{id}_1) \otimes ((A \otimes A) \circledast B)) \circledast (B \otimes A) \in \langle \Sigma \rangle$ is equivalent to $L_{[2,1,0,0,1,0]}^{[0,1,2,1,0,2]}([A, A, A, B, B, A])$ which is in normal form. These terms can be depicted as follows.



We introduce a procedure to turn any term into its normal form, relying on two new operators $\bullet$ and $\star$ allowing to compute the resulting normal form of a sequential and a parallel composition of two normal forms respectively. While the definition of $\star$ is straightforward, the definition of $\bullet$ requires mixing the layers of the two normal forms in order to satisfy the conditions explained in Definition 1.

Definition 3 (normalizing map). Let $\text{NF} : \langle \Sigma \rangle \rightarrow \langle \Sigma \rangle^{\text{nf}}$ be the normalizing map that assigns to each term $t : n \rightarrow m$ its normal form $\text{NF}(t) : n \rightarrow m$, inductively defined as

$$\begin{aligned} \text{NF}(\text{id}_n) &:= \text{id}_n & \text{NF}(g) &:= L_0^0(g) \\ \text{NF}(u \circledast v) &:= \text{NF}(u) \bullet \text{NF}(v) & \text{NF}(u \otimes v) &:= \text{NF}(u) \star \text{NF}(v) \end{aligned}$$

where $\bullet$ and $\star$ are inductively defined as follows, where $c = \text{cod}(L_{\ell s}^{ks}(gs))$ and $d = \text{dom}(L_{ss}^{rs}(hs))$.

$$\begin{aligned} \text{id}_n \bullet \text{id}_n &:= \text{id}_n \\ \text{id}_n \bullet L_{\ell s}^{ks}(gs) &:= L_{\ell s}^{ks}(gs) \\ L_{\ell s}^{ks}(gs) \bullet \text{id}_n &:= L_{\ell s}^{ks}(gs) \\ L_{\ell'}^{k'}(g') \bullet L_{\ell \# \ell s}^{k \# ks}(g \# gs) &:= \begin{cases} L_{\ell' \# \ell \# \ell s}^{k' \# k \# ks}(g' \# g \# gs) & \text{if } k' < k + \text{dom}(g) \\ L_{\ell + \text{def}(g')}^k(g) \circledast L_{\ell'}^{k' - \text{def}(g)}(g') & \text{else if } ks = \ell s = gs = [] \\ L_{\ell + \text{def}(g')}^k(g) \circledast (L_{\ell'}^{k' - \text{def}(g)}(g') \bullet L_{\ell s}^{ks}(gs)) & \text{otherwise} \end{cases} \\ L_{\ell \# \ell s}^{k \# ks}(g \# gs) \bullet L_{ss}^{rs}(hs) &:= L_{\ell}^k(g) \bullet (L_{\ell s}^{ks}(gs) \bullet L_{ss}^{rs}(hs)) \\ \text{id}_n \star \text{id}_m &:= \text{id}_{n+m} \\ \text{id}_n \star L_{\ell s}^{ks}(gs) &:= L_{\ell s}^{ks+n}(gs) \\ L_{\ell s}^{ks}(gs) \star \text{id}_n &:= L_{\ell s+n}^{ks}(gs) \\ L_{\ell s}^{ks}(gs) \star L_{ss}^{rs}(hs) &:= L_{(\ell s+d) @ ss}^{ks @ (rs+c)}(gs @ hs) \end{aligned}$$

Graphically, the notion of diagrammatic equivalence in PROs is known as *planar isotopy* [21, Theorem 3.1], and, roughly speaking, means that boxes can be moved horizontally without allowing wires to cross each other. To avoid graph-theoretic technicalities and keep the focus on term rewriting, we work under the assumption that the normalizing map NF yields exactly the notion of diagrammatic equivalence in PROs without showing that it relates to planar isotopy.

As stated in the following lemma, the normalizing map's definition ensures that all normal forms are fix points, meaning that normalizing a normal form does not change the normal form.

Lemma 4. $\text{NF}(t) = t$ for any $t \in \langle \Sigma \rangle^{\text{nf}}$.

Proof. The proof is by induction on the number of layers of t . The statement is true when $t = \text{id}_n$ and $t = L_\ell^k(g)$. Assume $\mathbf{L}_{\ell\# \ell s}^{k\# ks}(g\# gs)$ is in normal form, so in particular $k < ks_1 + \text{dom}(gs_1)$, which implies the following.

$$\begin{aligned} \text{NF}(\mathbf{L}_{\ell\# \ell s}^{k\# ks}(g\# gs)) &= \text{NF}(L_\ell^k(g)) \bullet \text{NF}(\mathbf{L}_{\ell s}^{ks}(gs)) \\ &\stackrel{\text{IH}}{=} L_\ell^k(g) \bullet \mathbf{L}_{\ell s}^{ks}(gs) = \mathbf{L}_{\ell\# \ell s}^{k\# ks}(g\# gs) \end{aligned}$$

This proof has been verified with Isabelle/HOL. $\square$

In order to simplify the proofs on the TRS and make the correspondence between generators and layers clearer, we apply a preprocessing step to terms that replaces each generator g by its corresponding layer $L_0^0(g)$. The set of pre-processed terms is denoted $\langle \Sigma \rangle^{\text{pp}}$ and satisfies $\langle \Sigma \rangle^{\text{nf}} \subset \langle \Sigma \rangle^{\text{pp}} \subset \langle \Sigma \rangle$.

3.2 Terminating and Confluent Rewriting System

Let $\mathcal{R}$ be the TRS containing the rewrite rules (R1) to (R11) defined in Figure 2. Rewriting can be applied to sub-terms, meaning that $K[L] \rightarrow_{\mathcal{R}} K[R]$ is a valid rewrite for any rewrite rule $(L \rightarrow R) \in \mathcal{R}$ and any $K[\cdot] \in \langle \Sigma \rangle_{\text{dom}(L), \text{cod}(L)}^{\text{pp}}$. Moreover, some rewrite rules ((R7), (R10) and (R11)) are subject to additional applicability conditions. All such conditions are decidable and stable under rewriting, ensuring that the normalization procedure is effective and suitable for automation. As stated in the following proposition, the rewriting system $\mathcal{R}$ is sound with respect to the coherence equations.

Proposition 5. $t_1 \rightarrow_{\mathcal{R}}^* t_2 \implies \mathbf{PRO}_{\Sigma} \vdash t_1 = t_2$ for any $t_1, t_2 \in \langle \Sigma \rangle^{\text{pp}}$.

Proof. For any rewrite rule $(L \rightarrow R) \in \mathcal{R}$ and any $K[\cdot] \in \langle \Sigma \rangle_{\text{dom}(L), \text{cod}(L)}^{\text{pp}}$, we prove $\mathbf{PRO}_{\Sigma} \vdash K[L] = K[R]$ by induction on $K[\cdot]$ where the only non-trivial case is $K[\cdot] = [\cdot]$. For most rewrite rules this is trivial, the only interesting cases (R10) and (R11) are proved using, in particular, Equation (6). $\square$

In the following, we prove that preprocessed terms are reduced into their normal form by applying $\mathcal{R}$. To do so, we show that $\mathcal{R}$ terminates and is confluent. We initially attempted to rely on automated termination and confluence provers such as AProVE [15] and the provers of the CoCo competition [17]. While these tools were useful for guiding early refinements of the rules, they ultimately did not succeed in producing termination or confluence certificates for our system (at least used from their web interface by our non-expert hands). We therefore carried out the proofs manually and formalize the most important steps with the proof assistant Isabelle/HOL. To keep the formalization light weight, we chose not to use the term rewriting infrastructure present in Isabelle/HOL, concentrating instead our formalizing efforts on the equalities and inequalities at the heart of the proofs.

$$\begin{array}{ll}
(t \circledast u) \circledast v \longrightarrow t \circledast (u \circledast v) & \text{(R1)} \quad (t \otimes u) \otimes v \longrightarrow t \otimes (u \otimes v) & \text{(R2)} \\
\text{id}_n \circledast t \longrightarrow t & \text{(R3)} \quad t \circledast \text{id}_n \longrightarrow t & \text{(R4)} \\
\text{id}_n \otimes \text{id}_m \longrightarrow \text{id}_{n+m} & \text{(R5)} \quad \text{id}_n \otimes (\text{id}_m \otimes t) \longrightarrow \text{id}_{n+m} \otimes t & \text{(R6)} \\
u \otimes v \longrightarrow (u \otimes \text{id}_{\text{dom}(v)}) \circledast (\text{id}_{\text{cod}(u)} \otimes v) & \text{if } u \neq \text{id}_n \wedge v \neq \text{id}_m & \text{(R7)} \\
\text{id}_n \otimes (u \circledast v) \longrightarrow (\text{id}_n \otimes u) \circledast (\text{id}_n \otimes v) & \text{(R8)} \\
(u \circledast v) \otimes \text{id}_n \longrightarrow (u \otimes \text{id}_n) \circledast (v \otimes \text{id}_n) & \text{(R9)} \\
L_{\ell_1}^{k_1}(g_1) \circledast L_{\ell_2}^{k_2}(g_2) \longrightarrow L_{\ell_2 + \text{def}(g_1)}^{k_2}(g_2) \circledast L_{\ell_1}^{k_1 - \text{def}(g_2)}(g_1) & \text{(R10)} \\
\text{if } k_1 \geq k_2 + \text{dom}(g_2) & \\
L_{\ell_1}^{k_1}(g_1) \circledast (L_{\ell_2}^{k_2}(g_2) \circledast t) \longrightarrow L_{\ell_2 + \text{def}(g_1)}^{k_2}(g_2) \circledast (L_{\ell_1}^{k_1 - \text{def}(g_2)}(g_1) \circledast t) & \text{(R11)} \\
\text{if } k_1 \geq k_2 + \text{dom}(g_2) &
\end{array}$$

Fig. 2: The rewrite rules are defined for any $n, m, k_1, k_2, \ell_1, \ell_2 \in \mathbb{N}$, $g_1, g_2 \in \Sigma$, $t, u, v \in \langle \Sigma \rangle$ and (R7), (R10) and (R11) are conditioned.

Proposition 6. $\mathcal{R}$ terminates.

Proof. We show that any rewrite sequence $t_1 \longrightarrow_{\mathcal{R}} t_2 \longrightarrow_{\mathcal{R}} t_3 \longrightarrow_{\mathcal{R}} \dots$ is finite. To do so, we define two positive quantities $\beta : \langle \Sigma \rangle \rightarrow \mathbb{N}$ and $\delta_D : \langle \Sigma \rangle \rightarrow \mathbb{R}^+$ (see Figure 3), where the former is parameterized by $D \in \mathbb{N}$, and prove that (β, δ_D) strictly decreases by lexicographical order, for a carefully chosen $D \in \mathbb{N}$. Notice that as δ_D takes real positive value, we must also show that it decreases by at least a constant at each step, otherwise the value could decrease infinitely. Given a term $t \in \langle \Sigma \rangle$, let $D(t) := 2 + \max_{g \in \text{gen}(t)} |\text{def}(g)|$. Notice that $D(\cdot)$ is preserved under rewriting, meaning that $D(t_i) = D(t_j)$ for any i, j . We prove that

$$\beta(K[L]) > \beta(K[R]) \quad \text{or} \quad \beta(K[L]) = \beta(K[R]) \wedge \delta_D(K[L]) \geq \delta_D(K[R]) + 1/D$$

for any rewrite rule $(L \longrightarrow R) \in \mathcal{R}$ and any context $K[\cdot] \in \langle \Sigma \rangle_{\text{dom}(L), \text{cod}(L)}^{\text{pp}}$, where we chose $D := D(K[L]) = D(K[R])$. This is straightforwardly proved by induction on $K[\cdot]$ where the base cases $K[\cdot] = [\cdot]$ have been verified with Isabelle/HOL. More precisely, we prove that (R1) to (R9) strictly decrease β while (R10) and (R11) preserve β and strictly decrease δ_D . $\square$

Remark 7. The fact that (R10) and (R11) strictly decrease δ_D is not trivial, and is only true with a careful choice of $D \in \mathbb{N}$. More precisely, $D := D(K[L]) = D(K[R])$ ensures that $D \geq 2 - \text{def}(g_2)$, which implies $1 - 1/D \geq 1/D(1 - \text{def}(g_2))$. Moreover, the condition $k_1 \geq k_2 + \text{dom}(g_2)$ with $\text{dom}(g_2) \geq 1$ yields the following.

$$\begin{aligned}
k_1 \geq k_2 + 1 &\implies (1 - 1/D)k_1 \geq (1 - 1/D)(k_2 + 1) \\
&\implies k_1 + 1/Dk_2 \geq k_2 + 1/Dk_1 + (1 - 1/D) \geq k_2 + 1/Dk_1 + 1/D(1 - \text{def}(g_2)) \\
&\implies k_1 + 1/Dk_2 \geq k_2 + 1/D(k_1 - \text{def}(g_2)) + 1/D \implies \delta_D(L) \geq \delta_D(R) + 1/D
\end{aligned}$$

	$\alpha : \langle \Sigma \rangle \rightarrow \mathbb{N}$	$\beta : \langle \Sigma \rangle \rightarrow \mathbb{N}$	$\gamma : \langle \Sigma \rangle \rightarrow \mathbb{N}$	$\delta_D : \langle \Sigma \rangle \rightarrow \mathbb{R}^+$
id_n	0	2	n	0
g	$nm^2 + 1$ if $g = \sigma_{n,m}$ 0 otherwise	5	0	0
$u \circ v$	$\alpha(u) + \alpha(v)$	$2\beta(u) + \beta(v) + 1$	$\gamma(u) + \gamma(v)$	$\delta_D(u) + \frac{1}{D}\delta_D(v)$
$u \otimes v$	$\alpha(u) + \alpha(v)$	$\beta(u)^2\beta(v)$	$\gamma(u) + 2\gamma(v)$	$n + \delta_D(v)$ if $u = \text{id}_n$ $\delta_D(u) + \delta_D(v)$ otherwise
$L_\ell^k(g)$	d if $g = \tau_d$ $\alpha(g)$ otherwise	200	$k + 4\ell$	k

Rewrite rules	α	β	γ	δ_D
(R1), (R2), (R3), (R4), (R5), (R6), (R7), (R8) and (R9)	=	>		
(R10) and (R11)		=		>
(R12), (R13) and (R14)	>			
(R15) and (R16)	=	=	>	
(R17) and (R18)	>			
(R19) and (R20)	>			
(R21) and (R22)	=	=	=	>

Fig. 3: The top table defines the positive quantities $\alpha, \beta, \gamma, \delta_D$, where the former is parameterized by $D \in \mathbb{N}$. These quantities are used to prove the termination of $\mathcal{R}$ and $\mathcal{R}_0$ in Propositions 6 and 19 respectively. The last line of the table gives the value of each quantity applied to layers. The bottom table describes which rewrite rules strictly decrease ($>$) or preserve ($=$) each quantity. All cases so indicated have been verified with Isabelle/HOL.

Lemma 8. $t_1 \rightarrow_{\mathcal{R}}^* t_2 \implies \text{NF}(t_1) = \text{NF}(t_2)$ for any $t_1, t_2 \in \langle \Sigma \rangle^{\text{pp}}$.

Proof. For any rewrite rule $(L \rightarrow R) \in \mathcal{R}$ and any $K[\cdot] \in \langle \Sigma \rangle_{\text{dom}(L), \text{cod}(L)}^{\text{pp}}$, the equality $\text{NF}(K[L]) = \text{NF}(K[R])$ is straightforwardly proved by induction on $K[\cdot]$ where the base cases $K[\cdot] = [\cdot]$ have been verified with Isabelle/HOL. $\square$

Lemma 9. $t \not\rightarrow_{\mathcal{R}} \implies t \in \langle \Sigma \rangle^{\text{nf}}$ for any $t \in \langle \Sigma \rangle^{\text{pp}}$.

Proof. By induction on t . The base cases $t = \text{id}_n$ for $n \in \mathbb{N}$ are direct because $\text{id}_n \not\rightarrow_{\mathcal{R}}$ and $\text{id}_n \in \langle \Sigma \rangle^{\text{nf}}$. The base cases $t \in \Sigma$ are trivial because $t \notin \langle \Sigma \rangle^{\text{pp}}$.

- Suppose $t = u \circ v$ and $t \not\rightarrow_{\mathcal{R}}$. Then $u, v \in \langle \Sigma \rangle^{\text{pp}}$ and $u \not\rightarrow_{\mathcal{R}}, v \not\rightarrow_{\mathcal{R}}$ which, by IH, implies $u, v \in \langle \Sigma \rangle^{\text{nf}}$. Notice that $u \neq \text{id}_n \neq v$ otherwise (R3) or (R4) applies. So, it must be the case that $u = L_\ell^k(g)$ and $v = L_{\ell s}^{ks}(gs)$ for some $k, \ell, p \in \mathbb{N}$, $g \in \Sigma$, $ks, \ell s \in \mathbb{N}^p$, $gs \in \Sigma^p$ such that $p > 0$, otherwise (R1) applies. Moreover, $k < ks_1 + \text{dom}(gs_1)$, otherwise (R10) or (R11) applies. Hence, $t = L_{\ell \# \ell s}^{k \# ks}(g \# gs) \in \langle \Sigma \rangle^{\text{nf}}$.

- Suppose $t = u \otimes v$ and $t \not\rightarrow_{\mathcal{R}}$. Then $u \not\rightarrow_{\mathcal{R}}, v \not\rightarrow_{\mathcal{R}}$. Notice that u and v are not necessarily in $\langle \Sigma \rangle^{\text{pp}}$, so we cannot use the IH. Nevertheless, we know that $u, v \notin \Sigma$, otherwise $t \notin \langle \Sigma \rangle^{\text{pp}}$. In fact, we know for sure that $u = \text{id}_k$ and $v = v_1 \otimes v_2$ for some $k \in \mathbb{N}$ and $v_1, v_2 \in \langle \Sigma \rangle$, otherwise a rule among (R2), (R5), (R7), (R8) and (R9) applies. For the same reason and the fact that (R6) is not applicable, it must be the case that $v_1 = g$ and $v_2 = \text{id}_\ell$ for some $g \in \Sigma$ and $\ell \in \mathbb{N}$. Hence, $t = \text{id}_k \otimes (g \otimes \text{id}_\ell) = L_\ell^k(g) \in \langle \Sigma \rangle^{\text{nf}}$. $\square$

Proposition 10. $\mathcal{R}$ is confluent in $\langle \Sigma \rangle^{\text{pp}}$.

Proof. Let $t \in \langle \Sigma \rangle^{\text{pp}}$. If $t \rightarrow_{\mathcal{R}}^* u$ and $t \rightarrow_{\mathcal{R}}^* v$, then by Proposition 6 there exist $u', v' \in \langle \Sigma \rangle^{\text{pp}}$ such that $u \rightarrow_{\mathcal{R}}^* u' \not\rightarrow_{\mathcal{R}}$ and $v \rightarrow_{\mathcal{R}}^* v' \not\rightarrow_{\mathcal{R}}$. Lemmas 4, 8 and 9 imply $\text{NF}(t) = u' = v'$, which concludes the proof. $\square$

As stated in the following theorem, the termination and confluence of $\mathcal{R}$ solve our problem: given two diagrammatically equivalence terms $t_1, t_2 \in \langle \Sigma \rangle^{\text{pp}}$, we can effectively rewrite t_1 into t_2 using the coherence equations.

Theorem 11. $\text{NF}(t_1) = \text{NF}(t_2) \implies t_1 \longleftrightarrow_{\mathcal{R}}^* t_2$ for any $t_1, t_2 \in \langle \Sigma \rangle^{\text{pp}}$.

Proof. Proposition 6 and Lemmas 4, 8 and 9 imply that $t_i \rightarrow_{\mathcal{R}}^* \text{NF}(t_i)$, and the assumption $\text{NF}(t_1) = \text{NF}(t_2)$ yields $t_1 \longleftrightarrow_{\mathcal{R}}^* t_2$. $\square$

While the rewrite rules can be applied in any order, it may be more efficient to follow a predefined rewriting strategy. In particular, (R1) to (R9) allow transforming any preprocessed term into a pre-normal form, and (R10) and (R11) allow transforming the layers to fulfill the normal form conditions. Hence, the two steps can be applied one after the other.

Notice that the normal form map NF is useful on its own. Indeed, checking diagrammatic equivalence between two terms $t_1, t_2 \in \langle \Sigma \rangle^{\text{pp}}$ can be directly achieved by checking $\text{NF}(t_1) = \text{NF}(t_2)$. Nevertheless, as we plan to integrate our TRSs in an overall rewriting-based certification pipeline, the TRS is more interesting to us than the normal form itself.

4 Term Rewriting in the PRO of Permutations

In this section, we consider the PRO of permutations $\mathbf{PRO}_{\Sigma_0, \mathcal{E}_0}$, whose generators are the swaps, i.e. $\Sigma_0 := \{\sigma_{n,m} : n, m \in \mathbb{N}\}$ and where $\mathcal{E}_0$ is the set containing Equations (7) to (10). $\mathbf{PRO}_{\Sigma_0, \mathcal{E}_0}$ is the simplest PROP that one can think about, namely $\mathbf{PROP}_\emptyset$. Intuitively, in this structure, the diagrams have no boxes and are just made of wires, which allows expressing any permutation. Notice that all terms in $\langle \Sigma_0 \rangle$ have an equal number of inputs and outputs. We solve the underlying diagrammatic equivalence problem: given two terms $t_1, t_2 \in \langle \Sigma_0 \rangle$ expressing the same permutation, how to effectively rewrite t_1 into t_2 using the coherence equations (Equations (1) to (10))? To do so, we refine the normal form of Definition 1 to define a normal form for the PRO of permutations—that we call *canonical form* to prevent confusion—and provide an associated rewriting system that we prove to be terminating and confluent.

4.1 Canonical Form and Interpretation

The permutations of size $n \in \mathbb{N}$, whose set is denoted $\mathbf{Perm}(n)$, are the bijective maps from $\{1, \dots, n\}$ to itself. A permutation $\pi \in \mathbf{Perm}(n)$ is denoted $(\pi(1), \dots, \pi(n))$. For instance, $(2, 1, 3) \in \mathbf{Perm}(3)$ is the permutation mapping 1 to 2, 2 to 1 and 3 to itself. There is only one permutation of size 0, denoted $()$, and only one permutation of size 1, denoted (1) . In the following, $|\pi|$ denotes the size of the permutation π and π^{-1} its inverse. Two permutations $\pi_1, \pi_2 \in \mathbf{Perm}(n)$ can be composed sequentially with $\circ$ as the usual composition, i.e. $(\pi_2 \circ \pi_1)(i) := \pi_2(\pi_1(i))$ for any $i \in \{1, \dots, n\}$. Two permutations $\pi_1 \in \mathbf{Perm}(n), \pi_2 \in \mathbf{Perm}(m)$ can be composed in parallel with the direct sum $\oplus$ defined as follows for any $i \in \{1, \dots, n+m\}$.

$$(\pi_1 \oplus \pi_2)(i) := \begin{cases} \pi_1(i) & \text{if } 1 \leq i \leq n \\ n + \pi_2(i - n) & \text{if } n < i \leq n + m \end{cases}$$

Let $\mathbf{Perm} := \cup_{n \in \mathbb{N}} \mathbf{Perm}(n)$ be the collection of all permutations. Then, each term $t : n \rightarrow n \in \langle \Sigma_0 \rangle$ can be interpreted as a permutation of size n .

Definition 12 (interpretation). Let $\llbracket \cdot \rrbracket : \langle \Sigma_0 \rangle \rightarrow \mathbf{Perm}$ be the interpretation, inductively defined as follows.

$$\begin{aligned} \llbracket \text{id}_n \rrbracket &:= (1, \dots, n) & \llbracket \sigma_{n,m} \rrbracket &:= (n+1, \dots, n+m, 1, \dots, n) \\ \llbracket u \circ v \rrbracket &:= \llbracket v \rrbracket \circ \llbracket u \rrbracket & \llbracket u \otimes v \rrbracket &:= \llbracket u \rrbracket \oplus \llbracket v \rrbracket \end{aligned}$$

Given $d \in \mathbb{N}$, let $\tau_d := \sigma_{d,1}$ be the *toboggan* of size d . Intuitively, the diagrammatic representation of τ_d lifts its last wire to the top, and its interpretation is $\llbracket \tau_d \rrbracket = (2, \dots, d+1, 1)$, which implies the following equality.

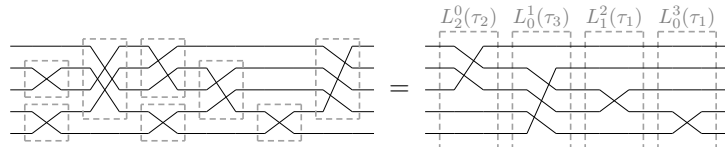
$$\llbracket L_\ell^k(\tau_d) \rrbracket = (1, \dots, k, k+2, \dots, k+d+1, k+1, k+d+2, \dots, k+d+1+\ell)$$

We now define the *canonical form* and its associated canonizing map. Given $ks, \ell s, ds \in \mathbb{N}^p$ with $p > 0$, we introduce the following notation.

$$\mathbf{C}_{\ell s}^{ks}(ds) := \mathbf{L}_{\ell s}^{ks}([\tau_{ds_i}]_{i \in \{1, \dots, p\}})$$

Definition 13 (canonical form). The set $\langle \Sigma_0 \rangle^{\text{cf}}$ of terms in canonical form is inductively defined as follows: id_n is in canonical form ; $L_\ell^k(\tau_d)$ is in canonical form if $d \geq 1$; and $\mathbf{C}_{\ell \# \ell s}^{k \# ks}(d \# ds)$ is in canonical form if $d \geq 1$ and $\mathbf{C}_{\ell s}^{ks}(ds)$ is in canonical form while satisfying $k < ks_1$.

Example 14. The term $(\text{id}_1 \otimes \tau_1 \otimes \tau_1) \circ (\sigma_{2,2} \otimes \text{id}_1) \circ (\sigma_{1,2} \otimes \tau_1) \circ (\text{id}_1 \otimes \sigma_{1,2} \otimes \text{id}_1) \circ (\text{id}_3 \otimes \tau_1) \circ (\tau_3 \otimes \text{id}_1)$ is equivalent to $\mathbf{C}_{[2,0,1,0]}^{[0,1,2,3]}([2,3,1,1])$, which is in canonical form. These terms can be graphically depicted as follows.



Definition 15 (canonizing map). Let $\text{CF} : \mathbf{Perm} \rightarrow \langle \Sigma_0 \rangle^{\text{cf}}$ be the canonizing map that assigns to each permutation $\pi \in \mathbf{Perm}(n)$ its term in canonical form $\text{CF}(\pi) : n \rightarrow n$, inductively defined as

$$\begin{aligned} \text{CF}(()) &:= \text{id}_0 & \text{CF}((1)) &:= \text{id}_1 \\ \text{CF}(\pi) &:= \begin{cases} \text{id}_1 \star \text{CF}(\pi|_1) & \text{if } \pi(1) = 1 \\ L_{|\pi|-\pi^{-1}(1)}^0(\tau_{\pi^{-1}(1)-1}) \bullet (\text{id}_1 \star \text{CF}(\pi|_1)) & \text{otherwise} \end{cases} \end{aligned}$$

where $\pi|_1 \in \mathbf{Perm}(|\pi|-1)$ is the permutation satisfying $(\pi|_1)(i) := \pi(i)-1$ when $1 \leq i < \pi^{-1}(1)$ and $(\pi|_1)(i) := \pi(i+1)-1$ when $\pi^{-1}(1) \leq i \leq |\pi|-1$.

Notice that $\langle \Sigma_0 \rangle^{\text{cf}} \subset \langle \Sigma_0 \rangle^{\text{nf}} \subset \langle \Sigma_0 \rangle^{\text{pp}} \subset \langle \Sigma_0 \rangle$. Moreover, as stated in the following lemma, the canonizing map's definition is compatible with $\llbracket \cdot \rrbracket$.

Lemma 16. $\text{CF}(\llbracket t \rrbracket) = t$ for any $t \in \langle \Sigma_0 \rangle^{\text{cf}}$.

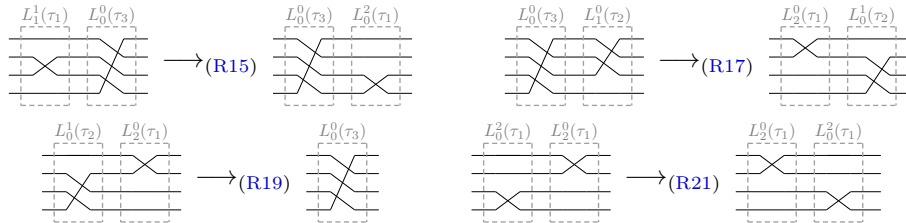
Proof. In the following we use the notation $\pi^{\oplus n} := \pi \oplus \dots \oplus \pi$ (with n compositions $\oplus$). If $t = \text{id}_n$, then $\text{CF}(\llbracket \text{id}_n \rrbracket) = \text{CF}((1)^{\oplus n}) = \text{id}_n$. Otherwise, we proceed by induction on the number of layers. If $t = L_\ell^k(\tau_d)$, then $\text{CF}(\llbracket L_\ell^k(\tau_d) \rrbracket) = \text{id}_k \star \text{CF}(\llbracket L_\ell^0(\tau_d) \rrbracket) = \text{id}_k \star (L_\ell^0(\tau_d) \bullet (\text{id}_1 \star \text{CF}((1)^{\oplus \ell+d}))) = L_\ell^k(\tau_d)$. Moreover, if $t = C_{\ell\# \ell s}^{k\# ks}(d\# ds)$, then we have the following proof.

$$\begin{aligned} \text{CF}(\llbracket C(k\# ks, \ell\# \ell s, d\# ds) \rrbracket) &= \text{id}_k \star \text{CF}(\llbracket C(0\#(ks-k), \ell\# \ell s, d\# ds) \rrbracket) \\ &= \text{id}_k \star (L_{\ell+d-d}^0(\tau_d) \bullet (\text{id}_1 \star \text{CF}(\llbracket C(ks-k-1, \ell s, ds) \rrbracket))) \\ &\stackrel{\text{IH}}{=} \text{id}_k \star (L_\ell^0(\tau_d) \bullet (\text{id}_1 \star C(ks-k-1, \ell s, ds))) = C(k\# ks, \ell\# \ell s, d\# ds) \quad \square \end{aligned}$$

4.2 Terminating and Confluent Rewriting System

We introduce the new rewrite rules in Figure 4. The rules (R1) to (R9) (see Figure 2) together with the rules (R12) to (R22) form the rewriting system $\mathcal{R}_0$. Notice that $\mathcal{R}_0$ contains all rules of $\mathcal{R}$ except (R10) and (R11), which have been replaced by the rules (R15) to (R22).

Remark 17. (R15), (R17), (R19) and (R21) explain how to transform a sequence of two layers $L_{\ell_1}^{k_1}(\tau_{d_1})$ and $L_{\ell_2}^{k_2}(\tau_{d_2})$ whenever $k_1 \geq k_2$, and (R16), (R18), (R20) and (R22) apply the same transformations when more layers follow. Here are some graphical examples.



As stated in the following proposition, the rewriting system $\mathcal{R}_0$ is sound with respect to the coherence equations.

$$\begin{aligned}
& \sigma_{0,n} \longrightarrow \text{id}_n \quad (\text{R12}) & \sigma_{n,0} \longrightarrow \text{id}_n \quad (\text{R13}) \\
& \sigma_{n+1,m+2} \longrightarrow (\sigma_{n+1,m+1} \otimes \text{id}_1) \circ (\text{id}_{m+1} \otimes \tau_{n+1}) \quad (\text{R14}) \\
& L_{\ell_1}^{k_1}(\tau_{d_1}) \circ L_{\ell_2}^{k_2}(\tau_{d_2}) \longrightarrow L_{\ell_2}^{k_2}(\tau_{d_2}) \circ L_{\ell_1-1}^{k_1+1}(\tau_{d_1}) \quad (\text{R15}) \\
& \text{if } d_1 \geq 1, d_2 \geq 1 \text{ and } k_2 \leq k_1 < k_2 + d_2 - d_1 \\
& L_{\ell_1}^{k_1}(\tau_{d_1}) \circ (L_{\ell_2}^{k_2}(\tau_{d_2}) \circ t) \longrightarrow L_{\ell_2}^{k_2}(\tau_{d_2}) \circ (L_{\ell_1-1}^{k_1+1}(\tau_{d_1}) \circ t) \quad (\text{R16}) \\
& \text{if } d_1 \geq 1, d_2 \geq 1 \text{ and } k_2 \leq k_1 < k_2 + d_2 - d_1 \\
& L_{\ell_1}^{k_1}(\tau_{d_1}) \circ L_{\ell_2}^{k_2}(\tau_{d_2}) \longrightarrow L_{\ell_2+1}^{k_2}(\tau_{d_2-1}) \circ L_{\ell_1-1}^{k_1+1}(\tau_{d_1-1}) \quad (\text{R17}) \\
& \text{if } d_1 \geq 1, d_2 \geq 1 \text{ and } k_2 \leq k_1 \wedge k_2 + d_2 - d_1 \leq k_1 < k_2 + d_2 \\
& L_{\ell_1}^{k_1}(\tau_{d_1}) \circ (L_{\ell_2}^{k_2}(\tau_{d_2}) \circ t) \longrightarrow L_{\ell_2+1}^{k_2}(\tau_{d_2-1}) \circ (L_{\ell_1-1}^{k_1+1}(\tau_{d_1-1}) \circ t) \quad (\text{R18}) \\
& \text{if } d_1 \geq 1, d_2 \geq 1 \text{ and } k_2 \leq k_1 \wedge k_2 + d_2 - d_1 \leq k_1 < k_2 + d_2 \\
& L_{\ell_1}^{k_1}(\tau_{d_1}) \circ L_{\ell_2}^{k_2}(\tau_{d_2}) \longrightarrow L_{\ell_1}^{k_2}(\tau_{d_1+d_2}) \quad (\text{R19}) \\
& \text{if } d_1 \geq 1, d_2 \geq 1 \text{ and } k_1 = k_2 + d_2 \\
& L_{\ell_1}^{k_1}(\tau_{d_1}) \circ (L_{\ell_2}^{k_2}(\tau_{d_2}) \circ t) \longrightarrow L_{\ell_1}^{k_2}(\tau_{d_1+d_2}) \circ t \quad (\text{R20}) \\
& \text{if } d_1 \geq 1, d_2 \geq 1 \text{ and } k_1 = k_2 + d_2 \\
& L_{\ell_1}^{k_1}(\tau_{d_1}) \circ L_{\ell_2}^{k_2}(\tau_{d_2}) \longrightarrow L_{\ell_2}^{k_2}(\tau_{d_2}) \circ L_{\ell_1}^{k_1}(\tau_{d_1}) \quad (\text{R21}) \\
& \text{if } d_1 \geq 1, d_2 \geq 1 \text{ and } k_1 > k_2 + d_2 \\
& L_{\ell_1}^{k_1}(\tau_{d_1}) \circ (L_{\ell_2}^{k_2}(\tau_{d_2}) \circ t) \longrightarrow L_{\ell_1}^{k_1}(\tau_{d_1}) \circ (L_{\ell_2}^{k_2}(\tau_{d_2}) \circ t) \quad (\text{R22}) \\
& \text{if } d_1 \geq 1, d_2 \geq 1 \text{ and } k_1 > k_2 + d_2
\end{aligned}$$

Fig. 4: The rewrite rules are defined for any $n, m, k_1, k_2, \ell_1, \ell_2, d_1, d_2 \in \mathbb{N}$ and (R15) to (R22) are conditioned.

Proposition 18. $t_1 \longrightarrow_{\mathcal{R}_0}^* t_2 \implies \mathbf{PRO}_{\Sigma_0, \varepsilon_0} \vdash t_1 = t_2$ for any $t_1, t_2 \in \langle \Sigma_0 \rangle^{\text{pp}}$.

Proof. For any rewrite rule $(L \longrightarrow R) \in \mathcal{R}_0$ and any $K[\cdot] \in \langle \Sigma_0 \rangle_{\text{dom}(L), \text{cod}(L)}^{\text{pp}}$, we prove $\mathbf{PRO}_{\Sigma_0, \varepsilon_0} \vdash K[L] = K[R]$ by induction on $K[\cdot]$ where the only non-trivial case is $K[\cdot] = [\cdot]$. For most rewrite rules this is trivial, the only interesting cases (R14) to (R22) are proved using, in particular, Equations (8) to (10). $\square$

In order to prove termination and confluence, we build upon the result of Section 3 and follow a similar proof flow.

Proposition 19. $\mathcal{R}_0$ terminates.

Proof. We build upon the proof of Proposition 6 by defining two additional positive quantities $\alpha, \gamma : \langle \Sigma_0 \rangle \rightarrow \mathbb{N}$ and show that $\mathcal{R}_0$ strictly decreases $(\alpha, \beta, \gamma, \delta_2)$ with lexicographical order (notice that, as $\text{def}(g) = 0$ for any $g \in \Sigma_0$, we can take $D = 2$ for δ_D). The definitions of α, β, γ and δ_2 are given in Figure 3 together with a sum up of which rewrite rules strictly decrease or preserve each quantity. Again, all the base cases have been verified with Isabelle/HOL. $\square$

Lemma 20. $t_1 \longrightarrow_{\mathcal{R}_0}^* t_2 \implies \llbracket t_1 \rrbracket = \llbracket t_2 \rrbracket$ for any $t_1, t_2 \in \langle \Sigma_0 \rangle^{\text{PP}}$

Proof. For any rewrite rule $(L \longrightarrow R) \in \mathcal{R}_0$ and any $K[\cdot] \in \langle \Sigma_0 \rangle_{\text{dom}(L), \text{cod}(L)}^{\text{PP}}$, the equality $\llbracket K[L] \rrbracket = \llbracket K[R] \rrbracket$ is straightforwardly proved by induction on $K[\cdot]$ where the base cases $K[\cdot] = [\cdot]$ have been verified with Isabelle/HOL. $\square$

Lemma 21. $t \not\rightarrow_{\mathcal{R}_0} \implies t \in \langle \Sigma_0 \rangle^{\text{cf}}$ for any $t \in \langle \Sigma_0 \rangle^{\text{PP}}$.

Proof. All rewrite rules of $\mathcal{R}$ are in $\mathcal{R}_0$ except (R10) and (R11), which exactly correspond to (R21) and (R22) in $\mathcal{R}_0$ when $g_i = \tau_{d_i}$ for some $d_i \geq 1$. Moreover, every generator $\sigma_{n,m}$ in t satisfies $n > 0$ and $m = 1$, otherwise a rule among (R12) to (R14) applies. Thus, Lemma 9 applies and we can assume that $t \in \langle \Sigma_0 \rangle^{\text{nf}}$. This implies that there exists $ks, \ell s, ds \in \mathbb{N}^p$ for some $p > 0$ satisfying $ds_i \geq 1$ for any $i \in \{1, \dots, p\}$, such that $t = \mathbf{C}_{\ell s}^{ks}(ds)$. But then $\forall i \in \{1, \dots, p-1\} : ks_i < ks_{i+1}$, otherwise a rule among (R15) to (R20) applies. Hence, $t \in \langle \Sigma_0 \rangle^{\text{cf}}$. $\square$

Proposition 22. $\mathcal{R}_0$ is confluent in $\langle \Sigma_0 \rangle^{\text{PP}}$.

Proof. Let $t \in \langle \Sigma_0 \rangle^{\text{PP}}$. If $t \longrightarrow_{\mathcal{R}_0}^* u$ and $t \longrightarrow_{\mathcal{R}_0}^* v$, then by Proposition 19 there exist $u', v' \in \langle \Sigma_0 \rangle^{\text{PP}}$ such that $u \longrightarrow_{\mathcal{R}_0}^* u' \not\rightarrow_{\mathcal{R}_0}$ and $v \longrightarrow_{\mathcal{R}_0}^* v' \not\rightarrow_{\mathcal{R}_0}$. Lemmas 16, 20 and 21 imply $\text{CF}(\llbracket t \rrbracket) = u' = v'$, which concludes the proof. $\square$

As stated in the following theorem, the termination and confluence of $\mathcal{R}_0$ solve our problem: given two terms $t_1, t_2 \in \langle \Sigma_0 \rangle^{\text{PP}}$ expressing the same permutation, we can effectively rewrite t_1 into t_2 using the coherence equations.

Theorem 23. $\llbracket t_1 \rrbracket = \llbracket t_2 \rrbracket \implies t_1 \longleftarrow_{\mathcal{R}_0}^* t_2$ for any $t_1, t_2 \in \langle \Sigma_0 \rangle^{\text{PP}}$.

Proof. Proposition 19 and Lemmas 16, 20 and 21 implies that $t_i \longrightarrow_{\mathcal{R}_0}^* \text{CF}(\llbracket t_i \rrbracket)$, and the assumption $\llbracket t_1 \rrbracket = \llbracket t_2 \rrbracket$ yields $t_1 \longleftarrow_{\mathcal{R}_0}^* t_2$. $\square$

5 Conclusion

In this work, we addressed the problem of diagrammatic equivalence for two classes of diagrams, namely $\mathbf{PRO}_{\Sigma}$ (for an arbitrary state-and-effect-free set of generators Σ) and $\mathbf{PRO}_{\Sigma_0, \mathcal{E}_0} = \mathbf{PROP}_{\emptyset}$, which can express any permutation. We used a term rewriting approach, providing terminating and confluent TRSs. Our plan is to combine our solutions to design a rewriting-based approach to handle diagrammatic equivalences in an arbitrary $\mathbf{PROP}_{\Sigma}$, the main bottleneck being that, in $\mathbf{PROP}_{\Sigma}$, generators can slide through swaps (see Equation (9)), which breaks the normal forms we designed. Since our normal forms require the assumption of a state-and-effect-free PRO, it would also be interesting to generalize the first TRS to arbitrary PROs. Also, we could leverage the existing term-rewriting infrastructure present in Isabelle/HOL [24] to obtain a verified checker for our TRSs, or create proof-checkers that compute the normal/canonical forms in a verified way following Sternagel and Thiemann [23]. Our main objective—initiated by this first contribution—is to obtain a fully verified pipeline for certifying equivalences of quantum circuits.

Acknowledgements

We thank Florian Frohn, Carsten Fuhs, René Thiemann, and Akihisa Yamada for insightful discussions on TRSs. We also thank the anonymous reviewers for their comments. This work is supported by the Plan France 2030 through the PEPR integrated project EPiQ ANR-22-PETQ-0007 and the HQI platform ANR-22-PNCQ-0002; and by the European project MSCA Staff Exchanges Qcomical HORIZON-MSCA-2023-SE-01. The project is also supported by the Maison du Quantique MaQuEst.

References

1. Abramsky, S., Coecke, B.: A categorical semantics of quantum protocols. In: Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (LICS) (2004). <https://doi.org/10.1109/LICS.2004.1319636>, <https://arxiv.org/abs/quant-ph/0402130>
2. Baader, F., Nipkow, T.: Term rewriting and all that. Cambridge University Press (1998), <https://dl.acm.org/doi/10.5555/280474>
3. Backens, M., Kissinger, A.: Zh: A complete graphical calculus for quantum computations involving classical non-linearity. In: Proceedings of the 15th International Conference on Quantum Physics and Logic (QPL) (2018). <https://doi.org/10.4204/EPTCS.287.2>, <https://arxiv.org/abs/1805.02175>
4. Bonchi, F., Gadducci, F., Kissinger, A., Sobocinski, P., Zanasi, F.: String diagram rewrite theory ii: Rewriting with symmetric monoidal structure. *Mathematical Structures in Computer Science* (2022). <https://doi.org/10.1017/S0960129522000317>
5. Bonchi, F., Sobociński, P., Zanasi, F.: Interacting bialgebras are frobenius. In: Proceedings of the 17th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS) (2014). https://doi.org/10.1007/978-3-642-54830-7_23, <https://hal.science/hal-00989174>
6. Clavel, M., Durán, F., Hendrix, J., Lucas, S., Meseguer, J., Ölveczky, P.: The maude formal tool environment. In: Proceedings of the 2nd International Conference on Algebra and Coalgebra in Computer Science (CALCO) (2007). https://doi.org/10.1007/978-3-540-73859-6_12
7. Clément, A., Delorme, N., Perdrix, S.: Minimal equational theories for quantum circuits. In: Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS) (2024). <https://doi.org/10.1145/3661814.3662088>, <https://arxiv.org/abs/2311.07476>
8. Clément, A., Delorme, N., Perdrix, S., Vilmart, R.: Quantum circuit completeness: Extensions and simplifications. In: Proceedings of the 32nd EACSL Annual Conference on Computer Science Logic (CSL) (2024). <https://doi.org/10.4230/LIPIcs.CSL.2024.20>, <https://arxiv.org/abs/2303.03117>
9. Clément, A., Heurtel, N., Mansfield, S., Perdrix, S., Valiron, B.: A complete equational theory for quantum circuits. In: Proceedings of the 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS) (2023). <https://doi.org/10.1109/LICS56636.2023.10175801>, <https://arxiv.org/abs/2206.10577>
10. Cockett, R., Comfort, C.: The category tof. *Electronic Proceedings in Theoretical Computer Science* (2019). <https://doi.org/10.4204/EPTCS.287.4>

11. Coecke, B.: The mathematics of text structure. Joachim Lambek: The Interplay of Mathematics, Logic, and Linguistics (2021). https://doi.org/10.1007/978-3-030-66545-6_6, <https://arxiv.org/abs/1904.03478>
12. Coecke, B., Duncan, R.: Interacting quantum observables. In: Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP) (2008). https://doi.org/10.1007/978-3-540-70583-3_25, <https://arxiv.org/abs/0906.4725>
13. Delorme, N., Perdrix, S.: Diagrammatic reasoning with control as a constructor, applications to quantum circuits. In: Proceedings of the 29th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS) (2026). https://doi.org/10.1007/978-3-032-22730-0_12, <https://arxiv.org/abs/2508.21756>
14. Delorme, N., Tourret, S.: Isabelle/HOL theory files companions of the IJCAR 2026 paper "Towards Term-based Verification of Diagrammatic Equivalence" (May 2026). <https://doi.org/10.5281/zenodo.20294284>
15. Giesl, J., Schneider-Kamp, P., Thiemann, R.: Aprove 1.2: Automatic termination proofs in the dependency pair framework. In: Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR) (2006). https://doi.org/10.1007/11814771_24
16. Hadzihanovic, A.: A diagrammatic axiomatisation for qubit entanglement. In: Proceedings of the 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS) (2015). <https://doi.org/10.1109/LICS.2015.59>, <https://arxiv.org/abs/1501.07082>
17. Hirokawa, N., Nagele, J., Middeldorp, A.: Cops and cocoweb: Infrastructure for confluence tools. In: Proceedings of the 9th International Joint Conference on Automated Reasoning (IJCAR) (2018). https://doi.org/10.1007/978-3-319-94205-6_23
18. Jeandel, E., Perdrix, S., Vilmart, R.: A complete axiomatisation of the ZX-calculus for Clifford+T quantum mechanics. In: Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science LICS (2018). <https://doi.org/10.1145/3209108.3209131>, <https://arxiv.org/abs/1705.11151>
19. Lafont, Y.: Towards an algebraic theory of boolean circuits. Journal of Pure and Applied Algebra (2003). [https://doi.org/10.1016/S0022-4049\(03\)00069-0](https://doi.org/10.1016/S0022-4049(03)00069-0), <https://www.sciencedirect.com/science/article/pii/S0022404903000690>
20. Lehmann, A., Caldwell, B., Shah, B., Spencer, W., Rand, R.: Vyzx: Formal verification of a graphical quantum language. ACM Transactions on Programming Languages and Systems (2026). <https://doi.org/10.1145/3807780>, <https://arxiv.org/abs/2311.11571>
21. Selinger, P.: A survey of graphical languages for monoidal categories (2011). https://doi.org/10.1007/978-3-642-12821-9_4
22. Sobocinski, P.: Graphical linear algebra. Mathematical blog. (2015), <https://graphicallinearalgebra.net>
23. Sternagel, C., Thiemann, R.: A framework for developing stand-alone certifiers. Electronic Notes in Theoretical Computer Science (2015). <https://doi.org/10.1016/j.entcs.2015.04.004>, <https://www.sciencedirect.com/science/article/pii/S1571066115000080>
24. Thiemann, R., Sternagel, C.: Certification of termination proofs using ceta. In: Proceedings of the 22nd International Conference on Theorem Proving in Higher Order Logics (TPHOLs) (2009), <https://arxiv.org/abs/1505.01337>

25. Zeng, W., Coecke, B.: Quantum algorithms for compositional natural language processing. In: Proceedings of the Workshop on Semantic Spaces at the Intersection of NLP, Physics and Cognitive Science (SLPCS) (2016). <https://doi.org/10.4204/EPTCS.221.8>, <https://arxiv.org/abs/1608.01406>
26. Zhang, Y., Wang, Y.R., Flatt, O., Cao, D., Zucker, P., Rosenthal, E., Tatlöck, Z., Willsey, M.: Better together: Unifying datalog and equality saturation. In: Proceedings of the ACM on Programming Languages (PACMPL) (2023). <https://doi.org/10.1145/3591239>