Termination of Rewriting on Reversible Boolean Circuits as a Free 3-Category Problem

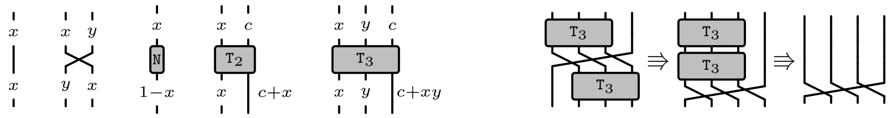
Adriano Barile, Stefano Berardi, Luca Roversi

Dipartimento di Informatica, Università di Torino

Abstract. Reversible Boolean Circuits are an interesting computational model under many aspects and in different fields, ranging from Reversible Computing to Quantum Computing. Our contribute is to describe a specific class of Reversible Boolean Circuits - which is as expressive as classical circuits - as a bi-dimensional diagrammatic programming language. We uniformly represent the Reversible Boolean Circuits we focus on as a free 3-category **Toff**. This formalism allows us to incorporate the representation of circuits and of rewriting rules on them, and to prove termination of rewriting. Termination follows from defining a non-identities-preserving functor from our free 3-category **Toff** into a suitable 3-category **Move** that traces the "moves" applied to wires inside circuits.

1 Introduction

The class of Reversible Boolean circuits (from now on, RBC) constitutes an interesting computational model, for many reasons. We name just some of them: once implemented, they may help to reduce electronic devices energy consumption [10], easing miniaturization, due to a limited heat dissipation; they are at the core of Cryptographic block cyphers analysis [16], and of quantum circuits synthesis [3,14]. Moreover, *reversibility* means that if we execute the circuits in the opposite direction, e.g. bottom-up instead of top-down, we are able to recover the input. RBC can nevertheless simulate all non-reversible classical circuits [17].



(a) Generators of RBC and their semantics

(b) An example of reduction on a Reversible Boolean Circuit

Fig. 1: Generators of Reversible Boolean Circuits and a reduction on them

Our focus is to study a class of RBC in the lines of [9], i.e. as a bi-dimensional diagrammatic formal language built by series and parallel composition of the generators; the diagrams can be rewritten by rewriting rules preserving generator interpretation, as given in Fig. 1a. The inputs of the generators are on top. We assume $x, y, c \in \{0, 1\}$, and from left to right we have the gates: "*Identity*", "*Swap*", "*Negation*" (which we also call "*Toffoli-one*"), "*Toffoli-two*", and "*Toffoli-three*". We denote the boolean XOR operation with $+$; the AND is by juxtaposition. Toffoli-two performs a *controlled negation* (`CNOT`): when the *controller* c is set to 1, x is negated, and is unchanged otherwise. The input x is always carried over as an output. Toffoli-three can be used to build conjuctions: if $c = 0$, $(x, y, 0) \mapsto (x, y, xy)$.

All classical boolean functions can be expressed in RBC, provided we add extra inputs/outputs in order to make the boolean functions invertible: in a reversible setting, the number of ouputs must be always equal to that of inputs.

All gates depicted in Fig. 1a represent invertible functions; moreover, they are self-inverse, i.e., if they yield an identity when applied twice.

Our goal in the long term is to answer specific domain questions related to the reversible language purposes. Such questions in the short term require to answer more classical questions about rewriting.

An example of specific reversibility question is: "Can we partition RBC into equivalence classes to find those containing the most efficient circuits, according to measures which depend on ancillae or the number of generators involved?". We recall that "ancillae" is a technical notion, to identify variables used as temporary storage in a circuit $C \in$ RBC. Ancillae allow C to compute the desired function while preserving the possibility of reverting the computation. A good reference to frame the role of ancillae in various contexts can be [13].

Another specific reversibility question can be: "How can we decide the equivalence of reversible circuits that we obtain by compilers which translate classical boolean circuits into reversible ones, driven by some heuristic aimed at optimizing specific parameters of the output?". The heuristic in [11] is just an example, of the many proposed in the literature, whose purpose, for instance, is to obtain good translations minimizing the quantity of ancillae.

Answering questions like the two above requires to formally and unambiguously know the representatives of equivalence classes in RBC. This amounts to asking basic questions about rewriting, such as: "What is and how do we get normal forms of RBC?"

Contributions. A natural strategy to answer the last question is to look for normal forms w.r.t. an interpretation-preserving rewriting on the circuits of RBC. Fig. 1b suggests what we mean by a simple example, in which we move a Toffoli-three gate next to another Toffoli-three so that they annihilate each other, because Toffoli-three are self-inverse operators. All *base* reversible gates are self-inverse, while, in general, reversible circuits do not need to be self-inverse.

Rewriting bi-dimensional diagrammatic formal languages has well-know difficulties, mainly concerning their relations with the usual (linear) syntaxes expressing the same problem.

We here explore the use of a free 3-category to represent both the reversible circuits and the rewriting rules on them. We build over Burroni's work [4], who introduced the notion of polygraphs to express algebraic theories and their reductions. Lafont [9] used bidimensional diagrammatic syntax on product categories and an informal measure on them to prove termination results on several classes of boolean functions; Guiraud used 3-polygraphs and diagrammatic reasoning to represent rewriting of circuit-like objects in [7,8], together with a formal method to build terminating measures. We focus on freely generated 3-categories instead of polygraphs, to have access to the rewriting paths as 3-morphisms; we call "*Toffoli* 3*-category*" the free 3-category we introduce, and we denote it with **Toff**.

We remind the reader that an *Abstract Rewriting System* (ARS) is said to be *terminating* when there are no infinite chains of subsequent reductions, i.e. all reductions eventually yield a (not necessarily unique) *normal form*. We show termination of the rewriting system in **Toff**: termination follows from defining a functor from **Toff** into a suitable 3-category **Move** of moves applied to wires inside circuits.

We remark that the functor we introduce simplifies the functorial and differential interpretations in [6,7,8]: our approach does not require to identify both a non-increasing measure on diagrams, which recalls a current flowing, and a strictly decreasing measure, connected to the "current", which recalls heat, generated by the "current" itself. We just need to identify a decreasing measure on a monoid of strings. The measure keeps track of the "moves" being applied to each individual wire, and assigns them a cost.

Plan of our work. We sketch the definition of 3-categories (§2), then we provide a bi-dimensional graphic representation for 3-categories representing circuits (§3). Next, we formally define the free 3-category **Toff** (§4), and a functor from **Toff** to a 3-category **Move** of movements through circuits (§5),which we show to be a decreasing measure of reductions. Eventually we prove termination of reductions on RBC (§6). The reduction rules we utilized are *syntactic* in nature; the normal form under algebraic equivalences remains an open problem [9].

2 3-Categories, 3-Functors and Free 3-Categories

In this section we will briefly go over the definition of n-categories. We provide enough details for the purposes of representing circuits and syntactic reductions on them (level 3). For a complete description of higher categories we refer to e.g. [2,5].

Motivations for using categories. The literature using product categories to describe circuits is extensive; we follow Burroni and Guiraud and use higher category theory to describe reduction systems on algebraic structures, with particular attention to circuits [4,8,9]. The "multi-level" structure of n-categories provides a suitable model for bi-dimensional objects such as circuits; the third level effortlessly captures the notion of a rewriting system on circuits. Moreover, the use of categories allows us to reason up to "bureaucratic" identities such as those invoked when shortening/lengthening the wires by adding/removing identities.

We give an anticipation on our 3-categorial model.

- The "base" level will contain a single token object $*$ representing the empty space between wires in a circuit.
- A first level will contain input and output wires, with a "monoidal" product to generate bundles of multiple wires. In our model, we will consider a single *wire* $|$ as a formal cell $* \to *$ between two 'empty spaces" $*$.
- A second level will contain circuits as morphisms between n input wires and m output wires (in our context $n = m$ because of reversibility), with appropriate series and parallel compositions.
- A third level will contain syntactic rewriting rules, considered as morphisms between circuits. Reductions should preserve the number of input/output wires of a circuit.

We have that each level consists of morphisms, named *cells*, with domain and codomain (here *source* and *target*) objects at the adjacent lower level. Source/target of a reduction is a circuit, source/target of a circuit is a set of I/O wires, and source/target of a wire is the empty space $*$. The *i-compositions* will provide an unified entity that captures both concatenation of wires, series and parallel composition of circuits, as well three different ways to compose reductions. Also, the properties of 0-, 1-, 2-composition will capture equations between circuits and between reductions. The levels up to the second one yield a description of circuits isomorphic to the product categories formalization.

We now provide an equational definition for n-categories following [15].

Definition 1 (n-categories). *A (strict) n-category $\mathcal{C}$ contains the following.*

- *A list of sets C_i, $0 \leq i \leq n$, called* levels*, whose elements are called i-*cells*.*
- *The maps $\mathtt{s}_i$, called i-*source*, and $\mathtt{t}_i$, called i-*target*, that associate to each j-cell x with $0 \leq i < j \leq n$ two i-cells $\mathtt{s}_i(x)$, $\mathtt{t}_i(x)$ we respectively call the i-source and the i-target of x;*
- *The j-cells $x \star_i y$, defined for all j-cells x, y and indices $0 \leq i < j \leq n$ such that $\mathtt{t}_i(x) = \mathtt{s}_i(y)$, called the i-*composition* of x and y, with k-source/target given by:*
  1. *$\mathtt{s}_k(x \star_i y) = \mathtt{s}_k(x)$ and $\mathtt{t}_k(x \star_i y) = \mathtt{t}_k(y)$ when $0 \leq k \leq i$;*
  2. *$\mathtt{s}_k(x \star_i y) = \mathtt{s}_k(x) \star_i \mathtt{s}_k(y)$ and $\mathtt{t}_k(x \star_i y) = \mathtt{t}_k(x) \star_i \mathtt{t}_k(y)$ when $j > k > i$.*

  The i-composition is denoted in diagrammatic order (left-to-right).

The data above define an n-category if they satisfy the following further conditions.

- ***Globularity.***

$$\mathtt{s}_{i-2}(\mathtt{s}_{i-1}(x)) = \mathtt{s}_{i-2}(\mathtt{t}_{i-1}(x)), \qquad \mathtt{t}_{i-2}(\mathtt{s}_{i-1}(x)) = \mathtt{t}_{i-2}(\mathtt{t}_{i-1}(x))$$

  for all i-cells x with $2 \leq i \leq n$. Globularity means that all i-cells connect two $(i-1)$-cells with the same $(i-2)$-source and $(i-2)$-target.
- ***Associativity** of each $\star_i$.*
- ***Local Units.** For all i-cells A with $0 \leq i < j \leq n$, there exists an identity j-cell $\mathtt{id}_{i,j,A}$, or $\mathtt{id}_A$ for short, such that $\mathtt{s}_i(\mathtt{id}_{i,j,A}) = \mathtt{t}_i(\mathtt{id}_{i,j,A}) = A$, the lower index source/targets of $\mathtt{id}_{i,j,A}$are those of A, and for all j-cells f we have*
  - *if $A = \mathtt{s}_i(f)$ then $\mathtt{id}_{i,j,A} \star_i f = f$, and*
  - *if $A = \mathtt{t}_i(f)$ then $f \star_i \mathtt{id}_{i,j,A} = f$.*

  Any i-composition of j-identity is a j-identity, for all $0 \leq i < j \leq n$.
- ***Exchange Rule.***

$$(\alpha \star_j \beta) \star_i (\gamma \star_j \delta) = (\alpha \star_i \gamma) \star_j (\beta \star_i \delta)$$

  for $\alpha, \beta, \gamma, \delta$ k-cells such that the above compositions are defined and all $0 \leq i < j < k \leq n$. □

To build the measure that proves termination of circuits reductions, we need the notion of 3-functor which we define for the general case.

Definition 2 (n-functor). *Let $\mathcal{C}$, $\mathcal{D}$ be n-categories. An n-functor $\varphi : \mathcal{C} \to \mathcal{D}$ is a map such that for all $0 \leq i \leq n$, φ sends i-cells of $\mathcal{C}$ into i-cells of $\mathcal{D}$, and such that for all i-cells f, g of $\mathcal{C}$, for all $0 \leq j < i \leq n$, and for all j-cells A we have:*

1. ***Source/Target preservation**. $\mathtt{s}_j(F(f)) = F(\mathtt{s}_j(f))$, $\mathtt{t}_j(F(f)) = F(\mathtt{t}_j(f))$*
2. ***Identity preservation (for i-cells and $\star_j$)**. $F(\mathtt{id}_{j,i,A}) = \mathtt{id}_{j,i,F(A)}$*
3. ***Composition preservation**. $F(f \star_j g) = F(f) \star_j F(g)$*

Free n-Categories

In the next section, we will represent circuits and reductions on them with a free 3-category. A free n-category consists of all well-formed expressions for objects in an n-category that are generated by a set of names for cells, identities and compositions, modulo all the equations we have for n-categories.

Definition 3 (Free n-Categories). *Let $\boldsymbol{G}_n$ be a* signature*, i.e. a list of sets $\boldsymbol{G}_n = (G_0, \ldots, G_n)$, with G_i sets of names for i-cells for $1 \leq i \leq n$, equipped with two maps $\mathtt{s}, \mathtt{t} : G_i \to G_{i-1}$, $1 \leq i \leq n$ such that the globularity requirement is met. We define the* free n-category $\mathcal{C}_n$ generated by the signature $\boldsymbol{G}_n$ *by induction on n.*

The 0-category $\mathcal{C}_0$ is just the set G_0. Assume we have defined the free $(n-1)$-category $\mathcal{C}_{n-1}$ generated by the signature $\boldsymbol{G}_{n-1}$ as the $(n-1)$-category with levels $C_0, C_1, \ldots, C_{n-1}$.

- *An* n-generator *of* C_n *is any name* $f \in \mathcal{C}_{n-1}$ *such that* $n = 1$ *or* $n \geq 2$ *and* f *satisfies the Globularity condition:* $\mathtt{s}_{n-2}(\mathtt{s}_{n-1}(f)) = \mathtt{s}_{n-2}(\mathtt{t}_{n-1}(f))$ *and* $\mathtt{t}_{n-2}(\mathtt{s}_{n-1}(f)) = \mathtt{t}_{n-2}(\mathtt{t}_{n-1}(f))$.
- *Let* E_n *be the set of* n-constants *of* C_n*, containing all* n*-generators and all expressions* $\mathtt{id}_{i,n,A}$ *denoting the* i*-identity on* A*, for* $0 \leq i \leq n-1$*,* $A \in C_i$*.*
- *Let* E_n^* *be the smallest set containing* E_n *and all expressions* $f \star_i g$ *such that* $f, g \in E_n^*$*,* $0 \leq i \leq n-1$ *and* $\mathtt{t}_i(f) = \mathtt{s}_i(g)$*. Source/target maps are defined on* E_n^* *by the source/target equations for* n*-categories.*

$\mathcal{C}_n$ *is defined as the* n*-category with levels* $C_0, \dots, C_{n-1}, C_n$ *and* $C_n = E_n^* / \sim$*, where* $\sim$ *is the smallest equivalence relation compatible with* $\star_0, \dots, \star_{n-1}$ *and including Associativity, Local Units and Exchange.*

Alternatively, the readers who are familiar with the definition of polygraphs [4,7,6] will notice that the free n-category generated by the signature G is isomorphic to the n-category generated by the corresponding n-polygraph, which itself lacks categorial structure at the n-th level [12].

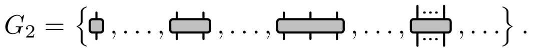
3 A Bi-dimensional Diagrammatic Syntax for 3-Categories

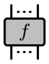
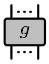
In this section we restrict to free 3-categories with a single 0-cell $*$ and a unique generator for 1-cell, i.e. the wire $|$, and we describe them as formal circuits and circuit reductions, by specifying what role the i-compositions take in the context of circuits. We introduce a diagrammatic syntax that represents such free 3-categories and the associated circuits. A great incentive to use a bi-dimensional syntax is that diagrams actually *look like* circuits. We stress the fact that the categorial setting together with a diagrammatic formalism give a thorough, compact and sound presentation for circuital theories. We invite the reader to think of i-compositions as a gluing operator of two objects along their common i-target and source, respectively.

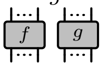
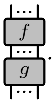
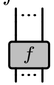
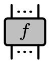
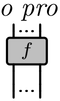
Definition 4 (Diagrams for a free 3-category). *Let us assume that* G *is a free* 3*-category with generator sets* $\boldsymbol{G} = (G_0, G_1, G_2)$*, source and target maps* $\mathtt{s}_i, \mathtt{t}_i$*, a single* 0*-cell and a unique generator for* 1*-cells. The diagrammatic representation of* G *is as follows.*

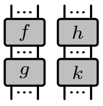
- ***Generators.***
  - $G_0 = \{*\}$ *consists of a unique* 0*-cell, representing a separator between input/output wires. We depict* $*$ *as a white area in the sheet of paper the diagram is drawn on.*
  - $G_1 = \{ | \}$ *consists of a unique* 1*-cell, which we call a* wire*. A wire is a formal cell between the two portions of sheet that it divides, both marked with* $*$*.*

  - *Each gate in G_2 is depicted as a circuit-like box with input and output wires representing* 1*-source and* 1*-target of the circuit.*

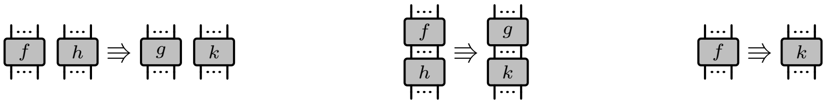
    If g is a gate with n input and output wires, we write $g : n \Rrightarrow n$. The diagrams depicting gates preserve the meaning of 2*-cells as formal maps between* 1*-cells (wires).*
  - 3*-cells in G_3, or reductions, are written as*  $\Rrightarrow$  *. We include no diagram for* 3*-cells: this would involve* 3*-dimensional objects [7].*

- *i**-cells.***
  - *The set G_0^* of all* 0*-cells is again $G_0 = \{*\}$.*
  - *The set G_1^* of all* 1*-cells consists of all possible* 0*-compositions $\star_0$ between* 1*-cell generators. Any element of G_1^* has the form $|* \mid *| \cdots *|$ or simply $| \quad | \ldots |$, freely generated as* 0*-composition along their common $*$ white area.*
  - *The set G_2^* of all* 2*-cells represents circuits, and it is obtained from gates in G_2 by closure w.r.t.* 0*-composition $\star_0$ (parallel composition, or composition along a common $*$ area), and* 1*-composition $\star_1$ (sequential composition, or composition along a common bundle of wires).*
  - ***Compositions of* 2*-cells.***
    - *Every two* 2*-cells are* 0*-composable, since they have as inputs and outputs* 1*-cells with the same source and target (the only* 0*-cell $*$). The* 0*-composition of* 2*-cells is depicted by putting the circuits next to each other along their common $*$ white area:*  *. This operation corresponds to the usual parallel composition and is read left-to-right.*
    - 1*-composition of* 2*-cells is defined for pairs of* 2*-cells such that* 1*-target of the first one (its output wires) is equal to the* 1*-source of the second one (its input wires).* 1*-composition vertically stacks circuits by connecting common I/O set of wires:*  *. This operation corresponds to series composition and is read top-to-bottom.*
  - ***Circuit equivalences.***
    - *The degenerate* 1*-composition $\mathtt{id}_A \star_1 f$ corresponds to prolonging the wires of $A = \mathtt{s}_1(f)$. The Unit rule*  $=$  $=$  *graphically shows that circuits are defined independently from the length of wires.*
    - *The Exchange rule $(f \star_1 g) \star_0 (h \star_1 k) = (f \star_0 h) \star_1 (g \star_0 k)$ says that the diagram*

      defines a single circuit which we can be equivalently read left-to-right, or top-to-bottom.

- ***3-cells**, or circuit reductions, can be composed with $\star_0, \star_1, \star_2$. If $\alpha : f \Rrightarrow g$ and $\beta : h \Rrightarrow k$, then the compositions behave as follows.*



$\alpha \star_0 \beta$ $\alpha \star_1 \beta$ (f, h *and* g, k 1*-composable)* $\alpha \star_2 \beta$ $(g = h)$

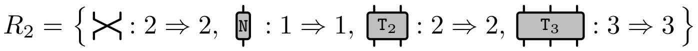
Remark 1. The diagrammatic notation, inspired by the string diagrams of categories, can be treated as a full 2-dimensional syntax for product categories, as explained in [1]. Again in [1] there is a termination result for the rewriting system where associativity, local units and the exchange rule are understood as rewriting rules and not as equivalences, and the concepts of *squeezed form* and *longest normal form* are introduced. In this paper, we shall limit ourselves to reasoning on diagrams modulo the above equivalences.

4 The Free 3-Category of Reversible Boolean Circuits

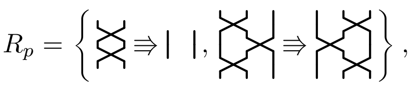
In this section we will describe, as a free 3-category **Toff**, both all reversible circuits we can obtain from the generators `SWAP`, `NOT`, `CNOT` (or $\mathtt{T}_2$, for Toffoli two), and $\mathtt{T}_3$ (or `CCNOT`), and a particular set of reductions on such circuits. This set of gates is proven to be universal (with ancillae) in [17]. The 3-category **Toff** includes, as its 3-generators, a reduction set composed of rules that replace each pair of consecutive `SWAP` or consecutive Toffoli gates by an identity, implementing involutivity of reversible gates. We also arrange circuits in a canonical form, with a "leaning" to moving a `SWAP` down, and when this is not possible, to the right. All reductions preserve the associated boolean function.

Definition 5 (The free 3-category Toff of Reversible Circuits). *The free* 3*-category of Reversible Circuits is the free* 3*-category specified by the following sets of* 0-, 1-, 2- *and* 3*-generators.*

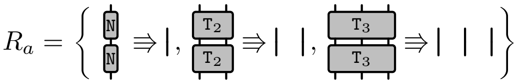
- $R_0 = \{*\}$, $R_1 = \{|\}$, *where* $| : * \to *$
- R_2 *contains the following generators for reversible circuits, in this order:* `SWAP`, `NOT`, $\mathtt{T}_2$, $\mathtt{T}_3$



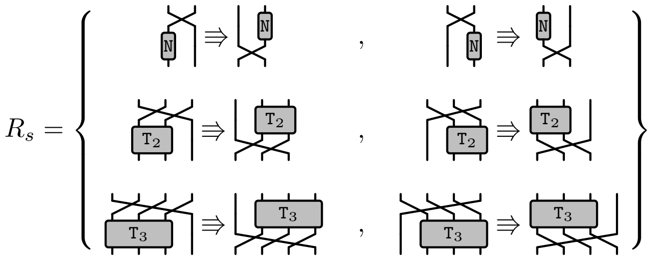
- $R_3 = R_p \cup R_a \cup R_s \cup R_t$, *where* R_p *contains the following* permutation rules*:*



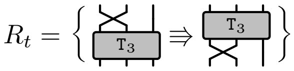
R_a *contains the following* annihilation rules*:*



R_s *contains the following* sliding rules*:*



and R_t *contains the following* Swapped Toffoli rule*:*



5 The Interpreting 3-Functor

In this section, we will define a 3-category **Move** of strings of "moves", each of them corresponding to the action of a gate on a single wire. Moves are elements of an ordered monoid that expresses the cost of series of singular moves along the wires of the circuit. The circuits are then measured by a componentwise order on all moves on all wires. The reduced circuit should describe an equivalent boolean function with a reduced total cost. The interpretation of circuits into moves will be given as a 3-functor from the free 3-category **Toff** of reversible circuits and a Toffoli base to **Move**.

Definition 6 (The ordered monoid $(\mathtt{M}, <_{\mathtt{M}})$ of moves).

1. *$\mathtt{M}$ is the free monoid of words generated from the letters $\mathtt{l}, \mathtt{r}, \mathtt{t}$.*
2. *We order $w_1, w_2 \in \mathtt{M}$ first by length, and when the lengths are the same, by the lexicographic order induced by the following order on letters: $\mathtt{t} <_{\mathtt{M}} \mathtt{r} <_{\mathtt{M}} \mathtt{l}$.*
3. *We write $<_{\mathtt{M}}$ for the order on $\mathtt{M}$.*

The letters $\mathtt{l}, \mathtt{r}, \mathtt{t}$ correspond to the three moves "left-to-right", "right-to-left" on the two wires of a `SWAP`, and to the move "Toffoli" on any wire of a Toffoli circuit. From $\mathtt{M}$ we define a 3-category **Move**.

Definition 7 (Move). **Move** *is a collection of the following sets and ordered sets of i-cells M_i, $0 \leq i \leq 3$.*

- *$M_0 = \{*\}$, a single element set.*
- *$(M_1, <_1)$ is the set of all cartesian powers $\mathtt{M}^n$ for $n \in \mathbb{N}$; the singleton $\mathtt{M}^0$ is the identical 1-cell, identified with the unique 0-cell $*$. We define an order on 1-cells as $\mathtt{M}^n <_1 \mathtt{M}^m$ if and only if $n < m$. We order vectors $\boldsymbol{w} \in \mathtt{M}^n$ componentwise, with the* product order *$<_{\mathtt{M}^n}$ on $\mathtt{M}^n$.*

- $(M_2, <_2)$ *is the set of all* $<_1$*-increasing maps* $f : \mathtt{M}^n \to \mathtt{M}^n$ *for some* $n \in \mathbb{N}$ *(all the identities* $\mathtt{id}_{\mathtt{M}^n}$ *are included). Let* $f, g : \mathtt{M}^n \to \mathtt{M}^n$ *be* 2*-cells with the same source and target. We define a* strictly pointwise *order* $f <_2 g$ *on* 2*-cells by:* $f <_2 g$ *if and only if* $f(x) <_{\mathtt{M}^n} g(x)$ *for all* $x \in \mathtt{M}^n$*. We write* $f \leq_2 g$ *for* $f = g \vee f <_2 g$*, that is: either* $f(\boldsymbol{x}) = g(\boldsymbol{x})$ *for all* $\boldsymbol{x}$*, or* $f(\boldsymbol{x}) <_{\mathtt{M}^n} g(\boldsymbol{x})$ *for all* $\boldsymbol{x} \in \mathtt{M}^n$*.*
- M_3 *contains all pairs* $r = \langle f, g \rangle$ *of* 2*-cells with the same source and target* $\mathtt{M}^n$ *for some* n*, such that* $f \geq_2 g$*. There is at most one* 3*-cell between* f, g*, which merely signals the existence of a* $\leq_2$*-relation between the two functions. We think of each pair* $\langle f, g \rangle$ *as a reduction from* f *to* g*. The reduction is identical if* $f = g$*, it is non-identical if* $f <_2 g$*.*

The i*-compositions on the* i*-cells are defined as follows.*

- 0***-composition on*** 1***-cells*** *is* $\mathtt{M}^n \star_0 \mathtt{M}^m = \mathtt{M}^{n+m}$*.*
- 0***-composition on*** 2***-cells*** *is the cartesian product of maps.*
- 1***-composition on*** 2***-cells*** *is the usual (sequential) composition of maps.*
- 0***-composition on*** 3***-cells*** *is defined as* $\langle f, g \rangle \star_0 \langle f', g' \rangle = \langle f \times f', g \times g' \rangle$*.*
- 1***-composition on*** 3***-cells*** *is defined as* $\langle f, g \rangle \star_1 \langle f', g' \rangle = \langle f'f, g'g \rangle$*.*
- 2***-composition on*** 3***-cells*** *is defined as* $\langle f, g \rangle \star_2 \langle g, h \rangle = \langle f, h \rangle$*.*

Remark 2. The order $<_{\mathtt{M}}$ on words is well-founded, and in fact $(\mathtt{M}, <_{\mathtt{M}})$ is order-isomorphic to the order $(\mathbb{N}, <)$ on natural numbers. The order $<_2$ on maps is well-founded. In fact, each decreasing sequence $f >_2 f' >_2 f" >_2 \ldots$ defines a decreasing sequence $f(\boldsymbol{\epsilon}) >_{\mathtt{M}^n} f'(\boldsymbol{\epsilon}) >_{\mathtt{M}^n} f"(\boldsymbol{\epsilon}) >_{\mathtt{M}^n} \ldots$, where $\boldsymbol{\epsilon} = (\epsilon, \epsilon, \ldots, \epsilon)$ and ϵ is the monoid unit (empty word). Therefore, every decresing sequence from f has length at most the number of words which are less than $f(\boldsymbol{\epsilon})$ in $\mathtt{M}^n$.

Proposition 1. **Move** *is a* 3*-category.*

Proof. We first prove that $\star_0$, $\star_1$ are increasing on 2-cells of **Move**. Assume $f \leq_2 f'$ and $g \leq_2 g'$, and either $f <_2 f'$ or $g <_2 g'$. Then $\star_0$ is increasing: for all $x \in \mathtt{M}^n = \mathtt{s}_1 f$, $y \in \mathtt{M}^m = \mathtt{s}_1 g$, we have $(f \times g)(x, y) = (f(x), g(y)) <_{\mathtt{M}^{n+m}} (f'(x), g'(y)) = (f' \times g')(x, y)$. $\star_1$ is increasing: if $\mathtt{t}_1(f) = \mathtt{M}^n = \mathtt{s}_1(g)$, then for all $x \in \mathtt{s}_1(f)$ we have $(gf)(x) = g(f(x)) \leq_{\mathtt{M}^n} g(f'(x)) \leq_{\mathtt{M}^n} g'(f'(x)$, and one of the two inequalities is strict, therefore $g(f(x)) <_{\mathtt{M}^n} g'(f'(x)$. We then have also that $\star_0$ and $\star_1$ are increasing on 3-cells of **Move**. As a consequence, i-cells are closed w.r.t. j-compositions. Associativity, unit and exchange axioms are straightforward. □

Definition 8. *A* 3*-functor* φ : **Toff** $\to$ **Move** *is* strict in a 3-cell α *if* α *is an identity* 3*-cell in* **Toff** *only if* $\varphi(\alpha)$ *is an identity* 3*-cell in* **Move***.* φ *is* strict *if* φ *is strict on all* 3*-cells* α *of* **Toff***.*

Proposition 2. *Assume* φ : **Toff** $\to$ **Move** *is a* 3*-functor which is strict on all generators of* α*. Then* φ *is strict.*

Proof. By induction on α, using the fact that φ is a 3-functor and 0-, 1-, 2- and 3-composition are increasing.

We define a 3-functor φ : **Toff** → **Move** which is strict on all generators for 0-, 1- ,2-cells. Recall that a word is written left-to-right, the last letter being the last move.

Definition 9 (Move interpretation). *We define a map φ by the following assignments on the generator gates of* **Toff***.*

$$\varphi(*) = * \qquad \varphi(|) = \mathtt{M} \qquad \varphi(\times)(v,w) = (w\mathtt{l}, v\mathtt{r})$$
$$\varphi(\boxed{\mathtt{N}})(v) = v\mathtt{t} \qquad \varphi(\boxed{\mathtt{T}_2})(v,w) = (v\mathtt{t}, w\mathtt{t})$$
$$\varphi(\boxed{\mathtt{T}_3})(v,w,z) = (v\mathtt{t}, w\mathtt{t}, z\mathtt{t}) \qquad \varphi(r) = \langle \varphi(\mathtt{s}_2 r), \varphi(\mathtt{t}_2 r)\rangle$$

Lemma 1. *The map φ of Def. 9 extends in a unique way to a 3-functor* **Toff** → **Move***.*

Lemma 2 (Termination Lemma). *If the free 3-category* **Toff** *has a 3-functor to* **Move** *which is strict on all generators for 3-cells, then all chains of non-identical 3-cells in* **Toff** *terminate.*

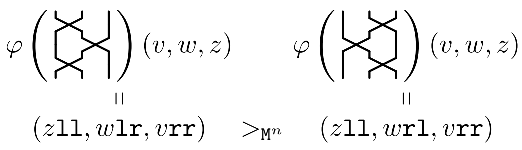
Toff is freely generated from a free 3-category, therefore φ is entirely and uniquely defined by the assignments on the generators of **Toff**, provided we recursively check that the 3-functor φ preserves sources and targets, and sources and targets of sources.

6 A Termination Result for Reversible Boolean Circuits

Theorem 1. *The free 3-category of Reversible Boolean Circuits terminates (all reduction sequences are finite).*

Proof (of Termination for Reversible Boolean Circuits with the Toffoli base). By Lemma 2, we have to prove that φ is strict on all generators for 3-cells. By definition of strictness, it is enough to prove that for all $\alpha \in R_3$ in **Toff** we have that $\varphi(\alpha)$ is not an identity. By definition, $\varphi(\alpha)$ is equal to the pair $\langle \varphi(\mathtt{s}_2\alpha), \varphi(\mathtt{t}_2\alpha)\rangle$. $\varphi(\alpha)$ is not an identity if $\varphi(\mathtt{s}_2\alpha) >_2 \varphi(\mathtt{t}_2\alpha)$. Suppose $\mathtt{s}_1(\alpha) = \mathtt{M}^n$. By definition of the pointwise order on 2-cells of **Move**, we have to prove that $\varphi(\mathtt{s}_2\alpha)(v_1, \dots, v_n) >_{\mathtt{M}^n} \varphi(\mathtt{t}_2\alpha)(v_1, \dots, v_n)$ for all $v_1, \dots, v_n \in \mathtt{M}$.

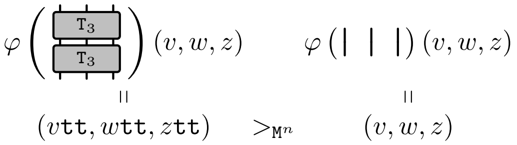
– **Permutation rules.**



$$\varphi\left(\quad\right)(v,w,z) \qquad \varphi\left(\quad\right)(v,w,z)$$
$$\begin{array}{ccc} \shortparallel & & \shortparallel \\ (z\mathtt{ll}, w\mathtt{lr}, v\mathtt{rr}) & >_{\mathtt{M}^n} & (z\mathtt{ll}, w\mathtt{rl}, v\mathtt{rr}) \end{array}$$

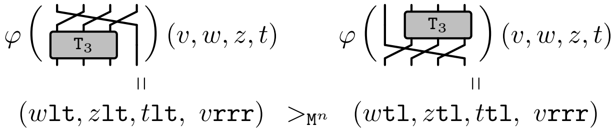
The thesis follows from $\mathtt{lr} > \mathtt{rl}$.

- **Annihilation rules.**



$$\varphi(\,\cdot\,)(v,w,z) = (v\mathtt{tt}, w\mathtt{tt}, z\mathtt{tt}) >_{\mathtt{M}^n} (v,w,z) = \varphi(|\;|\;|)(v,w,z)$$

  Words on the right-hand-size are all shorter and therefore smaller.
- **Left-to-Right Sliding rules.**
  The reduction moving the Toffoli gate upwards and to the right swaps the lowest letter of the monoid `t`, with a higher letter, `l`, in all movements v, w, z, t but v. Below, we consider the case of the circuit Toffoli 3.



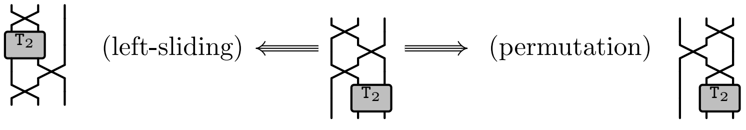
$$\varphi(\,\cdot\,)(v,w,z,t) = (w\mathtt{lt}, z\mathtt{lt}, t\mathtt{lt},\ v\mathtt{rrr}) >_{\mathtt{M}^n} (w\mathtt{tl}, z\mathtt{tl}, t\mathtt{tl},\ v\mathtt{rrr}) = \varphi(\,\cdot\,)(v,w,z,t)$$

  The thesis follows from $\mathtt{lt} >_{\mathtt{M}} \mathtt{tl}$. The Right-to-Left Sliding rules are the mirror case.
- **Swapped Toffoli.** This case follows from $\mathtt{lt} > \mathtt{tl}$ and $\mathtt{rt} > \mathtt{tr}$.

□

7 Conclusion

This work explores 3-categories as a unified formal framework for modeling two concepts. First, it examines Reversible Boolean Circuits, which are treated as diagrams generated from a base through the iterative application of series/parallel compositions. Then, it investigates the termination of a rewriting system on these circuits. **Toff** is the free 3-category which effectively formalizes circuits and rewriting rules. **Move** is the 3-category supplying the well founded-order in a monoid of strings. Termination follows from interpreting **Toff** into **Move** by means of a (strict) functor φ that intuitively represents traces of moves inside the circuit being rewritten, by exploiting the common 3-category structure. A possible extension of this work is to evaluate its effectiveness in proving the termination of reductions of free 3-polygraphs generated from alternative bases which include Fredkin and Peres gates. A second development must focus on confluence: rewriting in **Toff** is not. We claim that the following circuits has two non-confluent normal forms:



We are pursuing various directions in order to obtain confluence. However, the problem at hand is known to be non-obvious. Guiraud warns that a 3-polygraph may generate an infinite number of critical pairs which, in specific cases, can be categorized into a finite set of patterns, eventually leading to confluence [6].

References

1. Acclavio, M.: String Diagram Rewriting: Applications in Category and Prooftheory. Ph.D. thesis, Aix-Marseille Université (2016)
2. Baez, J.C.: An introduction to n-categories. In: Moggi, E., Rosolini, G. (eds.) Category Theory and Computer Science. pp. 1–33. Springer Berlin Heidelberg, Berlin, Heidelberg (1997)
3. Barenco, A., Bennett, C.H., Cleve, R., DiVincenzo, D.P., Margolus, N., Shor, P., Sleator, T., Smolin, J.A., Weinfurter, H.: Elementary gates for quantum computation. Phys. Rev. A **52**, 3457–3467 (Nov 1995). https://doi.org/10.1103/PhysRevA.52.3457, https://link.aps.org/doi/10.1103/PhysRevA.52.3457
4. Burroni, A.: Higher-dimensional Word Problems with Applications to Equational Logic. Theoret. Comput. Sci. **115**(1), 43–62 (1993). https://doi.org/https://doi.org/10.1016/0304-3975(93)90054-W
5. Cheng, E., Lauda, A.: Higher-dimensional categories: an illustrated guide book. https://eugeniacheng.com/wp-content/uploads/2017/02/cheng-lauda-guidebook.pdf (2004)
6. Guiraud, Y.: Termination orders for three-dimensional rewriting. Journal of Pure and Applied Algebra **207**(2), 341–371 (2006). https://doi.org/https://doi.org/10.1016/j.jpaa.2005.10.011
7. Guiraud, Y., Bonfante, G.: Polygraphic programs and polynomial-time functions. Log. Methods Comput. Sci. **5** (2009)
8. Guiraud, Y., Malbos, P.: Higher-dimensional Categories with Finite Derivation Type. Theory Appl. Categ. **22**, 420–478 (2009)
9. Lafont, Y.: Towards an algebraic theory of boolean circuits. Journal of Pure and Applied Algebra **184** (2003)
10. Landauer, R.: Irreversibility and heat generation in the computing process. IBM J. Res. Dev. **5**(3), 183–191 (1961)
11. Meuli, G., Soeken, M., Roetteler, M., Bjorner, N., Micheli, G.D.: Reversible Pebbling Game for Quantum Memory Management. In: 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 288–291 (2019)
12. Métayer, F.: Cofibrant objects among higher-dimensional categories. Homology, Homotopy and Applications **10**, 185–188 (2008). https://doi.org/10.4310/HHA.2008.v10.n1.a7
13. Perumalla, K.S.: Introduction to Reversible Computing. Chapman & Hall/CRC Computational Science, CRC Press (2013)
14. Saeedi, M., Markov, I.L.: Synthesis and optimization of reversible circuits—a survey. ACM Comput. Surv. **45**(2) (2013). https://doi.org/10.1145/2431211.2431220, https://doi.org/10.1145/2431211.2431220
15. Street, R.: The algebra of oriented simplexes. Journal of Pure and Applied Algebra **49**, 304–305 (1987)
16. Táborský, D., Larsen, K.F., Thomsen, M.K.: Encryption and reversible computations. In: Kari, J., Ulidowski, I. (eds.) Reversible Computation. pp. 331–338. Springer International Publishing, Cham (2018)
17. Toffoli, T.: Reversible computing. In: de Bakker, J.W., van Leeuwen, J. (eds.) Automata, Languages and Programming, 7th Colloquium, Noordweijkerhout, The Netherland, July 14-18, 1980, Proceedings. Lecture Notes in Computer Science, vol. 85, pp. 632–644. Springer (1980)