A Superposition-Based Calculus for Diagrammatic Reasoning
Nicolas Peltier, Mnacho Echenim, Rachid Echahed, Mehdi Mhalla

To cite this version:
A Superposition-Based Calculus for Diagrammatic Reasoning
(Long Version)

Rachid Echahed
Mnacho Echenim
Mehdi Mhalla
Nicolas Peltier
Univ. Grenoble Alpes, LIG, CNRS/GINP
F-38000, Grenoble, France

ABSTRACT
We introduce a class of rooted graphs which are expressive enough to encode various kinds of classical or quantum circuits. We then follow a set-theoretic approach to define rewrite systems over the considered graphs. Afterwards, we tackle the problem of equational reasoning with the graphs under study and we propose a new Superposition calculus to check the unsatisfiability of formulas consisting of equations or disequations over these graphs. We establish the soundness and refutational completeness of the calculus.

CCS CONCEPTS
• Theory of computation → Automated reasoning: Equational logic and rewriting.

KEYWORDS
Equational Reasoning, Graph, Superposition Calculus

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:

1 INTRODUCTION
Rewrite systems play an important rôle in defining operational semantics of declarative programming languages such as functional programming [3, 26, 32] or functional-logic programming [16, 24]. In general, the basic objects of such declarative languages are trees, e.g., lambda terms or first-order terms. In this paper, we consider rewrite systems operating on a class of rooted graphs. Such systems are tailored to model applications based on circuits building and simulation. The road to computing by means of graph rewrite systems started in the seventies [19] using the well-known algebraic double pushout approach. Since then, several approaches of graph rewriting have been proposed. One can distinguish two main streams of research : the algebraic approaches using constructs borrowed from category theory [14, 15, 18, 29] on one hand and the set-theoretic or algorithmic approaches [4, 17, 20] based on actual algorithms/operations involved in graph transformations on the other hand. All the proposed approaches define dedicated solutions to the delicate action of replacing a given subgraph by another in the rewriting process. Subgraph replacement is more difficult to achieve in general if compared to subterm (subtree) replacement. This difficulty is mainly due to the way one handles the possible edges connecting nodes of the subgraph to be replaced with the rest of the considered graph. In this work, we propose a set-theoretic approach to define a class of rewrite systems over rooted graphs. We introduce the basic operations over graphs needed to work out the machinery used to define rewriting steps. Due to the shape of the graphs we consider, our definition of graph rewriting is rather close to term rewriting and thus avoids the difficulties behind subgraph replacement actions.

Rule-based programming offers the possibility to write programs in a declarative way by stating rules or inferences behind the intended algorithm without caring for the control in a first step. Another benefit of rule-based programming is the underlying mathematical foundation which allows one to make some formal reasoning on programs. For example, in the case of term rewrite systems, equational reasoning has been developed intensively and now is routine in most theorem provers. Unfortunately, reasoning on graph structures is still in its infancy despite the progress made in using Hoare-like calculi [9, 22, 34] or model-checkers [35, 39]. In this paper, we also tackle the problem of reasoning with graphs and propose an extension to graphs of the well-known Superposition calculus.

The Superposition calculus [1] is the most successful automated proof procedure for equational reasoning. The calculus may be viewed as an extension of the Knuth-Bendix algorithm [28] to a wider class of equational formulæ, involving disjunctions (e.g., conditional rules) and quantifiers. It also generalizes the narrowing procedure [25] by handling equations that cannot be oriented and may be viewed as an extension of the Resolution calculus [31, 37], which is the most efficient proof procedure for predicate logic, to sets of clauses containing the equality predicate. The calculus is presented as a set of inference rules, deducing new assertions from axioms or previously generated assertions, together with a generic redundancy criterion that permits to prune the search space and discard many inferences. This calculus is sound, in the sense that every derived formula is a logical consequence of the premises, and
refutationally complete: if a set of axioms $E$ is contradictory, then a contradiction (i.e., an empty clause, equivalent to false) can always be generated by the calculus. It is also generic, and can be uniformly applied to any set of axioms that can be expressed in first-order logic. Most efficient theorem provers, such as Vampire [36], E [38] or Spass [42], are based on the Superposition calculus. Numerous extensions of the Superposition calculus have been proposed to handle specific theories or extensions of first-order logic, for instance higher-order logic [6], inductive and co-inductive datatypes [7], cancellative monoids [41], or general-purpose theory-reasoning [5]. In the present paper, we extend the calculus to perform equational reasoning on graphs instead of terms and we establish the soundness and refutational completeness of the obtained proof procedure.

Graphs are ubiquitous in programming and verification. They are often used for example to model pointer-based data-structures, and equations between graphs may be viewed as specifications of functions operating on such data-structures. In this context, proving general properties of these functions involves performing equational reasoning on graphs. In quantum computing, graphs have become particularly useful for the formal verification of quantum algorithms and protocols, and several such languages have been developed over the past years for such an analysis, including the ZW calculus [23], the ZH calculus [2], the ZX calculus [13], the SZX [10] calculus, or more recently the PBS-calculus [11]. The reason these languages are so useful is that standard approaches to verifying the correctness of quantum algorithms and protocols involve computations on complex matrices, and such approaches are non-intuitive and error-prone. On the other hand, graph-based languages permit to represent the same algorithms and protocols in a more intuitive way, by abstracting the numerical values and matrices, and replacing computations on these matrices by rewrite rules. Using graph-based languages to represent quantum algorithms thus permits to reduce many verification tasks to testing the equivalence of two graphs modulo a set of equations, which represent properties of the considered computations.

Extending the Superposition calculus to reason on graphs is thus a very desirable feature, with promising applications in programming. A first, rather straightforward, approach would be to encode graphs into first-order terms, stating all the necessary properties as axioms. However, such an approach would be very inefficient and the produced proofs would be extremely long and hardly readable, since they will be cluttered by the many reasoning steps necessary to handle trivial transformations related to associativity, commutativity, isomorphism etc. A second, more promising, approach is to tune the Superposition calculus so that it applies directly on graphs, which would replace terms as basic objects, with built-in algorithms for establishing the completeness of the Superposition calculus. Numerous extensions of the Superposition calculus have been proposed to handle specific theories or extensions of first-order logic, for instance higher-order logic [6], inductive and co-inductive datatypes [7], cancellative monoids [41], or general-purpose theory-reasoning [5]. In the present paper, we extend the calculus to perform equational reasoning on graphs instead of terms and we establish the soundness and refutational completeness of the obtained proof procedure.

Related Work

The very recent research report [21] is very close to our work. It pursues similar aims, namely to construct automatically equational derivations on diagrammatic structures, more precisely on ZX diagrams [12]. The considered approach, which is practically very successful, is different from ours: it relies on the use of the standard Superposition calculus [1] together with an encoding of ZX diagrams and rules for terms and equations between them (including an embedding of higher-order terms to cope with schematic rules). In our work, we bypass this encoding by defining a Superposition calculus that applies directly on graphs. The advantage is that we do not have to model graph properties by first-order axioms (such as the associativity and commutativity of graph constructors, see, e.g. [21, Equations 303-304, p. 80]), instead these properties are taken into account in the definition of the inference rules. We do not use schematic rules to cope with variadic gates, instead instances of these rules with fixed arities may be generated on demand during the proof search.
Numerous works in graph rewriting use an algebraic framework based on categories theory, using mostly the double pushout approach introduced in the seminal paper [19] (which has since been generalized in many directions). Here, we prefer to use an algorithmic, set theoretical framework. This low-level approach, although less general, is more suited to deal with the intricacies in the definition of the inference rules and in the completeness proof. The considered structures are also closer to those considered usually in superposition-based theorem proving, which is more convenient for extending the calculus to graphs.

In [8], it is proven that confluence is decidable for terminating rewrite systems on graphs with an interface, which embed those considered in the present paper (as mentioned above, confluence is undecidable in general in graph rewriting). In our results, the rewrite rules we consider are not necessarily terminating (and equations are not even orientable in general). In Section 7, confluence is used as a tool to construct models of saturated sets of clauses. It is easy to check that a terminating set of rules with no non joinable critical pair is always saturated (in the sense of Definition 53 and w.r.t. some adequate order), thus the decidability of confluence could be re-established for our particular class of graphs using Lemma 56 in the present paper.

We wish to emphasize that the notion of “completeness” considered in the present paper refers to the completeness of the inference rules, i.e., to their ability to detect that graphs are reducible to each other. It should not be confused with work asserting the completeness of the diagrammatic axiomatization w.r.t. a particular semantics and application domain (such as [40]).

The rest of the paper is organized as follows. Some basic definitions are recalled in the next section. The class of rooted graphs is introduced in Section 3 together with some basic operations on these graphs as well as the considered class of graph rewrite systems. Sections 4 and 5 define the graph formulas and their semantics. In Section 6, the new Superposition calculus is presented and its completeness is established in Section 7. Section 8 provides examples of the considered graphs and concluding remarks are made in Section 9.

2 BASIC DEFINITIONS AND NOTATIONS

For any partial function $f$, we denote by $\text{dom}(f)$ its domain, i.e., the set of elements $x$ such that $f(x)$ is defined and by $\text{img}(f)$ the set $\{f(x) \mid x \in \text{dom}(f)\}$. For any function $f$ and for any $D \subseteq \text{dom}(f)$, we denote by $f|_D$ the restriction of $f$ to $D$. The integer interval $\{a, a+1, \ldots, b\}$ is denoted by $[a,b]$ (it is empty if $a > b$). The function of domain $\{t_1, \ldots, t_n\}$ mapping $t_i$ to $t_j$ for all $i \in [1,n]$ is denoted by $(t_1 \mapsto t_1 | i \in [1,n])$.

Any partial function $f$ operating on some set $S$ may be extended into a function operating on tuples or sets of elements in $S$, using the relations: $f((t_1, \ldots, t_n)) \triangleq \langle f(t_1), \ldots, f(t_n) \rangle$ and $f((t_1, \ldots, t_n)) \triangleq f(f((t_1, \ldots, f(t_n)))$. If $t_i \not\in \text{dom}(f)$ for some $i = 1, \ldots, n$ then $f((t_1, \ldots, t_n))$ and $f((t_1, \ldots, t_n))$ are undefined. These relations may be applied recursively, e.g., if $t_1, \ldots, t_n$ are themselves sets or tuples of elements.

The notation $g \circ f$ denotes as usual the composition of $g$ and $f$, with $\text{dom}(g \circ f) = \{x \in \text{dom}(f) \mid f(x) \in \text{dom}(g)\}$ and $(g \circ f)(x) = g(f(x))$ for every $x \in \text{dom}(g \circ f)$. According to the previous convention, if functions are viewed as sets of pairs, then for every function $f$ and for every injective function $g$ such that $\text{img}(f) \subseteq \text{dom}(g)$, $g(f(x))$ is a function with $\text{dom}(g(f)) = g(\text{dom}(f))$, $\text{img}(g(f)) = g(\text{img}(f))$, and $(g(f)(g(x)) = g(f(x))$, for any $x \in \text{dom}(f)$. Note that, by definition, $g(f) = g \circ f \circ g^{-1}$.

With a slight abuse of notations, we sometimes use set notations on tuples when the order and duplication of components are irrelevant, i.e., we write $u \equiv \langle t_1, \ldots, t_n \rangle$ to state that $u \equiv \langle \{t_1, \ldots, t_n\} \cup E \rangle$ to denote the set $\{t_1, \ldots, t_n\} \cup E$ or $\langle t_1, \ldots, t_n \rangle = \langle t'_1, \ldots, t'_m \rangle \cup \langle t''_1, \ldots, t''_k \rangle$ for $\{t_1, \ldots, t_n\} = \{t'_1, \ldots, t'_m, t''_1, \ldots, t''_k\}$.

**Definition 1.** Let $\Sigma$ be a set of function symbols and let $\mathcal{V}$ be a countable set of variables. Each symbol $f \in \Sigma$ is associated with a unique arity $\text{ar}(f)$. The set of terms $\mathcal{T}$ is the least set such that $\mathcal{V} \subseteq \mathcal{T}$ and $t_1, \ldots, t_n \in \mathcal{T}$ implies $f(t_1, \ldots, t_n) \in \mathcal{T}$, for every $f \in \Sigma$ with $n = \text{ar}(f)$. We denote by $\mathcal{V}(t)$ the set of variables occurring in $t$. A term $t$ is ground if $\mathcal{V}(t) = \emptyset$.

A substitution $\sigma$ is a total mapping from $\mathcal{V}$ to $\mathcal{T}$. As usual, for any term $t$, $\sigma(t)$ denotes the term obtained from $t$ by replacing every variable $x$ by $\sigma(x)$. A substitution $\sigma$ is ground if $\sigma(x)$ is ground for every $x \in \text{dom}(\sigma)$. A unifier of a set of pairs $E$ is a substitution $\sigma$ such that $\sigma(t) = \sigma(s)$ holds for all pairs $(t, s) \in E$. $E$ is unifiable if it admits a unifier. A substitution $\sigma$ is more general than a substitution $\theta$ iff there exists a substitution $\sigma'$ such that $\theta = \sigma' \circ \sigma$. It is well known that every unifiable set of pairs admits a most general unifier (mgu).

3 ROOTED GRAPHS

3.1 Definitions

We define the class of graphs on which the adaptation of the Superposition calculus will be applied. Intuitively, this class consists of labeled graphs with a distinguished sequence of nodes which can be viewed as an interface that constrains the operations of replacing a subgraph by another graph that can be performed. More precisely, the replacement is possible only if all the edges between the original graph and the replaced subgraph have an incident node inside the interface.

Let $N$ be a fixed countable set of nodes, disjoint from the set of terms, and let $S$ be a set of sorts. We consider a function sort mapping every node $a \in N$ to a sort in $S$ and a pre-order $\preceq$ on $N$. We write $a \preceq a'$ if $a \preceq a'$ and $a' \preceq a$, and $a < a'$ for $a \preceq a'$ and $a' \not\preceq a$.

**Definition 2.** A DRL-graph (for Directed Rooted Labeled graph) $G$ is a tuple $\langle N_G, R_G, E_G, \ell_G \rangle$ where:

- $N_G$ is a finite subset of $N$.
- $R_G$ is a finite sequence of nodes in $N_G$ with no repetition, called the roots of $G$. We denote by $N_G^*$ the set of nodes $N_G \setminus R_G$.
- $E_G$ is a finite set of ordered pairs (written $(a \rightarrow b)$) of nodes in $N_G$, called edges. We write $(a \leftarrow b)$ to denote any edge $(a \rightarrow b)$ or $(b \rightarrow a)$.
- $\ell_G$ is a function mapping each node $a \in N_G^*$ to a term (called the label of $a$).

**Lemma 56**
Two DRL-graphs $G$ and $H$ are disjoint if $N_G \cap N_H = \emptyset$. We denote by $\mathcal{V}(G) \triangleq \bigcup_{\alpha \in N_G^*} \mathcal{V}(l_G(\alpha))$ the set of variables occurring in the labels of nodes in $G$. A DRL-graph $G$ is ground if $\mathcal{V}(G) = \emptyset$.

Note that the function $l_G$ is not defined over the nodes in $R_G$.

**Definition 3.** For any substitution $\sigma$ and DRL-graph $G$, $\sigma(G)$ denotes the DRL-graph $(\langle G; R_G; E_G, I' \rangle)$, where $I'(\alpha) = \sigma(l_G(\alpha))$, for all $\alpha \in N_G^*$.

**Definition 4.** A DRL-graph $H$ is a DRL-subgraph of $G$ (written $H \preceq G$) if the following conditions hold:

1. $N_H \subseteq N_G$;
2. We have $(\alpha \rightarrow \beta) \in E_H$ then $(\alpha \rightarrow \beta) \in E_G$.
3. If $(\alpha \rightarrow \beta) \in E_G$ and $\alpha, \beta \in N_H^*$ then $(\alpha \rightarrow \beta) \in E_H$.
4. If $(\alpha \rightarrow \beta) \in E_G$, $\beta \notin N_H$ and $\alpha \in N_H$, then $\alpha$ occurs in $R_H$.
5. If $\alpha$ occurs in $R_G$ and $\alpha \in N_H$, then $\alpha$ occurs in $R_H$.

**Proposition 5.** The relation $\preceq$ is transitive and reflexive.

**Proof.** It is immediate to check that $\preceq$ is reflexive. Assume that $I \preceq H \preceq G$. Then:

1. $N_I \subseteq N_H \subseteq N_G$ hence $N_I \subseteq N_G$.
2. We have $(\alpha \rightarrow \beta) \in E_I$ then $(\alpha \rightarrow \beta) \in E_G$.
3. $(\alpha \rightarrow \beta) \in E_G$ and $\alpha, \beta \in N_I^*$, then by Condition 5 of Definition 4 $\alpha, \beta \in N_H^*$ thus $(\alpha \rightarrow \beta) \in E_H$ and $(\alpha \rightarrow \beta) \in E_G$.
4. Assume that $\alpha \rightarrow \beta \in E_G$, $\alpha \in N_I$ and $\beta \notin N_I$. Then $(\alpha \rightarrow \beta) \in E_H$ and we deduce that $\alpha$ must occur in $R_I$ because $I \preceq H$. If $\beta \notin N_H$ then $\alpha$ occurs in $R_H$ since $H \preceq G$. Thus $\alpha \in R_I$, since $I \preceq H$.
5. $I_I = l_I|_{N_I^*} = (l_G|_{N_H^*})|_{N_I} = l_G|_{N_I^*}$ (because $N_I^* \subseteq N_H^*$).

**Proposition 6.** If $H \preceq G$ then $R_G \subseteq N_G \setminus N_H^*$.

**Proof.** Assume for a contradiction that $\alpha \in R_G \cap N_H^*$. Then by definition, $\alpha$ is a node in $N_H$ that does not occur in $R_H$, but this is impossible by Condition 5 of Definition 4.

In what follows, we define an equivalence relation on graphs to formalize the fact that the properties we prove do not depend on the actual nodes that occur in the graph.

**Definition 7.** An $N$-mapping $\mu$ is a partial injective mapping from nodes to nodes, such that $\text{sort}(\alpha) = \text{sort}(\mu(\alpha))$ and $\alpha \preceq \mu(\alpha)$. For technical convenience we assume that $\mu(t) = t$ holds for every term $t$, i.e., every $N$-mapping is extended to the identity on terms.

**Proposition 8.** If $G$ is a DRL-graph, then for every $N$-mapping $\mu$ with $\text{dom}(\mu) \supseteq N_G$, $\mu(G)$ is a DRL-graph. Moreover, $N_{\mu(G)} = \mu(N_G)$, $\mathcal{E}_{\mu(G)} = \mu(\mathcal{E}_G)$, $l_{\mu(G)} = \mu(l_G)$ and $l_{\mu(G)} = \mu(l_G) = \mu \circ l_G \circ \mu^{-1}$.

**Proof.** The result follows immediately from the definition of $\mu(G)$ (see Section 2) and from the fact that $\mu$ is injective.

Similarly, because $N$-mappings are injective, we have the following result:

**Proposition 9.** If $H \preceq G$ and $N_G \subseteq \text{dom}(\mu)$ for an $N$-mapping $\mu$, then $\mu(H) \preceq \mu(G)$.

**Definition 10.** We write $G \sim G'$ if there exists $\mu$ such that $\text{dom}(\mu) \supseteq N_G$ and $\mu = \mu(G')$.

**Proposition 11.** The relation $\sim$ is an equivalence relation.

**Proof.**

- **Reflexivity.** The identity on $N_G$ is an $N$-mapping, thus $G \sim G$.
- **Symmetry.** If $G \sim G'$, then $G = \mu(G')$, where $\mu$ is an $N$-mapping such that $\text{dom}(\mu) \supseteq N_G$. Since $\mu$ is injective, $\mu^{-1}$ is well-defined, $\mu^{-1}(G) = \mu^{-1}(G') = G'$ and $\text{dom}(\mu^{-1}) = \text{im}(\mu) \supseteq \mu(N_G) = N_G$. Hence $G \sim G'$.
- **Transitivity.** If $G \sim G'$, then there exist $N$-mappings $\mu$ and $\mu'$ such that $G = \mu(G')$, $G' = \mu'(G''')$, $\text{dom}(\mu) \supseteq N_G$ and $\text{dom}(\mu') \supseteq N_G$. Then since the composition of two injective functions is also injective, $\mu \circ \mu'$ is an $N$-mapping, and $G = (\mu \circ \mu')(G''')$ with $\text{dom}(\mu \circ \mu') \supseteq N_G$. Hence $G \sim G'''$.

We now define a replacement operation on the considered class of graphs. Intuitively, this replacement operation applied to a graph permits to replace one of its subgraphs by another graph, provided a so-called substitutability condition is satisfied. This condition will guarantee that the replacement operation on graphs enjoys properties similar to those of standard term rewriting.

**Definition 12.** Let $G, H$ and $H'$ be DRL-graphs such that $H \preceq G$, $R_H = \langle a_1, \ldots, a_m \rangle$ and $R_{H'} = \langle a'_1, \ldots, a'_m \rangle$. The DRL-graphs $H$ and $H'$ are root-compatible if $n = m$ and for every $i \in [1, n]$, we have $\text{sort}(a_i) = \text{sort}(a'_i)$ and $a_i \preceq a'_i$.

**Definition 13.** A DRL-graph $H'$ is substitutable for $H$ in $G$ if $H$ and $H'$ are root-compatible and $N_G \cap N_{H'} = \emptyset$.

An immediate consequence of this definition is the following property:

**Proposition 14.** If $H'$ is substitutable for $H$ in $G$ then $(N_G \setminus N_H) \cap N_{H'} = \emptyset$.

**Definition 15.** Let $G$, $H$ and $H'$ be DRL-graphs such that DRL-graph $H'$ is substitutable for $H$ in $G$, $H \preceq G$, $R_H = \langle a_1, \ldots, a_n \rangle$ and $R_{H'} = \langle a'_1, \ldots, a'_n \rangle$. The mapping from $H$ to $H'$ in $G$, is the function $\mu_{G \rightarrow H'} \triangleq \{ \alpha \mapsto \alpha' \mid 1 \leq i \leq n \} \cup \{ \alpha \mapsto \alpha \mid \alpha \in N_G \setminus N_H \}$.

Let $\mu = \mu_{G \rightarrow H'}$. We denote by $G[H'/H]$ the DRL-graph $G'$ defined as follows:

1. $N_{G'} = (N_G \setminus N_H) \cup N_{H'}$.
2. $R_{G'} = \mu(R_G)$.
3. $E_{G'} = \mu(E_G) \cup E_{H'}$.
4. For every $\alpha \in N_{G'}$:
   
   \[ l_{G'}(\alpha) = \begin{cases}  
   l_G(\alpha) & \text{if } \alpha \in N_G \setminus N_{H'} \\
   l_{H'}(\alpha) & \text{if } \alpha \in N_{H'} \\
   l_G(\mu^{-1}(\alpha)) & \text{if } \mu^{-1}(\alpha) \in N_G \cap R_H \\
   \text{undefined otherwise} 
   \end{cases} \]

**Example 16.** Consider the following DRL-graphs (see also Figure 1):
We assume that all nodes are of the same sort and that all nodes are distinct. Then $H$ is a subgraph of $G$, and $H'$ is substitutable for $H$ in $G$. The DRL-graph $G[H'/H]$ is represented in Figure 2, where $l_{G[H'/H]} = \{(a_1, a), (b_1, f(x)), (b_2, f(b)), (b_3, c), (b_4, d), (a_4, b)\}$.

**Proposition 17.** If $H'$ is substitutable for $H$ in $G$, then by letting $\mu^H_{G[H']} \triangleq \mu_{G[H'/H]}$ and $G' \triangleq G[H'/H]$, we have $\mu(\mathcal{R}_G) \subseteq \mathcal{R}_{G'}$ and $\mu(\mathcal{R}_{H'}) = \mathcal{R}_{H'}$.

**Proposition 18.** If $H'$ is substitutable for $H$ in $G$, then by letting $G' \triangleq G[H'/H]$, we have:
- $\mathcal{N}_{G'} \setminus \mathcal{N}_{H'} = \mathcal{N}_{G} \setminus \mathcal{N}_{H}$;
- $\mathcal{N}_{G'} \setminus \mathcal{N}_{H} = \mathcal{N}_{G} \setminus \mathcal{N}_{H'}$.

**Proof.** We have $\mathcal{N}_{G'} \setminus \mathcal{N}_{H'} = \{(N_G \setminus \mathcal{N}_{H'}) \cup \mathcal{N}_{H'} \} \setminus \mathcal{N}_{H'} = \{(N_G \setminus \mathcal{N}_{H'}) \setminus \mathcal{N}_{H'} \} \setminus \mathcal{N}_{H'}$ by Proposition 14. Consider a node $\alpha \in \mathcal{N}_G$; we show that $\alpha \in \mathcal{R}_G \setminus \mathcal{N}_{H}$ if and only if $\alpha \in \mathcal{R}_{G'} \setminus \mathcal{N}_{H'}$.

Observe that, with the notations of Definition 15, if $\alpha \in \mathcal{N}_{G'} \setminus \mathcal{N}_{H'}$, then necessarily $\mu(\alpha) = \sigma$, so that $\mu^{-1}(\sigma) = \alpha$ and $l_{G'}(\alpha) = l_G(\mu^{-1}(\alpha))$.

**Lemma 19.** Let $G$ be a DRL-graph, let $H$ be a DRL-subgraph of $G$ and let $H'$ be a DRL-graph substitutable for $H$ in $G$. Then, $G[H'/H]$ is a DRL-graph and $H' \preceq G[H'/H]$.

**Proof.** Observe that the function $\mu$ in Definition 15 is injective, since $\mathcal{R}_{H'}$ is repetition-free and no node in $\mathcal{N}_{G} \setminus \mathcal{N}_{H}$ may occur in $\mathcal{R}_{H'}$, because $\mathcal{N}_{G} \cap \mathcal{N}_{H'} \subseteq \mathcal{N}_{H}$ by hypothesis. We show that the domain of $\mu$ contains all nodes occurring either in $\mathcal{R}_{G}$ or in an edge $\mathcal{E}_{G'} \setminus \mathcal{E}_{H'}$. By Condition 5 of Definition 4, the nodes in $\mathcal{R}_{G}$ occur in either $\mathcal{R}_{H'}$ or $\mathcal{N}_{G} \setminus \mathcal{N}_{H}$.

The function $l_{G'}$ is well-defined: if $\alpha \in \mathcal{N}_{G'}$ and $\alpha = \mu(\alpha')$ for $\alpha' \in \mathcal{N}_{G} \setminus \mathcal{N}_{H}$, then we have $\alpha' \not\in \mathcal{N}_{H'}$ (because $\mathcal{N}_{G} \cap \mathcal{N}_{H'} \subseteq \mathcal{N}_{G}$), hence $\alpha' \neq \alpha$, and in this case, by definition of $\mu$, $\alpha$ occurs in $\mathcal{R}_{H'}$, hence may not occur in $\mathcal{N}_{H'}$, a contradiction. Thus it is straightforward to check that $G[H'/H]$ is a DRL-graph.

We prove that $H' \preceq G'$, with $G' \triangleq G[H'/H]$:

1. $\mathcal{N}_{H'} \subseteq \mathcal{N}_{G'} \subseteq (\mathcal{N}_{G} \setminus \mathcal{N}_{H}) \cup \mathcal{N}_{H'}$.
2. If $(\alpha \rightarrow \beta) \in \mathcal{E}_{H'}$, then $(\alpha \rightarrow \beta) \in \mathcal{E}_{G'} = \mu(\mathcal{E}_{G'} \setminus \mathcal{E}_{H'})$.
3. If $\alpha, \beta \in \mathcal{N}_{H'}$, then $(\alpha \rightarrow \beta) \in \mathcal{E}_{G'}$, by definition of $\mathcal{E}_{G'}$, $(\alpha \rightarrow \beta) \in \mathcal{E}_{H'}$ and the proof is completed, or $(\alpha \rightarrow \beta) \in \mathcal{E}_{G'} \setminus \mathcal{E}_{H'}$. In the latter case, we have $\alpha = \mu(\alpha'), \beta = \mu(\beta')$ with $(\alpha' \rightarrow \beta') \in \mathcal{E}_{G'}$ and $\mathcal{E}_{H'}$. We show that $\alpha' \not\in \mathcal{N}_{H'}$. If $\alpha = \alpha'$, then we have $\alpha \in \mathcal{N}_{H'} \cap \mathcal{N}_{G}$, thus $\alpha' \in \mathcal{N}_{H'}$ (since $H'$ is substitutable for $H$ in $G$). Otherwise, $\alpha \neq \alpha'$, hence by definition of $\mu$, $\alpha'$ occurs in $\mathcal{R}_{H'}$, thus $\alpha' \not\in \mathcal{N}_{H'}$. Similarly, $\beta' \not\in \mathcal{N}_{H'}$, and since $H \preceq G$ we deduce that $(\alpha' \rightarrow \beta') \in \mathcal{E}_{H'}$, contradicting the fact that $(\alpha' \rightarrow \beta') \in \mathcal{E}_{G'} \setminus \mathcal{E}_{H'}$.

4. Assume that $(\alpha \rightarrow \beta) \in \mathcal{E}_{G'}$, $\alpha \in \mathcal{N}_{H'}$ and $\beta \not\in \mathcal{N}_{H'}$. By definition of $\mathcal{E}_{G'}$, this entails that $(\alpha \rightarrow \beta) \in \mu(\mathcal{E}_{G'} \setminus \mathcal{E}_{H'})$ and $\alpha = \mu(\alpha'), \beta = \mu(\beta')$, for some $\alpha', \beta' \in \mathcal{N}_{G}$. If $\alpha = \alpha'$ then $\alpha \in \mathcal{N}_{H'} \cap \mathcal{N}_{G}$, and since $H'$ is substitutable for $H$ in $G$, we have $\alpha \in \mathcal{N}_{H'}$, thus, by definition of $\mu$, $\alpha$ occurs in $\mathcal{R}_{H}$ (since $\alpha \in \text{dom}(\mu)$). Otherwise, $\alpha \neq \alpha'$, and $\alpha$ occurs in $\mathcal{R}_{H'}$ by definition of $\mu$.

5. Let $\alpha \in \mathcal{N}_{H'}$ be a node occurring in $\mathcal{R}_{G'}$. We have $\alpha = \mu(\alpha')$ for some node occurring in $\mathcal{R}_{G'}$. By definition of $\mu$, either $\alpha$ occurs in $\mathcal{R}_{H'}$ and the proof is completed, or $\alpha = \alpha'$ with $\alpha \not\in \mathcal{N}_{H'}$, which contradicts the hypothesis that $H'$ is substitutable for $H$ in $G$.

6. By definition of $l_{G'}$, we have $l_{H'} = l_{G'}|_{\mathcal{N}_{H'}}$. □
3.2 Properties of the Replacement Operation

We establish some basic properties of the replacement operation, all of which are similar to their counterparts for the replacement operation on first-order terms.

**Proposition 20.** Let \( G \) be a DRL-graph and let \( H \subseteq G \). Then \( H \) is substitutable for \( H \) in \( G \) and \( G[H/H] = G \).

**Proof.** By definition, \( N_G \cap N_H \subseteq N_H \) thus \( H \) is substitutable for \( H \) in \( G \). The conditions of Definition 15 (with \( H = H' \)) entail that \( \mu \) is the identity on \( N_G \setminus N_H \) and that \( N_G \cap N_H = N_G \). \( \Re_G \subseteq \Re_H \subseteq (E_G \setminus \eu_H) \cup \eu_H = E_G \) and \( l_G = l_G \) (since \( l_H = l_G \mid_N \)).

**Proposition 21.** Let \( G, H \) and \( H' \) be DRL-graphs, with \( H \subseteq G \) and \( H' \) is substitutable for \( H \) in \( G \). Let \( \mu \) be an \( N \)-mapping with \( N_G \cap N_H \). Then \( \mu(H') \) is substitutable for \( \mu(H) \) in \( (G), \) and:

\[
\mu(G[H'/H]) = \mu(G)[\mu(H')/\mu(H)]
\]

**Proof.** We have \( \mathcal{N}_{\mu(G)} \cap \mathcal{N}_{\mu(H')} = \mu((N_G \setminus N_H) \cup N_H) = \mu(N_G \cap N_H) \) by injectivity of \( \mu \), thus \( \mathcal{N}_{\mu(G)} \cap \mathcal{N}_{\mu(H')} \subseteq \mu(N_G) = \mu(H) \) (since \( N_G \cap N_H \subseteq N_H \), as \( H' \) is substitutable for \( H \) in \( G \)). Consequently, \( \mu(H') \) is substitutable for \( \mu(H) \) in \( (G) \).

Let \( G' \models G[H'/H] \) and \( G'' \models \mu(G)[\mu(H')/\mu(H)] \). We show that \( G'' \models \mu(G) \).

\[
\mu' = \{ (a, b) \in A' \mid \exists \alpha \in \mathcal{N}_{\mu(G)} \cap \mathcal{N}_{\mu(H)} \}
\]

Observe that \( \mu'' = \mu \circ \mu' \circ \mu^{-1} \). We have:

\[
\mathcal{N}_{\mu(G')} = \mathcal{N}_{\mu(G)} = \mu((N_G \setminus N_H) \cup N_H)
\]

\[
\mathcal{N}_{\mu(G)} \subseteq \mathcal{N}_{\mu(H)} \cup \mu(N_H) \cup \mu(N_H)
\]

\[
\mathcal{N}_{\mu(H')} \cup \mu(N_H) \cup \mu(N_H)
\]

\[
\text{by injectivity of } \mu
\]

\[
\mathcal{N}_{\mu(G')} \subseteq \mathcal{N}_{\mu(G)} \cup \mathcal{N}_{\mu(H)} \cup \mathcal{N}_{\mu(H')}
\]

Further:

\[
\mathcal{E}_{\mu(G')} = \mu''(\mathcal{E}_{\mu(G)} \setminus \mathcal{E}_{\mu(H)} \cup \mathcal{E}_{\mu(H)})
\]

\[
= \mathcal{E}_{\mu''(\mu(G)) \setminus \mu''(\mu(H)) \cup \mu''(\mu(H))}
\]

\[
= \mathcal{E}_{\mu''(\mu(G)) \setminus \mu''(\mu(H)) \cup \mu''(\mu(H))}
\]

Hence \( \mathcal{E}_{\mu(G')} = \mathcal{E}_{\mu(G)} \).

**Proposition 22.** Let \( G, H \) and \( H' \) be DRL-graphs, where \( H \subseteq G \) and \( H' \) is substitutable for \( H \) in \( G \). Let \( \sigma \) be a substitution with domain \( \mathcal{V}(G) \cup \mathcal{V}(H') \). Then \( \sigma(H') \) is substitutable for \( \sigma(H) \) in \( \sigma(G) \), and:

\[
\sigma(G[H'/H]) = \sigma(G)[\sigma(H')/\sigma(H)]
\]

**Proof.** Since a substitution only affects the labels of the DRL-graphs, the first property is immediate, and we only have to prove that, for every node \( \alpha \in (N_G \setminus N_H) \cup N_H \), \( l_{\sigma(G[H'/H])}(\sigma(H')) \equiv l_{\sigma(G[H'/H])}(\sigma(H)) \). By definition, we have \( l_{\sigma(G[H'/H])}(\sigma(H')) = l_{\sigma(G[H'/H])}(\sigma(H)) \).

By definition, we distinguish several cases.

If \( \alpha \in N_H \), then \( l_{\sigma(G[H'/H])}(\sigma(H')) = l_{\sigma(H')} \). Moreover, we also have \( l_{\sigma(G[H'/H])}(\sigma(H')) \equiv l_{\sigma(G[H'/H])}(\sigma(H)) \) and the proof is completed.

Finally, if \( \alpha \notin N_H \), then by definition \( l_{\sigma(G[H'/H])}(\sigma(H')) = l_{\sigma(H')} \), and \( \alpha \notin N_{\sigma(H')} \). Thus \( l_{\sigma(G[H'/H])}(\sigma(H')) \equiv l_{\sigma(H')} \).

The next lemma shows that renaming the nodes of some subgraph does not affect the graphs obtained by replacing this subgraph.

**Lemma 23.** Let \( G, H \) and \( H' \) be DRL-graphs, with \( H \subseteq G \) and \( H' \) is substitutable for \( H \) in \( G \). Let \( \mu \) be an \( N \)-mapping with \( N_G \) such that \( \mu(\alpha) = \alpha \) if \( \alpha \in N_G \setminus N_H \). Then:

\[
\mu(G[H'/H]) = G[H'/H]
\]

**Proof.** Let \( G' \equiv G[H'/H], G'' \equiv \mu(G)[H'/H] \), \( \Re_G \equiv \{ (a, b) \mid \alpha \in \mathcal{N}_{\mu(G)} \} \), and consider the following \( N \)-mappings:

\[
\mu_1 \equiv \{ (a, b) \mid \alpha \in \mathcal{N}_{\mu(G)} \}
\]

\[
\mu_2 \equiv \{ (a, b) \mid \alpha \in \mathcal{N}_{\mu(G)} \}
\]

By injectivity of \( \mu \), we have \( \mathcal{N}_{\mu(G)} \cap \mathcal{N}_{\mu(H)} = \mathcal{N}_{\mu(G)} \cap \mathcal{N}_{\mu(H)} \). Thus \( \mu_1 \equiv \mathcal{N}_{\mu(G)} \cap \mathcal{N}_{\mu(H)} \).

Finally, if \( \alpha \notin N_G \), then again by Definition 15, we have \( l_{\mu(G[H'/H])}(\alpha) = l_{\mu(H)}(\alpha) \).

That \( \alpha' \notin N_H \), and that there is no \( \alpha'' \) such that \( \alpha = \mu(\alpha'') \). With \( \alpha'' \in N_G \setminus N_H \). Thus \( l_{\mu(G[H'/H])}(\alpha) \) and \( l_{\mu(G[H'/H])}(\alpha') \) are both undefined.

A similar property holds for substitutions.

**Proposition 24.** Let \( G, H \) and \( H' \) be DRL-graphs, where \( H \subseteq G \) and \( H' \) is substitutable for \( H \) in \( G \). Let \( \sigma \) be a substitution with domain \( \mathcal{V}(G) \cup \mathcal{V}(H') \). Then \( \sigma(H') \) is substitutable for \( \sigma(H) \) in \( \sigma(G) \), and:

\[
\sigma(G[H'/H]) = \sigma(G)[\sigma(H')/\sigma(H)]
\]

Proof. Since a substitution only affects the labels of the DRL-graphs, the first property is immediate, and we only have to prove that, for every node \( \alpha \in (N_G \setminus N_H) \cup N_H \), \( l_{\sigma(G[H'/H])}(\sigma(H')) \equiv l_{\sigma(G[H'/H])}(\sigma(H)) \). By definition, we have \( l_{\sigma(G[H'/H])}(\sigma(H')) = l_{\sigma(G[H'/H])}(\sigma(H)) \).

By definition, we distinguish several cases.

If \( \alpha \in N_H \), then \( l_{\sigma(G[H'/H])}(\sigma(H')) = l_{\sigma(H')} \). Moreover, we also have \( l_{\sigma(G[H'/H])}(\sigma(H')) \equiv l_{\sigma(G[H'/H])}(\sigma(H)) \) and the proof is completed.

Finally, if \( \alpha \notin N_H \), then by definition \( l_{\sigma(G[H'/H])}(\sigma(H')) = l_{\sigma(H')} \), and \( \alpha \notin N_{\sigma(H')} \). Thus \( l_{\sigma(G[H'/H])}(\sigma(H')) \equiv l_{\sigma(H')} \).

The next lemma shows that renaming the nodes of some subgraph does not affect the graphs obtained by replacing this subgraph.
This entails that the replacement of a subgraph by an isomorphic subgraph preserves isomorphism:

**Corollary 24.** Let G, H, H', H'' be DRL-graphs, where H ≤ G, H' and H'' are substitutable for H in G and H' ∼ H''. Then:

\[ G[H'_1/H] \sim G[H''_1/H] \]

**Proof.** Since H'_1 ∼ H''_1, there exists an N-mapping μ of domain N_H' such that H'_1 = μ(H''_1). Consider the extension μ' of μ to N_G, such that: μ'(α) = α if α ∈ N_G \ N_H and all the nodes in N_H \ N_H' are mapped to pairwise distinct nodes not occurring in G or H'_1. Observe that μ is well-defined, since N_G ∩ N_H' ⊆ N_H and H' is substitutable for H in G. We show that μ' is injective. Let α, α' such that μ'(α) = μ'(α'). If α, α' ∈ N_H, then μ'(α) = μ(α) and μ'(α') = μ(α') hence α = α' as μ is injective. If α, α' ∈ N_G \ N_H then μ'(α) = α and μ'(α') = α' hence α = α'. If α, α' ∈ N_H \ N_H' then we have α = α' by definition of μ'. If α ∈ N_G \ N_H and α' ∈ N_H \ N_H', then by definition of μ', μ'(α) ∈ N_G \ N_H' and μ'(α') ∈ N_G \ N_H', which contradicts the fact that μ'(α) = μ'(α'). The only remaining case is (by symmetry): α ∈ N_H and α' ∈ N_G \ N_H. Then μ'(α) = μ(α) ∈ N_H' and μ'(α') = α' ∈ N_G, hence, since H'_1 is substitutable for H in G, we must have α' ∈ N_H, contradicting the fact that α' ∈ N_G \ N_H.

We get, by Proposition 21, since μ coincides with μ on N_H:

\[ μ'(G[H'_1/H]) = μ'(G[H'_2/H]) = μ'(G[H'/H]) \]

By definition of μ', μ'(α) = α holds for any node α ∈ N_G \ N_H, hence, by Lemma 23, we deduce:

\[ μ'(G[H'_1/H]) = G[H'_1/H] \]

Therefore, G[H'_1/H] ∼ G[H'_2/H].

The next lemma forms a sort of transitivity of the replacement operation: replacing H by I and then I by I' is equivalent to replacing H by I'.

**Lemma 25.** Let G, H, I, I', where H ≤ G, I is substitutable for H in G and I' is substitutable for I in G[I/H]. Then I' is substitutable for H in G and G[I'/H] = G[I/H].

**Proof.** Let G' = G[I/H]. We show that I' is substitutable for H in G, i.e., that N_G \ N_H ⊆ N_H. By definition, N_{G'} = (N_G \ N_H) ∪ N_I, we show that (N_G \ N_H) ∩ N_I = 0. We have

\[ (N_G \ N_H) ∩ N_I ⊆ [(N_G \ N_H) \cap N_I] \cup N_I = N_G \cap N_I \subseteq N_I \]

because I' is substitutable for I in G' by hypothesis. We deduce that (N_G \ N_H) ∩ N_I = 0. But (N_G \ N_H) ∩ N_I = 0 by Proposition 14, hence the result.

We show that G'[I'/I] = G[I/H]. We define μ' = H → I, μ' = H → I and μ = H → I. Note that μ'' = μ' ∘ μ. Indeed, by definition we have dom(μ) = dom(μ'). Furthermore, for every α ∈ dom(μ), if α ∈ N_G \ N_H then μ(α) = μ'(α), and if α ∈ N_H (since I is substitutable for H in G), hence μ''(α) = α. If α ∈ dom(μ) and α ∈ N_G \ N_H, then α = α, for some i = 1, . . . , n, with R' = (a_1, . . . , a_n). By definition, μ(α_i) = (a_i', . . . , a''_n) and μ'(α_i) = (a_i', . . . , a''_n) ∩ R = (a_1', . . . , a_n').

(1) We have

\[ N_{G'[I'/I]} = (N_G \ N_I) \cup N_I = [[N_G \ N_I] \cup N_I] \cup N_I = (N_G \ N_I) \cup N_I = N_G \cap N_I \]

where the second to last equality is obtained using the fact that, by Proposition 14, (N_G \ N_I) ∩ N_I = 0.

(2) We have R_{G'[I'/I]} = μ'(R_{G'}) = μ'(μ(R_G)) = μ''(R_G).

(3) We show that μ(G'[E/H] \cap E) = 0. Suppose for a contradiction that (α' ≈ β') ∈ μ(G'[E/H] \cap E). Then α', β' ∈ N_H, and there exist nodes α, β such that α' = μ(α) and β' = μ(β), with (α ≈ β) ∈ E_G \ E_H. Since (α ≈ β) ∉ E_H, necessarily, (α, β) ∈ N_I, by Definition 4 (3). Assume w.l.o.g. that α ∈ N_G \ N_H. Then by definition α' = α ∈ N_I. But this is impossible because (N_G \ N_I) ∩ N_I = 0 by Proposition 14. We deduce that

\[ E_G[I'/I] = μ'(E_G \ E_I) \cup E_I = μ'(μ(E_G \ E_I) \cup E_I) \cup E_I = μ'(E_G \ E_H \cup E_I) \cup E_I = μ''(E_G \ E_I) \cup E_I = E_G[I'/I]. \]

where the second to last equality is obtained using the fact that μ(E_G \ E_H) \cap E_I = 0 and that μ'' = μ' ∘ μ.

(4) Let α ∈ N_{G'[I'/I]}. If α ∈ N_{G'[I'/I]} \ N_I, then by Proposition 18 we have α ∈ N_{G'[I'/I]} \ N_I if and only if α ∈ N_{G'[I'/I]} \ N_I if and only if α ∈ N_{G'[I'/I]} \ N_H if and only if α ∈ N_{G'[I'/I]} \ N_I. Thus l_{G'[I'/I]}(α) = l_{G'[I'/I]}(α).

• If α ∈ N_G, then l_{G'[I'/I]}(α) = l_{G'[I'/I]}(α).

• If β = μ''(α) ∈ N_{G'[I'/I]} \ N_R then by definition of μ, we must have μ''(β) = β ∈ N_R. We cannot have β ∈ N_R, because otherwise we would have β ∈ R_{G'[I'/I]} \ N_R, which is substitutable for H in G. Since μ'' = μ' ∘ μ, we deduce that μ''(β) ∈ N_{G'[I'/I]} \ N_R. Since μ''(β) = μ''(β), and l_{G'[I'/I]}(β) = l_{G'[I'/I]}(β), and l_{G'[I'/I]}(β) = l_{G'[I'/I]}(β).

By Proposition 18 we have α ∈ N_{G'[I'/I]} \ N_I if and only if α ∈ N_{G'[I'/I]} \ N_H. This proves that l_{G'[I'/I]}(α) = l_{G'[I'/I]}(α).

In particular, replacing H by H' and then H' by H is equivalent to doing nothing:

**Corollary 26.** Assume H' is substitutable for H in G and let G' = G[H'/H']. Then H is substitutable for H' in G' and G'[H'/H'] = G.

**Proof.** We have N_{G'} ∩ N_H = [(N_G \ N_H) ∪ N_{H'}] ∩ N_H = N_{H'} ∩ N_H ⊆ N_{H'}, which show that H is substitutable for H' in G'. By Lemma 25 and Proposition 20, we deduce that G[H'/H'] = G[H'/H'] = G.

The next lemma states that the replacement operation does not affect the subgraph that are disjoint from the replaced DRL-graph.
**Lemma 27.** Consider the graphs $G, H, H', I$, and assume that $H \preceq G$, $I \preceq G$ and $H'$ is substitutable for $H$ in $G$. If $N_{H} \cap N_{I} = \emptyset$ then $I \preceq G[H'/H]$. 

Proof. Let $G' \overset{\text{def}}{=} G[H'/H]$ and $\mu \overset{\text{def}}{=} \mu_{G}^{H \rightarrow H'}$. Note that since $H'$ is substitutable for $H$ in $G$, we have $N_{H'} \cap N_{I} \subseteq N_{H'} \cap N_{G} \subseteq N_{H} \cap N_{I} = \emptyset$, and that since $N_{H} \cap N_{I} = \emptyset$, if $\alpha \in N_{I}$ then $\mu(\alpha) = \alpha$. We verify that $I \preceq G'$:

1. Since $I \preceq G$, by definition $N_{I} \subseteq N_{G}$. By hypothesis $N_{H} \cap N_{I} = \emptyset$, thus $N_{I} \subseteq N_{G} \cap N_{H} \subseteq N_{G}$.

2. If $(\alpha \rightarrow \beta) \in E_{I}$ then $(\alpha \rightarrow \beta) \in E_{G}$, because $I \preceq G$. Since $N_{H} \cap N_{I} = \emptyset$ by hypothesis, $(\alpha \rightarrow \beta) \notin E_{H}$. Thus $(\alpha \rightarrow \beta) \in (E_{G} \setminus E_{H}) \cup E_{H'}$ and since $\mu(\alpha) = \alpha$ and $\mu(\beta) = \beta$, we deduce that $(\alpha \rightarrow \beta) \in (E_{G} \setminus E_{H}) \cup E_{H'} = E_{G'}$.

3. Let $\alpha, \beta \in N_{I}$, and assume that $(\alpha \rightarrow \beta) \in E_{G'} = \mu(E_{G} \setminus E_{H}) \subseteq E_{H'} \cup E_{H}$.

4. Assume that $(\alpha \rightarrow \beta) \in E_{G'} = \mu(E_{G} \setminus E_{H}) \subseteq E_{H'} \cup E_{H}$, where $\alpha \in N_{I}$ and $\beta \notin N_{I}$. Then $\mu(\alpha) = \alpha$ and since $N_{\text{nar}GH} \cap N_{I} = \emptyset$, we cannot have $(\alpha \rightarrow \beta) \in E_{H'}$. Let $\beta' \overset{\text{def}}{=} \mu^{-1}(\beta)$, note that we cannot have $\beta' \in N_{H}$ because otherwise we would have $\mu(\beta') = \beta' \notin N_{I}$. Thus, $(\alpha \rightarrow \beta') \in E_{G} \setminus E_{H} \subseteq E_{G}$, and since $I \preceq G$, we deduce that $\alpha \in R_{I}$.

5. Assume $\alpha \in R_{G'} \cap N_{I} = \mu(R_{G}) \cap N_{I}$. Then since $\mu(\alpha) = \alpha$, we have $\alpha \in R_{G} \cap N_{I}$, hence $\alpha \in R_{I}$ because $I \preceq G$.

6. Consider $N_{I} \subseteq N_{H}$ and by Proposition 18 we have $\alpha \in N_{G} \cap N_{H}$. By definition, $l_{G'}(\alpha) = l_{G}(\alpha) = l_{I}(\alpha)$, because $I \preceq G$.

Further, the result of the replacement of two disjoint subgraphs does not depend on the order in which the replacement operations are performed:

**Lemma 28.** Let $G, H, H', I$ be DRL-graphs for $i = 1, 2$, where $H_{i} \subseteq G$ and $H'_{i}$ is substitutable for $H_{i}$ in $G$. If $N_{H_{i}} \cap N_{H'}_{i} = \emptyset$ then, for $i, j \in \{1, 2\}$ with $i \neq j$, $H'_{i}$ is substitutable for $H_{j}$ in $G[H'_{i}/H_{j}]$ and $(G[H'_{i}/H_{j}])[H'_{j}/H_{i}] = (G[H'_{i}/H_{j}])[H'_{j}/H_{i}]$.

Proof. By Corollary 27, $H_{i}$ is substitutable for $H_{j}$ in $G[H'_{i}/H_{j}]$; the proof that $H'_{j}$ is substitutable for $H_{i}$ in $G[H'_{i}/H_{j}]$ is symmetric. Let $\alpha \in N_{H_{i}} \cap N_{H'_{j}} \subseteq N_{H'_{j}} \cap N_{H'_{i}} = \emptyset$, then $\mu(\alpha) = \alpha$ and $\mu_{G}^{H \rightarrow H'}(\alpha) = \mu(\alpha)$, hence the result.

Finally, replacements can be embedded:
Proof. We have $N_I \cap N_F \subseteq N_G \cap N_F \subseteq N_F$ (since $I'$ is substitutable for $I$ in $G$). Therefore, $I'$ is substitutable for $I$ in $H$. Similarly, $N_G \cap N_H \cup I [I' / I] \subseteq N_H \cup (N_I \cap N_F) = N_H \cup (N_G \cap N_F) \subseteq N_H \cup N_F = N_H$, hence $H[I' / I]$ is substitutable for $H$ in $G$.

Let $G' \equiv G[I' / I]$, $H' \equiv H[I' / I]$ and $G'' \equiv G[H' / H]$; we show that $G'' = G'''.

Let $\mu_1 \equiv \mu_1^{I' \rightarrow I}$, $\mu_2 \equiv \mu_2^{I' \rightarrow I}$ and $\mu_3 \equiv \mu_3^{H' \rightarrow H}$. Note that by definition, $\mu_3$ is the restriction of $\mu_3$ to the nodes in $(N_H \cap N_F) \cup R_I = N_H \setminus N'_H$. We show that for all $\alpha \in N_G \setminus N'_H = (N_G \setminus N_H) \cap R_H$, we have $\mu_3(\alpha) = \mu_2(\alpha)$. If $\alpha \in N_G \setminus N_H$ then since $N_I \subseteq N_F$, we have $\mu_3(\alpha) = \mu_2(\alpha) = \alpha$. Otherwise, $\alpha \in R_H$ and by Proposition 17, $\mu_3(\alpha) = \mu_2(\alpha)$. Since $R_H \subseteq N_H \setminus N'_H$ by Proposition 6, we deduce that $\mu_3(\alpha) = \mu_2(\alpha) = \mu_1(\alpha)$.

• By Definition 15 (1), $N_G' = (N_G \setminus N_H) \cup N_F = (N_G \setminus N_H) \cup (N_H \setminus N_I) \cup N_F = N_G \setminus N'_H$, $\mu_3 \equiv \mu_3(N_G' \setminus N_H)$. By Definition 15 (2), $R_G' = \mu_3(R_G)$ and $\mathbb{R}_G' = \mu_1(R_G)$. Since $R_G \subseteq N_G \setminus N_H$ by Proposition 6, we have $\mu_3(\alpha) = \mu_2(\alpha) = \mu_1(\alpha)$.

Consider an edge $(\alpha \leftrightarrow \beta)$ in $E_H \setminus E_I$. Then since $I \leq H$, we cannot have $(\alpha, \beta) \notin N_I$ and if $\alpha \in N_I$ then necessarily $\alpha \in R_I$. This shows that $(\alpha, \beta) \notin N_G \setminus N'_H$ and therefore, $\mu_1(\alpha) = \mu_2(\alpha) = \mu_3(\alpha)$. The fact that $\mu_3(E_G \setminus E_H) = \mu_1(E_G \setminus E_H)$ is proved in a similar way. We have

$$
E_G' = \mu_3(E_G \setminus E_H) \cup E_I = \mu_2(\mu_3(E_G \setminus E_H) \cup E_I) = \mu_1(\mu_2(\mu_3(E_G \setminus E_H) \cup E_I)) = \mu_1(E_G \setminus E_H) \cup E_I = E_G.
$$

Consider a node $\alpha \in N_G'$. Since $N_G' = N_G$, $\mathbb{R}_G' = \mathbb{R}_G$, we also have $\alpha \in N_G'$. First assume that $\alpha \notin N_H$, so that $I_G'(\alpha) = I_G(\alpha)$.

- If $\alpha \in N'_G \setminus N_H$, then by definition $I_G'(\alpha) = I_G(\alpha)$.

- If $\alpha \in N'_G \cap N_H$, then by hypothesis and $H \leq G$, we have $I_G'(\alpha) = I_H(\alpha) = I_G(\alpha)$.

- If $\alpha \notin N_H$, then $\mu_1^{-1}(\alpha) = \mu_2^{-1}(\alpha) = \alpha$, hence the result.

Assume that $\alpha \in N_H$. So that $I_G'(\alpha) = I_G(\alpha)$.

- We cannot have $\alpha \in N'_G \setminus N_H$ because $N'_G \subseteq N_H$.

- If $\alpha \in N'_G$, then $I_G'(\alpha) = I_H(\alpha) = I_G(\alpha)$ because $\alpha \in N'_H$.

- We cannot have $\mu_3^{-1}(\alpha) \in N'_G \cap R_H$ because otherwise we would have $\alpha \in \mu_3(R_H) = R_H$ and since $I' \leq H'$, necessarily $\alpha \in R_H$.

Now assume that $\alpha \in N'_G \cap R_H$, i.e., $\mu_1^{-1}(\alpha) \in N'_G \cap R_H$, so that $I_G'(\alpha) = I_G(\mu_1^{-1}(\alpha))$.

- We cannot have $\alpha \in N'_G \setminus N_H$ because $R_H \subseteq N_H$.

- If $\alpha \in N'_H$, then we have $I_G'(\alpha) = I_H(\mu_1^{-1}(\alpha)) = I_H(\mu_1^{-1}(\alpha))$.

- If $\mu_3^{-1}(\alpha) \in N'_G \cap R_H \subseteq N'_G \setminus N_H$ then we have $I_G'(\alpha) = I_G(\mu_3^{-1}(\alpha))$ because $\mu_1$ and $\mu_3$ coincide on $N'_G \setminus N'_H$.

□

3.3 Graph Merging

We introduce the notion of a merge of two graphs. Intuitively, the merge of graphs $G_1$ and $G_2$ will be any graph $G$ containing all nodes and edges occurring in either $G_1$ or $G_2$, possibly along with some additional edges.

DEFINITION 30. Two DRL-graphs $G_1$ and $G_2$ are label-compatible iff the set of pairs $\{(\alpha G_1(\beta), \beta G_2(\beta)) : \beta \in N_{G_1} \cap N_{G_2}\}$ admits an mgu.

Given two label-compatible DRL-graphs $G_1$, $G_2$ with mgu $\sigma$ and a set of edges $E$ of the form $(\alpha \leftrightarrow \beta)$ such that $\alpha \in R_{G_1} \cap N_{G_2}$ and $\beta \in R_{G_2} \cap N_{G_1}$, the E-merge of $G_1$ and $G_2$ is the graph $G$ defined as follows:

- $N_G = N_{G_1} \cup N_{G_2}$;
- $R_G = (R_{G_1} \cup R_{G_2}) \setminus (N_{G_1} \setminus N_{G_2})$ (the order of the nodes in $R_G$ is chosen arbitrarily);
- $E_G = E_{G_1} \cup E_{G_2} \cup E$;
- for every node $\alpha \in N_{G_1}$ with $i = 1, 2$, $l_G(\alpha) = l_{G_i}(\alpha)$.

A merge of two label-compatible DRL-graphs $G_1$, $G_2$ is a DRL-graph $G$ that is an E-merge of $G_1$ and $G_2$, for some set $E$ of edges of the form $(\alpha \leftrightarrow \beta)$ such that $\alpha \in R_{G_1} \cap N_{G_2}$ and $\beta \in R_{G_2} \cap N_{G_1}$.

It is easy to check that merges can be constructed using usual unification algorithms.

PROPOSITION 31. Assume $H, H' \leq G$ and consider the set of edges $E = \{(\alpha \leftrightarrow \beta) : \alpha \in R_H \setminus N_H, \beta \in R_{H'} \setminus N_H, (\alpha \leftrightarrow \beta) \in E_G\}$. Then $H$ and $H'$ are label-compatible and if $G' = G$ is the E-merge of $H$ and $H'$, then $G' \leq G$.

Proof. It is straightforward to verify that $H$ and $H'$ are label-compatible, with the empty mgu. We show that $G' \leq G$, using the fact that, by hypothesis, $H, H' \leq G$.

1. By Definition 30, we have $N_G = N_H \cup N_{H'} \subseteq N_G$.
2. By the edges in $E$ are also in $E_G$, it is thus straightforward to verify that $E_G \subseteq E_G$.
3. Assume that $(\alpha \to \beta) \in E_G$, where $\alpha, \beta \in N_G$. If $\alpha, \beta \in N_H$ or $\alpha \in N_H$ and $\beta \in N_{H'}$, then $(\alpha \to \beta) \in E_G$, since $H' \leq G$. Suppose that $\alpha \in N_H \setminus N_F$, $\beta \in N_{H'} \setminus N_H$. By Definition 4 (4), necessarily $\alpha \in R_H$ and $\beta \in R_{H'}$, hence $(\alpha \to \beta) \in E \subseteq E_G$. The case where $\alpha \in N_H \setminus N_{H'}$ and $\beta \in N_{H'} \setminus N_F$ is proved in a similar way.
4. Consider $(\alpha \leftrightarrow \beta) \in E_G$, where $\alpha \in N_F$ and $\beta \notin N_G$. Assume $\alpha \in N_H$, the case where $\alpha \in N_{H'}$ is proved in a similar way. Since $\beta \notin N_G$, we have $\beta \notin N_H$, hence $\alpha \in R_H$ by Definition 4 (4). It cannot be the case that $\alpha \in N_{H'}$, because otherwise we would have $\alpha \in N_{H'}$ and $\beta \notin N_{H'}$, so that $\alpha \in R_{H'}$, a contradiction. We conclude that $\alpha \in R_H$.
5. Consider a node $\alpha \in R_G \cap N_{G_2}$ and suppose w.l.o.g. that $\alpha \in N_H$. Since $H \leq G$, by Definition 4 (5) necessarily $\alpha \in R_H$. If $\alpha \in N_H$, then $\alpha \in R_H$, we deduce that $\alpha \in R_H \setminus (N_{G_1} \setminus N_{G_2}) \subseteq R_G$.
6. By definition (since the considered substitution $\sigma$ is empty) $I_G'$ is the restriction of $I_G$ to $N_{G_1}$. 

□
3.4 Restricting the Class of Graphs: C-Relations

Depending on the intended application, it is sometimes necessary to further restrict the considered class of graphs. For instance, if graphs are intended to model standard circuits obtained by composing gates associated with inputs and outputs, then it might be necessary to dismiss cyclic graphs, which have no obvious semantics in this particular context. We shall thus consider an arbitrary but fixed class of DRL-graphs C, which is assumed to be closed under isomorphism. The class C should be provided by a user, depending on the considered application (an example is provided in Section 8 to model possibly cyclic circuits).

To perform equational reasoning, we need to ensure that the replacement of a subgraph within a DRL-graph in C results in a DRL-graph that is still in C. It is clear that this property will not hold in general, even if the considered subgraph is also in C. For instance, the replacement of a subgraph in a non-cyclic DRL-graph may create new paths in the DRL-graph, which may create a cycle. To overcome this issue, we shall consider a restriction of the subgraph relation, denoting the particular subgraphs on which replacement is allowed. This relation is called a C-relation and will be denoted by $\subseteq_C$. In the following we only state the abstract properties that must be fulfilled by the relation $\subseteq_C$, its actual definition will depend on the considered application and on the class C. In most cases, $\subseteq_C$ will coincide with $\subseteq$ on graphs in C (this is the case for the class defined in Definition 62). However, we prefer, for the sake of generality, not to make this assumption.

**Definition 32.** Let C be a set of DRL-graphs, such that $G \in C \land G \sim G' \Rightarrow G' \in C$. A C-relation $\subseteq_C$ is a reflexive and transitive binary relation included in $\subseteq$ and satisfying the following properties:

1. If $H \subseteq_C G$ then $G \in C$ and $H \in C$.
2. If $H \subseteq_C G$, $H' \in C$ and $H'$ is substitutable for $H$ in $G$ then $H' \subseteq_G [H'/H]$ (thus $[H'/H] \in C$).
3. If $H \subseteq_C G$, then $\mu(H) \subseteq_C \mu(G)$, for every N-mapping $\mu$.
4. If $H \subseteq_C G$, $I \subseteq_C G$, $H$ and $I$ are disjoint, $I' \in C$ and $I'$ is substitutable for $I$ in $H$, then $H \subseteq_C G[I'/I]$.
5. If $G_i \subseteq_C G$, for $i = 1, 2$, then there exists a merge $G' \subseteq_G G_1$ and $G_2$, such that $G' \subseteq_C G$.
6. If $G \subseteq_C H$ then for every substitution $\sigma$, $\sigma(G) \subseteq_C \sigma(H)$.

In the following we consider a fixed set of DRL-graphs C, closed under isomorphism, and a fixed C-relation $\subseteq_C$. Note that in particular, if C is closed under merging, then the intersection of $\subseteq$ with $\subseteq_C$ is always a C-relation.

3.5 Graph Rewrite Systems

In what follows, we define the notion of a graph rewrite system that is based on the replacement operation. Note that the rewrite relation is parameterized by $\subseteq_C$, as the replacement of a subgraph $H$ within $G$ is allowed only if $H \subseteq_C G$.

**Definition 33.** A DRL-graph rewrite rule is a pair written $G \rightarrow H$, where $G$ and $H$ are DRL-graphs such that $R_G$ and $R_H$ are root-compatible. A DRL-graph rewrite system is a set of DRL-graph rewrite rules.

**Definition 34.** If $R$ is a DRL-graph rewrite system then we write $G \rightarrow_R G'$ if there exists a rule $H \rightarrow H'$ in $R$, a substitution $\sigma$ and an N-mapping $\mu$ of domain $N_H \cup N_{H'}$ such that the following conditions hold:

- $\sigma(\mu(H)) \subseteq_C G$.
- $\sigma(\mu(H'))$ is substitutable for $\sigma(\mu(H))$ in $G$, and
- $G' \sim G[\sigma(\mu(H'))/\sigma(\mu(H))]$.

We denote by $\rightarrow_R^+$ the transitive closure of $\rightarrow_R$. We write $G \rightarrow^k_R G'$ if there exists a sequence of DRL-graphs $H_i$ (for $i = 1, \ldots, k$) with $H_1 \sim G$, $H_k \sim G'$ and for every $i \in \{1, k-1\}$, $H_i \rightarrow_R H_{i+1}$. By a slight abuse of notation, we also denote by $\rightarrow_R^+$ the least relation containing $\rightarrow_R^*$ and $\sim$.

Note that $\rightarrow_R^*$ is the reflexive and transitive closure of $\rightarrow_R$, if the latter is viewed as a relation on equivalence classes of DRL-graphs w.r.t. $\sim$.

The next proposition and lemma state that $\rightarrow_R$ is compatible with isomorphism and graph embedding.

**Proposition 35.** If $G \rightarrow_R G'$ and $G'' \sim G$, then $G'' \rightarrow_R G'$.

**Proof.** By definition of the relation $\rightarrow_R$, there exists a rule $H \rightarrow H' \in R$, a substitution $\sigma$ and an N-mapping $\mu$ of domain $N_H \cup N_{H'}$ such that $\sigma(\mu(H)) \subseteq_C G$, $\sigma(\mu(H'))$ is substitutable for $\sigma(\mu(H))$ in G, and $G' \sim G[\sigma(\mu(H'))/\sigma(\mu(H))]$. Since $G'' \sim G$, there exists an N-mapping $\mu'$ such that $G'' = \mu'(G)$. By Definition 32 (3), we have $\mu'(\sigma(\mu(H))) \subseteq_C \mu'(G)$. By Proposition 21, $\mu'(\sigma(\mu(H'))) = \mu'(\sigma(\mu(H)))$ is substitutable for $\mu'(\sigma(\mu(H)))$ in $\mu'(G)$, and $\mu'(G)[\sigma(\mu(H'))/\sigma(\mu(H))] = \mu'(G)[\sigma(\mu(H')])/\mu'(\sigma(\mu(H)))] = G''[\sigma(\mu(H'))/\sigma(\mu(H))]$. Thus $G' \sim G''[\mu'(\sigma(\mu(H')))/\mu'(\sigma(\mu(H)))]$. Since $\mu'(\sigma(\mu(H))) = \mu'(\sigma(\mu(H'))) = \sigma(\mu'(\mu(H'))) = \sigma(\mu(\mu(H'))) = \sigma(\mu(\mu(H')))$, by Definition 34 we deduce that $G'' \rightarrow_R G'$.

**Lemma 36.** Let $R$ be a DRL-graph rewrite system. Let $G$ be a DRL-graph and $I$ be a DRL-subgraph of $G$, with $I \subseteq_C G$. If $I \rightarrow_R^k I'$ for some $k \geq 0$ and $I'$ is substitutable for $I$ in $G$, then $G \rightarrow_R^k[G[I'/I]]$.

**Proof.** The proof is by induction on $k$. If $k = 0$, then the proof is an immediate consequence of Proposition 20. Assume that $k > 0$. By Definition 33, there exist a rule $H \rightarrow H' \in R$, a substitution $\sigma$ and an N-mapping $\mu$ of domain $N_H \cup N_{H'}$ such that $\sigma(\mu(H)) \subseteq_C I$, $\sigma(\mu(H'))$ is substitutable for $\sigma(\mu(H))$ in $I$, and $I'' \sim I[\sigma(\mu(H'))/\sigma(\mu(H))]$ with $I'' \rightarrow R^k I'$. We assume w.l.o.g. that:

- The N-mapping $\mu$ is such that $N_H[\sigma(\mu(H'))] \cap N_G \subseteq N_I$, so that the graph $I[\sigma(\mu(H'))/\sigma(\mu(H))]$ is substitutable for $I$ in $G$ (such an N-mapping is guaranteed to exists because $N_G \subseteq N_I$);
- $N_{H'} \cap N_G = \emptyset$, so that $I''$ is substitutable for $I$ in $G$.

By Corollary 24 we have $G[I''/I] \sim G[I[\sigma(\mu(H'))]/\sigma(\mu(H))]/I]$, and by Lemma 29,

$$G[\sigma(\mu(H'))/\sigma(\mu(H))] = G[I[\sigma(\mu(H'))]/\sigma(\mu(H))]/I].$$

We deduce that $G[I''/I] \sim G[I[\sigma(\mu(H'))]/\sigma(\mu(H))].$

By Definition 32, $\subseteq_C$ is transitive and since we have $I \subseteq_C G$ and $\sigma(\mu(H)) \subseteq_C I$, we deduce that $\sigma(\mu(H)) \subseteq_C G$, which proves that $G \rightarrow_R G[I''/I]$. By Definition 32 (2) we have $I'' \subseteq_C G$ and...
by the induction hypothesis, $G[I''/I] \rightarrow_{R}^{k-1} G[I'/I]$. Therefore, $G \rightarrow_{R}^{k} G[I'/I]$.

4 GRAPH FORMULAS

In this section we introduce DRLG-literals, which are either equations or disequations between graphs and DRLG-formulas, that are sets of DRLG-literals.

Definition 37. A DRLG-equation is an unordered pair written $G \equiv H$, where $G, H$ are root-compatible DRLG-graphs. A DRLG-disequation is the negation of a DRLG-graph equation, written $G \nleq H$. A DRLG-literal can be either one of a DRLG-equation, a DRLG-disequation or $\perp$ (standing for false). The set of variables occurring in a DRLG-literal $L$ is defined as follows: $\mathcal{V}(G \equiv H) = \mathcal{V}(G) \cup \mathcal{V}(H)$ and $\mathcal{V}(\perp) = \emptyset$. All substitutions and $\pi$-mappings are extended to DRLG-literals as follows: $\mu(G \equiv H) = \mu(G) \equiv \mu(H)$, for $\equiv \in \{\equiv, \nleq\}$, and $\mu(\perp) = \perp$.

If $L, L'$ are DRLG-literals, we write $L \leadsto L'$ if either $L = L'$ or $L = (G \equiv H), L' = (G' \equiv H'), G \sim G'$ and $H \sim H'$.

A DRLG-formula is a set of DRLG-literals.

Remark 38. Note that disjunctions are not allowed in our syntax. Considering disjunctions of DRLG-literals would actually be straightforward (with the classical semantics), but it would not bring much generality in our context, because nodes could not be shared between the considered DRLG-literals (only variables could be shared). Stronger forms of disjunction could be considered but it is not clear how their semantics would be defined. This is why formulas are defined as conjunctions of literals. Depending on the application, disjunctions can of course be encoded at the object level by specific nodes within the graphs, with appropriate axioms. The graph languages considered in quantum computing (such as the ZX calculus) do not allow for disjunctions.

Definition 39. For any DRLG-formula $S$, we denote by $I_3(S)$ the set of ground instances of a DRLG-literal in $S$, i.e., the set of DRLG-literals $\sigma(L)$ where $L \leadsto L'$ for some $L' \in S$ and $\sigma$ is a ground substitution of domain $V(L)$.

We write $L \rightarrow_R L'$ if $L = G \equiv H$, $L' = G' \equiv H$ and $G \rightarrow_R G'$.

5 SEMANTICS

Depending on the considered application, graphs may be associated with various semantics. For instance, in quantum programming, ZX-diagrams are associated with complex matrices. In classical circuits, graphs may be associated with operators with multiple inputs and outputs. In the present paper, for the sake of generality, we make no assumption on the way DRL-graphs are interpreted, and we only assume that the chosen semantics is compatible with the replacement of DRL-subgraphs. This leads to the notion of a DRLG-congruence, which lifts the usual notion of congruence to DRL-graphs.

Definition 40. A binary relation $\equiv$ on DRL-graphs is closed under isomorphisms if $G \equiv G' \wedge H \sim H' \Rightarrow G \equiv H'$.

Definition 41. A binary relation $\equiv$ on DRL-graphs is closed under embeddings if for all DRL-graphs $G, H, H'$ such that $H \leq_{G} G$, $H'$ is substitutable for $H$ in $G$ and $H \equiv H'$, we have $G \equiv G[H'/H]$.

Definition 42. A DRLG-congruence $\equiv$ is an equivalence relation between ground DRL-graphs that is closed under isomorphisms and embeddings.

Definition 43. A DRLG-congruence $\equiv$ validates $\phi$ if:
- $\phi$ is a ground DRLG-equation $G \equiv H$ and $G \equiv H$;
- $\phi$ is a ground DRLG-disequation $G \nleq H$ and $G \nleq H$;
- $\phi$ is a formula and $\equiv$ validates all DRLG-literals in $I_3(\phi)$.

If a DRLG-congruence validates $\phi$ then it is a model of $\phi$, and we say that $\phi$ is satisfiable. We write $\phi \models \psi$ (and say that $\psi$ is a logical consequence of $\phi$) if every model of $\phi$ is a model of $\phi'$, and $\phi \equiv \psi$ if $\phi \models \psi$ and $\psi \models \phi$.

Proposition 44. For any set of DRLG-equations $E$, there exists a DRLG-congruence $\equiv_E$ that is the minimal (w.r.t. $\subseteq$) model of $E$.

Proof. It is easy to check that if $M$ is a set of models of $E$, then the relation defined as the intersection of all the relations in $M$ is also a model of $E$. Further, $E$ has at least one model, containing all pairs of root-compatible ground DRL-graphs. Hence $\equiv_E$ can be defined as the intersection of all the models of $E$.

We now prove that the satisfiability problem is undecidable for ground DRLG-formulas. The result emphasizes the difference with the usual ground equational logic (on terms) which is well-known to be decidable.

Theorem 45. The satisfiability problem is undecidable for ground DRLG-formulas.

Proof. The proof goes by a reduction from the halting problem for Turing machines (TM). Let $M$ be a deterministic TM of the form $(Q, \Gamma, \delta, \Sigma, \epsilon, q_0, q_f, \delta, F)$, where $Q$ is the set of states, $q_0$ is the initial state, $F \subseteq Q$ is the set of final states, $b$ is the blank symbol, $\Gamma$ is the alphabet, $\Sigma \subseteq \Gamma$ is the set of input symbols and $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ is the transition function. We assume, w.l.o.g., that $Q \cap \Gamma = \emptyset$. The configurations are tuples $c = (q, w, w')$, where $q$ is a state and $w, w'$ are non-empty substrings of the tape before and after the head, respectively (we assume that $w, w'$ are empty for technical convenience). Any such configuration, with $w = w_1 \ldots w_n$, $w' = w'_{m} \ldots w_m$ and $w_i, w'_j \in \Gamma$, may be encoded into a graph $G(c)$ as follows (where $s, h, e$ are pairwise distinct symbols not occurring in $\Gamma \cup \Sigma$):

$$
N_G(c) = \{a_0, a_1, a_2, a_3, b_1, \ldots, b_n, b'_1, \ldots, b'_m\} \\
R_G(c) = \emptyset \\
E_G(c) = \{(b_1 \Rightarrow b_{i+1}) \mid i = 1, \ldots, n - 1\} \\
\cup \{(b'_j \Rightarrow b'_{j+1}) \mid j = 1, \ldots, m - 1\} \\
\cup \{(a_1 \Rightarrow b_1), (b_2 \Rightarrow a_2), (a_2 \Rightarrow b'_1), (b'_m \Rightarrow a_3)\}
$$

$$
I_G(c) = w_i \text{ (for } i = 1, \ldots, n) \\
l_G(c) = w'_j \text{ (for } j = 1, \ldots, m) \\
l_G(c) = q \text{ if } \delta(q_0, \epsilon) = (q_1, \delta_1, R) \text{ and } k \in \Gamma.
$$

The transition function is encoded by the following DRLG-equations:

$$
\begin{align*}
\odot & \Rightarrow \odot \odot \odot \odot \\
\odot & \Rightarrow \odot \odot \odot \odot
\end{align*}
$$

for all $q_1, i, q_2, j, k$ such that $\delta(q_1, \epsilon) = (q_2, j, R)$ and $k \in \Gamma$. The second rule encodes the fact that the tape is infinite: if the head
is at the end of the word, a new blank symbol must be created
to ensure that there are always symbols after the head (i.e., $w'$
must be nonempty in any configuration $(q, w, w')$). Similar rules
are defined for the left movement (for all $q_i, i, q_{j, j, k}$ such that
$\delta(q_{i, i}) = (q_{j, j}, L)$ and $k, l \in \Gamma$):
\[
\begin{array}{c}
\delta(q_{i, i}) \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \right.
A Superposition-Based Calculus for Diagrammatic Reasoning (Long Version)

7 COMPLETENESS

In this section we prove the completeness of our calculus, in other words, that if a saturated DRLG-formula is unsatisfiable, then it necessarily contains \( \bot \). Throughout this section, we assume that \( S \) denotes a fixed DRLG-formula.

Our first goal is to show that if \( S \) is saturated and contains no contradiction then the relation \( \rightarrow_{R_S} \) is confluent (Lemma 57). This result will allow us to use this relation as a basis for defining a DRLG-congruence satisfying \( S \) (Definition 58). Note that showing local confluence is not sufficient since \( \rightarrow_{R_S} \) is not necessarily well-founded (this is an essential difference with the usual completeness proofs for standard Superposition calculi, for which total reduction orders exist). We consider a restriction of \( \rightarrow_{R_S} \) obtained by bounding the number of steps for which the rewritten graph is greater or equal to some given DRLG-graph \( G \).

Definition 54. Let \( G \) be a DRL-graph. We write \( H \rightarrow_{R_S}^n G \) iff there exists a sequence \( I_1(\pi_1, \ldots, \pi_n) \) with \( I_1 = H, I_m = H', I_i \rightarrow_{R_S} I_{i+1} \) for all \( i = 1, \ldots, m-1 \) and there are at most \( n \) indices \( i \) in \([1, m - 1]\) such that \( I_{i+1} \not\in S \).

Proposition 55. We have the following properties:

1. If \( H \rightarrow_{R_S}^n I \) then \( H \rightarrow_{R_S}^m I \), for some \( n \in \mathbb{N} \).
2. If \( H \rightarrow_{R_S}^n I \) then \( H \rightarrow_{R_S}^m I \), for all \( m \geq n \).
3. (If \( H < G \) and \( H \rightarrow_{R_S}^n I \) then \( H \rightarrow_{R_S}^0 I \).
4. If \( H \leq G \) and \( H \rightarrow_{R_S}^n I \) then \( H \rightarrow_{R_S}^0 I \).
5. If \( H \rightarrow_{R_S}^n I \), \( H' \rightarrow_{R_S}^n I \) then \( H \rightarrow_{R_S}^{n+m} I \).

Proof. The results follow immediately from the definition of \( \rightarrow_{R_S}^n \) and Proposition 51.

The following lemma states a form of local confluence:

Lemma 56. Assume that \( S \) is saturated and \( \bot \not\in S \). Let \( G, G_1, G_2 \) be ground DRL-graphs such that \( G \rightarrow_{R_S} G_i \) (for \( i = 1, 2 \)). There exists a DRL-graph \( G' \) and numbers \( \eta_i \in \{0, 1\} \) such that \( G_i \rightarrow_{R_S}^n G' \), for all \( i = 1, 2 \). Furthermore, if \( G > G_i \) for some \( i = 1, 2 \) then \( \eta_1 = \eta_2 = 0 \).

Proof. By definition of \( \rightarrow_{R_S} \) there exist rules \( H_i \rightarrow H'_i \) in \( R_S \), substitutions \( \sigma_i \) and \( N \)-mappings \( \mu_i \) such that, for \( i = 1, 2 \), \( \sigma_i(\mu_i(H_i)) \subseteq S \), \( \sigma_i(\mu_i(H'_i)) \) is substitutable for \( \sigma_i(\mu_i(H_i)) \) in \( G \), and \( G_i \rightarrow_{R_S} G_i(\sigma(\mu_i(H'_i)) / \sigma(\mu_i(H_i))) \).

Let \( i \in \{1, 2\} \). Since \( R_S \) is ground we have \( \sigma_1 = id \), and since \( I_5(S) \) (hence \( R_S \)) is closed under isomorphisms, we may assume that \( \mu_1 = id \), so that \( G_i = G_i(H'_i/H_i) \). Furthermore, by definition of \( R_S \), we have \( H_i \geq H'_i \) and since \( \geq \) is closed under embeddings, \( G \geq G_i \).

We distinguish two cases.

- Assume that \( N_{J_R} \cap N_{H_L} = \emptyset \). By Lemmas 27 and 28, for every \( i, j \in \{1, 2\} \) with \( i \neq j \), we have \( H_i \leq G_j, H'_i \) is substitutable for \( H_i \) in \( G_j \), and:

\[
G_j[H'_i/H_i] = G_j[H'_j/H_j].
\]

By Definition 34 (a), we have \( H_i \leq C, G_j \), thus \( G_i \rightarrow_{R} G_i(H'_i/H_j) \), so that \( G_i \rightarrow_{R} G_i[H'_i/H_j] \), where:

\[
\eta_i = \begin{cases} 1 & \text{if } G_i[H'_i/H_j] \geq G_i \\ 0 & \text{otherwise} \end{cases}
\]

Note that by Equation 1 we necessarily have \( \eta_1 = \eta_2 \). Since \( H_j \geq H'_j \) and \( \geq \) is closed under embeddings, we have \( G_j \geq G_i[H'_i/H_j] \). If \( G_j \geq G_i \), for some \( i = 1, 2 \), then \( G_j \geq G_i[H'_i/H_j] \), thus by definition \( \eta_1 = 0 \). By Equation 1, we obtain the stated result.

- Assume that \( N_{J_R} \cap N_{H_L} \neq \emptyset \). By definition of \( R_S \), there exist DRL-graphs \( L_i \in S \), groundsubstitutions \( \theta_i \) and \( N \)-mappings \( v_i \) (for \( i = 1, 2 \)) such that \( H_i \theta_i(\nu_i(\nu_i))) \) and \( H'_i = \theta_i(\nu_i(\nu_i'))) \). We may assume that \( J_i \leq I \) and \( I \leq I_2 \) are variable-disjoint, so that \( \theta_i \) and \( \theta_2 \) have disjoint domains. By Definition 32 (5), \( v_1(I_1) \) and \( v_2(I_2) \) admit a merge \( I \), with \( \mu_i(\theta_i) \) such that \( (\theta_i \cup \theta_2) = \theta_0 \Rightarrow \theta, \) for some substitution \( \theta_0 \). Then, \( \theta_0(I) \subseteq C \). Since \( I_5(S) \) is closed by node renaming, we may also assume, w.l.o.g., that \( v_l(I_1) \cap \neg G_2 \subseteq v_l(I_1) \), so that \( v_l(I_1) \) is substitutable for \( v_l(I_1) \) in \( I \) and also that \( I_5 \) (\( I_1 \)) (\( I_5 \)) (\( I_1 \)) (\( I_5 \)).

Let \( i \in \{1, 2\} \) and assume that \( I[\theta(\nu(i_1))] \theta(\nu(i_1))) \) (\( I_5 \)) (\( I_1 \)). Then, since the order \( > \) is closed under substitutions, we have \( \theta_0(I[\theta(v_1(I_1))] \theta(\nu(i_1))) \) (\( I_5 \)) (\( I_1 \)). and therefore \( \theta_0(I[\theta(v_1(I_1))] \theta(\nu(i_1))) \) (\( I_5 \)) (\( I_1 \)). We get:

\[
G'[\theta_0(I[\theta(v_1(I_1))] \theta(\nu(i_1))) / \theta(\nu(i_1))] > G
\]

because \( \geq \) is closed under embeddings, thus, by Lemma 29,

\[
G[\theta(\nu(v'_1(I_1))] \theta(\nu(v'_1(I_1))] > G,
\]

Therefore, \( G[H'_i/H_i] > G \), which contradicts the fact that \( H_j \geq H'_j \).

Consequently, we have \( I[\theta(\nu(i_1))] \theta(\nu(i_1))) \) (\( I_5 \)) (\( I_1 \)) (\( I_5 \)) (\( I_1 \)) (\( I_5 \)). This entails that the rule \( S^+ \) is applicable on \( I_1 \approx I_2 \) and \( I_2 \approx I_2 \) (up to the node renamings \( v_1, v_2 \)), yielding the DRLG-literal:

\[
I[\theta(v_1(I_1))] \theta(\nu(i_1))) \theta(\nu(i_1))) \approx I[\theta(v_2(I_2))] \theta(\nu(i_2))) \theta(\nu(i_2)))
\]

Since \( S \) is saturated and \( \bot \not\in S \), there exist DRL-graphs \( J_1 \) (\( i = 1, 2 \)) such that either \( J_1 \approx J_2 \) or \( J_1 \approx J_2 \in I_5(S)_p \) (\( i = 1, 2 \)), yielding the DRLG-literal:

\[
I[\theta(v_1(I_1))] \theta(\nu(i_1))) \theta(\nu(i_1))) \approx I[\theta(v_2(I_2))] \theta(\nu(i_2))) \theta(\nu(i_2)))
\]

Since \( S \) is saturated and \( \bot \not\in S \), the proof if \( J_1 = J_2 \) is similar and simpler. Since
\[\begin{align*}
&\text{If } I \subseteq G, \text{ then by minimality of } G \text{ there exists a DRL-graph } G' \text{ such that for } i = 1, 2, I_i \rightarrow_{R_5}^\ast G', \text{ hence } I_i \rightarrow_{R_5}^\eta G' \text{ by Proposition 55 (3). Since } G_i \rightarrow_{R_5}^\eta I_i, \text{ we deduce that } G_i \rightarrow_{R_5}^\eta G' \text{ by Proposition 55 (5). If } \eta_1 = \eta_2 = 0 \text{ then this entails that } G_i \rightarrow_{R_5}^\eta G' \text{ and the proof is completed. Otherwise, by definition of } \eta_i, \text{ we have } H_i \approx H_i \geq G, \text{ hence } n_1, n_2 > 0 \text{ and } n_1, n_2 \geq \eta_1, \eta_2, \text{ thus we also have } G_i \rightarrow_{R_5}^\eta G'.
\end{align*}\]

**Definition 58.** We denote by \(\ll\) the relation defined as follows: \(G \ll H\) if there exists a DRL-graph \(I\) such that \(G \ll_{R_5} I\) and \(H \rightarrow_{R_5} I\).

**Lemma 59.** If \(S\) is saturated and does not contain \(\bot\) then \(\ll_S\) is a DRL-congruence.

**Proof.** It is clear that \(\ll_S\) is reflexive and symmetric. Since \(\ll_S\) is transitive by definition, it is also closed under isomorphisms.

We now show that \(\ll_S\) is transitive. If \(G_1 \ll_S G_2 \ll_S G_3\) then there exist \(I_1, H_2, I_2\) such that \(G_1 \rightarrow_{R_5} I_1, G_2 \rightarrow_{R_5} H_2, G_3 \rightarrow_{R_5} I_2\). Since \(I_2 \rightarrow_{R_5} H_2\) by Lemma 57, we deduce that there exists a DRL-graph \(I_S\) such that \(H_1 \rightarrow_{R_5} I_S\). By transitivity of \(\ll_S\), we get \(G_1 \ll_S G_3\). There remains to prove that \(\ll_S\) is closed under embeddings. Consider \(G, H, H_1, H_2, H_1 \ll_S H_2\) for some \(H_1, H_2\) and let \(I_2 \rightarrow_{R_5} H_2\). By definition of \(\ll_S\), there exists a DRL-graph \(I_2\) such that \(G_1 \ll_{R_5} I_2\) and \(H_1 \ll_{R_5} I_2\). This proof is completed.

**Lemma 60.** If \(S\) is saturated and does not contain \(\bot\) then \(\ll_S\) is a model of \(S\).

**Proof.** Let \(G \approx G'\) be a DRLG-equation in \(S\) and let \(\theta\) be a ground substitution of the variables in \(G, G'\). By definition, we have \(\theta(G) \approx \theta(G') \in L(S)\). Since \(\approx\) is total on ground DRL-graphs, either \(\theta(G) \subseteq \theta(G')\) or \(\theta(G') \subseteq \theta(G)\). We assume by symmetry that \(\theta(G) \subseteq \theta(G')\), so that \(R_5\) contains a rule \(\theta(G) \rightarrow \theta(G')\). Then by definition of \(\ll_S\), we have \(\theta(G) \ll_S \theta(G')\), hence \(\ll_S\) is a model of \(G \approx G'\).

Now consider a ground DRLG-disequation \(G \neq G'\) that is redundant w.r.t. \(S\); we prove that \(G \not\ll_S G'\). This is sufficient to obtain the result, since all the ground instances of a DRLG-disequation occurring in \(S\) are redundant w.r.t. \(S\). Assume for a contradiction that \(G \ll_S G'\), so that there exists \(H\) such that \(G \rightarrow_{R_5} H\) and \(G' \rightarrow_{R_5} H\), for some \(n, n' \in \mathbb{N}\). The proof is by induction on the pair \((\{G, G'\}, \{n, n'\})\), ordered by the lexicographic and multiset...
extension of the ordering $<$ and of the usual order on natural numbers.

- If there exists $I$ such that $G \rightarrow_{RS}^I$ and $I \neq G'$ is redundant w.r.t. $S$ then by Proposition 51, $I < G$, thus by the induction hypothesis $I \not\rightarrow_{S} G'$. By definition of $\rightarrow_{S}$ we have $G \not\rightarrow_{S} I$, hence $G \not\rightarrow_{S} G'$.

- The proof is similar if there exists $I$ such that $G' \rightarrow_{RS}^I$ and $G \neq 1$ is redundant w.r.t. $S$.

- Otherwise, $G \neq G'$ must be an instance of a DRL-literal in $S$, i.e., there exist a DRL-disjunction $H \not\rightarrow H'$ occurring in $S$ (up to a renaming of nodes) and a substitution $\sigma$ such that $G = \sigma(H)$ and $G' = \sigma(H')$. We distinguish two cases.

  - If $n = n' = 0$ then $\sigma(H) \not\rightarrow \sigma(H')$, thus $H$ and $H'$ are label-unifiable (up to a renaming of nodes). Consequently, the rule $R$ applies. Since $S$ is saturated, this entails that $\perp \in S$, which contradicts the hypothesis of the lemma.

  - Otherwise, we have either $n > 0$ or $n' > 0$. Assume that $G > G'$, $n = 0$ and $n' > 0$. Then we have $G \sim H$, thus $G' \rightarrow_{RS} G$ and $G' \gneq G$ by Proposition 51, a contradiction. The case where $G' > G$, with $n > 0$ and $n' = 0$ is symmetric. Now assume that $n = 0$ and that $G \geq G'$ (the proof where $n > 0$ and $G' \geq G$ is symmetric). Then there exists a DRL-equation $J = J'$ in $S$ (up to a renaming of nodes) and a substitution $\theta$ such that $\theta(J) \not\rightarrow G$, $G \rightarrow_{RS} \theta(\sigma(0))/\theta(J) \rightarrow_{RS}^I$ and $\theta(J) \geq \theta(J')$. We assume, w.l.o.g. that $J = J'$ and $H \not\rightarrow H'$ share no variable, so that $\sigma$ and $\theta$ have disjoint domains. Let $y$ be a most general substitution such that $\gamma(J) \not\rightarrow \gamma(H)$. By definition, there exists $\gamma'$ such that $(\sigma \cup \gamma) = \gamma' \circ \gamma$. Since $\theta(J) \geq \theta(J')$, and is closed under substitutions, we have $\gamma(J') \not\rightarrow \gamma(G')$ (as otherwise we would have $\gamma'(\gamma(J')) > \gamma'(\gamma(G'))$, i.e., $\theta(J') > \theta(J)$). If $G \geq G'$ and $J$ is closed under substitutions, we conclude that $\gamma(H') \not\rightarrow \gamma(H)$. This entails that the rule $\perp$ is applicable on $H \not\rightarrow H'$ and $J = J'$, yielding: $\perp \rightarrow_{S} ^\perp \rightarrow_{S} G'$. This entails that the rule $\perp$ is applicable on $G' \rightarrow_{RS} G'$, hence $G \not\rightarrow_{S} G'$.

\[ \square \]

**Theorem 61 (Completeness).** Any saturated DRL-formula not containing $\perp$ is satisfiable.

**Proof.** This follows immediately from Lemma 60. $\square$

8 AN APPLICATION TO CIRCUITS

8.1 A Class of DRL-Graphs

In this section we consider a class of graphs, denoted by Circuits, which intuitively consists of graphs that represent gates. Informally, a gate with $n$ inputs and $m$ outputs is represented in this class by a graph with $2(n + m) + 1$ nodes:

- one node that contains the information about the number of entries and outputs;
- $n$ nodes that represent the entries and $m$ nodes that represent the outputs;
- $n + m$ root nodes that are used for rewriting operations.

We also define the notion of a subcircuit and show that it is a C-motion, thus guaranteeing the completeness of the proof procedure defined in Section 6.

**Definition 62.** We denote by Circuits the class of graphs $G$ satisfying the following properties:

1. The set of labels is $\mathbb{N} \setminus \{0\}$.
2. There are sorts gate$_m^n$ into and from denoting gates with $n$ entries and $m$ outputs, input ports, and output ports respectively.
3. For each node $\alpha$ of sort gate$_n^m$, there exist exactly $n$ nodes $\alpha_1, \ldots, \alpha_n$ of sort into labeled with $1, \ldots, n$ respectively and $m$ nodes $\alpha'_1, \ldots, \alpha'_m$ of sort from labeled with $1, \ldots, m$ respectively, and edges $(\alpha \rightarrow \alpha'_i)$ and $(\alpha' \rightarrow \alpha_j)$ for every $i \in [1, n]$ and $j \in [1, m]$.
4. All nodes of sort into have exactly one incoming and at most one outgoing edge; all nodes of sort from have exactly one outgoing and at most one incoming edge.
5. Every node of sort into which is labeled by $i \in \mathbb{N} \setminus \{0\}$ admits one outgoing edge, the target of which is a node of sort gate$_n^m$, where $i \leq n$.
6. Every node of sort from which is labeled by $i \in \mathbb{N} \setminus \{0\}$ admits one incoming edge, the source of which is a node of sort gate$_n^m$, where $i \leq m$.
7. All the other edges are of the form $(\alpha \rightarrow \beta)$ where $\alpha$ is of sort from and $\beta$ is of sort into.
8. $\mathcal{R}_G$ is of the form $R_f \cdot R_i$, where every node in the sequence $R_f$ is of sort from and admits no incoming edge, and every node in the sequence $R_i$ is of sort into and admits no outgoing edge.

The sequence $R_f$ is denoted by $R_{G}^{in}$, and $R_i$ by $R_{G}^{out}$.

Intuitively, $R_{G}^{in}$ denotes the inputs of $G$ and $R_{G}^{out}$ its outputs.

**Example 63.** The empty DRL-graph $\emptyset \triangleright= (\emptyset, \emptyset, \emptyset, \emptyset)$, is in Circuits, with $R_{G}^{in} = R_{G}^{out} = \emptyset$.

**Definition 64.** Given a node $\alpha$ in $R_{G}^{in}$, we denote by $\text{trg}_{G}(\alpha)$ (or simply $\text{trg}(\alpha)$ when there is no ambiguity) the node $\beta$ such that $(\alpha \rightarrow \beta) \in E_G$. Given a node $\alpha$ in $R_{G}^{out}$, we denote by $\text{src}_{G}(\alpha)$ (or simply $\text{src}(\alpha)$ when there is no ambiguity) the node $\beta$ such that $(\beta \rightarrow \alpha) \in E_G$.

Since every node $\alpha$ in $R_{G}^{in}$ is of sort from, the node $\beta$ such that $(\beta \rightarrow \alpha) \in E_G$ exists and is unique; similarly, if $\alpha$ in $R_{G}^{out}$ then the node $\beta$ such that $(\beta \rightarrow \alpha) \in E_G$ exists and is unique.

**Definition 65.** $H$ is a subcircuit of a DRL-graph $G$ in Circuits, denoted by $H \subseteq G$, if and only if $H \subseteq G$ and $H \in \text{Circuits}$.

**Lemma 66.** Assume that $H \subseteq G$, let $H'$ be a DRL-graph in Circuits that is substitutable for $H$ in $G$ and let $G'' \triangleright= G[H'/H]$. Then $G'' \in \text{Circuits}$.

**Proof.** Let $\mu \triangleright=_{G} H \rightarrow H'$ (see Definition 15), we prove that $G'' \in \text{Circuits}$. Items (1) and (2) of the Definition 62 are immediate to verify.

**Item 3** Let $\alpha \in N_G$ be a node of sort gate$_n^m$. Note that $\alpha$ cannot be in $R_{G}^{in} \cup R_{H}$ if $\alpha \in N_{H}$, then because $H' \in \text{Circuits},$
it is guaranteed that there exist exactly $n$ nodes $\alpha_1, \ldots, \alpha_n$ of sort into labeled with $1, \ldots, n$ respectively and $m$ nodes $\alpha'_1, \ldots, \alpha'_m$ of sort from labeled with $1, \ldots, m$ respectively, and edges $(\alpha_i \rightarrow \alpha)$ and $(\alpha \rightarrow \alpha'_j)$ in $E_H$ for all $i \in [1, n]$ and $j \in [1, m]$. Since $E_H \subseteq E_G$, we have the result.

Otherwise, $\alpha \in N_G \setminus N_H$. Let $i \in [1, n]$, we prove that there exists a node $\alpha'_i$ of sort into labeled with $i$ such that $(\alpha'_i \rightarrow \alpha) \in \mathcal{E}_G$. The proof for nodes of sort from is similar. Since $G$ is Circuits, there exists a node $\alpha_i \in N_G$ of sort into labeled with $i$ such that $(\alpha_i \rightarrow \alpha) \in \mathcal{E}_G$. Necessarily $(\alpha_i \rightarrow \alpha) \in \mathcal{E}_G \setminus E_H$, and we have the result by taking $a'_i \triangleq \mu(\alpha_i)$.

**Item 4** Consider a node $\alpha \in N_G$ of sort into and, first assume that $\alpha \in N_G \setminus N_H$, so that $\mu(\alpha) = \alpha$ and $\alpha \notin N_H$ by Proposition 14. By definition there exists a unique edge $(\alpha' \rightarrow \alpha) \in \mathcal{E}_G \setminus E_H$, and since there can be no edge of the form $(\alpha'' \rightarrow \alpha) \in \mathcal{E}_H$, $\alpha$ admits $\mu(\alpha' \rightarrow \alpha) = (\mu(\alpha'' \rightarrow \alpha))$ as a unique incoming edge. If $\alpha \notin N_H$ then by definition $\mu(\alpha) = \alpha$ and by Proposition 14, $\alpha \notin N_G \setminus N_H$ because $\alpha \notin N_H$. Since $H \in$ Circuits, the proof in this case is straightforward.

The proof that $\alpha$ admits at most one outgoing edge is carried out in a similar way, and so are the proofs when $\alpha$ is of sort from.

**Item 5** Consider a node $\alpha \in N_G$ of sort into and labeled by $i$. A case analysis depending on whether $\alpha \in N_G \setminus N_H$ or $\alpha \in N_H$ as in the previous point shows that the target of $\alpha$ is of sort gate$^m_\alpha$ for some $n \geq i$.

**Item 6** The proof is the same as in the previous case.

**Item 7** Because $\mu$ preserves sorts, it is straightforward to prove that all other edges are of the form $(\alpha \rightarrow \beta)$ where $\alpha$ is of sort from and $\beta$ is of sort into.

**Item 8** By definition we have $\mathcal{R} = \mu(\mathcal{R}_G)$, and since $G$ and $H'$ are both in Circuits, $\mathcal{R}_G$ is of the required form.

**Lemma 67.** Assume that $H \leq G$, $H' \leq G$ and that $H$ and $H'$ are label-compatible. Then the $E$-merge of $H$ and $H'$ with $E \triangleq \{ (\alpha \rightarrow \beta) \mid \alpha \in \mathcal{R}_H \setminus N_H, \beta \in \mathcal{R}_H \setminus N_H, (\alpha \rightarrow \beta) \notin \mathcal{E}_G \}$ is in Circuits.

**Proof.** Let $G'$ denote the $E$-merge of $H$ and $H'$; by Proposition 31, $G' \leq G$. It is straightforward to verify that $G' \in$ Circuits: the existence of nodes and edges satisfying the requirements of the definition of Circuits are deduced from the fact that $H$ and $H'$ are both in Circuits, and the uniqueness properties are consequences of the fact that $H$ and $H'$ are both subgraphs of $G$ which is in Circuits.

**Lemma 68.** The relation $\leq_G$ is a $C$-relation, with $C = \text{Circuits}$.

**Proof.** First note that since $H \in$ Circuits, if $\alpha \in N_G \setminus N_H$ is of sort into then there can be no edge $(\alpha_i \rightarrow \alpha) \in \mathcal{E}_H$. Indeed, necessarily, $\alpha_i \in \mathcal{R}_G^\text{in}$ and by definition, $\alpha_i$ cannot admit any outgoing edge. Similarly, if $\alpha \in N_G \setminus N_H$ is of sort from then there can be no edge $(\alpha \rightarrow \alpha_i) \in \mathcal{E}_H$.

(1) By definition, if $H \leq_G G$ then $G \in \text{Circuits}$.

(2) By Lemma 19 we have $H' \leq G[H'/H]$ and by Lemma 66 $G[H'/H] \in \text{Circuits}$.

(3) It is clear that if $G \in \text{Circuits}$ then $\mu(G) \in \text{Circuits}$ for all $N$-mappings $\mu$. By Proposition 9, we deduce that if $H \leq_G G$ then $\mu(H) \leq_G \mu(G)$.

(4) By Lemma 27 we have $H \leq G[I/I']$ and by Lemma 66, $G[I'/I] \in \text{Circuits}$.

(5) If $H \leq_G G$ and $H' \leq_G G$ then the merge $G'$ of $H$ and $H'$ considered in Lemma 67 is such that $G' \leq_G G$.

**8.2 Parallel and Sequential Composition**

It is natural to construct a circuit by composing simpler components. For instance the diagrams of the ZX calculus [12], formally defined as morphisms in a dagger compact category, are built from a set of generators via composition and monoidal product (modulo equalities induced by the considered structure). We thus introduce two standard composition operations: parallel and sequential composition. The empty graph is the neutral element for parallel composition, and we introduce the class of so-called identity graphs, which can be viewed as neutral elements for sequential composition.

**Definition 69.** Consider two graphs $G_1, G_2$ in Circuits, such that $N_{G_1} \cap N_{G_2} = \emptyset$. We define the parallel composition of $G_1$ and $G_2$, denoted by $G_1 \parallel G_2$, as the graph $G_3$ constructed as follows.

(1) $N_{G_3} \triangleq N_{G_1} \cup N_{G_2}$;

(2) $\mathcal{R}_{G_3} \triangleq \mathcal{R}_{G_1} \circ \mathcal{R}_{G_2}$;

(3) $\mathcal{E}_{G_{12}} \triangleq \mathcal{E}_{G_1} \cup \mathcal{E}_{G_2}$;

(4) $\mathcal{I}_{G_3} \triangleq \mathcal{I}_{G_1} \cup \mathcal{I}_{G_2}$.

It is straightforward to check that $G_2 \parallel G_1$ is a DRL-graph when $G_1$ and $G_2$ are disjoint.

**Proposition 70.** Parallel composition is an associative operation on mutually disjoint DRL-graphs, and the empty DRL-graph $\emptyset$ is its neutral element.

**Proof.** Immediate.

**Definition 71.** Consider two graphs $G_1, G_2$ in Circuits, such that $N_{G_1} \cap N_{G_2} = \emptyset$, where $\mathcal{R}_{G_{12}} = (\alpha_1, \ldots, \alpha_n)$ and $\mathcal{R}_{G_2} = (\beta_1, \ldots, \beta_m)$.

We let $\gamma_{G_1} \gamma_{G_2}$ denote the mapping on $N_{G_1} \cup N_{G_2}$ defined as follows:

- For all $i = 1, \ldots, n, \gamma_{G_1} \gamma_{G_2} (\alpha_i) \triangleq \text{trg}_{G_2} (\beta_i)$ and $\gamma_{G_1} \gamma_{G_2} (\beta_i) \triangleq \text{src}_{G_1} (\alpha_i)$;

- For all other nodes $\alpha \in N_{G_1} \cup N_{G_2}$, $\gamma_{G_1} \gamma_{G_2} (\alpha) \triangleq \alpha$.

We define the sequential composition of $G_1$ and $G_2$, denoted by $G_1 ; G_2$, as the graph $G_3$ constructed as follows.

(1) $N_{G_3} \triangleq \gamma_{G_1} \gamma_{G_2} (N_{G_1}) \cup \gamma_{G_2} (N_{G_2})$;

(2) $\mathcal{R}_{G_3} \triangleq \mathcal{R}_{G_1} \circ \mathcal{R}_{G_2}$;

(3) $\mathcal{E}_{G_{12}} \triangleq \gamma_{G_1} \gamma_{G_2} (\mathcal{E}_{G_1}) \cup \gamma_{G_2} (\mathcal{E}_{G_2})$;

(4) $\mathcal{I}_{G_3} \triangleq \mathcal{I}_{G_1} \cup \mathcal{I}_{G_2}$.

It is straightforward to check that $G_1 ; G_2$ is a DRL-graph when $G_1$ and $G_2$ satisfy the conditions of Definition 71.
A Superposition-Based Calculus for Diagrammatic Reasoning (Long Version)

If \( \alpha \in R_{G_1 \circ G_2}^{in} \), then \( \text{trg}_{G_2 \circ G_1}(\alpha) = Y_{G_1}^G(\text{trg}_{G_2}(\alpha)) \).

If \( \alpha \in R_{G_2 \circ G_1}^{out} \), then \( \text{src}_{G_2 \circ G_1}(\alpha) = Y_{G_1}^G(\text{src}_{G_2}(\alpha)) \).

**Proposition 74.** If the graphs \( G_1 \), \( G_2 \) and \( G_3 \) in \textit{Circuits} are mutually disjoint and such that \( G_3 \circ G_2 \circ G_1 \) are well-defined, then the sequential compositions \((G_3 \circ G_2) \circ G_1 \) and \( G_3 \circ (G_2 \circ G_1) \) are well-defined and equal.

**Proof.** We assume

\[
\text{\text{R}}_{G_1}^{\text{out}} = \langle \alpha_1, \ldots, \alpha_n \rangle \quad \text{R}_{G_2}^{\text{in}} = \langle \beta_1, \ldots, \beta_n \rangle
\]

Let \( H \equiv \text{G} \circ G_1 \) and \( I \equiv G_3 \circ G_2 \). We begin by proving the following statements.

1. If \( \alpha \in \mathcal{N}_{G_1} \), then \( Y_{G_1}^I \circ Y_{G_1}^G(\alpha) = Y_{G_1}^I(\alpha) \). First assume that \( \alpha \in \mathcal{R}_{G_1}^{\text{out}} \), i.e., that \( \alpha = \alpha_i \) for some \( i = 1, \ldots, n \). Then by construction we have \( Y_{G_1}^I \circ Y_{G_1}^G(\alpha) = Y_{G_1}^I(\text{trg}_{G_1}(\beta_i)) \), and by Proposition 73, \( Y_{G_1}^I(\alpha) = \text{trg}_{G_1}(\beta_i) = Y_{G_1}^I(\text{trg}_{G_1}(\beta_i)) \).

2. First assume that \( \alpha \in \mathcal{R}_{G_1}^{\text{in}} \), i.e., that \( \alpha = \beta_i \) for some \( i = 1, \ldots, n \). Then by Proposition 73 we have \( \mathcal{R}_{G_2}^{\text{out}} = \mathcal{R}_{G_2}^{\text{out}} \), therefore, \( Y_{G_1}^I(\text{trg}_{G_2}(\beta_i)) = Y_{G_1}^I(\text{src}_{G_2}(\beta_i)) \). Otherwise, since the considered graphs are pairwise disjoint and \( \mathcal{R}_{G_1}^{\text{out}} = \mathcal{R}_{G_2}^{\text{out}} \), we have

\[
Y_{G_1}^I \circ Y_{G_1}^G(\alpha) = Y_{G_1}^I(\alpha) = \alpha = Y_{G_1}^I(\alpha),
\]

to the result.

3. Now assume that \( \alpha \in \mathcal{R}_{G_2}^{\text{out}} \), i.e., that \( \alpha = \gamma_i \) for some \( i = 1, \ldots, m \). Then by Proposition 73 we have \( \mathcal{R}_{G_1}^{\text{in}} = \mathcal{R}_{G_2}^{\text{out}} \) and

\[
Y_{G_1}^I \circ Y_{G_1}^G(\alpha) = Y_{G_1}^I(\alpha) = \text{trg}_{G_1}(\delta_i) = Y_{G_1}^I(\text{trg}_{G_1}(\delta_i)) = Y_{G_1}^I \circ Y_{G_1}^G(\alpha).
\]

Otherwise we have

\[
Y_{G_1}^I \circ Y_{G_1}^G(\alpha) = Y_{G_1}^I(\alpha) = \alpha = Y_{G_1}^I(\alpha),
\]

to the result.

4. Now assume that \( \alpha \in \mathcal{R}_{G_2}^{\text{out}} \), i.e., that \( \alpha = \delta_i \) for some \( i = 1, \ldots, m \). Then by construction we have \( Y_{G_1}^I \circ Y_{G_1}^G(\alpha) = Y_{G_1}^I(\text{src}_{G_2}(\gamma_i)) \), and by Proposition 73, \( Y_{G_1}^I(\alpha) = \text{src}_{G_1}(\gamma_i) = Y_{G_1}^I(\text{src}_{G_2}(\gamma_i)) \). Still by Proposition 73 we have \( \mathcal{R}_{I}^{\text{in}} = \mathcal{R}_{G_1}^{\text{out}} \), therefore, \( Y_{G_1}^I(\text{trg}_{G_2}(\beta_i)) = Y_{G_1}^I \circ Y_{G_1}^G(\alpha) \).

Otherwise, since the considered graphs are pairwise disjoint and \( \mathcal{R}_{G_1}^{\text{out}} = \mathcal{R}_{G_2}^{\text{out}} \), we have

\[
Y_{G_1}^I \circ Y_{G_1}^G(\alpha) = Y_{G_1}^I(\alpha) = \alpha = Y_{G_1}^I(\alpha),
\]

to the result.
Using the statements above, we deduce that:
\[
\begin{align*}
E_{I_1G_1} &= Y_{G_1}^I(E_{G_1}) \cup Y_{G_1}^I(E_I) \\
&= Y_{G_1}^I(E_{G_1}) \cup Y_{G_1}^I \circ Y_{G_1}^I(E_{G_1}) \cup Y_{G_1}^I \circ Y_{G_1}^I(E_{G_1}) \\
&= Y_{G_1}^I \circ Y_{G_1}^I(E_{G_1}) \cup Y_{G_1}^I \circ Y_{G_1}^I(E_{G_1}) \cup Y_{G_1}^I \circ Y_{G_1}^I(E_{G_1}) \\
&= Y_{G_1}^H(E_{G_1}) \cup Y_{G_1}^H(E_{G_1}) \\
&= \mathcal{E}_{G_1^+H}.
\end{align*}
\]

We prove in a similar way that \(N_{I_1G_1} = N_{G_1^+H} \). Since \(R_{I_1G_1} = R_{G_1^+H} \) and \(l_{I_1G_1} = l_{G_1^+H} \) by construction, we have the result.

We have the following commutation rule that shows that parallel and sequential composition can be switched:

**Lemma 75.** Consider the pairwise disjoint graphs \(G_1, G_2, G_3, G_4 \) in \textsc{circuits}, and assume \([R_{I_1G_1}^{in}] = [R_{I_1G_2}^{out}] \) and \([R_{I_2G_3}^{in}] = [R_{I_2G_4}^{out}] \). Then we have:
\[
(G_2 \otimes G_4) \circ (G_1 \otimes G_3) = (G_2 \otimes G_1) \circ (G_4 \otimes G_3).
\]

**Proof.** We define the following graphs:
\[
\begin{align*}
H_1 &\overset{\text{def}}{=} G_2 \circ G_1 \\
H_2 &\overset{\text{def}}{=} G_4 \circ G_3 \\
H_3 &\overset{\text{def}}{=} H_1 \otimes H_2 \\
I_1 &\overset{\text{def}}{=} G_1 \otimes G_3 \\
I_2 &\overset{\text{def}}{=} G_2 \otimes G_4 \\
I_3 &\overset{\text{def}}{=} I_2 \circ I_1
\end{align*}
\]
and prove that \(H_3 = I_3 \).

First note that by construction, we have \(R_{I_1G_1}^{in} = R_{I_1G_2}^{out} \cdot R_{I_2G_3}^{out} \). Since \([R_{I_1G_1}^{in}] = [R_{I_1G_2}^{out}] \) and \([R_{I_2G_3}^{in}] = [R_{I_2G_4}^{out}] \) by hypothesis, \(I_3 \) is well-defined. Furthermore, because all graphs are pairwise disjoint, it is straightforward to verify that \(H_1 = G_1 \cup Y_{G_1} \).

\[
N_{H_1} = N_{H_1} \cup N_{H_2} = Y_{G_1}^G(N_{G_1}) \cup Y_{G_2}^G(N_{G_2}) \cup Y_{G_3}^G(N_{G_3}) \cup Y_{G_4}^G(N_{G_4}) = Y_{H_1}^G(N_{H_1}) \cup Y_{H_2}^G(N_{H_2}) = Y_{H_1}^G(N_{H_1}) \cup Y_{H_2}^G(N_{H_2}) = N_{I_2}.
\]

\[
R_{H_1} = R_{H_1}^{in} \cdot R_{H_2}^{out} \cdot R_{H_1}^{out} \\
= R_{H_1}^{in} \cdot R_{H_2}^{out} \\
= R_{H_1}.
\]

\[
E_{H_1} = E_{H_1} \cup E_{H_2} = Y_{G_1}^G(E_{G_1}) \cup Y_{G_2}^G(E_{G_2}) \cup Y_{G_3}^G(E_{G_3}) \cup Y_{G_4}^G(E_{G_4}) = Y_{H_1}^G(E_{H_1}) \cup Y_{H_2}^G(E_{H_2}) = Y_{H_1}^G(E_{H_1}) \cup Y_{H_2}^G(E_{H_2}) = E_{I_1}.
\]

We now define the class of identity graphs. Intuitively, these graphs can be viewed as parallel compositions of wires. Their sequential composition with a graph leaves the latter unchanged, up to a renaming of nodes.

**Definition 76.** A graph \(G \) is an id\(_k\)-graph if it is of the following form:
\[
\begin{align*}
N_G &= \{a_1, a_2\}, \\
R_G &= \langle a_1, a_2 \rangle, \\
E_G &= \{(a_1 \rightarrow a_2)\}, \\
I_G &= \emptyset,
\end{align*}
\]

\(a_1 \) is of sort from and \(a_2 \) is of sort into.

The empty graph is an id\(_0\)-graph and for \(k > 0\), a graph \(G \) is an id\(_k\)-graph if it is of the form \(H \otimes H'\), where \(H \) is an id\(_k\)-graph and \(H' \) is an id\(_{k-1}\)-graph.

It is easy to check that \(G \) is an id\(_k\)-graph iff it is an id\(_k\)-graph and that every id\(_k\)-graph is in \textsc{circuits}. In what follows, we will denote by \(I_k \) any id\(_k\)-graph.

**Proposition 77.** Consider a graph \(G \) in \textsc{circuits}, and let \(n \overset{\text{def}}{=} R_{I_kG_k}^{out} \) and \(m \overset{\text{def}}{=} R_{I_kG_k}^{in} \). Then \(G \approx (G \circ I_n) \) and \(G \approx (I_m \circ G) \).

**Proof.** Let \(H \overset{\text{def}}{=} G \circ I_n \), we prove that \(G \approx H\), the proof for the other statement is similar. Let \(R_{I_kG}^{in} = \langle a_1, \ldots, a_n \rangle, \quad R_{I_kG}^{out} = \langle \beta_1, \ldots, \beta_m \rangle \). Let \(Y \overset{\text{def}}{=} Y_{I_kG} \). Note that by construction, for all \(i = 1, \ldots, n \), we have \(Y_{\beta_i} = \text{trg}(a_i) \) and \(Y_{\alpha_i} = \text{src}(\beta_i) = \beta_i \).

Consider an \(N\)-mapping \(\mu\) such that for all \(i = 1, \ldots, n \), \(\mu(\beta_i) = a_i \) and for all \(a \in N_{G_k}, \mu(a) = a\). We show that \(\mu(H) = G\).

\[
\begin{align*}
N_{\mu(H)} &= \mu(N_G) \cup \mu(N_{I_kG}) \\
&= (N_G \setminus \{a_i \mid i = 1, \ldots, n\}) \cup (\{\beta_i \mid i = 1, \ldots, n\}) \\
&= (N_G \setminus \{a_i \mid i = 1, \ldots, n\}) \cup (\{\beta_i \mid i = 1, \ldots, n\}) \\
&= N_G.
\end{align*}
\]

\[
\begin{align*}
R_{\mu(H)} &= \mu(R_{I_kG}^{in}) \cdot R_{\mu(H)}^{out} \\
&= \mu(R_{I_kG}^{in}) \cdot \mu(R_{I_kG}^{out}) \\
&= R_{I_kG}^{in} \cdot R_{I_kG}^{out} \\
&= R_{I_kG}.
\end{align*}
\]

• Let \(E \overset{\text{def}}{=} \{(a_i \rightarrow \text{trg}(a_i)) \mid i = 1, \ldots, n\} \), so that no node \(a_i \) occurs in \(E_G \setminus E\). We have:
\[
\begin{align*}
E_{\mu(H)} &= \mu(Y(E_G) \cup Y(E_{I_kG})) \\
&= \mu(Y(E_G \setminus E)) \cup \mu(Y(E)) \cup \mu(Y(E_{I_kG})) \\
&= \mu(E_G \setminus E) \cup \mu((\beta_i \rightarrow \text{trg}(a_i)) \mid i = 1, \ldots, n) \\
&= (E_G \setminus E) \cup E \\
&= E_G.
\end{align*}
\]
We have defined an extension of the Superposition calculus to a class of graphs that is general enough to encode the graph-based languages that are used to perform diagrammatic reasoning on quantum algorithms and protocols. This calculus is complete, meaning that if a graph formula is unsatisfiable, then it is guaranteed that the calculus will generate \( \perp \). We now intend to adapt an implementation of the Saturation calculus to design a tool on which we will be able to evaluate the efficiency of the calculus and investigate how it can be improved for specific subclasses of graphs – for example, by defining suitable orderings for such subclasses in order to reduce the search space as much as possible.

There are several lines of future work that would be interesting to explore. We plan to investigate how variables representing graphs can be introduced into the calculus. During the construction of a refutation using a calculus that can handle such variables, these variables would be instantiated by graphs with specific properties that guarantee correctness, and we would for example permit to automatically synthesize circuits. Another promising topic is to extend graph literals with constraints. This would permit to represent algorithms in a more concise or natural way, while still being able to automatically verify these algorithms.

REFERENCES

[19] H. Ehrg, M. Pfender, and H. J. Schneider. Graph-grammars: An algebraic approach. In 16th Annual Symposium on Switching and Automata Theory, Iowa City,


