Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science 165 (2006) 201–212
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Reasoning with Graphs

Renata P. de Freitas[1]

Institute of Mathematics, UFF, Niterói, Brazil

Paulo A. S. Veloso[1,2]

Systems and Computer Engineering Program, COPPE/UFRJ
Rio de Janeiro, Brazil

Sheila R. M. Veloso[1,3]

Systems and Computer Engin. Dept., Faculty of Engineering, UERJ
Rio de Janeiro, Brazil

Petrucio Viana[1,4]

Institute of Mathematics, UFF, Niterói, Brazil

Abstract

In this paper we study the (positive) graph relational calculus. The basis for this calculus was introduced by S. Curtis and G. Lowe in 1996 and some variants, motivated by their applications to semantics of programs and foundations of mathematics, appear scattered in the literature. No proper treatment of these ideas as a logical system seems to have been presented. Here, we give a formal presentation of the system, with precise formulation of syntax, semantics, and derivation rules. We show that the set of rules is sound and complete for the valid inclusions, and prove a finite model result as well as decidability. We also prove that the graph relational language has the same expressive power as a first-order positive fragment (both languages define the same binary relations), so our calculus may be regarded as a notational variant of the positive existential first-order logic of binary relations. The graph calculus, however, has a playful aspect, with rules easier to grasp and use. This opens a wide range of applications which we illustrate by applying our calculus to the positive relational calculus (whose set of valid inclusions is not finitely axiomatizable), obtaining an algorithm for deciding the valid inclusions and equalities of the latter.

Keywords: Completeness, Decidability, Expressive power, Graph calculus, Relational Calculus.

[1] Research partly sponsored by CNPq and FAPERJ (grants E-26/152.395/2002 and E-26/171.180/2003).

[2] Partly sponsored by CNPq (grant 301163/91-0).

[3] Partly sponsored by CNPq (grants 471608/03-3 and 301526/2005-2), FAPERJ (grant E-26/131.180/2003 and Prociência).

[4] Email: petrucio@cos.ufrj.br

doi:10.1016/j.entcs.2006.05.046

1 Introduction

In this paper we study +RG, the (positive) relational calculus with graphs. The basis for the graph relational calculus was introduced by S. Curtis and G. Lowe [5]. They exemplified its strong expressive power, claimed soundness of their inference rules and left completeness as an open problem. Some variants of it, motivated by applications to semantics of programs and foundations of mathematics, appear scattered in the literature. In particular, D. Cantone et al. [4,7] deal with some questions about expressive power. D. Dougherty and C. Gutiérrez [6,9], and P.J. Freyd and A. Scedrov [8] apply a fragment of it to allegories. C. Brown and G. Hutton [3,10] present an approach for the introduction of *projections* and *parallelism* into the graph calculus. Although Curtis and Lowe give motivation and examples in [5], no proper treatment of these ideas as a logical system seems to have been presented.

The main issues addressed in this paper concern a proper formulation of the logical system +RG: a set of rules to derive graphs that is sound and complete with respect to the valid inclusions between graphs and a characterization of the graph relational language compared to a first-order positive fragment in the sense that both languages define the same binary relations.

Our formulation of the graph calculus leads to the following improvements: a proper treatment of the union operator by the introduction of the notion of a component of a graph; a more elaborated definition of homomorphism enabling both precise formulation and use of the homomorphism rule in proofs; a set of rules equivalent to the homomorphism rule providing a better understanding of it; a normal form for proofs resembling the familiar one in classical propositional logic; an analysis establishing the precise relationship among the positive relational calculus, the graph calculus and a positive fragment of the first-order language of binary relations. Despite being a notational variant of the latter, our graph calculus has a playful aspect, with rules easier to grasp and to use. Also, in contrast to the algebraic approach to relations, whose elements are relational terms, the graph approach deals with relational terms and points. This leads to a pictorial and smoother environment for relational calculus. Such an approach opens a wide range of applications and provides contributions to the areas of algebraic logic, algebraic semantics, theoretical computer science, and model theory. It also has some important practical consequences since it deals with relational formalisms that are widely applicable. We illustrate this aspect by using +RG to prove the valid inclusions and equalities of the positive relational calculus, a (non-finitely axiomatizable) decidable fragment of Tarski's relational calculus [18].

The paper is structured as follows. In Section 2, we present the syntax and semantics of the relational formalism based on graphs. In Section 3, we provide a set of rules to transform a graph in another one and prove that it is sound and complete with respect to the valid inclusions between graphs. In Section 4, we characterize the expressive power of the graph language in terms of the first-order language of binary relations. In Section 5, we apply the graph calculus to the positive fragment of the relational formalism presented in [17,16], proving the decidability of this system and presenting a set of axioms.

2 Syntax and semantics

The graph relational language uses familiar relational concepts. Its construction is based on the positive relational language, +RC, whose basic syntactical and semantical concepts of +RC are essentially those of [17,18,16] without complementation and empty relation.

The *terms* of +RC, typically noted R, S, T, are generated from the set of relational variables $\textsc{Rvar} = \{r_i : i \in \omega\}$ by applying the relational operators E, I, $^{\mathsf{T}}$, $\sqcap$, $\sqcup$, and $\circ$, according to the following grammar:

$$R ::= r_i \mid \mathsf{E} \mid \mathsf{I} \mid R^{\mathsf{T}} \mid R \sqcap S \mid R \sqcup S \mid R \circ S.$$

The models and the meaning $[\![R]\!]_{\mathfrak{M}}$ of a term R in a model $\mathfrak{M}$ are defined as in the relational case (excluding all references to the empty relation and to complementation). Formally, a *model* is a structure $\mathfrak{M} = \langle M, r_i^{\mathfrak{M}} \rangle_{i\in\omega}$, where $M \neq \emptyset$ and $r_i^{\mathfrak{M}} \subseteq M \times M$. Given a model $\mathfrak{M}$, E and I are interpreted, respectively, as the relations $M \times M$ and $\{(a, b) \in M \times M : a = b\}$; $^{\mathsf{T}}$, $\sqcap$, $\sqcup$, and $\circ$ as the conversion, intersection, union, and composition of relations, respectively.

Now, we present a relational language +RG, based on graphs. +RG is designed to *represent* relations using graphs of a special kind. Its language has two kinds of expressions: *components* and *graphs*. Components are (directed arc-labeled pseudo multi) graphs having a distinguished pair of nodes and arcs labeled by terms of +RC. Figure 1 shows three one-component graphs.

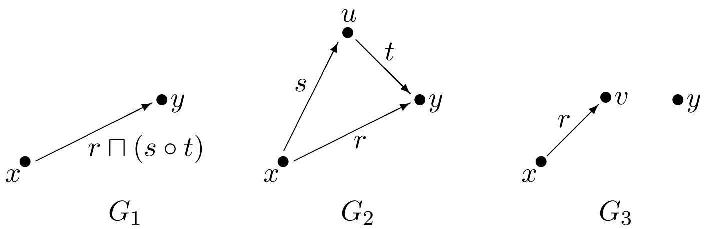

Fig. 1. One-component graphs.

Formally, we fix a set $\textsc{Inod} = \{x_n : n \in \omega\}$ of *individual nodes*, typically noted x, y, z, u, v, w. A *component* is a structure $C = (N, A, x, y)$, where N is a non-empty set of nodes, $A \subseteq N \times \mathfrak{T}^+ \times N$ is a set of labeled *arcs* ($\mathfrak{T}^+$ is the set of all +RC terms), x, y are, not necessarily distinct, distinguished nodes in N. The pair (x, y) is called the *distinguished pair* of C. Given a term R of +RC and nodes u, v, we note the arc (u, R, v) by uRv. A *positive relational graph*, or simply a *graph*, is a finite non-empty set of components (which may share nodes). We identify a component and a graph having only this component. Figure 2 shows a two-component graph, G_4.

Given a base set, considered as universe, a graph defines a binary relation on it, according to some conditions on its components. The label of an arc represents a restriction associated to the relation defined by the label. A path from a node to

G_4

Fig. 2. A two-component graph.

another one represents a restriction associated to the composition of the corresponding relations. A graph is a set of components. Each graph represents a restriction associated to the union of the relations corresponding to its components.

Formally, consider a component $C = (N, A, x, y)$ and a model $\mathfrak{M}$. An *assignment* for C in $\mathfrak{M}$ is a function $g : N \to M$ such that $(g(u), g(v)) \in [\![R]\!]_{\mathfrak{M}}$ whenever $uRv \in A$, which we denote by $g : C \to \mathfrak{M}$. The *meaning of* C *in* $\mathfrak{M}$ is the set $[\![C]\!]_{\mathfrak{M}} = \{(g(u), g(v)) \in M \times M : g : C \to \mathfrak{M}\}$. The *meaning of a graph* G, $[\![G]\!]_{\mathfrak{M}}$, is the union of the meanings of its components.

We define general notions of inclusion and equality for graphs, according to the relations they represent as follows. Let G, H be graphs of $+\mathsf{RG}$. We say that G is *included* in H, noted $\models G \sqsubseteq H$, when $[\![G]\!]_{\mathfrak{M}} \subseteq [\![H]\!]_{\mathfrak{M}}$, for every model $\mathfrak{M}$. We say that G and H are *equivalent* when $[\![G]\!]_{\mathfrak{M}} = [\![H]\!]_{\mathfrak{M}}$, for every model $\mathfrak{M}$.

3 Derivation system

We shall now present a graph relational calculus, i.e., a set of transformation rules for deriving a relational graph from another. Some rules, when applied to a graph, do not change the corresponding graph relation, while others alter the corresponding relation, transforming it into a larger one. The main idea behind the choice of the rules is to define a normal form for the graph language, and use it to prove that $\models G \sqsubseteq H$ by executing the following two major steps. First, reduce the graphs G and H to their simple normal forms SNFG and SNFH, respectively. This is accomplished by using the Introduction/Elimination rules (Table 1). Second, verify whether or not SNFH can be obtained from SNFG by a series of structural transformations (Table 2). These structural transformations are of a special kind and, as we will show, are equivalent to just one homomorphism rule (Table 3). In fact, as we will see, to obtain the completeness result we just need to show that our rules can execute the two major steps described.

3.1 Definition of the graph relational calculus

To state the transformation rules, we adopt some conventions. We note the insertion and removal of elements in a set by $+$ and $-$, respectively. We also note by $C_1 \ldots C_n$ the set $\{C_1, \ldots, C_n\}$ of components.

The *transformation rules* are given in Tables 1 and 2. Explanations follow.

All Introduction/Elimination rules and the first two structural rules can be applied in both upward and downward directions. The last two structural rules can be applied only in the downward direction.

$$\mathsf{Univ}\ \frac{N, A + u\mathsf{E}v, x, y}{N, A, x, y} \qquad \mathsf{Iden}\ \frac{N, A + u\mathsf{I}v, x, y}{\mathtt{ren}_u^v N, \mathtt{ren}_u^v A, \mathtt{ren}_u^v x, \mathtt{ren}_u^v y}$$

$$\mathsf{Conv}\ \frac{N, A + uR^{\mathsf{T}}v, x, y}{N, A + vRu, x, y} \qquad \mathsf{Int}\ \frac{N, A + uR \sqcap Sv, x, y}{N, A + uRv + uSv, x, y}$$

$$\mathsf{Uni}\ \frac{N, A + uR \sqcup Sv, x, y}{(N, A + uRv, x, y)\ \ (N, A + uSv, x, y)}$$

$$\mathsf{Comp}\ \frac{N, A + uR \circ Sv, x, y}{N + w, A + uRw + wSv, x, y},\ \text{if } w \notin N$$

Table 1
Introduction/Elimination Rules.

$$\mathsf{Splt}\ \frac{N, A, x, y}{N + u', \mathsf{spl}_{u'}^{u} A, x, y},\ \text{if } u' \notin N$$

$$\mathsf{EraN}\ \frac{N, A, x, y}{N - u, A, x, y},\ \text{if } u \text{ is isolated and } u \notin \{x, y\}$$

$$\mathsf{EraA}\ \frac{N, A, x, y}{N, A - uRv, x, y} \qquad \mathsf{AddC}\ \frac{C}{C\ \ C'}$$

Table 2
Structural Rules.

Rule Univ states that the meaning of a graph does not change by erasing an arc labeled by E from a component, leaving the rest of the graph untouched. Rule Iden states that the meaning of a graph does not change by erasing an arc $u\mathsf{I}v$ and node u, and renaming the component where they occur, redirecting arcs accordingly. This rule uses the function $\mathtt{ren}_u^v$ (rename u to v), described by the following definitions.

$$\mathtt{ren}_u^v w = \begin{cases} v & \text{if } w = u, \\ w & \text{otherwise.} \end{cases}$$

Given arbitrary set of nodes and arcs N and A, respectively, set $\mathtt{ren}_u^v N = \{\mathtt{ren}_u^v w : w \in N\}$ and $\mathtt{ren}_u^v A = \{\mathtt{ren}_u^v w R \mathtt{ren}_u^v w' : wRw' \in A\}$. Rule Conv states that the meaning of a graph does not change by replacing an arc $uR^{\mathsf{T}}v$ by vRu, inside a component where it occurs, leaving the rest of the graph untouched. Rule Int states that the meaning of a graph does not change by replacing an arc $uR \sqcap Sv$ by two others, uRv and uSv, inside a component where it occurs, leaving the rest of the graph untouched. Rule Uni states that the meaning of a graph does not change by replacing a component C_1 having occurrence of an arc $uR \sqcup Sv$, by two other components C_2 and C_3, each one of them obtained from C_1 by replacing the arc $uR \sqcup Sv$ by a new arc: uRv for C_2 and uSv for C_3, leaving the rest of the graph untouched. Finally, rule Comp states that the meaning of a graph does not change by replacing an arc $uR \circ Sv$ by two others, uRw and wSv, with a new node w, inside a component where it occurs, leaving the rest of the graph untouched. For instance, graph G_2 is obtained from graph G_1 by applying Int and Comp (down), whereas graph G_5 (in Figure 3) is obtained from G_4 by applying Univ, Comp and Uni (up).

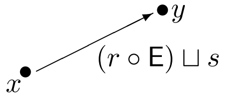

Fig. 3. Graph G_5.

Rule Splt states that the addition of a new node u' having adjacent to it the same arcs as a node u does not alter the meaning of the graph. This rule uses the function $\mathsf{spl}^u_{u'}$ (split u with u') transforming sets of arcs, defined by:

$$\mathsf{spl}^u_{u'} A = A \cup \{u'Rv : uRv \in A\} \cup \{vRu' : vRu \in A\} \cup \{u'Ru', uRu', u'Ru : uRu \in A\}$$

Rule EraN states that the meaning of a graph does not change by erasing an isolated node that is not distinguished in a component. A node is *isolated* when it is not linked to another node by an arc. Rule EraA states that the meaning of a graph obtained by erasing an arc in a component contains the meaning of the original one. Rule AddC states that the addition of components to a graph does not decrease its meaning. For instance, graph G_6 (in Figure 4) is obtained from G_2 by Splt (splitting node y to v).

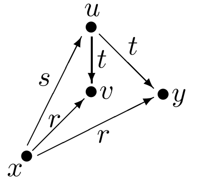

Fig. 4. Graph G_6.

Given components $C' = (N', A', x', y')$ and $C = (N, A, x, y)$ a *homomorphism* $\phi : C' \to C$ is a function $\phi : N' \to N$ for which $\phi x' = x$, $\phi y' = y$, and $\phi uR\phi v \in A$, for all $uRv \in A'$. Let G, H be graphs. We say that G *is homomorphic to* H, noted $G \leftarrow H$ when, for each component C of G, there is a component C' of H and a homomorphism $\phi : C' \to C$. For instance, the function ϕ given by $\phi(x) = x$

and $\phi(v) = \phi(y) = y$ is a homomorphism from the left component of G_4 to the component of G_2.

Rule Hom (Table 3) states that when $G \leftarrow H$, we can infer H from G.

$$\mathsf{Hom}\ \frac{G}{H},\ \text{if } G \leftarrow H$$

Table 3
Homomorphism rule.

The notion of proof is the standard one. We say that H is *derivable* from G in $+\mathsf{RG}$, noted $G \vdash H$, if there is a sequence $G_1, \ldots, G_n$ of graphs such that: (1) $G_1 = G$; (2) $G_n = H$; (3) for each $i, 1 < i \leq n$, the graph G_i is obtained from the graph G_{i-1} by application of one of the rules in Tables 1, 2 and 3.

Our set of rules is not minimal. For instance, the above homomorphism ϕ can be simulated by:

$$G_2 \xrightarrow{\mathsf{Splt}} G_6 \xrightarrow{\mathsf{EraA}} G_7 \xrightarrow{\mathsf{EraN}} G_3 \xrightarrow{\mathsf{AddC}} G_4,$$

where G_7 is the graph in Figure 5.

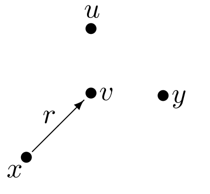

Fig. 5. Graph G_7.

Proposition 3.1 *Rule* Hom *is equivalent to the set of rules in Table 2.*

Proof. Clearly, the structural rules are instances of Hom. Conversely, let $G \leftarrow H$, then each component of G can be transformed into a component of H, by using rules Splt, EraA, and EraN. Finally, using AddC, we add the components of H not obtained by this process. $\square$

3.2 Soundness and completeness

We now examine soundness and completeness of $+\mathsf{RG}$. To this end, we implement the following strategy. First, we show that every graph can be transformed to an equivalent one in a normal form, by applications of rules in Table 1. Second, we show that the inclusion of graphs in normal form can be decided by testing the existence of a homomorphism from one graph to another. The combination of these steps will provide completeness.

Let G be a graph of $+\mathsf{RG}$. We say that G is *simple* if all its arcs are labeled by relational variables. A *simple normal form* of G is a simple graph H of $+\mathsf{RG}$ that can be obtained from G by applications of the elimination rules. In this case we write, $H = \mathrm{SNF}G$. Clearly, G and $\mathrm{SNF}G$ are equivalent. For instance, $G_2 = \mathrm{SNF}G_1$. These ideas lead to the next lemma guarantying the first step of the strategy.

Lemma 3.2 *Every graph of* $+\mathsf{RG}$ *has a simple normal form.*

The second step mentioned above can be established by constructing a (finite) canonical model. Given a component $C = (N, A, x, y)$, its *canonical model* is $\mathcal{C} = \langle N, r_i^{\mathcal{C}} \rangle_{i\in\omega}$, where $r_i^{\mathcal{C}} = \{(u, v) \in N \times N : ur_iv \in A\}$, for $i \in \omega$.

Proposition 3.3 *For simple graphs G and H, the following are equivalent:*
(a) $\models G \sqsubseteq H$,
(b) *for each component C of G, its distinguished pair (x_C, y_C) is in $[\![H]\!]_{\mathcal{C}}$,*
(c) *G is homomorphic to H.*

Proof. (a)⇒(b) is clear, as $(x_C, y_C) \in [\![G]\!]_{\mathcal{C}}$. (b)⇒(c): for some component D of H, $(x_C, y_C) \in [\![D]\!]_{\mathcal{C}}$, so we have an assignment $g : D \to \mathcal{C}$, which gives a homomorphism from D to C. (c)⇒(a) is clear, as rule Hom is sound. □

Corollary 3.4 *For graphs G and H of* $+\mathsf{RG}$*, the following are equivalent:*
(a) $G \vdash H$,
(b) $G \sqsubseteq H$,
(c) $\mathrm{SNF}G \leftarrow \mathrm{SNF}H$.

From this we obtain the following result.

Theorem 3.5 *Given graphs G and H of* $+\mathsf{RG}$.
(a) (*Soundness and Completeness*) $\models G \sqsubseteq H$ *iff* $G \vdash H$.
(b) (*Finite model property*) $\models G \sqsubseteq H$ *iff* $[\![C]\!]_{\mathcal{C}} \subseteq [\![H]\!]_{\mathcal{C}}$ *for each component C of* $\mathrm{SNF}G$ *with canonical model* $\mathcal{C}$.

Proof. (a) Lemma 3.2 and Proposition 3.3 yield (⇒) and (⇐) follows from the soundness of the rules. (b) As $[\![C]\!]_{\mathcal{C}} \subseteq [\![G]\!]_{\mathcal{C}} \subseteq [\![H]\!]_{\mathcal{C}}$, (⇒) is clear, and Proposition 3.3 yields (⇐), as $(x_C, y_C) \in [\![C]\!]_{\mathcal{C}}$. □

Corollary 3.4 also provides a normal form for proofs as in Table 4. For example, the sequence of graphs

$$G_1 \xrightarrow{\mathsf{Int;Comp}} G_2 \xrightarrow{\mathsf{Hom}} G_4 \xrightarrow{\mathsf{Univ;Comp;Uni}} G_5$$

represents the majors steps of a proof.

As a corollary we also obtain the decidability of the Validity Problem for inclusions and equalities of $+\mathsf{RG}$.

4 Expressive power

Given a model $\mathfrak{M}$, every term of $+\mathsf{RC}$, as well as every graph of $+\mathsf{RG}$, defines a binary relation on M. Given a model $\mathfrak{M}$ and $X \subseteq M \times M$, we say that X *is definable*

G	
$\downarrow$	elimination of operators (Table 1)
SNFG	
$\downarrow$	one application of Hom (Table 3)
SNFH	
$\downarrow$	introduction of operators (Table 1)
H	

Table 4
Normal form for proofs in $+\mathsf{RG}$.

in $+\mathsf{RC}$ if $X = [\![R]\!]_{\mathfrak{M}}$ for some term R of $+\mathsf{RC}$. Analogously, X *is definable in* $+\mathsf{RG}$ when $X = [\![G]\!]_{\mathfrak{M}}$ for some graph G of $+\mathsf{RG}$. We shall characterize these definable relations and investigate the exact relationship between definability in $+\mathsf{RC}$ and in $+\mathsf{RG}$.

Definability in $+\mathsf{RG}$ subsumes definability in $+\mathsf{RC}$. To prove this it suffices to associate to each term R a graph G_R that defines the same relation as the term. Let R be a term of $+\mathsf{RC}$. The *graph associated to* R is $G_R ::= (\{x, y\}, \{xRy\}, x, y)$. The next lemma is clear.

Lemma 4.1 *For every term R of $+\mathsf{RC}$, we have $[\![R]\!]_{\mathfrak{M}} = [\![G_R]\!]_{\mathfrak{M}}$, for any model $\mathfrak{M}$. Hence, given a model $\mathfrak{M}$, a relation $X \subseteq M \times M$ is definable in $+\mathsf{RC}$ only if X is definable in $+\mathsf{RG}$.*

We compare the expressive powers of $+\mathsf{RC}$ and $+\mathsf{RG}$ with that of first-order logic. For this purpose, the following version of first-order language seems to be quite adequate.

Let $\textrm{I}\textsc{var} = \{x_i : i \in \omega\}$ be a set of individual variables, typically noted x, y, z, and $\textrm{R}\textsc{var} = \{r_i : i \in \omega\}$ be a set of relational symbols, typically noted r, s, t. The *formulas* of $+\exists\mathsf{FOL}(\mathrm{R})$, typically noted φ, ψ, θ, are defined according to the following grammar:

$$\varphi ::= xry \mid x \approx y \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x\varphi.$$

The semantics for $+\exists\mathsf{FOL}(\mathrm{R})$ is just the first-order one restricted to the positive language. So, the models for $+\mathsf{RC}$, $+\mathsf{RG}$ and $+\exists\mathsf{FOL}(\mathrm{R})$ are the same. This simplifies the comparison of the expressive powers of these formalisms. We freely use all the syntactic notions, properties and conventions of first-order logic when restricted to $+\exists\mathsf{FOL}(\mathrm{R})$.

We will now characterize the expressive powers of $+\mathsf{RC}$ and $+\mathsf{RG}$ in terms of two fragments of $+\exists\mathsf{FOL}(\mathrm{R})$. Let $+\exists\mathsf{FOL}(\mathrm{R})^{xy}$ consist of the formulas of $+\exists\mathsf{FOL}(\mathrm{R})$ having at most x and y free, and let $+\exists\mathsf{FOL}(\mathrm{R})^{xy}_z$ consist of the formulas of $+\exists\mathsf{FOL}(\mathrm{R})^{xy}$ having at most x, y and z as variables.

The next result parallels the analogous one for the Tarski's relational formalism [18]. To prove it, just note the modularity present in both the forward and backward translations presented in [2].

Proposition 4.2 *There exist meaning-preserving translations from terms of* $+\mathsf{RC}$ *to formulas of* $+\exists\mathsf{FOL}(\mathrm{R})_z^{xy}$ *and vice-versa.*

Now, we show that the graph language and the positive existential first-order fragment define the same relations in any model $\mathfrak{M}$. The next result shows that the disjunctive normal form of formulas of $+\exists\mathsf{FOL}(\mathrm{R})$ are very close to graphs of $+\mathsf{RG}$ in simple normal form. We say that a graph and a first order formula are *equivalent* when they define the same relations in each model.

Theorem 4.3 *Each graph of* $+\mathsf{RG}$ *is equivalent to a formula of* $+\exists\mathsf{FOL}(\mathrm{R})^{xy}$*, and conversely, each formula of* $+\exists\mathsf{FOL}(\mathrm{R})^{xy}$ *is equivalent to a graph of* $+\mathsf{RG}$*.*

Proof. We generalize the notion of components: $L = (N, A, z_1, \ldots, z_n)$ with $z_1, \ldots, z_n$ in N as distinguished nodes with meaning the expected n-ary relation $[\![L]\!]_{\mathfrak{M}}$. This meaning can be defined by an existentially quantified conjunction of atomic formulas of $+\exists\mathsf{FOL}(\mathrm{R})^{xy}$. Conversely, given a disjunction-free formula φ of $+\exists\mathsf{FOL}(\mathrm{R})$, we construct a generalized component that defines the same relation as φ in each model, by induction on the structure of φ. $\square$

5 Decidability of the positive relational calculus

An *inclusion of* $+\mathsf{RC}$ is an expression of the form $R \sqsubseteq S$, where R, S are terms of $+\mathsf{RC}$. An inclusion $R \sqsubseteq S$ is *valid*, noted $\models R \sqsubseteq S$, when $[\![R]\!]_{\mathfrak{M}} \subseteq [\![S]\!]_{\mathfrak{M}}$ for every model $\mathfrak{M}$. In this section, $+\mathsf{RG}$ will be used to decide the valid inclusions of $+\mathsf{RC}$. The idea is to derive an inclusion $R \sqsubseteq S$ by using the corresponding graphs G_R and G_S. The work of R. Lyndon [12,13] yields that the inclusion $R \sqsubseteq S$, where R and S are, respectively, the following $+\mathsf{RC}$ terms:

$$p \sqcap (((q \circ r) \sqcap s) \circ (t \sqcap (a \circ b))) \text{ and}$$

$$q \circ ((((q^{\mathsf{T}} \circ p) \sqcap (r \circ t)) \circ b^{\mathsf{T}}) \sqcap (r \circ a) \sqcap (q^{\mathsf{T}} \circ ((p \circ b^{\mathsf{T}}) \sqcap (s \circ a)))) \circ b$$

although valid is not derived in the relational formalism, from the Tarski's axioms [17]. Within the graph calculus, this inclusion can be proved, because $G_R \vdash G_S$ (since $\mathrm{SNF}G_R \leftarrow \mathrm{SNF}G_S$). In general, we have soundness and completeness for $+\mathsf{RC}$.

Theorem 5.1 *For terms* R, S *of* $+\mathsf{RC}$*, we have* $\models R \sqsubseteq S$ *iff* $G_R \vdash G_S$*.*

The non-finite axiomatizability of the valid inclusions of $+\mathsf{RC}$ is a consequence of a general result of H. Andréka [1]. This does not preclude infinite axiomatizations and, in fact, the existence of a set of positive axioms follows from a result of B. M. Schein [14]. To the best of our knowledge, no explicit infinite set of axioms to $+\mathsf{RC}$ has been exhibited. Paralleling the results of B. Jónsson [11], we believe that no such a set should be described in simple terms. The quest for axiomatizability of $+\mathsf{RC}$ and some of its subreducts is one of the problems stated in [15]. We can combine the results of Section 3 with that reported in [6] to exhibit a set of equational axioms for +RC. Hence, axiomatization and decidability of +RC follows from our work in +RG.

6 Conclusion

We have given a formal treatment to +RG, a relational calculus based on graphs, presenting soundness and completeness results for the valid inclusions and obtaining a finite model property. We have also compared the expressive power of +RG to a first-order language fragment showing that both define the same binary relations. One may regard +RG as a notational variant of the positive existential first-order logic of binary relations. This perspective leaves open the possibility of developing positive first-order logic as graph calculus.

Our study of the graph calculus opens up several interesting problems. It is easy to extend +RG to deal with the empty relation. An interesting problem concerns the extension of +RG to deal with complementation. Another interesting problem is to make a deeper comparison between the role played by our rules and the ones presented in [5].

References

[1] H. Andréka. Representations of distributive lattice-ordered semigroups with binary relations. *Algebra Univers.*, 28:12–25, 1991.

[2] H. Andréka, I. Németi, and I. Sain. Algebraic logic. Technical report, 2003. http://www.math-inst.hu/pub/algebraic-logic/Contents.html.

[3] C. Brown and G. Hutton. Categories, allegories and circuit design. In *Proc. Ninth Annual IEEE Symposium on Logic in Computer Science*, 372–381, IEEE Computer Society Press, 1994.

[4] D. Cantone, A. Formisano, E.G. Omodeo, and C.G. Zarba. Compiling dyadic specifications into map algebra. *Theoret. Comput. Sci.*, 293:447–475, 2003.

[5] S. Curtis and G. Lowe. Proofs with graphs. *Sci. Comput. Programming*, 26:197–216, 1996.

[6] D. Dougherty and C. Gutiérrez. Normal forms and reduction theories of binary relations. In *Proc. LNCS 1833*, 95–109, Springer, 2000.

[7] A. Formisano, E.G. Omodeo, and M. Simeoni. A graphical approach to relational reasoning. *ENTCS*, 44:1–22, 2003.

[8] P.J. Freyd and A. Scedrov. *Categories, Allegories*. North-Holland, Amsterdam, 1990.

[9] C. Gutiérrez. Normal forms for connectedness in categories. *Ann. Pure Appl. Logic*, 108:237–247, 2001.

[10] G. Hutton. A relational derivation of a functional program. In *Lecture Notes of the STOP Summer School on Constructive Algorithms*, 1992.

[11] B. Jónsson. Representation of modular lattices and of relation algebras. *Trans. Amer. Math. Soc.*, 92:449–464, 1959.

[12] R.C. Lyndon. The representation of relational algebras. *Annals Math.*, 51:707–729, 1950.

[13] R.C. Lyndon. The representation of relational algebras, II. *Annals Math.*, 63:294–307, 1956.

[14] B.M. Schein. Relation algebras and function semigroups. *Semigroup Forum*, 1:1–62, 1970.

[15] B.M. Schein. Representation of subreducts of Tarski relation algebras. In *Algebraic Logic*, volume 54 of *Colloquia Mathematica Societatis János Bolyai*, 621–635, North-Holland, Amsterdam, 1991.

[16] G. Schmidt and T. Ströhlein. *Relation and Graphs: Discrete Mathematics for Computer Scientists*. Springer-Verlag, Berlin, 1993.

[17] A. Tarski. On the calculus of relations. *J. Symbolic Logic*, 6:73–89, 1941.

[18] A. Tarski and S. Givant. *A Formalization of Set Theory without Variables*. American Mathematical Society, Providence, Rhode Island, 1987.