# On Graph Refutation for Relational Inclusions*

Paulo A. S. Veloso
COPPE-UFRJ
Systems and Computer Engin. Program
UFRJ: Federal University of Rio de Janeiro
RJ, Brazil
pasveloso@gmail.com

Sheila R. M. Veloso
FEN-UERJ
Systems and Computer Engin. Dept., Fac. of Engineering
UERJ: State University of Rio de Janeiro
RJ , Brazil
sheila.murgel.bridge@gmail.com

We introduce a graphical refutation calculus for relational inclusions: it reduces establishing a relational inclusion to establishing that a graph constructed from it has empty extension. This sound and complete calculus is conceptually simpler and easier to use than the usual ones.

## 1 Introduction

We introduce a sound and complete goal-oriented graph calculus for relational inclusions.[1] Though somewhat richer, it is conceptually simpler and easier to use than the usual ones, as is its extension for handling hypotheses, due to goal-orientation.

Diagrams and figures are very important and useful in several branches of science, as well as in everyday life. Graphs and diagrams provide convenient visualization in many areas [1, 3, 4, 15, 17]. The heuristic appeal of diagrams is evident. Venn diagrams, for instance, may be very helpful in visualizing connections between sets. They are not, however, usually accepted as proofs: one has to embellish the connections discovered in terms of standard methods of reasoning. This is not the case with our graph calculi: there is no need to compile the steps into standard reasoning. Graph manipulations, provided with precise syntax and semantics, are proof methods.

Formulas are usually written down on a single line [6]. While the Polish parenthesis-free notation is more economical, the usual notation is more readable: e.g. compare $\rightarrow \wedge pq \vee rs$ and $(p \wedge q) \rightarrow (r \vee s)$. A basic idea behind graph calculi is a two-dimensional representation: e.g. the structure of $(x+y)\cdot(z-w)$ is more apparent in the notation $\begin{pmatrix} x \\ + \\ y \end{pmatrix} \cdot \begin{pmatrix} z \\ - \\ w \end{pmatrix}$ (see also [1]). Using (individual) nodes in graph calculi is crucial, as well (see Sections 2 and 3).

Using drawings for relations is a natural idea: represent the fact that $a$ is related to $b$ via relation $r$ by an arrow $a \xrightarrow{r} b$. Then, some operations on relations correspond to simple manipulations on arrows, e.g. transposal to arrow reversal, intersection to parallel arcs and relative product to consecutive arcs (see Example 2.1). So, one can reason about relations by manipulating their representations. This is a key idea underlying graph methods for reasoning about relations [5, 6, 7, 8, 9, 10, 11, 12]. Some relational operations (like complementation) are not so easy to handle.[2] In this paper, we intend to show that one can profit from complementation by proposing a refutational graph calculus for reasoning about relations: this goal-orientated calculus, having simpler concepts, is easier to use than the usual ones.

*Research partly sponsored by the Brazilian agencies CNPq and FAPERJ.

[1] Discussions with Petrucio Viana and Renata de Freitas are gratefully acknowledged.

[2] Complementation may be introduced by definition, if one can reason from hypotheses [9], or it can be handled via arcs labeled by boxes [12].

Ronchi della Rocca, S., Pimentel, E. (Eds.): 6th Workshop on
Logical and Semantic Frameworks with Applications 2011 (LSFA 2011).
EPTCS 81, 2012, pp. 47–62, doi:10.4204/EPTCS.81.4

The structure of this paper is as follows. In Section 2, we illustrate the ideas underlying our calculus for relational inclusions. In Section 3, we examine our graph language: syntax, semantics as well as some concepts and constructions. In Section 4, we introduce our refutation calculus and its rules, which we extend to handle inclusion hypotheses in Section 5. Finally, Section 6 presents some remarks about our approach and other relational calculi.

## 2 Motivation: underlying ideas

We now examine some basic ideas underlying our calculus for relational inclusions.

We wish to establish inclusions between relational terms. Relational terms are expressions like r, $\overline{\mathrm{s}}$, $\mathrm{r}\sqcap\overline{\mathrm{s}}$ and $\mathrm{r}^{\smile};\overline{(\mathrm{r};\mathrm{s})}$. The relational terms are (freely) generated from relation names by relational constants and operations, as usual [16]. We employ the RelMiCs notation [2].

- A relation name $\mathrm{r},\mathrm{s},\mathrm{t},\ldots$ corresponds to an arbitrary binary relation (over a set $M$).
- The constants $\mathbb{L}$, $\mathbb{T}$, $\mathbb{I}$ and $\mathbb{D}$ denote respectively the following 2-ary relations: empty $\emptyset$, square $M^2 := M\times M$, identity $I_M := \{(a,b)\in M^2 \,/\, a=b\}$ and diversity $I_M\tilde{}:= \{(a,b)\in M^2 \,/\, a\neq b\}$.
- The unary operations $^{-}$ and $^{\smile}$ stand for Boolean complementation $\tilde{}$ and Peircean transposition $^{\mathsf{T}}$. Recall that $\mathrm{R}\tilde{}:= \{(a,b)\in M^2 \,/\, (a,b)\notin \mathrm{R}\}$ and $\mathrm{R}^{\mathsf{T}} := \{(a,b)\in M^2 \,/\, (b,a)\in \mathrm{R}\}$.
- The binary operations $\sqcap$ and $\sqcup$ stand for Boolean intersection $\cap$ and union $\cup$, respectively. The binary operations ; and † stand for relative product $|$ and sum $\underline{|}$, respectively. For $(a,b)\in M^2$, we have: $(a,b)\in \mathrm{P}\,|\,\mathrm{Q}$ iff, for some $c\in M$, $(a,c)\in\mathrm{P}$ and $(c,b)\in\mathrm{Q}$, and $(a,b)\in \mathrm{P}\,\underline{|}\,\mathrm{Q}$ iff, for every $c\in M$, $(a,c)\in\mathrm{P}$ or $(c,b)\in\mathrm{Q}$.

We can now introduce the ideas of our graph methods (see also Sections 3 and 4). A graph is a finite set of alternative slices. A slice consists of finite sets of nodes and labeled arcs together with 2 distinguished nodes (marked $\rightarrow$). To establish an inclusion $\mathrm{P}\sqsubseteq\mathrm{Q}$ we start with the slice corresponding to $\mathrm{P}\sqcap\overline{\mathrm{Q}}$ and apply the rules so as to obtain a graph whose slices are inconsistent.

We now examine some simple examples illustrating our graph methods (see also Sections 3 and 4).

**Example 2.1.** *To establish* $\mathrm{P}^{\smile};\overline{\mathrm{P};\mathrm{Q}}\sqsubseteq\overline{\mathrm{Q}}$, *we show* $(\mathrm{P}^{\smile};\overline{\mathrm{P};\mathrm{Q}})\sqcap\overline{\overline{\mathrm{Q}}}\sqsubseteq\mathbb{L}$.

1. *First, we form a slice for* $(\mathrm{P}^{\smile};\overline{\mathrm{P};\mathrm{Q}})\sqcap\overline{\overline{\mathrm{Q}}}$, *with parallel arcs:* $\mathsf{S}_0 :=$ $\begin{tikzcd} \rightarrow \mathrm{x} \arrow[r, bend left, "\mathrm{P}^{\smile};\overline{\mathrm{P};\mathrm{Q}}"] \arrow[r, bend right, "\overline{\overline{\mathrm{Q}}}"'] & \mathrm{y}\rightarrow \end{tikzcd}$ .
2. *We now convert this slice* $\mathsf{S}_0$ *to a special form, as follows.*
   (a) *We eliminate double complementation, converting* $\mathsf{S}_0$ *to* $\mathsf{S}_1$ *as follows:*

$$\begin{tikzcd} \rightarrow \mathrm{x} \arrow[r, bend left, "\mathrm{P}^{\smile};\overline{\mathrm{P};\mathrm{Q}}"] \arrow[r, bend right, "\mathrm{Q}"'] & \mathrm{y}\rightarrow \end{tikzcd}$$

   (b) *Next, we eliminate* ; *by converting* $\mathsf{S}_1$ *to a* 3*-node slice* $\mathsf{S}_2$ *as follows:*

$$\begin{tikzcd} \rightarrow \mathrm{x} \arrow[r, "\mathrm{P}^{\smile}"] \arrow[dr, "\mathrm{Q}"'] & \mathrm{z} \arrow[d, "\overline{\mathrm{P};\mathrm{Q}}"] \\ & \mathrm{y}\rightarrow \end{tikzcd}$$

(c) *We eliminate* $\smile$ *from* $\mathsf{S}_2$*, inverting its arrow and giving* $\mathsf{S}_3$ *as follows:*

$$\begin{tikzcd} \to x \arrow[dr, "Q"'] & z \arrow[l, "P"'] \arrow[d, "\overline{P;Q}"] \\ & y\to \end{tikzcd}$$

(d) *We now convert* $\mathsf{S}_3$ *to* $\mathsf{S}_4$ *(with a complemented slice as arc label):*

$$\begin{tikzcd} \to x \arrow[dr, "Q"'] & z \arrow[l, "P"'] \arrow[d, "\overline{\boxed{\to z' \xrightarrow{P} x' \xrightarrow{Q} y' \to}}"] \\ & y\to \end{tikzcd}$$

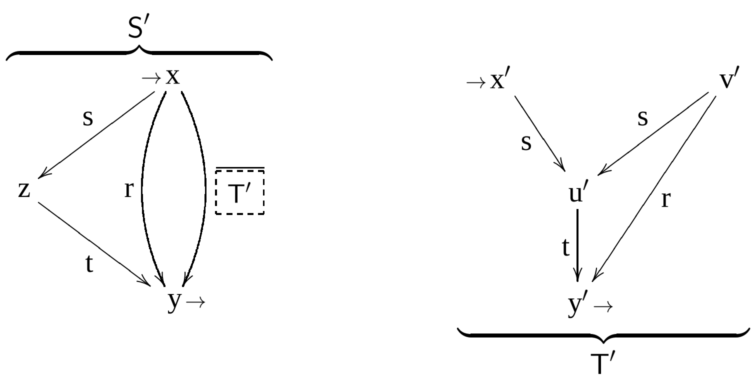
3. *Now, within slice* $\mathsf{S}_4$*, we have the following (parallel) paths from* z *to* y*:*
   - *positive path* $z \xrightarrow{P} x \xrightarrow{Q} y$ *(corresponding to the term* P;Q*) and*
   - *negative path* $z \xrightarrow{\overline{\boxed{\to z' \xrightarrow{P} x' \xrightarrow{Q} y' \to}}} y$ *(corresponding to the term* $\overline{P;Q}$*).*

   *Slice* $\mathsf{S}_4$ *represents an inconsistent situation, corresponding to the empty relation* $\emptyset$*.*

**Example 2.2.** *Consider the modular law* $P \sqsubseteq Q$*, where* $P := r \sqcap (s;t)$ *and* $Q := s;[(s^{\smile};r) \sqcap t]$ *(cf. [11]). We reduce it to* $P \sqcap \overline{Q} \sqsubseteq \mathbf{\perp}$*, which we can establish as follows (see Sections 3 and 4 for some details).*

1. *As before, we construct a slice with parallel arcs, namely* $\mathsf{S} := \begin{tikzcd} \to x \arrow[r, bend left, "P"] \arrow[r, bend right, "\overline{Q}"'] & y\to \end{tikzcd}$ .

2. *We can convert slice* $\mathsf{S}$ *(see 3.1 and 4.1) to an equivalent slice* $\mathsf{S}'$ *with slice* $\overline{\mathsf{T}'}$ *as arc label, where:*




3. *Now consider the mapping* $\theta$*, given by* $x', v' \mapsto x$*;* $u' \mapsto z$*;* $y' \mapsto y$*. It maps arcs of* $\mathsf{T}'$ *to arcs of* $\mathsf{S}'$*. Slice* $\mathsf{S}'$ *is inconsistent, corresponding to the empty relation* $\emptyset$ *(see 3.2). Informally speaking, in* $\mathsf{S}'$ *we find an image of* $\mathsf{T}'$ *in parallel with* $\overline{\mathsf{T}'}$*. Thus, we have the inclusion* $\mathsf{S}' \sqsubseteq \mathbf{\perp}$*, whence also the inclusions* $\mathsf{S} \sqsubseteq \mathbf{\perp}$*,* $P \sqcap \overline{Q} \sqsubseteq \mathbf{\perp}$ *and* $P \sqsubseteq Q$*.*

We will be able to convert every relational term to a graph (see 3.1 and 4.1). Consider, however, the following two slices $S'$ and $S''$:

$$\begin{tikzcd} & u \arrow[dr, "r"] & \\ \to x \arrow[ur, "p"] \arrow[dr, "q"'] & & y \to \\ & v \arrow[ur, "s"'] & \end{tikzcd} \qquad \begin{tikzcd} & u \arrow[dr, "r"] \arrow[dd, "t"] & \\ \to x \arrow[ur, "p"] \arrow[dr, "q"'] & & y \to \\ & v \arrow[ur, "s"'] & \end{tikzcd}$$

Slice $S'$ corresponds to the term $\begin{bmatrix} (p;r) \\ \sqcap \\ (q;s) \end{bmatrix}$, but one does not have a term corresponding to slice $S''$. So, graphs will turn out to be more expressive than relational terms.

# 3 Graph Language

We now introduce our graph language: syntax and semantics (in 3.1) and some constructions (in 3.2). Labels, slices and graphs will represent binary relations, whereas arcs will represent restrictions.

We will consider two fixed denumerably infinite sets: set $\mathbb{R}n$ of *relation names* and set $\mathbb{N}d$ of (individual) *nodes* (in alphabetical order: $x, y, z, \ldots$).

## 3.1 Syntax and semantics

We now examine the syntax and semantics of our graph language.

We introduce the syntax of our graph concepts (by mutual recursion).

**(L)** The *labels* are (freely) generated from the relation names, slices and graphs (see below), by relational operations and constants.

**(a)** An *arc* over a set $N \subseteq \mathbb{N}d$ is a triple $u\,L\,v$, where $u, v \in N$ and $L$ is a label.

**(Σ)** A *sketch* $\Sigma = \langle N, A \rangle$ consists of 2 sets: $N \subseteq \mathbb{N}d$ of nodes and $A$ of arcs over $N$.

**(D)** A *draft* $D$ is a sketch with finite sets of nodes and of arcs.

**(S)** A *slice* $S = \langle N, A : x_S, y_S \rangle$ consists of a draft $\underline{S} = \langle N, A \rangle$ (its *underlying draft*) together with a pair of distinguished nodes $x_S, y_S \in N$ (its *input* and *output* nodes). For instance, in Example 2.1, we have the slice $S_2 = \langle \{x, y, z\}, \{x\,P^{\smile}\,z, z\,\overline{P;Q}\,y, x\,Q\,y\} : x, y \rangle$.

**(G)** A graph is a finite set of slices. Example 4.4 (in 4.2) will show a 2-slice graph $G = \{S_+, S_-\}$.

The *empty graph* $\{\ \}$ has no slice. Note that every relational term is a label, as are slices and graphs. Drafts, slices and graphs are finite objects, whereas sketches are useful in some arguments (cf. 4.2).

An *inclusion* is a pair of labels, noted $L \sqsubseteq K$. The *difference slice* of a pair of labels $L$ and $K$ is the 2-arc slice $DS(L \setminus K) := \langle \{x, y\}, \{x\,L\,y, x\,\overline{K}\,y\} : x, y \rangle$ (where $x$ and $y$ are the first 2 nodes in $\mathbb{N}d$). The difference slice $DS(L \setminus K)$ has 2 parallel arcs: $\begin{tikzcd} \to x \arrow[r, bend left, "L"] \arrow[r, bend right, "\overline{K}"'] & y \to \end{tikzcd}$ (cf. Examples 2.1 and 2.2 in Section 2).

We now examine the semantics of our graph language. We use models for semantics: a model assigns a binary relation to each relation name. A *model* is a structure $\mathfrak{M} = \langle M, (r^{\mathfrak{M}})_{r \in \mathrm{R}n} \rangle$, consisting of a set $M$ and a binary relation $r^{\mathfrak{M}}$ on $M$, i.e. $r^{\mathfrak{M}} \subseteq M^2$, for each relation name $r \in \mathrm{R}n$. An *M-assignment* for a set $N \subseteq \mathbb{N}\mathrm{d}$ of nodes is a function $\mathrm{g} : N \to M$, assigning an element $\mathrm{w}^{\mathrm{g}} \in M$ to each node $\mathrm{w} \in N$.

We now introduce the semantics of our graph concepts (again by mutual recursion). Consider a given *M-model* $\mathfrak{M} = \langle M, (r^{\mathfrak{M}})_{r \in \mathrm{R}n} \rangle$.

**(L)** The *relation of label* L is the relation $[\mathrm{L}]_{\mathfrak{M}} \subseteq M^2$ obtained by extending the relations of the relation names by means of the concrete versions of the operations. More precisely, the relation of a label is the binary relation on $M$ defined as follows.

- **(0)** For a relation name $r$: $[r]_{\mathfrak{M}} := r^{\mathfrak{M}}$ (as given by model $\mathfrak{M}$). For the constants, we set $[\mathbb{L}]_{\mathfrak{M}} := \emptyset$, $[\mathbb{T}]_{\mathfrak{M}} := M^2$, $[\mathbb{I}]_{\mathfrak{M}} := I_M$ and $[\mathbb{D}]_{\mathfrak{M}} := I_M{}^{\sim}$. For a slice or a graph, we employ their extensions, namely: $[\mathsf{S}]_{\mathfrak{M}} := [\![S]\!]_{\mathfrak{M}}$ and $[\mathsf{G}]_{\mathfrak{M}} := [\![G]\!]_{\mathfrak{M}}$ (as defined below).
- **(1)** For the unary operations $^{-}$ and $^{\smile}$, we have Boolean complementation $^{\sim}$and Peircean transposition $^{\mathsf{T}}$, respectively; so we set $[\overline{\mathrm{L}}]_{\mathfrak{M}} := [\mathrm{L}]_{\mathfrak{M}}{}^{\sim}$and $[\mathrm{L}^{\smile}]_{\mathfrak{M}} := [\mathrm{L}]_{\mathfrak{M}}{}^{\mathsf{T}}$.
- **(2)** For the binary operations $\sqcap$, $\sqcup$, ; and †, we have intersection, union, relative product and relative sum, respectively; so we set $[\mathrm{L}\sqcap\mathrm{K}]_{\mathfrak{M}} := [\mathrm{L}]_{\mathfrak{M}} \cap [\mathrm{K}]_{\mathfrak{M}}$, $[\mathrm{L}\sqcup\mathrm{K}]_{\mathfrak{M}} := [\mathrm{L}]_{\mathfrak{M}} \cup [\mathrm{K}]_{\mathfrak{M}}$, $[\mathrm{L};\mathrm{K}]_{\mathfrak{M}} := [\mathrm{L}]_{\mathfrak{M}} \mid [\mathrm{K}]_{\mathfrak{M}}$ and $[\mathrm{L}\dagger\mathrm{K}]_{\mathfrak{M}} := [\mathrm{L}]_{\mathfrak{M}} \underline{\mid} [\mathrm{K}]_{\mathfrak{M}}$.

**(a)** An $M$-assignment $\mathrm{g} : N \to M$ *satisfies* an arc $\mathrm{u}\,\mathrm{L}\,\mathrm{v}$ in $\mathfrak{M}$ (noted $\mathrm{g} \Vdash_{\mathfrak{M}} \mathrm{u}\,\mathrm{L}\,\mathrm{v}$) iff the pair of values $\mathrm{u}^{\mathrm{g}}$ and $\mathrm{v}^{\mathrm{g}}$ belongs to the relation of the label, i.e. $\mathrm{u}, \mathrm{v} \in N$ and $(\mathrm{u}^{\mathrm{g}}, \mathrm{v}^{\mathrm{g}}) \in [\mathrm{L}]_{\mathfrak{M}}$.

**(Σ)** An assignment $\mathrm{g} : N \to M$ *satisfies* a sketch $\Sigma = \langle N_\Sigma, A_\Sigma \rangle$ in $\mathfrak{M}$ (noted $\mathrm{g} : \Sigma \to \mathfrak{M}$) iff it satisfies all its arcs, i.e. $\mathrm{g} \Vdash_{\mathfrak{M}} \mathsf{a}$, for every arc $\mathsf{a} \in A_\Sigma$.

**(S)** The *extension* of a slice $\mathsf{S} = \langle \underline{\mathsf{S}} : \mathrm{x}_{\mathsf{S}}, \mathrm{y}_{\mathsf{S}} \rangle$ is the binary relation on $M$ consisting of the pair of values of $\mathrm{x}_{\mathsf{S}}$ and $\mathrm{y}_{\mathsf{S}}$ for the assignments satisfying its underlying draft $\underline{\mathsf{S}}$, namely:

$$[\![\mathsf{S}]\!]_{\mathfrak{M}} := \{(\mathrm{x}_{\mathsf{S}}{}^{\mathrm{g}}, \mathrm{y}_{\mathsf{S}}{}^{\mathrm{g}}) \in M^2 \,/\, \mathrm{g} : \underline{\mathsf{S}} \to \mathfrak{M}\}.$$

**(G)** The *extension* of a graph G is the union of the extensions of its slices: $[\![\mathsf{G}]\!]_{\mathfrak{M}} := \bigcup_{\mathsf{S} \in \mathsf{G}} [\![\mathsf{S}]\!]_{\mathfrak{M}}$.

**Remark 3.1.** *A slice* S *has non-empty extension in an M-model* $\mathfrak{M}$ *iff some M-assignment satisfies (in* $\mathfrak{M}$*) its underlying draft* $\underline{\mathsf{S}}$.

An inclusion $\mathrm{L} \sqsubseteq \mathrm{K}$ *holds* in model $\mathfrak{M}$ (noted $\mathfrak{M} \models \mathrm{L} \sqsubseteq \mathrm{K}$) iff $[\mathrm{L}]_{\mathfrak{M}} \subseteq [\mathrm{K}]_{\mathfrak{M}}$. An inclusion is *valid* iff it holds in every model. For instance, the inclusions $\mathrm{P}^{\smile};\overline{\mathrm{P};\mathrm{Q}} \sqsubseteq \overline{\mathrm{Q}}$, $(\mathrm{P}^{\smile};\overline{\mathrm{P};\mathrm{Q}})\sqcap\overline{\overline{\mathrm{Q}}} \sqsubseteq \mathbb{L}$ and $\mathsf{S}_i \sqsubseteq \mathbb{L}$ (for $i = 0, 1, \ldots, 4$) in Example 2.1 are all valid. Label L is *null* iff it the inclusion $\mathrm{L} \sqsubseteq \mathbb{L}$ is valid. Clearly, the empty graph $\{\ \}$ (with no slice) and the constant $\mathbb{L}$ are null. Labels L and K are *equivalent* (noted $\mathrm{L} \equiv \mathrm{K}$) iff both inclusions $\mathrm{L} \sqsubseteq \mathrm{K}$ and $\mathrm{K} \sqsubseteq \mathrm{L}$ are valid. For instance, in Example 2.1, all slices $\mathsf{S}_0$ through $\mathsf{S}_4$ are equivalent labels. A slice S and a singleton graph $\{\mathsf{S}\}$ are equivalent, so one may identify them.

**Lemma 3.1.** *An inclusion* $\mathrm{L} \sqsubseteq \mathrm{K}$ holds *in a model* $\mathfrak{M}$ *(*$\mathfrak{M} \models \mathrm{L} \sqsubseteq \mathrm{K}$*) iff the difference slice* $\mathsf{DS}(\mathrm{L} \setminus \mathrm{K})$ *has empty extension in* $\mathfrak{M}$ *(*$[\![\mathsf{DS}(\mathrm{L} \setminus \mathrm{K})]\!]_{\mathfrak{M}} = \emptyset$*).*

*Proof.* The difference slice has extension $[\![\mathsf{DS}(\mathrm{L} \setminus \mathrm{K})]\!]_{\mathfrak{M}} = [\mathrm{L}]_{\mathfrak{M}} \setminus [\mathrm{K}]_{\mathfrak{M}}$. □

**Corollary 3.1.** *An inclusion* $\mathrm{L} \sqsubseteq \mathrm{K}$ *holds in an M-model iff no M-assignment satisfies the underlying draft of the difference slice* $\mathsf{DS}(\mathrm{L} \setminus \mathrm{K})$.

*Proof.* By Remark 3.1 and Lemma 3.1. □

## 3.2 Concepts and constructions

We will now examine some concepts and constructions.

We use the notation '$+$' for adding arcs to a sketch or to a slice. Given an arc $\mathrm{u\,L\,v}$: for a sketch $\Sigma = \langle N, A\rangle$, $\Sigma + \mathrm{u\,L\,v} := \langle N \cup \{\mathrm{u},\mathrm{v}\}, A \cup \{\mathrm{u\,L\,v}\}\rangle$; for a slice $\mathsf{S}$, $\mathsf{S} + \mathrm{u\,L\,v} := \langle \underline{\mathsf{S}} + \mathrm{u\,L\,v} : \mathrm{x_S}, \mathrm{y_S}\rangle$.

We now introduce morphisms for comparing sketches.

Consider sketches $\Sigma' = \langle N', A'\rangle$ and $\Sigma'' = \langle N'', A''\rangle$. A node renaming function $\theta : N' \to N''$ is a *morphism* from $\Sigma'$ to $\Sigma''$ (noted $\theta : \Sigma' \dashrightarrow \Sigma''$) iff it preserves arcs: for every arc $\mathrm{u\,L\,v} \in A'$, $\mathrm{u}^\theta\,\mathrm{L}\,\mathrm{v}^\theta$ is an arc in $A''$. For instance, Example 2.2 (in Section 2) shows a morphism $\theta : \underline{\mathsf{T}'} \dashrightarrow \underline{\mathsf{S}'}$. We will use $\mathsf{Mor}[\Sigma', \Sigma'']$ for the *set of morphisms* from $\Sigma'$ to $\Sigma''$.

Morphisms transfer satisfying assignments by composition.

**Lemma 3.2.** *Given a morphism $\theta : \Sigma' \dashrightarrow \Sigma''$ and a model $\mathfrak{M}$, for every assignment $\mathrm{g}$ satisfying $\Sigma''$ in model $\mathfrak{M}$, the composite $\mathrm{g}\cdot\theta$ is an assignment satisfying $\Sigma'$ in model $\mathfrak{M}$.*

*Proof.* For every arc $\mathrm{u\,L\,v} \in A_{\Sigma'}$, we have $\mathrm{u}^\theta\,\mathrm{L}\,\mathrm{v}^\theta \in A_{\Sigma''}$, so $(\mathrm{u}^{\mathrm{g}\cdot\theta}, \mathrm{v}^{\mathrm{g}\cdot\theta}) \in [\mathrm{L}]_{\mathfrak{M}}$. □

A sketch $\Sigma$ is *zero* iff, for some slice $\mathsf{T} = \langle \underline{\mathsf{T}} : \mathrm{x_T}, \mathrm{y_T}\rangle$, there exists a morphism $\theta : \underline{\mathsf{T}} \dashrightarrow \Sigma$, such that ${\mathrm{x_T}}^\theta\,\overline{\mathsf{T}}\,{\mathrm{y_T}}^\theta$ is an arc of $\Sigma$. A slice is *zero* iff its underlying draft is zero. For instance, in Example 2.2, draft $\underline{\mathsf{S}'}$ is a zero sketch and slice $\mathsf{S}'$ is a zero slice. A *zero graph* is a graph consisting of zero slices.

**Lemma 3.3.** *No assignment can satisfy a zero sketch.*

*Proof.* By Lemma 3.2. If $\mathrm{g} : \Sigma \to \mathfrak{M}$, then we have $\mathrm{g}\cdot\theta : \underline{\mathsf{T}} \to \mathfrak{M}$ (thus $({\mathrm{x_T}}^{\mathrm{g}\cdot\theta}, {\mathrm{y_T}}^{\mathrm{g}\cdot\theta}) \in [\![\mathsf{T}]\!]_{\mathfrak{M}}$) and $\mathrm{g} \Vdash_{\mathfrak{M}} {\mathrm{x_T}}^\theta\,\overline{\mathsf{T}}\,{\mathrm{y_T}}^\theta$ (whence $({\mathrm{x_T}}^{\mathrm{g}\cdot\theta}, {\mathrm{y_T}}^{\mathrm{g}\cdot\theta}) \notin [\mathsf{T}]_{\mathfrak{M}}$), giving a contradiction. □

Zero graphs have empty extensions in every model, thus being null.

**Corollary 3.2.** *A zero graph $\mathsf{H}$ is null: $[\![\mathsf{H}]\!]_{\mathfrak{M}} = \emptyset$, for every model $\mathfrak{M}$.*

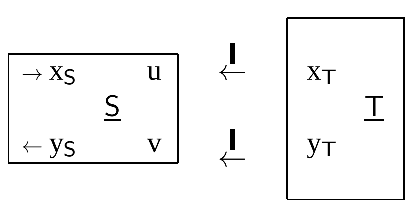
*Proof.* By Remark 3.1 (in 3.1) and Lemma 3.3. If $[\![\mathsf{H}]\!]_{\mathfrak{M}} \neq \emptyset$, then $[\![\mathsf{T}]\!]_{\mathfrak{M}} \neq \emptyset$, for some slice $\mathsf{T} \in \mathsf{H}$, whence some $M$-assignment satisfies the underlying draft $\underline{\mathsf{T}}$. □

We call a model $\mathfrak{M} = \langle M, (r^{\mathfrak{M}})_{r\in \mathrm{R}n}\rangle$ *natural* for a sketch $\Sigma = \langle N_\Sigma, A_\Sigma\rangle$ iff $M = N$ and, for each $r \in \mathrm{R}n$, $r^{\mathfrak{M}} = \{(\mathrm{w},\mathrm{z}) \in M^2 \,/\, \mathrm{w}\,r\,\mathrm{z} \in A\}$. For instance, a natural model $\mathfrak{M}$ for draft $\underline{\mathsf{S}'}$ (in Example 2.2 in Section 2) has $M = \{\mathrm{x},\mathrm{y},\mathrm{z}\}$, $\mathrm{r}^{\mathfrak{M}} = \{(\mathrm{x},\mathrm{y})\}$, $\mathrm{s}^{\mathfrak{M}} = \{(\mathrm{x},\mathrm{z})\}$ and $\mathrm{t}^{\mathfrak{M}} = \{(\mathrm{z},\mathrm{y})\}$. Natural models will be used for establishing completeness (in 4.2).

We will now examine some constructions: co-limits and pushouts [13].

We wish to glue a slice $\mathsf{T}$ onto a slice $\mathsf{S}$ via a designated pair of nodes. One can do this as follows.

1. First, use identity arcs to connect the input and output nodes of $\mathsf{T}$ to the designated nodes. One then obtains a slice with the following aspect:




2. Now, eliminate the identity arcs to obtain the glued slice $\mathsf{S}\frac{\mathrm{u}}{\mathrm{v}}\mathsf{T}$.

We now illustrate this construction.

One can eliminate an arc $w\,\mathbf{I}\,z$ from a slice $\mathsf{S}$ by renaming w to z (or z to w) throughout in $\mathsf{S}$. For instance, from the slice $\mathsf{S} := \langle\{x,u,v,y\},\{x\,t\,y, x\,r\,u, u\,\mathbf{I}\,v, v\,s\,y\} : x,y\rangle$, we obtain the (equivalent) slice $\mathsf{S}' := \langle\{x,v,y\},\{x\,t\,y, x\,r\,v, v\,s\,y\} : x,y\rangle$.

**Example 3.1.** *Consider the three slices* $\mathsf{S} := \ \to x \xrightarrow{r} u \xrightarrow{s} v \xrightarrow{t} y \to$ , $\mathsf{T} := \ \to w \xrightarrow{p} z \to$ *and* $\mathsf{T}' :=$
$$\begin{tikzcd} \downarrow\uparrow w \arrow[d, bend left, "p"] \\ z \arrow[u, bend left, "q"] \end{tikzcd}$$
*We then have the following three glued slices:* $\mathsf{S}\tfrac{u}{v}\mathsf{T} =$
$$\begin{tikzcd} \to x \arrow[r, "r"] & u \arrow[r, bend left, "s"] \arrow[r, bend right, "p"'] & v \arrow[r, "t"] & y \to \end{tikzcd},$$

$\mathsf{S}\tfrac{u}{v}\mathsf{T}' =$
$$\begin{tikzcd} \to x \arrow[r, "r"] & w \arrow[loop above, "s"] \arrow[r, "t"] \arrow[d, shift left, "p"] & y \to \\ & z \arrow[u, shift left, "q"] & \end{tikzcd}$$
*and* $\mathsf{S}\tfrac{x}{y}\mathsf{T}' =$
$$\begin{tikzcd} & & u \arrow[dd, "s"] \\ \rightleftarrows w \arrow[urr, "r"] \arrow[d, bend right, "p"'] & & \\ z \arrow[u, bend right, "q"'] & & v \arrow[ull, "t"] \end{tikzcd}.$$

The category of sketches and morphisms has co-limits. The co-limit of a diagram of sketches can be obtained as expected: obtain the co-limit of the sets of nodes and then transfer arcs (by using the functions to the co-limit node set). Thus, the pushout of drafts gives a draft.

Gluing involves an amalgamated sum (of drafts). Consider a slice $\mathsf{T}$. Given a draft $\mathsf{D} = \langle N,A\rangle$ and nodes $(u,v) \in \mathbb{N}\mathsf{d}^2$, the *glued draft* $\mathsf{D}\tfrac{u}{v}\mathsf{T}$ is the pushout of drafts $\mathsf{D}$ and $\underline{\mathsf{T}}$ over the arcless draft $\langle\{x,y\},\emptyset\rangle$ and natural morphisms ($\alpha : x \mapsto u, y \mapsto v$ and $\beta : x \mapsto x_{\mathsf{T}}, y \mapsto y_{\mathsf{T}}$) as follows:

$$\begin{tikzcd} & \mathsf{D} + \{u,v\} \arrow[dr, "\sigma"] & \\ \langle\{x,y\},\emptyset\rangle \arrow[ur, "\alpha"] \arrow[dr, "\beta"'] & & \mathsf{D}\tfrac{u}{v}\mathsf{T} \\ & \underline{\mathsf{T}} \arrow[ur, "\tau"'] & \end{tikzcd}$$

Given a slice $\mathsf{S} = \langle\underline{\mathsf{S}} : x_{\mathsf{S}}, y_{\mathsf{S}}\rangle$, we obtain the *glued slice* $\mathsf{S}\tfrac{u}{v}\mathsf{T}$ by transferring the input and output nodes of $\mathsf{S}$ to the glued draft $\underline{\mathsf{S}}\tfrac{u}{v}\mathsf{T}$: $\mathsf{S}\tfrac{u}{v}\mathsf{T} := \langle\underline{\mathsf{S}}\tfrac{u}{v}\mathsf{T} : {x_{\mathsf{S}}}^{\sigma}, {y_{\mathsf{S}}}^{\sigma}\rangle$. The glued draft and slice are unique up to isomorphism.[3] Also, we *glue* a graph naturally by gluing its slices: $\mathsf{S}\tfrac{u}{v}\mathsf{H} := \{\mathsf{S}\tfrac{u}{v}\mathsf{T} \,/\, \mathsf{T} \in \mathsf{H}\}$. Note that, for the empty graph: $\mathsf{S}\tfrac{u}{v}\{\ \} = \{\mathsf{S}\tfrac{u}{v}\mathsf{T} \,/\, \mathsf{T} \in \{\ \}\} = \{\ \}$.

# 4 Refutation Calculus

We now introduce our refutation calculus: label conversion and graph expansion. We will first examine basic objects, then rules of our calculus: conversion and its rules (in 4.1) and the expansion rule (in 4.2).

To establish that a label is null, we first convert it to a graph (by conversion rules) and then try to obtain a zero graph by repeatedly applying the expansion rule (cf. the examples in Section 2 and Examples 4.1 and 4.4).

We define basic labels, arcs, sketches, slices and graphs by mutual recursion. A label $\mathsf{L}$ is a *basic label* iff it is either a relation name in $\mathbb{R}\mathsf{n}$ or $\overline{\mathsf{T}}$, where $\mathsf{T}$ is a basic slice (see below). An arc $u\,\mathsf{L}\,v$ is a *basic*

[3] As isomorphic objects have the same behavior, we often consider a sketch or a slice up to isomorphism.

*arc* iff its label L is basic. A sketch is a *basic sketch* iff all its arcs are basic arcs. A slice $\mathsf{S}$ is a *basic slice* iff its underlying draft $\underline{\mathsf{S}}$ is a basic sketch. A graph is a *basic graph* iff all its slices are basic slices.

In Example 2.2 (in Section 2), slice $\mathsf{S}$ is not basic (as it has composite terms as labels), whereas slice $\mathsf{S}'$ is basic (as it has 4 basic arc labels : r, s, t and $\overline{\mathsf{T}'}$, where $\mathsf{T}'$ is a basic slice). Also, in Example 4.1 (in 4.1 below), both slices $\mathsf{S}$ and $\mathsf{T}$ are basic.

## 4.1 Label conversion

We now examine label conversion and its rules in our calculus.

**Example 4.1.** *Consider the inclusion* $\mathrm{P} \sqsubseteq \mathrm{Q}$ *(cf. [14]), with terms* $\mathrm{P} := \mathrm{a} \sqcap [(\mathrm{b}\,;\mathrm{c}) \sqcap \mathrm{d})\,;(\mathrm{e} \sqcap (\mathrm{f}\,;\mathrm{g}))]$ *and* $\mathrm{Q} := \mathrm{b}\,;[(((\mathrm{b}^{\smile};\mathrm{a})\sqcap(\mathrm{c}\,;\mathrm{e}));\mathrm{g}^{\smile})\sqcap(\mathrm{c}\,;\mathrm{f})\sqcap(\mathrm{b}^{\smile};((\mathrm{a}\,;\mathrm{g}^{\smile})\sqcap(\mathrm{d}\,;\mathrm{f}))]\,;\mathrm{g}$*, over relation names* $\mathrm{a},\mathrm{b},\mathrm{c},\mathrm{d},\mathrm{e},\mathrm{f},\mathrm{g}$*. Label* P *is equivalent to the graph* $\{\mathsf{S}\}$*, with the following basic slice* $\mathsf{S}$*:*

$$\begin{tikzcd}
\to x \arrow[rr, "a"] \arrow[d, "b"'] \arrow[dr, "d"] & & y \to \\
u \arrow[r, "c"] & v \arrow[ur, "e"] \arrow[r, "f"] & w \arrow[u, "g"']
\end{tikzcd}$$

*Label* Q *is equivalent to the graph* $\{\mathsf{T}\}$*, with the following basic slice* $\mathsf{T}$*:*

$$\begin{tikzcd}
 & x' \arrow[rr, "a"] \arrow[dd, "b"'] & & y' & \\
 & & v' \arrow[ur, "e"] & & \\
\to x \arrow[r, "b"] & u \arrow[ur, "c"] \arrow[r, "c"] & v \arrow[r, "f"] & w \arrow[uu, "g"'] \arrow[r, "g"] \arrow[dl, "g"] & y \to \\
 & & y'' & & \\
 & x'' \arrow[uu, "b"] \arrow[ur, "a"] \arrow[rr, "d"] & & v'' \arrow[uu, "f"'] &
\end{tikzcd}$$

*So, the difference slice* $\mathsf{DS}(\mathrm{P} \setminus \mathrm{Q})$ *is equivalent to the graph* $\{\mathsf{S} + x\overline{\mathsf{T}}y\}$*. Now, we have a morphism* $\theta : \underline{\mathsf{T}} \dashrightarrow \underline{\mathsf{S}}$ *given by* $x, x', x'' \mapsto x$*;* $u \mapsto u$*;* $v, v', v'' \mapsto v$*;* $w \mapsto w$ *and* $y, y', y'' \mapsto y$*. Thus,* $\{\mathsf{S} + x\overline{\mathsf{T}}y\}$ *is a zero graph, so inclusions* $\{\mathsf{S} + x\overline{\mathsf{T}}y\} \sqsubseteq \bot$*,* $\mathsf{DS}(\mathrm{P} \setminus \mathrm{Q}) \sqsubseteq \bot$ *and* $\mathrm{P} \sqsubseteq \mathrm{Q}$ *are all valid.*

The conversion rules will be of two kinds: operational and structural rules. The aim of these rules is converting every label to an equivalent basic graph (see Proposition 4.1).

The operational rules come from labels that are equivalent to graphs.[4] For the constants: $\bot$ is equivalent to the empty graph $\{\ \}$, $\mathbb{T}$ and $\mathbb{I}$ are equivalent to graphs with a single arcless slice, namely $\to x \;\; y\to$ and $\to x\to$; also, for diversity $\mathbb{D} \equiv \{\ \to x \xrightarrow{\overline{\boxed{\to x \to}}} y\to\ \}$. For the operations: $\mathrm{L}^{\smile} \equiv \{\to x \xleftarrow{\mathrm{L}} y\to\}$, $\mathrm{L}\sqcap\mathrm{K}$ is equivalent to the graph whose single slice consists of the 2 parallel arcs $\to x \xrightarrow{\mathrm{L}} y\to$ and $\to x \xrightarrow{\mathrm{K}} y\to$,

[4] Recall that x, y and z are the first 3 individual nodes (see Section 3).

$L \sqcup K$ is equivalent to the graph $\{\to x \xrightarrow{L} y \to, \to x \xrightarrow{K} y \to\}$, $L;K$ is equivalent to the graph with single slice $\to x \xrightarrow{L} z \xrightarrow{K} y \to$ and $L\dagger K$ is equivalent to the graph with single slice $\to x \xrightarrow{\overline{\boxed{\to x \xrightarrow{\overline{L}} z \xrightarrow{\overline{K}} y \to}}} y \to$. We have no such rule for complementation, but we do have $\overline{\overline{L}} \equiv L$. We will consider the *consecutive-arc slice* $\mathsf{Sl}(L{\to}K) := \langle\{x,y,z\},\{x\,L\,z, z\,K\,y\} : x,y\rangle$, i.e. the slice $\to x \xrightarrow{L} z \xrightarrow{K} y \to$.

Table 1 gives the 10 *operational rules*.

| | | | | |
|---|---|---|---|---|
| $(\perp\!\!\!\perp)$ | $\perp\!\!\!\perp$ | $\rhd$ | $\{\ \}$ | empty graph |
| $(\top\!\!\!\top)$ | $\top\!\!\!\top$ | $\rhd$ | $\{\langle\{x,y\},\emptyset : x,y\rangle\}$ | 2-node arcless slice $\to x \quad y \to$ |
| $(\mathbb{I})$ | $\mathbb{I}$ | $\rhd$ | $\{\langle\{x\},\emptyset : x,x\rangle\}$ | single-node arcless slice $\to x \to$ |
| $(\mathbb{D})$ | $\mathbb{D}$ | $\rhd$ | $\{\langle\{x,y\},\{\overline{\langle\{x\},\emptyset : x,x\rangle}\} : x,y\rangle\}$ | 2-node single-arc slice $\to x \xrightarrow{\overline{\boxed{\to x \to}}} y \to$ |
| $(=)$ | $\overline{\overline{L}}$ | $\rhd$ | $L$ | replace $\overline{\overline{L}}$ by $L$ |
| $(\smile)$ | $L^{\smile}$ | $\rhd$ | $\{\langle\{x,y\},\{y\,L\,x\} : x,y\rangle\}$ | reversed-arc slice $\to x \xleftarrow{L} y \to$ |
| $(\sqcap)$ | $L \sqcap K$ | $\rhd$ | $\{\langle\{x,y\},\{x\,L\,y, x\,K\,y\} : x,y\rangle\}$ | parallel-arc slice: $\to x \overset{L}{\underset{K}{\rightrightarrows}} y \to$ |
| $(\sqcup)$ | $L \sqcup K$ | $\rhd$ | $\left\{\begin{array}{c}\langle\{x,y\},\{x\,L\,y\} : x,y\rangle, \\ \langle\{x,y\},\{x\,K\,y\} : x,y\rangle\end{array}\right\}$ | alternative slices: $\to x \xrightarrow{L} y \to$ <br> $\to x \xrightarrow{K} y \to$ |
| $(;)$ | $L;K$ | $\rhd$ | $\{\mathsf{Sl}(L{\to}K)\}$ | consecutive-arc slice: $\to x \xrightarrow{L} z \xrightarrow{K} y \to$ |
| $(\dagger)$ | $L\dagger K$ | $\rhd$ | $\{\langle\{x,y\},\{x\,\overline{\mathsf{Sl}(\overline{L}{\to}\overline{K})}\,y\} : x,y\rangle\}$ | complemented label: $\overline{\to x \xrightarrow{\overline{L}} z \xrightarrow{\overline{K}} y \to}$ |

Table 1: Operational rules

By applying the operational rules (of Table 1) in any context, one can eliminate all relational constants and operations except complement, but complemented relation names (e.g $\overline{r}$) remain and slices or graphs and their complements as labels may appear.

**Example 4.2.** *The operational rules (in Table 1) give the following conversions.*

*1.* $r;\perp\!\!\!\perp \overset{(;)}{\rhd} \{\to x \xrightarrow{r} z \xrightarrow{\perp\!\!\!\perp} y \to\} \overset{(\perp\!\!\!\perp)}{\rhd} \left\{ \to x \xrightarrow{r} z \xrightarrow{\boxed{\{\ \}}} y \to \right\} = \mathsf{G}_1$.

*2.* $r;(s\sqcup t) \overset{(;)}{\rhd} \{\to x \xrightarrow{r} z \xrightarrow{s\sqcup t} y \to\} \overset{(\sqcup)}{\rhd} \left\{ \to x \xrightarrow{r} z \xrightarrow{\boxed{\left\{\begin{array}{c}\to x \xrightarrow{s} y \to, \\ \to x \xrightarrow{t} y \to\end{array}\right\}}} y \to \right\} = \mathsf{G}_2$.

*3.* $\mathsf{r};\overline{\mathsf{s}\sqcap\mathsf{t}} \overset{(;)}{\triangleright} \{\rightarrow\mathrm{x} \overset{\mathrm{r}}{\rightarrow} \mathrm{z} \overset{\overline{\mathsf{s}\sqcap\mathsf{t}}}{\rightarrow} \mathrm{y}\rightarrow\} \overset{(\sqcap)}{\triangleright} \left\{ \rightarrow\mathrm{x} \xrightarrow{\mathrm{r}} \mathrm{z} \xrightarrow{\overline{\left\{\boxed{\rightarrow\mathrm{x} \underset{\mathrm{t}}{\overset{\mathrm{s}}{\rightrightarrows}} \mathrm{y}\rightarrow}\right\}}} \mathrm{y}\rightarrow \right\} = \mathsf{G}_3.$

The 4 structural rules ($\overset{\cup}{\rightarrow}$), ($\overline{\cup}$), ($\overline{\cap}$) and ($\bar{r}$) will address such cases.[5]

($\overset{\cup}{\rightarrow}$) We can replace a graph arc by glued slices (cf. 3.2), as $\mathsf{S}+\mathrm{u}\,\mathsf{H}\,\mathrm{v} \equiv \{\mathsf{S}\,^{\mathrm{u}}_{\mathrm{v}}\,\mathsf{T}\,/\,\mathsf{T}\in\mathsf{H}\}$. For instance, with the slices $\mathsf{S} := \langle\{\mathrm{x},\mathrm{u},\mathrm{v},\mathrm{y}\},\{\mathrm{x\,r\,u},\mathrm{u\,s\,v},\mathrm{v\,t\,y}\} : \mathrm{x},\mathrm{y}\rangle$, $\mathsf{T} := \langle\{\mathrm{w},\mathrm{z}\},\{\mathrm{w\,p\,z}\} : \mathrm{w},\mathrm{z}\rangle$ and $\mathsf{T}' := \langle\{\mathrm{w},\mathrm{z}\},\{\mathrm{w\,p\,z},\mathrm{z\,q\,w}\} : \mathrm{w},\mathrm{w}\rangle$ (cf. Example 3.1 in 3.2), we have $\mathsf{S}+\mathrm{u}\,\{\mathsf{T},\mathsf{T}'\}\,\mathrm{v}$ equivalent to $\{\langle\{\mathrm{x},\mathrm{u},\mathrm{v},\mathrm{y}\},\{\mathrm{x\,r\,u},\mathrm{u\,s\,v},\mathrm{u\,p\,v},\mathrm{v\,t\,v}\} : \mathrm{x},\mathrm{y}\rangle,\langle\{\mathrm{x},\mathrm{v},\mathrm{z},\mathrm{y}\},\{\mathrm{x\,r\,v},\mathrm{v\,s\,v},\mathrm{v\,t\,y},\mathrm{v\,p\,z},\mathrm{z\,q\,v}\} : \mathrm{x},\mathrm{y}\rangle\}$.

($\overline{\cup}$) Also, we can replace a label that is a complemented graph by a slice, since $\overline{\mathsf{G}} \equiv \mathsf{Sl}[\mathsf{G}]$, where $\mathsf{Sl}[\mathsf{G}] := \langle\{\mathrm{x},\mathrm{y}\},\{\mathrm{x}\,\overline{\mathsf{S}}\,\mathrm{y}\,/\,\mathsf{S}\in\mathsf{G}\} : \mathrm{x},\mathrm{y}\rangle$ is the *slice of graph* $\mathsf{G}$. For a 2-slice graph $\mathsf{G} = \{\mathsf{S}_1,\mathsf{S}_2\}$,

$\mathsf{Sl}[\mathsf{G}]$ is the 2-arc slice $\rightarrow\mathrm{x} \underset{\overline{\mathsf{S}_3}}{\overset{\overline{\mathsf{S}_1}}{\rightrightarrows}} \mathrm{y}\rightarrow$ .

($\overline{\cap}$) Consider a slice $\mathsf{S} = \langle N,A : \mathrm{x}_{\mathsf{S}},\mathrm{y}_{\mathsf{S}}\rangle$. Call slice $\mathsf{S}$ *small* iff $N = \{\mathrm{x}_{\mathsf{S}},\mathrm{y}_{\mathsf{S}}\}$. An *I-O arc* of $\mathsf{S}$ is an arc $\mathrm{u\,L\,v} \in A$ with $\{\mathrm{u},\mathrm{v}\} \subseteq \{\mathrm{x}_{\mathsf{S}},\mathrm{y}_{\mathsf{S}}\}$. The *transformed* of I-O arc $\mathsf{a} = \mathrm{u\,L\,v}$ is the arc $\mathsf{a}^{\mathtt{tr}}$ obtained by replacing $\mathrm{x}_{\mathsf{S}}$ by x, $y_{\mathsf{S}}$ by y and label L by $\overline{\mathrm{L}}$. Now, the *graph of slice* $\mathsf{S}$ is the graph $\mathsf{Gr}(\mathsf{S})$ with a single-arc slice $\langle\{\mathrm{x},\mathrm{y}\},\{\mathsf{a}^{\mathtt{tr}}\} : \mathrm{x},\mathrm{y}\rangle$, for each I-O arc $\mathsf{a}$ of $\mathsf{S}$. For a 3-arc small slice $\mathsf{S} = \langle\{\mathrm{w},\mathrm{z}\},\{\mathrm{w\,r\,z},\mathrm{z\,s\,w},\mathrm{w\,t\,w}\} : \mathrm{w},\mathrm{z}\rangle$, $\mathsf{Gr}(\mathsf{S})$ is a graph with 3 slices, namely $\langle\{\mathrm{x},\mathrm{y}\},\{\mathrm{x}\,\bar{\mathrm{r}}\,\mathrm{y}\} : \mathrm{x},\mathrm{y}\rangle$, $\langle\{\mathrm{x},\mathrm{y}\},\{\mathrm{y}\,\bar{\mathrm{s}}\,\mathrm{x}\} : \mathrm{x},\mathrm{y}\rangle$ and $\langle\{\mathrm{x},\mathrm{y}\},\{\mathrm{x}\,\bar{\mathrm{t}}\,\mathrm{x}\} : \mathrm{x},\mathrm{y}\rangle$; pictorially, we have $\mathsf{Gr}(\rightarrow\overset{\overset{\mathrm{t}}{\circlearrowleft}}{\mathrm{w}} \underset{\mathrm{s}}{\overset{\mathrm{r}}{\rightleftarrows}} \mathrm{z}\rightarrow)$ as the graph $\{\rightarrow\mathrm{x} \overset{\bar{\mathrm{r}}}{\rightarrow} \mathrm{y}\rightarrow, \rightarrow\mathrm{x} \overset{\bar{\mathrm{s}}}{\leftarrow} \mathrm{y}\rightarrow, \rightarrow \overset{\overset{\bar{\mathrm{t}}}{\circlearrowleft}}{\mathrm{x}} \ \mathrm{y}\rightarrow\}$. Now, for a small slice, we can replace the complemented slice by a graph, moving complement inside, as $\overline{\{\mathsf{S}\}} \equiv \mathsf{Gr}(\mathsf{S})$.

($\bar{r}$) Finally, we can replace a label $\bar{r}$ by $\overline{\rightarrow\mathrm{x} \overset{r}{\rightarrow} \mathrm{y}\rightarrow}$ (since $\overline{\mathrm{L}} \equiv \overline{\rightarrow\mathrm{x} \overset{\mathrm{L}}{\rightarrow} \mathrm{y}\rightarrow}$).

**Example 4.3.** *The graphs* $\mathsf{G}_1$, $\mathsf{G}_2$ *and* $\mathsf{G}_3$ *in Example 4.2 have conversions as follows.*

*1. For* $\mathsf{G}_1$*:* $\left\{\rightarrow\mathrm{x} \overset{\mathrm{r}}{\rightarrow} \mathrm{z} \xrightarrow{\boxed{\{\ \}}} \mathrm{y}\rightarrow\right\} \overset{(\overset{\cup}{\rightarrow})}{\triangleright} \{\rightarrow\mathrm{x} \overset{\mathrm{r}}{\rightarrow} \mathrm{z} \xrightarrow{\boxed{\mathsf{T}}} y\rightarrow/\,\mathsf{T}\in\{\ \}\} = \{\ \}.$

*2. For* $\mathsf{G}_2$*:* $\left\{\rightarrow\mathrm{x} \xrightarrow{\mathrm{r}} \mathrm{z} \xrightarrow{\boxed{\left\{\begin{array}{l}\rightarrow\mathrm{x} \overset{\mathrm{s}}{\rightarrow} \mathrm{y}\rightarrow,\\ \rightarrow\mathrm{x} \overset{\mathrm{t}}{\rightarrow} \mathrm{y}\rightarrow\end{array}\right\}}} \mathrm{y}\rightarrow\right\} \overset{(\overset{\cup}{\rightarrow})}{\triangleright} \left\{\begin{array}{l}\rightarrow\mathrm{x} \overset{\mathrm{r}}{\rightarrow} \mathrm{z} \overset{\mathrm{s}}{\rightarrow} \mathrm{y}\rightarrow,\\ \rightarrow\mathrm{x} \overset{\mathrm{r}}{\rightarrow} \mathrm{z} \overset{\mathrm{t}}{\rightarrow} \mathrm{y}\rightarrow\end{array}\right\}.$

*3. For* $\mathsf{G}_3$*:* $\left\{\rightarrow\mathrm{x} \xrightarrow{\mathrm{r}} \mathrm{z} \xrightarrow{\overline{\left\{\boxed{\rightarrow\mathrm{x} \underset{\mathrm{t}}{\overset{\mathrm{s}}{\rightrightarrows}} \mathrm{y}\rightarrow}\right\}}} \mathrm{y}\rightarrow\right\} \overset{(\overline{\cap})}{\triangleright} \left\{\rightarrow\mathrm{x} \xrightarrow{\mathrm{r}} \mathrm{z} \xrightarrow{\boxed{\left\{\begin{array}{l}\rightarrow\mathrm{x} \overset{\bar{\mathrm{s}}}{\rightarrow} \mathrm{y}\rightarrow,\\ \rightarrow\mathrm{x} \overset{\bar{\mathrm{t}}}{\rightarrow} \mathrm{y}\rightarrow\end{array}\right\}}} \mathrm{y}\rightarrow\right\}$

[5] Recall that a slice $\mathsf{S}$ and its single-slice graph $\{\mathsf{S}\}$ are equivalent (cf. 3.1).

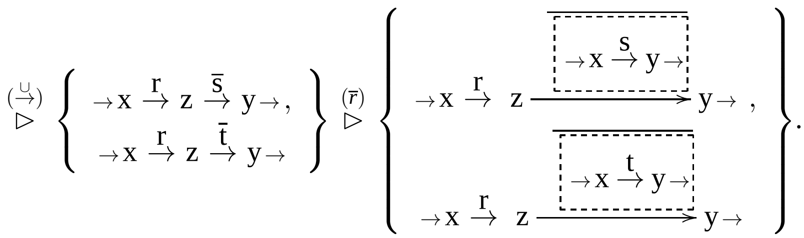


Table 2 gives the 4 *structural rules*.

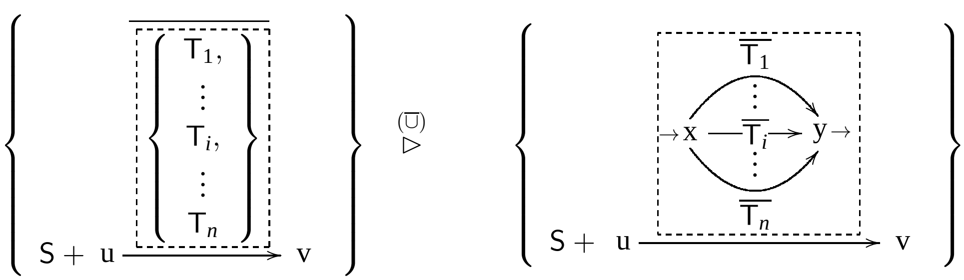
| | | | | |
|---|---|---|---|---|
| $(\overset{\cup}{\to})$ | $\{\mathsf{S} + \mathrm{u}\,\mathsf{H}\,\mathrm{v}\}$ | $\rhd$ | $\mathsf{S}\frac{\mathrm{u}}{\mathrm{v}}\mathsf{H}$ | replace graph arc by glued slices |
| $(\overline{\cup})$ | $\overline{\mathsf{G}}$ | $\rhd$ | $\mathsf{Sl}[\mathsf{G}]$ | replace $\overline{\mathsf{G}}$ by slice of $\mathsf{G}$ |
| $(\overline{\cap})$ small $\mathsf{S}$ | $\overline{\{\mathsf{S}\}}$ | $\rhd$ | $\mathsf{Gr}(\mathsf{S})$ | replace $\overline{\{\mathsf{S}\}}$ by graph of $\mathsf{S}$ |
| $(\overline{r})$ $r \in \mathrm{R}n$ | $\overline{r}$ | $\rhd$ | $\overline{\langle\{\mathrm{x},\mathrm{y}\},\{\mathrm{x}\,r\,\mathrm{y}\} : \mathrm{x},\mathrm{y}\rangle}$ | replace $\overline{r}$ by label $\overline{\to\mathrm{x} \overset{r}{\to} \mathrm{y}\to}$ |

Table 2: Structural rules

We also have a derived rule replacing a complemented graph arc by parallel complemented slice arcs:

$$(\overset{\overline{\cup}}{\to})\quad \{\mathsf{S} + \mathrm{u}\,\overline{\mathsf{H}}\,\mathrm{v}\} \quad \rhd \quad \{\mathsf{S} + \{\mathrm{u}\,\overline{\mathsf{T}}\,\mathrm{v} \,/\, \mathsf{T} \in \mathsf{H}\}\} \quad \text{replace } \mathrm{u} \overset{\overline{\mathsf{H}}}{\to} \mathrm{v} \text{ by } \{\mathrm{u} \overset{\overline{\mathsf{T}}}{\to} \mathrm{v} \,/\, \mathsf{T} \in \mathsf{H}\}$$

Derived rule $(\overset{\overline{\cup}}{\to})$ is obtained by applying rules $(\overline{\cup})$ and $(\overset{\cup}{\to})$ as follows:



$$\overset{(\overset{\cup}{\to})}{\rhd} \quad \left\{\ \mathsf{S} + \mathrm{u} \overset{\overline{\mathsf{T}_1}}{\to} \mathrm{v} + \ldots + \mathrm{u} \overset{\overline{\mathsf{T}_i}}{\to} \mathrm{v} + \ldots + \mathrm{u} \overset{\overline{\mathsf{T}_n}}{\to} \mathrm{v}\ \right\}$$

The 14 *conversion rules* (in Tables 1 and 2) can be applied in any context. We take the *eventual conversion* relation $\rhd^*$ as the reflexive-transitive closure of the immediate conversion relation $\rhd$ under relational operations as well as slice and graph formation. More precisely: if $\mathrm{L} \rhd^* \mathrm{K}$ then $\overline{\mathrm{L}} \rhd^* \overline{\mathrm{K}}$, $\mathrm{L}^{\smile} \rhd^* \mathrm{K}^{\smile}$ and $\mathsf{S} + \mathrm{u}\,\mathrm{L}\,\mathrm{v} \rhd^* \mathsf{S} + \mathrm{u}\,\mathrm{K}\,\mathrm{v}$; if $\mathrm{L}_1 \rhd^* \mathrm{K}_1$ and $\mathrm{L}_2 \rhd^* \mathrm{K}_2$ then $\mathrm{L}_1 \bullet \mathrm{L}_2 \rhd^* \mathrm{K}_1 \bullet \mathrm{K}_2$ (for a 2-ary operation $\bullet \in \{\sqcup, \sqcap, ;, \dagger\}$); if $\mathsf{T} \rhd^* \mathsf{T}'$ then $\mathsf{G} \cup \{\mathsf{T}\} \rhd^* \mathsf{G} \cup \{\mathsf{T}'\}$ and if $\mathsf{H} \rhd^* \mathsf{H}'$ then $\mathsf{G} \cup \mathsf{H} \rhd^* \mathsf{G} \cup \mathsf{H}'$.

One can apply the conversion rules in Tables 1 and 2 modularly (cf. Example 4.1).

**Remark 4.1.** *If* $\mathrm{L} \rhd^* \mathrm{L}'$ *and* $\mathrm{K} \rhd^* \mathrm{K}'$, *then* $\mathsf{DS}(\mathrm{L} \setminus \mathrm{K}) \rhd^* \mathsf{DS}(\mathrm{L}' \setminus \mathrm{K}')$.

**Proposition 4.1** (Conversion)**.** *Every label* $\mathrm{L}$ *can be eventually converted (by repeated applications of the conversion rules in Tables 1 and 2) to an equivalent basic graph* $\mathrm{L}^{\mathsf{bs}}$.

### 4.2 Graph expansion

We now examine graph expansion and its rule in our calculus.

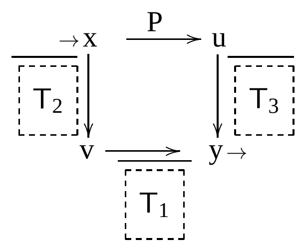
**Example 4.4.** *We now establish the inclusion* $\mathrm{P;(Q\dagger R)} \sqsubseteq \mathrm{(P;Q)\dagger R}$.

1. *As before, we begin with the difference slice* $\mathsf{DS}(\mathrm{P;(Q\dagger R)} \setminus \mathrm{(P;Q)\dagger R})$.
2. *We can convert it to a slice* $\mathsf{S}'$ *having complemented slices as arc labels. With the following slices*

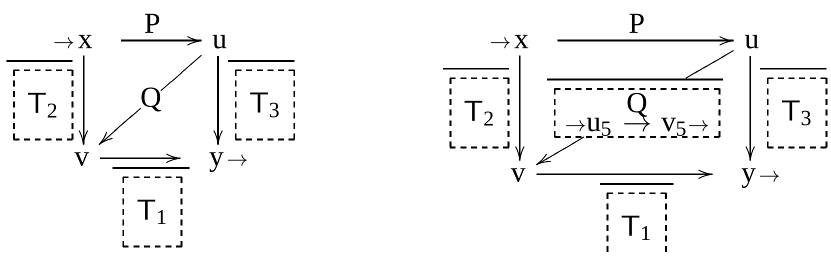
$$\mathsf{T}_1 := \to \mathrm{v}_1 \xrightarrow{\mathrm{R}} \mathrm{y}_1 \to, \mathsf{T}_2 := \to \mathrm{x}_2 \xrightarrow{\mathrm{P}} \mathrm{w}_2 \xrightarrow{\mathrm{Q}} \mathrm{y}_2 \to \textit{ and}$$

$$\mathsf{T}_3 := \to \mathrm{u}_3 \xrightarrow{\overline{\boxed{\to \mathrm{u}_4 \xrightarrow{\mathrm{Q}} \mathrm{z}_4 \to}}} \mathrm{v}_3 \xrightarrow{\overline{\boxed{\to \mathrm{z}_5 \xrightarrow{\mathrm{R}} \mathrm{y}_5 \to}}} \mathrm{y}_3 \to,$$

   *we have slice* $\mathsf{S}'$ *as follows:*



3. *This slice* $\mathsf{S}'$ *is not yet inconsistent. We can however expand it to a graph* $\mathsf{G}$ *consisting of* 2 *alternative slices* $\mathsf{S}_+$ *and* $\mathsf{S}_-$*, respectively as follows:*



*Now, both* $\mathsf{S}_+$ *and* $\mathsf{S}_-$ *can be seen to be zero slices: slice* $\mathsf{S}_+$ *has the arcs* $\mathrm{x\,P\,u}$, $\mathrm{u\,Q\,v}$ *and* $\mathrm{x}\,\overline{\mathsf{T}_2}\,\mathrm{v}$*, while slice* $\mathsf{S}_-$ *has the arcs* $\mathrm{u}\,\overline{\boxed{\to \mathrm{u}_5 \xrightarrow{\mathrm{Q}} \mathrm{v}_5 \to}}\,\mathrm{v}$, $\mathrm{v}\,\overline{\mathsf{T}_1}\,\mathrm{y}$ *and* $\mathrm{u}\,\overline{\mathsf{T}_3}\,\mathrm{y}$*. Therefore, we have established the inclusion* $\{\mathsf{S}_+,\mathsf{S}_-\} \sqsubseteq \mathbf{\perp}$*, whence also* $\{\mathsf{S}'\} \sqsubseteq \mathbf{\perp}$ *and* $\mathrm{P;(Q\dagger R)} \sqsubseteq \mathrm{(P;Q)\dagger R}$.

The expansion rule has an instance for slices $\mathsf{S}$ and $\mathsf{T}$ and pair of nodes $(\mathrm{u},\mathrm{v})$ of $\mathsf{S}$, which replaces the single-slice graph $\{\mathsf{S}\}$ by the 2-slice graph $\{\mathsf{S}\frac{\mathrm{u}}{\mathrm{v}}\mathsf{T}\} \cup \{\mathsf{S} + \mathrm{u}\,\overline{\mathsf{T}}\,\mathrm{v}\}$. The *expansion rule* is as follows:

$$(\mathtt{Exp}) \quad \frac{\{\mathsf{S}\}}{\{\mathsf{S}\frac{\mathrm{u}}{\mathrm{v}}\mathsf{T}, \mathsf{S} + \mathrm{u}\,\overline{\mathsf{T}}\,\mathrm{v}\}} \qquad (u,v) \in N_{\mathsf{S}}{}^2 \qquad \text{replace } \mathsf{S} \text{ by } \mathsf{S}\tfrac{\mathrm{u}}{\mathrm{v}}\mathsf{T} \;\&\; \mathsf{S} + \mathrm{u}\,\overline{\mathsf{T}}\,\mathrm{v}$$

We use $\lhd$ for the *immediate expansion* relation between graphs (e.g. $\{\mathsf{S}'\} \lhd \{\mathsf{S}_+,\mathsf{S}_-\}$ in Example 4.4) and $\lhd^*$ for its reflexive-transitive closure: the *eventual expansion* relation. A *derivation* is a sequence $\mathrm{L},\mathsf{G}_0,\ldots,\mathsf{G}_n$ of labels, such that, $\mathsf{G}_0,\ldots,\mathsf{G}_n$ are graphs, L eventually converts to $\mathsf{G}_0$ ($\mathrm{L} \rhd^* \mathsf{G}_0$) and, for each $i = 1,\ldots,n$, $\mathsf{G}_{i-1}$ converts or expands to $\mathsf{G}_i$ ($\mathsf{G}_{i-1}\,(\rhd \cup \lhd)\,\mathsf{G}_i$). Call a derivation *normal* iff applications of conversion rules precede applications of expansions.[6] We say that label L *derives* graph $\mathsf{H}$ (noted $\mathrm{L} \vdash \mathsf{H}$) iff there exists a derivation $\mathrm{L}, \mathsf{G}_0, \ldots, \mathsf{G}_n$ with $\mathsf{G}_n = \mathsf{H}$. Call a label *derivably zero* iff it derives some zero graph and *expansively zero* iff it eventually expands to some zero graph.

We have soundness and completeness of (normal) derivations.

**Theorem 4.1** (Correctness)**.** *Consider a label* L.

**(Sound)** *If label* L *is derivably zero, then* L *is null.*

**(Complete)** *If label* L *is a null basic graph, then* L *is expansively zero.*

Soundness is not difficult to see. For establishing completeness, we introduce (by mutual recursion) two measures of structural complexity: *rank* and *set of embedded slices*, with the aim of providing an appropriate inductive measure. For a relation name $r \in Rn$: $\mathsf{rk}(r) := 0$ and $\mathsf{ES}[r] := \emptyset$; for a basic slice $\mathsf{T}$: $\mathsf{rk}(\overline{\mathsf{T}}) := \mathtt{rk}(\mathsf{T}) + 1$ and $\mathsf{ES}[\overline{\mathsf{T}}] := \mathsf{ES}[\mathsf{T}] \cup \{\mathsf{T}\}$. For a basic label L: $\mathsf{rk}(u \,\mathrm{L}\, v) := \mathsf{rk}(\mathrm{L})$ and $\mathsf{ES}[u \mathrm{L} v] := \mathsf{ES}[\mathrm{L}]$. For a basic draft $\mathsf{D}$: $\mathtt{rk}(\mathsf{D}) := \sum_{\mathsf{a} \in A_{\mathsf{D}}} \mathsf{rk}(\mathsf{a})$ and for a basic sketch $\Sigma$: $\mathtt{ES}[\Sigma] := \bigcup_{\mathsf{a} \in A_\Sigma} \mathsf{ES}[\mathsf{a}]$. For a basic slice $\mathsf{S}$: $\mathtt{rk}(\mathsf{S}) := \mathtt{rk}(\underline{\mathsf{S}})$ and $\mathtt{ES}[\mathsf{S}] := \mathtt{ES}[\underline{\mathsf{S}}]$. Thus, for a basic draft $\mathsf{D} = \mathsf{D}' + \mathrm{u}\,\overline{\mathsf{T}}\,\mathrm{v}$, with $\mathrm{u}\,\overline{\mathsf{T}}\,\mathrm{v} \notin A_{\mathsf{D}'}$, we will have $\mathtt{rk}(\mathsf{D}) = \mathtt{rk}(\mathsf{D}') + \mathtt{rk}(\mathsf{T}) + 1$ and $\mathtt{ES}[\mathsf{D}] = \mathtt{ES}[\mathsf{D}'] \cup \mathtt{ES}[\mathsf{T}] \cup \{\mathsf{T}\}$.

We now indicate how one can establish completeness. Consider a basic graph $\mathsf{G}$ that is not expansively zero. Then, it has a slice $\mathsf{S} \in \mathsf{G}$ that is not zero, such that, for every $(\mathrm{u}, \mathrm{v}) \in {N_{\mathsf{S}}}^2$ and basic slice $\mathsf{T}$, $\{\mathsf{S} \frac{\mathrm{u}}{\mathrm{v}} \mathsf{T}\}$ or $\{\mathsf{S} + \mathrm{u}\,\overline{\mathsf{T}}\,\mathrm{v}\}$ is not expansively zero. Thus, we can then obtain a family $\mathcal{R}$ of non-zero basic slices (with underlying drafts connected by morphisms), which is saturated by applications of the expansion rule.[7] This family $\mathcal{R}$ can be used to obtain a co-limit sketch $\Sigma$, giving a natural model $\mathfrak{C}$ (cf. 3.2), which discriminates satisfying assignments as morphisms to $\Sigma$: for a basic draft $\mathsf{D}$ with $\mathtt{ES}[\mathsf{D}] \subseteq \mathtt{ES}[\Sigma]$, we have $\mathtt{g} : \mathsf{D} \to \mathfrak{C}$ iff $\mathtt{g} : \mathsf{D} \dashrightarrow \Sigma$ (by induction on $\mathtt{rk}(\mathsf{D})$). Hence, we have a counter-model: $[\![\mathsf{G}]\!]_{\mathfrak{C}} \supseteq [\![\mathsf{S}]\!]_{\mathfrak{C}} \neq \emptyset$.

We thus have a correct calculus for null labels and for valid label inclusions.

**(**L**)** A label L is null iff it its basic form $\mathrm{L}^{\mathsf{bs}}$ is expansively zero.

**(**$\sqsubseteq$**)** A label inclusion $\mathrm{L} \sqsubseteq \mathrm{K}$ is valid iff $\{\mathsf{DS}(\mathrm{L} \setminus \mathrm{K})\}^{\mathsf{bs}}$ is expansively zero.

## 5 Hypotheses

We now extend the preceding ideas to handle inclusions as hypotheses, by resorting to difference slices.

**Example 5.1.** *Consider the assertion:* "P;R′;Q $\sqsubseteq$ P;R″;Q *follows from* R′ $\sqsubseteq$ R″". *We reduce it to deriving* $(\mathrm{P;R';Q}) \sqcap \overline{\mathrm{P;R'';Q}} \sqsubseteq \bot\!\!\!\bot$ *from* $\mathrm{R'} \sqcap \overline{\mathrm{R''}} \sqsubseteq \bot\!\!\!\bot$.

*The difference slice* $\mathsf{DS}(\mathrm{R'} \setminus \mathrm{R''})$ *is equivalent to* $\mathsf{S}' := \begin{tikzcd} \to \mathrm{x} \arrow[r, bend left, "\mathrm{R'}"] \arrow[r, bend right, "\overline{\boxed{\to \mathrm{x} \xrightarrow{\mathrm{R''}} \mathrm{y} \to}}"'] & \mathrm{y} \to \end{tikzcd}$ .

1. *Begin with the graph* $\{\mathsf{DS}(\mathrm{P;R';Q} \setminus \mathrm{P;R'';Q})\}$*, with single slice* $\mathsf{S}_0$ *as follows:*

$$\mathsf{S}_0 \quad := \quad \begin{tikzcd} \to \mathrm{x} \arrow[r, bend left, "\mathrm{P;R';Q}"] \arrow[r, bend right, "\overline{\mathrm{P;R'';Q}}"'] & \mathrm{y} \to \end{tikzcd}$$

[6] The preceding examples use normal derivations: of the form $\mathrm{L} \rhd^* \mathsf{G} \lhd^* \mathsf{H}$.

[7] One may regard this as an analogue of Lindenbaum's Lemma: extending a consistent theory to a maximally consistent one.

2. *Slice* $\mathsf{S}_0$ *is equivalent to the following slice* $\mathsf{S}_1$*:*

$$\begin{tikzcd} \rightarrow x \arrow[r, "P"] \arrow[d, "\overline{\boxed{\rightarrow x' \xrightarrow{P} u' \xrightarrow{R''} v' \xrightarrow{Q} y' \rightarrow}}"'] & u \arrow[d, "R'"] \\ \leftarrow y & v \arrow[l, "Q"] \end{tikzcd}$$

3. *Now, expand graph* $\{\mathsf{S}_1\}$ *(with* $\mathsf{T} := \rightarrow x \xrightarrow{R''} y \rightarrow$*), obtaining a graph* $\mathsf{H} = \{\mathsf{S}_+, \mathsf{S}_-\}$*, where slices* $\mathsf{S}_+ := \mathsf{S}_1 \frac{u}{v} \mathsf{T}$ *and* $\mathsf{S}_- := \mathsf{S}_1 + u\overline{\mathsf{T}}v$ *are as follows:*

$$\mathsf{S}_+ \quad := \quad \begin{tikzcd} \rightarrow x \arrow[r, "P"] \arrow[d, "\overline{\boxed{\rightarrow x' \xrightarrow{P} u' \xrightarrow{R''} v' \xrightarrow{Q} y' \rightarrow}}"'] & u \arrow[d, bend right, "R'"'] \arrow[d, bend left, "R''"] \\ \leftarrow y & v \arrow[l, "Q"] \end{tikzcd}$$

$$\mathsf{S}_- \quad := \quad \begin{tikzcd} \rightarrow x \arrow[r, "P"] \arrow[d, "\overline{\boxed{\rightarrow x' \xrightarrow{P} u' \xrightarrow{R''} v' \xrightarrow{Q} y' \rightarrow}}"'] & u \arrow[d, bend right, "R'"'] \arrow[d, bend left, "\overline{\boxed{\rightarrow x \xrightarrow{R''} y \rightarrow}}"] \\ \leftarrow y & v \arrow[l, "Q"] \end{tikzcd}$$

*Now, consider the graph* $\mathsf{H} := \{\mathsf{S}_+, \mathsf{S}_-\}$.

- *Slice* $\mathsf{S}_+$ *is zero (because we have a morphism* $\theta$ *from* $\boxed{x' \xrightarrow{P} u' \xrightarrow{R''} v' \xrightarrow{Q} y'}$ *to* $\underline{\mathsf{S}_+}$*, given by* $x' \mapsto x, u' \mapsto u, v' \mapsto v, y' \mapsto y$*).*
- *As for slice* $\mathsf{S}_-$*, we have a morphism* $\theta' : \underline{\mathsf{S}'} \dashrightarrow \underline{\mathsf{S}_-}$*, given by* $x \mapsto u, y \mapsto v$.

*Thus,* $\mathsf{H}$ *has empty extension in any model where the hypothesis* $R' \sqsubseteq R''$ *holds.*

Given a set $\Lambda$ of inclusions, we say that $\Lambda$ *holds* in model $\mathfrak{M}$ (noted $\mathfrak{M} \models \Lambda$) iff every inclusion in $\Lambda$ holds in $\mathfrak{M}$. Now, we say that inclusion $L \sqsubseteq K$ *follows from* set $\Lambda$ of inclusions (noted $\Lambda \models L \sqsubseteq K$) iff $L \sqsubseteq K$ holds in every model $\mathfrak{M}$ where $\Lambda$ holds, i.e. $\mathfrak{M} \models L \sqsubseteq K$, whenever $\mathfrak{M} \models \Lambda$.

In Example 5.1, we have $\{\mathsf{S}_0\} \equiv \{\mathsf{S}_-, \mathsf{S}_+\}$, where $\mathsf{S}_+$ is a zero slice and one can erase slice $\mathsf{S}_-$.

Given a set $\Gamma$ of slices, call a slice $\mathsf{S}$ $\Gamma$-*erasable* iff $\mathsf{Mor}[\underline{\mathsf{S}'}, \underline{\mathsf{S}}] \neq \emptyset$ for some $\mathsf{S}' \in \Gamma$. The rule for hypothesis states that one can erase any $\Gamma$-erasable slice. The rule for hypothesis $\mathtt{Hyp}[\Gamma]$ is as follows:

$$(\mathtt{Hyp}[\Gamma]) \ \frac{\{\mathsf{S}\}}{\{\ \}} \qquad \text{if slice } \mathsf{S} \text{ is } \Gamma\text{-erasable}$$

One can also widen the goal to $\Gamma$-*zero* graphs, where each slice is zero or $\Gamma$-erasable. We have two versions of graph calculus with hypotheses. Given a set $\Gamma$ of slices and a (basic) graph $\mathsf{G}$, we have two ways of establishing that $\mathsf{G} \sqsubseteq \mathbb{L}$ follows from the set of assumed inclusions $\Lambda[\Gamma] := \{\mathsf{S}' \sqsubseteq \mathbb{L} \,/\, \mathsf{S}' \in \Gamma\}$.

- Derive a zero graph by using the rules $(\mathtt{Exp})$ and $(\mathtt{Hyp}[\Gamma])$, or
- derive a $\Gamma$-zero graph by using only the expansion rule $(\mathtt{Exp})$.

Both versions are sound and complete for a set $\Gamma$ consisting of basic slices.

**Theorem 5.1** (Hypotheses)**.** *Given a set $\Gamma$ of basic slices and a basic graph $\mathsf{G}$, the following 3 assertions are equivalent.*

1. *Inclusion $\mathsf{G} \sqsubseteq \mathbb{L}$ follows from $\Lambda[\Gamma] = \{\mathsf{S}' \sqsubseteq \mathbb{L} \,/\, \mathsf{S}' \in \Gamma\}$: $\Lambda[\Gamma] \models \mathsf{G} \sqsubseteq \mathbb{L}$.*
2. *From $\mathsf{G}$ one can derive a zero graph by applications of (*`Exp`*) and (*$\mathtt{Hyp}[\Gamma]$*).*
3. *From $\mathsf{G}$ one can derive a $\Gamma$-zero graph by applications of the rule (*`Exp`*).*

# 6 Conclusion

We now present some concluding remarks about graph calculi for relational inclusions.

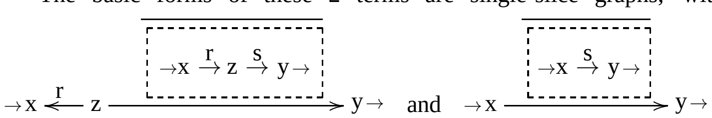
We have examined a sound and complete goal-oriented graphical calculus for inclusions: it reduces establishing a label inclusion to establishing that a graph constructed from it has empty extension. Relational terms, slices and graphs are labels and every label is equivalent to a basic graph and to a slice.[8]

Our goal-oriented calculus is simpler than some of the available graph relational calculi [7, 8, 9, 10, 11, 12]. It is conceptually simpler as it proceeds by eliminating relational operations and its rules require only the concept of (draft) morphism (rather than slice homomorphism – a draft morphism that respects input and output nodes – and graph cover [9]). Also, it manipulates a single graph trying to convert it to a zero graph (rather than two graphs and comparing them [12]). For instance, to establish directly the inclusion $\mathrm{r}^{\smile};\overline{\mathrm{r};\mathrm{s}} \sqsubseteq \overline{\mathrm{s}}$ (cf. Example 2.1), one would have to apply the expansion rule.[9] In fact, whenever there is a slice homomorphism from $\mathsf{T}$ to $\mathsf{S}$, the difference slice $\mathsf{DS}(\mathsf{S} \setminus \mathsf{T})$ is a zero slice.

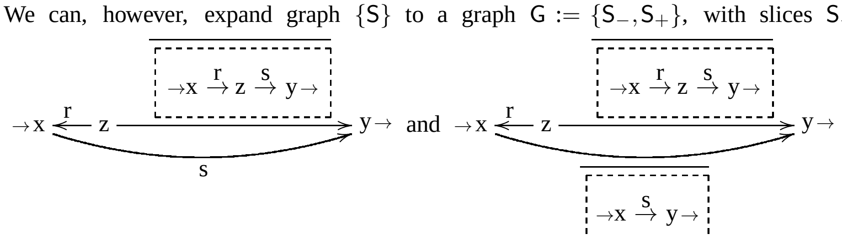
Also, the treatment of hypotheses is much simpler than in the usual calculi, as it resorts to erasing (rather than gluing) slices. The assertion in Example 5.1 can be established directly without the expansion rule (by means of the gluing rule for hypotheses). On the other hand, an assertion like "r $\sqsubseteq$ s follows from $\mathrm{r} \sqcap \overline{\mathrm{s}} \sqsubseteq \mathbb{L}$", which is trivial in our approach, will require using the expansion rule in the direct approach.[10]

[8] Also, any Boolean combination of inclusions is equivalent to an inclusion $\mathsf{L} \sqsubseteq \mathbb{L}$ [16].

[9] The basic forms of these 2 terms are single-slice graphs, with the following slices $\mathsf{S}$ and $\mathsf{T}$, respectively:

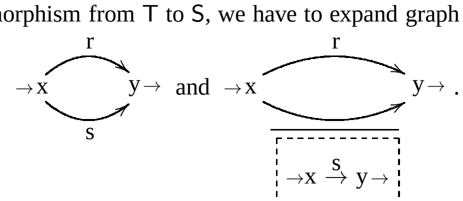


There is no homomorphism from $\mathsf{T}$ to $\mathsf{S}$. We can, however, expand graph $\{\mathsf{S}\}$ to a graph $\mathsf{G} := \{\mathsf{S}_-, \mathsf{S}_+\}$, with slices $\mathsf{S}_-$ and $\mathsf{S}_+$, respectively, as follows:



. Now, slice $\mathsf{S}_-$ is a zero slice (cf. slice $\mathsf{S}_4$ in Example 2.1 in Section 2) and we have a homomorphism from $\mathsf{T}$ to $\mathsf{S}_+$.

[10] With the slices $\mathsf{S} := \to \overset{\mathrm{r}}{\to} \mathrm{y}\to$ and $\mathsf{T} := \to \overset{\mathrm{s}}{\to} \mathrm{y}\to$, as there is no homomorphism from $\mathsf{T}$ to $\mathsf{S}$, we have to expand graph $\{\mathsf{S}\}$ to a graph $\mathsf{G} := \{\mathsf{S}_-, \mathsf{S}_+\}$, with slices $\mathsf{S}_-$ and $\mathsf{S}_+$, respectively, as follows:



Now, we have a homomorphism from $\mathsf{T}$ to $\mathsf{S}_-$ and slice $\mathsf{S}_+$ can be erased (as we have a homomorphism from $\mathsf{S}$ to $\mathsf{S}_+$).

Moreover, the idea of labels with embedded slices or graphs is rather powerful. Comparing with other graph calculi, we conjecture that there is not much gain or loss in complexity order, its main advantages are on the conceptual side: simpler concepts and goal orientation.

## References


[1] T. Barkowsky (2010): *Diagrams in the mind: visual or spatial?*. In A. K. Goel, M. Jamnik & N. H. Narayanan, editors: *Lecture Notes in Artificial Intelligence*, *Series* 6170, p. 1, Springer-Verlag, Berlin, doi:10.1007/978-3-540-92687-0.

[2] C. Brink, W. Kahl & G. Schmidt, editors (1997): *Relational Methods in Computer Science*. Springer-Verlag, Wien.

[3] C. Brown & G. Hutton (1994): *Categories, allegories and circuit design*. In *Proc. LICS 94*, IEEE-Computer Science, pp. 372–381, doi:10.1109/LICS.1994.316052.

[4] C. Brown & A. Jeffrey (1994): *Allegories of circuits*. In A. Nerode & Y. Matiyasevich, editors: *Lecture Notes in Computer Science*, Springer, *Series* 813, pp. 56–68, St. Petersburg, 1994, doi:10.1007/3-540-58140-5-7.

[5] S. Curtis & G. Lowe (1995): *A graphical calculus*. In B. Moller, editor: *Mathematics of Program Construction LNCS Series* 947, Springer-Verlag, Berlin, pp. 214–231, doi:10.1007/3-540-60117-1-12.

[6] S. Curtis & G. Lowe (1996): *Proofs with graphs*. In R. Backhouse, editor: *Science of Computer Programming*, Elsevier, volume (26), pp. 197–216, doi:10.1016/0167-6423(95)00025-9.

[7] R. Freitas, P. A. S. Veloso, S. R. M. Veloso & P. Viana (2006): *Reasoning with graphs*. In G. Mints & R. J. G. B. de Queiroz, editors: *Electronic Notes in Theoretical Computer Science*, Elsevier, *Series* 165, pp. 201–212, doi:10.1016/j.entcs.2006.05.046.

[8] R. Freitas, P. A. S. Veloso, S. R. M. Veloso & P. Viana (2007): *On positive relational calculi*. *Logic J. IGPL* volume (15) , pp. 577–601, doi:10.1093/jigpal/jzm054.

[9] R. Freitas, P. A. S. Veloso, S. R. M. Veloso & P. Viana (2008): *On a graph calculus for algebras of relations*. In W. Hodges & R. de Queiroz, editors: *Lecture Notes in Artificial Inelligence*, *Series* 5110, Springer-Verlag, Heiderberg, pp. 298–312, doi:10.1007/978-3-540-69937-8.

[10] R. Freitas, P. A. S. Veloso, S. R. M. Veloso & P. Viana (2009): *Positive fork graph calculus*. In S. Artemov, editor: *Lecture Notes in Computer Science*, *Series* 5407, Springer-Verlag, New York, pp. 152–163, doi:10.1007/978-3-540-92687-0.

[11] R. Freitas, P. A. S. Veloso, S. R. M. Veloso & P. Viana (2009): *On graph reasoning*. *Information and Computation*, volume (207), pp. 1000–1014, doi:10.1016/j.ic.2008.11.004.

[12] R. Freitas, P. A. S. Veloso, S. R. M. Veloso & P. Viana (2010): *A calculus for graphs with complement*. In A. K. Goel, M. Jamnik & N. H. Narayanan, editors: *Lecture Notes in Artificial Inelligence*, *Series* 6170, pp. 84–98, Springer-Verlag, Berlin, doi:10.1007/978-3-540-92687-0.

[13] S . MacLane (1998): *Categories for the Working Mathematician*, second edition, Springer-Verlag, Berlin.

[14] R. D. Maddux (1991): *The origin of relation algebras in the development and axiomatization of the calculus of relations*. *Studia Logica* volume (50), pp. 412–455, Springer-Verlag, doi:10.1007/BF00370681.

[15] R. D. Maddux (1996): *Relation-algebraic semantics*. *Theoretical Computer Science* pp.1–85, Elsevier doi:10.1016/0304-3975(95)00082-8.

[16] R. D.Maddux (2006): *Relation Algebras*. Elsevier, Amsterdam.

[17] G. Schmidt & T. Ströhlein (1993): *Relations and Graphs: Discrete Mathematics for Computer Science*. Springer-Verlag, Berlin.