



**HAL**  
open science

# Non axiomatisability of positive relation algebras with constants, via graph homomorphisms

Amina Doumane, Damien Pous

► **To cite this version:**

Amina Doumane, Damien Pous. Non axiomatisability of positive relation algebras with constants, via graph homomorphisms. *CONCUR*, 2020, Vienne, Austria. pp.1-23, 10.4230/LIPIcs.CONCUR.2020.49 . hal-02870687v2

**HAL Id: hal-02870687**

**<https://hal.science/hal-02870687v2>**

Submitted on 23 Jul 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Non axiomatisability of positive relation algebras with constants, via graph homomorphisms

**Amina Doumane**

Univ Lyon, CNRS, ENS de Lyon, UCB Lyon 1, LIP, Lyon, France  
amina.doumane@ens-lyon.fr

**Damien Pous**

Univ Lyon, CNRS, ENS de Lyon, UCB Lyon 1, LIP, Lyon, France  
damien.pous@ens-lyon.fr

---

## Abstract

We study the equational theories of composition and intersection on binary relations, with or without their associated neutral elements (identity and full relation). Without these constants, the equational theory coincides with that of semilattice-ordered semigroups. We show that the equational theory is no longer finitely based when adding one or the other constant, refuting a conjecture from the literature. Our proofs exploit a characterisation in terms of graphs and homomorphisms, which we show how to adapt in order to capture standard equational theories over the considered signatures.

**2012 ACM Subject Classification** Mathematics of computing → Discrete mathematics

**Keywords and phrases** Relation algebra, graph homomorphisms, (in)equational theories

**Related Version** This is the full version of the paper with the same title published in Proc. CONCUR 2020, available at <https://doi.org/10.4230/LIPIcs.CONCUR.2020.49>

**Funding** This work has been funded by the European Research Council (ERC-StG CoVeCe 678157) and the French National Research Agency (ANR-10-LABX-0070 ANR-11-IDEX-0007).

## 1 Introduction

Several operations and constants are used frequently on binary relations: set-theoretic operations (intersection, union, complement, empty and full relations), relational composition and identity relation, and converse (transpose). Amongst others, Tarski studied those operations and analysed their expressiveness and the equational laws they satisfy [24, 25, 23, 18]. It turns out that these basic operations already make it possible to encode Peano arithmetic in a purely algebraic setting, without variables. As a consequence, the corresponding equational theory is undecidable and not finitely based [20].

The situation changes when considering *positive* fragments [12, 4, 1, 19, 2], where the complement operation is removed. Indeed, the equational theory of those fragments is decidable, even in the presence of additional operations like reflexive transitive closure [6, 21]. However, results concerning finite axiomatisations are more on the negative side. Hodkinson and Mikuláš proved that one cannot obtain a finite firstorder axiomatisation whenever the operations of composition, intersection and converse are present [17]. When only two of those operations are considered, we get positive results: the problem is straightforward for converse and intersection; the case of composition and converse is more subtle and covered in [4, 10, 1]; and the equational theory of composition and intersection coincides with that of *semilattice-ordered semigroups* [3].

However, understanding the laws satisfied by the identity and the full-relation constants is difficult. They are neutral elements for composition and intersection, respectively, but they also satisfy rather unexpected laws, so that the equational theories depart from those of semilattice-ordered monoids and bounded-semilattice-ordered semigroups. The case of composition, intersection, and identity was thought to be finitely based, with an explicit

candidate [2], but there was an error in the completeness proof so that it remained as a conjecture. Our first contribution consists in refuting this conjecture: the equational theory of this fragment is not finitely based. Our second contribution is that adding the full relation to composition and intersection also yields an equational theory which is not finitely based.

The reasons for non finite axiomatisability of those two fragments are quite different. Still, our two proofs rely on a graph-theoretical characterisation the equational theory of binary relations [12, 1]. First, terms  $u, v$  built over a set of variables and the signature consisting of composition, intersection, and their neutral elements, can be used to denote graphs. The class of *expressible graphs*, those that may be denoted via a term, strongly depends on the considered fragment: they are always of treewidth at most two [8], they are also acyclic unless we have the identity constant, they are also connected unless we have the full-relation constant. The key result shared for all fragments is that a law  $u \leq v$  is valid for relations if and only if there exists a homomorphism from the graph of  $v$  to the graph of  $u$ .

We prove the first negative result as follows: we first show that if we had a finite and equational axiomatisation, then we would be able to decompose every homomorphism between two expressible graphs while remaining within the class of expressible graphs—a similar idea is used in [12] for representable allegories; we formalise it in Section 3. Then we provide a counter-example: an infinite sequence of homomorphisms that cannot be decomposed accordingly, by exploiting a necessary condition for a graph to be expressible (Section 4).

We do not think a similar argument can be used for the second negative result. Instead, we give directly an infinite sequence of homomorphisms and we show that for each of them, the corresponding law essentially has to be included into any sound and complete axiomatisation (Section 5).

As mentioned above, when neutral elements are taken into account, the equational theory of binary relations differs from that of natural algebraic structures extending semilattice-ordered semigroups (semilattice-ordered monoids, bounded-semilattice-ordered semigroups, and bounded-semilattice-ordered monoids). Our last contribution consists in providing graph-theoretical characterisations for those structures, yielding decidability in polynomial time of their equational theories (Section 6).

From the concurrency theory point of view, the structures considered here should not be confused with the ones studied in the literature on pomsets [14, 13] or concurrent Kleene algebra [16]. Indeed, two forms of composition are also put forward in those lines of work: sequential and parallel composition, and they resemble the operations of relational composition and intersection we consider in the present paper (e.g., they form monoids related by the ‘weak exchange’ law). However, the operation of intersection we use in the present paper is idempotent, and thus induces a partial order, which is not the case for parallel composition in concurrency theory. Accordingly, one should consider intersection as an operator for combining specifications rather than a program construction for concurrency, like with allegories and some of its extensions [12, 6, 22, 9].

## 2 Preliminaries

### 2.1 Terms

We fix in the rest of the paper an infinite alphabet  $A$ , and we let  $a, b \dots$  range over its letters. SP1T *terms* (series-parallel with one and  $\top$ ) are generated by the following syntax

$$e, f ::= e \cdot f \mid e \cap f \mid 1 \mid \top \mid a \quad (a \in A)$$

We denote their set by  $\text{SP1}\top$  and we often write  $ef$  for  $e \cdot f$ . We moreover assign priorities so that  $ab \cap c$  reads as  $(a \cdot b) \cap c$ . We define  $\text{SP1}$ ,  $\text{SPT}$  and  $\text{SP}$  to be respectively the set of  $\text{SP1}\top$  terms not containing  $\top$ ,  $1$  and neither of them. We also use those symbols to denote the associated signatures.

## 2.2 Relational interpretation

We are primarily interested in the relational interpretation of terms, where  $\cdot$  is relational composition,  $\cap$  is set-theoretic intersection,  $1$  is the identity relation, and  $\top$  is the full relation. Given an interpretation  $\sigma : A \rightarrow \mathcal{P}(S \times S)$  of letters into some space of binary relations, we write  $\widehat{\sigma} : \text{SP1}\top \rightarrow \mathcal{P}(S \times S)$  for the corresponding extension to terms.

An inequation between two terms  $u$  and  $v$  is *valid*, written  $\mathcal{R}el \models u \leq v$ , if for every such interpretation  $\sigma$  we have  $\widehat{\sigma}(u) \subseteq \widehat{\sigma}(v)$ . We call (*in*)*equational theory of binary relations* the set  $\mathcal{R}el$  of valid inequations. (We focus on inequations in the present work; note however that those are equivalent to equations: we have  $\mathcal{R}el \models u \leq v$  iff  $\mathcal{R}el \models u \cap v = u$ , where the latter symbol is defined as expected.)

## 2.3 Graphs

As explained in the introduction, terms also make it possible to denote graphs—more precisely, directed multigraphs with edges labelled in  $A$  and two designated vertices. Formally, those are tuples  $\langle V, E, s, t, l, \iota, o \rangle$  with  $V$  (resp.  $E$ ) a finite set of vertices (resp. edges),  $s, t : E \rightarrow V$  the *source* and *target* functions,  $l : E \rightarrow A$  the *labelling* function, and  $\iota, o \in V$  two distinguished vertices, respectively called *input* and *output*. We simply call them *graphs* in the sequel; we depict them as expected, with unlabelled ingoing and outgoing arrows to denote the input and the output, respectively.

Vertices distinct from input and output are called *inner* vertices. A vertex without incident edges is *isolated*.

Graphs can be composed in series or in parallel, as depicted below:

$$G \cdot H \triangleq \begin{array}{c} \rightarrow \circ \text{---} G \text{---} \circ \text{---} H \text{---} \circ \rightarrow \end{array} \quad G \cap H \triangleq \begin{array}{c} \rightarrow \circ \begin{array}{c} \text{---} G \text{---} \\ \text{---} H \text{---} \end{array} \circ \rightarrow \end{array}$$

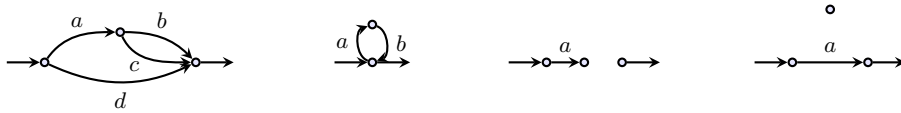
Those operations do have neutral elements, which are edge-less graphs:

$$1 \triangleq \begin{array}{c} \rightarrow \circ \rightarrow \end{array} \quad \top \triangleq \begin{array}{c} \rightarrow \circ \end{array} \quad \begin{array}{c} \circ \rightarrow \end{array}$$

We can thus recursively associate to every term  $u$  a graph  $\mathcal{G}(u)$  called the *graph of  $u$* , where the graph of a letter  $a \in A$  is

$$\mathcal{G}(a) \triangleq \begin{array}{c} \rightarrow \circ \xrightarrow{a} \circ \rightarrow \end{array}$$

Here are, from left to right, the graphs of  $a(b \cap c) \cap d$ ,  $ab \cap 1$ ,  $a\top$  and  $a \cap \top\top$ :



We say that a graph is  $\text{SP}$  (resp.  $\text{SP1}$ ,  $\text{SPT}$ ,  $\text{SP1}\top$ ) if it is the graph of some  $\text{SP}$  (resp.  $\text{SP1}$ ,  $\text{SPT}$ ,  $\text{SP1}\top$ ) term. The  $\text{SP}$  graphs are the acyclic and series-parallel graphs with all edges directed from the input towards the output. In a  $\text{SP1}$  graph, every vertex belongs to a directed path from the input towards the output; unlike  $\text{SP}$  graphs, they may contain cycles. In contrast,  $\text{SPT}$  graphs remain acyclic but they are not necessarily connected.  $\text{SP1}\top$  have treewidth at most two, i.e., they are  $K_4$ -free [8].

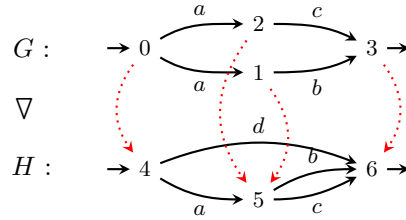
## 2.4 Homomorphisms

Graph homomorphisms play a central role in the paper; they are defined as follows:

► **Definition 1** (Graph homomorphism). *Given two graphs  $G = \langle V, E, s, t, l, \iota, o \rangle$  and  $G' = \langle V', E', s', t', l', \iota', o' \rangle$ , a (graph) homomorphism  $h : G \rightarrow H$  is a pair  $\langle h_v, h_e \rangle$  of functions  $h_v : V \rightarrow V'$  and  $h_e : E \rightarrow E'$  that respect the various components:  $s' \circ h_e = h_v \circ s$ ,  $t' \circ h_e = h_v \circ t$ ,  $l = l' \circ h_e$ ,  $\iota' = h_v(\iota)$ , and  $o' = h_v(o)$ .*

*We write  $H \triangleleft G$  if there exists a graph homomorphism from  $G$  to  $H$ . Like other relations on graphs, we sometimes use this relation directly on terms, writing  $u \triangleleft v$  for  $\mathcal{G}(u) \triangleleft \mathcal{G}(v)$ .*

The vertex component of such a homomorphism is depicted below



A pleasant way to think about graph homomorphisms is the following: we have  $H \triangleleft G$  if  $H$  is obtained from  $G$  by merging (or identifying) some vertices and some edges, and by adding some extra vertices and edges. For instance, the graph  $H$  in the example above is obtained from  $G$  by merging vertices 1 and 2 and the two  $a$ -labelled edges, and by adding a  $d$ -labelled edge from the input to the output.

We write  $h \circ g$  for the pointwise composition of the two components of two homomorphisms, which yields a homomorphism as expected.

A homomorphism is *injective* (resp. *surjective*, *bijective*) when its two components are so. We write  $G \hookrightarrow H$  if there exists an injective homomorphism from  $G$  to  $H$ ; in such a case, we say that  $G$  is a *subgraph* of  $H$ . We write  $G \simeq H$  when there exists a bijective homomorphism from  $G$  to  $H$  (an *isomorphism*). The aforementioned intuition about homomorphisms is reflected by the epi-mono factorisation property: every homomorphism  $h : G \rightarrow H$  factors uniquely into a surjective homomorphism followed by an injective homomorphism:

$$G \twoheadrightarrow h(G) \hookrightarrow H$$

The intermediate graph  $h(G)$  is the *image* of  $h$ ; it is a subgraph of  $H$  by definition.

The key result we exploit in the present paper is the following characterisation:

► **Theorem 2** ([1, Thm. 1], [12, p. 208]). *For all terms  $u, v$ ,  $\mathcal{R}el \models u \leq v$  iff  $u \triangleleft v$ .*

Thus, analysing the inequational theory of binary relations amounts to analysing homomorphism between graphs denoted by terms.

## 2.5 Closure under taking subgraphs

We show below that,  $\text{SPT}$  (resp.  $\text{SP1T}$ ), seen as a class of graphs, is the closure of  $\text{SP}$  (resp.  $\text{SP1}$ ) under taking subgraphs. This property is convenient in the sequel.

► **Proposition 3.**  *$G$  is  $\text{SPT}$  (resp.  $\text{SP1T}$ ) iff  $G$  is a subgraph of a  $\text{SP}$  (resp.  $\text{SP1}$ ) graph.*

**Proof.** We can reason mostly on terms. The forward implication is easy: given a SPT (resp. SP1T) term  $u$ , replacing all occurrences of  $\top$  with an arbitrary letter yields a SP (resp. SP1) term  $u'$  such that  $\mathcal{G}(u) \leftrightarrow \mathcal{G}(u')$ .

For the converse implication, we proceed in two steps. Assume  $h : G \leftrightarrow \mathcal{G}(u')$  for some SP (resp. SP1) term  $u'$ . First observe that the edges in  $\mathcal{G}(u')$  are in one-to-one correspondence with the occurrences of letters in  $u'$ . By replacing with  $\top$  all occurrences of letters in  $u'$  that are not in the image of  $h$  through this correspondence, we obtain a SPT (resp. SP1T) term  $u_0$  such that  $h$  corestricts to  $h_0 : G \leftrightarrow \mathcal{G}(u_0)$ , which is actually bijective on edges. It remains to get rid of the vertices  $\mathcal{G}(u_0)$  which are not in the image of  $h_0$ . Those are necessarily isolated inner vertices in  $\mathcal{G}(u_0)$  since  $h_0$  is bijective on edges. Roughly speaking, those arise via subterms of the shape  $\top\top$  in  $u_0$ , which we can replace with  $\top$  to obtain a SPT (resp. SP1T) term  $u$  whose graph is  $G$ . The formal argument is slightly more involved; we give it in Appendix A.  $\blacktriangleleft$

► **Corollary 4.** *The classes of graphs SPT and SP1T are closed under taking subgraphs.*

## 2.6 Inequational reasoning

Let us define what we mean by axiomatisation in the present context, where we focus on inequations rather than equations.

Assume a signature  $\Sigma$ , and consider in this subsection terms  $u, v$  built over this signature and variables in the alphabet  $A$ . We let  $\sigma, \theta$  range over *substitutions* assigning a terms to letters in  $A$ , and we write  $u\sigma$  for the result of applying such a substitution  $\sigma$  to a term  $u$ . A *renaming* is a possibly non-injective substitution whose range consists only of letters. We let  $C$  range over *contexts*, i.e., terms with exactly one occurrence of a special letter  $\bullet$  called the *hole*. We write  $C[u]$  for the term obtained by replacing the hole of a context  $C$  by a term  $u$ .

An *inequation* is a pair of terms, which we denote by  $u \leq v$ . An *inequational theory* is a set of inequations which forms a pre-order and which is stable under contexts and substitutions. For instance, the set of inequations such that  $\mathcal{R}el \models u \leq v$  is an inequational theory.

Given a set  $\mathcal{H}$  of inequations, the *axioms*, the *inequational theory of  $\mathcal{H}$*  is the least inequational theory containing  $\mathcal{H}$ . We write  $\mathcal{H} \vdash u \leq v$  when the inequation  $u \leq v$  belongs to the inequational theory of  $\mathcal{H}$ , or, equivalently, if it can be derived using the following rules:

$$\frac{}{u \leq v} \mathcal{H} \quad \frac{}{u \leq u} \quad \frac{u \leq v \quad v \leq w}{u \leq w} \quad \frac{u \leq v}{u\sigma \leq v\sigma} \quad \frac{u \leq v}{C[u] \leq C[v]}$$

An inequational theory is *finitely based* if it can be generated by a finite set of axioms.

### Standard algebraic structures

Inequational theories as defined above can be presented as equational theories as soon as the signature  $\Sigma$  contains a binary symbol  $\cap$  and  $\mathcal{H}$  contains the following finite set of inequations:

$$\mathcal{P} \triangleq \{a \leq a \cap a, a \cap b \leq a, a \cap b \leq b\}$$

Indeed, in such a case,  $\cap$  turns the partial order  $\leq$  into an *inf-semilattice*, and inf-semilattices can be defined algebraically as commutative idempotent semigroups: the partial order can be defined as  $u \leq v \triangleq (u \cap v = u)$ . The other operations in the signature must all be monotone, which can be expressed algebraically by adding equations of the form  $f(a \cap b) \cap f(a) = f(a)$ , say, for a unary symbol  $f$ . Conversely, any equational theory with a commutative idempotent

semigroup symbol and where all operations are monotone w.r.t. the associated partial order can be represented as an inequational theory in the previous sense. (Those conversions preserve the finiteness of the considered set of axioms, so that an inequational theory is finitely based iff its associated equational theory is finitely based, and vice versa.)

In particular, we capture other standard algebraic structures as follows. Define the following (finite) sets of inequations, where an equation is a shorthand for the corresponding two inequations:

$$\begin{aligned} \mathcal{SP} &\triangleq \mathcal{P} \cup \{a \cdot (b \cdot c) = (a \cdot b) \cdot c\} & \mathcal{SP}^\top &\triangleq \mathcal{SP} \cup \{a \leq \top\} \\ \mathcal{SP}^1 &\triangleq \mathcal{SP} \cup \{a \cdot 1 = a, 1 \cdot a = a\} & \mathcal{SP}^{1\top} &\triangleq \mathcal{SP}^1 \cup \mathcal{SP}^\top \end{aligned}$$

(Note that these sets are implicitly associated to the four signatures we consider in the present paper: for instance, when writing  $\mathcal{SP}^1 \vdash u \leq v$ , we mean that  $u$  and  $v$  are  $\mathcal{SP}^1$  terms and that the derivation mentions only  $\mathcal{SP}^1$  terms, contexts, and substitutions.)

We have that

- $\mathcal{SP}$  axiomatises *semilattice-ordered semigroups (sl-semigroups)*;
- $\mathcal{SP}^1$  axiomatises *semilattice-ordered monoids (sl-monoids)*;
- $\mathcal{SP}^\top$  axiomatises *bounded-semilattice-ordered semigroups (bsl-semigroups)*;
- $\mathcal{SP}^{1\top}$  axiomatises *bounded-semilattice-ordered monoids (bsl-monoids)*.

### Axiomatisability of relations

Given a subsignature  $X$  of  $\mathcal{SP}^{1\top}$ , we say that a set  $\mathcal{H}$  of inequations on  $X$  *axiomatises relations on  $X$*  if

$$\text{for all terms } u, v \text{ on } X, \quad \mathcal{H} \vdash u \leq v \quad \text{iff} \quad \mathcal{Rel} \models u \leq v .$$

Bredihin and Schein proved that the equational theory of relations on  $\mathcal{SP}$  coincides with that of sl-semigroups [3], which means in the above terminology that  $\mathcal{SP}$  axiomatises relations on  $\mathcal{SP}$ .

In contrast,  $\mathcal{SP}^1$  and  $\mathcal{SP}^\top$  do not suffice to axiomatise relations on  $\mathcal{SP}^1$  or  $\mathcal{SPT}$ . For instance, relations satisfy the laws below (by Theorem 2, this can be proved by providing appropriate homomorphisms—most of them actually are isomorphisms here) but there are bsl-monoids violating those laws<sup>1</sup>.

$$\mathcal{Rel} \vdash a\top \cap bc = (a\top \cap b)c \qquad \mathcal{Rel} \vdash \top a \top b \top = \top b \top a \top$$

$$\mathcal{Rel} \vdash a \cap b \cap 1 = (a \cap 1)(b \cap 1) \qquad \mathcal{Rel} \vdash (a \cap 1)b \cap c = (a \cap 1)(b \cap c) \qquad \mathcal{Rel} \vdash a \cap 1 \leq aa$$

In fact, as shown in the sequel,  $\mathcal{Rel}$  is not finitely based on  $\mathcal{SP}^1$ ,  $\mathcal{SPT}$ , and  $\mathcal{SP}^{1\top}$  (so that the corresponding equational theories are not finitely based either).

## 3 Decomposability

We fix a signature  $X \in \{\mathcal{SP}, \mathcal{SP}^1, \mathcal{SPT}, \mathcal{SP}^{1\top}\}$  in this section, and we provide a necessary condition for finite axiomatisability of relations on  $X$  (and thus existence of graph homomorphisms between  $X$  graphs). This is essentially the same condition as the one used by Freyd

<sup>1</sup> even finite ones, that can easily be found with tools such as Mace4.

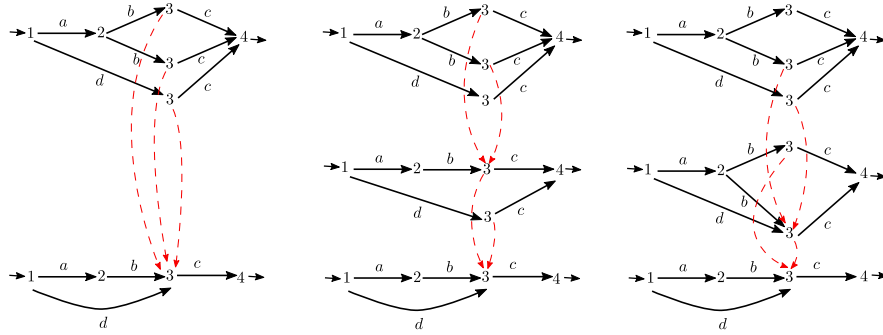
and Scedrov for representable allegories [12, pp 208-210]. We generalise it here so that it fits our needs, providing a different proof and more explicit treatment.

Recall that a homomorphism can be seen as the action of merging several vertices and edges of the source graph and adding some vertices and edges. The *degree* of a homomorphism is the number of vertices it merges:

► **Definition 5 (Degree).** Let  $h = (h_e, h_v)$  be a graph homomorphism. The degree of  $h$ , denoted  $\deg(h)$  is the number  $\#\{u \mid \exists w, w \neq u \text{ and } h_v(u) = h_v(w)\}$ . We write  $H \triangleleft_n G$  when there exists  $h : G \rightarrow H$  with  $\deg(h) \leq n$ .

Since the vertices of a graph can always be merged two at a time, every homomorphism can be decomposed into a sequence of homomorphisms of degree at most two. However, intermediate graphs in this decomposition are not necessarily in  $X$ , even if the endpoints are.

For instance, we depict on the left below a homomorphism of degree three between two SP graphs (top-down). This homomorphism can be decomposed into two sequences of homomorphisms of degree two, given in the middle and on the right. The intermediate graph in the middle is SP, while the intermediate graph on the right is not SP.



We write  $u \triangleleft_n^X v$  when  $u, v$  are terms of  $X$  and  $\mathcal{G}(u) \triangleleft_n \mathcal{G}(v)$ ; we write  $\triangleleft_n^{X^*}$  for the reflexive transitive closure of this relation.

In the above example, the valid law corresponding to  $(ab \cap d)c \triangleleft a(bc \cap bc) \cap dc$  can be decomposed into two valid laws  $(ab \cap d)c \triangleleft_2^{\text{SP}} abc \cap dc$  and  $abc \cap dc \triangleleft_2^{\text{SP}} a(bc \cap bc) \cap dc$ . These latter laws are intuitively simpler: they can be justified by homomorphisms with a smaller degree.

We need the following assumption about  $X$  to obtain Proposition 8 below.

► **Assumption 6.** There is an integer  $k$  such that for every term  $u$  of  $X$ , we have  $u \triangleleft_k^{X^*} u \cap u$ .

This is a rather mild assumption, which is satisfied in the context of the present paper:

► **Fact 7.** Assumption 6 is satisfied with  $k = 2$  for the four values of  $X$  considered here.

**Proof.** By a straightforward induction on  $u$  in each case. ◀

► **Proposition 8.** Under Assumption 6, if  $\text{Rel}$  is finitely based on  $X$ , then there exists an integer  $n$  such that for all terms  $u, v$  in  $X$ ,  $u \triangleleft v$  entails  $u \triangleleft_n^{X^*} v$ .

**Proof.** Suppose that we have a finite axiomatisation  $\mathcal{H}$ . By soundness (and Theorem 2), each axiom of  $\mathcal{H}$  gives rise to a graph homomorphism. Let  $n'$  be the maximal degree of these homomorphisms, and let  $n = \max\{k, n'\}$ . Now suppose  $u \triangleleft v$  for some terms  $u, v$  of  $X$ . By completeness (and Theorem 2), we get a derivation  $\mathcal{H} \vdash u \leq v$ . We prove  $u \triangleleft_n^{X^*} v$  by induction on this derivation. The only interesting case is that of the substitution rule. In this



case, we have  $u \triangleleft_n^{X^*} v$  by induction, and we must prove  $u\sigma \triangleleft_n^{X^*} v\sigma$  for a given substitution  $\sigma$ . W.l.o.g., we can assume  $u \triangleleft_n^X v$ , and we consider the underlying homomorphism  $h : v \rightarrow u$ , of degree at most  $n$ . Observe that  $h$  can be extended into a homomorphism  $h\sigma : v\sigma \rightarrow u\sigma$ . If  $h$  is injective on edges, then  $\deg(h\sigma) = \deg(h) \leq n$  and we are done. If instead  $h$  merges two  $a$ -labelled edges, then the degree of  $h\sigma$  is at least the number of inner vertices of  $\sigma(a)$ , which is not bounded by  $n$ , *a priori*. In this latter case, we use the homomorphism  $h' : v \rightarrow u'$  obtained from  $h$  by duplicating all edges in the image graph as many times as necessary to get injectivity on edges. The corresponding term  $u'$  is obtained from  $u$  by replacing some occurrences of letters, say  $a$ , with intersections of the same letter (e.g.,  $a \cap a \cap a$ ). The homomorphism  $h'\sigma : v\sigma \rightarrow u'\sigma$  satisfies  $\deg(h'\sigma) = \deg(h') = \deg(h) \leq n$ , so that  $u'\sigma \triangleleft_n^{X^*} v\sigma$ . We finally obtain  $u\sigma \triangleleft_k^{X^*} u'\sigma$  by repeatedly using Assumption 6.  $\blacktriangleleft$

By contraposition, to prove non-finite-axiomatisability, it suffices to find a sequence of homomorphisms  $(e_n \triangleleft f_n)_{n \in \omega}$  between terms of  $X$  such that for all  $n$ ,  $e_n \triangleleft_n^{X^*} f_n$  does not hold. We define such a sequence in the following section, for  $\text{SP1}$  and  $\text{SP1}\top$ .

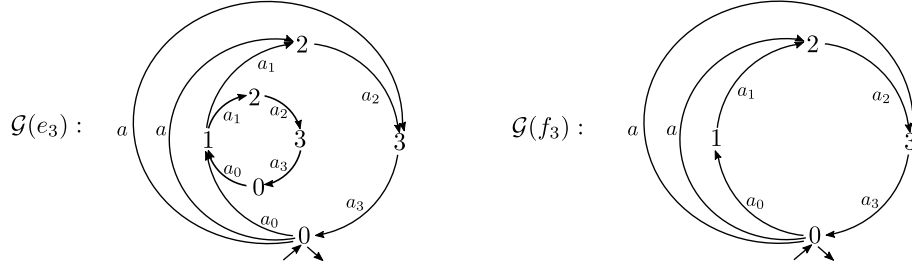
#### 4 The fragments $\text{SP1}$ and $\text{SP1}\top$

We show in this section that  $\mathcal{R}el$  is not finitely based on  $\text{SP1}$  and  $\text{SP1}\top$ .

► **Definition 9.** Let  $a, a_0, a_1, \dots$  be a fixed sequence of pairwise disjoint letters, and let  $n$  be a strictly positive integer. Let  $c_n$  be the term  $(a_1 a_2 \dots a_n a_0) \cap 1$ . We define the terms  $e_n$  and  $f_n$  with the help of the families  $(g_i^n)_{i \in [1, n-1]}$  and  $(h_i^n)_{i \in [1, n-1]}$  respectively as follows:

$$\begin{aligned} g_0^n &= a_0 c_n, & g_{i+1}^n &= (g_i^n a_{i+1}) \cap a, & e_n &= (g_{n-1}^n a_n) \cap 1. \\ h_0^n &= a_0, & h_{i+1}^n &= (h_i^n a_{i+1}) \cap a, & f_n &= (h_{n-1}^n a_n) \cap 1. \end{aligned}$$

For instance, the graphs of  $e_3$  and  $f_3$  are:



There is a homomorphism from  $\mathcal{G}(e_3)$  to  $\mathcal{G}(f_3)$ , which maps the nodes of  $\mathcal{G}(e_3)$  tagged by  $i \in [0, 3]$  to the node of  $\mathcal{G}(f_3)$  tagged  $i$ . More generally, it is not hard to see that, for every  $n \in \omega$ , there is a (unique) homomorphism from  $\mathcal{G}(e_n)$  to  $\mathcal{G}(f_n)$ , that is  $f_n \triangleleft e_n$ . Let us state the main proposition of this section:

► **Proposition 10.** For every  $n \in \omega$ ,  $f_n \not\triangleleft_n^{\text{SP1}\top^*} e_n$ .

Together with Proposition 8, this entails that  $\mathcal{R}el$  is not finitely based on  $\text{SP1}\top$ . Since  $e_n$  and  $f_n$  actually are  $\text{SP1}$  terms, this also entails that it is not finitely based on  $\text{SP1}$ .

We prove Proposition 10 below, by contradiction. In order to ease this proof, we first establish a property verified by  $\text{SP1}\top$  graphs; this will allow us to reach a contradiction in the two main cases of the proof.

► **Definition 11** (Back pattern). *A back pattern in a graph is a pair of distinct nodes  $m, n$  together with three directed paths:  $\pi$  from the input to  $m$ ,  $\kappa$  from  $n$  to  $m$ , and  $\rho$  from  $n$  to the output, such that  $\pi$  and  $\kappa$  intersect exactly on  $m$  and  $\kappa$  and  $\rho$  intersect exactly on  $n$ .*

Such a back pattern can be depicted as follows:  $\iota \xrightarrow{\pi} m \xleftarrow{\kappa} n \xrightarrow{\rho} o$ . They cannot arise in SP1T graphs. Intuitively, although SP1 graphs are not acyclic (unlike SP graphs), they are nevertheless oriented from the input towards the output.

► **Proposition 12.** *SP1T graphs do not contain back patterns.*

A proof is given found in Appendix B; it is an easy induction on the structure of SP1T terms, after adding some other forbidden patterns for the induction to go through. Proposition 12 can be extended to characterise SP1 graphs (and thus SP1T graphs via Corollary 4) using a slight relaxation of the definition of back patterns. Such a characterisation is not needed here, however, and we can now prove Proposition 10.

**Proof of Proposition 10.** Let  $h$  be the (unique) homomorphism from  $\mathcal{G}(e_n)$  to  $\mathcal{G}(f_n)$ . To simplify the presentation, we label the nodes of  $e_n$  and  $f_n$  by integers from  $[0, n]$ , in the same way as in the above example for  $n = 3$ : the input is labelled 0, and for every  $i \in [0, n - 1]$ , if a node is labelled  $i$ , then its  $a_i$  successors are labelled  $i + 1$ . Note that, for every  $i \in [1, n]$ , there is exactly one node labelled  $i$  in  $f_n$ ; its pre-image by  $h$  consists of those nodes labelled by  $i$  in  $\mathcal{G}(e_n)$ .

Suppose by contradiction that  $(f_n, e_n) \in \triangleleft_n^*$ . Thus, we can find SP1T terms  $(g_i)_{i \in [0, m+1]}$  and homomorphisms  $(h_i : g_i \rightarrow g_{i+1})_{i \in [0, m]}$  of degree at most  $n$ , and such that  $g_0 = e_n$  and  $g_{m+1} = f_n$ . The composition of the homomorphisms  $(h_i)_{i \in [0, m]}$  yields  $h$ .

Let  $k$  be an index such that  $e_n \hookrightarrow g_k$  and  $e_n \not\hookrightarrow g_{k+1}$ . This index exists since  $e_n \hookrightarrow g_0$  and  $e_n \not\hookrightarrow g_{m+1}$ . As  $\mathcal{G}(e_n)$  is a sub-graph of  $\mathcal{G}(g_k)$ , we label the nodes of  $\mathcal{G}(g_k)$  accordingly: keep the same labels for nodes in  $\mathcal{G}(e_n)$ , and do not label the other nodes.

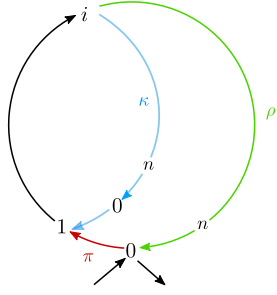
The fact that  $e_n \not\hookrightarrow g_{k+1}$  means that the homomorphism  $h_k$  merged some labelled nodes. Note that  $h_k$  cannot merge nodes tagged by different integers. Otherwise, the composition of the homomorphisms  $(h_i)_{i \in [0, m]}$  would also merge nodes tagged by different labels, which is not possible. We label the nodes of  $\mathcal{G}(g_{k+1})$  according to  $h_k$ : a node is labelled  $i$  if it is the image of a node labelled  $i$  by  $h_k$ .

Let us show that  $g_{k+1}$  cannot be an SP1T term. Let  $i$  be the largest integer in  $[1, n]$  such that for every  $j \in [1, i]$  the nodes labelled by  $j$  in  $\mathcal{G}(g_k)$  have been merged by  $h_k$ .

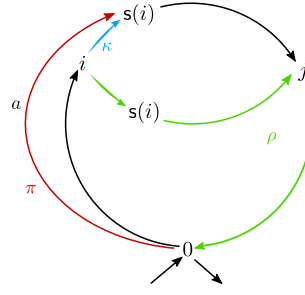
We define the function  $s : [1, n] \rightarrow [0, n]$  as follows:  $s(x) = x + 1$  if  $x \in [1, n - 1]$  and  $s(n) = 0$ . Note that the nodes labelled by  $s(i)$  in  $\mathcal{G}(g_k)$  are not merged by  $h_k$ . If this was the case, then either  $i \in [1, n - 1]$ , which would contradict maximality of  $i$ , or  $i = n$ , meaning that the degree of the homomorphism is at least  $n + 1$ , contradicting our hypothesis.

We distinguish two cases:

- For every  $j \in [i + 1, n] \cup \{0\}$ , the nodes labelled  $j$  in  $\mathcal{G}(g_k)$  are not merged by  $h_k$ . In this case,  $i \neq 1$ , otherwise no labelled nodes of  $\mathcal{G}(g_k)$  would be merged by  $h_k$ . There are two distinct directed paths from  $i$  to 1 in  $\mathcal{G}(g_{k+1})$ : one that visits the input and one that does not. We name the first one  $\kappa$ . The second can be decomposed into a path from  $i$  to the input, we name it  $\rho$ , and a path from the input to 1, we name it  $\pi$ . Then the nodes labelled  $i$  and 1, together with the paths  $\pi, \kappa, \rho$  form a back-pattern in  $\mathcal{G}(g_{k+1})$ , as illustrated in Figure 1.
- There is  $j \in [i + 1, n] \cup \{0\}$  such that the nodes labelled  $j$  are merged. Recall that the nodes labelled  $s(i)$  are not merged by  $h_k$ . We call  $m$  the node of  $\mathcal{G}(g_{k+1})$ , which is labelled  $s(i)$  and which is the  $a$  successor of the input. Note that since the nodes labelled  $j$  are



■ **Figure 1** First case in proof of Prop. 10



■ **Figure 2** Second case in proof of Prop. 10

merged, there is a path in  $\mathcal{G}(g_{k+1})$  connecting the node labelled  $i$  to the input (which in this case coincide with the output), which does not go through  $m$ . We call this path  $\rho$ . We call  $\pi$  the path labelled  $a$  from the input to  $m$  and  $\kappa$  the path labelled  $a_i$  from  $i$  to  $s(i)$ . The nodes  $s(i)$  and  $i$ , together with the paths  $\pi, \kappa$  and  $\rho$  form a back-pattern in  $\mathcal{G}(g_{k+1})$ , as illustrated in Figure 2. ◀

## 5 The fragment SPT

We show in this section that  $\mathcal{Rel}$  is not finitely based on SPT. We do not adopt the same strategy as for SP1<sup>2</sup>. Instead, our proof in this case is in two steps. First, we show that every axiomatisation can be turned into one with a very constrained shape, called *simple axiomatisation*. In a second step, we exhibit an infinite collection of inequations  $(f_n \leq e_n)_{n \in \omega}$  which any simple axiomatisation should contain in a certain sense.

### 5.1 Dealing with idempotency

Simple axiomatisations, which we introduce in the following, are axiomatisations where idempotency of intersection is used in a very controlled way, following Freyd and Scedrov' idea of *separatedness* [12, page 208].

We call *idempotency axiom* an inequation of the form  $a \leq a \cap a$  for some letter  $a$ .

► **Definition 13.** A term  $v$  is *simple* if every letter appears at most once in  $v$ . An inequation  $u \leq v$  is *simple* if  $v$  is simple. An axiomatisation  $\mathcal{H}$  is *simple* if contains only simple axioms and idempotency axioms.

The key intuition about simple axioms is that the corresponding homomorphisms cannot merge edges; in a sense, they are idempotency-free.

For every term  $v$ , there is a simple term  $v'$  and a renaming  $\theta_v$  such that  $v = v'\theta_v$  (e.g., for  $v = aa$ , take  $v' = a_1a_2$  and  $\theta_v = \{a_1, a_2 \mapsto a\}$ ). In such a case, write  $\theta_v^{-1}$  for the following substitution:  $\theta_v^{-1}(a) = \bigcap_{\theta_v(a')=a} a'$ . For every term  $u$  such that  $u \triangleleft v$ , we have  $u\theta_v^{-1} \triangleleft v'$ .

We can thus turn any sound axiomatisation  $\mathcal{H}$  into a simple axiomatisation  $\mathcal{H}'$  as follows: adjoin an idempotency axiom, and replace each non-simple axiom  $u \leq v$  of  $\mathcal{H}$  with  $u\theta_v^{-1} \leq v'$ . Every axiom of  $\mathcal{H}$  is derivable in  $\mathcal{H}'$  using its simple counterpart and the idempotency axiom; therefore if  $\mathcal{H}$  axiomatises  $\mathcal{Rel}$  then so does  $\mathcal{H}'$ . Note that  $\mathcal{H}'$  is finite whenever  $\mathcal{H}$  is finite.

<sup>2</sup> We actually conjecture that for all terms  $u, v$  in SPT,  $u \triangleleft v$  entails  $u \triangleleft_2^{\text{SPT}^*} v$ .

Idempotency can thus be isolated from the other axioms. We go further and show that its use may actually be pushed towards the leaves. Let  $\mathcal{I}$  be the following set of inequations

$$a \leq a \cap a \quad \top \leq \top \cap \top \quad (a \cap c)(b \cap d) \leq ab \cap cd \quad (a \cap c) \cap (b \cap d) \leq (a \cap b) \cap (c \cap d)$$

These axioms are sound w.r.t.  $\mathcal{R}el$ ; we need these axioms to obtain Proposition 15 below: every derivation can be seen as a sequence of rewriting steps where idempotency axioms are used only on letters (i.e., to merge parallel edges).

► **Definition 14.** *Given two terms  $e, f$ , if  $e = C[u\sigma]$  and  $f = C[v\sigma]$  for some context  $C$ , substitution  $\sigma$ , and terms  $u, v$ , we say that  $(C, \sigma)$  is a unifying context-substitution for  $e$  and  $f$  with inner terms  $u$  and  $v$ .*

► **Proposition 15.** *If  $\mathcal{I} \subseteq \mathcal{H}$  and  $\mathcal{H} \vdash f \leq e$ , then there is a sequence  $g_0, \dots, g_m$  of terms such that  $e = g_0$ ,  $g_m = f$ , and for all  $i < m$ , there is a unifying context-substitution for  $g_i$  and  $g_{i+1}$  with inner terms  $u$  and  $v$  such that  $(u \leq v) \in \mathcal{H}$ , and the substitution is a renaming if the latter axiom is an idempotency axiom.*

**Proof.** A simple induction on the derivation yields a sequence as in the statement, but without the constraint idempotency axioms. We refine this sequence by using the following property:

For every SPT term  $e$  there is a sequence  $g_0, \dots, g_m$  of terms such that  $e \cap e = g_0$ ,  $g_m = e$ , and for all  $i < m$ , there is a unifying context-substitution for  $g_i$  and  $g_{i+1}$  with inner terms  $u$  and  $v$  such that  $(u \leq v) \in \mathcal{I}$ , and the substitution is a renaming if the latter axiom is the idempotency axiom of  $\mathcal{I}$ .

This property is proved by an easy induction on  $e$ , using in each case the corresponding axiom of  $\mathcal{I}$ , e.g., in the product case,

$$ef \leq (e \cap e)f \leq (e \cap e)(f \cap f) \leq ef \cap ef$$

where the first two (sequences of) steps are obtained by induction hypothesis and the third step is an instance of the third axiom of  $\mathcal{I}$ . ◀

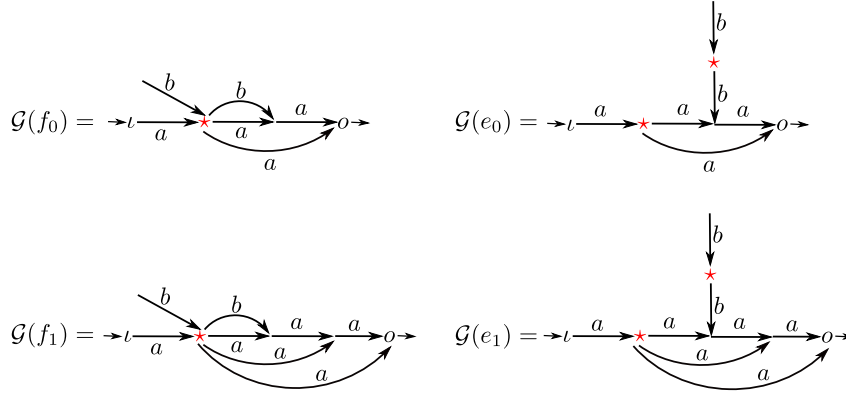
## 5.2 The counter-example

We can finally give the counter-example.

► **Definition 16.** *Let  $a, b$  be fixed letters. Given  $n \in \omega$ , the SPT terms  $e_n$  and  $f_n$  are defined as follows, with the help of two sequences  $(u_i^n)_{i \leq n}$   $(v_i^n)_{i \leq n}$  parameterised by  $n$ :*

$$\begin{aligned} u_0^n &= (\top bb \cap a)a \cap a & u_{i+1}^n &= (u_i^n a) \cap a & e_n &= au_n^n \\ v_0^n &= (a \cap b)a \cap a & v_{i+1}^n &= (v_i^n a) \cap a & f_n &= (\top b \cap a)v_n^n \end{aligned}$$

The graphs of  $f_0, e_0$  and  $f_1, e_1$  are depicted below.



There is a unique homomorphism  $!_n$  from  $e_n$  to  $f_n$ . This homomorphism is surjective, it just merges two vertices of the graph of  $e_n$ , which are tagged by red stars in the picture above.

The key property of the sequences  $(e_n)_{n \in \omega}$  and  $(f_n)_{n \in \omega}$  is stated in Proposition 18 below: every inequation  $e_n \leq f_n$  can be injected up-to renaming into an inequation of  $\mathcal{H}$ .

► **Definition 17.** We say that  $v$  injects up-to renaming into  $u$ , written  $v \hookrightarrow_\alpha u$ , if  $v \hookrightarrow u\theta$  for some renaming  $\theta$ .

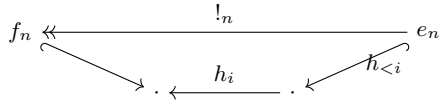
► **Proposition 18.** If  $\mathcal{H}$  is a simple axiomatisation of relations on  $\text{SPT}$ , then for every  $n \in \omega$ , there are  $\text{SPT}$  terms  $e'_n$  and  $f'_n$  such that  $f_n \hookrightarrow_\alpha f'_n$ ,  $e_n \hookrightarrow_\alpha e'_n$  and  $(f'_n \leq e'_n) \in \mathcal{H}$ .

As the size of the graphs of  $e_n$  and  $f_n$  grows infinitely, the set of inequations  $(f'_n \leq e'_n)_{n \in \omega}$  is infinite. Thus  $\mathcal{R}el$  is not finitely based on  $\text{SPT}$ . (Note that the above proposition is stronger than necessary: we could focus on one side of the equations.)

The proof of Proposition 18 relies on the three following lemmas, whose proofs can be found in Appendix C. The first one says that if  $!_n : e_n \twoheadrightarrow f_n$  decomposes into a sequence of homomorphisms, then of one them must essentially act like  $!_n$  under some irrelevant context.

► **Lemma 19.** Fix  $n \in \omega$  and assume  $!_n : e_n \twoheadrightarrow f_n$  decomposes into a sequence  $h_m \circ \dots \circ h_0$ . For  $i < m$ , write  $h_{<i}$  for the partial composition  $h_{i-1} \circ \dots \circ h_0$ . There exists an index  $i < m$  such that  $h_{<i}$  is injective and the image of  $h_{<i+1}$  is the graph of  $f_n$ .

In other words, in the above statement,  $h_{<i+1}$  factors through  $f_n$  as in the diagram below.



The second lemma essentially says that if there is a unifying context-substitution (Definition 14) for  $e_n$  and  $f_n$  whose inner terms are related by a homomorphism, then the context is necessarily trivial and  $e_n$  and  $f_n$  can be injected up-to renaming into the inner terms. We need to be slightly more flexible and we use the following notation: we write  $u \hookrightarrow v$  if there is a homomorphism from  $u$  to  $v$  which is bijective on edges and injective on vertices. In other words,  $u \hookrightarrow v$  when  $v$  is  $u$  with some additional isolated vertices.

► **Lemma 20.** Fix  $n \in \omega$  and suppose that there is a context  $C$ , a substitution  $\sigma$  and two terms  $u$  and  $v$  such that  $f_n \hookrightarrow C[v\sigma]$ ,  $e_n \hookrightarrow C[u\sigma]$ , and  $v \triangleleft u$ . Then  $f_n \hookrightarrow_\alpha v$  and  $e_n \hookrightarrow_\alpha u$ .

The last ingredient says that when a term  $e$  can be injected in a term of the form  $C[u\sigma]$ , for a simple term  $u$ , then we can restrict  $C, u$  and  $\sigma$  to match  $e$  up to some isolated vertices.

► **Lemma 21.** *If  $\iota : e \hookrightarrow C[u\sigma]$  with  $u$  simple, then there is a context  $C' \hookrightarrow C$ , a substitution  $\sigma' \hookrightarrow \sigma$  and a term  $u' \hookrightarrow u$  such that  $\iota$  decomposes into  $e \hookrightarrow C'[u'\sigma'] \hookrightarrow C[u\sigma]$ . (Where we extend  $\hookrightarrow$  to substitutions componentwise:  $\sigma \hookrightarrow \gamma$  if  $\text{dom}(\sigma) \subseteq \text{dom}(\gamma)$  and  $\forall a \in \text{dom}(\sigma), \sigma(a) \hookrightarrow \gamma(a)$ .)*

Note that the requirement that  $u$  must be simple cannot be dropped (because, e.g.,  $bc \hookrightarrow (b \cap c)(b \cap c) = (aa) \{a \mapsto b \cap c\}$ ).

**Proof of Proposition 18.** Let  $\mathcal{H}$  be a simple axiomatisation of relations on  $\text{SPT}$ . Since  $\mathcal{I}$  is simple and none of the  $f_n \leq e_n$  injects up-to renaming into  $\mathcal{I}$ , we can assume w.l.o.g that  $\mathcal{H}$  contains  $\mathcal{I}$ .

Let  $n \in \omega$ . In the rest of this proof we write  $e, f$ , and  $!$  for  $e_n, f_n$ , and  $!_n$ , respectively.

Since  $f \triangleleft e$ , we have  $\mathcal{H} \vdash f \leq e$  by completeness (and Theorem 2). By Proposition 15, we obtain a sequence  $g_0, \dots, g_m$  of terms such that  $e = g_0, g_m = f$ , and for all  $i < m$ , there is a unifying-context substitution  $(C_i, \sigma_i)$  for  $g_i$  and  $g_{i+1}$  with inner terms  $u_i$  and  $v_i$  (i.e.,  $g_i = C_i[u_i\sigma_i]$  and  $g_{i+1} = C_i[v_i\sigma_i]$ ) such that either  $(v_i \leq u_i) \in \mathcal{H}$  is simple or this is an idempotency axiom and  $\sigma_i$  is a renaming.

By soundness (and Theorem 2), there are homomorphisms  $h_i : u_i \rightarrow v_i$  for all  $i < m$ . Each of these homomorphisms can be extended, through the context  $C_i$  and the substitution  $\sigma_i$ , into a homomorphism  $h_i^\uparrow : g_i \rightarrow g_{i+1}$ . The composition of these homomorphisms must be the only homomorphism  $! : e \rightarrow f$ . Hence, by Lemma 19, there is an index  $i$  such that  $h_{<i}^\uparrow$  is injective and the image of  $h_{<i+1}^\uparrow$  is  $f$ . Since  $h_i$  merges two vertices, it cannot be the case that this step is an idempotency step: since those are restricted to letters, they only merge edges. Therefore  $u_i$  must be simple.

Since  $h_{<i}^\uparrow : e \hookrightarrow g_i = C_i[u_i\sigma_i]$ , Lemma 21 gives us a context  $C \hookrightarrow C_i$ , a substitution  $\sigma \hookrightarrow \sigma_i$  and a term  $u \hookrightarrow u_i$  such that  $\iota(e) \hookrightarrow C[u\sigma]$ . Call  $\iota$  the injective homomorphism  $u \hookrightarrow u_i$ .

The image of  $h_i \circ \iota : u \rightarrow v_i$ , as a subgraph of  $v_i$ , must be a  $\text{SPT}$  term  $v$  by Proposition 3. This decomposition corresponds to the commuting diagram on the left below, and assembling the various ingredients collected so far, we obtain the commuting diagram on the right.

$$\begin{array}{c}
v_i \xleftarrow{h_i} u_i \xleftarrow{\iota} u \\
\searrow \qquad \swarrow \\
\qquad \qquad v
\end{array}
\qquad
\begin{array}{c}
C_i[v_i\sigma_i] \xleftarrow{h_i^\uparrow} C_i[u_i\sigma_i] \xleftarrow{h_{<i}^\uparrow} C[u\sigma] \xleftarrow{\iota} e \\
\searrow \qquad \qquad \qquad \swarrow \\
C[v\sigma]
\end{array}$$

Since  $f$  is the image of  $h_{<i+1}^\uparrow$ , we deduce  $f \hookrightarrow C[v\sigma]$ .

We can finally use Lemma 20 to deduce  $e \hookrightarrow_\alpha u$  and  $f \hookrightarrow_\alpha v$ . As  $u \hookrightarrow u_i$  and  $v \hookrightarrow v_i$ , we also have that  $e \hookrightarrow_\alpha u_i$  and  $f \hookrightarrow_\alpha v_i$ . Since  $(v_i \leq u_i) \in \mathcal{H}$  this concludes the proof. ◀

## 6 Graph theoretical characterisation for natural structures

The characterisation of  $\text{Rel}$  in terms of graph homomorphisms (Theorem 2) works for all  $\text{SP1T}$  terms. This inequational theory coincides with that of sl-semigroups ( $\mathcal{SP}$ ) for  $\text{SP}$  terms, but we have seen that it departs from related algebraic structures (sl-monoids, bsl-semigroups and bsl-monoids) for  $\text{SP1}$ ,  $\text{SPT}$ , and  $\text{SP1T}$  terms. We show in this section that we can nevertheless obtain simple graph theoretical characterisations of the (in)equational theory of those three algebraic structures.

Let us focus on the  $\text{SP}$  case first, for which the notions we have used so far just work:

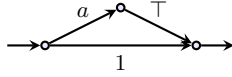
► **Proposition 22.** *For all  $u, v \in \mathcal{SP}$ ,  $\mathcal{SP} \vdash u \leq v$  iff  $\mathcal{G}(u) \triangleleft \mathcal{G}(v)$ .*

This result is a consequence of Theorem 2 and [3], but we give below a direct proof due to Brunet [5]. We say that a graph *goes forward* if every vertex belongs to a simple directed path from the input to the output, and so does every edge. All SP graphs go forward, which make it possible to obtain the following property:

► **Lemma 23.** *For all SP terms  $u, v_1, v_2$  such that  $\mathcal{G}(u) \triangleleft \mathcal{G}(v_1 v_2)$ , there are SP terms  $u_1, u_2$  such that  $\mathcal{G}(u_1) \triangleleft \mathcal{G}(v_1)$ ,  $\mathcal{G}(u_2) \triangleleft \mathcal{G}(v_2)$ , and  $\mathcal{SP} \vdash u \leq u_1 u_2$ .*

Proposition 22 follows easily (see Appendix D).

The above lemma cannot be adapted directly in the presence of 1 and  $\top$ , because the graphs no longer go forward. Instead, we start from a simpler interpretation of terms into graphs, ensuring that we keep forward graphs: we simply interpret 1 and  $\top$  as letters, so that graphs are now labelled in  $A \uplus \{1\} \uplus \{\top\}$ . We write  $\mathcal{G}'(\cdot)$  for the corresponding interpretation function. For instance,  $\mathcal{G}'(a \top \cap 1)$  is the following graph:



The previous notion of homomorphism is of course too strict, as it leaves the constants 1 and  $\top$  uninterpreted. We thus adjust the notion of homomorphism below. We first define some notations: given two vertices  $x, y$  in a graph as above, we write

- $x \Rightarrow y$  if there is a directed path from  $x$  to  $y$  along edges labelled with 1;
- $x \xrightarrow{a} y$  if  $x \Rightarrow x' \xrightarrow{a} y' \Rightarrow y$  for some vertices  $x'$  and  $y'$ ;
- $x \rightarrow^* y$  if there is a directed path from  $x$  to  $y$  along arbitrary edges;
- $x \rightarrow^+ y$  if there is a non-empty directed path from  $x$  to  $y$  along arbitrary edges.

► **Definition 24.** *Fix  $X$  in  $\{1, \top, 1\top\}$  and two graphs  $G, H$  as above. An  $X$ -homomorphism from  $G$  to  $H$  is a function  $h$  from the vertices of  $G$  to those of  $H$  preserving input and output, and such that:*

- (a) if  $x \xrightarrow{a} y$  in  $G$  then  $h(x) \xrightarrow{a} h(y)$  in  $H$ ;
- (1) for  $X \in \{1, 1\top\}$ , if  $x \xrightarrow{1} y$  in  $G$  then  $h(x) \Rightarrow h(y)$  in  $H$ ;
- ( $\top$ ) for  $X = \top$ , if  $x \xrightarrow{\top} y$  in  $G$  then  $h(x) \rightarrow^+ h(y)$  in  $H$ ;
- ( $\top'$ ) for  $X = 1\top$ , if  $x \xrightarrow{\top} y$  in  $G$  then  $h(x) \rightarrow^* h(y)$  in  $H$ ;

We write  $H \triangleleft^X G$  when there exists such a  $X$ -homomorphism.

We finally state our three characterisations, for sl-monoids, bsl-semigroups, and bsl-monoids.

► **Theorem 25.** *For all  $X$  in  $\{1, \top, 1\top\}$ , for all  $u, v \in \mathcal{SP}^X$ ,  $\mathcal{SP}^X \vdash u \leq v$  iff  $\mathcal{G}'(u) \triangleleft^X \mathcal{G}'(v)$ .*

The distinction in the clause for  $\top$ -labelled edges in the definitions of  $\top$ - and  $1\top$ -homomorphisms is required because bsl-monoids validate  $\top \leq \top\top$  (for instance, because  $\top = 1\top \leq \top\top$ ) while bsl-semigroups do not: allowing to use empty-paths with  $1\top$ -homomorphisms makes it possible to absorb one of the two edges of  $\mathcal{G}'(\top\top)$ , while this not possible with  $\top$ -homomorphisms.

Lemma 23 extends as follows:

► **Lemma 26.** *For all  $X$  in  $\{1, \top, 1\top\}$ , for all  $\mathcal{SP}^X$  terms  $u, v_1, v_2$  s.t.  $\mathcal{G}'(u) \triangleleft^X \mathcal{G}'(v_1 v_2)$ , there are  $\mathcal{SP}^X$  terms  $u_1, u_2$  such that  $\mathcal{G}'(u_1) \triangleleft^X \mathcal{G}'(v_1)$ ,  $\mathcal{G}'(u_2) \triangleleft^X \mathcal{G}'(v_2)$ , and  $\mathcal{SP} \vdash u \leq u_1 u_2$ .*

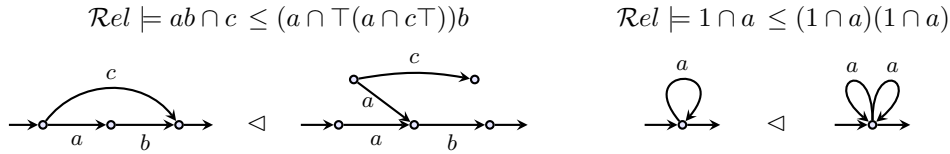
Like in the SP case, this lemma is proved by induction on the size of  $u$ , producing terms  $u_1, u_2$  such that  $\mathcal{G}'(u_1 u_2)$  is a subgraph of  $\mathcal{G}'(u)$  (in the strict sense), this is why we can get a derivation in  $\mathcal{SP}$  rather than in  $\mathcal{SP}^X$ .

Theorem 25 follows like in the SP case, by two inductions. The base case for letters is slightly more involved in the presence of 1; we give all details in Appendix D.

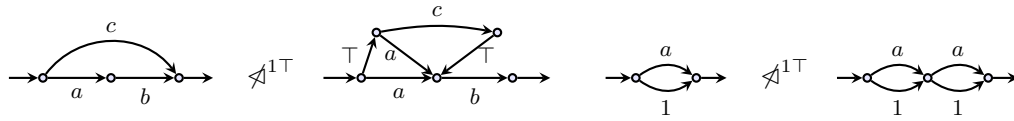
► **Corollary 27.** *The (in)equational theories of sl-monoids, bsl-semigroups, and bsl-monoids are decidable in polynomial time.*

**Proof.** The existence of an  $X$ -homomorphism from  $G$  to  $H$  is equivalent to the existence of a homomorphism from  $G$  to an appropriate closure of  $H$ . The graph of a term can obviously be computed in polynomial time, as well as its closure. The graph-homomorphism problem can be solved in polynomial time when the source graph has bounded treewidth [11, 7, 15], which is the case here (series-parallel graphs have treewidth at most two). ◀

As an example, consider the following homomorphisms, establishing two valid laws of  $\mathcal{Rel}$ :



Those are not laws of bsl-monoids: under the simpler interpretation  $\mathcal{G}'$ , we obtain the following pairs of graphs, which are not related by  $\triangleleft^{1\top}$ .



## 7 Conclusion

We have shown that on the signatures SP1, SP $\top$  and SP1 $\top$ ,  $\mathcal{Rel}$  is not finitely axiomatisable with a set of (in)equations. Does this change if we are more flexible on the shape of axioms? For example if Horn sentences or first-order formulas are allowed as axioms?

We have given a necessary condition on signatures for  $\mathcal{Rel}$  to be finitely based. This is the decomposability condition given in Proposition 8. Can we obtain a full characterisation?



## References

- 1 H. Andr eka and D. Bredikhin. The equational theory of union-free algebras of relations. *Algebra Universalis*, 33(4):516–532, 1995. doi:10.1007/BF01225472.
- 2 H. Andr eka and S. Mikul as. Axiomatizability of positive algebras of binary relations. *Algebra universalis*, 66(7), 2011. An erratum appears at <http://www.dcs.bbk.ac.uk/~szabolcs/AM-AU-err6.pdf>. doi:10.1007/s00012-011-0142-3.
- 3 D. A. Bredihin and B. M. Schein. Representations of ordered semigroups and lattices by binary relations. *Colloquium Mathematicae*, 39(1):1–12, 1978. URL: <http://eudml.org/doc/266458>.
- 4 D. A. Bredikhin. The equational theory of relation algebras with positive operations. *Izv. Vyssh. Uchebn. Zaved. Mat.*, 37(3):23–30, 1993. In Russian. URL: <http://mi.mathnet.ru/ivm4374>.
- 5 P. Brunet. The equational theory of algebras of languages. Talk at RAMiCS, Lyon, May 2017 (Special session on mechanised reasoning), 2017.
- 6 P. Brunet and D. Pous. Petri automata for Kleene allegories. In *LICS*, pages 68–79. ACM, 2015. doi:10.1109/LICS.2015.17.
- 7 C. Chekuri and A. Rajaraman. Conjunctive query containment revisited. *Theoretical Computer Science*, 239(2):211–229, 2000. doi:10.1016/S0304-3975(99)00220-0.
- 8 R. Diestel. *Graph Theory*. Graduate Texts in Mathematics. Springer Verlag, 2005.
- 9 A. Doumane and D. Pous. Completeness for identity-free Kleene lattices. In *CONCUR*, volume 118 of *LIPICs*, pages 18:1–18:17. Schloss Dagstuhl, 2018. doi:10.4230/LIPICs.CONCUR.2018.18.
- 10 Z.  sik and L. Bern atsky. Equational properties of Kleene algebras of relations with conversion. *Theoretical Computer Science*, 137(2):237–251, 1995. doi:10.1016/0304-3975(94)00041-G.
- 11 E. C. Freuder. Complexity of k-tree structured constraint satisfaction problems. In *NCAI*, pages 4–9. AAAI Press / The MIT Press, 1990. URL: <http://www.aaai.org/Library/AAAI/1990/aaai90-001.php>.
- 12 P. Freyd and A. Scedrov. *Categories, Allegories*. North Holland, 1990.
- 13 J. L. Gischer. The equational theory of pomsets. *Theoretical Computer Science*, 61(2):199–224, 1988. doi:10.1016/0304-3975(88)90124-7.
- 14 J. Grabowski. On partial languages. *Fundamenta Informaticae*, 4:427–498, 1981.
- 15 M. Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM*, 54(1):1:1–1:24, 2007. doi:10.1145/1206035.1206036.
- 16 T. Hoare, B. M oller, G. Struth, and I. Wehrman. Concurrent Kleene algebra and its foundations. *Journal of Logic and Algebraic Programming*, 80(6):266–296, 2011.
- 17 I. Hodkinson and S. Mikul as. Axiomatizability of reducts of algebras of relations. *Algebra Universalis*, 43:127–156, 2000. doi:10.1007/s000120050150.
- 18 R. Maddux. *Relation Algebras*. Elsevier, 2006.
- 19 S. Mikul as. Axiomatizability of algebras of binary relations. In *Classical and New Paradigms of Computation and their Complexity Hierarchies*, pages 187–205. Springer Netherlands, 2004. doi:10.1007/978-1-4020-2776-5\_11.
- 20 D. Monk. On representable relation algebras. *The Michigan mathematical journal*, 31(3):207–210, 1964. doi:10.1307/mmj/1028999131.
- 21 Y. Nakamura. Partial derivatives on graphs for kleene allegories. In *Lics*, pages 1–12. IEEE, 2017. doi:10.1109/LICS.2017.8005132.
- 22 D. Pous and V. Vignudelli. Allegories: decidability and graph homomorphisms. In *LiCS*, pages 829–838. ACM, 2018. doi:10.1145/3209108.3209172.
- 23 V. Pratt. Origins of the calculus of binary relations. In *LICS*, pages 248–254. IEEE Computer Society Press, 1992.
- 24 A. Tarski. On the calculus of relations. *J. of Symbolic Logic*, 6(3):73–89, 1941.
- 25 A. Tarski and S. Givant. *A Formalization of Set Theory without Variables*, volume 41 of *Colloquium Publications*. American Mathematical Society, Providence, Rhode Island, 1987.

## A Removing isolated vertices

We complete here the proof of Proposition 3 given in the main text. We deal with the  $\text{SP1}\top$  case first, which is slightly easier, then we explain how to deal with the  $\text{SPT}$  case.

### The $\text{SP1}\top$ case

W.l.o.g., it remains to show that for every term  $u_0$  with an isolated inner vertex, there is a term  $u$  such that  $\mathcal{G}(u)$  is  $\mathcal{G}(u_0)$  with the isolated vertex removed (equivalently,  $\mathcal{G}(u \cap \top\top) \simeq \mathcal{G}(u_0)$ , where  $\simeq$  denotes graph isomorphism).

In the sequel, we call *strict* a term  $u$  whose input and output in  $\mathcal{G}(u)$  are distinct.

Recall that given a term  $u$ , edges in  $\mathcal{G}(u)$  are in one-to-one correspondence with letter occurrences in  $u$ . There is a similar characterisation for inner vertices: inner vertices of  $\mathcal{G}(u)$  are in one-to-one correspondence with subterms of  $u$  of the shape  $v \cdot w$  such that  $v$  and  $w$  are strict. For instance, the graph of  $a \cdot ((b \cdot c) \cdot (d \cap 1))$  has two inner vertices, which can be associated to the first two occurrences of  $\cdot$  in the term; the third occurrence of  $\cdot$  does not count, because its second argument,  $(d \cap 1)$  is not strict.

Therefore, every isolated inner vertex in  $\mathcal{G}(u_0)$  corresponds to a subterm  $v \cdot w$  with both  $v$  and  $w$  strict, and such that the output of  $\mathcal{G}(v)$  and the input of  $\mathcal{G}(w)$  are isolated. It suffices to replace this subterm by  $(v \cap 1)\top(w \cap 1)$  to obtain the desired term  $u$ . ◀

For instance, the term  $(a\top)(\top b)$  denotes a graph with a single isolated inner vertex; we obtain the term  $(a\top \cap 1)\top(\top b \cap 1)$  using the above procedure.

### The $\text{SPT}$ case

While the correspondence about inner vertices is slightly simpler for  $\text{SPT}$  terms (they are all strict), the difficulty is that the term  $(v \cap 1)\top(w \cap 1)$  used to replace the subterm  $v \cdot w$  is not in  $\text{SPT}$ . We have to work a little bit more to find an equivalent  $\text{SPT}$  term (e.g., in the above example,  $a\top b$ ).

Let us say that a  $\text{SPT}$  term is *empty* when its graph is the graph of  $\top$  (equivalently, when this term is an arbitrary intersection of  $\top$  elements). We define the following auxiliary functions, defined by structural induction. The function  $r^o(\cdot)$  takes a non-empty term  $u$  such that the output of  $\mathcal{G}(u)$  is isolated, and returns a term such that  $\mathcal{G}(r^o(u)\top) \simeq \mathcal{G}(u)$ . Similarly for  $r^i(\cdot)$ , which takes a non-empty term  $u$  such that the input of  $\mathcal{G}(u)$  is isolated, and returns a term such that  $\mathcal{G}(\top r^i(u)) \simeq \mathcal{G}(u)$ .

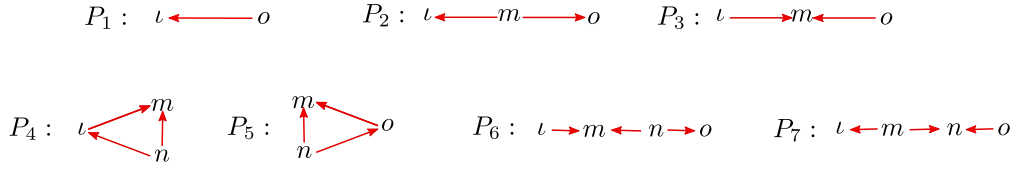
These functions are not defined on letters and  $\top$ : those terms do not satisfy the above constraints. In the second subcase in the intersection case,  $v$  must be non-empty since the argument is supposed to be non-empty; using  $u \cap r^o(v)$  (resp.  $u \cap r^i(v)$ ) in the case where both  $u$  and  $v$  are not empty would work equally well.

$$r^o(u \cap v) \triangleq \begin{cases} r^o(u) \cap v & \text{if } u \text{ is not empty} \\ r^o(v) & \text{otherwise} \end{cases} \quad r^o(u \cdot v) \triangleq \begin{cases} u & \text{if } v \text{ is empty} \\ u \cdot r^o(v) & \text{otherwise} \end{cases}$$

$$r^i(u \cap v) \triangleq \begin{cases} r^i(u) \cap v & \text{if } u \text{ is not empty} \\ r^i(v) & \text{otherwise} \end{cases} \quad r^i(u \cdot v) \triangleq \begin{cases} v & \text{if } u \text{ is empty} \\ r^i(u) \cdot v & \text{otherwise} \end{cases}$$

For example, we have  $r^o(a\top) = a$  and  $r^o(a\top \cap b\top) = a \cap b\top$ .

We finally resume with the main argument. If both  $v$  and  $w$  are empty, then it suffices to use  $\top$  in order to remove the isolated inner vertex. If both of them are non-empty, then we use  $r^o(v) \cdot r^i(w)$ . If only  $v$  is empty then we use  $\top \cdot r^i(w)$ . If only  $w$  is empty then we use  $r^o(v) \cdot \top$ . ◀



■ **Figure 3** Forbidden patterns.

## B Forbidden patterns for SP1T graphs

We prove here Proposition 12. The proof goes by induction on SP1T expressions, but we need to add some other forbidden patterns for the induction to go through. We call them back-patterns, and define them below.

► **Definition 28** (Back patterns). *Let  $G$  be a graph whose input is  $\iota$  and output is  $o$ . The back patterns of  $G$  are given in Figure 3. They should be understood this way: the nodes appearing in each pattern should be pairwise distinct, the red lines are simple paths of  $G$ , and the intersection of two paths is exactly the common nodes they have in the pattern.*

The back pattern introduced in the body of the paper (Definition 11) corresponds either to the pattern  $P_4$  (or  $P_5$ ) when  $\iota = o$ , or to  $P_6$  when  $\iota \neq o$  respectively.

► **Proposition 29** (Proposition 12 in the main text). *Graphs of SP1T expressions do not contain back patterns.*

**Proof.** We proceed by induction to prove Proposition 12. Actually, will show it for a larger class of expressions containing SP1T, which we denote by  $\text{SP1T}^\circ$  and define as follows

$$e, f ::= e \cdot f \mid e \cap f \mid 1 \mid \top \mid a \mid e^\circ \quad (a \in A)$$

The graph of an expressions is defined as usual, by induction on expressions, the graph of the expression  $e^\circ$  is obtained from the graph of  $e$  by identifying the input and the output.

We say that a graph is a *loop* if it is not strict, i.e., if its input is equal to its output. We call *loop* the new operator because it transforms graphs into loops. Note that if the graph of  $e \cap f$  is a loop, then

$$\mathcal{G}(e \cap f) = \mathcal{G}(e^\circ \cdot f^\circ)$$

This observation will allow us to analyse less cases in the induction, as we will see later. This is the reason why we introduced this loop operator.

Let  $e$  and  $f$  be some SP1T expressions. Let us start with some observations concerning the paths in the graphs of  $ef$ ,  $e \cap f$  and  $e^\circ$ .

► **Observation 30.** *We call checkpoint of  $\mathcal{G}(ef)$  the node of  $\mathcal{G}(ef)$  which is the output of  $\mathcal{G}(e)$  (and the input of  $\mathcal{G}(f)$ ). If  $u$  is a node of  $\mathcal{G}(e)$ ,  $v$  a node of  $\mathcal{G}(f)$  and  $p$  an undirected path in  $\mathcal{G}(ef)$  from  $u$  to  $v$ , then  $p$  goes necessarily through the checkpoint of  $\mathcal{G}(ef)$ .*

The reason is that the checkpoint is, by construction, the unique point of  $\mathcal{G}(ef)$  which is common to both  $\mathcal{G}(e)$  and  $\mathcal{G}(f)$ .

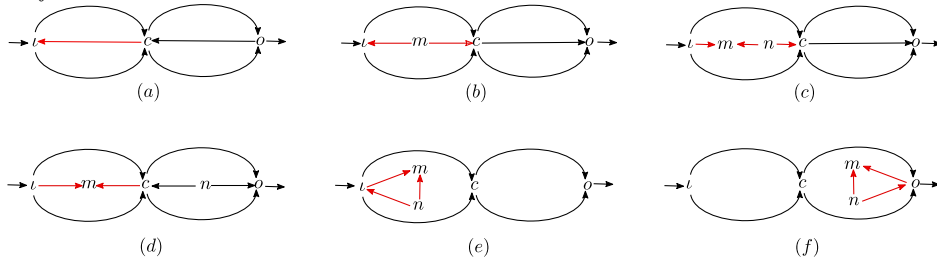
► **Observation 31.** *If  $u$  is a node of  $\mathcal{G}(e)$ ,  $v$  a node of  $\mathcal{G}(f)$  and  $p$  an undirected path in  $\mathcal{G}(e \cap f)$  from  $u$  to  $v$ , then  $p$  goes necessarily either through the input or the output of  $\mathcal{G}(ef)$ .*

► **Observation 32.** *If  $u$  is a node of  $\mathcal{G}(e^\circ)$  and  $p$  an undirected simple path in  $\mathcal{G}(e^\circ)$  from the input to  $u$ , then  $p$  is also a path of  $\mathcal{G}(e)$  either from the input or the output to  $u$ .*

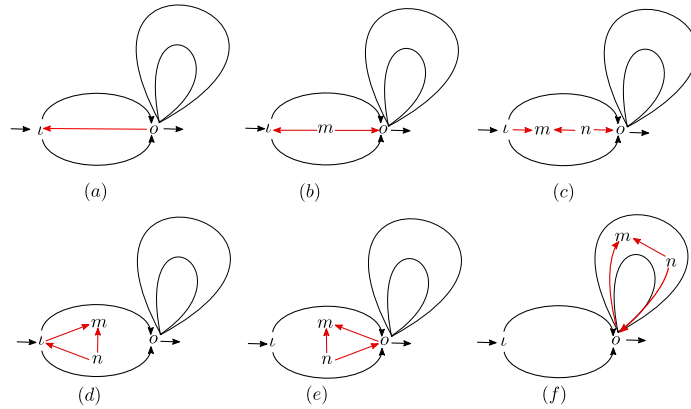
We prove Proposition 12 by induction on expressions. Clearly, the graphs of the expressions 1 and  $a$  where  $a \in A$  do not contain the back pattern. Consider now the expression  $e$  and  $f$ , both satisfying the induction hypothesis.

**Composition.** Let us show that  $ef$  satisfies also the induction hypothesis. For that we will show that a back pattern in  $ef$  induces necessarily a back pattern either in  $e$  or  $f$ , which contradicts the induction hypothesis. For that we will proceed by a case analysis on  $e$  and  $f$ : they are either both strict graphs, one of them is a loop and the other is strict or both are loops. For each of this situations, we see which bad patterns can happen in  $ef$  and how to extract bad patterns for  $e$  or  $f$  out of them. All these cases will be illustrated by pictures.

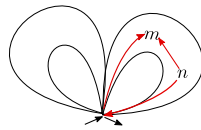
■  $e$  and  $f$  are strict.



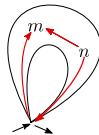
■  $e$  is strict and  $f$  is a loop. The case when  $e$  is a loop and  $f$  is strict is treated in a similar way.



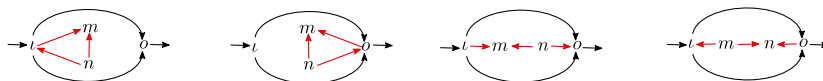
■  $e$  and  $f$  are loops.



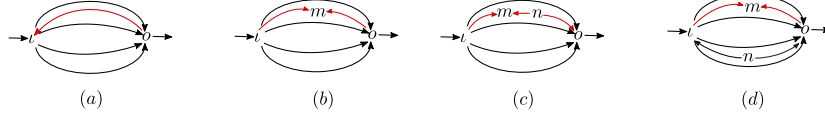
**Loops.** Let us show that  $e^\circ$  satisfies also the induction hypothesis. Note that since the graph of  $e^\circ$  is a loop, it can contain only one possible bad pattern, as illustrated below



Hence, by observation 32, the graph of  $e$  contains one of these back patterns



**Intersection.** Let us show that  $e \cap f$  satisfies the induction hypothesis. We can consider only the case when  $e \cap f$  is strict. Indeed, if this was not the case, we have that  $\mathcal{G}(e \cap f) = \mathcal{G}(e \circlearrowleft f \circlearrowright)$ . In this case, we can use the composition and loop cases.



## C Non-axiomatisability of relations on SPT

In this section, we prove the three lemmas needed to obtain Proposition 18.

► **Lemma 33** (Lemma 19 in the main txt). *Fix  $n \in \omega$  and assume  $!_n : e_n \rightarrow f_n$  decomposes into a sequence  $h_m \circ \dots \circ h_0$ . For  $i < m$ , write  $h_{<i}$  for the partial composition  $h_{i-1} \circ \dots \circ h_0$ . There exists an index  $i < m$  such that  $h_{<i}$  is injective and the image of  $h_{<i+1}$  is the graph of  $f_n$ .*

**Proof.** First observe that for all  $j \leq m+1$ ,  $h_{<j}$  is injective on edges, because  $!_n$  is so, and merges at most the two star vertices of  $e_n$ , because  $!_n$  only merges those two vertices. It suffices to take for  $i$  the least index  $i$  such that  $h_{<i+1}$  merges the two star vertices of  $e_n$ .  $h_{<i}$  is injective and the image of  $h_{<i+1}$  is  $f_n$ . ◀

► **Lemma 34** (Lemma 20 in the main text). *Fix  $n \in \omega$  and suppose that there is a context  $C$ , a substitution  $\sigma$  and two terms  $u$  and  $v$  such that  $f_n \dot{\leftrightarrow} C[v\sigma]$ ,  $e_n \dot{\leftrightarrow} C[u\sigma]$ , and  $v \triangleleft u$ . Then  $f_n \dot{\leftrightarrow}_\alpha v$  and  $e_n \dot{\leftrightarrow}_\alpha u$ .*

**Proof.** We first show that if  $e_n$  and  $f_n$  have a unifying context, then it is necessarily a hole. Notice first that  $e_n$  and  $f_n$  are of the following form

$$e_n = a \cdot (a \cap X) \quad f_n = (a \cap Y) \cdot (a \cap Z)$$

where  $X, Y, Z$  are SPT terms. Thus, if  $e_n$  (resp.  $f_n$ ) is written as an intersection  $f \cap g$ , then  $f \simeq \top$  or  $g \simeq \top$ . Similarly, if  $e_n$  (resp.  $f_n$ ) is written as a composition  $g \cdot h$ , then necessarily  $g = a$  and  $h = (a \cap X)$  (resp.  $g = a \cap Y$  and  $h = (a \cap Z)$ ).

We prove by induction on  $C$  that if there are two terms  $U, V$  such that  $e_n \dot{\leftrightarrow} C[U]$  and  $f_n \dot{\leftrightarrow} C[V]$ , then  $\bullet \dot{\leftrightarrow} C$ . The context  $C$  cannot be  $\top$  or a letter other than  $\bullet$ . If  $C = C_1 \cdot C_2$ , then by the remark above

$$a \dot{\leftrightarrow} C_1[U], \quad a \cap X \dot{\leftrightarrow} C_2[U], \quad a \cap Y \dot{\leftrightarrow} C_1[V], \quad a \cap Z \dot{\leftrightarrow} C_2[V].$$

If the hole is in  $C_1$ , then  $C_2[U] = C_2 = C_2[V]$ , but  $a \cap X \neq a \cap Z$  (even up to isolated vertices). If the hole is in  $C_2$ , then  $C_1[U] = C_1 = C_1[V]$ , but  $a \neq a \cap Y$  (idem). Therefore, either  $C = \bullet$ , in which case we are done, or  $C = C_1 \cap C_2$ , then by the remark above, either  $\top \dot{\leftrightarrow} C_1$  or  $\top \dot{\leftrightarrow} C_2$ . Say  $\top \dot{\leftrightarrow} C_1$ ; we deduce  $e_n \dot{\leftrightarrow} C_2[U]$  and  $f_n \dot{\leftrightarrow} C_2[V]$ , and thus  $\bullet \dot{\leftrightarrow} C_2$  by induction. We finally deduce  $\bullet \simeq \top \cap \bullet \dot{\leftrightarrow} C_1 \cap C_2 = C$ .

Thus  $\bullet \dot{\leftrightarrow} C$ , so that  $e_n \dot{\leftrightarrow} u\sigma$  and  $f_n \dot{\leftrightarrow} v\sigma$ . We can moreover assume w.l.o.g. that the letters in  $\text{dom}(\sigma)$  appear either in  $u$  or in  $v$ .

Let us say that a term  $u$  is *simple* if  $a \dot{\leftrightarrow} u$ ,  $b \dot{\leftrightarrow} u$ , or  $\top \dot{\leftrightarrow} u$ . Observe that the only common sub-terms of  $e_n$  and  $f_n$  modulo graph isomorphism are simple. We prove that  $\forall x \in \text{dom}(\sigma)$ ,  $\sigma(x)$  is simple.

- if  $x \in \text{dom}(\sigma)$  appears in  $u$ , then since there is a homomorphism from  $u$  to  $v$ ,  $x$  also appears in  $v$ . Thus  $e_n$  and  $f_n$  share the sub-term  $\sigma(x)$ , which must be simple by the observation above.
- if  $x \in \text{dom}(\sigma)$  appears in  $v$  but not in  $u$ , then suppose by contradiction that  $\sigma(x)$  is not simple. Then we obtain a non-surjective homomorphism from  $e_n$  to  $f_n$ , which is not possible.

Let  $\theta$  be the renaming defined as follows: for all  $x \in \text{dom}(\sigma)$  such that  $a \hookrightarrow \sigma(x)$  (resp.  $b \hookrightarrow \sigma(x)$ ), we set  $\theta(x) = a$  (resp.  $\theta(x) = b$ ). Since  $e_n$  and  $f_n$  do not contain isolated vertices, we have  $e_n \hookrightarrow u\theta$  and  $f_n \hookrightarrow v\theta$ , which concludes the proof. ◀

The remaining lemma is:

► **Lemma 35** (Lemma 21 in the main text). *If  $\iota : e \hookrightarrow C[u\sigma]$  with and  $u$  simple, then there is a context  $C' \hookrightarrow C$ , a substitution  $\sigma' \hookrightarrow \sigma$  and a term  $u' \hookrightarrow u$  such that  $\iota$  decomposes into  $e \hookrightarrow C'[u'\sigma'] \hookrightarrow C[u\sigma]$ . (Where we extend  $\hookrightarrow$  to substitutions componentwise:  $\sigma \hookrightarrow \gamma$  if  $\text{dom}(\sigma) \subseteq \text{dom}(\gamma)$  and  $\forall a \in \text{dom}(\sigma), \sigma(a) \hookrightarrow \gamma(a)$ .)*

This lemma follows by applying iteratively the following one. Indeed, applying a substitution  $\sigma$  to a simple term can be done one letter at a time, seeing the term to substitute as a context whose hole is the occurrence of the considered letter.

► **Lemma 36.** *If  $\iota : e \hookrightarrow C[u]$ , then there is a context  $C' \hookrightarrow C$  and a term  $u' \hookrightarrow u$  such that  $\iota$  decomposes into  $e \hookrightarrow C'[u'] \hookrightarrow C[u]$ .*

**Proof.** First notice that the edges of  $C[u]$  belong either to  $u$  or to  $C$ , and that the inner vertices of  $C[u]$  are either inner vertices of  $e$ , or inner vertices of  $C$  (in both cases in an exclusive manner). Call  $x$  and  $y$  the endpoints of the  $\bullet$ -labelled edge in  $C$ .

We rely on Corollary 4 to prove the lemma: since the class of  $\text{SP}\top$  graphs is closed under subgraphs, we can define terms by considering subgraphs of other terms.

Let  $u'$  be the term obtained by considering the subgraph of  $u$  where only the edges and vertices in the image of  $\iota$  are kept. Accordingly, let  $C'$  be the context obtained from  $C$  by keeping only the edges and vertices in the image of  $\iota$ , as well as the  $\bullet$ -labelled edge if  $u'$  is not empty.  $\iota$  corestricts to  $C'[u']$  by definition, and this corestriction is surjective on edges by definition, so that  $e \hookrightarrow C'[u']$ .

(Note that we cannot get  $\iota(e) \simeq C'[u']$  in general, due to situations where the input (resp. output) of  $u'$  is isolated as well as the source (resp. target) of the  $\bullet$ -edge in  $C'$ . This happens for instance with  $e = \top a \cap b$ ,  $C = \top \bullet \cap b$ ,  $u = \top a$ , where the above strategy yields  $C' = C$  and  $u' = u$ .) ◀

## D Graph theoretical characterisation for natural structures

We give detailed proofs for Section. 6. First note that when  $u$  is a  $\text{SP}$  term,  $\mathcal{G}(u)$  and  $\mathcal{G}'(u)$  coincide. We extend the notion of  $X$ -homomorphism and the associated notation  $\triangleleft^X$  to the case where  $X$  is empty, by using plain homomorphisms and  $\triangleleft$  in that case. This makes it possible to capture both Proposition 22 and Theorem 25 at once, as well as Lemmas 23 and 26.

► **Lemma 37** (Lemmas 23 and 26 in the main text). *For all  $X$  in  $\{, 1, \top, 1\top\}$ , for all  $\text{SP}X$  terms  $u, v_1, v_2$  s.t.  $\mathcal{G}'(u) \triangleleft^X \mathcal{G}'(v_1 v_2)$ , there are  $\text{SP}X$  terms  $u_1, u_2$  such that  $\mathcal{G}'(u_1) \triangleleft^X \mathcal{G}'(v_1)$ ,  $\mathcal{G}'(u_2) \triangleleft \mathcal{G}'(v_2)$ , and  $\mathcal{SP} \vdash u \leq u_1 u_2$ .*

**Proof.** We proceed by induction on the size of  $u$  and we write  $u = u^1 \cdots u^n$  such that none of the  $u^i$  start with a product. Then we perform a case analysis on the image of the vertex  $x$  in the middle of  $\mathcal{G}(v_1) \cdot \mathcal{G}(v_2)$  through the homomorphism.

- if it is the vertex shared by the subgraphs of  $u^i$  and  $u^{i+1}$  for some  $i$ , then since the graphs are forward, the starting homomorphism restricts to two homomorphisms  $\mathcal{G}(u^1 \dots u^i) \triangleleft \mathcal{G}(v_1)$  and  $\mathcal{G}(u^{i+1} \dots u^n) \triangleleft \mathcal{G}(v_2)$  so that we can conclude by associativity of  $\cdot$ .
- if it is the input (of  $u$ ), then necessarily  $X = 1$  or  $X = 1\top$  and we can return  $u_1 = 1$  and  $u_2 = u$  (because: 1/ the graph  $\mathcal{G}'(v_1)$  is strict and forward in the SP and SP $\top$  cases and the corresponding notions of homomorphisms do not permit to map it to a non-strict graph and 2/ in the other cases,  $v_1$  may not contain any letter by the same argument, so that we do have  $\mathcal{G}'(1) \triangleleft^X \mathcal{G}'(v_1)$ ).
- the case where  $x$  is mapped to the output is symmetrical.
- otherwise,  $x$  must be mapped to an inner vertex of the subgraph of  $u^i$  for some  $i$ . Since the graph of  $u^i$  contains an inner vertex, it cannot be just a letter, 1, or  $\top$ . Since it does not start with a product, it must be an intersection  $u_0^i \cap u_1^i$ . W.l.o.g., the image of  $x$  belongs to the graph of  $u_0^i$ . Then we observe that since the considered graphs go forward, the starting homomorphism corestricts to the graph of  $u' \triangleq u^1 \dots u^{i-1} u_0^i u^{i+1} \dots u^n$ , where the subgraph corresponding to  $u_1^i$  has been removed (for this observation to hold, it is crucial that we work here with forward graphs). Since  $u'$  is smaller than  $u$  we can use the induction hypothesis and conclude using associativity and monotonicity of  $\cdot$ :  $u = u^1 \dots u^{i-1} (u_0^i \cap u_1^i) u^{i+1} \dots u^n \leq u^1 \dots u^{i-1} u_0^i u^{i+1} \dots u^n = u'$ . ◀

We need the following two lemmas to prove the main theorem: a term whose graph contains a 1-labelled path from input to output is provably smaller than 1, and similarly, terms whose graphs contain a  $\overset{a}{\Rightarrow}$  path from input to output are provably smaller than  $a$ .

► **Lemma 38.** *For all  $u \in \text{SP}1\top$  such that  $\iota \Rightarrow o$  in  $\mathcal{G}'(u)$ , we have  $\text{SP}^1 \vdash u \leq 1$ .*

**Proof.** By induction on  $u$ .

- the assumption is not satisfied when  $u$  is a letter or  $\top$ , and the conclusion is trivial when  $u$  is 1;
- if  $u = v \cap w$ , then the path  $\iota \Rightarrow o$  must belong either to  $\mathcal{G}'(v)$  or to  $\mathcal{G}'(w)$ , so that we deduce either  $\text{SP}^1 \vdash v \leq 1$  or  $\text{SP}^1 \vdash w \leq 1$  by induction, and we conclude by transitivity with  $\text{SP}^1 \vdash u \leq v$  or  $\text{SP}^1 \vdash u \leq w$ ;
- if  $u = v \cdot w$ , then there must be paths  $\iota \Rightarrow o$  in both  $\mathcal{G}'(v)$  and  $\mathcal{G}'(w)$ , so that we deduce  $\text{SP}^1 \vdash v \leq 1$  and  $\text{SP}^1 \vdash w \leq 1$  by induction, from which we conclude by monotonicity and  $\text{SP}^1 \vdash 1 \cdot 1 \leq 1$ . ◀

► **Lemma 39.** *Fix  $X$  in  $\{1, \top, 1\top\}$ . For all  $u \in \text{SP}X$  such that  $\iota \overset{a}{\Rightarrow} o$  in  $\mathcal{G}'(u)$ , we have  $\text{SP}^X \vdash u \leq a$ .*

**Proof.** By induction on  $u$ .

- the assumption is not satisfied when  $u$  is 1,  $\top$ , or a letter distinct from  $a$ , and the conclusion is trivial when  $u$  is  $a$ ;
- if  $u = v \cap w$ , then the path  $\iota \overset{a}{\Rightarrow} o$  must belong either to  $\mathcal{G}'(v)$  or to  $\mathcal{G}'(w)$ , so that we deduce either  $\text{SP}^X \vdash v \leq a$  or  $\text{SP}^X \vdash w \leq a$  by induction, and we conclude by transitivity with  $\text{SP}^X \vdash u \leq v$  or  $\text{SP}^X \vdash u \leq w$ ;
- if  $u = v \cdot w$ , then necessarily  $X \in \{1, 1\top\}$ , and there must a path  $\iota \overset{a}{\Rightarrow} o$  in  $\mathcal{G}'(v)$  and a path  $\iota \Rightarrow o$  in  $\mathcal{G}'(w)$  (or vice-versa), so that we deduce  $\text{SP}^X \vdash v \leq a$  and  $\text{SP}^1 \vdash w \leq 1$  by induction and Lemma 38 (or vice-versa), from which we conclude by monotonicity and  $\text{SP}^1 \vdash a \cdot 1 \leq a$  (or  $\text{SP}^1 \vdash 1 \cdot a \leq a$ ). ◀

► **Theorem 40** (Proposition 22 and Theorem 25 in the main text). *For all  $X$  in  $\{1, \top, 1\top\}$ , for all  $u, v \in \mathcal{SP}X$ ,  $\mathcal{SP}^X \vdash u \leq v$  iff  $\mathcal{G}'(u) \triangleleft^X \mathcal{G}'(v)$ .*

**Proof.** For the forward direction, we proceed by induction on the derivation: the various axioms of  $\mathcal{SP}X$  all give rise to simple  $X$ -homomorphisms,  $X$ -homomorphisms properly compose, and they are preserved under substitutions and contexts. (To show preservation under substitutions, we need to exploit the fact that the manipulated graphs are strict and go forward in the  $\mathcal{SPT}$  case.)

For the converse implication, we proceed by structural induction on  $v$ .

- If  $v$  is 1 or a letter, we use Lemma 38 or Lemma 39.
- If  $v = \top$  then  $X = \top$  or  $X = 1\top$ , and  $\mathcal{SP}^X \vdash u \leq \top = v$ .
- If  $v = v_1 \cap v_2$ , then  $\mathcal{G}(u) \triangleleft^X \mathcal{G}(v)$  yields both  $\mathcal{G}(u) \triangleleft^X \mathcal{G}(v_1)$  and  $\mathcal{G}(u) \triangleleft^X \mathcal{G}(v_2)$ . We deduce  $\mathcal{SP}^X \vdash u \leq v_1$  and  $\mathcal{SP}^X \vdash u \leq v_2$  by induction, and thus  $\mathcal{SP}^X \vdash u \leq u \cap u \leq v_1 \cap v_2 = v$ .
- If  $v = v_1 \cdot v_2$ , then Lemma 37 gives  $u_1, u_2$  such that  $\mathcal{G}'(u_1) \triangleleft^X \mathcal{G}'(v_1)$ ,  $\mathcal{G}'(u_2) \triangleleft \mathcal{G}'(v_2)$ , and  $\mathcal{SP} \vdash u \leq u_1 u_2$ . We deduce  $\mathcal{SP}^X \vdash u_1 \leq v_1$  and  $\mathcal{SP}^X \vdash u_2 \leq v_2$  by induction and we conclude by monotonicity. ◀