Completeness Theorems for Kleene Algebra with Top

Damien Pous
Plume, LIP, CNRS, ENS de Lyon, France

Jana Wagemaker
University of Nijmegen, The Netherlands

Abstract

We prove two completeness results for Kleene algebra with a top element, with respect to languages and binary relations. While the equational theories of those two classes of models coincide over the signature of Kleene algebra, this is no longer the case when we consider an additional constant "top" for the full element. Indeed, the full relation satisfies more laws than the full language, and we show that those additional laws can all be derived from a single additional axiom. We recover that the two equational theories coincide if we slightly generalise the notion of relational model, allowing sub-algebras of relations where top is a greatest element but not necessarily the full relation.

We use models of closed languages and reductions in order to prove our completeness results, which are relative to any axiomatisation of the algebra of regular events.

2012 ACM Subject Classification Theory of computation → Equational logic and rewriting; Theory of computation → Logic and verification; Theory of computation → Regular languages

Keywords and phrases Kleene algebra, Hypotheses, Completeness, Closed languages

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2022.26

Funding This work was supported by the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX- 0007) operated by the French National Research Agency (ANR).

Acknowledgements The authors would like to thank Paul Brunet, Amina Doumane, and Jurriaan Rot for the discussions that eventually led to this work, and the CONCUR reviewers for all their comments.

1 Introduction

The axiomatic treatment of regular expressions and languages was developed extensively by Conway [9], after earlier work of Kleene [16]. He raised there a difficult question: how to axiomatise the equations between regular expressions that hold under their standard interpretation as formal languages? Redko had proved that every purely equational axiomatisation must be infinite [32]. Conway proposed such an infinite axiomatisation, which Krob proved to be complete twenty years later [24]. Conway had also proposed finite quasi-equational axiomatisations, one of which Kozen proved to be complete the same year [21] – this axiomatisation is now commonly called *Kleene algebra*. By an additional remark of Boffa [5], this latter completeness result can also be obtained as a consequence of Krob's completeness result. In the end, all finite quasi-equational axiomatisations proposed by Conway, as well as a few other ones, are actually complete [24, 6].

In symbols, writing $[e]$ for the language of a regular expression e and $\mathsf{KA} \vdash e = f$ when the equation $e = f$ is derivable in any of the aforementioned axiomatisations, we have that for all regular expressions e, f,

$$\mathsf{KA} \vdash e = f \iff [e] = [f]$$

The above equivalence extends with two more clauses. When an equation is derivable, it must hold in all models of the chosen axiomatisation. These include in particular language models (LANG) and relational models (REL), for which we actually have an equivalence: writing $\mathcal{X} \models e = f$ when the equation $e = f$ holds in all members of a class of models $\mathcal{X}$, we actually have:

$$\mathsf{KA} \vdash e = f \iff \mathsf{REL} \models e = f \iff \mathsf{LANG} \models e = f \iff [e] = [f]$$

Completeness w.r.t. LANG is immediate given the previous equivalence: the language interpretation of a regular expression lies in LANG. This is less obvious for REL: completeness comes from a nice trick due to Pratt showing that every member of LANG embeds into a member of REL [31, third page].

As an immediate consequence of the above equivalence, the equational theory of REL (or LANG) is decidable – more precisely, PSPACE-complete. This has important applications in program verification: Kleene algebras and their extension to Kleene algebras with tests [17] make it possible to represent and reason about the big-step semantics of while programs, algebraically. This was used for instance to analyse compiler optimisations [19]. The decidability result was also implemented in proof assistants such as Coq and Isabelle/HOL, in order to automate some reasoning steps about binary relations and Hoare logic on while programs [28, 23].

The above-mentioned results apply to the *regular* operations and constants: composition, union, Kleene star, identity, emptiness. A natural question is whether they extend to other operations or constants, such as intersection, converse, fullness. The case of converse was dealt with by Ésik et al.: the equational theories of REL and LANG differ in the presence of converse but both can be axiomatised [4, 13], and they remain PSPACE-complete [8]. The case of intersection (with or without converse or the various constants) is significantly more difficult, and remains partly open, see [2, 7, 26, 12]. In this paper we focus on the addition of a constant $\top$, interpreted as the full language in LANG and as the full relation in REL.

The usefulness of adding such a constant was demonstrated recently in the context of Kleene algebras with tests (KAT), to model *incorrectness logic* [33]. Indeed, while KAT alone makes it possible to model Hoare triples for partial correctness [18], the addition of a full element makes it possible to compare the (co)domains of relations, and thus to encode *incorrectness triples* [27, Section 5.3]. KAT with a top element was also used earlier, as an intermediate structure to characterise a semantics for abnormal termination [25, Definition 12].

As expected, one should consider an axiom expressing that $\top$ is a greatest element:

$$x \leq \top \tag{T}$$

(Where $x \leq y$ is a shorthand for $x + y = y$.) Together with the Kleene algebra axioms, axiom (T) yields a complete axiomatisation w.r.t. language models: we sketched a proof in [30, Example 3.4], which we make fully explicit here in Section 3 (Theorem 3.5). This proof gives us as a byproduct that the equational theory of Kleene algebras with a greatest element remains PSPACE-complete.

Unfortunately, the previous axiom is not enough to deal with relational models. In fact, in the presence of $\top$, the equational theories of LANG and REL differ. Indeed, there are laws such as $\top x \top y \top = \top y \top x \top$ [29, page 13], or $\top x \top x = \top x$ [33, page 14], which are valid in REL, but not in LANG.

In the present paper, we show that it suffices to further add the following axiom in order to obtain a complete axiomatisation for REL (Theorem 4.16):

$$x \leq x \cdot \top \cdot x \tag{F}$$

This inequation is mentioned in [33, page 14]; it holds in relational models, but not in language ones. Thanks to (T), axiom (F) may be seen as a consequence of Ésik et al.'s axiom $x \leq x \cdot x^{\circ} \cdot x$ for dealing with converse $(\cdot^{\circ})$ in relational models [13, 4]. How to use axiom (F) in an equational proof is not so intuitive: it does not give rise to a natural notion of normal form, and it must often be used in conjunction with (T) in order to compensate the fact that it duplicates subterms. For instance here is how we can prove the first of the aforementioned laws:

$$\begin{aligned}
\top x \top y \top &\leq \top x \top y \top \ \top \ \top x \underline{\top y \top} && \text{(by (F))}\\
&\leq \top x \top y \underline{\top \top \top} x \top && \text{(by (T))}\\
&\leq \underline{\top x \top} y \top x \top && \text{(by (T))}\\
&\leq \top y \top x \top && \text{(by (T))}
\end{aligned}$$

(We wrote compositions by juxtaposition, skipped the associativity steps, and underlined the subterms to be simplified by axiom (T) – the converse inequation is derived symmetrically.) Our completeness proof actually goes via a factorisation property (Proposition 4.7) intuitively asserting that one can always proceed in this way to reason about star-free expressions: expand the expressions using (F) a number of times, then remove spurious subterms using (T). Combining such a technique together with Kleene algebra reasoning for star is the second challenge we address in the present work. To get a grasp on the difficulties, the reader may try to find a proof of the following valid law of REL, using KA and axiom (F):

$$a^{+} \leq o(a + \top o)\top a^{+} \tag{$\star$}$$

There, o and a^{+} are shorthands for $(aa)^{*}a$ and $a^{*}a$; we give a solution in Examples 4.13-4.15.

Finally, we show that the difference between the equational theories of language and relational models can be blurred if we slightly generalise the notion of relational model, allowing $\top$ to be any greatest relation rather than the full one[1] (Corollary 5.2).

We prove our two main theorems using the concept of *closed language model* for Kleene algebra with hypotheses [11], and the reduction technique made explicit in [30, 15][2]. Intuitively, we establish reductions from KA with (T) and KA with (T, F) to plain KA, so that we can deduce completeness and decidability of the former theories from completeness and decidability of the latter one.

While the first reduction is relatively straightforward – this is a syntactical linear reduction, the second one is not. We exploit the aforementioned factorisation result (Proposition 4.7) and Kleene's theorem in order to show that regular languages are preserved by a certain closure operation, and that this preservation property can be justified algebraically (Proposition 4.14). Moreover, in order to establish the correspondence between the closed languages used there and relational models, we resort to a graph theoretical characterisation of the equational theory of REL [7, Theorem 6] (whose main ingredient dates back to the works of Freyd and Scedrov [14, page 208] and Andréka and Bredikhin [3, Theorem 1]).

[1] considering subalgebras where only certain relations are kept, since otherwise the only greatest relation is the full one.

[2] Such a technique is somehow implicit in Kozen and Smith's completeness proof for KAT [20] and Ésik et al.'s completeness proof for Kleene algebra with converse [4, 13].

Related work

Zhang et al. give a completeness result for KAT together with axiom (T), in terms of guarded string languages [33, Theorem 9]. They observe that this axiomatisation is incomplete for REL, that it does not suffice to properly express *incorrectness triples*, and they leave the existence of a complete axiomatisation for relational models open. Our Theorem 4.16 gives a positive answer to this question, in the more primitive setting of plain Kleene algebra, without tests. We believe that a similar answer holds also for KAT; if this is the case, then we would obtain a system where we can reason purely equationally about incorrectness triples, as envisioned by Zhang et al.

For the weaker theory of KAT with (T), the main completeness results of Zhang et al. [33, Theorems 7 and 9] are wrong: the model of guarded strings they designed equates too many expressions (namely, Σ^* and $\top$ – see Remark 3.6). Our Theorem 3.5 uses a different language model, and we believe our simple (and linear) reduction from KA with (T) to KA is also a reduction from KAT with (T) to KAT, so that, e.g., [33, Theorem 10] about the complexity of KAT with (T) remains true.

Zhang et al. also give a completeness result w.r.t. generalised relational models [33, Theorem 8]. Their proof is problematic because it relies on their Theorem 7, but the key idea remains valid: adapting Pratt's trick to embed language models into relational ones. We use the very same technique to obtain Corollary 5.2.

Outline

We setup and recall basic notation for regular expressions, formal languages and universal algebra in Section 2. Then we deal with language models in Section 3, and relational models in Section 4. While the language case was already sketched in [30, Example 3.4], we find it useful to treat it explicitly here, before dealing with the more involved case of relations: it illustrates the reduction method in a simpler setting, and we build on the reduction for languages to establish the reduction for relations. We finally prove completeness with respect to generalised relational models in Section 5.

2 Preliminaries

Given a set X, we write X^* for the set of *words* over X: finite sequences of elements of X. We let u, v range over words, we write ϵ for the empty word, and uv for the concatenation of two words u, v. A *language* is a set of words. We let e, f range over *regular expressions over* X, generated by the following grammar:

$$e, f ::= e + f \mid e \cdot f \mid e^* \mid 0 \mid 1 \mid x \in X$$

We sometimes omit the dots in regular expressions, writing, e.g., ab^* for $a \cdot b^*$. As usual, we associate a language $[e]$ to every regular expression e, the *language of* e. A language is *regular* if it is the language of a regular expression.

We fix a finite set Σ of *letters*, ranged over using a, b. We write $\Sigma_\top$ for the set Σ extended with a new element $\top$. We call the regular expressions over $\Sigma_\top$ *regular expressions with top* (or often just *expressions*, since we are mostly concerned with these). We shall sometimes see words over $\Sigma_\top$ as regular expressions with top. E.g., the word $a\top$ can be seen as the expression $a \cdot \top$.

We consider signatures $S \triangleq \{+_2, \cdot_2, \cdot^*{}_1, 0_0, 1_0\}$ and $S_\top \triangleq S \cup \{\top_0\}$. Given an S-algebra A and a valuation $\sigma : \Sigma \to A$, we write $\hat{\sigma}$ for the unique homomorphism extending σ to regular expressions over Σ. Similarly, given an $S_\top$-algebra A and a valuation $\sigma : \Sigma \to A$, we write $\hat{\sigma}$ for the unique homomorphism extending σ to regular expressions with top. (Note in that case that the domain of the valuation is only Σ, and that $\hat{\sigma}(\top) = \top_A$ by definition: $\top$ is a constant, not a variable.)

Given a class $\mathcal{X}$ of S-algebras and two regular expressions e, f over Σ, we write $\mathcal{X} \models e = f$ if for all members A of $\mathcal{X}$ and all valuations $\sigma : \Sigma \to A$, we have $\hat{\sigma}(e) = \hat{\sigma}(f)$. We use similar notations for classes of $S_\top$-algebras and regular expressions with top.

An *equation* is a pair of regular expressions e, f, written $e = f$. We write $e \leq f$, an *inequation*, as a shorthand for the equation $e + f = f$. An *axiomatisation* is a set of equations (or implications between equations). Given such a set $\mathcal{E}$, we write $\mathcal{E} \vdash e = f$ when the equation $e = f$ is derivable from $\mathcal{E}$ using the rules of equational reasoning (where letters from Σ appearing in the equations of $\mathcal{E}$ can be substituted by arbitrary terms).

We let KA stand for any axiomatisation over plain regular expressions which is sound and complete w.r.t. the regular language interpretation, i.e., such that for all regular expressions e, f (without top), we have[3]

$$\mathsf{KA} \vdash e = f \iff [e] = [f] \tag{$\dagger$}$$

As explained in the introduction, valid candidates for KA include Conway's infinite but purely equational axiomatisation [9, page 116] (proved complete by Krob [24]), Kozen's Kleene algebras [21], left-handed Kleene algebras [22, 10], and Boffa's algebras [6].

Also note that the above requirement is equivalent to the following one, since $L \subseteq K$ iff $L \cup K = K$ for all languages L, K:

$$\mathsf{KA} \vdash e \leq f \iff [e] \subseteq [f] \tag{$\ddagger$}$$

3 Languages

We let L, K range over languages on some alphabet X, and $\mathcal{P}(X^*)$ denotes the set of all such languages. Languages on X form a $S_\top$-algebra with the operations defined as follows:

$$\begin{aligned}
L + K &\triangleq L \cup K \\
L \cdot K &\triangleq \{uv \mid u \in L \ \wedge\ v \in K\} \\
L^* &\triangleq \{u_0 \dots u_{n-1} \mid \exists n \in \mathbb{N},\ \forall i < n,\ u_i \in L\} \\
0 &\triangleq \emptyset \\
1 &\triangleq \{\epsilon\} \\
\top &\triangleq X^*
\end{aligned}$$

(That is, $+$ is set-theoretic union, $\cdot$ is language concatenation, $\cdot^*$ is Kleene star, 0 and $\top$ are the empty and full languages, respectively, and 1 is the singleton language that contains the empty word.) We write LANG for the class of all $S_\top$-algebras of the above shape.

Let KA_T, *Kleene Algebra with a Top element*, denote the union of the axioms from KA and axiom (T). We prove in this section that KA_T is sound and complete for LANG.

Following the strategy from [11, 30], the first step consists of defining the closure operation below, according to the axiom (T) we add to Kleene algebra:

[3] Actually, we require slightly more if the axiomatisation contains implications: those implications should be valid in the models of languages and binary relations.

▶ **Definition 3.1** (Language closure C_T). *Given two words u, v over $\Sigma_\top$, we write $u \leftsquigarrow_T v$ if u is obtained from v by replacing an occurrence of $\top$ with an arbitrary word $w \in {\Sigma_\top}^*$. Given a language L over $\Sigma_\top$, we call T-*closure of L *the following language*

$$C_T(L) \triangleq \{u \mid u \leftsquigarrow_T^* v \text{ for some } v \in L\}$$

C_T is indeed a closure operator, and $C_T(L)$ may alternatively be described as the set of words obtained by replacing occurrences of $\top$ in a word of L with arbitrary words over $\Sigma_\top$.

▶ **Lemma 3.2.** *C_T is an $S_\top$-algebra homomorphism.*

Proof. By a routine verification; the case for composition follows from the fact that we replace single letters. ◀

▶ **Definition 3.3** (Expression closure r). *Let r be the unique S-algebra homomorphism on expressions with top such that $r(a) = a$ for all letters $a \in \Sigma$, and $r(\top) = \mathbf{\Sigma}_\top^*$ (where $\mathbf{\Sigma}_\top^*$ is a regular expression with top for the full language – e.g., $(a + b + \cdots + \top)^*$).*

▶ **Proposition 3.4.** *For all expressions e, we have*

(i) $[r(e)] = C_T[e]$, *and*
(ii) $\mathsf{KA}_T \vdash e = r(e)$.

Proof.

(i) $[r(\cdot)]$ and $C_T[\cdot]$ are S-algebra homomorphisms, and they agree on $\Sigma_\top$.
(ii) We proceed by induction on e; the only interesting case is when $e = \top$, for which we have $\mathsf{KA}_T \vdash r(\top) \le \top$ by axiom (T), and $\mathsf{KA}_T \vdash \top \le r(\top)$ by completeness of KA (‡), since $[\top] = \{\top\} \subseteq {\Sigma_\top}^* = [r(\top)]$. ◀

▶ **Theorem 3.5.** *For all regular expressions with top e, f, we have*

$$\mathsf{LANG} \models e = f \iff C_T[e] = C_T[f] \iff \mathsf{KA}_T \vdash e = f$$

Proof. We have

$$\begin{aligned}
&\mathsf{LANG} \models e = f \\
\Rightarrow\ & C_T[e] = C_T[f] && (C_T[\cdot] \text{ is an interpetation into a member of } \mathsf{LANG}\text{, by Lemma 3.2}) \\
\Leftrightarrow\ & [r(e)] = [r(f)] && (\text{Proposition 3.4(i)}) \\
\Leftrightarrow\ & \mathsf{KA} \vdash r(e) = r(f) && (\text{completeness of } \mathsf{KA}\ (\dagger)) \\
\Rightarrow\ & \mathsf{KA}_T \vdash e = f && (\text{transitivity and Proposition 3.4(ii)}) \\
\Rightarrow\ & \mathsf{LANG} \models e = f && (\text{soundness of } \mathsf{KA}_T \text{ axioms w.r.t. } \mathsf{LANG})
\end{aligned}$$

(In the last step, soundness w.r.t. LANG comes from our assumption about KA, and a trivial verification for axiom (T).) ◀

Note that the first equivalence in the above theorem can be obtained in a more direct way, without resorting to completeness of some axiomatisation; moreover the right-to-left implication of the second equivalence is an instance of a general property of closed language models [11, Theorem 2]. The reduction r is used only for the left-to-right implication of this second equivalence.

According to the above proof, we could complete the statement with "… $\iff [r(e)] = [r(f)]$". Doing so gives us a PSPACE algorithm: compute the regular expressions $r(e)$ and $r(f)$, and compare them for language equivalence.

▶ Remark 3.6. Note that it is crucial that $r(\top)$ be defined as a regular expression $\mathbf{\Sigma}_\top^*$ for the full language on $\Sigma_\top$ rather than an expression $\mathbf{\Sigma}^*$ for the full language on just Σ: otherwise we would equate $\mathbf{\Sigma}^*$ and $\top$, while those are different in LANG (e.g., for a counterexample when $\Sigma = \{a, b\}$, interpret both a and b as the empty language on some non-empty alphabet).

4 Relations

Given a set X, a *relation on* X is a set of pairs of elements from X. We let R, S range over such relations, whose set is written $\mathcal{P}(X \times X)$, and we write $x\,R\,y$ for $\langle x, y\rangle \in R$. Relations on X form an $S_\top$-algebra with the operations defined as follows:

$$\begin{aligned} R + S &\triangleq R \cup S \\ R \cdot S &\triangleq \{\langle x, z\rangle \mid \exists y \in X,\ x\,R\,y \ \wedge\ y\,S\,z\} \\ R^* &\triangleq \{\langle x_0, x_n\rangle \mid \exists n \in \mathbb{N}, x_1, \ldots, x_{n-1},\ \forall i < n,\ x_i\,R\,x_{i+1}\} \\ 0 &\triangleq \emptyset \\ 1 &\triangleq \{\langle x, x\rangle \mid x \in X\} \\ \top &\triangleq X \times X \end{aligned}$$

($+$ is set-theoretic union, $\cdot$ is relational composition, $\cdot^*$ is reflexive transitive closure, 0, 1 and $\top$ are the empty, identity and full relations, respectively.) We write REL for the class of all $S_\top$-algebras of the above shape.

Let KA_F, *Kleene Algebra with a Full element*, denote the union of the axioms from KA_T and axiom (F). Let us emphasise that despite the abbreviation, KA_F extends KA_T and thus contains axiom (T). We prove in this section that KA_F is sound and complete for REL. The proof consists of two parts. First we characterise the equational theory of REL in terms of closed languages (Section 4.1, Proposition 4.8), then we use reductions to show completeness of KA_F w.r.t. this closed language interpretation and obtain our main result (Section 4.2, Theorem 4.16).

4.1 Characterisation via closed languages

We start by extending the previous closure function (Definition 3.1), in order to take into account the new axiom (F):

▶ **Definition 4.1** (Language closure C_F). *Given two words u, v over $\Sigma_\top$, we write $u \leftsquigarrow_F v$ if either $u \leftsquigarrow_T v$, or u is obtained by replacing a subword of the shape $w\top w$ in v, with w (for some word $w \in {\Sigma_\top}^*$). Given a language L over $\Sigma_\top$, we call F-*closure of L *the language*

$$C_F(L) \triangleq \{u \mid u \leftsquigarrow_F^* v \text{ for some } v \in L\}$$

C_F is a closure operator, but unlike C_T in the previous section, C_F is not a homomorphism – e.g., $C_F(\{a\} \cdot \{\top a\})$ contains the word a while $C_F(\{a\}) \cdot C_F(\{\top a\})$ does not. Moreover, an elementary description of C_F requires more work than for C_T in the previous section.

Let E be the following function on languages over $\Sigma_\top$, where for a word u and a natural number n, we write u^n for the word obtained by concatenating n copies of u:

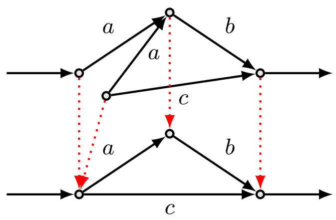
$$E(L) \triangleq \{w \mid \exists n,\ w(\top w)^n \in L\}$$

We shall prove that $C_F = E \circ C_T$, and that C_F can be characterised in terms of certain graph homomorphisms (Proposition 4.7 below). Before doing so, we need to define the type of graphs we will use in what follows.

▶ **Definition 4.2** (Graph, graph homomorphism). *A* graph *is a tuple $\langle V, E, \iota, o\rangle$, where V is a set of* vertices*, $E \subseteq V \times \Sigma \times V$ is a set of labelled edges, and $\iota, o \in V$ are two distinguished vertices, respectively called* input *and* output*.*

A graph homomorphism *from the graph G to the graph H is a function from vertices of G to vertices of H that preserves labelled edges, input, and output. We write $H \lhd G$ when there exists a homomorphism from G to H.*

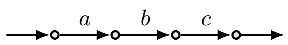
The relation $\lhd$ on graphs is a preorder. We depict graphs as usual, using an unlabelled ingoing (resp. outgoing) arrow to indicate the input (resp. output); we use dotted red arrows to depict graph homomorphisms. For instance, we depict two finite connected graphs below, and a homomorphism between them:

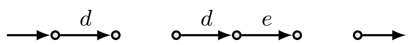


▶ **Definition 4.3** (Graph of a word). *We associate to each word $u \in {\Sigma_\top}^*$ the graph $g(u)$ defined as follows:*

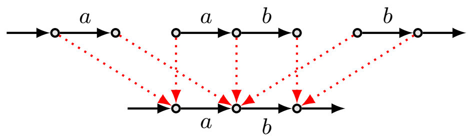
- *the vertices are the natural numbers smaller or equal to the length n of u;*
- *for $a \in \Sigma$ there is an a-labelled edge from i to $i + 1$ if the i-th letter of u is a;*
- *the input is 0 and the output is n.*

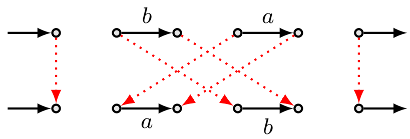
Graphs of words are rather simple: graphs as depicted above do not arise as graphs of words. For words not containing $\top$, they are just directed paths from the input to the output. For words containing $\top$, they are collections of (possibly empty) directed paths where the input is the starting-point of some path and the output is the end-point of some path. For example, the graphs of abc and $d\top de\top$ are depicted below:



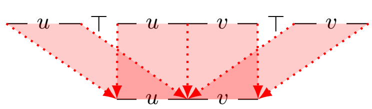


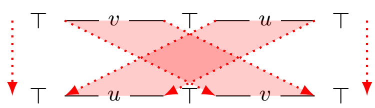
Nevertheless, homomorphisms between graphs of words may be non-trivial. For instance, we have $g(ab) \lhd g(a\top ab\top b)$ and $g(\top a\top b\top) \lhd g(\top b\top a\top)$, as witnessed below:





In the sequel, we shall represent homomorphisms between graphs of words in a slightly more compact way, starting directly from the natural writing of the words, and using horizontal lines and shaded parallelograms to emphasise distinguished subwords and mappings between them. For instance, the above homomorphisms can be generalised to $g(uv) \lhd g(u\top uv\top v)$ and $g(\top u\top v\top) \lhd g(\top v\top u\top)$ for arbitrary words u, v, which we can represent as follows:





Our main interest in graphs and homomorphisms comes from the following characterisation of the equational theory of REL. This characterisation appeared first in [7, Theorem 6], for the syntax of Kleene allegories. Its (trivial) extension to Kleene allegories with top then appeared in [29, Theorem 16].

▶ **Theorem 4.4** ([7, Theorem 6]). *For all regular expressions with top e, f, we have:*

$$\mathsf{REL} \models e \leq f \iff \forall u \in [e],\ \exists v \in [f],\ g(u) \lhd g(v)$$

Proof. Cf. the above references. That we need the theorem only in a small fragment here (without intersection and converse) does not seem to enable substantial simplifications. In particular, we still need to consider arbitrary graphs, and a variant of [3, Lemma 3] with top. We give a proof in Appendix A for the sake of completeness. ◀

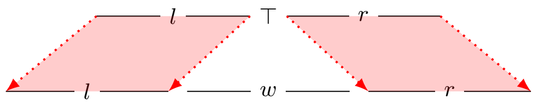
▶ Remark 4.5. For words u, v without top, we have $g(u) \lhd g(v)$ iff $u = v$. Therefore, for regular expressions e, f without top (whose languages only contain words without top), the above theorem reduces to $\mathsf{REL} \models e \leq f \iff [e] \subseteq [f]$, a standard variant of one of the equivalences recalled in the introduction.

Thanks to Theorem 4.4, it suffices to relate homomorphisms between graphs of words to the notion of C_F-closure. We do so in the following lemma.

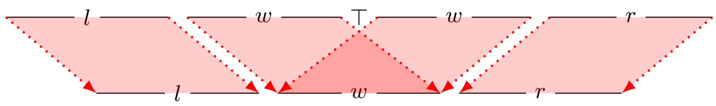
▶ **Lemma 4.6.** *For all words $u, v \in {\Sigma_\top}^*$, the following are equivalent:*
(i) $u \leftsquigarrow_F^* v$,
(ii) $g(u) \lhd g(v)$,
(iii) $u \in E(C_T\,\{v\})$.

Proof. We show (i) ⇒ (ii) ⇒ (iii) ⇒ (i). For the first implication, since $\lhd$ is a preorder, it suffices to show that $u \leftsquigarrow_F v$ entails $g(u) \lhd g(v)$. There are two cases to consider.

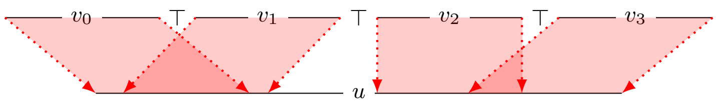
- either the rewriting rule associated to axiom (T) was used, i.e., $u = lwr$ and $v = l\top r$ for some words $l, w, r \in {\Sigma_\top}^*$. In that case we have the following homomorphism from the graph of v to the graph of u:



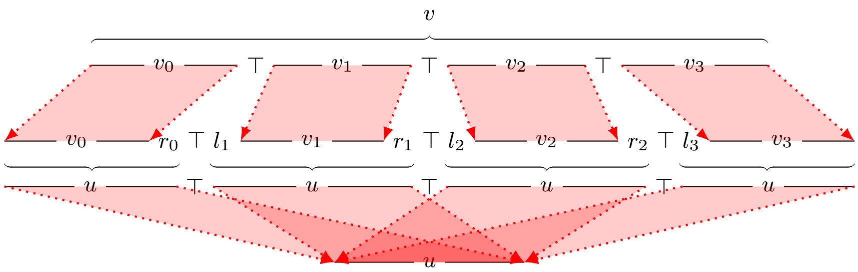
- or the rewriting rule associated to axiom (F) was used, i.e., $u = lwr$ and $v = lw\top wr$ for some words $l, w, r \in {\Sigma_\top}^*$. In that case we have the following homomorphism from the graph of v to the graph of u:



For the second implication, assume $g(u) \lhd g(v)$. Let n be the number of occurrences of $\top$ in v, and let $v_0, \ldots, v_n$ be top-free words such that $v = v_0 \top v_1 \top \cdots \top v_n$. Since they are top-free, those subwords must be mapped linearly to u, and thus be subwords of u. For instance, when $n = 3$, the homomorphism may look as follows:



For all $0 \leq i \leq n$, let l_i, r_i be the words such that $u = l_i v_i r_i$. We have that l_0 and r_n must be the empty word since inputs and outputs must be preserved by homomorphisms. We have $u(\top u)^n \leftsquigarrow^n_T v$: we can obtain $u(\top u)^n$ from v by replacing the ith occurrence of $\top$ in v with the word $r_{i-1}\top l_i$, for $0 < i \leq n$. This suffices to conclude that $u \in E(C_T\{v\})$: we have proven (ii) $\Rightarrow$ (iii). As an example, when $n = 3$, the situation may be depicted as follows:



For the last implication, assume that $u \in E(C_T\{v\})$. There exists n such that $u(\top u)^n \leftsquigarrow^*_T v$, an thus in particular $u(\top u)^n \leftsquigarrow^*_F v$. Finally observe that $u \leftsquigarrow^*_F u(\top u)^n$ using n rewriting steps using (F), so that we can conclude by transitivity: $u \leftsquigarrow^*_F u(\top u)^n \leftsquigarrow^*_F v$. ◀

The above lemma has two important immediate consequences. First we have the announced factorisation of the closure C_F, and second, combined with Theorem 4.4, we obtain a characterisation of the equational theory of REL in terms of closed languages:

▶ **Proposition 4.7.** *We have $C_F = E \circ C_T$.*

▶ **Proposition 4.8.** *For all regular expressions with top e, f, we have:*

$$\mathsf{REL} \models e = f \iff C_F[e] = C_F[f]$$

Proof. For all e, f, we have:

$$\begin{aligned} \mathsf{REL} \models e \leq f &\iff \forall u \in [e],\ \exists v \in [f],\ g(u) \lhd g(v) && \text{(by Theorem 4.4)} \\ &\iff [e] \subseteq C_F[f] && \text{(by Lemma 4.6)} \end{aligned}$$

The initial statement follows by antisymmetry and the fact that C_F is a closure (so that for all languages L, K, $L \subseteq C_F(K)$ iff $C_F(L) \subseteq C_F(K)$). ◀

4.2 Completeness w.r.t. closed languages

It remains to show that KA_F is complete w.r.t. the previous closed language interpretation. We use reductions in order to do so: we find a counterpart to the function r from Section 3 (Definition 3.3), for the F-closure rather than the T-closure. By Proposition 4.7, and since we already have the function r for T-closure, it actually suffices to find a function s that corresponds to the function E, i.e., such that for all expressions e, $s(e)$ is an expression whose language is $E[e]$.

To this end, we use Kleene's theorem stating that a language is regular if and only if it is recognisable by a finite automaton, and the fact that regular languages are closed under union and intersections. Using those tools, we show that the language $E(L)$ is regular whenever L is a regular language, by forming unions and intersections of regular languages extracted from some finite automaton for L.

We first recall standard notions from finite automata theory.

▶ **Definition 4.9** (Non-deterministic finite automaton). *Let X be a finite set. A* non-deterministic finite automaton (NFA) *over the alphabet X is a tuple $\mathcal{A} = \langle Q, i, \Delta, F\rangle$ where:*

- *Q is a finite set of* states*;*
- *$i \in Q$ is an* initial state*;*
- *$\Delta : X \to \mathcal{P}(Q\times Q)$ is the* transition relation*, associating to each letter of X a relation on states;*
- *$F \subseteq Q$ is a subset of* accepting states.

We extend the transition relation Δ into a function Δ' on words as follows (where as before, 1 is the identity relation on Q and $\cdot$ is relation composition):

$$\begin{cases} \Delta'(\epsilon) \triangleq 1 \\ \Delta'(xu) \triangleq \Delta(x)\cdot\Delta'(u) & \textit{for } x \in X \textit{ and } u \in X^* \end{cases}$$

The language of $\mathcal{A}$ from states p to q, *written $\mathcal{L}_{\mathcal{A}}(p,q)$ or just $\mathcal{L}(p,q)$ when $\mathcal{A}$ is clear from the context, is defined as follows:*

$$\mathcal{L}_{\mathcal{A}}(p,q) \triangleq \{u \in X^* \mid \langle p,q\rangle \in \Delta'(u)\}$$

The language of $\mathcal{A}$, *written $\mathcal{L}_{\mathcal{A}}$ is finally obtained as $\bigcup_{f\in F}\mathcal{L}_{\mathcal{A}}(i,f)$.*

Intuitively, the language from p to q consists of those words that label a path from p to q in the automaton, and the language of the automaton consists of those words labelling a path from the initial state to some accepting state.

We will also need a function which is intuitive in the end, but cumbersome to define. Let us use the standard notations for lists: $[]$ for the empty list, $x :: q$ for the insertion of an element x in front of a list q, and $[x; y; \dots; z]$ for concrete lists. Given a set Q, two elements $p, q \in Q$, and a list $l \in (Q\times Q)^*$ of pairs elements of Q, we write $\mathrm{pr}(p,l,q)$ for the list of pairs of elements of Q defined as follows, by recursion on l:

$$\begin{cases} \mathrm{pr}(p,[],q) \triangleq \{\langle p,q\rangle\} \\ \mathrm{pr}(p,\langle r,s\rangle :: k, q) \triangleq \langle p,r\rangle :: \mathrm{pr}(s,k,q) & \text{for } r,s \in Q, \text{ and } k \in (Q\times Q)^* \end{cases}$$

Intuitively, $\mathrm{pr}(p,l,q)$ shifts the pairs found in l, integrating p at the beginning and q at the end. For instance, we have $\mathrm{pr}(p,[\langle q,r\rangle;\langle s,t\rangle],u) = [\langle p,q\rangle;\langle r,s\rangle;\langle t,u\rangle]$.

▶ **Example 4.10.** The function pr is useful for the following reason. Consider an automaton with initial state i, a single final state f, $\Delta(a) = \{\langle r,s\rangle\}$, and $\Delta(b) = \{\langle t,u\rangle\}$. Suppose we want to characterise the set of words w such that $wawbw$ is accepted. Those are precisely those words in the intersection $\mathcal{L}(i,r)\cap\mathcal{L}(s,t)\cap\mathcal{L}(u,f)$. The terms from this intersection are easily described using pr: we have $\mathrm{pr}(i,[\langle r,s\rangle;\langle t,u\rangle],f) = [\langle i,r\rangle;\langle s,t\rangle;\langle u,f\rangle]$. ◀

We finally write $X^{\circledast}$ for the set of duplicate-free finite sequences over X (i.e., such that every element of X appears at most once). When X is finite, so is $X^{\circledast}$.

We now have all that we need to characterise the image of E on regular languages:

▶ **Proposition 4.11.** *Let $\mathcal{A} = \langle Q, i, \Delta, F\rangle$ be a NFA over $\Sigma_\top$ with language $\mathcal{L}$. We have*

$$E(\mathcal{L}) = \bigcup\left\{\ \bigcap\{\mathcal{L}(p,q) \mid \langle p,q\rangle \in \mathrm{pr}(i,l,f)\}\ \middle|\ l \in \Delta(\top)^{\circledast},\ f \in F\right\}$$

Proof. We prove the two inclusions separately.

To prove the inclusion from left to right, assume $w \in E(\mathcal{L})$. Let m be the length of w, and let n be the least natural number such that $w(\top w)^n \in \mathcal{L}$. By definition, there is some $f \in F$ and a path from i to f labelled with $w(\top w)^n$ in $\mathcal{A}$. Call a $\top$-*transition* a pair $\langle p, q\rangle$ belonging to $\Delta(\top)$. Let l be the sequence of $\top$-transitions used in this path at positions $m+1$, $2m+1$, $\ldots$, $nm+1$. This sequence is duplicate-free by minimality of n: if the same $\top$-transition was appearing twice, we would find a smaller witness for the membership of w in $E(\mathcal{L})$. We check easily that for all pairs $\langle p, q\rangle \in \mathrm{pr}(i, l, f)$, we have $w \in \mathcal{L}(p, q)$. Therefore, w belongs to the right-hand side expression.

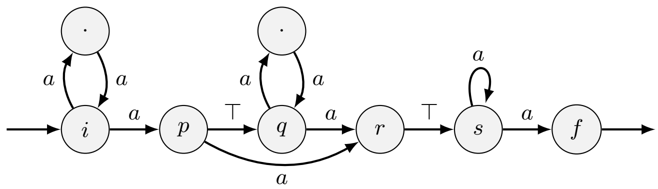
To prove the inclusion from right to left, let w, l, f such that for all $\langle p, q\rangle \in \mathrm{pr}(i, l, f)$, we have $w \in \mathcal{L}(p, q)$. Let n be the length of l. We can construct a path from i to f labelled by $w(\top w)^n$ in $\mathcal{A}$. Therefore we have $w(\top w)^n \in \mathcal{L}$, whence $w \in E(\mathcal{L})$. ◀

The above formula expresses $E(\mathcal{L})$ as a finite union of finite intersections of languages of the form $\mathcal{L}(p, q)$, which are all regular by Kleene's theorem. Since regular languages are closed under unions and intersections, we deduce that $E(\mathcal{L})$ is regular. In other words, the function E preserves regularity of languages over $\Sigma_\top$.

▶ **Definition 4.12** (Expression closure s). *Given a regular expression with top e, we define the regular expression with top $s(e)$ as follows:*

1. *construct a NFA $\langle Q, i, \Delta, F\rangle$ whose language is $[e]$;*
2. *for all $l \in \Delta(\top)^{\circledast}$ and all $f \in F$, compute a regular expression with top $g_{l,f}$ for the regular language $\bigcap\{\mathcal{L}(p, q) \mid \langle p, q\rangle \in \mathrm{pr}(i, l, f)\}$;*
3. *set $s(e) \triangleq \sum_{l,f} g_{l,f}$.*

▶ **Example 4.13.** Call e the expression $o(a + \top o)\top a^+$ from the introduction ($\star$), where $o \triangleq (aa)^*a$ is an expression for the set of words of as of odd length, and $a^+ \triangleq a^*a$ is an expression for the set of non-empty words of as. Let us compute $s(e)$ using the following automaton for $[e]$, where i is the initial state, and f is the only final state.



We can easily describe various languages of interest in this automaton:

x	y	$\mathcal{L}(x, y)$
i	p	$[o]$
q	r	$[o]$
s	f	$[a^+]$

x	y	$\mathcal{L}(x, y)$
i	r	$[o(a + \top o)]$
q	f	$[o\top a^+]$
s	p	$\emptyset$

There are exactly two $\top$-transitions, so that we have five lists in $\Delta(\top)^{\circledast}$ to consider for $s(e)$:

1. the empty list, contributing $\mathcal{L}(i, f) = [e]$ to $E[e]$;
2. $[\langle p, q\rangle]$, not contributing since $\mathcal{L}(i, p) \cap \mathcal{L}(q, f) = [o] \cap [o\top a^+] = \emptyset$;
3. $[\langle r, s\rangle]$, contributing $\mathcal{L}(i, r) \cap \mathcal{L}(s, f) = [o(a + \top o)] \cap [a^+] = [oa]$;
4. $[\langle p, q\rangle; \langle r, s\rangle]$, contributing $\mathcal{L}(i, p) \cap \mathcal{L}(q, r) \cap \mathcal{L}(s, f) = [o] \cap [o] \cap [a^+] = [o]$;
5. $[\langle r, s\rangle; \langle p, q\rangle]$, not contributing since $\mathcal{L}(i, r) \cap \mathcal{L}(s, p) \cap \mathcal{L}(q, f) = \emptyset$;

So in the end, $s(e)$ can simply be taken to be $e + oa + o$. ◀

▶ **Proposition 4.14.** *For all expressions e, we have*
(i) $[s(e)] = E[e]$, *and*
(ii) $\mathsf{KA}_F \vdash e = s(e)$.

Proof. The first item holds by definition of s and Proposition 4.11. We prove two inequations for the second item. Taking $n = 0$ in the definition of E, we have $[e] \subseteq E[e] = [s(e)]$, so that $\mathsf{KA} \vdash e \leq s(e)$ by completeness of KA (‡). For the converse implication, it suffices to prove that for all $l \in \Delta(\top)^{\circledast}$ and all $f \in F$, the expression $g_{l,f}$ from Definition 4.12 is provably smaller than e in KA_F. Let us abbreviate $g_{l,f}$ as g, let n be the length of l, and let $\langle p_j, q_j\rangle_{j\leq n}$ be the successive elements of $\mathrm{pr}(i, l, f)$ (so that $p_0 = i$ and $q_n = f$). We have

$$
\begin{aligned}
& [g(\top g)^n] \\
&= [g] \cdot \{\top\} \cdot [g] \cdot \ \dots \ \cdot \{\top\} \cdot [g] && ([\cdot] \text{ is a homomorphism}) \\
&\subseteq \mathcal{L}(p_0, q_0) \cdot \{\top\} \cdot \mathcal{L}(p_1, q_1) \cdot \ \dots \ \cdot \{\top\} \cdot \mathcal{L}(p_n, q_n) \\
& && (\text{by definition of } g, [g] \text{ is contained in each } \mathcal{L}(p_j, q_j)) \\
&\subseteq \mathcal{L}(p_0, q_0) \cdot \mathcal{L}(q_0, p_1) \cdot \mathcal{L}(p_1, q_1) \cdot \ \dots \ \cdot \mathcal{L}(q_{n-1}, p_n) \cdot \mathcal{L}(p_n, q_n) \\
& && (\text{since } l \in \Delta(\top)^{\circledast}, \text{ we have } \langle q_j, p_{j+1}\rangle \in \Delta(\top), \text{ and thus } \top \in \mathcal{L}(q_j, p_{j+1})) \\
&\subseteq \mathcal{L}(i, f) \subseteq [e] && (p_0 = i,\ q_n = f, \text{ and definition of } \mathcal{L})
\end{aligned}
$$

We deduce $\mathsf{KA} \vdash g(\top g)^n \leq e$ by completeness of KA (‡), and we conclude by prepending n applications of axiom (F): $\mathsf{KA}_F \vdash g \leq g(\top g)^n \leq e$. ◀

▶ **Example 4.15.** Continuing Example 4.13, we check that both $oa \leq oa\top oa \leq e$ and $o \leq o\top o\top o \leq e$ are derivable in KA_F, in both cases using axiom (F) for the the first inequation (once or twice), and KA completeness for the second one. Also observe that $[a^+] = [oa + o]$, so that $\mathsf{KA} \vdash a^+ = oa + o$ once again by KA completeness. Putting everything together, we obtain a derivation of the following shape for the law ($\star$) from the introduction.

$$\mathsf{KA}_F \vdash a^+ = oa + o \leq oa\top oa + o\top o\top o \leq e$$ ◀

More generally, we can combine all the above results to obtain our main theorem:

▶ **Theorem 4.16.** *For all regular expressions with top e, f, we have*

$$\mathsf{REL} \models e = f \iff C_F[e] = C_F[f] \iff \mathsf{KA}_F \vdash e = f$$

Proof. We have

$$
\begin{aligned}
& \mathsf{REL} \models e = f \\
\Leftrightarrow\ & C_F[e] = C_F[f] && (\text{Proposition 4.8}) \\
\Leftrightarrow\ & E(C_T[e]) = E(C_T[f]) && (\text{by Proposition 4.7}) \\
\Leftrightarrow\ & [s(r(e))] = [s(r(f))] && (\text{by Propositions 3.4(i) and 4.14(i)}) \\
\Leftrightarrow\ & \mathsf{KA} \vdash s(r(e)) = s(r(f)) && (\text{by completeness of } \mathsf{KA}\ (\dagger)) \\
\Rightarrow\ & \mathsf{KA}_F \vdash e = f && (\text{by transitivity and Propositions 3.4(ii) and 4.14(ii)}) \\
\Rightarrow\ & \mathsf{REL} \models e = f && (\text{soundness of } \mathsf{KA}_F \text{ axioms w.r.t. } \mathsf{REL})
\end{aligned}
$$

◀

The above proof follows the same strategy as the one for Theorem 3.5. Like there, the right-to-left implication of the second equivalence is an instance of [11, Theorem 2], and we use reductions only for the left-to-right part of this equivalence.

Like for Theorem 3.5, we could complete the statement of Theorem 4.16 with "… $\iff [s(r(e))] = [s(r(f))]$". This gives decidability since the function s is computable, but this does not give a reasonable algorithm: given an expression e, the size of $s(e)$ (or of an automaton for it) might be very big. We leave open the question of whether there is a better algorithm, hopefully in PSPACE.

5 Relations with a greatest element

A *generalised $S_\top$-algebra of relations* is an S-subalgebra A of an algebra of relations such that A has a greatest element, seen as an $S_\top$-algebra by using this greatest element for the constant $\top$. We write REL' for the class of all generalised $S_\top$-algebras of relations.

Intuitively, REL' consists of models of binary relations where $\top$ is not necessarily the full relation, only a greatest element. As an example, consider relations R over the natural numbers such that $i \le j$ whenever $i\,R\,j$. Those form an S-algebra with greatest element the order relation $\le$ itself, which is not the full relation.

In the literature, REL' is sometimes preferred over REL because it is closed under taking subalgebras and products, and actually forms a quasivariety [1]. (In contrast, it is not clear whether REL is closed under products: the two obvious ways of embedding a pair of relations into a new relation fail to preserve either union or top – REL as defined here is not closed under taking subalgebras either, but defining it in such a way would not change the results from the present paper.)

The equational theory of REL' differs from that of REL. For instance, the previous example of ordered relations shows that $\mathsf{REL}' \not\models x \le x \cdot \top \cdot x$. Indeed, for $x = \{\langle 0, 1\rangle\}$, $x \cdot \top \cdot x$ is empty since $\top$ does not relate 1 to 0.

We show below that the equational theory of REL' actually coincides with that of LANG, and can thus be axiomatised by KA_T.

▶ **Proposition 5.1.** *Every member of* LANG *embeds into a member of* REL'*.*

Proof. We adapt the technique used by Pratt for Kleene algebras (without top) [31, third page] and later reused by Kozen and Smith for Kleene algebras with tests [20, Lemma 5]. For a set X, let $M(X)$ be the set of relations R on X^* such that for all words u, v, u is a prefix of v whenever $u\,R\,v$. The S-operations on relations restrict to $M(X)$, so that $M(X)$ is an S-algebra, and setting $\top \triangleq \{\langle u, uv\rangle \mid u, v \in X^*\}$ turns it into a member of REL'. We embed a member $\mathcal{P}(X^*)$ of LANG into $M(X)$ as follows:

$$\begin{aligned} \iota &: \mathcal{P}(X^*) \to M(X) \\ &L \mapsto \{\langle u, uv\rangle \mid u \in X^*,\ v \in L\} \end{aligned}$$

The function ι is easily shown to be an $S_\top$-algebra homomorphism, and it is injective (since, e.g., $L = \{u \mid \langle \epsilon, u\rangle \in \iota(L)\}$). ◀

Note that it is crucial that we consider REL' rather than REL here: the above construction would not give an $S_\top$-algebra homomorphism if we were not restricting to relations of a certain shape: $\top$ would not be preserved.

▶ **Corollary 5.2.** *For all regular expressions with top, we have*

$$\mathsf{LANG} \models e = f \iff \mathsf{REL}' \models e = f \iff \mathsf{KA}_T \vdash e = f$$

Proof. That $\mathsf{REL}' \models e = f$ entails $\mathsf{LANG} \models e = f$ is a direct consequence of Proposition 5.1. That $\mathsf{KA}_T \vdash e = f$ entails $\mathsf{REL}' \models e = f$ follows from the soundness of KA_T axioms w.r.t. REL'. We conclude by Theorem 3.5. ◀

Similarly to REL', we can define a class LANG' of $S_\top$-algebras which is closed under taking subalgebras and where $\top$ is not necessarily the full language. However, unlike with REL' and REL, the equational theory of LANG' coincides with that of LANG (and REL'). Indeed the axioms of KA_T remain sound for LANG'.

References

1 H. Andréka and S. Mikulás. Axiomatizability of positive algebras of binary relations. *Algebra Universalis*, 66(1):7–34, 2011. `doi:10.1007/s00012-011-0142-3`.

2 H. Andréka, S. Mikulás, and I. Németi. The equational theory of Kleene lattices. *Theoretical Computer Science*, 412(52):7099–7108, 2011. `doi:10.1016/j.tcs.2011.09.024`.

3 H. Andréka and D.A. Bredikhin. The equational theory of union-free algebras of relations. *Algebra Universalis*, 33(4):516–532, 1995. `doi:10.1007/BF01225472`.

4 Stephen L. Bloom, Zoltán Ésik, and Gheorghe Stefanescu. Notes on equational theories of relations. *Algebra Universalis*, 33(1):98–126, 1995. `doi:10.1007/BF01190768`.

5 Maurice Boffa. Une remarque sur les systèmes complets d'identités rationnelles. *Informatique Théorique et Applications*, 24:419–428, 1990. URL: `http://archive.numdam.org/article/ITA_1990__24_4_419_0.pdf`.

6 Maurice Boffa. Une condition impliquant toutes les identités rationnelles. *Informatique Théorique et Applications*, 29(6):515–518, 1995. URL: `http://www.numdam.org/article/ITA_1995__29_6_515_0.pdf`.

7 Paul Brunet and Damien Pous. Petri automata for Kleene allegories. In *LICS*, pages 68–79. ACM, 2015. `doi:10.1109/LICS.2015.17`.

8 Paul Brunet and Damien Pous. Algorithms for Kleene algebra with converse. *Journal of Logical and Algebraic Methods in Programming*, 85(4):574–594, 2016. `doi:10.1016/j.jlamp.2015.07.005`.

9 J.H. Conway. *Regular Algebra and Finite Machines*. Chapman and Hall mathematics series. Chapman and Hall, 1971. URL: `https://books.google.nl/books?id=1KAXc5TpEV8C`.

10 Anupam Das, Amina Doumane, and Damien Pous. Left-handed completeness for Kleene algebra, via cyclic proofs. In *LPAR*, volume 57 of *EPiC Series in Computing*, pages 271–289. Easychair, 2018. `doi:10.29007/hzq3`.

11 Amina Doumane, Denis Kuperberg, Pierre Pradic, and Damien Pous. Kleene algebra with hypotheses. In *FoSSaCS*, volume 11425 of *LNCS*, pages 207–223. Springer, 2019. `doi:10.1007/978-3-030-17127-8_12`.

12 Amina Doumane and Damien 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`.

13 Zoltán Ésik and Laszlo Bernátsky. 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`.

14 P. Freyd and A. Scedrov. *Categories, Allegories*. North Holland, 1990.

15 Tobias Kappé, Paul Brunet, Alexandra Silva, Jana Wagemaker, and Fabio Zanasi. Concurrent Kleene algebra with observations: from hypotheses to completeness. In *FoSSaCS*, volume 12077 of *LNCS*, pages 381–400. Springer, 2020. `doi:10.1007/978-3-030-45231-5_20`.

16 S. C. Kleene. Representation of events in nerve nets and finite automata. In *Automata Studies*, pages 3–41. Princeton University Press, 1956. URL: `http://www.rand.org/pubs/research_memoranda/2008/RM704.pdf`.

17 D. Kozen. Kleene algebra with tests. *ACM Transactions on Programming Languages and Systems*, 19(3):427–443, May 1997. doi:10.1145/256167.256195.

18 D. Kozen. On Hoare logic and Kleene algebra with tests. *ACM Transactions on Computational Logic*, 1(1):60–76, 2000. doi:10.1145/343369.343378.

19 D. Kozen and M.-C. Patron. Certification of compiler optimizations using Kleene algebra with tests. In *CL2000*, volume 1861 of *LNAI*, pages 568–582. Springer, 2000. doi:10.1007/3-540-44957-4_38.

20 D. Kozen and F. Smith. Kleene algebra with tests: Completeness and decidability. In *CSL*, volume 1258 of *LNCS*, pages 244–259. Springer, September 1996. doi:10.1007/3-540-63172-0_43.

21 Dexter Kozen. A completeness theorem for Kleene Algebras and the algebra of regular events. In *LICS*, pages 214–225. IEEE Computer Society, 1991. doi:10.1109/LICS.1991.151646.

22 Dexter Kozen and Alexandra Silva. Left-handed completeness. In *RAMiCS*, volume 7560 of *LNCS*, pages 162–178. Springer, 2012. doi:10.1007/978-3-642-33314-9_11.

23 A. Krauss and T. Nipkow. Proof pearl: Regular expression equivalence and relation algebra. *Journal of Automated Reasoning*, 49(1):95–106, 2012. doi:10.1007/s10817-011-9223-4.

24 D. Krob. Complete systems of B-rational identities. *Theoretical Computer Science*, 89(2):207–343, 1991. doi:10.1016/0304-3975(91)90395-I.

25 Konstantinos Mamouras. Equational theories of abnormal termination based on Kleene algebra. In *FoSSaCS*, volume 10203 of *LNCS*, pages 88–105. Springer, 2017. doi:10.1007/978-3-662-54458-7_6.

26 Yoshiki Nakamura. Partial derivatives on graphs for Kleene allegories. In *LICS*, pages 1–12. IEEE, 2017. doi:10.1109/LICS.2017.8005132.

27 Peter W. O'Hearn. Incorrectness logic. *Proc. ACM Program. Lang.*, 4(POPL):10:1–10:32, 2020. doi:10.1145/3371078.

28 Damien Pous. Kleene Algebra with Tests and Coq tools for while programs. In *ITP*, volume 7998 of *LNCS*, pages 180–196. Springer, 2013. doi:10.1007/978-3-642-39634-2_15.

29 Damien Pous. On the positive calculus of relations with transitive closure. In *STACS*, volume 96 of *LIPIcs*, pages 3:1–3:16. Schloss Dagstuhl, 2018. doi:10.4230/LIPIcs.STACS.2018.3.

30 Damien Pous, Jurriaan Rot, and Jana Wagemaker. On tools for completeness of Kleene algebra with hypotheses. In *RAMiCS*, volume 13027 of *LNCS*, pages 378–395. Springer, 2021. doi:10.1007/978-3-030-88701-8_23.

31 V. R. Pratt. Dynamic algebras and the nature of induction. In *ACM Symposium on Theory of Computing*, STOC '80, pages 22–28, New York, NY, USA, 1980. Association for Computing Machinery. doi:10.1145/800141.804649.

32 V.N. Redko. On the algebra of commutative events. *Ukrainian Math. Zh.*, 16:185–195, 1964.

33 Cheng Zhang, Arthur Azevedo de Amorim, and Marco Gaboardi. On incorrectness logic and Kleene algebra with top and tests. *Proc. ACM Program. Lang.*, 6(POPL):1–30, 2022. doi:10.1145/3498690.

A Proof of Theorem 4.4

We give here a proof of Theorem 4.4. Variants of this theorem appeared for Kleene allegories without top in [7, Theorem 6], and for Kleene allegories with top in [29, Theorem 16]. The latter variant subsumes Theorem 4.4: it deals with a strictly larger fragment of relation algebra. We nevertheless give a proof here for the sake of completeness.

First we observe that valuations into relational models are very close to (potentially infinite) graphs in the sense of Definition 4.2: it suffices to adjoin to them an input and an output.

▶ **Definition A.1** (Graph of a valuation). *Let $\sigma : \Sigma \to \mathcal{P}(X \times X)$ be a valuation of Σ into relations on some set X. For all elements $i, j \in X$, we define the graph $\langle \sigma, i, j\rangle \triangleq \langle X, F, i, j\rangle$ where $F \triangleq \{\langle x, a, y\rangle \mid a \in \Sigma,\ \langle x, y\rangle \in \sigma(a)\}$.*

The first key lemma characterises evaluation of expressions not using $0, +, \cdot^*$ in a relational model, in terms of graph homomorphisms. In our case, expressions not using $0, +, \cdot^*$ can be represented by words with top. Such a lemma appeared first in [3, Lemma 3] for a signature including intersection and converse, but not top. Under its original formulation, its extension to cover top is trivial once we realise that the graph of $\top$ should simply be a graph without edges and exactly two vertices (the input and the output).

▶ **Lemma A.2.** *Let $\sigma : \Sigma \to \mathcal{P}(X \times X)$ be a valuation of Σ into a member of* REL*. For all words $u \in {\Sigma_\top}^*$, we have*

$$\langle i, j\rangle \in \hat{\sigma}(u) \iff \langle \sigma, i, j\rangle \lhd g(u)$$

Proof. By induction on u.

- if u is empty, then both sides reduce to the condition $i = j$;
- if u is a letter a, then both sides reduce to the condition $\langle i, j\rangle \in \sigma(a)$;
- if u is $\top$, then both sides hold independently of i, j;
- if $u = vw$ for two smaller words v, w then we have

$$\begin{aligned}
&\langle i, j\rangle \in \hat{\sigma}(vw) \\
\Leftrightarrow\ & \exists k,\ \langle i, k\rangle \in \hat{\sigma}(v)\ \wedge\ \langle k, j\rangle \in \hat{\sigma}(w) && \text{(by definition)} \\
\Leftrightarrow\ & \exists k,\ \langle \sigma, i, k\rangle \lhd g(v)\ \wedge\ \langle \sigma, k, j\rangle \lhd g(w) && \text{(by induction hypothesis on } v \text{ and } w\text{)} \\
\Leftrightarrow\ & \langle \sigma, i, j\rangle \lhd g(vw)
\end{aligned}$$

(The last equivalence comes from a simple analysis of the homomorphisms whose source is a sequential composition of two graphs – see, e.g., [3, Lemma 2(ii)].) ◀

The second key lemma characterises the evaluation of an arbitrary expression in terms of (the evaluations of) the words in the language of that expression. Variants of such a lemma often appear in the literature for *star-continuous* models, rather than just relational ones (e.g., [20, Lemma 4]).

▶ **Lemma A.3.** *Let $\sigma : \Sigma \to \mathcal{P}(X \times X)$ be a valuation of Σ into a member of* REL*. For all regular expressions with top e, we have*

$$\hat{\sigma}(e) = \bigcup_{u \in [e]} \hat{\sigma}(u)$$

Proof. By an easy induction on e, using distributivity of $\cdot$ over arbitrary unions in REL. ◀

Equipped with those two lemmas, we obtain the announced theorem.

▶ **Theorem A.4.** *For all regular expressions with top e, f, we have:*

$$\mathsf{REL} \models e \leq f \iff \forall u \in [e],\ \exists v \in [f],\ g(u) \lhd g(v)$$

Proof. For the forward implication, assume REL $\models e \leq f$ and let $u \in [e]$. Let n be the length of u and consider relations on $[0; n]$, a member of REL. Define $\sigma : \Sigma \to \mathcal{P}([0; n] \times [0; n])$ by $\langle i, j\rangle \in \sigma(a)$ if the i-th letter of u is a and $j = i+1$. The graph $g(u)$ is nothing but $\langle \sigma, 0, n\rangle$, so that we have $\langle 0, n\rangle \in \hat{\sigma}(u)$ by Lemma A.2, using the identity graph homomorphism. Thus we consecutively get $\langle 0, n\rangle \in \hat{\sigma}(e)$ by Lemma A.3, $\langle 0, n\rangle \in \hat{\sigma}(f)$ by assumption, and $\langle 0, n\rangle \in \hat{\sigma}(v)$ for some $v \in [f]$ by Lemma A.3 again. Lemma A.2 finally gives $g(u) = \langle \sigma, 0, n\rangle \lhd g(v)$, as required.

For the backward implication, assume the right-hand side and let $\sigma : \Sigma \to \mathcal{P}(X \times X)$ be a valuation into a member of REL. For all $i, j \in X$, we have

$$\begin{aligned}
& \langle i, j\rangle \in \hat{\sigma}(e) \\
\Leftrightarrow\ & \langle i, j\rangle \in \hat{\sigma}(u) \text{ for some } u \in [e] && \text{(by Lemma A.3)} \\
\Leftrightarrow\ & \langle \sigma, i, j\rangle \lhd g(u) \text{ for some } u \in [e] && \text{(by Lemma A.2)} \\
\Rightarrow\ & \langle \sigma, i, j\rangle \lhd g(u) \text{ for some } u, v \text{ s.t. } v \in [f] \text{ and } g(u) \lhd g(v) && \text{(by assumption)} \\
\Rightarrow\ & \langle \sigma, i, j\rangle \lhd g(v) \text{ for some } v \in [f] && \text{(by transitivity of } \lhd) \\
\Leftrightarrow\ & \langle i, j\rangle \in \hat{\sigma}(v) \text{ for some } v \in [f] && \text{(by Lemma A.2)} \\
\Leftrightarrow\ & \langle i, j\rangle \in \hat{\sigma}(f) && \text{(by Lemma A.3)}
\end{aligned}$$

Whence $\hat{\sigma}(e) \subseteq \hat{\sigma}(f)$, and thus $\mathsf{REL} \models e \leq f$ as required. ◀