

A cut-free cyclic proof system for Kleene algebra

Anupam Das and Damien Pous

Plume team — Univ. Lyon, CNRS, ENS de Lyon, UCB Lyon 1, LIP

Abstract. We introduce a sound non-wellfounded proof system whose regular proofs are complete for (in)equations between regular expressions. We achieve regularity by using *hypersequents* rather than usual sequents, with more structure in the succedent, and relying on the discreteness of *rational languages* to drive proof search. By inspection of the proof search space we extract a PSPACE bound for the system, which is optimal for deciding such (in)equations.

1 Introduction

Kleene algebra is a finite quasi-equational theory over regular expressions [9], which admits *formal languages* and *binary relations* as free models. Indeed, Kroh and Kozen independently proved its completeness: every equation which is universally valid in one of those models, or equivalently, whose members denote the same rational language, is provable from the axioms of Kleene algebra [18] [25]. This theorem is important in practice since it shows that the equational theory of Kleene algebra is decidable, and actually PSPACE-complete: it reduces to the problem of comparing rational languages. Thanks to the model of binary relations, Kleene algebra and its extensions have been used to reason abstractly about program correctness [21,22,2,14,1]. The aforementioned decidability result actually made it possible to automatise many reasoning steps in proof assistants [5,23,28].

Following work in substructural logics about residuated lattices [26], Jipsen proposed a sequent system for Kleene algebra and asked whether the cut-rule is admissible in this system [16]—Buszkowski proved it is not [8]. Wurm recently proposed a different sequent system [31]; unfortunately his cut-admissibility theorem does not hold (see App. A). Proofs in these two systems are finite.

Palka proposed a sequent system for *star-continuous action lattices* [27], and thus in particular for Kleene algebra. She proved completeness and cut-elimination. Her system is wellfounded and relies on an ‘ ω -rule’ for Kleene star with infinitely many premisses, in the traditional school of infinitary proof theory [30]. Doing so has the advantage of being simple, but it does not make it possible to consider ‘regular’ proofs: every proof is necessarily infinite. As a consequence, such a system cannot be used for proof search.

We introduce in this paper a non-wellfounded sequent system HKA^∞ for Kleene algebra, which we prove sound and complete (Sects. 5 and 6). This system is cut-free and admits the subformula property. We actually prove that its regular fragment—those proofs which can be described as potentially cyclic finite

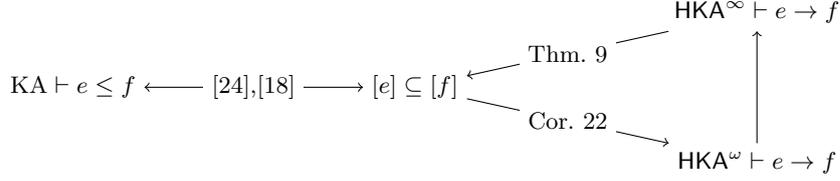


Fig. 1. Algebraic and proof-theoretic views on rational language equivalence.

graphs—is complete. We give a diagrammatic summary of these contribution in Fig. 1, where we write HKA^ω to denote the regular fragment of HKA^∞ .

Starting from Palka’s system, a natural idea when looking for a regular system consists in replacing her infinitary rules for Kleene star by finitary ones, and allowing non-wellfounded proofs. Doing so, we obtain the system LKA^∞ we describe in Sect. 3: proofs that are well-founded but of infinite width in Palka’s system become finitely branching but infinitely deep in LKA^∞ . This new system admits an elegant proof theory, but we show that its regular fragment is not complete: there are valid inequalities which require lines of unbounded size. We solve this problem by allowing slightly more structure in the succedents of sequents, moving to *hypersequents* to design the system HKA (Sect. 4). The system HKA^ω makes it possible to give an alternative proof that the equational theory of rational languages is in PSPACE, without relying on automata-theoretic arguments (Sect. 7). We conclude this paper with some further comments and directions for future work (Sect. 8).

2 Kleene algebra

We consider *regular expressions* over a finite *alphabet* A :

$$e, f ::= e \cdot e \mid e + e \mid e^* \mid 1 \mid 0 \mid a \in A$$

Sometimes we simply write ef for $e \cdot f$. Each expression e denotes a rational language $[e] \subseteq A^*$, defined in the usual way [17]. A *Kleene algebra* is a tuple $(K, 0, 1, +, \cdot, *)$ where $(K, 0, 1, +, \cdot)$ is an idempotent semiring and:¹

- (a) $1 + xx^* \leq x^*$;
- (b) if $xy \leq y$ then $x^*y \leq y$;
- (c) if $yx \leq y$ then $yx^* \leq y$.

There are several equivalent variants of this definition [9]. Intuitively we have that x^*y (resp. yx^*) is the least fixpoint of $z \mapsto y + xz$ (resp. $z \mapsto y + zx$).

We write $\text{KA} \vdash e \leq f$ if $e \leq f$ is provable from the axioms of Kleene Algebra, i.e. is true in all Kleene algebras (by completeness of first-order logic). We have the following completeness result, independently due to Kozen and Krob:

¹ Here we write $x \leq y$ as a shorthand for $x + y = y$.

$$\begin{array}{cccc}
\frac{id}{e \rightarrow e} & \frac{0-l}{\Gamma, 0, \Delta \rightarrow e} & \frac{1-l}{\Gamma, 1, \Delta \rightarrow e} & \frac{1-r}{\rightarrow 1} \\
\frac{-l}{\frac{\Gamma, e, f, \Delta \rightarrow g}{\Gamma, e \cdot f, \Delta \rightarrow g}} & \frac{+l}{\frac{\Gamma, e, \Delta \rightarrow g \quad \Gamma, f, \Delta \rightarrow g}{\Gamma, e + f, \Delta \rightarrow g}} & \frac{*l}{\frac{\Gamma, \Delta \rightarrow f \quad \Gamma, e, e^*, \Delta \rightarrow f}{\Gamma, e^*, \Delta \rightarrow f}} & \\
\frac{-r}{\frac{\Gamma \rightarrow e \quad \Delta \rightarrow f}{\Gamma, \Delta \rightarrow e \cdot f}} & \frac{+r}{\frac{\Gamma \rightarrow e_i}{\Gamma \rightarrow e_1 + e_2} \quad i \in \{1, 2\}} & \frac{*r}{\rightarrow e^*} & \frac{*r}{\frac{\Gamma \rightarrow e \quad \Delta \rightarrow e^*}{\Gamma, \Delta \rightarrow e^*}}
\end{array}$$

Fig. 2. The rules of the system LKA.

Theorem 1 ([18,25]). $\text{KA} \vdash e \leq f$ if and only if $[e] \subseteq [f]$.

Formal languages, i.e. subsets of A^* , form a Kleene algebra, so the left-right implication is straightforward. The converse, completeness, is much harder.

3 An intrinsically non-regular system: LKA^∞

A sequent is an expression $\Gamma \rightarrow e$, where Γ is a list of regular expressions and e is a regular expression. For such a sequent we refer to Γ as the *antecedent* and e as the *succedent*. We say that a sequent $e_1, \dots, e_n \rightarrow e$ is *valid* if $[e_1 \cdots e_n] \subseteq [e]$. i.e. the comma is interpreted as sequential composition, and the sequent arrow as containment. We refer to expressions as ‘formulae’ when it is more natural proof theoretic terminology, e.g. ‘subformula’ or ‘principal formula’.

The system LKA is defined by the rules given in Fig. 2. Without the $*$ -rules, LKA is essentially a fragment of non-commutative intuitionistic linear logic [13],² or alternatively the Lambek calculus [26], restricted to the following connectives: multiplicative conjunction (\cdot), additive disjunction ($+$) and additive falsity (0) (for which there is no right rule). The rules for Kleene star can be seen as arising from the characterisation of e^* as a fixed point: $e^* = \mu x.(1 + ex)$. In contrast, Palka [27] interprets Kleene star as an infinite sum $e^* = \Sigma_i e^i$, corresponding to $*$ -continuity in a Kleene algebra, whence her left rule for Kleene star with infinitely many premisses and the infinitely many corresponding right rules.

All of the rules of LKA are *sound*: if each premiss of a rule is true in a Kleene algebra then so is its conclusion. LKA also has the *subformula property*: every expression occurring in a premiss of a rule instance is a subformula of an expression in its conclusion.

LKA, construed as a system of usual finite proofs, is not complete: there are valid sequents which conclude no finite derivation of LKA rules. For instance $(b + c)^* \rightarrow (c + b)^*$, cf. Ex. 5 later, has no finite proof in LKA. Accordingly, we consider *non-wellfounded* proofs. Intuitively, these are infinite trees built from the rules of LKA. More formally:

² This logic is non-commutative because there is no exchange rule, and intuitionistic since there is exactly one formula on the right-hand side.

Definition 2. A (binary, possibly infinite) tree is a prefix-closed subset of $\{0, 1\}^*$. An LKA^∞ -preproof is a labelling π of a tree by sequents such that, for every node v with children v_1, \dots, v_n ($n = 0, 1, 2$), the expression $\frac{\pi(v_1) \cdots \pi(v_n)}{\pi(v)}$ is an instance of an LKA rule. A preproof is regular if it has only finitely many distinct subtrees, i.e. it can be expressed as the infinite unfolding of a finite graph.

Preproofs are not always sound (hence the terminology). Consider, for instance, the preproof on the right deriving a non-valid sequent, where we use the symbol \boxtimes to indicate a circularity. Fortunately, we may rule out such behaviours by a simple fairness criterion:

$$\frac{\frac{1-r}{\rightarrow 1} \quad \frac{\dots}{a \rightarrow 1^*} \boxtimes}{*r \frac{a \rightarrow 1^*}{a \rightarrow 1^*} \boxtimes}$$

Definition 3. A proof is a preproof that is fair for $*l$, i.e. where every infinite branch contains infinitely many occurrences of $*l$. We write $\text{LKA}^\infty \vdash e \leq f$ if there is an LKA^∞ proof of $e \rightarrow f$.

This criterion is somewhat simpler than the ones from other works, e.g. [12] and [10], which require a finer analysis of formula occurrences in infinite branches. However, for our purposes, the condition above suffices and, indeed, leads to a simpler correctness checking procedure for a preproof, cf. Sect. 7.

Theorem 4 (Soundness). If $\text{LKA}^\infty \vdash e_1, \dots, e_n \rightarrow e$, then $[e_1 \cdots e_n] \subseteq [e]$.

Proof (idea). Similar to the proof we give in Sect. 5 for the system HKA^∞ . \square

Let us consider an example of a proof in LKA^∞ whose conclusion admits no finite LKA proof. Again, we will use the symbol \boxtimes to indicate circularities.

Example 5. Here is an LKA^∞ proof of $a^*(b+c)^* \leq a^*(c+b)^*$:

$$\frac{\frac{\frac{\frac{id}{b \rightarrow b} \quad \frac{id}{c \rightarrow c}}{+r \frac{b \rightarrow c+b}{b \rightarrow c+b}} \quad \frac{\dots}{c \rightarrow c+b}}{+l \frac{b+c \rightarrow c+b}{b+c \rightarrow c+b}} \quad \frac{\dots}{(b+c)^* \rightarrow (c+b)^*} \boxtimes}{*r \frac{b+c, (b+c)^* \rightarrow (c+b)^*}{(b+c)^* \rightarrow (c+b)^*} \boxtimes} \quad \frac{id}{a^* \rightarrow a^*} \quad \frac{*r}{\rightarrow (c+b)^*} \quad \frac{*l}{(b+c)^* \rightarrow (c+b)^*}}{*r \frac{a^*, (b+c)^* \rightarrow a^*(c+b)^*}{a^*(b+c)^* \rightarrow a^*(c+b)^*}}$$

Note that while LKA^∞ satisfies the subformula property, the size and number of sequents occurring in a proof is not *a priori* bounded, due to the $*l$ rule. In fact, this system does not admit finite regular proofs for all valid sequents. An example is the inequality $aa^* \leq a^*a$, whose only LKA^∞ proof is the following:

$$\frac{\frac{\frac{\frac{id}{a \rightarrow a} \quad \frac{*r}{\rightarrow a^*}}{*r \frac{a \rightarrow a^*}{a \rightarrow a^*}} \quad \frac{id}{a \rightarrow a} \quad \dots}{*r \frac{a, a \rightarrow a^*a}{a, a \rightarrow a^*a}} \quad \frac{*l}{a, a, a, a^* \rightarrow a^*a}}{*l \frac{a, a^* \rightarrow a^*a}{aa^* \rightarrow a^*a}} \quad (1)$$

This proof necessarily will contain all sequents of the form $a, \dots, a, a^* \rightarrow a^*a$.

Many cases of non-regularity can be avoided by adding *symmetric* versions of the $*$ rules in LKA:

$${}_{*-l'} \frac{\Gamma, \Delta \rightarrow f \quad \Gamma, e^*, e, \Delta \rightarrow f}{\Gamma, e^*, \Delta \rightarrow f} \quad {}_{*-r'} \frac{\Gamma \rightarrow e^* \quad \Delta \rightarrow e}{\Gamma, \Delta \rightarrow e^*} \quad (2)$$

For instance, using these rules, it is not hard to see that the situation (1) above can be avoided by a finite proof that is not even cyclic (see App. B).

However, adding the rules from (2) above does not always suffice for regularity. Consider the valid sequent $a^* \rightarrow (aa)^* + a(aa)^*$. It is not hard to see that any proof of this must contain a path of just $*-l$ steps, since we are never able to apply a $+ -r$ step while there remains an a^* on the left. Thus it admits no regular proof in LKA^∞ , even with the rules from (2).

Similarly, consider the valid sequent $(a + b)^* \rightarrow a^*(ba^*)^*$. Any proof of this sequent, even with symmetric rules, must contain some path of sequents whose antecedents denote languages containing $a^m(a + b)^*a^n$, for sufficiently large m, n . Along such a path a $-r$ step is never valid and so one is forced to apply $*-l$ and $+ -l$ rules forever, again yielding a non-regular proof.

The next section presents a system where we can avoid these issues by reasoning *underneath* \cdot and $+$ in the succedent, and thus arrive at a general completeness theorem for regular proofs. We come back to the problem cases discussed above at the end of the next section, in Ex. 23.

4 A calculus HKA^ω of regular proofs

We denote lists of formulae by Γ, Δ etc. as before. We will use X, Y, Z to vary over multisets of lists. A *hypersequent* is an expression $\Gamma \rightarrow X$, where Γ is a list and X is a multiset of lists. Henceforth we may simply say ‘sequent’ instead of ‘hypersequent’ when it is not ambiguous. We use the comma, ‘,’ for both delimiting lists and denoting union of multisets, using angled brackets $\langle \cdot \rangle$ to distinguish lists in a multiset. Here is the general form of a hypersequent:

$$e_1, \dots, e_l \rightarrow \langle f_{11}, \dots, f_{1n_1} \rangle, \dots, \langle f_{m1}, \dots, f_{mn_m} \rangle$$

We extend the notion of language of a regular expression to lists of expressions by setting $[\langle e_1, \dots, e_n \rangle] = [e_1 \cdots e_n]$, and to multisets of such lists by setting $[\langle \Gamma_1 \rangle, \dots, \langle \Gamma_n \rangle] = \bigcup_i [\Gamma_i]$. The hypersequent $\Gamma \rightarrow X$ is *valid* if $[\Gamma] \subseteq [X]$.

If $X = \langle \Delta_1 \rangle, \dots, \langle \Delta_n \rangle$, we write $\langle \Sigma \rangle X$ for the set $\langle \Sigma, \Delta_1 \rangle, \dots, \langle \Sigma, \Delta_n \rangle$. When Σ is a singleton $\langle e \rangle$ we simply write eX instead of $\langle e \rangle X$, as an abuse of notation.

The rules of the system HKA are given in Fig. 3. Notice that these rules satisfy the subformula property. The left logical rules are exactly those of LKA, lifted to hypersequents. The right logical rules slightly differ, to take advantage of the richer structure of the sequents. Weakening and contractions are allowed on the right of the sequents; the identity axiom from LKA is decomposed into an axiom for the empty lists, and a ‘modal’ rule (K).

Left logical rules:

$$\begin{array}{c}
\frac{}{0-l \Gamma, 0, \Delta \rightarrow} \quad \frac{\Gamma, \Delta \rightarrow X}{1-l \Gamma, 1, \Delta \rightarrow X} \quad \frac{\Gamma, e, f, \Delta \rightarrow X}{\cdot-l \Gamma, e \cdot f, \Delta \rightarrow X} \\
\frac{\Gamma, e, \Delta \rightarrow X \quad \Gamma, f, \Delta \rightarrow X}{+l \Gamma, e + f, \Delta \rightarrow X} \quad \frac{\Gamma, \Delta \rightarrow X \quad \Gamma, e, e^*, \Delta \rightarrow X}{*-l \Gamma, e^*, \Delta \rightarrow X}
\end{array}$$

Right logical rules:

$$\begin{array}{c}
\frac{\Gamma \rightarrow X, \langle \Delta, \Sigma \rangle}{1-r \Gamma \rightarrow X, \langle \Delta, 1, \Sigma \rangle} \quad \frac{\Gamma \rightarrow X, \langle \Delta, e, f, \Sigma \rangle}{\cdot-r \Gamma \rightarrow X, \langle \Delta, e \cdot f, \Sigma \rangle} \\
\frac{\Gamma \rightarrow X, \langle \Delta, e, \Sigma \rangle, \langle \Delta, f, \Sigma \rangle}{+r \Gamma \rightarrow X, \langle \Delta, e + f, \Sigma \rangle} \quad \frac{\Gamma \rightarrow X, \langle \Delta, \Sigma \rangle, \langle \Delta, e, e^*, \Sigma \rangle}{*-r \Gamma \rightarrow X, \langle \Delta, e^*, \Sigma \rangle}
\end{array}$$

Identity, modal and structural rules:

$$\frac{}{id \rightarrow \langle \rangle} \quad \frac{\Gamma \rightarrow X}{K e, \Gamma \rightarrow eX} \quad \frac{\Gamma \rightarrow X}{wk \Gamma \rightarrow X, \langle \Delta \rangle} \quad \frac{\Gamma \rightarrow X, \langle \Delta \rangle, \langle \Delta \rangle}{cntr \Gamma \rightarrow X, \langle \Delta \rangle}$$

Fig. 3. Rules for the system HKA.

Definition 6. Preproofs and proofs of HKA^∞ are defined analogously to LKA^∞ ; in particular proofs require fairness of $*-l$. We write HKA^ω to denote the fragment of regular proofs of HKA^∞ , i.e. ones with only finitely many distinct subtrees.

Remark 7 (Invertibility and cancellation). All rules of HKA except wk and K are *strongly* invertible: truth of the conclusion implies truth of all premisses in any Kleene algebra. K is not strongly invertible, even in its atomic form, due to the possible existence of 0-divisors. It is however weakly invertible when e is atomic: the validity of the conclusion implies the validity of the premiss.³ On the other hand, the non-invertibility of wk turns out to be crucial for completeness, from a complexity theoretic point of view, cf. Sect. 7.

Example 8 (Atomic modal steps). We can reduce K steps to atomic form in HKA^ω . This is proved by structural induction on the modal expression; the key case is when it is a $*$ -formula, where non-wellfoundedness appears:

$$\frac{\frac{\frac{\frac{\vdots}{*-l \frac{e^*, \Gamma \rightarrow e^* X}{\text{✕}}}}{IH \frac{e, e^*, \Gamma \rightarrow \langle e, e^* \rangle X}{\text{✕}}}}{\frac{\Gamma \rightarrow X}{*-r, wk \frac{\Gamma \rightarrow e^* X}{\text{✕}}}}{\frac{\Gamma \rightarrow e^* X}{*-l \frac{e^*, \Gamma \rightarrow e^* X}{\text{✕}}}}$$

The derivation marked IH is obtained from the inductive hypothesis on e .

³ Note that atomicity of e really is required for this, even in the usual discrete language model. For instance, we have $[a^*ab] \subseteq [a^*b(a+b)^*]$, but $[ab] \not\subseteq [b(a+b)^*]$.

5 Soundness

We now show that HKA^∞ proofs derive only valid sequents. Throughout this section and later, we use standard proof theoretic terminology about *ancestry* in proofs, e.g., from [7].

Theorem 9 (Soundness). *If $\text{HKA}^\infty \vdash \Gamma \rightarrow X$, then $[\Gamma] \subseteq [X]$.*

Before giving the proof, we need the following intermediate result.

Lemma 10. *If $\text{HKA}^\infty \vdash \Gamma, e^*, \Delta \rightarrow X$ then, for $n \in \mathbb{N}$, $\text{HKA}^\infty \vdash \Gamma, e^n, \Delta \rightarrow X$.⁴*

Proof. We define appropriate preproofs by induction on n . Replace every direct ancestor of e^* by e^n , adjusting origins as follows,

$$*_l \frac{\Gamma, \Delta \rightarrow X \quad \Gamma, e, e^*, \Delta \rightarrow X}{\Gamma, e^*, \Delta \rightarrow X} \quad \mapsto \quad *_l \frac{\Gamma, \Delta \rightarrow X}{\Gamma, 1, \Delta \rightarrow X} \quad \text{or} \quad *_l \frac{\Gamma, e, e^{n-1}, \Delta \rightarrow X}{\Gamma, e^n, \Delta \rightarrow X}$$

when $n = 0$ or $n > 0$, respectively. In the latter case we appeal to the inductive hypothesis. Notice that, on branches where e^* is never principal, this is simply a global substitution of e^n for e^* everywhere along the branch. The preproof resulting from this entire construction is fair since every infinite branch will share a tail with the proof we began with. \square

Now we define a measure with which Thm. 9 will be proved by induction.

Definition 11 (Measure of a sequent). *The $*$ -height of a regular expression e , denoted $h_*(e)$, is the maximum nesting of $*$ in its term tree. Formally:*

- $h_*(0) = h_*(1) = h_*(a) = 0$.
- $h_*(e \cdot f) = h_*(e + f) = \max(h_*(e), h_*(f))$.
- $h_*(e^*) = h_*(e) + 1$.

The weighted $$ -height of a list Γ of expressions, denoted $wh_*(\Gamma)$ is the multiset $\{h(e) : e \in \Gamma\}$. We totally order such multisets under a well-known ordering [11]: for two multisets N, N' we say $N < N'$ if there is a multiset surjection $f : N \rightarrow N'$ such that, for each $n \in N'$, we have either $f^{-1}(n) = \{n\}$ or, for every $m \in f^{-1}(n)$, $m < n$.*

Fact 12 *For every rule of HKA^∞ except $*_l$, the antecedent of each premiss has weighted $*$ -height bounded by that of the antecedent of the conclusion.*

For the $*_l$ rule also notice that, bottom-up, the maximum $*$ -height of an expression in the antecedent does not increase. We now prove our soundness result:

Proof (of Thm. 9). Let π be an HKA^∞ proof of $\Sigma \rightarrow X$ and let us proceed by induction on the weighted $*$ -height of the antecedent Σ . For each infinite branch of π take the least $*_l$ step that occurs and notice that their conclusions, together with any initial sequents in π , form a bar B through the infinite tree of π . Since

⁴ Strictly speaking, we should bracket e^n as $e(e(\dots(ee)))$ and set e^0 to 1.

π labels a binary tree, the prefix closure of B must be finite by König's Lemma and thus, if each of the sequents of B is valid then so is the conclusion of π by the soundness of finite HKA derivations.

Now, consider a subproof π' that derives a sequent in B . If it is initial it is valid; otherwise it must derive a sequent $\Gamma, f^*, \Delta \rightarrow Y$ where f^* is principal for the concluding $*$ - l -step of π' . By construction and Fact 12 notice that $wh_*(\Gamma, f^*, \Delta) \leq wh_*(\Sigma)$. Now, by Lemma 10, π' can be transformed into proofs π'_n of $\Gamma, f^n, \Delta \rightarrow Y$ for each $n \in \mathbb{N}$. Each π'_n derives a sequent whose antecedent has lower weighted $*$ -height than that of Σ , and so is sound by the inductive hypothesis. Finally, this means that $\Gamma, f^*, \Delta \rightarrow Y$ is valid, by definition of Kleene star for languages, and hence $\Sigma \rightarrow X$ is valid after all. \square

\square

6 Completeness

Infinite non-wellfounded proofs are easily seen to be complete: bottom-up, simply apply left rules forever (they are invertible); the only normal forms of this procedure will have a finite word as the antecedent, whence we may perform the correct finite sequence of right steps to finish proof search.

In this section we give a more sophisticated argument showing that the *regular* fragment of HKA^∞ is complete: each valid inclusion has a finite circular proof.

6.1 A regular class of proofs

We first define a class of proofs which can be made regular in a systematic way.

Definition 13. *A preproof is leftmost if the principal formula of every logical step is at the beginning of a list, either in the antecedent or the succedent.*

For regularity, the most useful property of a leftmost proof is the following:

Theorem 14. *A leftmost preproof contains only lists of length linear in the size of the end-sequent. Hence only finitely many lists occur in a leftmost preproof.⁵*

Before we can prove this, let us recall some basic notions regarding terms. An *occurrence* in e is a subformula of e together with its position in e . We often omit this positional information when it is unambiguous.

Definition 15 (Total order on occurrences). *Given a fixed term, we define a relation \preceq on the occurrences in it as follows: $e \preceq f$ if f contains e , or if e and f are disjoint and e occurs to the left of f .*

Due to the tree structure of a term, any two occurrences in a term are either disjoint or one is contained in the other, so we have the following:

Proposition 16. *\preceq is a total order on the occurrences in a term.*

⁵ *A priori*, this could still be exponentially many in the size of the end-sequent.

In a preproof, let us identify every expression occurring as an occurrence of a term in the end-sequent in the natural way, due to the subformula property and via the usual notions of proof ancestry. In this way, we can meaningfully compare any two expressions in a preproof under \preceq . We have the following:

Lemma 17. *In any leftmost preproof every list is strictly increasing under \preceq .*

Now we can prove the bound on the size of lists in leftmost preproofs:

Proof (of Thm. 14). Every term in a preproof is an ancestor of an occurrence in a term of the end sequent, by the subformula property and usual notions of proof ancestry. Moreover, no occurrence can appear twice in the same list, otherwise we would contradict Lemma 17. \square

We still do not quite have regularity, since in the succedent we may have multisets with arbitrarily many occurrences of the same list. Naturally, we appeal to the right structural rules to ‘merge’ occurrences in such a situation:

Corollary 18. *A leftmost preproof in HKA^∞ can be transformed into one of the same end-sequent that contains only finitely many distinct sequents.*

Proof. By Thm. 14 only finitely many distinct lists occur in a leftmost preproof. Thanks to contraction and weakening, we can always write succedents with at most two copies of each distinct list, of which there are only finitely many. \square

It remains to show that we may place backpointers while preserving correctness:

Corollary 19. *A leftmost proof in HKA^∞ can be transformed into a regular proof, i.e. in HKA^ω , with the same end-sequent.*

Proof. Assuming only finitely many distinct sequents occur, by Cor. 18 above, in each infinite branch some sequent occurs infinitely often, by the pigeonhole principle. This means that, due to fairness, for each infinite branch we may identify two instances of the same sequent with a $*-l$ -step in between, whence we may correctly place a backpointer and preserve fairness. \square

6.2 Completeness of leftmost proofs

Thanks to Cor. 19, for completeness of HKA^ω it now suffices to show that any valid hypersequent admits a leftmost proof in HKA^∞ . We do so by providing a leftmost proof search strategy for which we need the following important result:

Lemma 20 (Productivity on the right). *Suppose there is a (finite) HKA derivation of right logical rules of the following format,⁶*

$$\frac{\Gamma \rightarrow X, \langle e^*, \Delta \rangle \quad \left| \begin{array}{c} \pi \\ \Gamma \rightarrow Y, \langle \Delta \rangle, \langle e, e^*, \Delta \rangle \end{array} \right.}{\Gamma \rightarrow Y, \langle e^*, \Delta \rangle} \text{*}-r$$

⁶ Notice that right logical rules do not branch.

such that the list $\langle e^*, \Delta \rangle$ in the initial sequent is an ancestor of that from the end sequent. If the end sequent is valid, then so is $\Gamma \rightarrow X$.

Proof. Since all right logical rules of HKA are invertible, it suffices to show that $\langle e^*, \Delta \rangle$ in the initial sequent is redundant, i.e. that already $[X] \supseteq [\langle e^*, \Delta \rangle]$. For this, we appeal to soundness of fair preproofs, Thm. 9, and show that HKA^∞ proves the corresponding sequent: $e^*, \Delta \rightarrow X$.⁷ We construct an appropriate proof π' bottom-up by induction on the length of π where, for each right logical rule in π , we apply the analogous left logical rule in π' along the appropriate branch. Each leaf of π' will be of the form $\Sigma \rightarrow X$, where Σ is a list occurring in the succedent of the premiss of π , by construction. Now, if $\Sigma \in X$ then we can conclude by weakening, K and identity; otherwise Σ is $\langle e^*, \Delta \rangle$, whence we can conclude by circularity. Notice that π' is fair due to the fact that the bottommost step is a $*-l$ due to the analogous $*-r$ beneath π . \square

We can now prove our main completeness result:

Theorem 21. *Every valid hypersequent has a leftmost proof in HKA^∞ .*

Proof. Construct a leftmost HKA^∞ preproof bottom-up as follows:

- (i) Apply leftmost left logical rules as long as possible. After this any leaves will be valid, by invertibility of logical rules, and of the form:

$$\rightarrow X \quad \text{or} \quad a, \Gamma \rightarrow X$$

- (ii) Apply leftmost right logical rules until the succedent contains only lists beginning with a $*$ -term that have already been decomposed⁸ or lists for which no leftmost right logical rule applies. This terminates after finitely many steps due to Thm. 14 and since only $*-r$ can increase the length of a list in the succedent. All resulting leaves must be valid, again by invertibility.
- (iii) Now we apply wk to weaken any appropriate lists in the succedent that have already been decomposed. Leaves remain valid due to Lemma 20 and must be of the form:

$$\rightarrow (\langle \rangle, \langle a_1, X_1 \rangle, \dots, \langle a_n, X_n \rangle) \quad \text{or} \quad a, \Gamma \rightarrow (\langle \rangle, \langle a_1, X_1 \rangle, \dots, \langle a_n, X_n \rangle)$$

In the former case, since we have preserved validity going upwards, we must have that the empty list occurs in the succedent, whence we can close the branch by several wk steps and id .

In the latter case, again since we have preserved validity going upwards, we must be able to weaken any list that begins with an a_i that is not a and preserve validity. Now any remaining leaves are of the form,

$$a, \Gamma \rightarrow aX$$

whence we can apply K and preserve validity by Rmk. 7. Now go back to (i) and repeat the entire procedure.

⁷ This argument is akin to applying a cut, which is sound since we are only applying it once, and at the meta-level.

⁸ Here we mean in the sense that it is identical to a descendant, as in Lemma 20.

7 Complexity matters and algorithms for proof search

We present in this section a brief overview of the complexity theoretic aspects of our system HKA^ω .

Checking validity of a regular preproof. When a preproof is given as a tree with backpointers, it is not difficult to see that checking validity is feasible (i.e. in polynomial time), since we may simply exhaust the paths of the tree, of which there are linearly many, to exclude the existence of a $*-l$ -free loop. When the preproof is given as an arbitrary graph the problem is a little more subtle, but remains feasible. Construing sequents as nodes and inference steps as edges, let us delete any edge that corresponds to a $*-l$ step. Notice that the original preproof was valid just if there are no infinite paths in the resulting graph, i.e. it is *acyclic*. This can be decided by computing its transitive closure, hence:

Proposition 25. *Validity of a HKA^ω -preproof, given as an arbitrary directed graph, is polynomial-time decidable.*

Notice that this bound is lower than those for circular proofs in other systems, e.g. [6,12], since logics with more sophisticated fixed points require a more general correctness criterion reducing to the inclusion of *Büchi automata*, a problem that is PSPACE-complete.

Complexity of proof search. HKA^ω proof search yields an optimal bound for deciding equations of Kleene algebra via the induced loop-checking procedure:

Proposition 26. *HKA^ω proof search induces a PSPACE decision procedure for inequalities between regular expressions.*

Proof (sketch). For a leftmost proof we give a polynomial bound on the depth until a loop occurs. Notice that succedents only grow polynomially in depth and $*-height$, by inspection of HKA , and so this indeed yields a PSPACE bound.

Each time a K step is applied, bottom-up, it is on an atom occurrence that may not reoccur, unless we have already formed a loop, namely by unfolding the same $*-expression$, which by construction contains a $*-l$. Every other leftmost step decreases the size of the leftmost term in a list. Thus, any path in a leftmost proof will hit a loop within polynomially many steps. \square

Notice that, while almost every step in HKA is invertible, it is the crucial applications of weakening in the procedure of Thm. 21, justified by Lemma 20, which requires proof search to operate in PSPACE rather than CONP. Indeed, it is the number of wk steps along any proof path that allows search complexity to climb up the polynomial hierarchy. This cannot be uniformly bounded since deciding inequalities of regular expressions is known to be PSPACE-complete.

8 Conclusions and further work

We proposed a regular and cut-free hypersequent system HKA^ω , which we proved sound and complete for rational language inclusion, and thus for Kleene algebra. We conclude with further comments and directions for future work.

Richer systems for theorem proving. Now that we have a completeness theorem for HKA^ω , we could envisage enriching the system with more (sound) rules that might be more natural from the point of view of theorem proving. For instance, we might imagine alternative right logical rules for $+$ and $*$ as follows,

$$\frac{\Gamma \rightarrow X, \langle \Delta, e_i, \Sigma \rangle}{\Gamma \rightarrow X, \langle \Delta, e_1 + e_2, \Sigma \rangle} \quad \frac{\Gamma \rightarrow X, \langle \Delta, \Sigma \rangle}{\Gamma \rightarrow X, \langle \Delta, e^*, \Sigma \rangle} \quad \frac{\Gamma \rightarrow X, \langle \Delta, e, \Sigma \rangle}{\Gamma \rightarrow X, \langle \Delta, e^*, \Sigma \rangle} \quad \frac{\Gamma \rightarrow X, \langle \Delta, e^*, e^*, \Sigma \rangle}{\Gamma \rightarrow X, \langle \Delta, e^*, \Sigma \rangle}$$

Such systems are more expressive since they can encode not only the rules of HKA but also symmetric variants, e.g. unfolding $*$ to the right rather than the left.⁹ An illustrative example is the inequality $a^*a \leq a^*$, which was one source of irregularity for LKA^∞ . Contrast the following two proofs, the left of which follows a leftmost strategy in HKA^ω , the right of which uses the rules above and is not even circular:

$$\begin{array}{c} \frac{id}{\rightarrow \langle \rangle} \\ \frac{wk}{\rightarrow \langle \rangle, \langle a, a^* \rangle} \\ \frac{*r}{\rightarrow a^*} \\ \frac{K}{a \rightarrow \langle a, a^* \rangle} \\ \frac{wk}{a \rightarrow \langle \rangle, \langle a, a^* \rangle} \\ \frac{*r}{a \rightarrow \langle a^* \rangle} \\ \frac{*l}{a^*, a \rightarrow \langle a^* \rangle} \\ \frac{..l}{a^*a \rightarrow \langle a^* \rangle} \end{array} \quad \begin{array}{c} \vdots \\ \frac{*l}{a^*, a \rightarrow \langle a^* \rangle} \boxtimes \\ \frac{K}{a, a^*, a \rightarrow \langle a, a^* \rangle} \\ \frac{wk}{a, a^*, a \rightarrow \langle \rangle, \langle a, a^* \rangle} \\ \frac{*r}{a, a^*, a \rightarrow \langle a^* \rangle} \boxtimes \\ \frac{..l}{a^*, a \rightarrow \langle a^* \rangle} \\ \frac{..l}{a^*a \rightarrow \langle a^* \rangle} \end{array} \quad \begin{array}{c} \frac{id}{\rightarrow \langle \rangle} \\ \frac{K}{a \rightarrow \langle a \rangle} \\ \frac{K}{a^*, a \rightarrow \langle a^*, a \rangle} \\ \frac{K}{a^*, a \rightarrow \langle a^*, a^* \rangle} \\ \frac{..l}{a^*, a \rightarrow \langle a^* \rangle} \\ \frac{..l}{a^*a \rightarrow \langle a^* \rangle} \end{array}$$

Extensions of Kleene algebra. Kleene algebra can be extended with operations such as *meet* [19], *residuals* [29], or *tests* [20]. One can thus ask whether we can obtain regular sequent systems for such extensions. Meets (\cap) and residuals (\multimap) correspond to additive conjunction and linear implications in (non-commutative) linear logic; they could easily be added to LKA^∞ (Palka actually includes them in her system [27]), but it is unclear how to add them to our hypersequent system while preserving regular cut-free completeness. An important difficulty here is that the free model for such structures is not the obvious language model.¹⁰ In contrast, Kleene algebra with tests, whose free model is that of *guarded string* languages [20], could be handled using our approach. It would also be interesting to try adapt our systems to ω -regular expressions, which denote languages of *infinite words* and for which automaton models and notions of derivative are well-defined.

Cut-elimination. By completeness, any reasonable ‘cut rule’ is admissible in HKA^ω . A natural question is whether one can prove a direct cut-elimination result,

⁹ Notice that the $*$ rules here correspond in fact to an alternative fixed point definition of e^* : $\mu x.(1 + e + xx)$.

¹⁰ Notice also that while it would be natural to enrich the antecedent structure for \cap as we did in succedents for $+$, there is a difficult asymmetry in that $x(y + z) = xy + xz$ but $x(y \cap z) \not\leq xy \cap xz$.

using proof theoretic methods. There are several difficulties here: first one has to define a general enough notion of cut for the hypersequent system; second one has to come up with an appropriate correctness criterion for preproofs with cuts (fairness as in Dfn. 6 is not enough to guarantee soundness); finally, the regular system being complete, one would certainly like to prove that cut-elimination preserves regularity. Such a cut-elimination result would make it possible to interpret Kleene algebra proofs directly into HKA^ω , without going through the free model (languages). This could be helpful to handle extensions of Kleene algebras whose free model is unknown, for instance with meet or with residuals.

Towards an alternative completeness result for KA. Conversely to the previous comments, an interesting question is whether our completeness result for the regular system HKA^ω can be used to obtain an *alternative* proof of the completeness of Kleene algebra, Thm. 1. Namely, can we prove directly that for all regular expressions e, f , $\text{HKA}^\omega \vdash e \rightarrow f$ entails $\text{KA} \vdash e \leq f$, in a direct manner? We believe this is possible, by encoding cycles in a leftmost proof as specific instances of the ‘induction’ axioms (b) and (c) from Sect. 2.¹¹ For instance a loop in a regular derivation might be transformed as follows:

$$\begin{array}{ccc}
 \begin{array}{c} e^*, f \rightarrow g \\ \left| \pi \right. \\ \text{*-l} \frac{f \rightarrow g \quad e, e^*, f \rightarrow g}{e^*, f \rightarrow g} \end{array} & \rightsquigarrow & \begin{array}{c} \text{id} \frac{}{g \rightarrow g} \\ \left| \pi[g/(e^*, f)] \right. \\ \text{(b)} \frac{e, g \rightarrow g}{e^*, g \rightarrow g} \\ \text{cut} \frac{f \rightarrow g \quad e^*, g \rightarrow g}{e^*, f \rightarrow g} \end{array}
 \end{array}$$

Generalising this idea into a full alternative proof of Kozen’s and Krob’s results is the subject of ongoing work.

References

1. C. J. Anderson, N. Foster, A. Guha, J.-B. Jeannin, D. Kozen, C. Schlesinger, and D. Walker. NetKAT: semantic foundations for networks. In *Proc. POPL*, pages 113–126. ACM, 2014.
2. A. Angus and D. Kozen. Kleene algebra with tests and program schematology. Technical Report TR2001-1844, CS Dpt., Cornell University, July 2001.
3. V. M. Antimirov. Partial derivatives of regular expressions and finite automaton constructions. *TCS*, 155(2):291–319, 1996.
4. F. Bonchi and D. Pous. Checking NFA equivalence with bisimulations up to congruence. In *Proc. POPL*, pages 457–468. ACM, 2013.
5. T. Braibant and D. Pous. An efficient Coq tactic for deciding Kleene algebras. In *Proc. 1st ITP*, volume 6172 of *LNCS*, pages 163–178. Springer, 2010.
6. J. Brotherston and A. Simpson. Sequent calculi for induction and infinite descent. *J. Log. and Comp.*, 21(6):1177–1216, 2011.

¹¹ Note that the broader problem of whether cyclic proofs can be simulated by ‘inductive’ proofs for a certain framework has no known general solution, cf. [6].

7. S. R. Buss. An introduction to proof theory. *Handbook of proof theory*, 137:1–78, 1998.
8. W. Buszkowski. On action logic: Equational theories of action algebras. *J. Log. Comput.*, 17(1):199–217, 2007.
9. J. H. Conway. *Regular algebra and finite machines*. Chapman and Hall, 1971.
10. C. Dax, M. Hofmann, and M. Lange. A proof system for the linear time μ -calculus. In *Proc. FSTTCS*, volume 4337 of *LNCS*, pages 273–284. Springer, 2006.
11. N. Dershowitz and Z. Manna. Proving termination with multiset orderings. *C. ACM*, 22(8):465–476, 1979.
12. A. Doumane, D. Baelde, L. Hirschi, and A. Saurin. Towards completeness via proof search in the linear time μ -calculus: The case of Büchi inclusions. In *Proc. LICS*, pages 377–386. ACM, 2016.
13. J.-Y. Girard. Linear logic. *TCS*, 50:1–102, 1987.
14. C. A. R. Hoare, B. Möller, G. Struth, and I. Wehrman. Concurrent Kleene Algebra. In *Proc. CONCUR*, volume 5710 of *LNCS*, pages 399–414. Springer, 2009.
15. J. E. Hopcroft and R. M. Karp. A linear algorithm for testing equivalence of finite automata. Technical Report 114, Cornell Univ., 1971.
16. P. Jipsen. From semirings to residuated Kleene lattices. *Studia Logica*, 76(2):291–303, 2004.
17. S. C. Kleene. Representation of events in nerve nets and finite automata. In *Automata Studies*, pages 3–41. Princeton University Press, 1956.
18. D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. In *Proc. LICS*, pages 214–225. IEEE, 1991.
19. D. Kozen. On action algebras. In J. van Eijck and A. Visser, editors, *Logic and Information Flow*, pages 78–88. MIT Press, 1994.
20. D. Kozen. Kleene algebra with tests. *Transactions on Programming Languages and Systems*, 19(3):427–443, May 1997.
21. D. Kozen. On Hoare logic and Kleene algebra with tests. *ACM Trans. Comput. Log.*, 1(1):60–76, 2000.
22. D. Kozen and M.-C. Patron. Certification of compiler optimizations using Kleene algebra with tests. In *Proc. CL2000*, volume 1861 of *LNAI*, pages 568–582. Springer, 2000.
23. A. Krauss and T. Nipkow. Proof pearl: Regular expression equivalence and relation algebra. *JAR*, 49(1):95–106, 2012.
24. D. KroB. A Complete System of B-Rational Identities. In *Proc. ICALP*, volume 443 of *LNCS*, pages 60–73. Springer, 1990.
25. D. KroB. Complete systems of B-rational identities. *TCS*, 89(2):207–343, 1991.
26. J. Lambek. The mathematics of sentence structure. *The American Mathematical Monthly*, 65:154–170, 1958.
27. E. Palka. An infinitary sequent system for the equational theory of $*$ -continuous action lattices. *Fundam. Inform.*, pages 295–309, 2007.
28. D. Pous. Kleene Algebra with Tests and Coq tools for while programs. In *Proc. ITP*, volume 7998 of *LNCS*, pages 180–196. Springer, 2013.
29. V. Pratt. Action logic and pure induction. In *Proc. JELIA*, volume 478 of *LNCS*, pages 97–120. Springer, 1990.
30. K. Schütte. *Proof Theory*. Grundlehren der mathematischen Wissenschaften 225. Springer Berlin Heidelberg, 1977. Translation of *Beweistheorie*, 1968.
31. C. Wurm. Kleene algebras, regular languages and substructural logics. In *Proc. GandALF*, EPTCS, pages 46–59, 2014.

A Failure of cut-admissibility in the system KL from [31]

The system KL proposed by Wurm [31] consists of finite derivations built from the rules of LKA_{cut}^∞ except for its left rule for Kleene star, and the following four rules:

$$\begin{array}{c}
 \frac{\Gamma \rightarrow e(f + g)}{\Gamma \rightarrow ef + eg} \quad D \\
 \frac{e, f \rightarrow f \quad \Gamma \rightarrow f}{e^*, \Gamma \rightarrow f} \quad *I1 \qquad \frac{f, e \rightarrow f \quad \Gamma \rightarrow f}{\Gamma, e^* \rightarrow f} \quad *I2 \qquad \frac{\Gamma \rightarrow e^* \quad \Delta \rightarrow e}{\Gamma, \Delta \rightarrow e^*} \quad I*2
 \end{array}$$

Rule D is there to “make sure distributivity holds”, although it is derivable from the other rules (with cut). Rule $I*2$ is the symmetrical version of our second right rule for Kleene star. Rules $*I1$ and $*I2$ correspond to the two implications used to define Kleene algebra.

The author proves completeness of this system (Theorem 1), mentioning on page 7 that “for the completeness part of the proof, the cut-rule is not needed, in fact it is not even mentioned!”. He later proves cut-admissibility using this observation (Theorem 10). Unfortunately, the following valid sequent cannot be proved without cut.

$$a, b^*, a \rightarrow a^*(ba^*)^*$$

Indeed, what would be the last rule? It cannot be one of the rules for Kleene star for syntactic reasons: the star on the left-hand side is in the middle of two expressions, and the ones on the right hand side are not at toplevel. For similar syntactic reasons, the only potential rule is the right rule for product (\cdot - r). But every application of this rule yield a sequent which is not valid (i.e., $a, b^*, a \rightarrow (ba^*)^*$; $b^*, a \rightarrow (ba^*)^*$; $a, b^* \rightarrow a^*$; or $a, b^*, a \rightarrow a^*$.)

This sequent is provable in KL with cut, e.g., by going through $(a + b)^*$.

B Further examples of proofs in LKA^∞

As in many common sequent systems, initial identity steps can be reduced to atomic form in LKA^∞ , although for this we crucially rely on access to non-wellfounded proofs. The interesting case is for the Kleene star:

$$\frac{\frac{\frac{\frac{\vdots}{e^* \rightarrow e^*} \quad *l}{e^* \rightarrow e^*} \quad \boxtimes}{e \rightarrow e} \quad id}{e \rightarrow e} \quad *r}{\rightarrow e^*} \quad *r \quad \frac{e, e^* \rightarrow e^*}{e^* \rightarrow e^*} \quad *l \quad \boxtimes$$

