Non-wellfounded proof theory for (Kleene+action)(Algebras+lattices)

Das, Anupam; Pous, Damien

DOI:
10.4230/LIPIcs.CSL.2018.19

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (Harvard):
Das, A & Pous, D 2018, Non-wellfounded proof theory for (Kleene+action)(Algebras+lattices). in DR Ghica & A Jung (eds), *27th EACSL Annual Conference on Computer Science Logic 2018 (CSL 2018).*, 19, Leibniz International Proceedings in Informatics, LIPIcs, vol. 119, Schloss Dagstuhl, pp. 19:1 - 19:18, 27th Annual EACSL Conference Computer Science Logic, CSL 2018, Birmingham, United Kingdom, 4/09/18. https://doi.org/10.4230/LIPIcs.CSL.2018.19

Non-Wellfounded Proof Theory For (Kleene+Action)(Algebras+Lattices)

Anupam Das
University of Copenhagen, Copenhagen, Denmark
anupam.das@di.ku.dk

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

Abstract

We prove cut-elimination for a sequent-style proof system which is sound and complete for the equational theory of Kleene algebra, and where proofs are (potentially) non-wellfounded infinite trees. We extend these results to systems with meets and residuals, capturing 'star-continuous' action lattices in a similar way. We recover the equational theory of all action lattices by restricting to regular proofs (with cut) – those proofs that are unfoldings of finite graphs.

2012 ACM Subject Classification Theory of computation → Proof theory, Theory of computation → Regular languages

Keywords and phrases Kleene algebra, proof theory, sequent system, non-wellfounded proofs

Digital Object Identifier 10.4230/LIPIcs.CSL.2018.19

Related Version Long version at `https://hal.archives-ouvertes.fr/hal-01703942`.

Funding This work has been funded by the European Research Council (ERC) under the European Union's Horizon 2020 programme (*CoVeCe*, grant agreement No. 678157, and *MiLC*, grant agreement No. 753431). 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).

1 Introduction

The axioms of *Kleene algebras* are sound and complete for the theory of regular expressions under language equivalence [24, 27, 4]. As a consequence, the equational theory of Kleene algebras is decidable (in fact PSPACE-complete). Models of these axioms of particular interest include formal languages and binary relations. For binary relations, the Kleene star is interpreted as reflexive transitive closure, whence the axioms of Kleene algebra make it possible to reason abstractly about program correctness [22, 23, 3, 19, 1]. The aforementioned decidability result moreover makes it possible to automate interactive proofs [5, 26, 30].

There are however important extensions of Kleene algebras which are not yet fully understood. These include *action algebras* [31], where two 'residual' operations are added, *Kleene lattices*, where a 'meet' operation is added, and *action lattices* [25], where all three operations are added. Pratt introduced residuals in order to *internalise* the induction rules of the Kleene star, as we explain later; they allow us to express properties of relations such as well-foundedness in a purely algebraic way [12]. Kozen added the meet operation to action algebra to obtain a structure closed under taking matrices. In the context of program verification, meets are useful since they allow us to express conjunctions of local specifications.

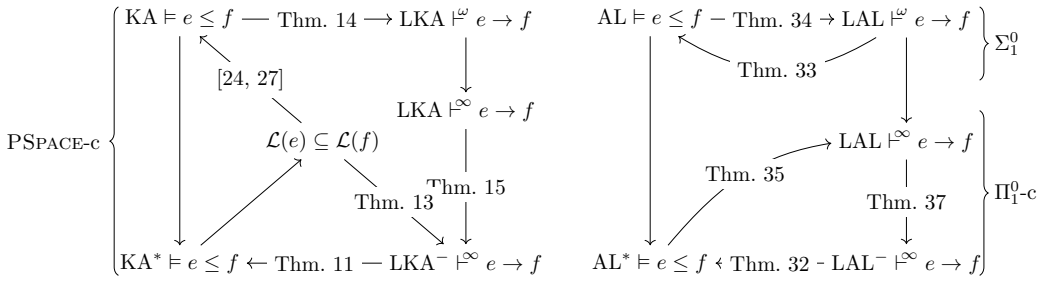

Figure 1 Context and contributions for Kleene algebra and action lattices.

Unfortunately, the decidability of the three corresponding equational theories is still open, and there is no known notion of 'free model' for them that is analogous to the rational languages for Kleene algebra. In this paper, we explore a proof-theoretic approach to such questions: we provide sequent calculi that capture these theories which we show admit a form of cut-elimination. Although this does not (yet) give us decidability, it does improve our understanding of these theories:

- we obtain a computational interpretation of proofs of inequalities in our systems as program transformers, which could prove useful to describe free models;
- we recover two conservativity results: action lattices are conservative over (star-continuous) Kleene lattices and action algebra, thanks to the sub-formula property; (these results are also implied by [29]).
- we obtain structural properties, e.g., as Whitman did when he proved cut-elimination for the theory of lattices, which we aim to exploit in consequent research.

We first focus on pure Kleene algebra, which is easier to handle and enables a simpler presentation. Being a well-established theory, we are able to relate our results to existing ones in the literature, identifying which issues become relevant when moving to extensions.

Kleene algebra

In our sequent system, called LKA, proofs are finitely branching, but possibly infinitely deep (i.e. not wellfounded). To prevent fallacious reasoning, we give a simple validity criterion for proofs with cut, and prove that the corresponding system admits cut-elimination. The difficulty in the presence of infinitely deep proofs consists in proving that cut-elimination is productive; we do so by using the natural interpretation of regular expressions as data types for parse-trees [15], and by giving an interpretation of proofs as parse-tree transformers. Such an idea already appears in [18] but in a simpler setting, for a finitary natural deduction system rather than for a non-wellfounded sequent calculus.

The results we prove about LKA are summarised in Fig. 1(left). In addition to cut-elimination (Thm. 15), we prove that the system is sound for all *star-continuous* Kleene algebras (Thm. 11), and conversely, that it is complete w.r.t. the language theoretic interpretation of regular expressions (Thm. 13). We actually refine this latter result by showing that every proof from Kleene algebra axioms can be translated into a *regular* proof with cut (Thm. 14), i.e., a proof with cut which is the unfolding of a finite graph. Note, however, that regularity is not preserved by cut-elimination: the class of cut-free regular proofs in LKA is incomplete w.r.t. Kleene algebra.

Action algebras, Kleene lattices, and action lattices

Despite its finite quasi-equational presentation, the equational theory of Kleene algebra is not finitely based: Redko proved that any finite set of equational axioms must be incomplete [32]. However, by adding two binary operations to the signature, Pratt showed how to obtain a finitely based extension which is conservative over the equational theory of Kleene algebras [31]. These two operations, called *left residual* ($\backslash$) and *right residual* ($/$), are 'adjoint' to sequential composition and, as we mentioned, such structures are called *action algebras*. Kozen then proposed *action lattices* [25], where the signature is extended further to include a binary meet operation ($\cap$). We call *Kleene lattices* the structures consisting of Kleene algebra extended just with meets.

While both action algebras and action lattices are finitely based and conservatively extend Kleene algebra, they bring some difficulties. By definition, their equational theories are at most Σ^0_1, so that they must differ from their star-continuous variants which are Π^0_1-complete [7, 29]. (Buszkowski proved the lower bound and Palka proved the upper bound.) In contrast, by Kozen's completeness result we have that Kleene algebra and star-continuous Kleene algebra give rise to the same equational theory, which is PSPACE-complete. This matter remains open for Kleene lattices since Buszkowski's lower bound does not apply.

Residuals and meets naturally correspond to linear implication and additive conjunction [20, 29], from (non-commutative intuitionistic) linear logic [17]. They are also essential connectives in the Lambek-calculus and related substructural logics [28]. We extend LKA accordingly into a system LAL and obtain the results summarised in Fig. 1 (right): LAL is complete for star continuous action lattices (Thm. 35); it still admits cut-elimination (Thm. 37); thus it is also sound w.r.t. star continuous action lattices (Thms. 32). Furthermore we are able to show that its regular fragment with cut is in fact sound and complete for all action lattices (Thm. 33); this somewhat surprising result gives us a nontrivial yet finite proof theoretic representation of the theory of action lattices. The proof of soundness reasons inductively on the cycle structure of such regular proofs, and we crucially exploit the availability of both residuals and meets: for action algebra and Kleene lattices, it remains open whether the corresponding regular fragments with cut are sound.

Thms. 32, 34, and 37 are proved by extending the proofs of Thms. 11, 14, and 15 to deal with the additional connectives. Amongst those, cut-elimination is the most delicate extension, relying on higher types to interpret residuals, and proving that LAL proofs still yield terminating programs in such a setting. Thm. 13 cannot be extended directly, due to the lack of a free model analogous to the regular languages for Kleene algebra when adding residuals or meet. This is why we instead rely on cut-elimination for completeness.

As explained above, while all notions are equivalent in the case of Kleene algebra (Fig. 1 (left)), complexity arguments make it possible to separate the lower and upper parts of Fig. 1 (right), except for Kleene lattices. Whether the upper part is decidable remains open, but it is interesting to note that we managed to characterise action lattices in such a way that the non-regular/regular distinction at the proof-level corresponds precisely to the difference between the star continuous and general cases, respectively. One potentially fruitful direction towards the decidability of action lattices is to characterise the image of regular proofs under cut-elimination. We aim to explore this possibility in future work.

Related work

We briefly discussed the cut-free variant of the system LKA in [10] (with a simpler validity criterion), observing that its regular fragment is incomplete (due to the absence of cut). Our main contribution was a variant of it based on 'hypersequents', HKA, whose regular fragment is sound and complete without cut, and admits a PSPACE proof search procedure.

Palka proposed a sequent system for star continuous action lattices in [29], for which she proved cut-elimination. Its non-star rules are precisely those of LKA, but the system is wellfounded and relies on an 'omega-rule' for Kleene star with infinitely many premisses, in the traditional school of infinitary proof theory, cf. [33]. Such an approach does not admit a notion of finite proof analogous to our regular proofs, corresponding to the upper parts of Fig. 1. Wurm also proposed a (finite, and thus wellfounded) sequent system for Kleene algebra [34]. Unfortunately his cut-admissibility theorem does not hold – see [10].

The normalisation theory of *linear logic* with (least and greatest) fixed point operators has been studied in [14] and, more comprehensively, in [13]. While the latter is a rather general framework, its exposition still differs significantly from the current work for various reasons. One immediate difference is that their setting is commutative while ours is non-commutative, and so those results are not directly applicable. A more important difference is that they do not have any atoms in their language, reasoning only on closed formulae. This is rather significant from the point of view of normalisation, since the convergence of cut-elimination becomes more complicated in presence of atoms. The argument we give in Sect. 4 uses different ideas that are closely related to the language-based models of our algebras and the natural interpretation of language inclusions as programs [18]. A *game semantics* approach to cut-elimination for non-wellfounded proofs is given in [8], though in that work only finitely many cuts in a proof are considered and so it does not seem sufficient to handle the star rules in this work.

2 Preliminaries on Kleene algebra and extensions

Let A be a finite *alphabet*. *Regular expressions* [21] are generated as follows:

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

Sometimes we may simply write ef instead of $e \cdot f$. Each expression e generates a rational language $\mathcal{L}(e) \subseteq A^*$, defined in the usual way.

A *Kleene algebra* is a tuple $(K, 0, 1, +, \cdot, ^*)$ where $(K, 0, 1, +, \cdot)$ is an idempotent semiring and where the following properties hold, where $x \leq y$ is a shorthand for $x + y = y$.

$$1 + xx^* \leq x^* \qquad \text{if } xy \leq y \text{ then } x^*y \leq y \qquad \text{if } yx \leq y \text{ then } yx^* \leq y \tag{1}$$

There are several equivalent variants of this definition [9]. Intuitively we have that x^*y (resp., yx^*) is the least fixpoint of the function $z \mapsto y + xz$ (resp., $z \mapsto y + zx$). We write $\mathrm{KA} \vDash e \leq f$ if the inequality $e \leq f$ holds universally in all Kleene algebras – or, equivalently, if it is derivable from the axioms of Kleene algebra. Kozen [24] and Krob [27] showed that this axiomatisation is complete for language inclusions, corresponding to the right-to-left implication in the following characterisation (the other direction is routine).

▶ **Theorem 1** ([24, 27]). $\mathrm{KA} \vDash e \leq f$ *if and only if* $\mathcal{L}(e) \leq \mathcal{L}(f)$.

A Kleene algebra is *star-continuous* if for all elements x, y, z, xy^*z is the least upper bound of the sequence $(xy^iz)_{i \in \mathbb{N}}$, where $y^0 = 1$ and $y^{i+1} = yy^i$. In presence of the other laws, star-continuity is equivalent to the following condition:

$$\forall xyzt, \ (\forall i \in \mathbb{N}, \ xy^iz \leq t) \Rightarrow xy^*z \leq t \ .$$

We write $\mathrm{KA}^* \vDash e = f$ when the equality $e = f$ holds in all star-continuous Kleene algebras. Formal languages form a star-continuous Kleene algebra, and so by completeness of Kleene algebra w.r.t. rational languages, we have $\mathrm{KA}^* \vDash e = f$ iff $\mathrm{KA} \vDash e = f$; this is the triangle on the left in Fig. 1.

$$cut\ \frac{\Delta \to e \quad \Gamma, e, \Sigma \to f}{\Gamma, \Delta, \Sigma \to f} \qquad id\ \frac{}{e \to e} \qquad 0\text{-}l\ \frac{}{\Gamma, 0, \Delta \to e} \qquad 1\text{-}l\ \frac{\Gamma, \Delta \to e}{\Gamma, 1, \Delta \to e} \qquad 1\text{-}r\ \frac{}{\to 1}$$

$$\cdot\text{-}l\ \frac{\Gamma, e, f, \Delta \to g}{\Gamma, e \cdot f, \Delta \to g} \qquad +\text{-}l\ \frac{\Gamma, e, \Delta \to g \quad \Gamma, f, \Delta \to g}{\Gamma, e + f, \Delta \to g} \qquad *\text{-}l\ \frac{\Gamma, \Delta \to f \quad \Gamma, e, e^*, \Delta \to f}{\Gamma, e^*, \Delta \to f}$$

$$\cdot\text{-}r\ \frac{\Gamma \to e \quad \Delta \to f}{\Gamma, \Delta \to e \cdot f} \qquad +\text{-}r_i\ \frac{\Gamma \to e_i}{\Gamma \to e_1 + e_2}\ i \in \{1, 2\} \qquad *\text{-}r_1\ \frac{}{\to e^*} \qquad *\text{-}r_2\ \frac{\Gamma \to e \quad \Delta \to e^*}{\Gamma, \Delta \to e^*}$$

Figure 2 The rules of LKA.

An *action lattice* is a Kleene algebra with three additional binary operations, left and right *residuals* $(\backslash, /)$, and *meet* $(\cap)$ defined by the following equivalences:

$$\forall xyz,\ \ y \leq x\backslash z \ \Leftrightarrow\ xy \leq z \ \Leftrightarrow\ x \leq z/y \qquad \text{and} \qquad \forall xyz,\ \ z \leq x \cap y \ \Leftrightarrow\ z \leq x \wedge z \leq y$$

An *action algebra* is a Kleene algebra with residuals, a *Kleene lattice* is a Kleene algebra with meets. We extend regular expressions accordingly, writing $\mathrm{AL} \vDash e \leq f$ when the inequation $e \leq f$ holds in all action lattices, and $\mathrm{AL}^* \vDash e \leq f$ when it holds in all star continuous action lattices. Note that while rational languages are closed under residuals and intersection, thus forming an action lattice, they are not the 'free' one: Thm. 1 fails. The equational theories generated by all action lattices and by the star-continuous ones actually differ, cf. [7, 29].

3 The sequent system LKA

A *sequent* is an expression $\Gamma \to e$, 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*, or simply the 'left' and 'right' hand sides, respectively. We say that a sequent $e_1, \ldots, e_n \to e$ is *valid* if $\mathrm{KA}^* \vDash e_1 \cdot \cdots \cdot e_n \leq e$. I.e., the comma is interpreted as sequential composition, and the sequent arrow as inclusion. We may refer to expressions as 'formulae' when it is more natural from a proof theoretic perspective, e.g. 'subformula' or 'principal formula'.

The rules of LKA are given in Fig. 2. We call LKA^- the subset of LKA where the *cut* rule is omitted (which corresponds to the system called LKA in [10]). Leaving the $*$-rules aside, these rules are those of the non-commutative variant of intuitionistic linear logic [17], 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 understood as those arising from the characterisation of e^* as a fixed point: $e^* = \mu x.(1 + ex)$. In contrast, Palka [29] follows the alternative interpretation of Kleene star as an infinite sum, $e^* = \Sigma_i e^i$, whence her left rule for Kleene star with infinitely many premisses, and the infinitely many right rules she uses for this operation.

As previously mentioned, we consider infinitely deep proofs, so it is necessary to impose a validity criterion to ensure that derivations remain sound.

▶ **Definition 2.** A (binary, possibly infinite) *tree* is a prefix-closed subset of $\{0, 1\}^*$, which we view with the root, ε, at the bottom; elements of $\{0, 1\}^*$ are called *nodes*. A *preproof* is a labelling π of a tree by sequents such that, for every node v with children $v_1, \ldots v_n$ $(n = 0, 1, 2)$, the expression $\dfrac{\pi(v_1)\ \cdots\ \pi(v_n)}{\pi(v)}$ is an instance of an LKA rule. Given a node v in a preproof π, we write π_v for the sub-preproof rooted at v, defined by $\pi_v(w) = \pi(vw)$. A

preproof is *regular* if it has finitely many distinct subtrees, i.e. it can be expressed as the infinite unfolding of a finite graph. A preproof is *cut-free* if it does not use the *cut*-rule.

We will use standard proof theoretic terminology about *principal formulas* and *ancestry* in proofs, e.g. from [6]. (see [11, App. A] for further details). The notion of validity below is similar to [13], adapted to our setting.

▶ **Definition 3.** A *thread* is a maximal path through the graph of (immediate) ancestry in a preproof. By definition it must start at a conclusion formula or at a cut formula and it only goes upwards. A thread is *valid* if it is principal for a $*$-l step infinitely often. A preproof is *valid* if every infinite branch eventually has a valid thread. A *proof* is a valid preproof. We write $\mathrm{LKA} \vdash^{\infty} \Gamma \to e$ if the sequent $\Gamma \to e$ admits a proof, $\mathrm{LKA} \vdash^{\omega} \Gamma \to e$ if it admits a regular proof, and $\mathrm{LKA}^{-} \vdash^{\infty} \Gamma \to e$ if it admits a cut-free proof.

Notice that every valid thread eventually follows a unique (star) formula, by the subformula property. Let us consider some examples of (pre)proofs. In all cases, we will use the symbol $\bullet$ to indicate circularities (i.e. to identify roots of the same subtree), colours to mark some of the threads, and double lines to denote finite derivations.

▶ **Example 4.** Here is a regular and cut-free proof of $(b+c)^* \to (c+b)^*$:

$$\dfrac{*\text{-}r_1\dfrac{}{\to (c+b)^*} \quad *\text{-}r_2\dfrac{\dfrac{}{b+c \to c+b} \quad *\text{-}l\dfrac{\vdots}{(b+c)^* \to (c+b)^*}\bullet}{b+c, (b+c)^* \to (c+b)^*}}{*\text{-}l\dfrac{}{(b+c)^* \to (c+b)^*}\bullet}$$

▶ **Example 5** (Atomicity of identity)**.** As in many common sequent systems, initial identity steps can be reduced to atomic form, although for this we crucially rely on access to non-wellfounded (yet regular) proofs. As usual, we proceed by induction on the size of an identity step, whence the crucial case is for the Kleene star,

$$\dfrac{*\text{-}r_1\dfrac{}{\to e^*} \quad *\text{-}r_2\dfrac{\overset{IH}{\triangledown} \quad *\text{-}l\dfrac{\vdots}{e^* \to e^*}\bullet}{e, e^* \to e^*}}{*\text{-}l\dfrac{}{e^* \to e^*}\bullet}$$

where the derivation marked *IH* is obtained by the inductive hypothesis.

Note that while LKA^{-} satisfies the subformula property, the size and number of sequents occurring in a cut-free proof is not a priori bounded, due to the $*$-l rule:

▶ **Example 6** (A non-regular proof)**.** The only cut-free proof of the sequent $a, a^* \to a^*a$ is the one on the left below:

$$\dfrac{\dfrac{}{a \to a^*a} \quad *\text{-}l\dfrac{\dfrac{}{a, a \to a^*a} \quad *\text{-}l\dfrac{\vdots}{a, a, a, a^* \to a^*a}}{a, a, a^* \to a^*a}}{*\text{-}l\dfrac{}{a, a^* \to a^*a}} \qquad \dfrac{\dfrac{}{a \to a^*a} \quad cut\dfrac{*\text{-}l\dfrac{}{a, a^* \to a^*a}\bullet \quad \dfrac{}{a, a^*a \to a^*a}}{a, a, a^* \to a^*a}}{*\text{-}l\dfrac{}{a, a^* \to a^*a}\bullet}$$

This proof contains all sequents of the form $a, \ldots, a, a^* \to a^*a$, whence non-regularity. A regular proof with cuts is given on the right above; see [10] for more details on the lack of regularity in LKA^{-} and how to recover regularity in a cut-free setting, using 'hypersequents'.

▶ **Example 7** (Two invalid preproofs)**.** The following preproofs are not valid; they derive invalid sequents.

$$\dfrac{\dfrac{}{\to 1}\,{\scriptstyle 1\text{-}r} \qquad \dfrac{\vdots}{a \to 1^*}\,{\scriptstyle *\text{-}r_2}}{a \to 1^*}\,{\scriptstyle *\text{-}r_2} \qquad\qquad \dfrac{\dfrac{}{\to b^*} \qquad \dfrac{\dfrac{\dfrac{}{a \to a}\,{\scriptstyle id} \qquad \dfrac{}{a^* \to a^*}\,{\scriptstyle id}}{a, a^* \to a^*}\,{\scriptstyle *\text{-}r_2} \qquad \dfrac{\vdots}{a^* \to b^*}\,{\scriptstyle *\text{-}l}}{a, a^* \to b^*}\,{\scriptstyle cut}}{a^* \to b^*}\,{\scriptstyle *\text{-}l}$$

The left preproof is cut-free and infinite; since it does not contain any $*$-l-rule, it cannot be valid. On the right the principal formula of the $*$-l-rule is the cut formula of the *cut*-rule so that the only infinite thread is the one along the occurrences of b^*, and this formula is never principal for a $*$-l step.

The notion of validity we use here actually generalises the notion of *fairness* we used in [10], where we were working only with cut-free preproofs:

▶ **Proposition 8.** *A cut-free preproof is valid if and only if it is* fair *for $*$-l, i.e. every infinite branch contains infinitely many occurrences of $*$-l.*

Proof sketch. The left-right implication is immediate. Conversely, every infinite path in a fair cut-free preproof has infinitely many $*$-l steps, but there are only finitely many possible principal formulae by the subformula property. One can thus extract a valid thread. ◀

An alternative criterion for cut-free preproofs is obtained as follows:

▶ **Proposition 9.** *A cut-free preproof is valid if and only if it has no infinite branch with a tail of only $*$-r_2-steps.*

Proof. Define the 'weight' of a sequent to be the multiset of its formulae, ordered by the subformula relation. This measure strictly decreases when reading LKA^- rules bottom-up, except for the right premisses of rules $*$-l and $*$-r_2; for the latter, it either remains unchanged (when Γ is empty) or it strictly decreases. Thus every infinite branch of a cut-free preproof either contains infinitely many $*$-l steps, or eventually contains only $*$-r_2 steps. ◀

Observe that the proof on the left in Ex. 7 does not satisfy this condition.

The cut-free system LKA^- is sound and complete for Kleene algebras. Thanks to the completeness theorem for Kleene algebras, Thm. 1, it suffices to prove soundness with respect to star-continuous Kleene algebra. We first prove the following lemma:

▶ **Lemma 10.** *If $\mathrm{LKA}^- \vdash^\infty \Gamma, e^*, \Delta \to f$ then, for each $n \in \mathbb{N}$, $\mathrm{LKA}^- \vdash^\infty \Gamma, e^n, \Delta \to f$.*

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

$$\dfrac{\Gamma, \Delta \to f \quad \Gamma, e, e^*, \Delta \to f}{\Gamma, e^*, \Delta \to f}\,{\scriptstyle *\text{-}l} \qquad \mapsto \qquad \dfrac{\Gamma, \Delta \to f}{\Gamma, 1, \Delta \to f}\,{\scriptstyle 1\text{-}l} \quad \text{or} \quad \dfrac{\Gamma, e, e^{n-1}, \Delta \to f}{\Gamma, e^n, \Delta \to f}\,{\scriptstyle \cdot\text{-}l}$$

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 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. ◀

We can now prove soundness w.r.t. star continuous Kleene algebra:

▶ **Theorem 11** (Soundness). *If* $\mathrm{LKA}^- \vdash^\infty e_1, \dots, e_n \to e$, *then* $\mathrm{KA}^* \vDash e_1 \cdot \dots \cdot e_n \leq e$.

Proof. First observe that every rule of LKA is sound: if its premisses are valid then so is its conclusion. Let π be an LKA^- proof of $\Sigma \to f$. We proceed by structural induction on the multiset of formulae in its conclusion, via case analysis on the last rule. For all but two cases, we just use soundness of the rule and the induction hypotheses. The first remaining case is $*\text{-}r_2$, where we must appeal to a sub-induction since the measure does not always strictly decrease in the right premiss (Prop. 9). The last case is $*\text{-}l$, where $\Sigma = \Gamma, e^*, \Delta$. By Lem. 10, π can be transformed into proofs π_n of $\Gamma, f^n, \Delta \to f$ for each $n \in \mathbb{N}$. Each π_n derives a sequent whose weight is strictly smaller than that of $\Sigma \to f$, which is thus valid by the inductive hypothesis. Finally, this means that $\Gamma, f^*, \Delta \to g$ is valid, by star-continuity. ◀

For completeness of LKA^-, we can get a direct proof by starting from the free model of rational languages (Fig. 1). This strategy is no longer possible for Kleene lattices, action algebras and action lattices, for which we will need to go through cut-elimination. We first prove completeness for sequents whose antecedent is a word:

▶ **Lemma 12.** *If* $a_1 \dots a_n$ *is a word in* $\mathcal{L}(e)$ *for some expression* e, *then there is a finite proof of the sequent* $a_1, \dots, a_n \to e$ *using only right logical rules.*

Proof. By a straightforward induction on e. ◀

▶ **Theorem 13** (Completeness). *If* $\mathcal{L}(e_1 \cdot \dots \cdot e_n) \subseteq \mathcal{L}(e)$ *then* $\mathrm{LKA}^- \vdash^\infty e_1, \dots, e_n \to e$.

Proof. This is proved like in [10] for HKA: all left rules of LKA^- are invertible so that they can be applied greedily; doing so, one obtains an infinite tree whose leaves are sequents of the shape $a_1, \dots, a_k \to e$, with $k \geq 0$, where $a_1 \dots a_k$ is a word in $\mathcal{L}(e_1 \cdot \dots \cdot e_n)$ and thus in $\mathcal{L}(e)$ by assumption. Those leaves can be replaced by finite derivations using by Lem. 12. Notice, that we obtain fairness, since any infinite branch of only left rules must contain $*\text{-}l$ infinitely often. ◀

The previous proof builds infinite and non-regular derivations whenever the language of the starting antecedent is infinite. For instance, it would yield the proof given on the left in Ex. 6. By using a different technique, we show in the following theorem, that we can get regular proofs if we allow the cut-rule.

▶ **Theorem 14** (Regular completeness). *If* $\mathrm{KA} \vDash e \leq f$ *then* $\mathrm{LKA} \vdash^\omega e \to f$.

Proof. We prove the statement for equalities. Consider the relation $\equiv$ defined by $e \equiv f$ if $\mathrm{LKA} \vdash^\omega e \to f$ and $\mathrm{LKA} \vdash^\omega f \to e$. This relation is an equivalence on regular expressions thanks to the cut rule, and it is easily shown to be preserved by all contexts (i.e. it is a congruence). Also remark that we have $e + f \equiv f$ iff $\mathrm{LKA} \vdash^\omega e \to f$, thanks to the cut-rule and the rules about sum. It then suffices to show that regular expressions quotiented by $\equiv$ form a Kleene algebra. The (in)equational axioms defining KA can be proved by finite derivations. The only difficulty is in dealing with the two implications from the definition of Kleene algebra (1). We implement them as follows:

$$\dfrac{\dfrac{}{f \to f}\,id \qquad \dfrac{\dfrac{\vdots}{e^*, f \to f}\,{*\text{-}l} \bullet \qquad \overset{IH}{\triangledown}\ e, f \to f}{e, e^*, f \to f}\,cut}{e^*, f \to f}\,{*\text{-}l} \bullet$$

$$\dfrac{\dfrac{}{f \to f}\,id \qquad \dfrac{\overset{IH}{\triangledown}\ f, e \to f \qquad \dfrac{\vdots}{f, e^* \to f}\,{*\text{-}l} \bullet}{f, e, e^* \to f}\,cut}{f, e^* \to f}\,{*\text{-}l} \bullet$$

where the derivations marked *IH* are obtained from the inductive hypothesis. The preproofs we construct in this way are valid and regular, by inspection. In particular, the only infinite branch not in *IH* in the above derivations has a valid thread on e^*, coloured in green. ◀

Note the asymmetry when we interpret the two implications: the premisses of the cut rule are swapped when we move from one to the other. This asymmetry comes from the fact that we have a single left rule for Kleene star, which unfolds the star from the left.

4 Cut-elimination for LKA

This section is devoted to proving the following cut-elimination theorem.

▶ **Theorem 15.** *If* $\mathrm{LKA} \vdash^{\infty} \Gamma \to e$ *then* $\mathrm{LKA}^{-} \vdash^{\infty} \Gamma \to e$.

Combined with Thm. 11, it establishes the soundness of our criterion for proofs with cuts. This serves as a 'warm-up' for the analogous result for the extended system (Sect. 6), which is obtained using the same template.

We show that proofs can be considered as certain *transducers*, transforming *parse-trees* of input words of languages computed by terms. We design them so that a given computation only explores a finite prefix of the proof, which we call the *head*. We then prove that cut-reductions, restricted to the head of a proof, preserve these computations, always terminate, and eventually produce some non-cut rules. We can then repeatedly apply this procedure to remove all cuts from an infinite proof, in a productive way.

4.1 Programs from proofs

We first define programs and their reduction semantics, based on which we prove cut-elimination, in Sect. 4.2. We fix in this section a (valid) LKA proof π and we let v range over its nodes, which we recall are elements of $\{0,1\}^*$, cf. Dfn. 2.

▶ **Definition 16** (Programs). *Programs* are defined by the following syntax, where x ranges over a countable set of *variables*, and i ranges over $\{1,2\}$.

$$M, N ::= x \mid \star \mid \langle M, N\rangle \mid \mathrm{in}_i M \mid [] \mid M :: N \mid v[\vec{M}]$$

Intuitively, programs compute parse-trees for words belonging to the language of an expression. Given a node v of π such that $\pi(v) = \Gamma \to e$, the last entry corresponds to the application of the subproof π_v, rooted at v, to a list $\vec{M}$ of programs for the antecedent (Γ); it should eventually return a parse-tree for the succedent (e). This is formalised using the following notion of types.

▶ **Definition 17** (Typing environment). A *typing environment*, written E, is a list of pairs of variables and expressions (written $x : e$), together with a finite antichain of nodes: for any two nodes v and w in the antichain, v is not a prefix of w. We write E, F for the concatenation of two typing environments, which is defined only when this antichain condition on nodes is preserved.

Intuitively, typing environments keep track of which variables and proof nodes are used in a program, to impose linearity constraints; these constraints become crucial when we add residuals and meets, in Sect. 5.

▶ **Definition 18** (Types). A program M has type e in an environment E, written $E \vdash M : e$ if this judgement can be derived from the rules in Fig. 3.

$$\frac{}{x : e \vdash x : e} \qquad \frac{}{\vdash \star : 1} \qquad \frac{E \vdash M : e \qquad F \vdash N : f}{E, F \vdash \langle M, N \rangle : ef} \qquad \frac{E \vdash M : e_i}{E \vdash \mathrm{in}_i M : e_1 + e_2} \qquad \frac{}{\vdash [] : e^*}$$

$$\frac{E \vdash M : e \qquad E' \vdash N : e^*}{E, E' \vdash M :: N : e^*} \qquad \frac{\forall i, E_i \vdash M_i : e_i \qquad \pi(v) = e_1, \ldots, e_n \to f}{v, E_1, \ldots, E_n \vdash v[\vec{M}] : f}$$

Figure 3 Typing rules for programs.

▶ **Example 19.** With the proof from Ex. 4, letting ε denote the root node, we have:

$$\varepsilon, x : b, y : c \vdash \varepsilon[\mathrm{in}_0 x :: \mathrm{in}_1 y :: []] : (c + b)^*$$
$$\varepsilon, z : b + c, z' : b + c, q : (b + c)^* \vdash \varepsilon[z :: z' :: q] : (c + b)^*$$

▶ **Observation 20.** *Let $x_1, \ldots, x_n$ be variables. We have $a_1 \ldots a_n \in \mathcal{L}(e)$ iff there exists a program M such that $x_1 : a_1, \ldots, x_n : a_n \vdash M : e$. This (unused) observation has no counterpart when considering extensions of Kleene algebra, where there is no longer an appropriate notion of 'language' for expressions that constitutes a free model.*

▶ **Definition 21** (Reduction)**.** *Reduction*, written $\rightsquigarrow$, is the closure under all contexts of the following rules defined by case analysis on the last step of the subproof π_v rooted at v. These rules are written concisely for lack of space; in each case, $v0$ and $v1$ are the nodes of the premisses, when they exist. We moreover assume that the sizes of the vectors match those that arise from the various rules. See [11, App. B] for an extensive definition.

$$\begin{aligned}
&id : v[M] \rightsquigarrow M && cut : v[\vec{M}, \vec{N}, \vec{P}] \rightsquigarrow v1[\vec{M}, v0[\vec{N}], \vec{P}] \\
&1\text{-}l : v[\vec{M}, \star, \vec{N}] \rightsquigarrow v0[\vec{M}, \vec{N}] && 1\text{-}r : v[] \rightsquigarrow \star \\
&\cdot\text{-}l : v[\vec{M}, \langle M, N \rangle, \vec{N}] \rightsquigarrow v0[\vec{M}, M, N, \vec{N}] && \cdot\text{-}r : v[\vec{M}, \vec{N}] \rightsquigarrow \langle v0[\vec{M}], v1[\vec{N}] \rangle \\
&+\text{-}l : v[\vec{M}, \mathrm{in}_i M, \vec{N}] \rightsquigarrow vi[\vec{M}, M, \vec{N}] && +\text{-}r_i : v[\vec{M}] \rightsquigarrow \mathrm{in}_i(v0[\vec{M}]) \\
&*\text{-}l : v[\vec{M}, [], \vec{N}] \rightsquigarrow v0[\vec{M}, \vec{N}] \text{ and} && *\text{-}r_1 : v[] \rightsquigarrow [] \\
&\quad v[\vec{M}, M :: N, \vec{N}] \rightsquigarrow v1[\vec{M}, M, N, \vec{N}] && *\text{-}r_2 : v[\vec{M}, \vec{N}] \rightsquigarrow v0[\vec{M}] :: v1[\vec{N}]
\end{aligned}$$

When useful we write, say, $\rightsquigarrow_{cut}$ to indicate a reduction according to the *cut* rule above.

▶ **Example 22.** Continuing with the proof from Ex. 4 we have the following complete reductions. The second program still contains calls to proofs in the end because the inputs were under-specified.

$$\begin{aligned}
&\varepsilon[\mathrm{in}_0 x :: \mathrm{in}_1 y :: []] \\
\rightsquigarrow\ & 1[\mathrm{in}_0 x, \mathrm{in}_1 y :: []] \\
\rightsquigarrow\ & 10[\mathrm{in}_0 x] :: 11[\mathrm{in}_1 y :: []] \\
\rightsquigarrow\ & 100[x] :: 11[\mathrm{in}_1 y :: []] \\
\rightsquigarrow\ & \mathrm{in}_1 x :: 11[\mathrm{in}_1 y :: []] \\
\rightsquigarrow\ & \mathrm{in}_1 x :: 111[\mathrm{in}_1 y, []] \\
\rightsquigarrow\ & \mathrm{in}_1 x :: 1110[\mathrm{in}_1 y] :: 1111[[]] \\
\rightsquigarrow^4\ & \mathrm{in}_1 x :: \mathrm{in}_0 y :: []
\end{aligned}$$

$$\begin{aligned}
&\varepsilon[z :: z' :: q] \\
\rightsquigarrow\ & 1[z, z' :: q] \\
\rightsquigarrow\ & 10[z] :: 11[z' :: q] \\
\rightsquigarrow\ & 10[z] :: 111[z', q] \\
\rightsquigarrow\ & 10[z] :: 1110[z'] :: 1111[q]
\end{aligned}$$

As one might expect, we have subject reduction. We need the following notion of *extension* to state it properly.

▶ **Definition 23** (Extension). Given two typing environments E, E', we say that E' *extends* E if E and E' coincide after removing all nodes, and if all nodes in E' are either already in E, or are immediate successors of some nodes in E.

▶ **Proposition 24** (Subject reduction). *If $E \vdash M : e$ and $M \rightsquigarrow M'$, then $E' \vdash M' : e$ for some environment E' extending E.*

For instance, along the reductions on the left in Ex. 22, the antichain part of the typing environment evolves as follows: $\{\varepsilon\}, \{1\}, \{10, 11\}, \{100, 11\}, \{11\}, \{111\}, \{1110, 1111\}, \emptyset$.

Our objective now is to prove that well-typed programs terminate. For the sake of simplicity, we work in the sequel with the 'leftmost innermost' strategy: a redex $v[\vec{M}]$ is fired only when the programs in $\vec{M}$ are irreducible and there are no other redexes to its left.

▶ **Definition 25** (Runs). The *run* of a program M is the sequence of nodes corresponding to the redexes fired during the (potentially infinite) leftmost innermost reduction of M.

▶ **Lemma 26.** *If $E \vdash M : e$ then every node w appears at most once in the run of M; in this case we have that $w = uv$ for some nodes u, v with u in E and, for every prefix v' of v, uv' appears in the run of M before w.*

Proof. These are immediate consequences of Prop. 24. ◀

In particular, the run of a well-typed program has finitely many connected components. We finally obtain that well typed programs terminate, thanks to the validity criterion.

▶ **Proposition 27.** *If $E \vdash M : e$, then the run of M is finite.*

Proof. Suppose the run of M is infinite. Then by Lem. 26 and König's Lemma one can extract an infinite branch of π which is contained in the run. By validity, this branch must eventually have a thread along a star formula f^* which is infinitely often principal. By analysis of the reduction rules, and thanks to the innermost strategy, we may find an infinite sequence of programs of type f^* whose sizes are strictly decreasing, which is impossible. ◀

4.2 Cut reduction

Our cut-elimination argument is driven by a standard set of cut reduction rules, which we do not have space to present in the main text. These include key and commutative cases, as usual, and are fully presented in [11, App. D]. To produce an infinite cut-free proof, we must show that we may produce proofs with arbitrarily large cut-free prefixes in a continuous manner. The main difficulty is to show that such a procedure is *productive*, i.e., eventually produces non-cut steps. To this end, we use the previous notion of 'run' to drive cut-reductions.

▶ **Definition 28** (Head). Let π be a proof of $\Gamma \to e$. The *head* of π, written $hd(\pi)$, is the run of the program $\varepsilon[\vec{x}]$ in π, where $\vec{x}$ is a list of variables of the same length as Γ.

Note that the above program is well-typed in the appropriate environment. The head is a sequence of nodes, but we shall sometimes see it as the underlying sequence of programs. Also note that the nodes of a cut step appearing in the head correspond to program reductions where the redex is a cut ($\rightsquigarrow_{cut}$).

▶ **Definition 29** (Weight). The *weight* of a proof π, written $w(\pi)$, is the multiset of cut-reductions in its head, ordered by their distance to the end of the head.

▶ **Lemma 30.** *Let π' be obtained from a proof π by a cut-reduction. We have that:*
(i) *π' is a valid proof;*
(ii) *$|hd(\pi')| \leq |hd(\pi)|$, where $|s|$ is the length of a sequence s;*
(iii) *if the reduced cut was the last $\rightsquigarrow_{cut}$ step in $hd(\pi)$, then $w(\pi') < w(\pi)$.*

Proof sketch. By case analysis; key cases strictly decrease the length of the head while it is only conserved by commutative cases. We list and discuss all cases in [11, App. D]; one of the two $*$-key cases is the following one:

$$\dfrac{\dfrac{\Delta \to e \quad \Sigma \to e^*}{\Delta, \Sigma \to e^*}\,{}_{*\text{-}r_2} \quad \dfrac{\Gamma, \Pi \to f \quad \Gamma, e, e^*, \Pi \to f}{\Gamma, e^*, \Pi \to f}\,{}_{*\text{-}l}}{\Gamma, \Delta, \Sigma, \Pi \to f}\,{}_{cut} \quad \mapsto \quad \dfrac{\Delta \to e \quad \dfrac{\Sigma \to e^* \quad \Gamma, e, e^*, \Pi \to f}{\Gamma, e, \Sigma, \Pi \to f}\,{}_{cut}}{\Gamma, \Delta, \Sigma, \Pi \to f}\,{}_{cut}$$

If the reduced cut (with conclusion at v) occurs in the head of π then the heads of the two proofs only differ by the following subsequences, inside some evaluation context:

$$\begin{array}{ll} & v[\vec{M}, \vec{N}, \vec{O}, \vec{P}] \\ \rightsquigarrow_{cut_{e^*}} & v1[\vec{M}, v0[\vec{N}, \vec{O}], \vec{P}] \\ \rightsquigarrow & v1[\vec{M}, v00[\vec{N}] :: v01[\vec{O}], \vec{P}] \\ \rightsquigarrow^n & v1[\vec{M}, N' :: v01[\vec{O}], \vec{P}] \\ \rightsquigarrow^o & v1[\vec{M}, N' :: O', \vec{P}] \\ \rightsquigarrow & v11[\vec{M}, N', O', \vec{P}] \end{array} \qquad \begin{array}{ll} & v[\vec{M}, \vec{N}, \vec{O}, \vec{P}] \\ \rightsquigarrow_{cut_e} & v1[\vec{M}, v0[\vec{N}], \vec{O}, \vec{P}] \\ \rightsquigarrow^n & v1[\vec{M}, N', \vec{O}, \vec{P}] \\ \rightsquigarrow_{cut_{e^*}} & v11[\vec{M}, N', v10[\vec{O}], \vec{P}] \\ \rightsquigarrow^o & v11[\vec{M}, N', O', \vec{P}] \end{array}$$

(Note that the programs $\vec{M}, \vec{N}, \vec{O}, \vec{P}$ are irreducible due to the innermost strategy, and that $v00$ in the starting proof and $v0$ in the resulting one both point to the same subproof: $\pi_{v00} = \pi'_{v0}$.) The new head is shorter by one step, and the initial cut on e^* is replaced by two cuts which are closer to the end of the head.

Commutative cases do not always shorten the head, but either they move the cut closer to its end, or the head no longer visits it. For instance, when the left premiss of the reduced cut ends with a $\cdot$-l step, the rule is:

$$\dfrac{\dfrac{\dfrac{\Delta, e, f, \Sigma \to g}{\Delta, ef, \Sigma \to g}\,{}_{\cdot\text{-}l} \quad \Gamma, g, \Pi \to h}{\Gamma, \Delta, ef, \Sigma, \Pi \to h}\,{}_{cut} \quad \mapsto \quad \dfrac{\Delta, e, f, \Sigma \to g \quad \Gamma, g, \Pi \to h}{\Gamma, \Delta, e, f, \Sigma, \Pi \to h}\,{}_{cut}}{\Gamma, \Delta, ef, \Sigma, \Pi \to h}\,{}_{\cdot\text{-}l}$$

If the head of π goes through the step $v[\vec{M}, \vec{N}, O, \vec{P}, \vec{Q}] \rightsquigarrow_{cut_{ef}} v1[\vec{M}, v0[\vec{N}, O, \vec{P}], \vec{Q}]$, then there are two cases to consider:

- either $O = \langle O_1, O_2 \rangle$ and the sequence continues with $v1[\vec{M}, v00[\vec{N}, O_1, O_2, \vec{P}], \vec{Q}]$; then in π' we get $v[\vec{M}, \vec{N}, O, \vec{P}, \vec{Q}] \rightsquigarrow v0[\vec{M}, \vec{N}, O_1, O_2, \vec{P}, \vec{Q}] \rightsquigarrow_{cut_{ef}} v01[\vec{M}, v00[\vec{N}, O_1, O_2, \vec{P}], \vec{Q}]$; the length is preserved and the cut has been pushed towards the end;
- or not, and the head of π' stops earlier, without visiting the cut on ef anymore, thus decreasing the weight.

For (iii), the assumption that the cut-reduction took place on the last cut of the head is used in some of the cases to ensure that the weights of the other cuts in the head do not increase (e.g., in some of the right $\cdot$-r and $*$-r_2 cases). ◀

▶ **Proposition 31** (Productive cut-reduction)**.** *For a proof π, there exists a proof π' obtained from π by a sequence of cut-reductions, which does not start with a cut.*

Proof. By induction on the weight, reduce the last cut visited by the head until the head no longer contains any cut. The resulting proof cannot start with a cut, by definition. ◀

We can finally prove cut-elimination.

Proof of Thm. 15. Focus on a lowest cut, at node v, and apply Prop. 31 to the corresponding subproof π_v. By iterating this process, we obtain in the limit a cut-free preproof π' with the same conclusion as the starting one. Moreover, thanks to Lem. 30.(i), all heads of subproofs of π' are finite, so that π' is valid by Prop. 9: an infinite branch of $*$-r_2 steps would give rise to a subproof with an infinite head. ◀

5 Action algebras, Kleene lattices, and action lattices

We now consider extensions of Kleene algebra by residuals and meets, as axiomatised in [31] and [25]. We first extend the system LKA with the following rules, which are standard from substructural logic [28, 16]. We write LAL for the corresponding system.

$$\backslash\text{-}l\ \frac{\Delta \to e \quad \Gamma, f, \Sigma \to g}{\Gamma, \Delta, e\backslash f, \Sigma \to g} \qquad /\text{-}l\ \frac{\Delta \to e \quad \Gamma, f, \Sigma \to g}{\Gamma, f/e, \Delta, \Sigma \to g} \qquad \cap\text{-}l_i\ \frac{\Gamma, e_i, \Delta \to f}{\Gamma, e_1 \cap e_2, \Delta \to f}\ i \in \{1,2\}$$

$$\backslash\text{-}r\ \frac{e, \Gamma \to f}{\Gamma \to e\backslash f} \qquad /\text{-}r\ \frac{\Gamma, e \to f}{\Gamma \to f/e} \qquad \cap\text{-}r\ \frac{\Gamma \to e \quad \Gamma \to f}{\Gamma \to e \cap f}$$

We define judgements as previously. Except for Thm. 33, the results below also hold for action algebras and Kleene lattices using the appropriate fragment of LAL. We prove soundness w.r.t. star-continuous models exactly like for Kleene algebra (Thm. 11).

▶ **Theorem 32** (Soundness). *If* $\mathrm{LAL}^- \vdash^\infty e_1, \dots, e_n \to e$, *then* $\mathrm{AL}^* \vDash e_1 \cdot \dots \cdot e_n \leq e$.

As announced in the introduction, regular proofs are sound for all (non-necessarily star-continuous) action lattices. We prove it using proof-theoretical arguments to translate every regular proof into an inductive proof from the axioms of action lattices.

▶ **Theorem 33** (Regular soundness). *If* $\mathrm{LAL} \vdash^\omega e_1, \cdots, e_n \to f$ *then* $\mathrm{AL} \vDash e_1 \cdot \dots \cdot e_n \leq f$.

Proof. We prove the statement for all regular proofs in **-normal form*, where every back-pointer points to a 'validating' $*$-l-step: every infinite branch of the starting proof has a valid thread; since the proof is regular, this thread must be infinitely often principal for $*$-l-step of some sequent of the branch; cut the infinite branch by using a backpointer the second time this sequent appears in the branch.

We proceed by induction on the number of simple cycles in such a proof π. The interesting case is when π ends with a $*$-l step that is the target of a backpointer. Colour red all ancestors of its principal formula that are the same expression, say e^*. Let $\{\Gamma_i, e^*, \Delta_i \to f_i\}_{i\in I}$ be the set of all sequents in π with e^* principal and let $\{\pi_i^l : \Gamma_i, \Delta_i \to f_i\}_{i\in I}$ and $\{\pi_i^r : \Gamma_i, e, e^*, \Delta_i \to f_i\}_{i\in I}$ be the corresponding subproofs rooted at their left and right premisses, respectively.

Define expressions $g_i = \prod \Gamma_i$, $d_i = \prod \Delta_i$, $h_i = (g_i\backslash f_i)/d_i$, and $h = \bigcap_{i\in I} h_i$. For $i \in I$, construct proofs $\pi_i^{r\prime}$ from π_i^r by replacing each e^* by h, modifying critical steps as follows:

$$*\text{-}l\ \frac{\Gamma_j, \Delta_j \to f_j \quad \Gamma_j, e, e^*, \Delta_j \to f_j}{\Gamma_j, e^*, \Delta_j \to f_j} \quad \mapsto \quad \left.\begin{array}{l} \cdot\text{-}r\ \dfrac{}{\Delta_j \to d_j} \qquad \backslash\text{-}l\ \dfrac{\cdot\text{-}r\ \dfrac{}{\Gamma_j \to g_j} \quad \dfrac{}{f_j \to f_j}}{\Gamma_j, g_j\backslash f_j \to f_j} \\ /\text{-}l\ \dfrac{}{\Gamma_j, h_j, \Delta_j \to f_j} \\ \cap\text{-}l\ \dfrac{}{\Gamma_j, h, \Delta_j \to f_j} \end{array}\right\} \rho_j$$

Note that the proofs π_i^l and $\pi_i^{r\prime}$ have fewer simple cycles than π, so that by the induction hypothesis we have that $g_i d_i \leq f_i$ and $g_i e h d_i \leq f_i$ hold universally in action lattices, for all $i \in I$. From here we deduce $1 \leq h$ and $eh \leq h$ using the laws about residuals and conjunction. Thus we have $e^* \leq h$ by star induction (1). Finally note that following the above proof ρ_j we have in action lattices that $g_j h d_j \leq f_j$ and thus $g_j e^* d_j \leq f_j$. We conclude by choosing the appropriate j such that $(\Gamma_j, \Delta_j, f_j)$ is (Γ, Δ, f). ◀

Note that we crucially rely on the presence of both residuals and meet to compute invariants for Kleene stars in the above proof, so that it does not immediately carry over to action algebras and Kleene lattices.

Conversely, the regular fragment of LAL (with cut) is complete for action lattices.

▶ **Theorem 34** (Regular completeness). *If* $\mathrm{AL} \vDash e \leq f$ *then* $\mathrm{LAL} \vdash^{\omega} e \to f$.

Proof. The axioms defining meet and residual immediately translate to finite derivations in LAL, so we may simply extend the proof of Thm. 14. ◀

Note that the regular fragment cannot be complete for star continuous models: a regular proof is a finite verifiable object and the equational theory of star-continuous action lattices is Π_1^0-hard [7]. The full, non-regular system is however complete for star-continuous models:

▶ **Theorem 35** (Star-continuous completeness). *If* $\mathrm{AL}^* \vDash e \leq f$ *then* $\mathrm{LAL} \vdash^{\infty} e \to f$.

Proof. As for Thms. 14 and 34, consider the relation $\equiv'$ defined by $e \equiv' f$ if $\mathrm{LAL} \vdash^{\infty} e \to f$ and $\mathrm{LAL} \vdash^{\infty} f \to e$. Expressions quotiented by this slightly larger relation also form an action lattice, which we prove star-continuous using the natural simulation of an ω-rule for Kleene star: combine proofs $(\pi_i)_{i\in\mathbb{N}}$ of the sequents $(\Gamma, e^i, \Delta \to f)_{i\in\mathbb{N}}$ as follows:

$$\dfrac{\pi_0\ \Gamma, e, \Delta \to f}{\Gamma, e^*, \Delta \to f}\,{}^{*\text{-}l} \qquad \dfrac{\dfrac{\pi_1\ \Gamma, e, e, \Delta \to f}{\Gamma, e, e, e^*, \Delta \to f}\,{}^{*\text{-}l} \qquad \dfrac{\pi_2\ \Gamma, e, e, \Delta \to f \quad \iddots}{\Gamma, e, e, e^*, \Delta \to f}\,{}^{*\text{-}l}}{\Gamma, e, e^*, \Delta \to f}$$

◀

The remaining property to establish is cut-elimination: combined with Thm. 32 it gives soundness of proofs with cut w.r.t. star-continuous models, and combined with Thm. 35 it gives completeness of LAL^- w.r.t. these models.

6 Cut-elimination in LAL

The main alteration to the proof for LKA is that we need a more sophisticated notion of programs. We associate linear functions to residuals, and additive pairs to meets: a program for $e \cap f$ waits to see whether the environment wants a value for e or a value for f – but not both, and reacts accordingly. We thus extend the syntax of programs (Dfn. 16) to include λ-abstractions, which will be used for residuals, and a new kind of pairs for meets.

$$M, N ::= x \mid \star \mid \langle M, N\rangle \mid \mathrm{in}_i M \mid [] \mid M :: N \mid \pi[\vec{M}] \mid \lambda x.M \mid \langle\!\langle M, N\rangle\!\rangle$$

The type system (Fig. 3) is extended by the following rules, where in the final rule, E_1 and E_2 are extensions of E (cf. Dfn. 23).

$$\frac{x : e, E \vdash M : f}{E \vdash \lambda x.M : e\backslash f} \qquad \frac{E, x : e \vdash M : f}{E \vdash \lambda x.M : f/e} \qquad \frac{E_1 \vdash M : e \qquad E_2 \vdash N : f}{E \vdash \langle\!\langle M, N\rangle\!\rangle : e \cap f}$$

▶ **Lemma 36** (Substitution lemma). *If $E \vdash N : e$ and $F, x : e, F' \vdash M : f$ with F, E, F' defined, then $F, E, F' \vdash M\{N/x\} : f$, where $M\{N/x\}$ is M with x substituted by N.*[1]

The following reductions are added, using the same conventions as in Dfn. 21:

$$\cap\text{-}l_i : v[\vec{M}, \langle\!\langle N_1, N_2 \rangle\!\rangle, \vec{P}] \rightsquigarrow v0[\vec{M}, N_i, \vec{P}] \qquad \cap\text{-}r : v[\vec{M}] \rightsquigarrow \langle\!\langle v0[\vec{M}], v1[\vec{M}] \rangle\!\rangle$$
$$\backslash\text{-}l : v[\vec{M}, \vec{N}, \lambda x.F, \vec{P}] \rightsquigarrow v1[\vec{M}, F\{v0[\vec{N}]/x\}, \vec{P}] \qquad \backslash\text{-}r : v[\vec{M}] \rightsquigarrow \lambda x.v0[x, \vec{M}]$$
$$/\text{-}l : v[\vec{M}, \lambda x.F, \vec{N}, \vec{P}] \rightsquigarrow v1[\vec{M}, F\{v0[\vec{N}]/x\}, \vec{P}] \qquad /\text{-}r : v[\vec{M}] \rightsquigarrow \lambda x.v0[\vec{M}, x]$$

One has to be careful about what we deem to be evaluation contexts: lambda abstractions and additive pairs are *not* considered evaluation contexts. This is crucial to obtain subject-reduction: otherwise some redexes duplicated by the $\cap$-r rule can be active at the same time, thus breaking the property of Lem. 26 used in our termination proof that a given node appears at most once in the run of a program.

Despite this subtlety, Prop. 24 (subject reduction) and Prop. 27 (termination) are proved for this extended system exactly as in the Kleene algebra case – see [11, App. C]. It thus remains to show that the new cut reductions do not increase the length of heads, and strictly decrease the weight (Lem. 30). The key cases are easy: they strictly decrease the length and replace the cut by smaller ones. Amongst the commutative cases, some care is required when a right introduction rule appears on the right of the cut. For instance, for meet:

$$\dfrac{\Delta \to f \qquad \dfrac{\Gamma, f, \Sigma \to e_1 \quad \Gamma, f, \Sigma \to e_2}{\Gamma, f, \Sigma \to e_1 \cap e_2}\,\cap\text{-}r}{\Gamma, \Delta, \Sigma \to e_1 \cap e_2}\,cut \quad \mapsto \quad \dfrac{\dfrac{\Delta \to f \quad \Gamma, f, \Sigma \to e_1}{\Gamma, \Delta, \Sigma \to e_1}\,cut \qquad \dfrac{\Delta \to f \quad \Gamma, f, \Sigma \to e_2}{\Gamma, \Delta, \Sigma \to e_1}\,cut}{\Gamma, \Delta, \Sigma \to e_1 \cap e_2}\,\cap\text{-}r$$

If the head of π contains the sequence,

$$v[\vec{M}, \vec{N}, \vec{O}] \rightsquigarrow v1[\vec{M}, v0[\vec{N}], \vec{O}] \rightsquigarrow^n v1[\vec{M}, N', \vec{O}] \rightsquigarrow \langle\!\langle v10[\vec{M}, N', \vec{O}], v11[\vec{M}, N', \vec{O}] \rangle\!\rangle$$

where v is the reduced cut-node, then in the head of π' we just get:

$$v[\vec{M}, \vec{N}, \vec{O}] \rightsquigarrow \langle\!\langle v0[\vec{M}, \vec{N}, \vec{O}], v1[\vec{M}, \vec{N}, \vec{O}] \rangle\!\rangle$$

Here we see the need for $\langle\!\langle -, - \rangle\!\rangle$ not being an evaluation context: the computations involving $\vec{N}$ would otherwise be duplicated, thus potentially increasing the length of the run. If the head of π never touches the produced additive pair, then the head of π' is strictly shorter, and the cut on $e_1 \cap e_2$ is not visited anymore. Otherwise, this pair can only be destroyed by a $\cap$-l_i rule: $\langle\!\langle v10[\vec{M}, N', \vec{O}], v11[\vec{M}, N', \vec{O}] \rangle\!\rangle \rightsquigarrow v1i[\vec{M}, N', \vec{O}]$, and the head of π' can 'catch up' by doing:

$$\langle\!\langle v0[\vec{M}, \vec{N}, \vec{O}], v1[\vec{M}, \vec{N}, \vec{O}] \rangle\!\rangle \rightsquigarrow vi[\vec{M}, \vec{N}, \vec{O}] \rightsquigarrow vi1[\vec{M}, vi0[\vec{N}], \vec{O}] \rightsquigarrow^n vi1[\vec{M}, N', \vec{O}]$$

The size of the head has not changed, but the cut is closer to the end. The analogous case for residuals is similar since the creation of a λ-abstraction temporarily blocks reductions; other cases can be found in [11, App. D]. Finally, by the same argument as for Thm. 15 we obtain:

▶ **Theorem 37** (Cut elimination). *If* $\mathrm{LAL} \vdash^\infty \Gamma \to e$ *then* $\mathrm{LAL}^- \vdash^\infty \Gamma \to e$.

One useful application of this cut-elimination result is the following alternative proof of the upper bound result of Palka for star-continuous action lattices:

[1] More precisely, the occurrences of x selected by the typing derivation of M.

▶ **Corollary 38** (Palka [29]). AL^* *is in* Π^0_1.

Proof. We say that a sequent $\Gamma \to e$ has a *d-derivation*, for $d \in \mathbb{N}$, if there is a LAL^- derivation ending in $\Gamma \to e$ for which each branch has length d, or otherwise terminates at a correct initial sequent in length $< d$. To avoid validity issues, we assume that the left premiss of every $*$-r_2 step has nonempty antecedent, so that all preproofs become valid without sacrificing provability (cf. Prop. 9). We define a Π^0_1 predicate $\mathsf{Prov}(\Gamma \to e)$ as $\forall d \in \mathbb{N}$."there is a d-derivation of $\Gamma \to e$". Notice that this is indeed Π^0_1 since the size of a d-derivation is exponentially bounded. Furthermore, if $\mathsf{Prov}(\Gamma \to e)$ then, by the infinite pigeonhole principle, we may recover an infinite proof of $\Gamma \to e$, by inductively choosing premisses resulting in larger derivations that nonetheless prefix infinitely many d-derivations. ◀

7 Conclusions

We presented a simple sequent system LKA that admits non-wellfounded proofs and showed it to be sound and complete for Kleene algebra, KA, by consideration of the free model of rational languages. We showed that its regular fragment is already complete, in the presence of cut, by a direct simulation of KA. We also gave a cut-elimination result for LKA, obtaining an alternative proof of completeness of its cut-free fragment.

We were able to generalise these arguments to an extended system LAL of Kleene algebras with residuals and meets, resulting in a sound and complete cut-free system for the equational theory of star-continuous action lattices, AL^*. Thanks to the subformula property for cut-free proofs, this also gives us proof-theoretical characterisations of star-continuous action algebras and Kleene lattices. This yields alternative proofs of several results of Palka [29], namely conservativity of AL^* over its fragments, as well as their membership in Π^0_1.

Finally, we characterised the theory of all action lattices by just the regular proofs of LAL. Whether the equational theory of action lattices is decidable remains open. It would be interesting to see if techniques such as interpolants for our system LAL, or a characterisation of the image of cut-elimination on cut-free proofs, might yield decidability.

It would be natural to consider systems which are commutative and/or contain arbitrary fixed points, bringing the subject matter closer to that of [13]. We would however not be able to arrive at a similar subformula property once fixed point formulae are allowed to contain meets and residuals, since this property is essentially thanks to the presence of only 'positive' connectives in KA, from the point of view of focusing [2].

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. doi:10.1145/2535838.2535862.

2 J.-M. Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.

3 A. Angus and D. Kozen. Kleene algebra with tests and program schematology. Technical Report TR2001-1844, CS Dpt., Cornell University, July 2001. URL: http://hdl.handle.net/1813/5831.

4 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.

5 Thomas Braibant and Damien Pous. An efficient Coq tactic for deciding Kleene algebras. In *Proc. 1st ITP*, volume 6172 of *Lecture Notes in Computer Science*, pages 163–178. Springer Verlag, 2010. doi:10.1007/978-3-642-14052-5_13.

6 Samuel R. Buss. An introduction to proof theory. *Handbook of proof theory*, 137:1–78, 1998.

7 Wojciech Buszkowski. On action logic: Equational theories of action algebras. *J. Log. Comput.*, 17(1):199–217, 2007. doi:10.1093/logcom/exl036.

8 Pierre Clairambault. Least and greatest fixpoints in game semantics. In *Proc. FoSSaCS*, pages 16–31, 2009. doi:10.1007/978-3-642-00596-1_3.

9 J. H. Conway. *Regular algebra and finite machines*. Chapman and Hall, 1971.

10 Anupam Das and Damien Pous. A cut-free cyclic proof system for Kleene algebra. In *Proc. TABLEAUX*, volume 10501 of *Lecture Notes in Computer Science*, pages 261–277. Springer Verlag, 2017. doi:10.1007/978-3-319-66902-1_16.

11 Anupam Das and Damien Pous. Non-Wellfounded Proof Theory For (Kleene+Action)(Algebras+Lattices). Full version of this extended abstract, 2018. URL: https://hal.archives-ouvertes.fr/hal-01703942.

12 H. Doornbos, R. Backhouse, and J. van der Woude. A calculational approach to mathematical induction. *Theoretical Computer Science*, 179(1-2):103–135, 1997. doi:10.1016/S0304-3975(96)00154-5.

13 Amina Doumane, David Baelde, and Alexis Saurin. Infinitary proof theory: the multiplicative additive case. In *CSL*, volume 62 of *LIPIcs*, pages 42:1–42:17, 2016. doi:10.4230/LIPIcs.CSL.2016.42.

14 Jérôme Fortier and Luigi Santocanale. Cuts for circular proofs: semantics and cut-elimination. In *Proc. CSL*, volume 23 of *LIPIcs*, pages 248–262, 2013. doi:10.4230/LIPIcs.CSL.2013.248.

15 Alain Frisch and Luca Cardelli. Greedy regular expression matching. In *Proc. ICALP*, volume 3142 of *Lecture Notes in Computer Science*, pages 618–629. Springer Verlag, 2004. doi:10.1007/978-3-540-27836-8_53.

16 N. Galatos, P. Jipsen, T. Kowalski, and H. Ono. *Residuated Lattices: An Algebraic Glimpse at Substructural Logics*. Elsevier, 2007.

17 J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

18 Fritz Henglein and Lasse Nielsen. Regular expression containment: coinductive axiomatization and computational interpretation. In *Proc. POPL 2011*, pages 385–398. ACM, 2011. doi:10.1145/1926385.1926429.

19 C. A. R. Hoare, Bernhard Möller, Georg Struth, and Ian Wehrman. Concurrent Kleene Algebra. In *Proc. CONCUR*, volume 5710 of *Lecture Notes in Computer Science*, pages 399–414. Springer Verlag, 2009. doi:10.1007/978-3-642-04081-8_27.

20 P. Jipsen. From semirings to residuated Kleene lattices. *Studia Logica*, 76(2):291–303, 2004. doi:10.1023/B:STUD.0000032089.54776.63.

21 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.

22 D. Kozen. On Hoare logic and Kleene algebra with tests. *ACM Trans. Comput. Log.*, 1(1):60–76, 2000. doi:10.1145/343369.343378.

23 D. Kozen and M.-C. Patron. Certification of compiler optimizations using Kleene algebra with tests. In *Proc. CL2000*, volume 1861 of *Lecture Notes in Artificial Intelligence*, pages 568–582. Springer Verlag, 2000. doi:10.1007/3-540-44957-4_38.

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

25 Dexter Kozen. On action algebras. In J. van Eijck and A. Visser, editors, *Logic and Information Flow*, pages 78–88. MIT Press, 1994.

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

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

28 Joachim Lambek. The mathematics of sentence structure. *The American Mathematical Monthly*, 65:154–170, 1958.

29 Ewa Palka. An infinitary sequent system for the equational theory of *-continuous action lattices. *Fundamenta Informaticae*, pages 295–309, 2007. URL: http://iospress.metapress.com/content/r5p53611826876j0/.

30 Damien Pous. Kleene Algebra with Tests and Coq tools for while programs. In *Proc. ITP*, volume 7998 of *Lecture Notes in Computer Science*, pages 180–196. Springer Verlag, 2013. doi:10.1007/978-3-642-39634-2_15.

31 V. Pratt. Action logic and pure induction. In *Proc. JELIA*, volume 478 of *Lecture Notes in Computer Science*, pages 97–120. Springer Verlag, 1990. doi:10.1007/BFb0018436.

32 Volodimir Nikiforovych Redko. On defining relations for the algebra of regular events. *Ukrainskii Matematicheskii Zhurnal*, 16:120–126, 1964.

33 Kurt Schütte. *Proof Theory*. Grundlehren der mathematischen Wissenschaften 225. Springer Berlin Heidelberg, 1977. Translation of Beweistheorie, 1968.

34 Christian Wurm. Kleene algebras, regular languages and substructural logics. In *Proc. GandALF*, EPTCS, pages 46–59, 2014. doi:10.4204/EPTCS.161.7.