



HAL
open science

Non-wellfounded proof theory for Kleene algebra and extensions

Anupam Das, Damien Pous

► **To cite this version:**

Anupam Das, Damien Pous. Non-wellfounded proof theory for Kleene algebra and extensions. 2018. hal-01703942v1

HAL Id: hal-01703942

<https://hal.science/hal-01703942v1>

Preprint submitted on 8 Feb 2018 (v1), last revised 29 Jun 2018 (v3)

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

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

Non-wellfounded proof theory for Kleene algebra and extensions

Anupam Das¹ and Damien Pous²

1 University of Copenhagen

2 CNRS, Plume team, LIP, ENS Lyon, France

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.

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

1 Introduction

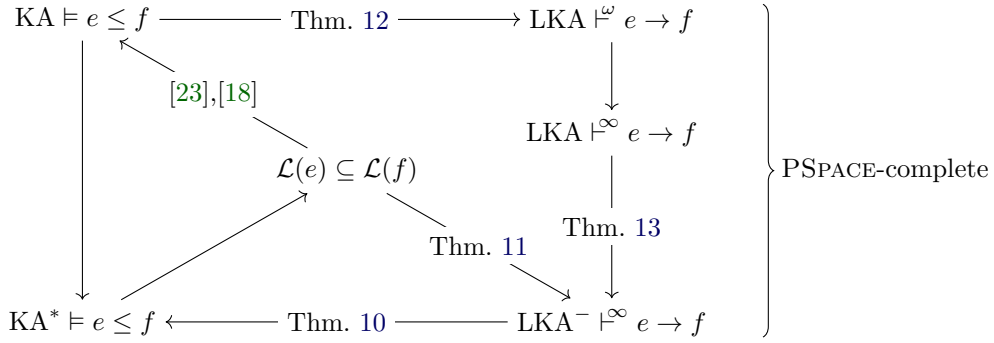
The axioms of *Kleene algebra* are sound and complete for the theory of regular expressions under language equivalence [18, 24]. As a consequence, the equational theory of Kleene algebra is decidable (actually PSPACE-complete). Models of these axioms of particular interest include formal languages and binary relations. For binary relations, Kleene star is interpreted as reflexive transitive closure, and the axioms of Kleene algebra make it possible to reason abstractly about program correctness [20, 21, 3, 15, 1]. The aforementioned decidability result moreover makes it possible to design automation tactics in proof assistants [4, 22, 27].

In previous work [8], we briefly discussed a sequent system for Kleene algebra, LKA, where proofs are finitely branching, but possibly infinitely deep (not well-founded). We observed that this system is sound w.r.t. Kleene algebra but that its *regular* fragment—those proofs which are unfoldings of finite graphs—is incomplete without cut. Our main contribution there was a variant of it based on ‘hypersequents’, whose regular fragment is sound and complete without cut.

In this paper we step back and study the proof theory of LKA, namely proving cut-elimination. This completes the corresponding proof-theoretical account of Kleene algebra and we show how to exploit this method to characterise important extensions of Kleene algebra: *action algebra* [28], whose language includes residuals, and *action lattices* [19], whose language further includes meets.

Kleene algebra

We focus on pure Kleene algebra first, which is easier to handle and enables a simpler presentation. Our key contribution here is that we give a simple validity criterion for proofs with cut, and prove that the corresponding system admits cut-elimination (Thm. 13). 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 [11], and by giving an interpretation of proofs as parse-tree transformers. (Such an idea already appears in [14] but for a finitary natural deduction system rather than for a non-wellfounded sequent calculus). Cut-elimination is crucial later to obtain



■ **Figure 1** Context and contributions for Kleene algebra.

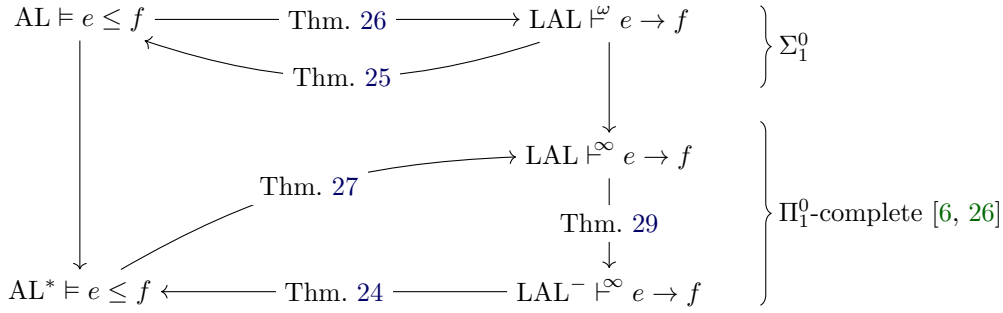
completeness results for the aforementioned extensions of Kleene algebra. We also prove that the system is sound for any *star-continuous* Kleene algebra (Thm. 10), and conversely, that it is complete w.r.t. the language theoretic interpretation of regular expressions (Thm. 11). We actually refine this latter result by showing that any proof from Kleene algebra axioms can be translated into a regular proof with cut (Thm. 12). Note however that regularity is not preserved by cut-elimination and, as mentioned above, that the cut-free regular system is actually incomplete w.r.t. Kleene algebra. These results are summarised in Fig. 1 (notations are formally defined in Sect. 2 and 3).

Action algebra 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 [29]. 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 [28]. These two operations, called *left residual* (\backslash) and *right residual* ($/$), are adjoint to sequential composition. Algebras of this kind are called *action algebras*. Kozen then proposed to extend the signature further to include a binary meet operation (\cap), so as to obtain a class of algebras, *action lattices*, which is closed under the formation of matrices [19].

While both action algebra and action lattices are finitely based and conservatively extend Kleene algebra, they bring some difficulties. By definition, their equational theories are at most Σ_1^0 , so that they must differ from their star-continuous variants, which Buszkowski and Palka proved to be Π_1^0 -complete [6, 26]. (Buszkowski proved the lower bound and Palka proved the upper bound.) It is actually not known whether the equational theories of action algebra and action lattices are decidable, and what their free models are. 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, as we mentioned, is PSPACE-complete).

Residuals and meets naturally correspond to linear implication and additive conjunction [16, 26], from (non-commutative) linear logic [13]. They are also essential connectives in the Lambek-calculus and related substructural logics [25]. We extend LKA accordingly into a system LAL and obtain the following additional results: LAL is complete for star continuous action lattices (Thm. 27); it still admits cut-elimination (Thm. 29); thus it is also sound w.r.t. star continuous action lattices (Thms. 24). Furthermore we show that its regular fragment with cut is in fact sound and complete for all action lattices (Thm. 25). These results are summarised in Fig. 2. Thms 24, 26, and 29 are proved by extending the proofs of Thms. 10, 12, and 13 to deal with the additional connectives. Amongst those,



■ **Figure 2** Context and contributions for action lattices.

cut-elimination is the most delicate extension, relying on higher types to interpret residuals, and proving that LAL proofs still yield finite programs in such a setting. Thm. 11 cannot be extended directly, due to the lack of a counterpart to languages when adding residuals and meet, which is why we instead rely on cut-elimination for completeness.

Lastly, we prove soundness of regular proofs with cuts with respect to all action lattices by reasoning on the cycle structure of such regular proofs (Thm. 25). There we exploit in an essential way the operations of residuals and meets: for action algebra and Kleene lattices, it remains open whether the corresponding regular fragments with cut are sound.

As explained above, while all notions are equivalent in the case of Kleene algebra (Fig. 1), complexity arguments make it possible to separate the lower and upper parts of Fig. 2. Whether the upper part is decidable remains open. An interesting point to notice here is that we manage to characterise action lattices in such a way that regularity makes it possible to tell the difference between the star continuous and the general cases.

Related work

Palka proposed a sequent system for star continuous action lattices, for which she proved cut-elimination [26]. Its non-star rules are precisely those of LKA, but the system is well-founded and relies on an ‘omega-rule’ for Kleene star with infinitely many premisses, in the traditional school of infinitary proof theory, cf. [30]. Doing so does not make it possible to consider ‘regular’ proofs, i.e., finite objects, as we do in the upper parts of Fig. 1 and 2. Wurm also proposed a (finite, and thus wellfounded) sequent system for Kleene algebra [31]. Unfortunately his cut-admissibility theorem does not hold—see [8].

The normalisation theory of *linear logic* with (least and greatest) fixed point operators has been studied in [10] and, more comprehensively, in [9]. 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 somewhat more complicated in the 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 [14]. Finally, due to our more bespoke setting, namely with $*$ as the only fixed point operator, we are able to arrive at cut-free systems with a genuine subformula property and simpler correctness criteria.

2 Preliminaries on Kleene algebra and extensions

Let A be a finite *alphabet*. *Regular expressions* [17] 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 denotes 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 $x + y = y$.

- (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 [7]. 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 $\text{KA} \models e = f$ if the equation $e = f$ universally holds in all Kleene algebras—equivalently, if it is derivable from the axioms of Kleene algebra. We use a similar notation for inequations. Kroh [23] and Kozen [18] have shown that this axiomatisation is complete for language inclusions. This corresponds to the right-to-left implication in the following characterisation.

► **Theorem 1** ([23, 18]). $\text{KA} \models e = f$ if and only if $\mathcal{L}(e) = \mathcal{L}(f)$.

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 implication:

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

We write $\text{KA}^* \models e = f$ 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 $\text{KA}^* \models e = f$ iff $\text{KA} \models e = f$; this is the triangle on the left in Fig. 1.

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:

$$\forall xyz, y \leq x \backslash z \Leftrightarrow xy \leq z \Leftrightarrow x \leq z / y \quad \forall xyz, z \leq x \cap y \Leftrightarrow z \leq x \wedge z \leq y$$

An *action algebra* is an action algebra with residuals only, a *Kleene lattice* is a Kleene algebra with meets only. We extend regular expressions accordingly, writing $\text{AL} \models e = f$ when the equation $e = f$ holds in all action lattices, and $\text{AL}^* \models e = f$ when it holds in all star-continuous action lattices.

Notice 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. [6, 26].

3 The sequent 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*, or simply the ‘left’ and ‘right’ hand sides, respectively. We say that a sequent $e_1, \dots, e_n \rightarrow e$

$$\begin{array}{c}
\text{cut} \frac{\Delta \rightarrow e \quad \Gamma, e, \Sigma \rightarrow f}{\Gamma, \Delta, \Sigma \rightarrow f} \quad \text{id} \frac{}{e \rightarrow e} \quad \text{0-l} \frac{}{\Gamma, 0, \Delta \rightarrow e} \quad \text{1-l} \frac{\Gamma, \Delta \rightarrow e}{\Gamma, 1, \Delta \rightarrow e} \quad \text{1-r} \frac{}{\rightarrow 1} \\
\text{-l} \frac{\Gamma, e, f, \Delta \rightarrow g}{\Gamma, e \cdot f, \Delta \rightarrow g} \quad \text{+l} \frac{\Gamma, e, \Delta \rightarrow g \quad \Gamma, f, \Delta \rightarrow g}{\Gamma, e + f, \Delta \rightarrow g} \quad \text{*l} \frac{\Gamma, \Delta \rightarrow f \quad \Gamma, e, e^*, \Delta \rightarrow f}{\Gamma, e^*, \Delta \rightarrow f} \\
\text{-r} \frac{\Gamma \rightarrow e \quad \Delta \rightarrow f}{\Gamma, \Delta \rightarrow e \cdot f} \quad \text{+r}_i \frac{\Gamma \rightarrow e_i}{\Gamma \rightarrow e_1 + e_2} \quad i \in \{1, 2\} \quad \text{*r}_1 \frac{}{\rightarrow e^*} \quad \text{*r}_2 \frac{\Gamma \rightarrow e \quad \Delta \rightarrow e^*}{\Gamma, \Delta \rightarrow e^*}
\end{array}$$

■ **Figure 3** The rules of LKA.

is *valid* if $\text{KA}^* \models e_1 \cdots e_n \leq e$. i.e., the comma is interpreted as sequential composition, and the sequent arrow as containment. We may refer to expressions as ‘formulae’ when it is more natural proof theoretic terminology, e.g. ‘subformula’ or ‘principal formula’.

The rules of LKA are given in Fig. 3. We call LKA^- the subset of LKA where the *cut* rule is omitted, which corresponds to the version of LKA discussed in [8]. Leaving the *-rules aside, these rules are those of the non-commutative variant of intuitionistic linear logic [13], 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 [26] follows the alternative interpretation of Kleene star as an infinite sum, $e^* = \sum_i e^i$, whence her left rule for Kleene star with infinitely many premisses, and the infinitely many right rules she uses for this operation.

As announced before, we allow infinitely deep proofs; we however need to impose a validity criterion to make sure derivations remain sound.

► **Definition 2.** A (binary, possibly infinite) *tree* is a prefix-closed subset of $\{0, 1\}^*$. A *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 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 [5].

► **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 $\text{LKA} \vdash^\infty \Gamma \rightarrow e$ if the sequent $\Gamma \rightarrow e$ admits a proof, $\text{LKA} \vdash^\omega \Gamma \rightarrow e$ if it admits a regular proof, and $\text{LKA}^- \vdash^\infty \Gamma \rightarrow e$ 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, colours to highlight some of the threads, and double lines to denote finite derivations.

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

$$\frac{\frac{\frac{\frac{\vdots}{b+c \rightarrow c+b} \quad *l \frac{\vdots}{(b+c)^* \rightarrow (c+b)^*}}{b+c, (b+c)^* \rightarrow (c+b)^*} \quad *r2}{\rightarrow (c+b)^*} \quad *r1}{(b+c)^* \rightarrow (c+b)^*} \quad *l$$

► **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,

$$\frac{\frac{\frac{\frac{\text{IH}}{e \rightarrow e} \quad *l \frac{\vdots}{e^* \rightarrow e^*}}{e, e^* \rightarrow e^*} \quad *r2}{\rightarrow e^*} \quad *r1}{e^* \rightarrow e^*} \quad *l$$

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 $a, a^* \rightarrow a^*a$ is the one on the left below:

$$\frac{\frac{\frac{\frac{\vdots}{a, a \rightarrow a^*a} \quad *l \frac{\vdots}{a, a, a, a^* \rightarrow a^*a}}{a, a, a^* \rightarrow a^*a} \quad *r1}{a \rightarrow a^*a} \quad *l}{a, a^* \rightarrow a^*a} \quad *l \quad \frac{\frac{\frac{\frac{\vdots}{a, a^* \rightarrow a^*a} \quad *l \frac{\vdots}{a, a^*a \rightarrow a^*a}}{a, a, a^* \rightarrow a^*a} \quad \text{cut}}{a \rightarrow a^*a} \quad *l}{a, a^* \rightarrow a^*a} \quad *l$$

This proof contains all sequents of the form $a, \dots, a, a^* \rightarrow a^*a$, whence non-regularity. A regular proof with cuts is given on the right; see [8] 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.

$$\frac{\frac{\frac{\vdots}{\rightarrow 1} \quad *r2}{a \rightarrow 1^*} \quad *r1}{a \rightarrow 1^*} \quad *r2 \quad \frac{\frac{\frac{\frac{\text{id}}{a \rightarrow a} \quad \text{id}}{a, a^* \rightarrow a^*} \quad *r2}{a, a^* \rightarrow a^*} \quad \text{cut}}{\rightarrow b^*} \quad *l}{a^* \rightarrow b^*} \quad *l$$

On the right-hand side, note that 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 [8], 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. The direct implication is immediate. Conversely, every infinite path in a fair cut-free preproof has infinitely many $*l$ steps, and their principal formulas are finitely many by the subformula property. One can thus extract a valid thread. ◀

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

► **Proposition 9.** *Let π be a cut-free preproof. If the antecedent of the left premiss of every $*-r_2$ step is non-empty, then π is valid.*

Proof. See App. C. ◀

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 Krob and Kozen’s completeness theorem for Kleene algebras (the triangle on the left in Fig. 1), it suffices to prove soundness with respect to star-continuous Kleene algebra.

► **Theorem 10 (Soundness).** *If $\text{LKA}^- \vdash^\infty e_1, \dots, e_n \rightarrow e$, then $\text{KA}^* \models e_1 \cdots e_n \leq e$.*

Proof sketch. Note that finite derivations in LKA are easily proven sound; the difficulty consists in handling infinite derivations. The proof follows the same strategy as the one given in [8] for the hypersequent system HKA . We proceed by induction on the multiset of the star heights of the formulas in the antecedent, relying on star-continuity and the following lemma for $*-l$ -steps.

$$\text{If } \text{LKA}^- \vdash^\infty \Gamma, e^*, \Delta \rightarrow f \text{ then, for all } n \in \mathbb{N}, \text{LKA}^- \vdash^\infty \Gamma, e^n, \Delta \rightarrow f.^1$$

See App. A for a detailed proof. ◀

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 action algebras and lattices, for which we will need to go through cut-elimination.

► **Theorem 11 (Completeness).** *If $\mathcal{L}(e_1 \cdots e_n) \subseteq \mathcal{L}(e)$ then $\text{LKA}^- \vdash^\infty e_1, \dots, e_n \rightarrow e$.*

Proof sketch. Again this can be proved like in [8] 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 $a_1, \dots, a_k \rightarrow e$, with $k \geq 0$, where $a_1 \dots a_k$ is a word in $\mathcal{L}(e_1 \cdots e_n)$ and thus in $\mathcal{L}(e)$ by assumption. Those leaves can be replaced by finite derivations using only right rules. See App. B for a detailed proof. ◀

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 12 (Regular completeness).** *If $\text{KA} \models e \leq f$ then $\text{LKA} \vdash^\omega e \rightarrow f$.*

Proof. We prove the statement for equalities. Consider the relation \equiv defined by $e \equiv f$ if $\text{LKA} \vdash^\omega e \rightarrow f$ and $\text{LKA} \vdash^\omega f \rightarrow e$. 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 $\text{LKA} \vdash^\omega e \rightarrow f$, 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, thanks to soundness, Thm. 10. The (in)equational axioms defining KA can be proved by finite derivations. The only difficulty is in dealing with the two implications (b) and (c) from the definition of Kleene algebra. We implement them as follows:

¹ Where $e^0 = 1$ and $e^{n+1} = e \cdot e^n$.

$$\begin{array}{c}
\vdots \\
\text{*} \text{-} l \frac{}{e^*, f \rightarrow f} \bullet \\
\text{IH} \\
\text{cut} \frac{\text{id} \frac{}{f \rightarrow f} \quad e, f \rightarrow f}{e, e^*, f \rightarrow f}}{\text{*} \text{-} l \frac{}{e^*, f \rightarrow f} \bullet}
\end{array}
\qquad
\begin{array}{c}
\text{IH} \\
\text{cut} \frac{\text{id} \frac{}{f \rightarrow f} \quad f, e \rightarrow f}{f, e, e^* \rightarrow f} \quad \text{*} \text{-} l \frac{}{f, e^* \rightarrow f} \bullet \\
\text{*} \text{-} l \frac{}{f, e^* \rightarrow f} \bullet
\end{array}$$

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

Note the asymmetry when we interpret the implications (b) and (c): 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 13.** *If $\text{LKA} \vdash^\infty \Gamma \rightarrow e$ then $\text{LKA}^- \vdash^\infty \Gamma \rightarrow e$.*

Combined with Thm. 10, it establishes the soundness of our criterion for proofs with cuts. This serves as a ‘warm-up’ for the analogous result for action lattices, in Sect. 6, where the proof is considerably more technical, although follows a similar template.

It will be useful to consider proofs as certain *transducers*, transforming *parse-trees* of input words of languages computed by terms. This leads to the notion of a ‘run’ that allows us to identify finite parts of proofs that are later used as certain invariants during the cut-elimination argument.

We do not give full proofs here but further details may be found in App. C.

4.1 Programs from proofs

We first define programs and their reduction semantics, based on which we prove cut-elimination, in Sect. 4.2.

► **Definition 14 (Programs).** *Programs* are defined by the following syntax, where x ranges over countable set of *variables*, i ranges over 1, 2, and π ranges over (valid) LKA proofs.

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

Intuitively, programs compute parse-trees for words belonging to the language of an expression. Given a proof π of the sequent $\Gamma \rightarrow e$, the last entry corresponds to the application of π 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 15 (Types).** A *typing environment*, written E , is a list of pairs of variables and expressions, written $x : e$. We say that a program M has type e in environment E , written $E \vdash M : e$ if this judgement can be derived from the rules in Fig. 4.

Note that the type system is linear; this is crucial for handling residuals in Sect. 5.

$$\begin{array}{c}
\frac{}{x : e \vdash x : e} \quad \frac{}{\vdash \star : 1} \quad \frac{E \vdash M : e \quad F \vdash N : f}{E, F \vdash \langle M, N \rangle : ef} \quad \frac{E \vdash M : e_i}{E \vdash \text{in}_i M : e_1 + e_2} \quad \frac{}{\vdash [] : e^*} \\
\\
\frac{E \vdash M : e \quad E' \vdash N : e^*}{E, E' \vdash M :: N : e^*} \quad \frac{\pi \text{ ends with } e_1, \dots, e_n \rightarrow f \quad \forall i, E_i \vdash M_i : e_i}{\vec{E} \vdash \pi[\vec{M}] : f}
\end{array}$$

■ **Figure 4** Typing rules for programs.

► **Observation 16.** Let x_1, \dots, x_n be variables. We have $a_1 \dots a_n \in \mathcal{L}(e)$ iff there exists a ground program M (i.e. without proof symbols π) such that $x_1 : a_1, \dots, x_n : a_n \vdash M : e$. This observation actually also holds for non-ground programs a posteriori, as a consequence of soundness (Thm. 10) and cut-elimination (Thm. 13 below).

Also note that one can always provide trivial arguments to a given proof:

► **Observation 17.** For every proof π of $e_1, \dots, e_n \rightarrow f$, we have $x_1 : e_1, \dots, x_n : e_n \vdash \pi[\vec{x}] : f$.

► **Definition 18** (Reduction). *Reduction*, written \rightsquigarrow , is the closure under all contexts of the following rules based on the last step of a proof π . Those rules are written concisely for lack of space; in each case, π' and π'' denote the premisses (when they exist), ordered from left to right. We moreover assume that the sizes of the vectors match those that arise from the various rules. See App. E for an extensive definition.

- *id*: $\pi[M] \rightsquigarrow M$.
- *cut*: $\pi[\vec{M}, \vec{N}, \vec{P}] \rightsquigarrow \pi''[\vec{M}, \pi'[\vec{N}], \vec{P}]$.
- *1-r*: $\pi[] \rightsquigarrow \star$.
- *-r*: $\pi[\vec{M}, \vec{N}] \rightsquigarrow \langle \pi'[\vec{M}], \pi''[\vec{N}] \rangle$.
- *+r_i*: $\pi[\vec{M}] \rightsquigarrow \text{in}_i(\pi'[\vec{M}])$.
- **-r₁*: $\pi[] \rightsquigarrow []$.
- **-r₂*: $\pi[\vec{M}, \vec{N}] \rightsquigarrow \pi'[\vec{M}] :: \pi''[\vec{N}]$.
- *1-l*: $\pi[\vec{M}, \star, \vec{N}] \rightsquigarrow \pi'[\vec{M}, \vec{N}]$.
- *-l*: $\pi[\vec{M}, \langle M, N \rangle, \vec{N}] \rightsquigarrow \pi'[\vec{M}, M, N, \vec{N}]$.
- *+l*: $\pi[\vec{M}, \text{in}_1 M, \vec{N}] \rightsquigarrow \pi'[\vec{M}, M, \vec{N}]$ and $\pi[\vec{M}, \text{in}_2 M, \vec{N}] \rightsquigarrow \pi''[\vec{M}, M, \vec{N}]$.
- **-l*: $\pi[\vec{M}, [], \vec{N}] \rightsquigarrow \pi'[\vec{M}, [], \vec{N}]$ and $\pi[\vec{M}, M :: N, \vec{N}] \rightsquigarrow \pi''[\vec{M}, M, N, \vec{N}]$.

Notice that the proofs appearing in a reduct of a program are all subproofs of the proofs already appearing in that program. As one might expect, we have subject reduction:

► **Proposition 19** (Subject reduction). *If $E \vdash M : e$ and $M \rightsquigarrow N$, then $E \vdash N : e$.*

For the sake of simplicity, we assume in the sequel the following (weak) reduction strategy: the second reduction rule for **-l* is applicable only once M and N are irreducible.

► **Definition 20** (Supports and runs). The *support* of a program M is the set of proofs appearing as subterms of any program reachable from M by a sequence of reductions. If the proofs occurring in M are subproofs of some global proof π , the *run* of M is the subgraph of π induced by the conclusions of proofs in the support of M .

The following result is the key observation:

► **Proposition 21.** *If $E \vdash M : e$, for some environment E and expression e , then the support (and so also the run) of M is finite.*

Proof sketch. First observe that the run of M has finitely many connected components. Then suppose by contradiction that the run of M is not finite. By König's Lemma one can extract an infinite branch through the run and thus, by validity, a thread along a star formula e^* which is infinitely often principal. By analysis of the reduction rules, we find a sequence of programs of type e^* whose sizes are strictly decreasing, which is impossible. ◀

Although this is not needed, one could also prove that reductions from well-typed programs are strongly normalising.

4.2 Cut reduction: production of cut-free prefixes

Our cut-elimination argument will be 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 App. D.

To produce an infinite cut-free proof, we must show that we may produce arbitrarily large cut-free proofs in a continuous manner. One issue here is to show that such a procedure is *productive*, i.e., eventually produces non-cut steps. To this end we use the following notion of run and we show that they are in some sense 'preserved' by cut-reduction:

► **Definition 22 (Bar).** A *bar* in a proof π is a set B of sequents that conclude disjoint subproofs of π , such that every infinite branch of π intersects B .

Notice that, by definition, a bar must be finite.

► **Lemma 23 (Cut-reduction lemma).** *For any proof π with a designated bar B , there is a proof π' of the same conclusion with a designated bar B' under which it is cut-free and satisfies the condition of Prop. 9 for $*-r_2$ rules. Furthermore, appealing to Obs. 17, if the run of $\pi[\vec{x}]$ lies below B in π , then the run of $\pi'[\vec{x}]$ lies below B' in π' .*

Proof sketch. First remove all the (useless) instances of rule $*-r_2$ below the bar that do not satisfy the condition, by using directly their right premiss. Then proceed by induction on the multiset of cut degrees underneath bar. By appealing to a sub-induction on the distance of a cut step from the bar, we may operate on a topmost cut step to reduce this measure by applying cut reduction steps from App. D. For instance, one interesting case is the key $*-cut$ reduction:

$$\begin{array}{c}
 \begin{array}{ccc}
 \begin{array}{c} \triangle \\ \pi_e \end{array} & \begin{array}{c} \triangle \\ \pi_{e^*} \end{array} & \begin{array}{c} \triangle \\ \pi_f^1 \end{array} \\
 \Delta \rightarrow e & \Sigma \rightarrow e^* & \Gamma, \Pi \rightarrow f \\
 \hline
 \Delta, \Sigma \rightarrow e^* & & \Gamma, e^*, \Pi \rightarrow f \\
 \text{\textit{cut}} & &
 \end{array}
 \quad \text{\textit{*-}r_2} \quad
 \begin{array}{ccc}
 \begin{array}{c} \triangle \\ \pi_e \end{array} & \begin{array}{c} \triangle \\ \pi_{e^*} \end{array} & \begin{array}{c} \triangle \\ \pi_f^+ \end{array} \\
 \Delta \rightarrow e & \Sigma \rightarrow e^* & \Gamma, e, e^*, \Pi \rightarrow f \\
 \hline
 \Delta \rightarrow e & \Gamma, e, \Sigma, \Pi \rightarrow f & \\
 \text{\textit{cut}} & &
 \end{array}
 \quad \text{\textit{*-}l}
 \end{array}
 \mapsto
 \begin{array}{c}
 \begin{array}{ccc}
 \begin{array}{c} \triangle \\ \pi_e \end{array} & \begin{array}{c} \triangle \\ \pi_{e^*} \end{array} & \begin{array}{c} \triangle \\ \pi_f^+ \end{array} \\
 \Delta \rightarrow e & \Sigma \rightarrow e^* & \Gamma, e, e^*, \Pi \rightarrow f \\
 \hline
 \Delta \rightarrow e & \Gamma, e, \Sigma, \Pi \rightarrow f & \\
 \text{\textit{cut}} & &
 \end{array}
 \end{array}$$

Here we must apply the sub-inductive hypothesis to the remaining e^* cut. Notice that any run on the left can have no branch through π_f^1 , by construction, and so is indeed preserved on the right. In this case its size has decreased, and B' is defined by restricting B to the subproofs π_e, π_{e^*} and π_f^+ . ◀

We can now prove cut-elimination.

Proof sketch of Thm. 13. We show that we can define cut-free prefixes of proofs under arbitrarily high bars. Given a bar bounding a cut-free prefix of a proof, consider a sequent in the bar concluding some subproof π ending in a *cut*-step (in the worst case). Choose any new bar above the run of $\pi[\vec{x}]$ and apply Lemma 23 above to produce a proof with a

bar under which it is cut-free and contains the run. By definition of runs, we must have added some non-*cut* steps beneath the bar, and so we have defined a larger cut-free prefix, as required. Finally, the preproof at the limit of this construction satisfies the condition of Prop. 9 (since it is a local condition), so it must be valid. ◀

5 Action algebras, Kleene lattices, and action lattices

We now consider extensions of Kleene algebra by residuals and meets, as axiomatised in [28] and [19]. We first extend the system LKA with the following rules, which are standard from substructural logic [25, 12], writing LAL for the corresponding system.

$$\begin{array}{c}
\frac{\Delta \rightarrow e \quad \Gamma, f, \Sigma \rightarrow g}{\Gamma, \Delta, e \setminus f, \Sigma \rightarrow g} \quad \frac{\Delta \rightarrow e \quad \Gamma, f, \Sigma \rightarrow g}{\Gamma, f/e, \Delta, \Sigma \rightarrow g} \quad \frac{\Gamma, e_i, \Delta \rightarrow f}{\Gamma, e_1 \cap e_2, \Delta \rightarrow f} \quad i \in \{1, 2\} \\
\frac{e, \Gamma \rightarrow f}{\Gamma \rightarrow e \setminus f} \quad \frac{\Gamma, e \rightarrow f}{\Gamma \rightarrow f/e} \quad \frac{\Gamma \rightarrow e \quad \Gamma \rightarrow f}{\Gamma \rightarrow e \cap f}
\end{array}$$

We define judgements as previously. Unless stated explicitly, the results below also hold for action algebras and Kleene lattices using the appropriate fragment of LAL.

To prove soundness w.r.t. star-continuous models, we need to slightly refine the measure we use, to take negative occurrences of star in the succedent into account.

► **Theorem 24 (Soundness).** *If $LAL^- \vdash^\infty e_1, \dots, e_n \rightarrow e$, then $AL^* \models e_1 \cdots e_n \leq e$.*

Proof sketch. The proof is similar to that for star-continuous Kleene algebra, Thm. 10, only when defining the multiset measure we must take into account star sub-expressions that occur in negative context, even on the right, due to the presence of residuals. A detailed proof is given in App. A. ◀

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. We crucially rely on the presence of both residuals and intersection to compute invariants for Kleene stars here, so that our proof does not immediately carry over to action algebras and Kleene lattices.

► **Theorem 25 (Regular soundness).** *If $LAL \stackrel{\omega}{\vdash} e_1, \dots, e_n \rightarrow f$ then $AL \models e_1 \cdots e_n \leq f$.*

Proof. We prove the statement for all regular proofs in **-normal form*, where every backpointer points to a **-l*-step. We proceed by induction on the number of simple cycles in such a proof π . W.l.o.g., suppose that π ends with a **-l* step that is the target of a backpointer, and colour in red all ancestors of its principal formula that are the same expression, e^* . Let $\{\Gamma_i, e^*, \Delta_i \rightarrow f_i\}_{i \in I}$ be the set of all sequents in π with e^* principal (thus for a **-l* step) and let $(\pi_i^l : \Gamma_i, \Delta_i \rightarrow f_i)_{i \in I}$ and $(\pi_i^r : \Gamma_i, e, e^*, \Delta_i \rightarrow f_i)_{i \in I}$ be the set of subproofs rooted at their left and right premisses, respectively.

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

$$\frac{\frac{\Gamma_j, \Delta_j \rightarrow f_j \quad \Gamma_j, e, e^*, \Delta_j \rightarrow f_j}{\Gamma_j, e^*, \Delta_j \rightarrow f_j}}{\Gamma_j, \Delta_j \rightarrow f_j} \quad \mapsto \quad \left. \frac{\frac{\frac{\Gamma_j \rightarrow g_j \quad f_j \rightarrow f_j}{\Gamma_j, g_j \setminus f_j \rightarrow f_j}}{\Delta_j \rightarrow d_j}}{\Gamma_j, h_j, \Delta_j \rightarrow f_j}}{\Gamma_j, h, \Delta_j \rightarrow f_j} \right\} \rho_j$$

Note that the proofs π_i^l and $\pi_i^{r'}$ have fewer simple cycles than π , so that by 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 those 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 (b). 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 j such that $(\Gamma_j, \Delta_j, f_j) = (\Gamma, \Delta, f)$. ◀

Although we cannot have a counterpart to Thm. 11, the regular fragment of LAL is indeed complete in the presence of *cut* for action lattices.

► **Theorem 26** (Regular completeness). *If $\text{AL} \models e \leq f$ then $\text{LAL} \vdash^\omega e \rightarrow f$.*

Proof. The axioms defining meet and residual immediately translate to finite derivations in LAL, one can thus extend the proof of Thm. 12. ◀

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 -complete [6]. The full system of non-wellfounded proofs is however complete for star-continuous models:

► **Theorem 27** (Star-continuous completeness). *If $\text{AL}^* \models e \leq f$ then $\text{LAL} \vdash^\infty e \rightarrow f$.*

Proof. As for Thms. 12 and 26, consider the relation \equiv' defined by $e \equiv' f$ if $\text{LAL} \vdash^\infty e \rightarrow f$ and $\text{LAL} \vdash^\infty f \rightarrow e$. 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 $(\Gamma, e^i, \Delta \rightarrow f)_{i \in \mathbb{N}}$ as follows:

$$\frac{\frac{\frac{\frac{\Gamma, e, \Delta \rightarrow f}{\pi_0}}{\Gamma, e, \Delta \rightarrow f} \quad \frac{\frac{\Gamma, e, e, \Delta \rightarrow f}{\pi_1}}{\Gamma, e, e, \Delta \rightarrow f} \quad \frac{\frac{\Gamma, e, e, \Delta \rightarrow f}{\pi_2}}{\Gamma, e, e, \Delta \rightarrow f} \quad \dots}{\Gamma, e, e, \Delta \rightarrow f} \quad \dots}{\Gamma, e, e^*, \Delta \rightarrow f} \quad \dots}{\Gamma, e^*, \Delta \rightarrow f} \quad \dots$$

The remaining property to establish is cut-elimination: combined with Thm. 24 it gives soundness of the system with cut w.r.t. star-continuous models, and combined with Thm. 27 it gives completeness of LAL^- w.r.t. those models.

6 Cut-elimination in LAL

The main alteration to the proof for Kleene algebra is that we need a more sophisticated notion of parse-trees and programs. For this we associate with a residual a functional that transforms (generalised) parse-trees appropriately. Meets are easier to handle: a parse-tree for $e \cap f$ is given by a parse-tree for e and another one for f .

We first extend the syntax of programs (Dfn. 14) to include λ -abstractions, which will be used for residuals. We do not add anything for meets since we can reuse tuples.

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

The type system (Fig. 4) is extended by the following rules:

$$\frac{x : e, E \vdash M : f}{E \vdash \lambda x.M : e \setminus f} \quad \frac{E, x : e \vdash M : f}{E \vdash \lambda x.M : f/e} \quad \frac{E \vdash M : e \quad E \vdash N : f}{E \vdash \langle M, N \rangle : e \cap f}$$

► **Lemma 28** (Substitution lemma). *If $F, x : e, F' \vdash M : f$ and $E \vdash N : e$ then we have $F, E, F' \vdash M\{N/x\} : f$, where $M\{N/x\}$ is M with occurrences of x substituted by N .²*

The following reductions are added, using the same conventions as in Dfn. 18:

- $\cap\text{-}r$: $\pi[\vec{M}] \rightsquigarrow \langle \pi'[\vec{M}], \pi''[\vec{M}] \rangle$.
- $\cap\text{-}l_i$: $\pi[\vec{M}, \langle M_1, M_2 \rangle, \vec{N}] \rightsquigarrow \pi'[\vec{M}, M_i, \vec{N}]$.
- $\backslash\text{-}r$: $\pi[\vec{M}] \rightsquigarrow \lambda x. \pi'[x, \vec{M}]$.
- $\backslash\text{-}l$: $\pi[\vec{M}, \vec{N}, \lambda x. F, \vec{P}] \rightsquigarrow \pi''[\vec{M}, F\{\pi'[\vec{N}]/x\}, \vec{P}]$.
- $/\text{-}r$: $\pi[\vec{M}] \rightsquigarrow \lambda x. \pi'[\vec{M}, x]$.
- $/\text{-}l$: $\pi[\vec{M}, \lambda x. F, \vec{N}, \vec{P}] \rightsquigarrow \pi''[\vec{M}, F\{\pi'[\vec{N}]/x\}, \vec{P}]$.

Props. 19 and 21, still hold in this setting (see App. C), so that it remains to show that cuts can be removed from finite proofs while preserving runs (Lem. 23). Here is the key meet case:

$$\frac{\frac{\frac{\pi_1}{\Delta \rightarrow e_1} \quad \frac{\pi_2}{\Delta \rightarrow e_2}}{\Delta \rightarrow e_1 \cap e_2} \quad \frac{\pi}{\Gamma, e_i, \Sigma \rightarrow f}}{\Gamma, \Delta, \Sigma \rightarrow f} \quad \mapsto \quad \frac{\frac{\pi_i}{\Delta \rightarrow e_i} \quad \frac{\pi}{\Gamma, e_i, \Sigma \rightarrow f}}{\Gamma, \Delta, \Sigma \rightarrow f}$$

This reduction reduces the size of a run by throwing away the decomposition information from one of the branches, and so the bar B' is just the restriction of B to π and π_i . This is a dual phenomenon to the key case for $+$.

Here is the key left residual case:

$$\frac{\frac{\frac{\pi}{e, \Delta \rightarrow f} \quad \frac{\pi_e}{\Delta \rightarrow e \setminus f}}{\Gamma_1, \Gamma_2, \Delta, \Sigma \rightarrow g} \quad \frac{\frac{\pi_e}{\Gamma_2 \rightarrow e} \quad \frac{\pi_f}{\Gamma_1, f, \Sigma \rightarrow g}}{\Gamma_1, \Gamma_2, e \setminus f, \Sigma \rightarrow g}}{\Gamma_1, \Gamma_2, \Delta, \Sigma \rightarrow g} \quad \mapsto \quad \frac{\frac{\pi_e}{\Gamma_2 \rightarrow e} \quad \frac{\frac{\pi}{e, \Delta \rightarrow f} \quad \frac{\pi_f}{\Gamma_1, f, \Sigma \rightarrow g}}{\Gamma_1, e, \Delta, \Sigma \rightarrow g}}{\Gamma_1, \Gamma_2, \Delta, \Sigma \rightarrow g}$$

This reduction again reduces the size of runs by directly composing the associated runs rather than first abstracting and Currying. In this case, we may simply preserve the same bounding bar.

The other cases can be found in App. D. After arriving at this generalised form of Lem. 23, the rest of the proof follows the same template as Thm. 13.

► **Theorem 29** (Cut elimination). *If $\text{LAL} \vdash^\infty \Gamma \rightarrow e$ then $\text{LAL}^- \vdash^\infty \Gamma \rightarrow e$.*

One useful application of this cut-elimination result is the following alternative proof of Palka's result that the equational theory of star-continuous action lattices is co-recursively-enumerable, by proof search.

► **Corollary 30** (Palka [26]). *AL^* is in Π_1^0 .*

Proof sketch. We say that a sequent $\Gamma \rightarrow d$ has a d -derivation, for $d \in \mathbb{N}$, if there is a LAL^- derivation ending in $\Gamma \rightarrow e$ 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 derivation

² Formally speaking, those occurrences that are selected by the typing derivation of M .

must satisfy the side-condition on $*-r_2$ steps from Prop. 9, that the left premiss has nonempty antecedent, so that all preproofs become valid without sacrificing provability.

We define a Π_1^0 predicate $\text{Prov}(\Gamma \rightarrow e)$ as $\forall d \in \mathbb{N}$. “there is a d -derivation of $\Gamma \rightarrow e$ ”. Notice that this is indeed Π_1^0 since the size of a d -derivation is exponentially bounded. Furthermore, if $\text{Prov}(\Gamma \rightarrow e)$ then, by the infinite pigeonhole principle, we may in fact recover an infinite proof of $\Gamma \rightarrow e$, by inductively choosing premisses resulting in larger derivations that nonetheless prefix infinitely many d -derivations. \blacktriangleleft

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 sub-formula property in cut-free proofs, this also gives us proof-theoretical characterisations for star-continuous action algebras and Kleene lattices, and conservativity of AL^* over those two theories. Furthermore, this gives rise to an alternative proof of Palka’s result that AL^* and its fragments are in Π_1^0 .

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: despite regularity, our characterisation does not allow for a proof search algorithm due to the need for the cut rule. It would be interesting to see if techniques such as interpolants for our LAL could 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 [9]. We would however not be able to arrive at a similar sub-formula 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 focussing [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.
- 2 J.-M. Andreoli. Logic programming with focusing proofs in linear logic. *J. Log. and Comp.*, 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.
- 4 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.
- 5 S. R. Buss. An introduction to proof theory. *Handbook of proof theory*, 137:1–78, 1998.
- 6 W. Buszkowski. [On action logic: Equational theories of action algebras](#). *J. Log. Comput.*, 17(1):199–217, 2007.
- 7 J. H. Conway. *Regular algebra and finite machines*. Chapman and Hall, 1971.
- 8 A. Das and D. Pous. [A cut-free cyclic proof system for Kleene algebra](#). In *Proc. TABLEAUX*, volume 10501 of *LNCS*, pages 261–277. Springer, 2017.

- 9 A. Doumane, D. Baelde, and A. Saurin. Infinitary proof theory: the multiplicative additive case. In *Proc. CSL*, volume 62 of *LIPICs*, pages 42:1–42:17, Sept. 2016.
- 10 J. Fortier and L. Santocanale. Cuts for circular proofs: semantics and cut-elimination. In *Proc. CSL*, volume 23 of *LIPICs*, pages 248–262, 2013.
- 11 A. Frisch and L. Cardelli. Greedy regular expression matching. In *Proc. ICALP*, volume 3142 of *LNCS*, pages 618–629. Springer, 2004.
- 12 N. Galatos, P. Jipsen, T. Kowalski, and H. Ono. *Residuated Lattices: An Algebraic Glimpse at Substructural Logics*. Elsevier, 2007.
- 13 J.-Y. Girard. Linear logic. *TCS*, 50:1–102, 1987.
- 14 F. Henglein and L. Nielsen. Regular expression containment: coinductive axiomatization and computational interpretation. In *Proc. POPL, 2011*, pages 385–398. ACM, 2011.
- 15 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.
- 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. On Hoare logic and Kleene algebra with tests. *ACM Trans. Comput. Log.*, 1(1):60–76, 2000.
- 21 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.
- 22 A. Krauss and T. Nipkow. Proof pearl: Regular expression equivalence and relation algebra. *JAR*, 49(1):95–106, 2012.
- 23 D. KroB. A Complete System of B-Rational Identities. In *Proc. ICALP*, volume 443 of *LNCS*, pages 60–73. Springer, 1990.
- 24 D. KroB. Complete systems of B-rational identities. *TCS*, 89(2):207–343, 1991.
- 25 J. Lambek. The mathematics of sentence structure. *The American Mathematical Monthly*, 65:154–170, 1958.
- 26 E. Palka. An infinitary sequent system for the equational theory of *-continuous action lattices. *Fundam. Inform.*, pages 295–309, 2007.
- 27 D. Pous. Kleene Algebra with Tests and Coq tools for while programs. In *Proc. ITP*, volume 7998 of *LNCS*, pages 180–196. Springer, 2013.
- 28 V. Pratt. Action logic and pure induction. In *Proc. JELIA*, volume 478 of *LNCS*, pages 97–120. Springer, 1990.
- 29 V. N. Redko. On defining relations for the algebra of regular events. *Ukrainskii Matematicheskii Zhurnal*, 16:120–126, 1964.
- 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 Soundness of LAL without cut w.r.t. star continuous models

We give here a detailed proof for Thms. 10 and 24: soundness of cut-free proofs w.r.t. star continuous models. The proofs are independent from the considered fragment (Kleene algebra, action algebra, action lattices) thanks to the sub-formula property.

► **Observation 31.** Every rule of LAL is sound (for Kleene algebra, action algebra, or action lattices depending on the connectives it mentions): if its premisses are valid then so is its conclusion.

We first prove the lemma mentioned in the proof sketch we gave for Thm. 10.

► **Lemma 32.** *If $\text{LAL}^- \vdash^\infty \Gamma, e^*, \Delta \rightarrow f$ then, for each $n \in \mathbb{N}$, $\text{LAL}^- \vdash^\infty \Gamma, e^n, \Delta \rightarrow f$.*

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

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

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. ◀

Now we formally define the measure on sequents which allows us to prove soundness of infinite proofs by induction.

► **Definition 33** (Weight of a sequent). The two functions h_*^+ and h_*^- from expressions to natural numbers, are defined by mutual recursion:

- $h_*^s(0) = h_*^s(1) = h_*^s(a) = 0$
- $h_*^s(e \cdot f) = h_*^s(e + f) = h_*^s(e \cap f) = \max(h_*^s(e), h_*^s(f))$
- $h_*^s(e \setminus f) = h_*^s(f / e) = \max(h_*^{-s}(e), h_*^s(f))$
- $h_*^-(e^*) = h_*^-(e) + 1$
- $h_*^+(e^*) = h_*^+(e)$

The *weight* of a sequent is the multiset $wh_*(e_1, \dots, e_n \rightarrow f) = \{h_*^-(e_1), \dots, h_*^-(e_n), h_*^+(f)\}$.

When e is a regular expression (without residuals or intersection), $h_*^+(e) = 0$ and $h_*^-(e)$ is the *star height* of the expression e , i.e., the maximum nesting of $*$ in its term tree. In presence of residuals, h_*^- (resp. h_*^+) counts the maximum nesting of $*$ appearing in positive (resp. negative) position.

► **Observation 34.** For every rule of LAL^- except $*-l$, the antecedent of each premiss has weight bounded by that of the conclusion.

We can now prove our soundness result:

Proof of Thms. 10 and 24. Let π be an LAL^- proof of $\Sigma \rightarrow e$ and let us proceed by induction on the weight of its conclusion. 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 π labels a binary tree, the prefix closure of B must be finite by König's Lemma. By using Obs. 31 a finite number of times, it thus suffices to show that every sequent in B is valid.

Let us consider a subproof π' that derives a sequent in B . If it is initial then it is necessarily valid; otherwise it must derive a sequent $\Gamma, f^*, \Delta \rightarrow g$ with a distinguished star-expression which is principal for the concluding $*-l$ -step of π' . By construction and Obs. 34 notice that $wh_*(\Gamma, f^*, \Delta \rightarrow g) \leq wh_*(\Sigma \rightarrow e)$. Now, by Lem. 32, π' can be transformed into proofs π'_n of $\Gamma, f^n, \Delta \rightarrow g$ for each $n \in \mathbb{N}$. Each π'_n derives a sequent whose weight is strictly smaller than that of $\Sigma \rightarrow e$, and so is sound by the inductive hypothesis. Finally, this means that $\Gamma, f^*, \Delta \rightarrow g$ is valid, by star-continuity. \blacktriangleleft

B Completeness of LKA without cut for Kleene algebra

► **Observation 35.** All left rules of LKA^- are (*weakly*) *invertible*: the validity of their conclusion implies the validity of all their premisses.

► **Lemma 36.** *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 $a_1, \dots, a_n \rightarrow e$ using only right logical rules.*

Proof. By a straightforward induction on e . \blacktriangleleft

Proof of Thm. 11. Suppose $\mathcal{L}(e_1 \cdot \dots \cdot e_n) \subseteq \mathcal{L}(e)$. We construct an LKA^- proof of $e_1, \dots, e_n \rightarrow e$ bottom-up, by first apply left-rules forever. By invertibility every sequent encountered will be valid, although this will not necessarily terminate due to $*-l$. Notice, however that we obtain fairness, since any infinite branch of only left rules must contain $*-l$ infinitely often.

The branches which terminate to a normal form have leaves of the form $a_1, \dots, a_n \rightarrow e$. Since such a leaf must be valid, we have $a_1 \dots a_n \in \mathcal{L}(e)$ and we can replace this leaf by a finite proof using Lem. 36. \blacktriangleleft

C Proofs for the cut-elimination arguments of Sects. 4 and 6

Here we present some further proof details of the intermediate results required for cut-elimination, all formulated for the general LAL case, as required in Sect. 6.

We first prove Prop. 9, which we need to assess the validity of the proofs produced by cut-elimination: every cut-free preproof such that the antecedent of the left premiss of every $*-r_2$ rule is non-empty is valid.

Proof of Prop. 9. Let the size of a sequent $\Gamma \rightarrow e$ be the multiset of the sizes of e and the formulas in Γ . By Prop. 8 it suffices to show that π is fair. Suppose by contradiction it contains an infinite branch containing only finitely many $*-l$ step. We get a contradiction by observing that bottom-up, from the point where this branch no longer contains any $*-l$ step, the size of the sequents strictly decreases. (The only interesting case is the $*-r_2$ rule, for which we know that $\Delta \rightarrow e^*$ is smaller than $\Gamma, \Delta \rightarrow e^*$ thanks to the condition we impose on this rule.) \blacktriangleleft

Now we establish the basic properties of the type system for programs: substitution lemma and subject reduction. We use the following notation:

► **Notation 37.** We write $\vec{E} \vdash \vec{M} : \Gamma$ when Γ has the same length as \vec{E} and \vec{M} and for all i , we have $E_i \vdash M_i : \Gamma_i$.

Proof sketch for Lemma 28. Given the typing derivation for $F, x : e, F' \vdash M : f$, replace every ancestor of $x : e$ on the left by E and every ancestor of x in M by N . The only critical case is the initial step, which can simply be replaced by a typing derivation for $E \vdash N : e$. \blacktriangleleft

Proof sketch of Prop. 19. We argue by induction on the structure of a typing derivation for $E \vdash M : e$. If the reduction takes place under a context, we use the induction hypothesis and the substitution lemma. Otherwise, it suffices to conduct a ‘last step analysis’, extracting the immediate subderivations to form a new typing derivation for $E \vdash N : e$. Some interesting cases are the following:

- $cut : \pi[\vec{M}, \vec{N}, \vec{P}] \rightsquigarrow \pi''[\vec{M}, \pi'[\vec{N}], \vec{P}]$. A typing derivation for the LHS has the form on the left below, so we recover a typing derivation for the RHS on the right below.

$$\frac{\vec{E} \vdash \vec{M} : \Gamma \quad \vec{F} \vdash \vec{N} : \Delta \quad \vec{G} \vdash \vec{P} : \Sigma}{\vec{E}, \vec{F}, \vec{G} \vdash \pi[\vec{M}, \vec{N}, \vec{P}] : f} \quad \frac{\vec{F} \vdash \vec{N} : \Delta}{\vec{F} \vdash \pi'[\vec{N}] : e} \quad \vec{E} \vdash \vec{M} : \Gamma \quad \vec{G} \vdash \vec{P} : \Sigma}{\vec{E}, \vec{F}, \vec{G} \vdash \pi''[\vec{M}, \vec{N}, \vec{P}] : f}$$

- $\backslash-l : \pi[\vec{M}, \vec{N}, \lambda x.P, \vec{Q}]$. A typing derivation for the LHS has the following form,

$$\frac{\vec{E} \vdash \vec{M} : \Gamma \quad \vec{F} \vdash \vec{N} : \Delta \quad \frac{x : e, \vec{G} \vdash P : f}{\vec{G} \vdash \lambda x.P : e \backslash f} \quad \vec{H} \vdash \vec{Q} : \Sigma}{\vec{E}, \vec{F}, \vec{G}, \vec{H} \vdash \pi[\vec{M}, \vec{N}, \lambda x.P, \vec{Q}] : g}$$

so we recover a typing derivation for the RHS as,

$$\frac{\vec{E} \vdash \vec{M} : \Gamma \quad \frac{\vec{F} \vdash \vec{N} : \Delta}{\vec{F} \vdash \pi'[\vec{N}] : e} \quad \frac{x : e, \vec{G} \vdash P : f}{\vec{F}, \vec{G} \vdash P\{\pi'[\vec{N}]/x\} : f} \quad \vec{H} \vdash \vec{Q} : \Sigma}{\vec{E}, \vec{F}, \vec{G}, \vec{H} \vdash \pi''[\vec{M}, P\{\pi'[\vec{N}]/x\}, \vec{Q}] : g} \text{ subst}$$

where we mark by **subst** an application of the substitution lemma. The case of $/-l$ is similar.

- $\backslash-r : \pi[\vec{M}] \rightsquigarrow \lambda x.\pi'[x, \vec{M}]$. A typing derivation for the LHS has the form on the left below, so we recover a typing derivation for the RHS on the right below.

$$\frac{\vec{E} \vdash \vec{M} : \Gamma}{\vec{E} \vdash \pi[\vec{M}] : e \backslash f} \quad \frac{\overline{x : e \vdash x : e} \quad \vec{E} \vdash \vec{M} : \Gamma}{\vec{E} \vdash \lambda x.\pi'[x, \vec{M}] : e \backslash f} \quad \frac{x : e, \vec{E} \vdash \pi'[x, \vec{M}] : f}{\vec{E} \vdash \lambda x.\pi'[x, \vec{M}] : e \backslash f}$$

The case of $/-r$ is similar. ◀

We now show that the support of a program is finite (Prop. 21). We need the following intermediate result.

► **Proposition 38 (Connectedness).** *Runs have only finitely many connected components.*

Proof. Notice that any term M has only finitely many proofs occurring in it, say k ; we show by induction on a reduction sequence from $M \rightsquigarrow^* N$ that N has at most k connected components. For each reduction step $N \rightsquigarrow N'$, N' may only contain proofs that either already occur in N or are immediate subproofs of those already occurring in N . Thus the number of connected components cannot increase. ◀

Proof of Prop. 21. Suppose otherwise, and let M be a program whose run ρ is infinite. Since runs have only finitely many connected components, ρ must contain an infinite branch,

along which there must be a valid thread (since ρ is a subgraph of the valid proof π). Let $\Gamma_i, e^*, \Delta_i \rightarrow f_i$ be this infinite branch, with the valid thread on e^* , and π_i be the associated subproofs. By assumption, we have that $M \rightsquigarrow^* C_i[\pi_i[\vec{M}_i, N_i, \vec{P}_i]]$ for some $C_i, \vec{M}_i, N_i, \vec{P}_i$ well-typed and with N_i irreducible, due to the evaluation strategy (i.e. the next step would be $*-l$ on π_i). The resulting reduction sequence reduces the size of N_i infinitely often, due to the fact that the thread on e^* is valid, which is a contradiction. \blacktriangleleft

D Cut elimination rules

D.1 Key cases for cut reduction

We give the key cases for cut elimination, when the two cut formulae are principal for a logical or initial step. Referring to cut-reduction, Lemma 23, all bars are preserved in these cases due to the definition of a run: the bar cannot cross a key case of a run that it bounds.

Identity:

$$\text{id} \frac{}{e \rightarrow e} \quad \text{cut} \frac{\frac{}{e \rightarrow e} \quad \frac{\pi}{\Gamma, e, \Delta \rightarrow f}}{\Gamma, e, \Delta \rightarrow f}}{\Gamma, e, \Delta \rightarrow f} \mapsto \frac{\pi}{\Gamma, e, \Delta \rightarrow f}$$

Product:

$$\text{-r} \frac{\frac{\frac{\pi_e}{\Delta \rightarrow e} \quad \frac{\pi_f}{\Sigma \rightarrow f}}{\text{cut} \frac{\Delta, \Sigma \rightarrow ef}} \quad \frac{\pi_g}{\Gamma, e, f, \Pi \rightarrow g}}{\text{-l} \frac{\Gamma, e, f, \Pi \rightarrow g}}{\Gamma, \Delta, \Sigma, \Pi \rightarrow g} \mapsto \frac{\frac{\pi_e}{\Delta \rightarrow e} \quad \frac{\frac{\pi_f}{\Sigma \rightarrow f} \quad \frac{\pi_g}{\Gamma, e, f, \Pi \rightarrow g}}{\text{cut} \frac{\Gamma, e, \Sigma, \Pi \rightarrow g}}}{\text{cut} \frac{\Delta \rightarrow e}{\Gamma, \Delta, \Sigma, \Pi \rightarrow g}}$$

Sum:

$$\text{+r}_i \frac{\frac{\pi_i}{\Delta \rightarrow e_i} \quad \frac{\pi_f^1}{\Gamma, e_1, \Sigma \rightarrow f} \quad \frac{\pi_f^2}{\Gamma, e_2, \Sigma \rightarrow f}}{\text{+l} \frac{\Gamma, e_1, \Sigma \rightarrow f \quad \Gamma, e_2, \Sigma \rightarrow f}}{\text{cut} \frac{\Delta \rightarrow e_1 + e_2}{\Gamma, \Delta, \Sigma \rightarrow f}} \mapsto \frac{\frac{\pi_i}{\Delta \rightarrow e_i} \quad \frac{\pi_f^i}{\Gamma, e_i, \Sigma \rightarrow f}}{\text{cut} \frac{\Delta \rightarrow e_i}{\Gamma, \Delta, \Sigma \rightarrow f}}$$

Kleene star:

$$\text{*r}_1 \frac{}{\rightarrow e^*} \quad \text{*l} \frac{\frac{\pi_f^1}{\Gamma, \Delta \rightarrow f} \quad \frac{\pi_f^+}{\Gamma, e, e^*, \Delta \rightarrow f}}{\text{cut} \frac{\Gamma, \Delta \rightarrow f}{\Gamma, e^*, \Delta \rightarrow f}} \mapsto \frac{\pi_f^1}{\Gamma, \Delta \rightarrow f}$$

This case eliminates the e^* component in the flow graph.

Other Kleene star:

$$\text{*r}_2 \frac{\frac{\pi_e}{\Delta \rightarrow e} \quad \frac{\pi_{e^*}}{\Sigma \rightarrow e^*}}{\text{cut} \frac{\Delta, \Sigma \rightarrow e^*}} \quad \text{*l} \frac{\frac{\pi_f^1}{\Gamma, \Pi \rightarrow f} \quad \frac{\pi_f^+}{\Gamma, e, e^*, \Pi \rightarrow f}}{\text{*l} \frac{\Gamma, \Pi \rightarrow f \quad \Gamma, e, e^*, \Pi \rightarrow f}}{\Gamma, e^*, \Pi \rightarrow f} \mapsto \frac{\frac{\pi_e}{\Delta \rightarrow e} \quad \frac{\frac{\pi_{e^*}}{\Sigma \rightarrow e^*} \quad \frac{\pi_f^+}{\Gamma, e, e^*, \Pi \rightarrow f}}{\text{cut} \frac{\Gamma, e, \Sigma, \Pi \rightarrow f}}}{\text{cut} \frac{\Delta \rightarrow e}{\Gamma, \Delta, \Sigma, \Pi \rightarrow f}}$$

One:

$$\frac{\frac{1-r}{\rightarrow 1} \quad \frac{1-l}{\Gamma, \Delta \rightarrow f} \quad \frac{\pi}{\Gamma, \Delta \rightarrow f}}{\text{cut} \quad \Gamma, \Delta \rightarrow f} \mapsto \frac{\pi}{\Gamma, \Delta \rightarrow f}$$

Zero has no key case since it does not have any right introduction rule.

Meet:

$$\frac{\frac{\frac{\pi_1}{\Delta \rightarrow e_1} \quad \frac{\pi_2}{\Delta \rightarrow e_2}}{\text{cut} \quad \Delta \rightarrow e_1 \cap e_2} \quad \frac{\frac{\pi}{\Gamma, e_i, \Sigma \rightarrow f}}{\cap-l_i \quad \Gamma, e_1 \cap e_2, \Sigma \rightarrow f}}{\text{cut} \quad \Gamma, \Delta, \Sigma \rightarrow f} \mapsto \frac{\frac{\pi_i}{\Delta \rightarrow e_i} \quad \frac{\pi}{\Gamma, e_i, \Sigma \rightarrow f}}{\text{cut} \quad \Gamma, \Delta, \Sigma \rightarrow f}$$

Left residual:

$$\frac{\frac{\frac{\pi}{e, \Delta \rightarrow f}}{\backslash-r \quad \Delta \rightarrow e \setminus f} \quad \frac{\frac{\pi_e}{\Gamma_2 \rightarrow e} \quad \frac{\pi_f}{\Gamma_1, f, \Sigma \rightarrow g}}{\backslash-l \quad \Gamma_1, \Gamma_2, e \setminus f, \Sigma \rightarrow g}}{\text{cut} \quad \Gamma_1, \Gamma_2, \Delta, \Sigma \rightarrow g} \mapsto \frac{\frac{\pi_e}{\Gamma_2 \rightarrow e} \quad \frac{\frac{\pi}{e, \Delta \rightarrow f} \quad \frac{\pi_f}{\Gamma_1, f, \Sigma \rightarrow g}}{\text{cut} \quad \Gamma_1, e, \Delta, \Sigma \rightarrow g}}{\text{cut} \quad \Gamma_1, \Gamma_2, \Delta, \Sigma \rightarrow g}$$

Right residual: symmetrical to the left residual

D.2 Commutative cases

We give the commutative cases for cut-elimination, when at least one of the cut formulae is not principal for a logical step. In these cases, it is possible that the bounding bar is moved below the *cut*-step, since the run may lie below the cut after commutation. This is actually why cut-elimination is productive at all.

Left \cdot -l:

$$\frac{\frac{\frac{\pi_1}{\Delta, e, f, \Sigma \rightarrow g}}{\cdot-l \quad \Delta, e, f, \Sigma \rightarrow g} \quad \frac{\pi_2}{\Gamma, g, \Pi \rightarrow h}}{\text{cut} \quad \Gamma, \Delta, e, f, \Sigma \rightarrow h} \mapsto \frac{\frac{\pi_1}{\Delta, e, f, \Sigma \rightarrow g} \quad \frac{\pi_2}{\Gamma, g, \Pi \rightarrow h}}{\text{cut} \quad \Gamma, \Delta, e, f, \Sigma \rightarrow h} \quad \cdot-l \quad \Gamma, \Delta, e, f, \Sigma \rightarrow h$$

Right \cdot -l:

$$\frac{\frac{\pi_1}{\Sigma \rightarrow g} \quad \frac{\frac{\pi_1}{\Gamma, e, f, \Delta, g, \Pi \rightarrow h}}{\cdot-l \quad \Gamma, e, f, \Delta, g, \Pi \rightarrow h}}{\text{cut} \quad \Gamma, e, f, \Delta, \Sigma, \Pi} \mapsto \frac{\frac{\pi_1}{\Sigma \rightarrow g} \quad \frac{\pi_1}{\Gamma, e, f, \Delta, g, \Pi \rightarrow h}}{\text{cut} \quad \Gamma, e, f, \Delta, \Sigma, \Pi \rightarrow h} \quad \cdot-l \quad \Gamma, e, f, \Delta, \Sigma, \Pi \rightarrow h$$

Right $\cdot\text{-}r$:

$$\frac{\frac{\frac{\pi_1}{\Delta \rightarrow e} \quad \frac{\frac{\pi_2}{\Gamma, e, \Sigma_1 \rightarrow f} \quad \frac{\pi_3}{\Sigma_2 \rightarrow g}}{\Gamma, e, \Sigma_1, \Sigma_2 \rightarrow fg}}{\Gamma, \Delta, \Sigma_1, \Sigma_2 \rightarrow fg} \text{cut}}{\Gamma, \Delta, \Sigma_1, \Sigma_2 \rightarrow fg} \mapsto \frac{\frac{\frac{\pi_1}{\Delta \rightarrow e} \quad \frac{\pi_2}{\Gamma, e, \Sigma_1 \rightarrow f}}{\Gamma, \Delta, \Sigma_1 \rightarrow f} \quad \frac{\pi_3}{\Sigma_2 \rightarrow g}}{\Gamma, \Delta, \Sigma_1, \Sigma_2 \rightarrow fg} \text{cut}$$

(also an analogous case when e appears in the right premiss of $\cdot\text{-}r$)

Left $\text{+}l$:

$$\frac{\frac{\frac{\pi_1}{\Delta, e, \Sigma \rightarrow g} \quad \frac{\pi_2}{\Delta, f, \Sigma \rightarrow g}}{\Delta, e + f, \Sigma \rightarrow g} \quad \frac{\pi_3}{\Gamma, g, \Pi \rightarrow h}}{\Gamma, \Delta, e + f, \Sigma, \Pi \rightarrow h} \text{cut} \mapsto \frac{\frac{\frac{\pi_1}{\Delta, e, \Sigma \rightarrow g} \quad \frac{\pi_3}{\Gamma, g, \Pi \rightarrow h}}{\Gamma, \Delta, e, \Sigma, \Pi \rightarrow h} \quad \frac{\frac{\pi_2}{\Delta, f, \Sigma \rightarrow g} \quad \frac{\pi_3}{\Gamma, g, \Pi \rightarrow h}}{\Gamma, \Delta, f, \Sigma, \Pi \rightarrow h}}{\Gamma, \Delta, e + f, \Sigma, \Pi \rightarrow h} \text{cut}$$

(analogous cases for $\text{+}l$ on the right hand side)

Case for $\text{+}r_i$ is simple.

Left $\text{*}l$:

$$\frac{\frac{\frac{\pi_1}{\Delta, \Sigma \rightarrow f} \quad \frac{\pi_2}{\Delta, e, e^*, \Sigma \rightarrow f}}{\Delta, e^*, \Sigma \rightarrow f} \quad \frac{\pi_3}{\Gamma, f, \Pi \rightarrow g}}{\Gamma, \Delta, e^*, \Sigma, \Pi \rightarrow g} \text{cut} \mapsto \frac{\frac{\frac{\pi_1}{\Delta, \Sigma \rightarrow f} \quad \frac{\pi_3}{\Gamma, f, \Pi \rightarrow g}}{\Gamma, \Delta, \Sigma, \Pi \rightarrow g} \quad \frac{\frac{\pi_2}{\Delta, e, e^*, \Sigma \rightarrow f} \quad \frac{\pi_3}{\Gamma, f, \Pi \rightarrow g}}{\Gamma, \Delta, e, e^*, \Sigma, \Pi \rightarrow g}}{\Gamma, \Delta, e^*, \Sigma, \Pi \rightarrow g} \text{cut}$$

(analogous case for $\text{*}l$ on the right hand side)

Right $\text{*}r_2$:

$$\frac{\frac{\frac{\pi_1}{\Delta \rightarrow e} \quad \frac{\frac{\pi_2}{\Gamma, e, \Sigma_1 \rightarrow f} \quad \frac{\pi_3}{\Sigma_2 \rightarrow f^*}}{\Gamma, e, \Sigma_1, \Sigma_2 \rightarrow f^*}}{\Gamma, \Delta, \Sigma_1, \Sigma_2 \rightarrow f^*} \text{cut}}{\Gamma, \Delta, \Sigma_1, \Sigma_2 \rightarrow f^*} \mapsto \frac{\frac{\frac{\pi_1}{\Delta \rightarrow e} \quad \frac{\pi_2}{\Gamma, e, \Sigma_1 \rightarrow f}}{\Gamma, \Delta, \Sigma_1 \rightarrow f} \quad \frac{\pi_3}{\Sigma_2 \rightarrow f^*}}{\Gamma, \Delta, \Sigma_1, \Sigma_2 \rightarrow f^*} \text{cut}$$

(If Γ, Δ, Σ_1 is empty, just use π_3 to satisfy the condition from Prop. 9 we impose on $\text{*}r_2$ rules during cut-elimination.)

Other right $\text{*}r_2$:

$$\frac{\frac{\frac{\pi_1}{\Delta \rightarrow e} \quad \frac{\frac{\pi_2}{\Gamma \rightarrow f} \quad \frac{\pi_3}{\Sigma_1, e, \Sigma_2 \rightarrow f^*}}{\Gamma, \Sigma_1, e, \Sigma_2 \rightarrow f^*}}{\Gamma, \Sigma_1, \Delta, \Sigma_2 \rightarrow f^*} \text{cut}}{\Gamma, \Sigma_1, \Delta, \Sigma_2 \rightarrow f^*} \mapsto \frac{\frac{\frac{\pi_2}{\Gamma \rightarrow f} \quad \frac{\frac{\pi_1}{\Delta \rightarrow e} \quad \frac{\pi_3}{\Sigma_1, e, \Sigma_2 \rightarrow f^*}}{\Sigma_1, \Delta, \Sigma_2 \rightarrow f^*}}{\Gamma, \Sigma_1, \Delta, \Sigma_2 \rightarrow f^*} \text{cut}}$$

(Here we know that Γ is non-empty by construction, and preserve the invariant.)

Left $\cap\text{-}l_i$:

$$\frac{\frac{\frac{\pi}{\Delta, e_i, \Sigma \rightarrow f}}{\text{cut}} \frac{\frac{\pi'}{\Gamma, f, \Pi \rightarrow g}}{\Gamma, \Delta, e_1 \cap e_2, \Sigma \rightarrow f}}{\Gamma, \Delta, e_1 \cap e_2, \Sigma, \Pi \rightarrow g}}{\text{cut}} \quad \mapsto \quad \frac{\frac{\frac{\pi}{\Delta, e_i, \Sigma \rightarrow f} \quad \frac{\pi'}{\Gamma, f, \Pi \rightarrow g}}{\text{cut}} \quad \frac{\Gamma, \Delta, e_i, \Sigma, \Pi \rightarrow g}}{\Gamma, \Delta, e_1 \cap e_2, \Sigma, \Pi \rightarrow g}}{\text{cut}} \quad \frac{\Gamma, \Delta, e_i, \Sigma, \Pi \rightarrow g}}{\Gamma, \Delta, e_1 \cap e_2, \Sigma, \Pi \rightarrow g}}{\text{cut}} \quad \frac{\Gamma, \Delta, e_i, \Sigma, \Pi \rightarrow g}}{\Gamma, \Delta, e_1 \cap e_2, \Sigma, \Pi \rightarrow g}}{\text{cut}}$$

(analogous case for $\cap\text{-}l_i$ on the right hand side)

Right $\cap\text{-}r$:

$$\frac{\frac{\frac{\pi}{\Delta \rightarrow f}}{\text{cut}} \quad \frac{\frac{\frac{\pi_1}{\Gamma, f, \Sigma \rightarrow e_1} \quad \frac{\pi_2}{\Gamma, f, \Sigma \rightarrow e_2}}{\text{cut}} \quad \frac{\Gamma, f, \Sigma \rightarrow e_1 \cap e_2}}{\Gamma, \Delta, \Sigma \rightarrow e_1 \cap e_2}}{\text{cut}} \quad \frac{\Gamma, f, \Sigma \rightarrow e_1 \cap e_2}}{\Gamma, \Delta, \Sigma \rightarrow e_1 \cap e_2}}{\text{cut}} \quad \mapsto \quad \frac{\frac{\frac{\pi}{\Delta \rightarrow f} \quad \frac{\pi_1}{\Gamma, f, \Sigma \rightarrow e_1}}{\text{cut}} \quad \frac{\frac{\pi}{\Delta \rightarrow f} \quad \frac{\pi_2}{\Gamma, f, \Sigma \rightarrow e_2}}{\text{cut}} \quad \frac{\Gamma, \Delta, \Sigma \rightarrow e_1}}{\Gamma, \Delta, \Sigma \rightarrow e_1 \cap e_2}}{\text{cut}} \quad \frac{\Gamma, \Delta, \Sigma \rightarrow e_1 \cap e_2}}{\Gamma, \Delta, \Sigma \rightarrow e_1 \cap e_2}}{\text{cut}}$$

Left $\setminus\text{-}l$:

$$\frac{\frac{\frac{\frac{\pi_e}{\Delta_2 \rightarrow e} \quad \frac{\pi_f}{\Delta_1, f, \Sigma \rightarrow g}}{\text{cut}} \quad \frac{\pi}{\Gamma, g, \Pi \rightarrow h}}{\text{cut}} \quad \frac{\Gamma, g, \Pi \rightarrow h}}{\Gamma, \Delta_1, \Delta_2, e \setminus f, \Sigma, \Pi \rightarrow h}}{\text{cut}} \quad \mapsto \quad \frac{\frac{\frac{\pi_e}{\Delta_2 \rightarrow e} \quad \frac{\frac{\pi_f}{\Delta_1, f, \Sigma \rightarrow g} \quad \frac{\pi}{\Gamma, g, \Pi \rightarrow h}}{\text{cut}} \quad \frac{\Gamma, \Delta_1, f, \Sigma, \Pi \rightarrow h}}{\Gamma, \Delta_1, \Delta_2, e \setminus f, \Sigma, \Pi \rightarrow h}}{\text{cut}} \quad \frac{\Gamma, \Delta_1, f, \Sigma, \Pi \rightarrow h}}{\Gamma, \Delta_1, \Delta_2, e \setminus f, \Sigma, \Pi \rightarrow h}}{\text{cut}}$$

Right $\setminus\text{-}l$:

$$\frac{\frac{\frac{\frac{\pi}{\Gamma \rightarrow g}}{\text{cut}} \quad \frac{\frac{\frac{\pi_e}{\Delta_2, g, \Delta_3 \rightarrow e} \quad \frac{\pi_f}{\Delta_1, f, \Sigma \rightarrow h}}{\text{cut}} \quad \frac{\Delta_1, \Delta_2, g, \Delta_3, e \setminus f, \Sigma \rightarrow h}}{\Delta_1, \Delta_2, \Gamma, \Delta_3, e \setminus f, \Sigma, \Pi \rightarrow h}}{\text{cut}} \quad \frac{\Delta_1, \Delta_2, g, \Delta_3, e \setminus f, \Sigma \rightarrow h}}{\Delta_1, \Delta_2, \Gamma, \Delta_3, e \setminus f, \Sigma, \Pi \rightarrow h}}{\text{cut}} \quad \mapsto \quad \frac{\frac{\frac{\frac{\pi}{\Gamma \rightarrow g} \quad \frac{\pi_e}{\Delta_2, g, \Delta_3 \rightarrow e}}{\text{cut}} \quad \frac{\pi_f}{\Delta_1, f, \Sigma \rightarrow h}}{\text{cut}} \quad \frac{\Delta_1, \Delta_2, \Gamma, \Delta_3 \rightarrow e}}{\Delta_1, \Delta_2, \Gamma, \Delta_3, e \setminus f, \Sigma \rightarrow h}}{\text{cut}} \quad \frac{\Delta_1, \Delta_2, \Gamma, \Delta_3 \rightarrow e}}{\Delta_1, \Delta_2, \Gamma, \Delta_3, e \setminus f, \Sigma \rightarrow h}}{\text{cut}}$$

Other right $\setminus\text{-}l$:

$$\frac{\frac{\frac{\frac{\pi}{\Gamma \rightarrow g}}{\text{cut}} \quad \frac{\frac{\frac{\pi_e}{\Delta_3 \rightarrow e} \quad \frac{\pi_f}{\Delta_1, g, \Delta_2, f, \Sigma \rightarrow h}}{\text{cut}} \quad \frac{\Delta_1, g, \Delta_2, \Delta_3, e \setminus f, \Sigma \rightarrow h}}{\Delta_1, \Gamma, \Delta_2, \Delta_3, e \setminus f, \Sigma, \Pi \rightarrow h}}{\text{cut}} \quad \frac{\Delta_1, g, \Delta_2, \Delta_3, e \setminus f, \Sigma \rightarrow h}}{\Delta_1, \Gamma, \Delta_2, \Delta_3, e \setminus f, \Sigma, \Pi \rightarrow h}}{\text{cut}} \quad \mapsto \quad \frac{\frac{\frac{\frac{\pi_e}{\Delta_3 \rightarrow e} \quad \frac{\pi}{\Gamma \rightarrow g}}{\text{cut}} \quad \frac{\pi_f}{\Delta_1, g, \Delta_2, f, \Sigma \rightarrow h}}{\text{cut}} \quad \frac{\Delta_1, \Gamma, \Delta_2, f, \Sigma \rightarrow h}}{\Delta_1, \Gamma, \Delta_2, \Delta_3, e \setminus f, \Sigma \rightarrow h}}{\text{cut}} \quad \frac{\Delta_1, \Gamma, \Delta_2, f, \Sigma \rightarrow h}}{\Delta_1, \Gamma, \Delta_2, \Delta_3, e \setminus f, \Sigma \rightarrow h}}{\text{cut}}$$

(Last case of a right $\setminus\text{-}l$ is similar.)

Right $\setminus\text{-}r$:

$$\frac{\frac{\frac{\frac{\pi}{\Gamma \rightarrow g}}{\text{cut}} \quad \frac{\frac{\pi'}{e, \Delta, g, \Sigma \rightarrow f}}{\text{cut}} \quad \frac{\Delta, \Gamma, \Sigma \rightarrow e \setminus f}}{\Delta, \Gamma, \Sigma \rightarrow e \setminus f}}{\text{cut}} \quad \frac{\Delta, \Gamma, \Sigma \rightarrow e \setminus f}}{\Delta, \Gamma, \Sigma \rightarrow e \setminus f}}{\text{cut}} \quad \mapsto \quad \frac{\frac{\frac{\frac{\pi}{\Gamma \rightarrow g} \quad \frac{\pi'}{e, \Delta, g, \Sigma \rightarrow f}}{\text{cut}} \quad \frac{e, \Delta, \Gamma, \Sigma \rightarrow f}}{\Delta, \Gamma, \Sigma \rightarrow e \setminus f}}{\text{cut}} \quad \frac{e, \Delta, \Gamma, \Sigma \rightarrow f}}{\Delta, \Gamma, \Sigma \rightarrow e \setminus f}}{\text{cut}}$$

(Cases for right residuals are symmetrical to those for left residuals.)

E Summary of the reduction rules

- If π is $\frac{id}{e \rightarrow e}$
then $\pi[\vec{M}] \rightsquigarrow M$.
- If π is $\frac{1-r}{\rightarrow 1}$
then $\pi[\] \rightsquigarrow \star$.
- If π is $\frac{\frac{\pi'}{\Gamma \rightarrow e} \quad \frac{\pi''}{\Delta \rightarrow f}}{\rightarrow-r} \Gamma, \Delta \rightarrow ef$
then $\pi[\vec{M}, \vec{N}] \rightsquigarrow \langle \pi'[\vec{M}], \pi''[\vec{N}] \rangle$.
- If π is $\frac{\frac{\pi'}{\Gamma \rightarrow e_i}}{\rightarrow-r_i} \Gamma \rightarrow e_1 + e_2$
then $\pi[\vec{M}] \rightsquigarrow \text{in}_i(\pi'[\vec{M}])$.
- If π is $\frac{\rightarrow-r_1}{\rightarrow f^*}$
then $\pi[\] \rightsquigarrow [\]$.
- If π is $\frac{\frac{\pi'}{\Gamma \rightarrow e} \quad \frac{\pi''}{\Delta \rightarrow e^*}}{\rightarrow-r_2} \Gamma, \Delta \rightarrow e^*$
then $\pi[\vec{M}, \vec{N}] \rightsquigarrow \pi'[\vec{M}] :: \pi''[\vec{N}]$.
- If π is $\frac{\frac{\pi'}{\Gamma \rightarrow f_1} \quad \frac{\pi''}{\Gamma \rightarrow f_2}}{\rightarrow-r} \Gamma \rightarrow f_1 \cap f_2$
then $\pi[\vec{M}] \rightsquigarrow \langle \pi'[\vec{M}], \pi''[\vec{M}] \rangle$.
- If π is $\frac{\frac{\pi'}{e, \Gamma \rightarrow f}}{\rightarrow-r} \Gamma \rightarrow e \setminus f$
then $\pi[\vec{M}] \rightsquigarrow \lambda x. \pi'[x, \vec{M}]$.
- If π is $\frac{\frac{\pi'}{\Gamma, e \rightarrow f}}{\rightarrow-r} \Gamma \rightarrow f/e$
then $\pi[\vec{M}] \rightsquigarrow \lambda x. \pi'[\vec{M}, x]$.
- If π is $\frac{\frac{\pi'}{\Delta \rightarrow e} \quad \frac{\pi''}{\Gamma, e, \Sigma \rightarrow f}}{cut} \Gamma, \Delta, \Sigma \rightarrow f$
then $\pi[\vec{M}, \vec{N}, \vec{P}] \rightsquigarrow \pi''[\vec{M}, \pi'[\vec{N}], \vec{P}]$.
- If π is $\frac{\frac{\pi'}{\Gamma, \Delta \rightarrow f}}{1-l} \Gamma, 1, \Delta \rightarrow f$
then $\pi[\vec{M}, \star, \vec{N}] \rightsquigarrow \pi'[\vec{M}, \vec{N}]$.
- If π is $\frac{\frac{\pi'}{\Gamma, e, f, \Delta \rightarrow f}}{\rightarrow-l} \Gamma, ef, \Delta \rightarrow f$
then $\pi[\vec{M}, \langle M, N \rangle, \vec{N}] \rightsquigarrow \pi'[\vec{M}, M, N, \vec{N}]$.
- If π is $\frac{\frac{\pi'}{\Gamma, e_1, \Delta \rightarrow f} \quad \frac{\pi''}{\Gamma, e_2, \Delta \rightarrow f}}{\rightarrow+l} \Gamma, e_1 + e_2, \Delta \rightarrow f$
then $\pi[\vec{M}, \text{in}_1 M, \vec{N}] \rightsquigarrow \pi'[\vec{M}, M, \vec{N}]$
and $\pi[\vec{M}, \text{in}_2 M, \vec{N}] \rightsquigarrow \pi''[\vec{M}, M, \vec{N}]$.
- If π is $\frac{\frac{\pi'}{\Gamma, \Delta \rightarrow f} \quad \frac{\pi''}{\Gamma, e, e^*, \Delta \rightarrow f}}{\rightarrow-l} \Gamma, e^*, \Delta \rightarrow f$
then $\pi[\vec{M}, [\], \vec{N}] \rightsquigarrow \pi'[\vec{M}, [\], \vec{N}]$
and $\pi[\vec{M}, M :: N, \vec{N}] \rightsquigarrow \pi''[\vec{M}, M, N, \vec{N}]$.
- If π is $\frac{\frac{\pi'}{\Gamma, e_i, \Delta \rightarrow f}}{\rightarrow-l_i} \Gamma, e_1 \cap e_2, \Delta \rightarrow f$
then $\pi[\vec{M}, \langle M_1, M_2 \rangle, \vec{N}] \rightsquigarrow \pi'[\vec{M}, M_i, \vec{N}]$.
- If π is $\frac{\frac{\pi'}{\Delta \rightarrow e} \quad \frac{\pi''}{\Gamma, f, \Sigma \rightarrow g}}{\rightarrow-l} \Gamma, \Delta, e \setminus f, \Sigma \rightarrow g$
then $\pi[\vec{M}, \vec{N}, \lambda x. F, \vec{P}] \rightsquigarrow \pi''[\vec{M}, F\{\pi'[\vec{N}]/x\}, \vec{P}]$.
- If π is $\frac{\frac{\pi'}{\Delta \rightarrow e} \quad \frac{\pi''}{\Gamma, f, \Sigma \rightarrow g}}{\rightarrow-l} \Gamma, f/e, \Delta, \Sigma \rightarrow g$
then $\pi[\vec{M}, \lambda x. F, \vec{N}, \vec{P}] \rightsquigarrow \pi''[\vec{M}, F\{\pi'[\vec{N}]/x\}, \vec{P}]$.