Complexity of Reasoning in Kleene and Action Algebras

Stepan Kuznetsov

ESSLLI 2022 · Galway, Ireland · Gallimh, Éire

Lecture 3

Leftmost HKA_∞

- Recall that we are developing a finitary and cut-free proof system for deciding inclusion of regular languages: $A \to B$ should be derivable iff $\nu(A) \subseteq \nu(B)$.

Leftmost $\mathbf{HKA}_\infty$

- Recall that we are developing a finitary and cut-free proof system for deciding inclusion of regular languages: $A \to B$ should be derivable iff $\nu(A) \subseteq \nu(B)$.
- Such a calculus will be the leftmost fragment of $\mathbf{HKA}_\infty$ (Das & Pous 2017).

Leftmost $\mathbf{HKA}_\infty$

- Recall that we are developing a finitary and cut-free proof system for deciding inclusion of regular languages: $A \to B$ should be derivable iff $\nu(A) \subseteq \nu(B)$.
- Such a calculus will be the leftmost fragment of $\mathbf{HKA}_\infty$ (Das & Pous 2017).
- $\mathbf{HKA}_\infty$ is a hypersequent calculus, whose derivable objects are of the form $\Pi \to \mathcal{X}$, where Π is a sequence of formulae (reg. exp.) and $\mathcal{X} = \Delta_1 \mid \ldots \mid \Delta_k$ is a multiset of sequences of formulae.

Leftmost $\mathbf{HKA}_\infty$

The rules of $\mathbf{HKA}_\infty$, which allow only leftmost proofs, are as follows:

$$\frac{\Gamma \to \mathcal{X} \quad A, A^*, \Gamma \to \mathcal{X}}{A^*, \Gamma \to \mathcal{X}} *L \qquad \frac{A, B, \Gamma \to \mathcal{X}}{A \cdot B, \Gamma \to \mathcal{X}} \cdot L$$

$$\frac{A, \Gamma \to \mathcal{X} \quad B, \Gamma \to \mathcal{X}}{A + B, \Gamma \to \mathcal{X}} +L \qquad \frac{\Gamma \to \mathcal{X}}{1, \Gamma \to \mathcal{X}} 1L \qquad \frac{}{0, \Gamma \to \mathcal{X}} 0L$$

$$\frac{\Pi \to \Delta \mid A, A^*, \Delta \mid \mathcal{X}}{\Pi \to A^*, \Delta \mid \mathcal{X}} *R \qquad \frac{\Pi \to A, B, \Delta \mid \mathcal{X}}{\Pi \to A \cdot B, \Delta \mid \mathcal{X}} \cdot R$$

$$\frac{\Pi \to A, \Delta \mid B, \Delta \mid \mathcal{X}}{\Pi \to A + B, \Delta \mid \mathcal{X}} +R \qquad \frac{\Pi \to \Delta \mid \mathcal{X}}{\Pi \to 1, \Delta \mid \mathcal{X}} 1R$$

$$\frac{}{\varepsilon \to \varepsilon} Ax \qquad \frac{\Gamma \to \Delta_1 \mid \ldots \mid \Delta_m}{a, \Gamma \to a, \Delta_1 \mid \ldots \mid a, \Delta_m} K$$

$$\frac{\Pi \to \mathcal{X}}{\Pi \to \Delta \mid \mathcal{X}} WR \qquad \frac{\Pi \to \Delta \mid \Delta \mid \mathcal{X}}{\Pi \to \Delta \mid \mathcal{X}} CR$$

Leftmost $\mathbf{HKA}_\infty$

- $\mathbf{HKA}_\infty$ allows non-well-founded proofs, i.e., infinite paths starting from the goal sequent.

Leftmost $\mathbf{HKA}_\infty$

- **$\mathbf{HKA}_\infty$** allows non-well-founded proofs, i.e., infinite paths starting from the goal sequent.
- Such proofs should obey the **fairness condition:** any infinite path should traverse $*L$, turning right, infinitely many times.

Leftmost $\mathbf{HKA}_\infty$

- **$\mathbf{HKA}_\infty$** allows non-well-founded proofs, i.e., infinite paths starting from the goal sequent.
- Such proofs should obey the **fairness condition:** any infinite path should traverse $*L$, turning right, infinitely many times.
- It is illegal to produce an infinite path using only $*R$.

Leftmost $\mathbf{HKA}_\infty$

- **$\mathbf{HKA}_\infty$** allows non-well-founded proofs, i.e., infinite paths starting from the goal sequent.
- Such proofs should obey the **fairness condition:** any infinite path should traverse $*L$, turning right, infinitely many times.
- It is illegal to produce an infinite path using only $*R$.
- In a **circular** proof, each infinite branch has a repeated copy of the sequent already seen before, with an $*L$ between these occurrences.

Leftmost $\mathbf{HKA}_\infty$

- $\mathbf{HKA}_\infty$ allows non-well-founded proofs, i.e., infinite paths starting from the goal sequent.
- Such proofs should obey the **fairness condition:** any infinite path should traverse $*L$, turning right, infinitely many times.
- It is illegal to produce an infinite path using only $*R$.
- In a **circular** proof, each infinite branch has a repeated copy of the sequent already seen before, with an $*L$ between these occurrences.
- In such cases, further development of the path is replaced by a **backlink,** and the whole structure becomes finite.

Theorem (Das & Pous 2017)

A sequent is derivable in $\mathbf{HKA}_\infty$ *iff it is valid in the standard interpretation. Moreover, any valid sequent has a circular leftmost proof.*

- The "only if" direction (soundness) was proved yesterday.

Productivity Lemma

- For completeness, we use the following lemma.

Productivity Lemma

- For completeness, we use the following lemma.

Lemma

If there is a subderivation

$$\dfrac{\dfrac{\Gamma \to A^*, \Delta \mid \mathscr{X}}{\vdots\ \pi}}{\dfrac{\Gamma \to \Delta \mid A, A^*, \Delta \mid \mathscr{Y}}{\Gamma \to A^*, \Delta \mid \mathscr{Y}}\ *R}$$

where π *consists only of right logical rules, and the sequent* $\Gamma \to A^*, \Delta \mid \mathscr{Y}$ *is valid in the standard interpretation, then so is* $\Gamma \to \mathscr{X}$ *(i.e.,* A^*, Δ *there is redundant).*

Productivity Lemma

- For completeness, we use the following lemma.

Lemma

If there is a subderivation

$$\dfrac{\dfrac{\dfrac{\Gamma \to A^*, \Delta \mid \mathscr{X}}{\vdots\ \pi}}{\Gamma \to \Delta \mid A, A^*, \Delta \mid \mathscr{Y}}}{\Gamma \to A^*, \Delta \mid \mathscr{Y}} *R$$

where π consists only of right logical rules, and the sequent $\Gamma \to A^*, \Delta \mid \mathscr{Y}$ is valid in the standard interpretation, then so is $\Gamma \to \mathscr{X}$ (i.e., A^*, Δ there is redundant).

- This was also established yesterday.

Constructing Circular Proofs

Now we are going to prove the following two statements, which give the necessary result:

1. If $\nu(A) \subseteq \nu(B)$, then $A \to B$ has a leftmost proof in $\mathbf{HKA}_\infty$.
2. Each leftmost proof can be transformed into a circular one.

Constructing Leftmost Proofs

- Let a hypersequent $\Pi \to \mathscr{X}$ be valid in the standard model (with , interpreted as $\cdot$ and $|$ interpreted as $+$). We shall construct a leftmost proof for it.

Constructing Leftmost Proofs

- Let a hypersequent $\Pi \to \mathscr{X}$ be valid in the standard model (with , interpreted as $\cdot$ and $|$ interpreted as $+$). We shall construct a leftmost proof for it.
- First **(Stage 1)**, we eagerly apply leftmost left logical rules.

Constructing Leftmost Proofs

- Let a hypersequent $\Pi \to \mathscr{X}$ be valid in the standard model (with , interpreted as $\cdot$ and $|$ interpreted as $+$). We shall construct a leftmost proof for it.
- First **(Stage 1)**, we eagerly apply leftmost left logical rules.
- The rules are invertible, so all leaves are valid in the standard interpretation.

Constructing Leftmost Proofs

- Let a hypersequent $\Pi \to \mathscr{X}$ be valid in the standard model (with , interpreted as $\cdot$ and $|$ interpreted as $+$). We shall construct a leftmost proof for it.
- First **(Stage 1)**, we eagerly apply leftmost left logical rules.
- The rules are invertible, so all leaves are valid in the standard interpretation.
- Each leaf is either $\to \mathscr{X}$ or $a, \Gamma \to \mathscr{X}$.

Constructing Leftmost Proofs

- Let a hypersequent $\Pi \to \mathscr{X}$ be valid in the standard model (with , interpreted as $\cdot$ and $|$ interpreted as $+$). We shall construct a leftmost proof for it.
- First **(Stage 1)**, we eagerly apply leftmost left logical rules.
- The rules are invertible, so all leaves are valid in the standard interpretation.
- Each leaf is either $\to \mathscr{X}$ or $a, \Gamma \to \mathscr{X}$.
- An important thing to mention here is that Stage 1 could generate infinite paths, e.g.:

$$\dfrac{\Gamma \to \mathscr{X} \qquad \dfrac{\dfrac{\vdots}{1^*, \Gamma \to \mathscr{X}}}{1, 1^*, \Gamma \to \mathscr{X}}\ 1L}{1^*, \Gamma \to \mathscr{X}}\ {*L}$$

Constructing Leftmost Proofs

- Such infinite paths are fine, since they obey the fairness condition (i.e., traverse $*L$ infinitely often).

Constructing Leftmost Proofs

- Such infinite paths are fine, since they obey the fairness condition (i.e., traverse $*L$ infinitely often).
- We shall show that they are actually regular, but by now we do not care about it.

Constructing Leftmost Proofs

- Such infinite paths are fine, since they obey the fairness condition (i.e., traverse $*L$ infinitely often).
- We shall show that they are actually regular, but by now we do not care about it.
- On **Stage 2** we apply right logical rules.

Constructing Leftmost Proofs

- Such infinite paths are fine, since they obey the fairness condition (i.e., traverse $*L$ infinitely often).
- We shall show that they are actually regular, but by now we do not care about it.
- On **Stage 2** we apply right logical rules.
- These rules are also invertible.

Constructing Leftmost Proofs

- Such infinite paths are fine, since they obey the fairness condition (i.e., traverse $*L$ infinitely often).
- We shall show that they are actually regular, but by now we do not care about it.
- On **Stage 2** we apply right logical rules.
- These rules are also invertible.
- However, here we cannot go eagerly, since this would result in proofs violating fairness:

$$\dfrac{\dfrac{\dfrac{\vdots}{\Pi \to \Delta \mid 1^*, \Delta \mid \mathscr{X}}}{\Pi \to \Delta \mid 1, 1^*, \Delta \mid \mathscr{X}}\ 1R}{\Pi \to 1^*, \Delta \mid \mathscr{X}}\ *R$$

Constructing Leftmost Proofs

- Due to this reason, we impose the following restriction: we apply leftmost right logical rules to all occurrences, except for A^*'s which were already obtained by decomposition $(*R)$.

Constructing Leftmost Proofs

- Due to this reason, we impose the following restriction: we apply leftmost right logical rules to all occurrences, except for A^*'s which were already obtained by decomposition $(*R)$.
- Now in our leaves we have multisets $\mathscr{X}$, and each sequent of $\mathscr{X}$ is either b, Δ, or empty, or A^*, Δ, where A^* was obtained by decomposition.

Constructing Leftmost Proofs

- Due to this reason, we impose the following restriction: we apply leftmost right logical rules to all occurrences, except for A^*'s which were already obtained by decomposition ($*R$).
- Now in our leaves we have multisets $\mathscr{X}$, and each sequent of $\mathscr{X}$ is either b, Δ, or empty, or A^*, Δ, where A^* was obtained by decomposition.
- The latter get weakened out by Productivity Lemma (weakenings form **Stage 3**).

Constructing Leftmost Proofs

- Due to this reason, we impose the following restriction: we apply leftmost right logical rules to all occurrences, except for A^*'s which were already obtained by decomposition ($*R$).
- Now in our leaves we have multisets $\mathscr{X}$, and each sequent of $\mathscr{X}$ is either b, Δ, or empty, or A^*, Δ, where A^* was obtained by decomposition.
- The latter get weakened out by Productivity Lemma (weakenings form **Stage 3**).
- Recall that our leaves are $\to \mathscr{X}$ or $a, \Gamma \to \mathscr{X}$.

Constructing Leftmost Proofs

- Due to this reason, we impose the following restriction: we apply leftmost right logical rules to all occurrences, except for A^*'s which were already obtained by decomposition $(*R)$.
- Now in our leaves we have multisets $\mathcal{X}$, and each sequent of $\mathcal{X}$ is either b, Δ, or empty, or A^*, Δ, where A^* was obtained by decomposition.
- The latter get weakened out by Productivity Lemma (weakenings form **Stage 3**).
- Recall that our leaves are $\to \mathcal{X}$ or $a, \Gamma \to \mathcal{X}$.
- In the first case, by validity, we have $\varepsilon \in v(\mathcal{X})$. Thus, $\mathcal{X}$ contains the empty sequence. Everything else gets weakened out.

Constructing Leftmost Proofs

- For the second case, we weaken all sequences in $\mathscr{X}$, which are the empty sequence or begin with $b \neq a$.

Constructing Leftmost Proofs

- For the second case, we weaken all sequences in $\mathscr{X}$, which are the empty sequence or begin with $b \neq a$.
- Next, we apply K **(Stage 4).**

Constructing Leftmost Proofs

- For the second case, we weaken all sequences in $\mathcal{X}$, which are the empty sequence or begin with $b \neq a$.
- Next, we apply K **(Stage 4).**
- Above K, we have new valid sequents, on which we again start with Stage 1.

Constructing Leftmost Proofs

- For the second case, we weaken all sequences in $\mathscr{X}$, which are the empty sequence or begin with $b \neq a$.
- Next, we apply K **(Stage 4).**
- Above K, we have new valid sequents, on which we again start with Stage 1.
- Notice that there is not (transfinite) induction here, we do not care for reducing any sort of parameter.

Constructing Leftmost Proofs

- For the second case, we weaken all sequences in $\mathscr{X}$, which are the empty sequence or begin with $b \neq a$.
- Next, we apply K **(Stage 4).**
- Above K, we have new valid sequents, on which we again start with Stage 1.
- Notice that there is not (transfinite) induction here, we do not care for reducing any sort of parameter.
- The only thing we should care about is that the proof never gets stuck, and fairness is maintained.

Constructing Leftmost Proofs

- To prove fairness, for each infinite path we consider two possibilities.

Constructing Leftmost Proofs

- To prove fairness, for each infinite path we consider two possibilities.
- Case 1: the branch eventually ends in one stage. Then it should be Stage 1, and fairness is maintained (see above).

Constructing Leftmost Proofs

- To prove fairness, for each infinite path we consider two possibilities.
- Case 1: the branch eventually ends in one stage. Then it should be Stage 1, and fairness is maintained (see above).
- Case 2: the branch traverses Stages 1–4 infinitely.

Constructing Leftmost Proofs

- To prove fairness, for each infinite path we consider two possibilities.
- Case 1: the branch eventually ends in one stage. Then it should be Stage 1, and fairness is maintained (see above).
- Case 2: the branch traverses Stages 1–4 infinitely.
- In this case, consider the size of the antecedent. This size gets strictly reduced at each Stage 4, thus, it should be restored, which can be done only by $*L$.

Making Proofs Circular

- Now let us regularise our leftmost proof.

Making Proofs Circular

- Now let us regularise our leftmost proof.
- For this, we shall need yet another order, now on occurrences of subformulae in the goal sequent.

Making Proofs Circular

- Now let us regularise our leftmost proof.
- For this, we shall need yet another order, now on occurrences of subformulae in the goal sequent.
- We have $A \preccurlyeq B$, if A is either a subformula of B or is located to the left of B. This is a linear (total) order.

Making Proofs Circular

- Now let us regularise our leftmost proof.
- For this, we shall need yet another order, now on occurrences of subformulae in the goal sequent.
- We have $A \preccurlyeq B$, if A is either a subformula of B or is located to the left of B. This is a linear (total) order.
- Moreover, thanks to subformula property, the order can be propagated to occurrences in sequents throughout the proof.

Making Proofs Circular

- Now let us regularise our leftmost proof.
- For this, we shall need yet another order, now on occurrences of subformulae in the goal sequent.
- We have $A \preccurlyeq B$, if A is either a subformula of B or is located to the left of B. This is a linear (total) order.
- Moreover, thanks to subformula property, the order can be propagated to occurrences in sequents throughout the proof.

Lemma

In a leftmost proof, each sequence is strictly increasing w.r.t. $\preccurlyeq$.

Making Proofs Circular

- Now let us regularise our leftmost proof.
- For this, we shall need yet another order, now on occurrences of subformulae in the goal sequent.
- We have $A \preccurlyeq B$, if A is either a subformula of B or is located to the left of B. This is a linear (total) order.
- Moreover, thanks to subformula property, the order can be propagated to occurrences in sequents throughout the proof.

Lemma

In a leftmost proof, each sequence is strictly increasing w.r.t. $\preccurlyeq$.

- Corollary: the length of such a sequence is bounded by the size of the goal sequent.

Making Proofs Circular

- Finally, using WR and CR, we can assure that in each antecedent we have no more than 2 copies of a sequence.

Making Proofs Circular

- Finally, using WR and CR, we can assure that in each antecedent we have no more than 2 copies of a sequence.
- Therefore, the total number of sequents which could appear in our proof is finite.

Making Proofs Circular

- Finally, using WR and CR, we can assure that in each antecedent we have no more than 2 copies of a sequence.
- Therefore, the total number of sequents which could appear in our proof is finite.
- Thus, each infinite path will eventually reach a sequent which already occurred earlier.

Making Proofs Circular

- Finally, using WR and CR, we can assure that in each antecedent we have no more than 2 copies of a sequence.
- Therefore, the total number of sequents which could appear in our proof is finite.
- Thus, each infinite path will eventually reach a sequent which already occurred earlier.
- We identify two such instances with an application of $*L$ in between, and make a cycle.

Making Proofs Circular

- Finally, using WR and CR, we can assure that in each antecedent we have no more than 2 copies of a sequence.
- Therefore, the total number of sequents which could appear in our proof is finite.
- Thus, each infinite path will eventually reach a sequent which already occurred earlier.
- We identify two such instances with an application of $*L$ in between, and make a cycle.
- This makes $\mathbf{KA}_\omega$ r.e., and by Post's theorem we obtain decidability.

Making Proofs Circular

- Finally, using WR and CR, we can assure that in each antecedent we have no more than 2 copies of a sequence.
- Therefore, the total number of sequents which could appear in our proof is finite.
- Thus, each infinite path will eventually reach a sequent which already occurred earlier.
- We identify two such instances with an application of $*L$ in between, and make a cycle.
- This makes $\mathbf{KA}_\omega$ r.e., and by Post's theorem we obtain decidability.
- A more accurate consideration shows that each path reaches its cycle in polynomial number of steps, which makes $\mathbf{KA}_\omega$ PSPACE.

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{\varepsilon \to \varepsilon}{\varepsilon \to \varepsilon \mid aa, (aa)^* \mid a(aa)^*}\; WR
}{\varepsilon \to (aa)^* \mid a(aa)^*}\; {*R}
}{\varepsilon \to (aa)^* + a(aa)^*}\; {+R}
\qquad
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{a^* \to a, (aa)^* \mid (aa)^*}{\vdots}
}{a^* \to a, (aa)^* \mid (aa)^*}
}{a, a^* \to a, a, (aa)^* \mid a, (aa)^*}\; K
}{a, a^* \to \varepsilon \mid a, a, (aa)^* \mid a, (aa)^*}\; WR
}{a, a^* \to \varepsilon \mid aa, (aa)^* \mid a(aa)^*}\; {\cdot R}
}{a, a^* \to (aa)^* \mid a(aa)^*}\; {*R}
}{a, a^* \to (aa)^* + a(aa)^*}\; {+R}
}{a^* \to (aa)^* + a(aa)^*}\; {*L}
$$

Entailment

- The situation changes dramatically when we allow *hypotheses*.

Entailment

- The situation changes dramatically when we allow *hypotheses.*
- Let $\mathscr{H}$ be a finite set of sequents of the form $A \to B$, and we ask whether a sequent $C \to D$ follows from $\mathscr{H}$.

Entailment

- The situation changes dramatically when we allow *hypotheses.*
- Let $\mathscr{H}$ be a finite set of sequents of the form $A \to B$, and we ask whether a sequent $C \to D$ follows from $\mathscr{H}$.
- We could think of this both syntactically and semantically.

Entailment

- The situation changes dramatically when we allow *hypotheses.*
- Let $\mathscr{H}$ be a finite set of sequents of the form $A \to B$, and we ask whether a sequent $C \to D$ follows from $\mathscr{H}$.
- We could think of this both syntactically and semantically.
- Syntax: there exists a (possibly infinite) derivation in $\mathbf{KA}_\omega$, where sequents from $\mathscr{H}$ are allowed to be used as axioms.

Entailment

- The situation changes dramatically when we allow *hypotheses*.
- Let $\mathscr{H}$ be a finite set of sequents of the form $A \to B$, and we ask whether a sequent $C \to D$ follows from $\mathscr{H}$.
- We could think of this both syntactically and semantically.
- Syntax: there exists a (possibly infinite) derivation in $\mathbf{KA}_\omega$, where sequents from $\mathscr{H}$ are allowed to be used as axioms.
    - Cut is also allowed.

Entailment

- The situation changes dramatically when we allow *hypotheses.*
- Let $\mathscr{H}$ be a finite set of sequents of the form $A \to B$, and we ask whether a sequent $C \to D$ follows from $\mathscr{H}$.
- We could think of this both syntactically and semantically.
- Syntax: there exists a (possibly infinite) derivation in $\mathbf{KA}_\omega$, where sequents from $\mathscr{H}$ are allowed to be used as axioms.
    - Cut is also allowed.
- Semantics: for any interpretation, on a *-continuous Kleene algebra, where all sequents from $\mathscr{H}$ are valid, $C \to D$ is also valid.

Entailment

- The situation changes dramatically when we allow *hypotheses.*
- Let $\mathscr{H}$ be a finite set of sequents of the form $A \to B$, and we ask whether a sequent $C \to D$ follows from $\mathscr{H}$.
- We could think of this both syntactically and semantically.
- Syntax: there exists a (possibly infinite) derivation in $\mathbf{KA}_\omega$, where sequents from $\mathscr{H}$ are allowed to be used as axioms.
    - Cut is also allowed.
- Semantics: for any interpretation, on a *-continuous Kleene algebra, where all sequents from $\mathscr{H}$ are valid, $C \to D$ is also valid.
    - Notice that this is an abstract *-continuous KA, not the standard interpretation.

Entailment

- There is a **strong soundness-and-completeness** theorem which states that these two notions are equivalent.

Entailment

- There is a **strong soundness-and-completeness** theorem which states that these two notions are equivalent.
- The proof is by a standard Lindenbaum–Tarski canonical model argument.

Entailment

- There is a **strong soundness-and-completeness** theorem which states that these two notions are equivalent.
- The proof is by a standard Lindenbaum–Tarski canonical model argument.
- Recall that an abstract *-continuous KA is an algebraic structure in which axioms and rules of $\mathbf{KA}_\omega$ are valid.

Entailment

- There is a **strong soundness-and-completeness** theorem which states that these two notions are equivalent.
- The proof is by a standard Lindenbaum–Tarski canonical model argument.
- Recall that an abstract *-continuous KA is an algebraic structure in which axioms and rules of $\mathbf{KA}_\omega$ are valid.
- One can consider only algebras of regular subsets of monoids (but not necessarily the free monoid Σ^*).

Complexity Results

The following complexity results for entailment in *-continuous KAs are due to Kozen (2002):

$\mathscr{H}$ includes only monoid equations	Π^0_2-complete
$\mathscr{H}$ includes only length-preserving monoid equations	Π^0_1-complete
$\mathscr{H}$ is arbitrary	Π^1_1-complete

Upper Bounds

- For arbitrary $\mathscr{H}$'s, we have one $\forall$ quantifier over infinite structures (*-continuous KAs).

Upper Bounds

- For arbitrary $\mathscr{H}$'s, we have one $\forall$ quantifier over infinite structures (*-continuous KAs).
- This can also be done syntactically, by considering derivations in $\mathbf{KA}_{\omega}$.

Upper Bounds

- For arbitrary $\mathscr{H}$'s, we have one $\forall$ quantifier over infinite structures (*-continuous KAs).
- This can also be done syntactically, by considering derivations in $\mathbf{KA}_\omega$.
- For monoid equations, one can show that entailment in arbitrary KAs from such $\mathscr{H}$ is equivalent to validity in $\mathsf{REG}(\Sigma^*)/\mathscr{H}$ (i.e., the factor-algebra of the algebra of regular languages).

Upper Bounds

- For arbitrary $\mathscr{H}$'s, we have one $\forall$ quantifier over infinite structures (*-continuous KAs).
- This can also be done syntactically, by considering derivations in $\mathbf{KA}_\omega$.
- For monoid equations, one can show that entailment in arbitrary KAs from such $\mathscr{H}$ is equivalent to validity in $\mathsf{REG}(\Sigma^*)/\mathscr{H}$ (i.e., the factor-algebra of the algebra of regular languages).
- Now the validity of $C \to D$ is equivalent to

$$\forall x\big(x \in v(C) \supset \exists y(y \equiv_{\mathscr{H}} x \,\&\, y \in v(D))\big).$$

Upper Bounds

- For arbitrary $\mathscr{H}$'s, we have one $\forall$ quantifier over infinite structures (*-continuous KAs).
- This can also be done syntactically, by considering derivations in $\mathbf{KA}_\omega$.
- For monoid equations, one can show that entailment in arbitrary KAs from such $\mathscr{H}$ is equivalent to validity in $\mathsf{REG}(\Sigma^*)/\mathscr{H}$ (i.e., the factor-algebra of the algebra of regular languages).
- Now the validity of $C \to D$ is equivalent to

$$\forall x\big(x \in v(C) \supset \exists y(y \equiv_{\mathscr{H}} x \,\&\, y \in v(D))\big).$$

- This is Π^0_2.

Upper Bounds

- For arbitrary $\mathcal{H}$'s, we have one $\forall$ quantifier over infinite structures (*-continuous KAs).
- This can also be done syntactically, by considering derivations in $\mathbf{KA}_\omega$.
- For monoid equations, one can show that entailment in arbitrary KAs from such $\mathcal{H}$ is equivalent to validity in $\mathsf{REG}(\Sigma^*)/\mathcal{H}$ (i.e., the factor-algebra of the algebra of regular languages).
- Now the validity of $C \to D$ is equivalent to

$$\forall x \big(x \in v(C) \supset \exists y (y \equiv_{\mathcal{H}} x \mathbin{\&} y \in v(D)) \big).$$

- This is Π^0_2.
- If the equations are length-preserving, then the $\exists y$ quantifier is decidable, thus Π^0_1.

Encoding Turing Machines

- In order to prove lower bounds, we encode **Turing computations,** following Davis (1958).

Encoding Turing Machines

- In order to prove lower bounds, we encode **Turing computations,** following Davis (1958).
- A Turing machine (TM) operates on a **tape,** potentially infinite in both directions, and keeps some finite information in its **state.**

Encoding Turing Machines

- In order to prove lower bounds, we encode **Turing computations,** following Davis (1958).
- A Turing machine (TM) operates on a **tape,** potentially infinite in both directions, and keeps some finite information in its **state.**
- The current **configuration** of a TM will be encoded by $\vdash xqy \dashv$, where x, y are words and q is a state.

Encoding Turing Machines

- In order to prove lower bounds, we encode **Turing computations,** following Davis (1958).
- A Turing machine (TM) operates on a **tape,** potentially infinite in both directions, and keeps some finite information in its **state.**
- The current **configuration** of a TM will be encoded by $\vdash xqy \dashv$, where x, y are words and q is a state.
- Equations encoding the TM:

Encoding Turing Machines

- In order to prove lower bounds, we encode **Turing computations,** following Davis (1958).
- A Turing machine (TM) operates on a **tape,** potentially infinite in both directions, and keeps some finite information in its **state.**
- The current **configuration** of a TM will be encoded by $\vdash xqy \dashv$, where x, y are words and q is a state.
- Equations encoding the TM:
    1. for each transition $(p, a) \to (b, \text{right}, q)$, add $pa = bq$;

Encoding Turing Machines

- In order to prove lower bounds, we encode **Turing computations,** following Davis (1958).
- A Turing machine (TM) operates on a **tape,** potentially infinite in both directions, and keeps some finite information in its **state.**
- The current **configuration** of a TM will be encoded by $\vdash xqy \dashv$, where x, y are words and q is a state.
- Equations encoding the TM:
    1. for each transition $(p, a) \to (b, \text{right}, q)$, add $pa = bq$;
    2. for each transition $(p, a) \to (b, \text{left}, q)$, add $cpa = qcb$ for each letter c;

Encoding Turing Machines

- In order to prove lower bounds, we encode **Turing computations,** following Davis (1958).
- A Turing machine (TM) operates on a **tape,** potentially infinite in both directions, and keeps some finite information in its **state.**
- The current **configuration** of a TM will be encoded by $\vdash xqy \dashv$, where x, y are words and q is a state.
- Equations encoding the TM:
  1. for each transition $(p, a) \to (b, \text{right}, q)$, add $pa = bq$;
  2. for each transition $(p, a) \to (b, \text{left}, q)$, add $cpa = qcb$ for each letter c;
  3. $\vdash \llcorner\!\lrcorner = \vdash$ and $\llcorner\!\lrcorner \dashv = \dashv$.

Encoding Turing Machines

Theorem

If t is the halting state, then $\vdash xqy \dashv \equiv_{\mathscr{H}} \vdash ztw \dashv$ if and only if $\vdash ztw \dashv$ is obtained from $\vdash xqy \dashv$ by the run of our TM.

- This is not obvious, because equiations (unlike TM execution) are undirected.

Encoding Turing Machines

Theorem

If t is the halting state, then $\vdash xqy \dashv \equiv_{\mathscr{H}} \vdash ztw \dashv$ if and only if $\vdash ztw \dashv$ is obtained from $\vdash xqy \dashv$ by the run of our TM.

- This is not obvious, because equiations (unlike TM execution) are undirected.
- However, this is true, due to determinism of the TM and the fact that t is the halting state.

Encoding Turing Machines

Theorem

If t is the halting state, then $\vdash xqy \dashv \equiv_{\mathscr{H}} \vdash ztw \dashv$ if and only if $\vdash ztw \dashv$ is obtained from $\vdash xqy \dashv$ by the run of our TM.

- This is not obvious, because equiations (unlike TM execution) are undirected.
- However, this is true, due to determinism of the TM and the fact that t is the halting state.
- An accurate proof is given, e.g., in Davis' textbook.

Encoding Turing Machines

Theorem

If t is the halting state, then $\vdash xqy \dashv \equiv_{\mathscr{H}} \vdash ztw \dashv$ if and only if $\vdash ztw \dashv$ is obtained from $\vdash xqy \dashv$ by the run of our TM.

- This is not obvious, because equiations (unlike TM execution) are undirected.
- However, this is true, due to determinism of the TM and the fact that t is the halting state.
- An accurate proof is given, e.g., in Davis' textbook.
- Thus, using only monoid equations, we can encode the halting problem, which gives Σ^1_0-hardness even without Kleene star (in particular, for the finitary **KA** theory).

Π^0_2-hardness

- Using Kleene star, we can do more and encode **totality** for Turing machines.

Π^0_2-hardness

- Using Kleene star, we can do more and encode **totality** for Turing machines.
- Totality is a Π^0_2-complete problem, and it means halting on **any** input.

Π^0_2-hardness

- Using Kleene star, we can do more and encode **totality** for Turing machines.
- Totality is a Π^0_2-complete problem, and it means halting on **any** input.
- The input will be a natural number n, encoded by a^n.

Π^0_2-hardness

- Using Kleene star, we can do more and encode **totality** for Turing machines.
- Totality is a Π^0_2-complete problem, and it means halting on **any** input.
- The input will be a natural number n, encoded by a^n.
- The encoding is as follows:

$$\mathscr{H} \vDash \vdash q_0 a^* \dashv \preceq \vdash t \dashv.$$

Π^0_2-hardness

- Using Kleene star, we can do more and encode **totality** for Turing machines.
- Totality is a Π^0_2-complete problem, and it means halting on **any** input.
- The input will be a natural number n, encoded by a^n.
- The encoding is as follows:

$$\mathscr{H} \models \vdash q_0 a^* \dashv \preceq \vdash t \dashv.$$

- For length-preserving equalities, we shall have totality for linearly bounded Turing machines, which models, e.g., totality of context-free grammars (which is Π^0_1-hard).

Π^1_1-hardness

- The more interesting result is Π^1_1-hardness of entailment from arbitrary $\mathscr{H}$'s.

Π^1_1-hardness

- The more interesting result is Π^1_1-hardness of entailment from arbitrary $\mathscr{H}$'s.
- This is also due to Kozen (2002), and the Π^1_1-hard problem encoded is **well-foundedness of infinite recursive graphs** (WF).

Π^1_1-hardness

- The more interesting result is Π^1_1-hardness of entailment from arbitrary $\mathscr{H}$'s.
- This is also due to Kozen (2002), and the Π^1_1-hard problem encoded is **well-foundedness of infinite recursive graphs** (WF).
- Vertices of the graph G are natural numbers and the fact whether there is an edge (m, n) is determined by a TM.

Π^1_1-hardness

- The more interesting result is Π^1_1-hardness of entailment from arbitrary $\mathscr{H}$'s.
- This is also due to Kozen (2002), and the Π^1_1-hard problem encoded is **well-foundedness of infinite recursive graphs** (WF).
- Vertices of the graph G are natural numbers and the fact whether there is an edge (m, n) is determined by a TM.
- The WF problem is to determine whether all paths in G which start from 0 are finite.

Π^1_1-hardness

- In $\mathscr{H}$, we encode the TM which is used for defining graph G.

Π^1_1-hardness

- In $\mathscr{H}$, we encode the TM which is used for defining graph G.
- s is the starting states, t, r, and u are halting states.

Π^1_1-hardness

- In $\mathscr{H}$, we encode the TM which is used for defining graph G.
- s is the starting states, t, r, and u are halting states.
- The machine works as follows:

$$\vdash a^m s a^n \dashv \;\rightarrow^*_M \begin{cases} \vdash a^n t \dashv & \text{if } (m,n) \text{ is an edge} \\ \vdash r \dashv & \text{if not} \end{cases}$$

(u is used when the input is malformed).

Π^1_1-hardness

- In $\mathscr{H}$, we encode the TM which is used for defining graph G.
- s is the starting states, t, r, and u are halting states.
- The machine works as follows:

$$\vdash a^m s a^n \dashv \ \to_M^* \begin{cases} \vdash a^n t \dashv & \text{if } (m, n) \text{ is an edge} \\ \vdash r \dashv & \text{if not} \end{cases}$$

  (u is used when the input is malformed).
- Additionally, we include the inequation

$$t \preceq s a^*$$

  into $\mathscr{H}$.

Π^1_1-hardness

- Now we claim that WF holds iff $\mathscr{H}$ entails $\vdash t \dashv \preceq \vdash r \dashv$.
- Suppose WF holds. This means
  $0 \in W = \{k \mid \text{all paths from } k \text{ are finite}\}$.
- We show that if $k \in W$, then $\mathscr{H}$ entails $\vdash a^k t \dashv \preceq \vdash r \dashv$.
- In the simple case k has just no outgoing edges.
- Then we entail $\vdash a^k t \dashv \preceq \vdash a^k s a^* \dashv$, and next $\vdash a^k s a^* \dashv \preceq \vdash r \dashv$ is obtained by the ω-rule (no (k, m) is an edge).

Π^1_1-hardness

- In the general case, we proceed by transfinite induction on W.
- If $k \in W$, then for each edge (k, m) we have $m \in W$, and, moreover, we may apply induction hypothesis to m.
- Thus, we have $\vdash a^m t \dashv \preceq \vdash r \dashv$.
- Next, the desired $\vdash a^k s a^* \dashv \preceq \vdash r \dashv$ is derived by the ω-rule from $\vdash a^k s a^m \dashv \preceq \vdash r \dashv$.
- If (k, m) is not an edge, it is immediate, otherwise by transitivity with $\vdash a^m t \dashv$.

Π^1_1-hardness

- The "if" direction, from entailment to WF, is performed via a specially constructed *-continuous KA.
- In this *-continuous KA, $\mathscr{H}$ holds, and $\vdash a^k t \dashv \preceq \vdash r \dashv$ implies $k \in W$.
- This KA is a variant of the algebra of languages.
- In fact, for a language A over our alphabet, we define $\sigma(A)$ as the operator which extends A by words obtained using (in)equations from $\mathscr{H}$.
- In particular, $t \preceq sa^*$ allows adding utv, if there is already $usa^k v$ for any k.
- If $x \equiv_{\mathscr{H}_0} y$, where $\mathscr{H}_0$ is the monoid part of $\mathscr{H}$, then x may be replaced by y in one step of σ.
- The closure of A under infinite iteration of σ is denoted by $\overline{A}$.

Π^1_1-hardness

- Now the operations and constants of our KA are defined as follows:

$$A \oplus B = \overline{A \cup B} \qquad 0 = \emptyset$$

$$A \odot B = \overline{A \cdot B} \qquad 1 = \{\varepsilon\}$$

$$A^{\circledast} = \overline{A^*}$$

Π^1_1-hardness

- Now the operations and constants of our KA are defined as follows:

$$A \oplus B = \overline{A \cup B} \qquad\qquad 0 = \emptyset$$
$$A \odot B = \overline{A \cdot B} \qquad\qquad 1 = \{\varepsilon\}$$
$$A^{\circledast} = \overline{A^*}$$

- Each letter a is interpreted as $\overline{\{a\}}$.

Π^1_1-hardness

- Now the operations and constants of our KA are defined as follows:

$$
\begin{aligned}
A \oplus B &= \overline{A \cup B} & \qquad 0 &= \emptyset \\
A \odot B &= \overline{A \cdot B} & \qquad 1 &= \{\varepsilon\} \\
A^{\circledast} &= \overline{A^*} & &
\end{aligned}
$$

- Each letter a is interpreted as $\overline{\{a\}}$.
- This is a *-continuous Kleene algebra, and it satisfies $\mathscr{H}$.

Π^1_1-hardness

- Now the operations and constants of our KA are defined as follows:

$$
\begin{aligned}
A \oplus B &= \overline{A \cup B} & \qquad 0 &= \emptyset \\
A \odot B &= \overline{A \cdot B} & \qquad 1 &= \{\varepsilon\} \\
A^{\circledast} &= \overline{A^*}
\end{aligned}
$$

- Each letter a is interpreted as $\overline{\{a\}}$.
- This is a *-continuous Kleene algebra, and it satisfies $\mathscr{H}$.
- In this algebra, we have $A \preceq B$ if $A \subseteq \overline{B}$.

Π^1_1-hardness

- Now suppose that $\vdash a^k t \dashv \preceq \vdash r \dashv$ is true in this KA.
- This means that $\vdash a^k t \dashv \in \overline{\{\vdash r \dashv\}}$.
- Again, we proceed by transfinite induction on the closure $\overline{\{\vdash r \dashv\}}$.
- $\vdash a^k t \dashv$ was obtained from $\vdash a^k s a^m \dashv$ for all m.
- The latter was either obtained immediately from $\vdash r \dashv$, which means that (k, m) is not an edge, or from $\vdash a^m t \dashv$, to which we apply the induction hypothesis yielding $m \in W$.
- Thus, $k \in W$.

Π^1_1-hardness

- Thus, as we see, adding possibility to derive from hypotheses increases complexity dramatically.
- And, in particular, now the finitary system **KA** (or circular fragment of $\mathbf{HKA}_\infty$) is not a complete axiomatisation.