Kleene Algebra with Hypotheses

Amina Doumane, Denis Kuperberg, Damien Pous, Pierre Pradic

Kleene Algebra with Hypotheses*

Amina Doumane[1,2], Denis Kuperberg[1], Damien Pous[1], Pierre Pradic[1,2]

[1] Univ Lyon, EnsL, UCBL, CNRS, LIP, F-69342, LYON Cedex 07, France.
[2] Warsaw University, MIMUW

Abstract. We study the Horn theories of Kleene algebras and star continuous Kleene algebras, from the complexity point of view. While their equational theories coincide and are PSPACE-complete, their Horn theories differ and are undecidable. We characterise the Horn theory of star continuous Kleene algebras in terms of downward closed languages and we show that when restricting the shape of allowed hypotheses, the problems lie in various levels of the arithmetical or analytical hierarchy. We also answer a question posed by Cohen about hypotheses of the form $1 = S$ where S is a sum of letters: we show that it is decidable.

Keywords: Kleene Algebra · Hypotheses · Horn theory · Complexity.

1 Introduction

Kleene algebras [10,6] are idempotent semirings equipped with a unary operation *star* such that x^* intuitively corresponds to the sum of all powers of x. They admit several models which are important in practice: formal languages, where L^* is the Kleene star of a language L; binary relations, where R^* is the reflexive transitive closure of a relation R; matrices over various semirings, where M^* can be used to perform flow analysis.

A fundamental result is that their equational theory is decidable, and actually PSPACE-complete. This follows from a completeness result which was proved independently by Kozen [11] and Krob and Boffa [17,3], and the fact that checking language equivalence of two regular expressions is PSPACE-complete: given two regular expressions, we have

$$\mathsf{KA} \vdash e \leq f \quad \text{iff} \quad [e] \subseteq [f]$$

(where $\mathsf{KA} \vdash e \leq f$ denotes provability from Kleene algebra axioms, and $[e]$ is the language of a regular expression e).

Because of their interpretation in the algebra of binary relations, Kleene algebras and their extensions have been used to reason abstractly about program

* Full version of the extended abstract to appear in Proc. FoSSaCS 2019. This work has been supported by the European Research Council (ERC) under the European Union's Horizon 2020 programme (CoVeCe, grant agreement No 678157) and by the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).

correctness [12,15,2,9,1]. For instance, if two programs can be abstracted into two relational expressions $(R^*;S)^*$ and $((R \cup S)^*;S)^=$, then we can deduce that these programs are equivalent by checking that the regular expression $(a^*b)^*$ and $(a+b)^*b+1$ denote the same language. This technique made it possible to automate reasoning steps in proof assistants [4,16,19].

In such a scenario, one often has to reason under assumptions. For instance, if we can abstract our programs into relational expressions $(R+S)^*$ and $S^*;R^*$, then we can deduce algebraically that the starting programs are equal if we know that $R;S = R$ (i.e., that S is a no-op when executed after R). When doing so, we move from the equational theory of Kleene algebras to their Horn theory: we want to know whether a given set of equations, the *hypotheses*, entails another equation in all Kleene algebras. Unfortunately, this theory is undecidable in general [13]. In this paper, we continue the work initiated by Cohen [5] and pursued by Kozen [13], by characterising the precise complexity of new subclasses of this general problem.

A few cases have been shown to be decidable in the literature, when we restrict the form of the hypotheses :

- when they are of the form $e = 0$ [5],
- when they are of the form $a \leq 1$ for a a letter [5],
- when they are of the form $1 = w$ or $a = w$ for a a letter and w a word, provided that those equations seen as a word rewriting system satisfy certain properties [18,14]; this includes equations like idempotency ($x = xx$) or self-invertibility ($1 = xx$).

(In the first two cases, the complexity can be shown to remain in PSpace.) We add one positive case, which was listed as open by Cohen [5], and which is typically useful to express that a certain number of predicates cover all cases:

- when hypotheses are of the form $S = 1$ for S a sum of letters.

Conversely, Kozen also studied the precise complexity of various undecidable sub-classes of the problem [13]. For those, one has to be careful about the precise definition of Kleene algebras. Indeed, these only form a quasi-variety (their definition involves two implications), and one often consider $*$-*continuous* Kleene algebras [6], which additionally satisfy an infinitary implication (We define these formally in Sect. 2). While the equational theory of Kleene algebras coincides with that of $*$-continuous Kleene algebras, this is not the case for their Horn theories: there exist Horn sentences which are valid in all $*$-continuous Kleene algebras but not in all Kleene algebras.

Kozen [13] showed for instance that when hypotheses are of the form $pq = qp$ for pairs of letters (p, q), then validity of an implication in all $*$-continuous Kleene algebras is Π^0_1-complete, while it is only known to be ExpSpace-hard for plain Kleene algebras. In fact, for plain Kleene algebras, the only known negative result is that the problem is undecidable for hypotheses of the form $u = v$ for pairs (u, v) of words (Kleene star plays no role in this undecidability result: this is just the word problem). We show that it is already undecidable, and in fact

	$1 = \sum a$	$a \leq \sum b$	$a \leq \sum w$	$a \leq g$
$\mathsf{KA}_H \vdash u \leq f$	Decidable	EXPTIME − complete	Σ_1^0-complete	Σ_1^0-complete
$\mathsf{KA}_H \vdash e \leq f$	Decidable	Undecidable	Σ_1^0-complete	Σ_1^0-complete
$\mathsf{KA}^*_H \vdash u \leq f$	Decidable	EXPTIME − complete	Σ_1^0-complete	Π_1^1-complete
$\mathsf{KA}^*_H \vdash e \leq f$	Decidable	Π_1^0-complete	Π_2^0-complete	Π_1^1-complete

Fig. 1. Summary of the main results.

Σ_1^0-complete when hypotheses are of the form $a \leq S$ where a is a letter and S is a sum of letters. We use a similar encoding as in [13] to relate the Horn theories of KA and KA^* to runs of Turing Machines and alternating linearly bounded automata. This allows us to show that deciding whether an inequality $w \leq f$ holds where w is a word, in presence of sum-of-letters hypotheses, is EXPTIME-complete. We also refine the Π_1^1-completeness result obtained in [13] for general hypotheses, by showing that hypotheses of the form $a \leq g$ where a is a letter already make the problem Π_1^1-complete.

The key notion we define and exploit in this paper is the following: given a set H of equations, and given a language L, write $\mathrm{cl}_H(L)$ for the smallest language containing L such that for all hypotheses $(e \leq f) \in H$ and all words u, v,

$$\text{if} \quad u[f]v \subseteq \mathrm{cl}_H(L) \quad \text{then} \quad u[e]v \subseteq \mathrm{cl}_H(L) \ .$$

This notion makes it possible to characterise the Horn theory of $*$-continuous Kleene algebras, and to approximate that of Kleene algebras: we have

$$\mathsf{KA}_H \vdash e \leq f \quad \Rightarrow \quad \mathsf{KA}^*_H \vdash e \leq f \quad \Leftrightarrow \quad [e] \subseteq \mathrm{cl}_H([f])$$

where $\mathsf{KA}_H \vdash e \leq f$ (resp. $\mathsf{KA}^*_H \vdash e \leq f$) denotes provability in Kleene algebra (resp. $*$-continuous Kleene algebra). We study downward closed languages and prove the above characterisation in Sect. 3.

The first implication can be strengthened into an equivalence in a few cases, for instance when the regular expression e and the right-hand sides of all hypotheses denote finite languages, or when hypotheses have the form $1 = S$ for S a sum of letters. We obtain decidability in those cases (Sect. 4).

Then we focus on cases where hypotheses are of the form $a \leq e$ for a a letter, and we show that most problems are already undecidable there. We do so by exploiting the characterisation in terms of downward closed languages to provide encodings of various undecidable problems on Turing machines, total Turing machines, and linearly bounded automata (Sect. 5).

We summarise our results in Fig. 1. The top of each column restricts the type of allowed hypotheses. Variables e, f stand for general expressions, u, w for words, and a, b for letters. Grayed statements are implied by non-grayed ones.

Notations We let a, b range over the letters of a finite alphabet Σ. We let u, v, w range over the words over Σ, whose set is written Σ^*. We write ϵ for the empty

word; uv for the concatenation of two words u, v; $|w|$ for the length of a word w.We write Σ^+ for the set of non-empty words. We let e, f, g range over the regular expressions over Σ, whose set is written Exp_Σ. We write $[e]$ for the language of such a an expression e: $[e] \subseteq \Sigma^*$. We sometimes implicitly regard a word as a regular expression. If X is a set, $\mathcal{P}(X)$ (resp. $\mathcal{P}_{\text{fin}}(X)$) is the set of its subsets (resp. finite subsets) and $|X|$ for its cardinality.

2 The systems KA and KA*

Definition 1 ($\mathsf{KA}, \mathsf{KA}^*$). *A* Kleene algebra *is a tuple $(M, 0, 1, +, \cdot, *)$ where $(M, 0, 1, +, \cdot)$ is an idempotent semiring and the following axioms and implications, where the partial order $\leq$ is defined by $x \leq y$ if $x + y = y$, hold for all $x, y \in M$.*

$$1 + xx^* \leq x^* \qquad\qquad xy \leq y \;\Rightarrow\; x^*y \leq y$$

$$1 + x^*x \leq x^* \qquad\qquad yx \leq y \;\Rightarrow\; yx^* \leq y$$

A Kleene algebra is $*$-continuous if it satisfies the following implication:

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

A hypothesis *is an inequation of the form $e \leq f$, where e and f are regular expressions. If H is a set of hypotheses, and e, f are regular expressions, we write $\mathsf{KA}_H \vdash e \leq f$ (resp. $\mathsf{KA}^*_H \vdash e \leq f$) if $e \leq f$ is derivable from the axioms and implications of KA (resp. KA^*) as well as the hypotheses from H. We omit the subscript when H is empty.*

Note that the letters appearing in the hypotheses are constants: they are not universally quantified. In particular if $H = \{aa \leq a\}$, we may deduce $\mathsf{KA}_H \vdash a^* \leq a$ but not $\mathsf{KA}_H \vdash b^* \leq b$.

Languages over the alphabet Σ form a $*$-continuous Kleene algebra, as well as binary relations over an arbitrary set.

In absence of hypotheses, provability in KA is coincides with provability in KA^* and with language inclusion:

Theorem 1 (Kozen [11]).

$$\mathsf{KA} \vdash e \leq f \quad\Leftrightarrow\quad \mathsf{KA}^* \vdash e \leq f \quad\Leftrightarrow\quad [e] \subseteq [f]$$

We will classify the theories based on the shape of hypotheses we allow; we list them below (I is a finite non-empty set):

Name of the hypothesis	Its shape		
$(1 = \sum x) -$ hypothesis	$1 = \sum_{i\in I} a_i$	where	$a_i \in \Sigma$
$(w \leq \sum w) -$ hypothesis	$v \leq \sum_{i\in I} v_i$	where	$v, v_i \in \Sigma^*$
$(x \leq \sum w) -$ hypothesis	$a \leq \sum_{i\in I} v_i$	where	$a \in \Sigma, v_i \in \Sigma^*$
$(x \leq \sum x) -$ hypothesis	$a \leq \sum_{i\in I} a_i$	where	$a, a_i \in \Sigma$
$(1 \leq \sum x) -$ hypothesis	$1 \leq \sum_{i\in I} a_i$	where	$a_i \in \Sigma$
$(x \leq 1) -$ hypothesis	$a \leq 1$	where	$a \in \Sigma$

We call *letter hypotheses* any class of hypotheses where the left-hand side is a letter (the last four ones). In the rest of the paper, we study the following problem from a complexity point of view: given a set of C-hypotheses H, where C is one of the classes listed above, and two expressions $e, f \in \mathsf{Exp}_\Sigma$, can we decide whether $\mathsf{KA}_H \vdash e \leq f$ (resp. $\mathsf{KA}^*_H \vdash e \leq f$) holds? We call it the problem of **deciding KA (resp. KA*) under C-hypotheses**.

3 Closure of regular languages

It is known that provability in KA and KA* can be characterised by language inclusions (Thm. 1). In the presence of hypotheses, this is not the case anymore: we need to take the hypotheses into account in the semantics. We do so by using the following notion of *downward closure* of a language.

3.1 Definition of the closure

Definition 2 (H-closure). *Let H be a set of hypotheses and $L \subseteq \Sigma^*$ be a language. The* H-closure of L, *denoted* $\mathrm{cl}_H(L)$, *is the smallest language K such that $L \subseteq K$ and for all hypotheses $e \leq f \in H$ and all words $u, v \in \Sigma^*$, we have*

$$u[f]v \subseteq C \qquad \Rightarrow \qquad u[e]v \subseteq K$$

Alternatively, $\mathrm{cl}_H(L)$ can be defined as the least fixed point of the function $\phi_L : \mathcal{P}(\Sigma^*) \to \mathcal{P}(\Sigma^*)$ defined by $\phi_L(X) = L \cup \psi_H(X)$, where

$$\psi_H(X) = \bigcup_{(e\leq f)\in H} \{u[e]v \mid u, v \in \Sigma^*, u[f]v \subseteq X\}.$$

Example 1. If $H = \{ab \leq ba\}$ then $\mathrm{cl}_H([b^*a^*]) = [(a+b)^*]$. while $\mathrm{cl}_H([a^*b^*]) = [a^*b^*]$.

In order to manipulate closures more conveniently, we introduce a syntactic object witnessing membership in a closure: derivation trees.

Definition 3. *Let H be a set of hypotheses and L a regular language. We define an infinitely branching proof system related to $\mathrm{cl}_H(L)$, where statements are regular expressions, and rules are the following, called respectively* axiom, extension, *and* hypothesis*:*

$$\frac{}{u}\ u \in L \qquad \frac{(u)_{u\in[e]}}{e} \qquad \frac{ufv}{uwv}\ w \in [e],\ e \leq f \in H$$

We write $\vdash_{H,L} e$ if e is derivable in this proof system, i.e. if there is a well-founded tree using these rules, with root e and all leaves labelled by words in L. Such a tree will be called a derivation tree *for $[e] \subseteq \mathrm{cl}_H(L)$ (or $e \in \mathrm{cl}_H(L)$ if e is a word).*

Example 2. The following derivation is a derivation tree for $bababa \in \mathrm{cl}_H([b^*a^*])$, where $H = \{ab \leq ba\}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{bbbaaa}}{bb\boldsymbol{ab}aa}}{bba\boldsymbol{ab}a}}{b\boldsymbol{ab}aba}$$

Derivation trees witness membership to the closure as shown by the following proposition.

Proposition 1. $[e] \subseteq \mathrm{cl}_H(L)$ *iff* $\vdash_{H,L} e$.

3.2 Properties of the closure operator

We summarise in this section some useful properties of the closure. Lem. 1 shows in particular that the closure is idempotent, monotonic (both for the set of hypotheses and its language argument) and invariant by context application. Lem. 2 shows that internal closure operators can be removed in the evaluation of regular expressions.

Lemma 1. *Let $A, B, U, V \subseteq \Sigma^*$. We have*

1. $A \subseteq \mathrm{cl}_H(A)$
2. $\mathrm{cl}_H(\mathrm{cl}_H(A)) = \mathrm{cl}_H(A)$
3. $A \subseteq B$ *implies* $\mathrm{cl}_H(A) \subseteq \mathrm{cl}_H(B)$
4. $H \subseteq H'$ *implies* $\mathrm{cl}_H(A) \subseteq \mathrm{cl}_{H'}(A)$
5. $\mathrm{cl}_H(A) \subseteq \mathrm{cl}_H(B)$ *if and only if* $A \subseteq \mathrm{cl}_H(B)$.
6. $A \subseteq \mathrm{cl}_H(B)$ *implies* $UAV \subseteq \mathrm{cl}_H(UBV)$.

Lemma 2. *Let $A, B \subseteq \Sigma^*$, then*

1. $\mathrm{cl}_H(A + B) = \mathrm{cl}_H(\mathrm{cl}_H(A) + \mathrm{cl}_H(B))$,
2. $\mathrm{cl}_H(AB) = \mathrm{cl}_H(\mathrm{cl}_H(A)\mathrm{cl}_H(B))$,
3. $\mathrm{cl}_H(A^*) = \mathrm{cl}_H(\mathrm{cl}_H(A)^*)$

3.3 Relating closure and provability in KA_H and KA^*_H

We show that provability in KA^* can be characterized by closure inclusions. In KA, provability implies closure inclusions but the converse is not true in general.

Theorem 2. *Let H be a set of hypotheses and e, f be two regular expressions.*

$$\mathsf{KA}_H \vdash e \leq f \qquad \Rightarrow \qquad \mathsf{KA}^*_H \vdash e \leq f \qquad \Leftrightarrow \qquad [e] \subseteq \mathrm{cl}_H([f])$$

Proof. Let $\mathsf{CReg}_{H,\Sigma} = \{\mathrm{cl}_H(L) \mid L \in \mathsf{Reg}_\Sigma\}$, on which we define the following operations:

$$X \oplus Y = \mathrm{cl}_H(X + Y) \qquad X \odot Y = \mathrm{cl}_H(X \cdot Y) \qquad X^{\circledast} = \mathrm{cl}_H(X^*).$$

We define the *closure model* $F_{H,\Sigma} = (\mathsf{CReg}_{H,\Sigma}, \emptyset, \{\epsilon\}, \oplus, \odot, \circledast)$.

We write $\leq$ for the inequality induced by $\oplus$ in $F_{H,\Sigma}$: $X \leq Y$ if $X \oplus Y = Y$.

Lemma 3. *$F_{H,\Sigma} = (\mathsf{CReg}_{H,\Sigma}, \emptyset, \{\epsilon\}, \oplus, \odot, \circledast)$ is a $*$-continuous Kleene algebra. The inequality $\leq$ of $F_{H,\Sigma}$ coincides with inclusion of languages.*

Proof. By Lem. 2, the function $\mathrm{cl}_H : (\mathcal{P}(\Sigma^*), +, \cdot, *) \to (\mathsf{CReg}_{H,\Sigma}, \oplus, \odot, \circledast)$ is a homomorphism. We show that $F_{H,\Sigma}$ is a $*$-continuous Kleene algebra. First, identities of $\mathsf{Lang}_\Sigma = (\mathcal{P}(\Sigma^*), +, \cdot, *)$ are propagated through the morphism cl_H, so only Horn formulas defining $*$-continuous Kleene algebras remain to be verified.It suffices to prove that $F_{H,\Sigma}$ satisfies the $*$-continuity implication, because the implication $xy \leq y \to x^*y \leq y$ and its dual can be deduced from it. Let $A, B, C \in F_{H,\Sigma}$ such that for all $i \in \mathbb{N}$, $A \odot B^{\textcircled{i}} \odot C \leq D$, where $B^{\textcircled{i}} = B \odot \cdots \odot B$. By Lem. 2, $A \odot B^{\textcircled{i}} \odot C = \mathrm{cl}_H(AB^iC)$, so we have $\mathrm{cl}_H(AB^iC) \leq D$, and in particular $AB^iC \leq D$ for all i. By $*$-continuity of Lang_Σ, we obtain $AB^*C \leq D$. By Lem. 1 and using $D = \mathrm{cl}_H(D)$, we obtain $\mathrm{cl}_H(AB^*C) \leq D$ and finally by Lem. 2, $A \odot B^{\circledast} \odot C \leq D$. This achieves the proof that $F_{H,\Sigma}$ is a $*$-continuous Kleene algebra.

Let $A, B \in \mathsf{CReg}_{H,\Sigma}$. We have $A \leq B \Leftrightarrow A \oplus B = B \Leftrightarrow \mathrm{cl}_H(A + B) = B \Leftrightarrow A \subseteq B$. Finally, if $e \leq f$ is a hypothesis from H, then we have $\mathrm{cl}_H[e] \subseteq \mathrm{cl}_H([f])$, so the hypothesis is verified in $F_{H,\Sigma}$. □

The implications $\mathsf{KA}^{(*)}_H \vdash e \leq f \Rightarrow [e] \subseteq \mathrm{cl}_H(f)$ follow from the fact that if an inequation $e \leq f$ is derivable in KA_H (resp. KA^*_H) then it is true in every model, in particular in the model $F_{H,\Sigma}$, thus $\mathrm{cl}_H([e]) \subseteq \mathrm{cl}_H([f])$ or, equivalently. $[e] \subseteq \mathrm{cl}_H([f])$.

Let us prove that for any regular expressions e, f, if $[e] \subseteq \mathrm{cl}_H([f])$ then $\mathsf{KA}^*_H \vdash e \leq f$. Let e, f be two such expressions and let T be a derivation tree for $[e] \subseteq \mathrm{cl}_H([f])$, i.e. witnessing $\vdash_{H,L} e \leq f$. We show that we can transform this tree T into a proof tree in KA^*_H. The extension rule is an occurrence of Lem. 12. Finally, the hypothesis rule is also provable in KA^*_H, using the hypothesis $e \leq f$ together with compatibiliy of $\leq$ with concatenation, and completeness of KA^* for membership of $u \in [e]$. We can therefore build from the tree T a proof in KA^*_H witnessing $\mathsf{KA}^*_H \vdash e \leq f$. □

When we restrict the shape of the expression e to words, and hypotheses to $(w \leq \sum w)$-hypotheses, we get the implication missing from Thm. 2.

Proposition 2. *Let H be a set of $(w \leq \sum w)$-hypotheses, $w \in \Sigma^*$ and $f \in \mathsf{Exp}_\Sigma$.*

$$\mathsf{KA}_H \vdash w \leq f \qquad \Leftrightarrow \qquad w \in \mathrm{cl}_H([f])$$

Proof. Let us show that $w \in \mathrm{cl}_H([f])$ implies $\mathsf{KA}_H \vdash w \leq f$. We proceed by induction on the height of a derivation tree for $w \in \mathrm{cl}_H([f])$. If this tree is just a leaf, then $w \in [f]$ and by Thm. 1 $\mathsf{KA} \vdash w \leq f$. Otherwise, this derivation starts with the following steps:

$$\dfrac{\dfrac{\left(\dfrac{\cdots}{uw_iv}\right)_i}{u(\sum_i w_i)v}}{uwv}\ w \leq \textstyle\sum_i w_i \in H$$

Our inductive assumption is that $\mathsf{KA}_H \vdash uw_iv \leq f$ for all i, thus $\mathsf{KA}_H \vdash \sum_i uw_iv \leq f$. We also have $\mathsf{KA}_H \vdash w \leq (\sum_i w_i)$ hence $\mathsf{KA} \vdash w \leq f$ by distributivity. □

4 Decidability of KA and KA* with $(1 = \sum x)$-hypotheses

In this section, we answer positively the decidability problem of KA_H, where H is a set of $(1 = \sum x)$-hypotheses, posed by Cohen [5]:

Theorem 3. *If H is a set of $(1 = \sum x)$-hypotheses, then KA_H is decidable.*

To prove this theorem we show that in the case of $(1 = \sum x)$-hypotheses:

(P1) $\mathsf{KA}_H \vdash e \leq f$ if and only if $[e] \subseteq \mathrm{cl}_H([f])$.
(P2) $\mathrm{cl}_H([f])$ is regular and we can compute effectively an expression for it.

Decidability of KA_H follows immediately from (P1) and (P2), since it amounts to checking language inclusion for two regular expressions.

To show $(P1)$ and $(P2)$, it is enough to prove the following result:

Theorem 4. *Let H be a set of $(1 = \sum x)$-hypotheses and let f be a regular expression. The language $\mathrm{cl}_H([f])$ is regular and we can compute effectively an expression c such that $[c] = \mathrm{cl}_H([f])$ and $\mathsf{KA}_H \vdash c \leq f$.*

(P2) follows immediatly from Thm. 4. To show (P1), it is enough to prove that $[e] \subseteq \mathrm{cl}_H([f])$ implies $\mathsf{KA}_H \vdash e \leq f$, since the other implication is always true (Thm. 2). Let e, f such that $[e] \subseteq \mathrm{cl}_H([f])$. If c is the expression given by Thm 4, we have $\mathsf{KA}_H \vdash c \leq f$ and $[e] \subseteq [c]$ so by Thm. 1 $\mathsf{KA} \vdash e \leq c$, and this concludes the proof.

To prove Thm. 4, we first show that the closure of $(1 = \sum x)$-hypotheses can be decomposed into the closure of $(x \leq 1)$-hypotheses followed by the closure of $(1 \leq \sum x)$-hypotheses:

Proposition 3 (Decomposition result). *Let $H = \{1 = S_j \mid j \in J\}$ be a set of $(1 = \sum x)$-hypotheses.*

We set $H_{sum} = \{1 \leq S_j \mid j \in J\}$ and $H_{id} = \{a \leq 1 \mid a \in [S_j], j \in J\}$. For every language $L \subseteq \Sigma^*$, we have $\mathrm{cl}_H(L) = \mathrm{cl}_{H_{sum}}(\mathrm{cl}_{H_{id}}(L))$.

Sketch. We show that rules from H_{id} can be locally permuted with rules of H_{sum} in a derivation tree. This allows to compute a derivation tree where all rules from H_{id} occur after (i.e. closer to leaves than) rules from H_{sum}. □

Now, we will show results similar to Thm. 4, but which apply to $(x \leq 1)$-hypotheses and $(1 \leq \sum x)$-hypotheses (Prop. 5 and 6 below). To prove Thm. 4, the idea is to decompose H into H_{id} and H_{sum} using the decomposition property Prop. 3, then applying Prop. 5 and Prop. 6 to H_{id} and H_{sum} respectively.

To show these two propositions, we make use of a result from [7]:

Definition 4. *Let $\mathcal{A} = (Q, \Delta, \iota, F)$ be an NFA , H be a set of hypotheses and $\varphi : Q \to \mathsf{Exp}_\Sigma$ a function from states to expressions. We say that φ is H-*compatible *with $\mathcal{A}$ if:*

- $\mathsf{KA}_H \vdash 1 \leq \varphi(q)$ *whenever* $q \in F$,
- $\mathsf{KA}_H \vdash a\varphi(r) \leq \varphi(q)$ *for all transitions* $(q, a, r) \in \Delta$.

We set $\varphi^{\mathcal{A}} = \varphi(\iota)$.

Proposition 4 ([7]). *Let $\mathcal{A}$ be a NFA, H be a set of hypothesis and φ be a function H-compatible with $\mathcal{A}$. We can construct a regular expression $f_{\mathcal{A}}$ such that:*

$$[f_{\mathcal{A}}] = [\mathcal{A}] \quad \text{and} \quad \mathsf{KA}_H \vdash f_{\mathcal{A}} \leq \varphi^{\mathcal{A}}$$

Proposition 5. *Let H be a set of $(x \leq 1)$-hypotheses and let f be a regular expression. The language $\mathrm{cl}_H([f])$ is regular and we can compute effectively an expression c such that $[c] = \mathrm{cl}_H([f])$ and $\mathsf{KA}_H \vdash c \leq f$.*

Proof. Let $K = \mathrm{cl}_H([f])$ and $\Gamma = \{a \mid (a \leq 1) \in H\}$, we show that K is regular. If $\mathcal{A}$ is a NFA for f, a NFA $\mathcal{A}_{id}$ recognizing K can be built from $\mathcal{A}$ by adding a Γ-labelled loop on every state. It is straightforward to verify that the resulting NFA recognizes K, by allowing to ignore any letter from Γ.

For every $q \in Q$, let f_q be a regular expression such that $[f_q] = [q]_{\mathcal{A}}$, where $[q]_{\mathcal{A}}$ denotes the language accepted from q in $\mathcal{A}$. Let $\varphi : Q \to \mathsf{Exp}_\Sigma$ which maps each state q of $\mathcal{A}_{id}$ (which is also a state of $\mathcal{A}$) to $\varphi(q) = f_q$. Let us show that φ is H-compatible with $\mathcal{A}$. If $q \in F$, then $1 \in [f_q]$, so by completeness of KA, we have $\mathsf{KA} \vdash 1 \leq f_q$. Let (p, a, q) be a transition of $\mathcal{A}_{id}$. Either $(p, a, q) \in \Delta$, in which case we have $a[f_q] \subseteq [f_p]$, and so by Thm. 1 $\mathsf{KA} \vdash af_q \leq f_p$. Or $p = q$ (this transition is a loop that we added). Then $\mathsf{KA}_H \vdash a \leq 1$, so $\mathsf{KA}_H \vdash af_p \leq f_p$, and this concludes the proof.

By Prop. 4, we can now construct a regular expression c which satisfies the desired properties. □

Definition 5. *Let Γ be a set of letters. A language L is said to be* Γ-closed *if:*

$$\forall u, v \in \Sigma^*, \forall a \in \Gamma \qquad uv \in L \qquad \Rightarrow \qquad uav \in L$$

If $H = \{1 \leq S_i \mid i \in I\}$ is a set of $(1 \leq \sum x)$-hypotheses, we say that a language L is H-closed if if it is Γ-closed where $\Gamma = \cup_{i \in I}[S_i]$.

Remark 1. If H is a set of $(x \leq 1)$-hypothesis, and $\Gamma = \{a \mid (a \leq 1) \in H\}$, then $\mathrm{cl}_H(L)$ is Γ-closed for every language L.

Proposition 6. *Let H be a set of $(1 \leq \sum x)$-hypotheses and let f be a regular expression whose language is H-closed. The language $\mathrm{cl}_H([f])$ is regular and we can compute effectively an expression c such that $[c] = \mathrm{cl}_H([f])$ and $\mathsf{KA}_H \vdash c \leq f$.*

Proof. We set $L = [f]$, $H = \{1 \leq S_j \mid j \in J\}$ and $\Gamma = \{a \mid a \in [S_j], j \in J\}$.

Let us show that $\mathrm{cl}_H(L)$ is regular. The idea is to construct a set of words $L_\sharp$, where each word $u_\sharp$ is obtained from a word u of $\mathrm{cl}_H(L)$, by adding at the position where a rule $(1 \leq S_j)$ is applied in the derivation tree for $\mathrm{cl}_H(L) \vdash u$, a new symbol $\sharp_j$. We will show that this set satisfies the two following properties:

- $\mathrm{cl}_H(L)$ is obtained from $L_\sharp$ by erasing the symbols $\sharp_j$.
- $L_\sharp$ is regular.

Since the operation that erases letters preserves regularity, we obtain as a corolary that $\mathrm{cl}_H(L)$ is regular.

Let us now introduce more precisely the language $L_\sharp$ and show the properties that it satisfies. Let $\Theta_\sharp = \{\sharp_j \mid j \in J\}$ be a set of new letters and $\Sigma_\sharp = \Sigma \cup \Theta_\sharp$ be the alphabet Σ enriched with these new letters.

We define the function $exp : \Sigma_\sharp \to \mathcal{P}(\Sigma)$ that expands every letter $\sharp_j$ into the sum of the letters corresponding to its rule in H as follows:

$$\begin{array}{ll} exp(a) = a & \text{if } a \in \Sigma \\ exp(\sharp_j) = \{a \mid a \in [S_j]\} & \forall j \in J \end{array}$$

This function can naturally be extended to $exp : (\Sigma_\sharp)^* \to \mathcal{P}(\Sigma^*)$.
If $L \subseteq \Sigma^*$, we define $L_\sharp \subseteq (\Sigma_\sharp)^*$ as follows:

$$L_\sharp = exp^{-1}(\mathcal{P}(L)) = \{u \in (\Sigma_\sharp)^* \mid exp(u) \subseteq L\}$$

We define the morphism $\pi : (\Sigma_\sharp)^* \to \Sigma^*$ that erases the letters from $\Theta_\sharp$ as follows: $\pi(a) = a$ if $a \in \Sigma$ and $\pi(\sharp_j) = \epsilon$ for all $j \in J$. Our goal is to prove that $\mathrm{cl}_H(L) = \pi(L_\sharp)$ and that $L_\sharp$ is regular. To prove the first part, we need an alternative presentation of $L_\sharp$ as the closure of a new set of hypotheses $H_\sharp$ which we define as follows:

$$H_\sharp = \{\sharp_j \leq S_j \mid j \in J\} \cup \{\sharp_j \leq 1 \mid j \in J\}$$

Lemma 4. *We have $L_\sharp = \mathrm{cl}_{H_\sharp}(L)$. In particular $L_\sharp$ is $\Theta_\sharp$-closed.*

See [8, App. B] for a detailed proof of Lem. 4.

Lemma 5. $\mathrm{cl}_H(L) = \pi(L_\sharp)$.

Proof. If $u \in \pi(L_\sharp)$, let $v \in L_\sharp$ such that $u = \pi(v)$. By Lem. 4, there is a derivation tree T_v for $v \in \mathrm{cl}_{H_\sharp}(L)$. Erasing all occurences of $\sharp_j$ in T_v yields a derivation tree for $u \in \mathrm{cl}_H(L)$.

Conversely, if $u \in \mathrm{cl}_H(L)$ is witnessed by some derivation tree T_u, we show by induction on T_u that there exists $v \in L_\sharp \cap \pi^{-1}(u)$. If T_u is a single leaf, we have $u \in L$, and therefore it suffices to take $v = u$.

Otherwise, the rule applied at the root of T_u partitions u into $u = wz$, and has premises $\{wbz \mid b \in [S_j]\}$ for some $j \in J$ and $w, z \in \Sigma^*$. By induction hypothesis, for all $b \in [S_j]$, there is $v_b \in L_\sharp \cap \pi^{-1}(wbz)$. Let $w = w_1 \dots w_n$ and $z = z_1 \dots z_m$ be the decompositions of w, z into letters of Σ. By definition of π, for all $b \in [S_j]$, v_b can be written $v_b = \alpha_{b,1} w_1 \alpha_{b,2} w_2 \dots w_n \alpha_{b,n} b \alpha_{b,n+1} z_1 \alpha_{b,n+2} \dots z_m \alpha_{b,n+m+3}$, with $\alpha_{b,0} \dots \alpha_{b,n+m+3} \in (\Theta_\sharp)^*$. For each $k \in [0, n+m+3]$, let $\alpha_k = \Pi_{b \in [S_j]} \alpha_{b,k}$. Let $w' = \alpha_0 w_1 \alpha_1 \dots w_n \alpha_{n+1}$ and $z' = \alpha_{n+2} z_1 \alpha_{n+3} \dots z_m \alpha_{n+m+3}$ By Lem. 4, $L_\sharp$ is $\Theta_\sharp$-closed, so for each $b \in [S_j]$ the word $v'_b = w' b z'$ is in $L_\sharp$, since v'_b is obtained from v_b by adding letters from $\Theta_\sharp$. We can finally build $v = w' \sharp_j z'$. We have $exp(v) = \bigcup_{b \in [S_j]} exp(v'_b) \subseteq L$, and $\pi(v) = \pi(w')\pi(z') = wz = u$. □

Lemma 6. *$L_\sharp$ is a regular language, computable effectively.*

Sketch. From a DFA $\mathcal{A} = (\Sigma, Q, q_0, F, \delta)$ for for L, we first build a DFA $\mathcal{A}_\wedge = (\Sigma, \mathcal{P}(Q), q_0, \mathcal{P}(F), \delta_\wedge)$,which corresponds to a powerset construction, except that accepting states are $\mathcal{P}(F)$. This means that the semantic of a state P is the conjunction of its members. We then build $\mathcal{A}_\sharp = (\Sigma, \mathcal{P}(Q), q_0, \mathcal{P}(F), \delta_\sharp)$ based on $\mathcal{A}_\wedge$, which can additionally read letters of the form $\sharp_j$, by expanding them using the powerset structure of $\mathcal{A}_\wedge$. □

Lemma 7. *We can construct a regular expression c such that $[c] = \mathrm{cl}_H(L)$ and $\mathsf{KA}_H \vdash c \leq f$.*

Proof. Let $\mathcal{A}_\sharp$ be the DFA constructed for $L_\sharp$ in the proof of Lem. 6. We will use the notations of this proof in the following.

Let $\pi(\mathcal{A}_\sharp) = (\Sigma, \mathcal{P}(Q), q_0, \mathcal{P}(F), \pi(\delta_\sharp))$ be the NFA obtained from $\mathcal{A}_\sharp$ by replacing every transition $\delta_\sharp(P, \sharp_j) = R$, where $j \in J$, by a transition $\pi(\delta_\sharp)(P, \epsilon) = R$. By Lem. 5, the automaton $\pi(\mathcal{A}_\sharp)$ recognizes the language $\mathrm{cl}_H(L)$. Let us construct a regular expression c for this automaton such that $\mathsf{KA}_H \vdash c \leq f$.

For every $P \in \mathcal{P}(Q)$, let f_P be a regular expression such that $[f_P] = [P]_{\mathcal{A}_\wedge}$.

Let $\varphi : \mathcal{P}(Q) \to \mathsf{Exp}_\Sigma$ be the function which maps each state P of $\pi(\mathcal{A}_\sharp)$ to $\varphi(P) = f_P$. Let us show that φ is H-compatible.

If $P \in \mathcal{P}(F)$, then P is a final state of $\mathcal{A}_\wedge$, so $1 \in [f_P]$, and by completeness of KA, $\mathsf{KA} \vdash 1 \leq f_P$. Let $(P, a, R) \in \pi(\Delta_\sharp)$. Either $a \in \Sigma$, so $(P, a, R) \in \Delta_\wedge$ and $a[f_R] \subseteq [f_P]$, so by Thm. 1 $\mathsf{KA} \vdash a f_R \leq f_P$. Or $a = \epsilon$ so there is $j \in J$ such that $(P, \sharp_j, R) \in \Delta_\sharp$. This means that $R = \cup_{b \in [S_j]} R_b$ where $\delta_\wedge(P, b) = R_b, \forall b \in [S_j]$. We have then that $b[f_{R_b}] \subseteq [f_P]$ for all $b \in [S_j]$. Note that for all $b \in [S_j]$, $R_b \subseteq R$, so $[f_R] \subseteq [f_{R_b}]$ and then $S_j[f_R] \subseteq [f_P]$. By Thm. 1 $\mathsf{KA} \vdash S_j f_R \leq f_P$. We have also that $\mathsf{KA}_H \vdash \sharp_j \leq S_j$, so $\mathsf{KA}_H \vdash \sharp_j f_R \leq f_P$.

By Prop. 4, we can construct the desired regular expression c. □

□

5 Complexity results for letter hypotheses

In this section, we give a recursion-theoretic characterization of KA_H and KA^*_H where H is a set of letter hypotheses or $(w \leq \sum w)$-hypotheses. In all the section, by "deciding $\mathsf{KA}^{(*)}_H$" we mean deciding whether $\mathsf{KA}^{(*)}_H \vdash e \leq f$, given e, f, H as input.

Theses various complexity classes will be obtained by reduction from some known problems concerning Turing Machines (TM) and alternating linearly bounded automata (LBA), such as halting problem and universality.

To obtain these reductions, we build on a result which bridges TMs and LBAs on one hand and closures on the other: the set of co-reachable configurations of a TM (resp. LBA) can be seen as the closure of a well-chosen set of hypotheses.

We present this result in Section 5.1, and show in Section 5.2 how to instantiate it to get our complexity classes.

5.1 Closure and co-reachable states of TMs and LBAs

Definition 6. *An* alternating Turing Machine *over Σ is a tuple $\mathcal{M} = (Q, Q_F, \Gamma, \iota, B, \Delta)$ consisting of a finite set of states Q and final states $Q_F \subseteq Q$, a finite set of states Q, a finite working alphabet $\Gamma \supseteq \Sigma$, an initial state $\iota \in Q$, $B \in \Gamma$ the blank symbol and a transition function $\Delta : (Q \setminus Q_F) \times \Gamma \to \mathcal{P}(\mathcal{P}(\{L, R\} \times \Gamma \times Q))$. Let $\#_L, \#_R \notin \Gamma$ be fresh symbols to mark the ends of the tape, and $\Gamma_\# = \Gamma \cup \{\#_L, \#_R\}$.*

A configuration *is a word $uqav = \#_L \Gamma^* Q \Gamma^+ \#_R$, where $\#_L$ and $\#_R$ are special symbols not in Γ, meaning that the head of the TM points to the letter a. We denote by C the set of configurations of $\mathcal{M}$. A configuration is* final *if it is of the form $\#_L \Gamma^* Q_F \Gamma^+ \#_L$.*

The execution of the TM $\mathcal{M}$ over input $w \in \Sigma$ may be seen as a game-like scenario between two players ∃loise *and* ∀belard *over a graph $C \sqcup (C \times \mathcal{P}(\{L, R\} \times \Gamma \times Q))$, with initial position ιw which proceeds as follows.*

- *over a configuration $uqav$ with $a \in \Gamma$, $u, v \in \Gamma^*_\#$,* ∃loise *picks a transition $X \in \Delta(q, a)$ to move to position $(uqav, X)$*
- *over a position $(uqav, X)$ with $a \in \Gamma$, $u, v \in \Gamma^*$,* ∀belard *picks a triple $(d, c, r) \in X$ to move in configuration*
  - *$ucrB\#_R$ if $v = \#_R$ and $d = R$*
  - *$ucrv$ if $v \neq \#_R$ and $d = R$*
  - *$\#_L rBcv$ if $u = \#_L$ and $d = L$*
  - *$u'rbcv$ if $u = \#_R u'b$ and $d = L$*

Given a subset of configurations $D \subseteq C$, we define $\mathrm{Attr}^{\exists\text{loise}}(D)$ *the* ∃loise *attractor for D as the set of configurations from which* ∃loise *may force the execution to go through D.*

A deterministic *TM $\mathcal{M}$ is one where every $\Delta(q,a) \subseteq \{\{(d,c,r)\}\}$ for some $(d,c,r) \in \{L,R\} \times \Gamma \times Q$ In such a case, we may identify $\mathcal{M}$ with the underlying partial function $[\mathcal{M}] : \Sigma^* \rightharpoonup Q_F$.*

An alternating linearly bounded automaton *over the alphabet Σ is a tuple $\mathcal{A} = (Q, Q_F, \Gamma, \iota, \Delta)$ where $(Q, Q_F, \Gamma \sqcup \{B\}, \iota, B, \Delta)$ is a TM that does not insert B symbols. This means that the head can point to $\sharp_d$, and for every $X \in \Delta(q, \#_d)$ and $(d', a, r) \in X$, we have $d \neq d'$ and $a = \#_d$.*

An LBA is deterministic if its underlying TM is.

Definition 7. *A set of $(w \leq \sum w)$-hypotheses is said to be* length-preserving *if for every $(v \leq \sum_{i \in I} v_i) \in H$, we have that $|v| = |v_i|$ for all $i \in I$.*

The following lemma generalizes a similar construction from [13].

Lemma 8. *For every TM $\mathcal{M}$ of working alphabet Γ, there exists a set of $(w \leq \sum w)$-hypotheses $H_{\mathcal{M}}$ over the alphabet $\Theta = Q \cup \Gamma$ such that, for any set of configurations $D \subseteq C$ we have that: $\mathrm{cl}_{H_{\mathcal{A}}}(D) = \mathrm{Attr}^{\exists \mathrm{loise}}(D)$. Furthermore, this reduction is polytime computable, and $H_{\mathcal{A}}$ is length-preserving if $\mathcal{M}$ is an LBA.*

A configuration c is *co-reachable* if ∃loise has a strategy to reach a final configuration from c. Lem. 8 shows that the set of co-reachable configurations can be seen as the closure by $(w \leq \sum w)$-hypotheses. Since we are also interested in $(x \leq \sum x)$-hypotheses, we will show that $(w \leq \sum w)$ hypotheses can be transformed into letter hypotheses. Moreover, this transformation preserves the length-preserving property.

Theorem 5. *Let Σ be an alphabet, H be a set of $(w \leq \sum w)$-hypotheses over Σ. There exists an extended alphabet $\Sigma' \supseteq \Sigma$, a set of $(x \leq \sum w)$-hypotheses H' over Σ' and a regular expression $h \in \mathsf{Exp}_{\Sigma'}$ such that the following holds for every $f \in \mathsf{Exp}_{\Sigma}$ and $w \in \Sigma^*$.*

$$w \in \mathrm{cl}_H([f]) \qquad \textit{if and only if} \qquad w \in \mathrm{cl}_{H'}([f + h])$$

Furthermore, we guarantee the following:

- *(Σ', H', h) can be computed in polynomial time from (Σ, H).*
- *H' is length-preserving whenever H is.*

5.2 Complexity results

Lemma 9. *If H is a set of length-preserving $(w \leq \sum w)$-hypotheses (resp. a set of $(x \leq \sum x)$-hypotheses), $w \in \Sigma^*$ and $f \in \mathsf{Exp}_{\Sigma}$, deciding $\mathsf{KA}_H \vdash w \leq f$ is* EXPTIME − complete.

Proof. We actually show that our problem is complete in alternating-PSPACE (APSPACE), which enables us to conclude as EXPTIME and APSPACE coincide. First, notice that by completeness of KA_H over this fragment (Prop. 2), we have $\mathsf{KA}_H \vdash w \leq f \Leftrightarrow w \in \mathrm{cl}_H([f])$. Hence, we work directly with the latter

notion. It suffices to show hardness for the $(x \leq \sum x)$ case and membership for the $(w \leq \sum w)$ case.

Given an arbitrary alternating Turing Machine $\mathcal{M}$ in APSPACE there exists a polynomial $p \in \mathbb{N}[X]$ such that executions of $\mathcal{M}$ over words w are bisimilar to executions of the LBA($\mathcal{M}$) over $wB^{p(|w|)}$. Hence, by Lem. 8 and Thm. 5, the problem with $(x \leq \sum x)$-hypotheses is APSPACE-hard. Conversely, we may show that our problem with $(w \leq \sum w)$-hypotheses falls into APSPACE. On input w, the alternating algorithm first checks whether $w \in [f]$ in linear time. If it is the case, it returns "yes". Otherwise, it non-deterministically picks a factorization $w = uxv$ with $x \in \Sigma^*$ and a hypothesis $x \leq \sum_i y_i$. It then universally picks $y_i \in \Sigma^{|x|}$, and replaces x by y_i on the tape, so that the new tape content is $w' = uy_iv$. Then the algorithm loops back to its first step. In parallel, we keep track of the number of steps and halt by returning "no" as soon as we reach $|\Sigma|^{|w|}$ steps. This is correct because, if there is a derivation tree witnessing $w \in \mathrm{cl}_H([f])$, there is one where on every path, all nodes have distinct labels, so the nondeterministic player can play according to this tree, while the universal player selects a branch. □

Theorem 6. *Deciding* KA^*_H *is* Π^0_1-complete *for* $(x \leq \sum x)$*-hypotheses.*

Proof. By Lem. 9 and the fact that regular expressions are in recursive bijection with natural numbers, our set is clearly Π^0_1. To show completeness, we effectively reduce the set of universal LBAs, which is known to be Π^0_1-complete, to our set of triples. Indeed, by Lem. 8, an LBA $\mathcal{A}$ is universal if and only if $\#_L\{\iota\}\Sigma^*\#_R \subseteq \mathrm{cl}_H(C_F)$ where C_F is the set of final configurations. □

Theorem 7. *If* H *is a set of* $(x \leq \sum w)$*-hypotheses,* $w \in \Sigma^*$ *and* $f \in \mathsf{Exp}_\Sigma$*, deciding* $\mathsf{KA}^{(*)}_H \vdash w \leq f$ *is* Σ^0_1-complete*.*

Proof. As KA_H is a recursively enumerable theory, our set is Σ^0_1. By the completeness theorem (Prop. 2), we have $\mathsf{KA}_H \vdash w \leq f \Leftrightarrow \mathsf{KA}^*_H \vdash w \leq f \Leftrightarrow w \in \mathrm{cl}_H([f])$, so we may work directly with closure. In order to show completeness, we reduce the halting problem for Turing machines (on empty input) to this problem. Let $\mathcal{M}$ be a Turing machine with alphabet Σ and final state q_f, and $H_\mathcal{M}$ be the set of $(w \leq \sum w)$-hypotheses given effectively by Lem. 8. Let $f = \Sigma^* q_f \Sigma^*$, by Lem. 8 we have $\mathcal{M}$ halts on empty input if and only if $q_0 \in \mathrm{cl}_{H_\mathcal{M}}(f)$. Notice that hypotheses of H' are of the form $u \leq V$ where $u \in \Theta^3$ and $V \subseteq \Theta^3$. By Thm. 5, we can compute a set H' of $(x \leq \sum x)$-hypotheses, and an expression h on an extended alphabet such that $q_0 \in \mathrm{cl}_{H_\mathcal{M}}([f]) \Leftrightarrow q_0 \in \mathrm{cl}_{H'}([f + h])$. □

Theorem 8. *Deciding* KA^*_H *is* Π^0_2-complete *for* $(x \leq \sum w)$*-hypotheses.*

Proof. This set is Π^0_2 by Thm. 7. It is complete by reduction from the set of Turing Machines accepting all inputs, which is known to be Π^0_2. Indeed, let $\mathcal{M}$ be a Turing Machine on alphabet Σ with final state q_f, by Lem. 8, we can compute a set of $(w \leq \sum w)$-hypotheses $H_\mathcal{M}$ with finite language in second components such that $c \in \mathrm{cl}_{H_\mathcal{M}}(c')$ if and only if configuration c' is reachable

from c. As before, by Thm. 5, we can compute a set of letter hypotheses H' with finite languages in second components, and a regular expression h on an extended alphabet, such that for any $\mathrm{cl}_{H'}([f + h]) \cap \Theta^* = \mathrm{cl}_H([f])$ for any $f \in \mathsf{Exp}_\Theta$. Let $C_f = \Sigma^* q_f \Sigma^*$, we obtain that $\mathcal{M}$ accepts all inputs if and only if $[q_0\Sigma^*] \subseteq \mathrm{cl}_{H'}([C_f + h])$, which achieves the proof of Π^0_2-completeness. □

Theorem 9. *Deciding* KA^*_H *is* Π^1_1-complete *for* $(x \leq g)$*-hypotheses* $(g \in \mathsf{Exp}_\Sigma)$*.*

Sketch. It is shown in [13] that the problem is complete with hypotheses of the form $H = H_w \cup \{x \leq g\}$, where H_w is a set of length-preserving $(w \leq \sum w)$ hypotheses. A slight refinement of Thm. 5 allows us to reduce this problem to hypotheses of the form $x \leq g$. □

5.3 Undecidability of KA_H for sums of letters

Fix an alphabet Σ, a well-behaved coding function $\lceil\cdot\rceil$ of Turing machines with final states $\{0, 1\}$ into Σ^* and a recursive pairing function $\langle\cdot,\cdot\rangle : \Sigma^* \times \Sigma^* \to \Sigma^*$. A *universal total* $F : \Sigma^* \to \{0, 1\}$ is a function such that, for every total Turing machine $\mathcal{M}$ and input $w \in \Sigma^*$ we have $F(\langle\lceil\mathcal{M}\rceil, w\rangle) = [M](w)$. In particular, F should be total and is not uniquely determined over codes of partial Turing machines. The next folklore lemma follows from an easy diagonal argument.

Lemma 10. *There is no universal total Turing machine.*

Our strategy is to show that decidability of KA_H with $(x \leq \sum x)$ hypotheses would imply the existence of a universal total TM. To do so, we need one additional lemma.

Lemma 11. *Suppose that* $\mathcal{M} = (Q, Q_F, \Gamma, \iota, B, \Delta)$ *is a total Turing machine with final states* $\{0, 1\}$ *and initial state* ι*. Let* $w \in \Sigma^*$ *be an input word for* $\mathcal{M}$*.*

Then there is effectively a set of length-preserving $(w \leq \sum w)$*-hypotheses* H *and expressions* e_w, h *such that* $[\mathcal{M}](w) = 1$ *if and only if* $\mathsf{KA}_H \vdash e_w \leq h$

Theorem 10. KA_H *is undecidable for* $(x \leq \sum x)$*-hypotheses.*

Proof. Assume that KA_H is decidable. This means that we have an algorithm $\mathcal{A}$ taking tuples (Σ, w, f, H), with H consisting only of sum-of-letters hypotheses and returning true when $\mathsf{KA}_H \vdash w \leq f$ and false otherwise. Without loss of generality, we can assume that $\mathcal{A}$ is total. By Thm. 5, we may even provide an algorithm $\mathcal{A}'$ taking as input tuples (w, f, H) where H is a set of length-preserving $(w \leq \sum w)$-hypotheses with a similar behaviour: $\mathcal{A}'$ returns true when $\mathsf{KA}_H \vdash w \leq f$ and false otherwise.

Given $\mathcal{A}'$, consider $\mathcal{M}$ defined so that $[\mathcal{M}](\lceil\mathcal{N}\rceil, w) = [\mathcal{A}'](e_w, h, H)$. where the last tuple is given by Lem. 11. We show that $\mathcal{M}$ is a total universal Turing machine. Since such a machine cannot exist by Lem. 10, this is enough to conclude. Since $\mathcal{A}'$ is total, so is $\mathcal{M}$. For total Turing Machines $\mathcal{N}$, Lem. 11 guarantees that $[\mathcal{N}](w) = 1$ if and only if $[\mathcal{A}'](e_w, h, H) = [\mathcal{M}](\lceil\mathcal{N}\rceil, w) = 1$. Since both $[\mathcal{A}']$ and $[\mathcal{M}]$ are total with codomain $\{0, 1\}$, we really have $[\mathcal{M}](\lceil\mathcal{N}\rceil, w) = [\mathcal{N}](w)$. □

References

1. C. J. Anderson, N. Foster, A. Guha, J.-B. Jeannin, D. Kozen, C. Schlesinger, and D. Walker. NetKAT: semantic foundations for networks. In *Proc. POPL*, pages 113–126. ACM, 2014.
2. A. Angus and D. Kozen. Kleene algebra with tests and program schematology. Technical Report TR2001-1844, CS Dpt., Cornell University, July 2001.
3. M. Boffa. Une remarque sur les systèmes complets d'identités rationnelles. *Informatique Théorique et Applications*, 24:419–428, 1990.
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. E. Cohen. Hypotheses in Kleene algebra. Technical report, Bellcore, Morristown, N.J., 1994.
6. J. H. Conway. *Regular algebra and finite machines*. Chapman and Hall, 1971.
7. A. Das, A. Doumane, and D. Pous. Left-handed completeness for kleene algebra, via cyclic proofs. In *Proc. LPAR*, volume 57 of *EPiC Series in Computing*, pages 271–289. EasyChair, 2018.
8. A. Doumane, D. Kuperberg, D. Pous, and P. Pradic. Kleene algebra with hypotheses. Full version of this extended abstract, available at `https://hal.archives-ouvertes.fr/TOBEFILLED`, 2019.
9. 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.
10. S. C. Kleene. Representation of events in nerve nets and finite automata. In *Automata Studies*, pages 3–41. Princeton University Press, 1956.
11. D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. *Inf. and Comp.*, 110(2):366–390, 1994.
12. D. Kozen. On Hoare logic and Kleene algebra with tests. *ACM Trans. Comput. Log.*, 1(1):60–76, 2000.
13. D. Kozen. On the complexity of reasoning in Kleene algebra. *Inf. and Comp.*, 179:152–162, 2002.
14. D. Kozen and K. Mamouras. Kleene algebra with equations. In *Proc. ICALP*, volume 8573 of *LNCS*, pages 280–292. Springer, 2014.
15. 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.
16. A. Krauss and T. Nipkow. Proof pearl: Regular expression equivalence and relation algebra. *JAR*, 49(1):95–106, 2012.
17. D. Krob. Complete systems of B-rational identities. *TCS*, 89(2):207–343, 1991.
18. K. Mamouras. *Extensions of Kleene Algebra for Program Verification*. PhD thesis, Cornell University, Ithaca, NY, 2015.
19. D. Pous. Kleene Algebra with Tests and Coq tools for while programs. In *Proc. ITP*, volume 7998 of *LNCS*, pages 180–196. Springer, 2013.

A Proofs of section 3

Lemma 12. *Let $(M, +, \cdot, *)$ be a model of KA^*, and $\sigma : \Sigma \to M$ be an interpretation of letters in M, naturally extended to $\bar{\sigma} : \mathsf{Exp}_\Sigma \to M$. Then for all $x, y, z \in M$ and $e \in \mathsf{Exp}_\Sigma$, we have:*

$$\frac{\forall u \in [e], x\bar{\sigma}(u)y \leq z}{x\bar{\sigma}(e)y \leq z}$$

Proof. We prove this by induction on $(h(e), |e|)$, where $h(e)$ is the star-height of e, i.e. the number of nested Kleene stars in e. If $h(e) = 0$, i.e. $[e]$ is a finite language, the result is trivial, using rules of idempotent semirings.

If $e = e_1 + e_2$, the assumption $\forall u \in [e], x\bar{\sigma}(u)y \leq z$ entails $\forall u \in [e_i], x\bar{\sigma}(u)y \leq z$ for $i \in \{1, 2\}$. By induction hypothesis we obtain that $x\bar{\sigma}(e_1)y \leq z$ and $x\bar{\sigma}(e_2)y \leq z$, hence $x\bar{\sigma}(e)y \leq z$.

If $e = e_1 e_2$, noting IH_{e_i} for the induction hypothesis on e_i, we have

$$\dfrac{\dfrac{\dfrac{\dfrac{\forall u \in [e], x\bar{\sigma}(u)y \leq z}{\forall u_1 \in [e_1], \forall u_2 \in [e_2], x\bar{\sigma}(u_1)\bar{\sigma}(u_2)y \leq z}\,\mathrm{IH}_{e_2}}{\forall u_1 \in [e_1], x\bar{\sigma}(u_1)\bar{\sigma}(e_2)y \leq z}\,\mathrm{IH}_{e_1}}{x\bar{\sigma}(e_1)\bar{\sigma}(e_2)y \leq z}}{x\bar{\sigma}(e)y \leq z}$$

Finally, if $e = f^*$, we have

$$\dfrac{\dfrac{\dfrac{\forall u \in [e], x\bar{\sigma}(u)y \leq z}{\forall i \in \mathbb{N}, \forall u \in [f^i], x\bar{\sigma}(u)y \leq z}\,\mathrm{HI}_{f^i}}{\forall i \in \mathbb{N}, x\bar{\sigma}(f^i)y \leq z}\,*\text{-continuity}}{x\bar{\sigma}(e)y \leq z}$$

□

We prove Prop. 1:

Proposition 1. $[e] \subseteq \mathrm{cl}_H(L)$ *iff* $\vdash_{H,L} e$.

Proof. Assume $[e] \subseteq \mathrm{cl}_H(L)$. This means there is an ordinal α such that $[e] \subseteq \phi_L^\alpha(\emptyset)$, by definition of $\mathrm{cl}_H(L)$ as the least fixed point of ϕ_L and Knaster-Tarski theorem. We prove by transfinite induction on α that $\mathrm{cl}_H(L) \vdash_{H,L} e$, i.e. there is a derivation tree for $[e] \subseteq \mathrm{cl}_H(L)$. The case $\alpha = 0$ is trivial, as $\phi_L^0(\emptyset) = \emptyset$. If $\alpha > 0$, we get that $[e] \subseteq \phi_L(\bigcup_{\beta<\alpha} \phi_L^\beta(\emptyset))$. We build the tree T in the following way: let $w \in [e]$, we want to build a tree T_w for $w \in \mathrm{cl}_H(L)$. If $w \in L$, then the tree T_w is just the single leaf w, an axiom of the system. If $w \in \phi_L^\beta(\emptyset)$ for some $\beta < \alpha$, we can conclude by induction hypothesis. The last possibility is $w \in \psi_H(\phi_L^\beta(\emptyset)$ for some $\beta < \alpha$. This means that there is an hypothesis $e_H \leq f_H \in H$ and words $u, v \in \Sigma^*$ such that $w \in u[e_H]v$, and $u[f_H]v \subseteq \phi_L^\beta(\emptyset)$ for some $\beta < \alpha$. By induction hypothesis, there is a derivation tree T' for $u[f_H]v \subseteq \phi_L^\beta(\emptyset)$. We can therefore build a tree T_w for $w \in \mathrm{cl}_H(L)$, by appending a hypothesis rule at the root of T'. Using an extension rule combining all these tres T_w, we finally build the derivation tree T for $[e] \subseteq \mathrm{cl}_H(L)$.

Conversely, assume there is a well-founded derivation tree T for $[e] \subseteq \mathrm{cl}_H(L)$, we want to show that it is indeed true that $[e] \subseteq \mathrm{cl}_H(L)$. Again, this can be shown by induction on the transfinite height α of the tree. If the tree is an axiom, then e is a word of L so it is true that $[e] \in \mathrm{cl}_H(L)$. Otherwise, consider the rule applied

to the root of the tree. If it is an extension rule, then by induction hypothesis, for all $u \in [e]$ we have $u \in \mathrm{cl}_H(L)$, so we have $[e] \subseteq \mathrm{cl}_H(L)$. If it is a hypothesis rule, then there is a hypothesis $e_H \leq f_H \in H$ and words $u, v \in \Sigma^*$ such that e is a word $w \in u[e_H]v$, and there is a tree T' for $u[f_H]v \subseteq \mathrm{cl}_H(L)$. Then by induction hypothesis we have indeed $u[f_H]v \subseteq \mathrm{cl}_H(L)$, and by definition of ϕ_L we have $w \in \mathrm{cl}_H(L)$. □

A.1 Proof of closure properties

We prove Lem. 1:

Lemma 1. *Let $A, B, U, V \subseteq \Sigma^*$. We have*

1. $A \subseteq \mathrm{cl}_H(A)$
2. $\mathrm{cl}_H(\mathrm{cl}_H(A)) = \mathrm{cl}_H(A)$
3. $A \subseteq B$ *implies* $\mathrm{cl}_H(A) \subseteq \mathrm{cl}_H(B)$
4. $H \subseteq H'$ *implies* $\mathrm{cl}_H(A) \subseteq \mathrm{cl}_{H'}(A)$
5. $\mathrm{cl}_H(A) \subseteq \mathrm{cl}_H(B)$ *if and only if* $A \subseteq \mathrm{cl}_H(B)$.
6. $A \subseteq \mathrm{cl}_H(B)$ *implies* $UAV \subseteq \mathrm{cl}_H(UBV)$.

Proof. The first four items follow from the definition of cl_H as the smallest fixed point of ϕ_A (or ϕ_B). If $A \subseteq \mathrm{cl}_H(B)$, we have $\mathrm{cl}_H(A) \subseteq \mathrm{cl}_H(\mathrm{cl}_H(B)) = \mathrm{cl}_H(B)$.

Finally, assume $A \subseteq \mathrm{cl}_H(B)$ and let $(u, w, v) \in U \times A \times V$. We need to show that $uwv \in \mathrm{cl}_H(UBV)$. Consider a derivation tree T for $w \in \mathrm{cl}_H(B)$. Applying the mapping $x \mapsto uxv$ to all nodes of T yields a derivation tree for $uwv \in \mathrm{cl}_H(uBv) \subseteq \mathrm{cl}_H(UBV)$. By Prop. 1, we obtain $uwv \in \mathrm{cl}_H(uBv) \subseteq \mathrm{cl}_H(UBV)$. □

We prove Lem. 2:

Lemma 2. *Let $A, B \subseteq \Sigma^*$, then*

1. $\mathrm{cl}_H(A + B) = \mathrm{cl}_H(\mathrm{cl}_H(A) + \mathrm{cl}_H(B))$,
2. $\mathrm{cl}_H(AB) = \mathrm{cl}_H(\mathrm{cl}_H(A)\mathrm{cl}_H(B))$,
3. $\mathrm{cl}_H(A^*) = \mathrm{cl}_H(\mathrm{cl}_H(A)^*)$

Proof. Using Lem. 1, to prove the first item it suffices to prove that $\mathrm{cl}_H(A) + \mathrm{cl}_H(B) \subseteq \mathrm{cl}_H(A + B)$. This follows from $\mathrm{cl}_H(A) \subseteq \mathrm{cl}_H(A + B)$ and $\mathrm{cl}_H(B) \subseteq \mathrm{cl}_H(A + B)$, by monotonicity of cl_H.

To show the second item, again it suffices to show $\mathrm{cl}_H(A)\mathrm{cl}_H(B) \subseteq \mathrm{cl}_H(AB)$. By stability under concatenation (last item of Lem. 1), for any $X \subseteq \Sigma^*$, we have $X\mathrm{cl}_H(B) \subseteq \mathrm{cl}_H(XB)$, so $\mathrm{cl}_H(A)\mathrm{cl}_H(B) \subseteq \mathrm{cl}_H(\mathrm{cl}_H(A)B)$. Using this stability again, we can now show $\mathrm{cl}_H(A)B \subseteq \mathrm{cl}_H(AB)$, and thus by Lem. 1, $\mathrm{cl}_H(\mathrm{cl}_H(A)B) \subseteq \mathrm{cl}_H(AB)$, thereby concluding the second item.

We finally show the last item, by proving $\mathrm{cl}_H(A)^* \subseteq \mathrm{cl}_H(A^*)$. Let $u \in \mathrm{cl}_H(A)^*$, there is $i \in \mathbb{N}$ such that $u \in \mathrm{cl}_H(A)^i$. As we just proved, this implies $u \in \mathrm{cl}_H(A^i) \subseteq \mathrm{cl}_H(A^*)$. □

B Proofs of section 4

We prove Prop. 3:

Proposition 3 (Decomposition result)**.** *Let $H = \{1 = S_j \mid j \in J\}$ be a set of $(1 = \sum x)$-hypotheses.*

We set $H_{sum} = \{1 \leq S_j \mid j \in J\}$ and $H_{id} = \{a \leq 1 \mid a \in [S_j], j \in J\}$. For every language $L \subseteq \Sigma^*$, we have that:

$$\mathrm{cl}_H(L) = \mathrm{cl}_{H_{sum}}(\mathrm{cl}_{H_{id}}(L))$$

Proof. We have that $\mathrm{cl}_{H_{sum}}(\mathrm{cl}_{H_{id}}(L)) \subseteq \mathrm{cl}_H(L)$ using the monotonicity of the closure (items 3 and 4, lem. 1). Let us show the other inclusion. Let $u \in \mathrm{cl}_H(L)$, and T be a derivation tree witnessing this membership. Note that T is finite since it is well-founded and finitely branching. We show first that T can be transformed into a derivation tree for $u \in \mathrm{cl}_H(L)$, where the application of the rules from H_{id} are delayed after the rules H_{sum}. In other words, no rule from H_{sum} appears after a rule H_{id}. For that, we define a rewriting system where a redex is a pattern of an application of a rule from H_{id} followed immediately by a rule from H_{sum}, followed by an expansion rule. Thus a redex is a derivation of one of the following forms:

$$\dfrac{\dfrac{\left(\dfrac{\pi_i}{ubvw}\right), b \in [S_j]}{u(S_j)vw}}{\dfrac{uvw}{uvaw}} \qquad \text{or} \qquad \dfrac{\dfrac{\left(\dfrac{\pi_i}{uvbw}\right), b \in [S_j]}{uvS_jw}}{\dfrac{uvw}{uavw}}$$

We define the following rewriting rules, which delays the application of the hypothesis rule from H_{id}.

$$\dfrac{\dfrac{\left(\dfrac{\pi_i}{ubvw}\right), b \in [S_j]}{uS_jvw}}{\dfrac{uvw}{uvaw}} \quad \Rightarrow \quad \dfrac{\left(\dfrac{\pi_i}{\dfrac{ubvw}{ubvaw}}\right), b \in [S_j]}{\dfrac{uS_jvaw}{uvaw}}$$

$$\dfrac{\dfrac{\left(\dfrac{\pi_i}{uvbw}\right), b \in [S_j]}{uvS_jw}}{\dfrac{uvw}{uavw}} \quad \Rightarrow \quad \dfrac{\left(\dfrac{\pi_i}{\dfrac{uvbw}{uavbw}}\right), b \in [S_j]}{\dfrac{uavS_jw}{uavw}}$$

Using these rewriting rules, we can transform T into a redex-free derivation T'.

Let T'' be the subtree of T' (with the same root) such that:

- No hypothesis rule from H_{id} is applied in T''.
- For every leaf l of T'', the subtree of T' rooted in l does not contain a hypothesis rule from H_{sum}. We denote this subtree T_l.

This decomposition of T' is possible because all the rule applications of H_{id} are delayed after those of H_{sum}. Note that T_l is a derivation tree for $l \in \mathrm{cl}_{H_{id}}(L)$. Thus T'' is a derivation tree for $u \in cl_{H_{sum}}(\mathrm{cl}_{H_{id}}(L))$. □

Proof of Lem. 4

Proof. If $v \in (\Sigma_\sharp)^*$, let $|v|_\sharp$ be the number of letters in v from $\Theta_\sharp$. We show by induction on $|v|_\sharp$ that for all $v \in (\Sigma_\sharp)^*$, $v \in L_\sharp \Leftrightarrow v \in \mathrm{cl}_{H_\sharp}(L)$. If $|v|_\sharp = 0$, then $v \in L_\sharp \Leftrightarrow v \in L \Rightarrow v \in \mathrm{cl}_{H_\sharp}(L)$. We also show that $v \in \mathrm{cl}_{H_\sharp}(L) \Rightarrow v \in L$, completing the base case of the induction. Consider a derivation tree T_v for $v \in \mathrm{cl}_{H_\sharp}(L)$. Since v does not contain any occurence of $\sharp_j$ for any $j \in J$, no hypothesis from $H_\sharp$ can be applied at the root of T_v, so T_v is necessarily a single leaf, and $v \in L$.

We now proceed to the induction case, and consider $v \in (\Sigma_\sharp)^*$ with $|v|_\sharp > 0$.

Asssume $v \in L_\sharp$. There is $u \in \Sigma^*$, $j \in J$ and $v' \in (\Sigma_\sharp)^*$ such that $v = u\sharp_j v'$ and $|v'|_\sharp = |v|_\sharp - 1$. For all $b \in [S_j]$, let $w_b = ubv'$. Then $exp(v) = \bigcup_{b\in[S_j]} exp(w_b) \subseteq L$. This means that for all $b \in [S_j]$, $w_b \in L_\sharp$ and by induction hypothesis, $w_b \in \mathrm{cl}_{H_\sharp}(L)$. Applying the rule $\sharp_j \leq \sum_{b\in[S_j]} b$ from $H_\sharp$ yields $v \in \mathrm{cl}_{H_\sharp}(L)$.

Conversely, assume $v \in \mathrm{cl}_{H_\sharp}(L)$, witnessed by a derivation tree T_v.

- If the rule applied at the root of T_v is of the form $\sharp_j \leq 1$, then it partitions v into $v_1\sharp_j v_2$, and by induction hypothesis we have $exp(v_1v_2) \subseteq L$. Since L is Γ-closed, and letters from $\Gamma \subseteq \Sigma$ are preserved by exp, for each $i \in I_j$, we have $exp(v_1y_{i,j}v_2) \subseteq K$. Since $exp(v) = \bigcup_{b\in[S_j]} exp(v_1bv_2)$, we obtain $exp(v) \subseteq L$ i.e. $v \in L_\sharp$.
- Or the rule applied at the root of T_v partitions v into $v_1\sharp_j v_2$ for some $j \in J$ and has premises $\{v_1bv_2 \mid b \in [S_j]\}$. By induction hypothesis, for all $b \in [S_j]$, $v_1bv_2 \in L_\sharp$, i.e. $exp(v_1bv_2) \subseteq L$. As before, this yields $v \in L_\sharp$.

□

Proof of Lem. 6:

Lemma 6. *$L_\sharp$ is a regular language, computable effectively.*

Proof. Let $\mathcal{A} = (\Sigma, Q, q_0, F, \delta)$ be a DFA for L. Let $\mathcal{A}_\wedge = (\Sigma, \mathcal{P}(Q), q_0, \mathcal{P}(F), \delta_\wedge)$ be the DFA where $\delta_\wedge$ is defined as follows:

$$\delta_\wedge(P,a) = Q \quad \Leftrightarrow \quad \begin{array}{l} \forall p \in P, \exists q \in Q \text{ such that } \delta(p,a) = q, \text{and} \\ \forall q \in Q, \exists p \in P \text{ such that } \delta(p,a) = q. \end{array}$$

If q is the state of an automaton $\mathcal{A}$, we denote by $[q]_\mathcal{A}$ the language of this automaton with initial state q. Note that if P is a state of $\mathcal{A}_\wedge$, then $[P]_{\mathcal{A}_\wedge} = \cap_{q\in P}[q]_\mathcal{A}$.

Let us construct $\mathcal{A}_\sharp$, the automaton for $L_\sharp$. We set $\mathcal{A}_\sharp = (\Sigma_\sharp, \mathcal{P}(Q), q_0, \mathcal{P}(F), \delta_\sharp)$ be the DFA where $\delta_\sharp$ is defined as follows:

$$\begin{array}{llll} \forall a \in \Sigma & \delta_\sharp(P,a) = Q & \Leftrightarrow & \delta_\wedge(P,a) = Q \\ \forall j \in J, & \delta_\sharp(\sharp_j, a) = Q & \Leftrightarrow & Q = \cup_{b\in[S_j]} Q_b \text{ where } \delta_\wedge(P,b) = Q_b, \forall b \in [S_j] \end{array}$$

We will show that for all $u \in \Sigma_\sharp^*$, for all $P \in \mathcal{P}(Q)$, $u \in [P]_{A_\sharp} \Leftrightarrow exp(u) \subseteq [P]_{\mathcal{A}_\wedge}$, by induction on $|u|$. This implies in particular $[A_\sharp] = L_\sharp$.

- If $u = \epsilon$, then $\epsilon \in [P]_{A_\sharp} \Leftrightarrow P \in \mathcal{P}(F) \Leftrightarrow \epsilon \in [P]_{A_\wedge} \Leftrightarrow exp(\epsilon) \subseteq [P]_{A_\wedge}$.
- If $u = av$ with $a \in \Sigma$, let $Q = \delta_\wedge(P, a)$. By induction hypothesis, $v \in [Q]_{A_\sharp} \Leftrightarrow exp(v) \subseteq [Q]_{A_\wedge}$. Since $exp(u) = a exp(v)$ and $\delta_\sharp(P, a) = Q$, we obtain $u \in [P]_{A_\sharp} \Leftrightarrow v \in [Q]_{A_\sharp} \Leftrightarrow exp(v) \subseteq [Q]_{A_\wedge} \Leftrightarrow a exp(v) \subseteq a[Q]_{A_\sharp} \Leftrightarrow a exp(v) \subseteq [P]_{A_\sharp} \Leftrightarrow exp(u) \subseteq [P]_{A_\sharp}$.
- If $u = \sharp_j v$ with $j \in J$, let $Q_b = \delta(P, b)$ for all $b \in [S_j]$. Then $u \in [P]_{A_\sharp} \Leftrightarrow \forall b \in S_j, v \in [Q_b]_{A_\sharp} \Leftrightarrow \forall b \in [S_j], exp(v) \subseteq [Q_b]_{A_\wedge}$ by induction hypothesis. Since $exp(u) = \bigcup_{b \in [S_j]} b exp(v)$, we obtain $u \in [P]_{A_\sharp} \Leftrightarrow exp(u) \subseteq [P]_{A_\wedge}$.

□

C Simulating Turing Machines

C.1 From word hypotheses to letter hypotheses

We prove Thm. 5

Theorem 5. *Let Σ be an alphabet, H a set of word hypotheses over Σ. There exists an extended alphabet $\Sigma' \supseteq \Sigma$, a set of letter hypotheses H' over Σ' and a regular expression $h \in \mathsf{Exp}_{\Sigma'}$ such that the following holds for every $f \in \mathsf{Exp}_\Sigma$ and $w \in \Sigma^*$.*

$$w \in \mathrm{cl}_H([f]) \qquad \textit{if and only if} \qquad w \in \mathrm{cl}_{H'}([f + h])$$

Furthermore, we guarantee the following:

- *(Σ', H', h) can be computed in polynomial time from (Σ, H).*
- *H' is length-preserving whenever H is.*

Proof. Let Σ be an alphabet and $H \in \mathcal{P}_{\mathrm{fin}}(\Sigma^+ \times \mathcal{P}_{\mathrm{fin}}(\Sigma^*))$ be a set of word hypotheses.

We will show that it is possible to design h and a set of letter hypotheses H' such that performing derivations for $\mathrm{cl}_{H'}([f + h])$ simulates derivations for $\mathrm{cl}_H([f])$. This is done by simulating hypotheses (w, X) from H by many hypotheses in H', one for each letter of w, and the expression h controlling that we process the letters in the right order. This also needs extra alphabets to store information about which hypothesis (w, X) and which position in w we are currently processing.

Call Θ the set $\{(w, X, w', i) \mid (w, X) \in H,\ w' \in X,\ i < |w|\}$ and $\perp\!\!\!\perp = \{\perp_s\} \cup \{\perp_{(w,X)} \mid (w, X) \in H\}$ a set of fresh letters. We let $\Sigma' = \Sigma \sqcup \Theta \sqcup \perp\!\!\!\perp$. Let us call the letters $a \in \Theta$ *transition letters*. For $w \in \Sigma^+$, $(w, X) \in H$ and $w' \in X$, the (w, X, w')-*transition word* $\delta_{w,X,w'} \subseteq \Theta^+$ is the word $(w, X, w', 0) \dots (w, X, w', |w| - 1)$. Call $T \subseteq \Theta^+$ the set of all such transitions

words. We define auxiliary functions source $s : \Theta \to \Sigma$ and target $t : \Theta \to \Sigma^*$ as follows.

$$s : \quad \Theta \quad \to \Sigma \qquad t : \quad \Theta \quad \to \Sigma^*$$
$$(w, X, w', i) \mapsto w_i \qquad (w, X, w', i) \mapsto \begin{cases} w'_i \dots w'_{|w'|-1} & \text{if } i = |w| - 1 \\ \epsilon & \text{if } i \geq |w'| \\ w'_i & \text{otherwise} \end{cases}$$

Notice that if H is length-preserving, then t is actually a function $\Theta \to \Sigma$. Furthermore, if we extend s and t to monoid morphisms, we have $s(\delta_{w,X,w'}) = w$ and $t(\delta_{w,X,w'}) = w'$.

We also define the function $\mathit{Hypo} : \Theta \cup (\perp\!\!\!\perp \setminus \perp_s) \to H$ by $\mathit{Hypo}(w, X, w', i) = (w, X)$ and $\mathit{Hypo}(\perp_{\mathit{Hyp}}) = \mathit{Hyp}$.

If $\mathit{Hyp} \in H$, let $\Theta_{\mathit{Hyp}} = \{(w, X, w', i) \in \Theta \mid (w, X) = \mathit{Hyp}\}$

The set H' is defined as follows.

$$\begin{array}{l} H' = H'_1 \cup H'_2 \cup H'_3 \qquad \text{where} \\ H'_1 = \{(\perp, \{\perp\}) \mid \perp \in \perp\!\!\!\perp\} \\ H'_2 = \{(a, \{b \in \Theta_{\mathit{Hyp}} \mid s(b) = a\} \cup \{\perp_s\}) \mid a \in \Sigma, \mathit{Hyp} \in H\} \\ H'_3 = \{(b, \{t(b), \perp_{\mathit{Hypo}(b)}\}) \mid b \in \Theta\} \end{array}$$

H' is length-preserving as soon as H is.

Let $h_{\overline{0}} = \{(w, X, w', k) \in \Theta \mid k \neq 0\}$ and $h_{succ} = \{(w, X, w', k)(w, X, w', k + 1) \in \Theta^2\}$. For each $\mathit{Hyp} \in H$, we define $h_{\mathit{Hyp},\overline{succ}} = (\Theta_{\mathit{Hyp}})^*(\Theta^2_{\mathit{Hyp}} \setminus h_{succ})$, which can be written as a regular expression of polynomial size over alphabet Θ_{Hyp}. Finally, h is a sum $h_1 + h_2$ where

$$h_1 = \sum_{w \in T} \Big(\sum_{\substack{u,v \in \Theta^* \\ a \in \Theta \\ w = uav}} \Sigma^* u \perp_s s(v) \Sigma^* + \Sigma^* t(u) \perp_{\mathit{Hypo}(a)} v \Sigma^* \Big)$$

$$h_2 = \Sigma^* (h_{\overline{0}} + \sum_{\mathit{Hyp} \in H} h_{\mathit{Hyp},\overline{succ}}) \Sigma^*$$

It is straightforward to verify that (Σ', H', h) is computable in polynomial time from (Σ, H). We need to show the announced equivalence, for arbitrary $u \in \Sigma^*$ and $f \in \mathsf{Exp}_\Sigma$:

$$u \in \mathrm{cl}_H([f]) \qquad \text{if and only if} \qquad u \in \mathrm{cl}_{H'}([f + h])$$

We start with the left to right implication. To do so, we first show an auxiliary lemma.

Lemma 13. *For every $(w, X) \in H$, we have $w \in \mathrm{cl}_{H'}([X + h])$.*

Proof. We first show that, for every transition word $\delta_{w,X,w'}$ and factorization $\delta_{w,X,w'} = uv$, we have $t(u)v \in \mathrm{cl}_{H'}([X + h])$ by induction over the length of v. If v is the empty word, we have $t(u)v = t(\delta_{w,X,w'}) = w' \in X \subseteq \mathrm{cl}_{H'}([X + h])$. Otherwise, $v = av'$ with $a = (w, X, w', i)$ for some $i \in \mathbb{N}$. By the inductive hypothesis,

$t(ua)v' \in \mathrm{cl}_{H'}([X+h])$. It is thus sufficient to prove $t(u)av' \in \mathrm{cl}_{H'}([t(ua)v'+h])$. By definition of H', we have $(a, \{t(a), \bot_{(w,X)}\}) \in H'$. Clearly, $t(u)\bot_{(w,X)}v' \in [h]$ and $t(u)t(a)v' = t(ua)v'$ so we may conclude.

Now, we show that for every factorization $\delta_{w,X,w'} = uv$ we have $us(v) \in \mathrm{cl}_{H'}([X+h])$ by another induction over $|v|$. If v is empty, then this is given by the previous induction by taking a trivial factorization. Otherwise, $v = av'$ with $a = (w, X, w', i)$ for some $i \in \mathbb{N}$. Let $Hyp = (w, X)$. We have two subcases.

- If $i > 0$ and u is non-empty, then we show $uw_i s(v') \in \mathrm{cl}_{H'}([uas(v')+h])$, from which me may conclude using the inductive hypothesis and the properties of the closure. By definition of H' and setting $Y = \{b \in \Theta_{Hyp} \mid s(b) = w_i\} \cup \{\bot_s\}$, we have $(w_i, Y) \in H'$. It thus suffices to check that $uys(v') \in [h]$ for all $y \in Y \setminus \{a\}$ to conclude.
  - Either $y = \bot_s$, in which case $uys(v') \in [h_1]$.
  - Otherwise, since u is non-empty and $y \neq a$ we necessarily have $uy \in h_{Hyp,\overline{succ}}$ and thus $uys(v') \in [h_2]$.
- Otherwise $v = \delta_{w,X,w'}$ and u is empty. In such a case, note that for every $w'' \in X$, we have $s(\delta_{w,X,w'}) = w = s(\delta_{w,X,w''})$. For any $w'' \in X$, call $b^0_{w''} = (w, X, w'', 0)$ the first letter of $\delta_{w,X,w''}$ and factorize w as av' with $a \in \Sigma$ and $v' \in \Sigma^*$. By definition of H', setting $Y = \{b \in \Theta_{Hyp} \mid s(b) = a\} \cup \{\bot_s\}$, we have $(a, Y) \in H'$. By the inductive hypothesis, we have $\{b^0_{w''}v' \mid w'' \in X\} \subseteq \mathrm{cl}_{H'}([X+h])$. Thus it is sufficient to show $yv' \in \{b^0_{w''}v' \mid w'' \in X\} \cup [h]$ for all $y \in Y$ to conclude. If $y = \bot_s$, then clearly $yv' \in [h_1]$. Otherwise, $y = (w, X, w'', j)$ with $w'' \in X$. If $j = 0$, then $y = b^0_{w''}$ and we are done. Otherwise $y \in h_{\overline{0}}$ and thus $yv' \in [h_2]$.

This concludes our induction, showing that $w = s(\delta_{w,X,w'}) \in \mathrm{cl}_{H'}([X+h])$. □

With Lem. 13, we then proceed by induction over the derivation of $u \in \mathrm{cl}_H([f])$, to show the wanted implication, i.e. $u \in \mathrm{cl}_H([f]) \Rightarrow u \in \mathrm{cl}_{H'}([f+h])$. If $u \in [f]$, then we also have $u \in [f+h]$ and we are done. Otherwise, we have some factorization $u = u'wu''$ for $u', u'' \in \Sigma^*$, $w \in \Sigma^+$, some X such that $(w, X) \in H$ and $u'Xu'' \subseteq \mathrm{cl}_{H'}([f+h])$ by the induction hypothesis. By Lem. 13, we know that $w \in \mathrm{cl}_{H'}([X+h])$, and by stability under concatenation (last item of Lem. 1), we have $u'wu'' \in \mathrm{cl}_{H'}([u'(X+h)u''])$. But since we have $[\Sigma^* h \Sigma^*] = [h]$, this amounts to $u'wu'' \in \mathrm{cl}_{H'}([u'Xu''+h])$. Since we have, $[u'Xu''+h] \subseteq \mathrm{cl}_{H'}([f+h])$ we can conclude $u \in \mathrm{cl}_{H'}([f+h])$.

To prove the converse, i.e. the right to left implication of Thm. 5, we first need auxiliary lemmas concerning $\perp\!\!\!\perp$ letters and consistency of transitions.

Lemma 14. *If $w \in (\Sigma \cup \Theta)^* \perp\!\!\!\perp (\Sigma \cup \Theta)^*$, then $w \in \mathrm{cl}_{H'}([f+h])$ if and only if $w \in [h]$.*

Proof. The right-to-left direction is easy. For the left-to-right, first notice that H' does not allow to remove letters from $\bot$, hence $(\Sigma \cup \Theta)^* \perp\!\!\!\perp (\Sigma \cup \Theta)^* \perp\!\!\!\perp (\Sigma \cup \Theta)^* \cap \mathrm{cl}_{H'}([f+h])$ is empty. Moreover, any rule of H' either leaves the word unchanged, or introduces a new letter from $\bot$. This means if a word $w \in (\Sigma \cup \Theta)^* \perp\!\!\!\perp (\Sigma \cup \Theta)^*$ is

in $w \in \mathrm{cl}_{H'}([f+h])$, by the previous remark it must be in $[f+h]$. Since $[f] \subseteq \Sigma^*$, we get $w \in [h]$. □

Lemma 15. *If $w \in \mathrm{cl}_{H'}([f+h])$ has two letters $a, b \in \Theta \cup (\perp\!\!\!\perp \setminus \perp_s)$, then $Hypo(a) = Hypo(b)$.*

Proof. Straightforward induction. □

The following lemma states that if we commpletely process a word hypothesis from H letter by letter according to $\mathrm{cl}_{H'}([f+h])$, we indeed performed a step according to $\mathrm{cl}_H([f])$.

Lemma 16. *Let $\delta_{w,X,w'}$ be a transition word, and $u, v \in \Theta^*$ such that $uv = \delta_{w,X,w'}$. Then we have, for every $u', v' \in \Sigma^*$*

$$\left.\begin{array}{c} u, v \neq \epsilon \text{ and } u'us(v)v' \in \mathrm{cl}_{H'}([f+h])) \\ \text{or} \\ u't(u)vv' \in \mathrm{cl}_{H'}([f+h]) \end{array}\right\} \Rightarrow u't(uv)v' \in \mathrm{cl}_H([f]).$$

Proof. We will show the result for all (w, X, w'), and proceed by induction. Towards notational convenience, let us define the following sets and function, for any $(w, X) \in H$.

$$\begin{array}{ll} Y_{w,X,1} & = \{u'us(v)v' \mid u', v' \in \Sigma^*, \exists \beta \in X, uv = \delta_{w,X,w'}, u, v \neq \epsilon\} \\ Y_{w,X,2} & = \{u't(u)vv' \mid u', v' \in \Sigma^*, \exists \beta \in X, uv = \delta_{w,X,w'}\} \\ Y & = \bigcup_{(w,X)\in H} Y_{w,X,1} \cup Y_{w,X,2} \end{array}$$

$$\begin{array}{rl} F : Y & \to \Sigma^* \\ u'us(v)v' & \mapsto u't(uv)v' \\ u't(u)vv' & \mapsto u't(uv)v' \end{array}$$

Notice that F is well-defined because u is non-empty in the first case, and therefore contains the information $w = uv$. In the second case, it suffices to project letters from v according to t. The function F describes the result after we finish processing the current word hypothesis from H letter-by-letter.

We show that for every $\alpha \in \mathrm{cl}_{H'}([f+h])$, if $\alpha \in Y$, then $F(\alpha) \in \mathrm{cl}_H([f])$. We proceed by induction on the derivation tree T_α for $\alpha \in \mathrm{cl}_{H'}([f+h])$. Note that $Y_1 \cap [f+h] = \emptyset$, $Y_2 \cap [h] = \emptyset$, and $Y_2 \cap [f] = [f]$. Furthermore, for all $\alpha \in [f]$ we have $F(\alpha) = \alpha$. Putting the pieces together, if $\alpha \in [f+h]$ and $\alpha \in Y$, we necessarily have $F(\alpha) \in [f] \subseteq \mathrm{cl}_H([f])$, which takes care of the base step where T_α is a single leaf. For the inductive step, suppose that the root of T_α uses the rule $(a, X') \in H'$ in the following way, with $\alpha = xay$:

$$\frac{\forall a' \in X', xa'y}{xay}$$

where by induction hypothesis, for each $a' \in X'$, if $xa'y \in Y$, then $F(xa'y) \in \mathrm{cl}_H([f])$. We perform a case analysis according to which component of Y the word xay belongs to.

- If $xay \in Y_{w,X,1}$ for some $(w, X) \in H$, we have $xay = u'us(v)v'$ with $u, v \neq \epsilon$. According to hypotheses H', Lem. 14 and the shape of h_1, we necessarily have: $a \in \Sigma$, $x = u'u$, $s(v) = as(v'')$, and $y = s(v'')v'$ for some $v'' \in \Theta^*$. We also have $x \bot_s y \in h_1$. By Lem. 15 and since $u \neq \epsilon$, we must have $a' \in X'$ such that $\delta_{w,X,w'} = ua'v''$. Hence we have $xa'y = u'ua's(v'')v' \in Y$ and thus, by the induction hypothesis, $F(xa'y) \in \mathrm{cl}_H([f])$. We can then conclude since $F(xay) = F(xa'y)$.
- If $xay \in Y_{w,X,2}$ for some $(w, X) \in H$, we have $xay = u't(u)vv'$. We make a case distinction on whether $v = \epsilon$.
  - If $v \neq \epsilon$, we necessarily have $a \in \Theta$, $x = u't(u)$, $v = av''$, and $y = v''v'$ for some $v'' \in \Theta^*$, with $x \bot_{Hypo(a)} y \in h_1$. But then $xt(a)y \in Y_{w,X,2}$ and, by the inductive hypothesis $F(xt(a)y) \in \mathrm{cl}_H([f])$. We can conclude by $F(xt(a)y) = F(xay)$.
  - Otherwise, we actually have $xay \in \Sigma^*$. This corresponds to the case where we finished treating a word hypothesis from H, and start treating a new one. By using Lem. 14, we can show that there exists a factorization $y = \alpha y'$ and an hypothesis $Hyp = (a\alpha, A) \in H$ such that $X' = \{b \in \Theta_{Hyp} \mid s(b) = a\} \cap \bot_s$. For each $\beta \in A$, let $\theta_\beta = (a\alpha, A, \beta, 0) \in \Theta$. We have $x\theta_\beta\alpha y' \in Y_{a\alpha,A,1}$ for every $\beta \in A$, so, by the inductive hypothesis $F(x\theta_\beta\alpha y') \in \mathrm{cl}_H([f])$. Moreover, for each $\beta \in A$, notice that $F(x\theta_\beta\alpha y') = x\beta y'$. Hence, by the definition of the H-closure, we have $xa\alpha y \in \mathrm{cl}_H([f])$.

□

This achieves the proof of Thm. 5, since taking $u = v = \epsilon$ in Lem. 16 yields the right-to-left implication in Thm. 5. □

C.2 Simulating an LBA with closure

We prove Lem. 8.

Lemma 8. *For every alternating LBA $\mathcal{A}$, there exists a set of length-preserving word hypotheses $H_\mathcal{A}$ over the alphabet $\Theta = Q \cup \Gamma_\#$ such that the following holds. For any subset of configurations $D \subseteq C$*

$$\mathrm{cl}_{H_\mathcal{A}}(D) = \mathrm{Attr}^{\exists\mathrm{loise}}(D)$$

Furthermore, this reduction is polytime computable.

Proof. As transitions over configurations are made at a local level, regarding $H_{\mathcal{A}}$ as a subset of $\mathcal{P}_{\text{fin}}(\Theta^3 \times \mathcal{P}_{\text{fin}}(\Theta^3))$, define

$$H_{mid} = \bigcup_{\substack{b\in\Gamma_{\#}\\ q\in Q\\ a\in\Gamma}} \{(bqa, \bigcup_{X\in\Delta(q,a)} \{bcr \mid (R,c,r)\in X\} \cup \{rbc \mid (L,c,r)\in X\})\}$$

$$H_{right} = \bigcup_{b\in\Gamma_{\#}, q\in Q} \{(bq\#_R, \bigcup_{X\in\Delta(q,\#_R)} \{rb\#_R \mid (L,c,r)\in X\}\}$$

$$H_{left} = \bigcup_{q\in Q} \{(q\#_L u, \bigcup_{X\in\Delta(q,\#_L)} \{\#_L ru \mid (R,c,r)\in X\})\}$$

$$H_{\mathcal{A}} = H_{mid} \cup H_{right} \cup H_{left}$$

It is then easy to check that $\text{cl}_{H_{\mathcal{A}}}$, when restricted to C and $\text{Attr}^{\exists}(D)$ are fixed points of the same operator. Notice that if $\mathcal{A}$ is an LBA, $H_{\mathcal{A}}$ is length-preserving.

For Turing machines, the hypotheses are no more left preserving, and as shown in [13] rules H_{right}, H_{left} are replaced with rules $H_B = \{\#_L \leq \#_L B, \#_R \leq B\#_R\}$.

□

C.3 Proof of Lem. 11

Lemma 11. *Suppose that $\mathcal{M} = (Q, Q_F, \Gamma, \iota, B, \Delta)$ is a total Turing machine with final states $\{0,1\}$ and initial state ι. Let $w \in \Sigma^*$ be an input word for $\mathcal{M}$.*

Then there is effectively a set of length-preserving word hypotheses H and expressions e_w, h such that $[\mathcal{M}](w) = 1$ if and only if $\mathsf{KA}_H \vdash e_w \leq h$

Proof. Consider the linearly bounded automaton $\text{LBA}(\mathcal{M})$ associated with $\mathcal{M}$ and take $H = H_{\text{LBA}}(\mathcal{M})$ to be the set of length-preserving hypotheses given by Lem. 8. Notice that this LBA is stuck on configurations where the head reaches an extremity of the configurations. Take accordingly h to be the sum $h_1 + h_2 + h_3$ where

$$h_1 = Q\#_L\Gamma^*\#_R \qquad h_2 = \#_L\Gamma^*Q\#_R \qquad \text{and} \qquad h_3 = \#_L\Gamma^*\{1\}\Gamma^+\#_R$$

By the semantics, the right-to-left implication is trivial in light of Lem. 8. Indeed, as soon as the number of B symbols in the lefthand side is sufficient, the expression h forces the result of $\mathcal{M}$ to be 1. For the left-to-right, suppose that $[\mathcal{M}](w) = 1$. Then, from ιw, $\mathcal{M}$ may execute to a final configuration $u1v$. Considering the execution of $\text{LBA}(\mathcal{M})$ over $\#_L B^k \iota w B^{k'} \#_R$, we may show that we have $n, n' \in \mathbb{N}$ such that a stuck configuration c occurs in one of the following patterns:

- if $k \geq n$ and $k' \geq n'$, then the computation faithfully simulates the execution of $\mathcal{M}$ and $c \in [h_3]$

- if $k < n$ and $k' \geq n'$, then the computation cannot faithfully the execution of $\mathcal{M}$ because of lack of space on the left of the tape and $c \in [h_1]$
- if $k \geq n$ and $k' < n'$, $c \in [h_2]$ for similar reasons
- if $k < n$ and $k' < n'$, we have $c \in [h_2 + h_1]$

Let $e_w = \#_L B^* \iota w B^* \#_R$. We partition e_w into the following $(n+1)(n'+1)$ regular expressions e for which we can prove $\mathsf{KA}_H \vdash e \leq h$. We detail below the different cases.

- The expressions $\#_L B^k \iota w B^{k'} \#_R$ with $k < n$ and $k' < n'$. The wanted inequality can be shown in KA_H by Cor. 9.
- The expressions $\#_L B^k \iota w v' B^{n'} B^* \#_R$ with $k < n$. Using the proof of Lem. 8, we can show that $\#_L B^k \iota w B^{n'} \in \mathrm{cl}_H(Q \#_L \Gamma^*)$, which by Prop. 2 establishes that $\mathsf{KA}_H \vdash \#_L B^k \iota w v' B^{n'} \leq Q \#_L \Gamma^*$. Then, we have $\mathsf{KA} \vdash B^* \#_R \leq \Gamma^* \#_R$, thus by concatenation and $\mathsf{KA} \vdash \Gamma^* \Gamma^* \leq \Gamma^*$, we have $\mathsf{KA}_H \vdash \#_L B^k \iota w B^{n'} B^* \#_R \leq Q \#_L \Gamma^* \#_R = h_1$.
- The expressions $\#_L B^* B^n \iota w B^{k'} \#_R$ with $k' < n'$ are treated in the same way.
- The expression $\#_L B^* B^n \iota w B^{n'} B^* \#_R$ also gets a fairly similar treatment: $\mathsf{KA}_H \vdash B^n \iota w B^{n'} \leq \Gamma^* 1$, from which we conclude by cutting with a proof in KA.

□

C.4 Proof of Lem. 10

Lemma 17. *There is no universal total Turing machine.*

Proof. Suppose that $\mathcal{M}$ is a universal total Turing machine. Consider the diagonal function $\mathcal{D}(w) = 1 - \mathcal{M}(\langle w, w \rangle)$. Notice that $\mathcal{D}$ is total. So, by universality, we have a contradiction.

$$[\mathcal{D}](\lceil \mathcal{D} \rceil) = 1 - [\mathcal{M}](\lceil \mathcal{D} \rceil, \lceil \mathcal{D} \rceil) = 1 - [\mathcal{D}](\lceil \mathcal{D} \rceil)$$

□