Ground Reducibility is EXPTIME-complete

Hubert Comon, Florent Jacquemard

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Ground Reducibility is EXPTIME-complete

Hubert Comon , Florent Jacquemard

N°3800

Novembre 1999

THÈME 2

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--3800--FR+ENG

Ground Reducibility is EXPTIME-complete

Hubert Comon[*] , Florent Jacquemard[†]

Thème 2 — Génie logiciel
et calcul symbolique
Projet PROTHEO

Rapport de recherche n° 3800 — Novembre 1999 — 26 pages

Abstract: We prove that ground reducibility is EXPTIME-complete in the general case. EXPTIME-hardness is proved by encoding the emptiness problem for the intersection of recognizable tree languages. It is more difficult to show that ground reducibility belongs to DEXPTIME. We associate first a tree automaton with disequality constraints to a rewrite system and a term. This automaton is deterministic and accepts a non-empty tree language iff the given term is not ground reducible by the system. The number of states of the automaton is exponential in the size of the term and the system, and the size of its constraints is polynomial in the size of the term and the system. Then we prove some new pumping lemmas, using a total ordering on the computations of the automaton. Thanks to these lemmas, we can show that emptiness for a tree automaton with disequality constraints can be decided in a time which is polynomial in the number of states and exponential in the size of the constraints. Altogether, we get a simply exponential time deterministic algorithm for ground reducibility decision.

Key-words: Ground Reducibility, Term Rewriting, Tree Automata, Complexity, Inductive Theorem Proving

(Résumé : tsvp)

[*] CNRS and LSV, 61 Av. du Pdt Wilson 94235 Cachan cedex, France. email: Hubert.Comon@lsv.ens-cachan.fr, http://www.lsv.ens-cachan.fr/~comon

[†] LORIA and INRIA, 615 rue du Jardin Botanique, B.P. 101, 54602 Villers-les-Nancy Cedex, France. email:Florent.Jacquemard@loria.fr http://www.loria.fr/~jacquema

Unité de recherche INRIA Lorraine
Technopôle de Nancy-Brabois, Campus scientifique,
615 rue de Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY (France)
Téléphone : 03 83 59 30 30 - International : +33 3 3 83 59 30 30
Télécopie : 03 83 27 83 19 - International : +33 3 83 27 83 19
Antenne de Metz, technopôle de Metz 2000, 4 rue Marconi, 55070 METZ

La réductibilité inductive est EXPTIME-complète

Résumé : Nous prouvons que la réductibilité inductive est une propriété EXPTIME-complète dans le cas général. La EXPTIME-difficulté est montrée par réduction du problème du vide de l'intersection de langages d'arbres reconnaissables. La preuve que la réductibilité inductive appartient à la classe DEXPTIME est plus difficile. Nous associons un automate d'arbres à contraintes diségalitaires à un système de réécriture et à un terme. Cet automate est déterministe et reconnaît un langage non vide si et seulement si le terme donné est inductivement réductible par le système. Le nombre d'états de l'automate est exponentiel en la taille du système et du terme et la taille de ses contrainte est polynomiale en cette mesure. Nous démontrons alors certains lemmes de pompage, basés sur un ordre total total sur les calculs d'automate. À l'aide de ces lemmes, nous pouvons montrer que le problème de la décision du vide pour automates d'arbres à contraintes diségalitaires est décidable en temps polynomial en le nombre d'états de l'automate et exponentiel en la taille de ses contraintes. Mis ensemble, ces résultats donnent un algorithme en temps simplement exponentiel déterministe pour la décision de la réductibilité inductive.

Mots-clé : Réductibilité indutive, Réécriture de termes, Automates d'arbres, Complexité, Démonstration automatique par récurrence

1 Introduction

Ground reducibility (GR for short) of a term t w.r.t. a term rewriting system $\mathcal{R}$ expresses that all ground instances (instances without variables) of t are reducible by $\mathcal{R}$. This property is fundamental in automating inductive proofs in equational theories without constructors [9]. It is also related to *sufficient completeness* in algebraic specifications (see *e.g.* [11]). Roughly, it expresses that all cases have been covered by $\mathcal{R}$ and that t will be reducible for any inputs. Many papers have been devoted to decision of ground reducibility. Let us report a brief history of the milestones, starting only in 1985 with the general case.

Ground reducibility was first shown decidable by D. Plaisted [13]. The algorithm is however quite complex: a tower of 9 exponentials though there is no explicit complexity analysis in the paper. D. Kapur *et al* gave another decidability proof [11] which is conceptually simpler, though still very complicated, and whose complexity is a tower of 7 exponentials in the size of $\mathcal{R}$, t. More precisely, they show that checking the reducibility of all ground instances of t can be reduced to checking the reducibility of all ground instances of t of depth smaller than $N(\mathcal{R})$ where $N(\mathcal{R})$ is a tower of 5 exponentials in the size of $\mathcal{R}$. A third proof was proposed by E. Kounalis in [12]. The result is generalized to co-ground reducibility and the expected complexity is 5 exponentials, though there is no explicit complexity analysis in the paper. These three algorithms use combinatorial arguments and some "pumping property": if there is a deep enough irreducible instance of t, then there is also a smaller instance which is also irreducible. This yielded the idea of making explicit the pumping argument as a pumping lemma in some tree language. In support of this idea, when both t and the left members of $\mathcal{R}$ are *linear*, *i.e.* each variable appears only once, then the set of reducible instances of t is accepted by a finite tree automaton [8]. Hence the set of irreducible ground instances is also accepted by a tree automaton, by complement. This easily gives a simply exponential algorithm in the linear case. (As we will see this algorithm is optimal).

H. Comon expressed first the problem of ground reducibility as an emptiness problem for some tree language [3]. He also gave a decision proof whose complexity is even worse than the former ones. A.-C. Caron, J.-L. Coquidé and M. Dauchet proved a very beautiful result in 1993 [2, 5], enlighting the pumping properties and their difficulty. They actually show a more general result: the first-order theory of unary encompassment predicates is decidable. And it turns out that ground reducibility can be expressed as a simple formula in this logic. Their technique consists in associating an automaton with each formula, in the spirit of Buchi's and Rabin's method. The kind of automata which is appropriate here is what they call *reduction automata*, a particular case of *automata with constraints* introduced by M. Dauchet in 1981. Such tree automata have the ability to check for equality or disequality of some subtrees before applying a transition rule. In general, emptiness of languages recognized by such automata is undecidable. However, when we only allow a fixed number of equality tests on each computation branch, then emptiness becomes decidable. Unfortunately, their result does not give any information about possible efficient algorithms. The complexity which results from their proof is not better than Plaisted's bound. We tried to specialize the tree automata technique for ground reducibility and we got in this way a triple exponential bound [4]. This is better than previous methods, but still far from the lower bound.

The problem in all works about ground reducibility is that they give a bound on the depth of a minimal irreducible instance of t (or a minimal term accepted by the automaton). However, after establishing carefully such an upper bound, they use a brute-force algorithm, checking the reducibility of all terms of depth smaller than the bound, which increases the complexity by a double exponential.

We use here a different approach. We still rely on automata with disequality constraints. However, we do not try to give a bound on the depth of an accepted term. Rather, we show a stronger result: with an appropriate notion of minimality, a minimal term accepted by the automaton contains at most an exponential number of distinct subterms. To prove this, we use a generalization of pumping to arbitrary replacements for which the term is decreasing according to some well chosen well founded ordering. With a few more ingredients, this yields an algorithm for deciding the emptiness of an automaton with disequality constraints which runs in polynomial time w.r.t. the number of states and in exponential time w.r.t. the size of the constraints. On the other hand, we show that ground reducibility of t w.r.t. $\mathcal{R}$ can be reduced to the emptiness problem for an automaton $\mathcal{A}$ with disequality constraints whose number of states is an exponential in the size of $\mathcal{R}$ and t and whose constraints are polynomial in size. Altogether, we have a simply exponential algorithm for ground reducibility.

This result is optimal since ground reducibility is EXPTIME-hard, already for linear rewrite systems and linear t. A $O(2^{\frac{n}{\log n}})$ lower bound was proved by D. Kapur *et al* [10]. We give here a simple proof of EXPTIME-hardness. It is known that the emptiness problem for the intersection of n recognizable languages is EXPTIME-complete, see [7], [14]. We show here that this problem is reducible to ground reducibility in polynomial time.

In section 2, we recall the definition of automata with disequality constraints. In section 3, we show how to construct an automaton with disequality constraints whose emptiness is equivalent to the ground reducibility of t w.r.t. $\mathcal{R}$ and we analyze carefully the complexity of such a construction, and the size of the automaton. Section 4 is devoted to to pumping lemmas for automata with disequality constraints. These lemmas are applied in section 5 to derive an optimal algorithm which checks the emptiness of the (language recognized by) an automaton with disequality constraints. Finally, we study the lower bound of ground reducibility in section 6.

2 Automata with Disequality Constraints

$\mathcal{F}$ will always be a fixed finite set of function symbols (together with their arity), and $\mathcal{X}$ a set of variables. The set of terms built on $\mathcal{F}$ is written $\mathcal{T}(\mathcal{F},\mathcal{X})$ and its subset of ground terms is written $\mathcal{T}(\mathcal{F})$. A *position* is a string of positive integers. The concatenation of two positions p and p' is denoted pp' and Λ is the empty string. The length of a string p is $|p|$. Positions are ordered according the prefix ordering: $p \prec p'$ iff there is a string p'' such that $pp'' = p'$. The position p, p' are called *parallel*, $p \parallel p'$, iff $p \not\preccurlyeq p'$ and $p' \not\preccurlyeq p$.

As usual, a finite term t can be viewed as a mapping from its set of positions $Pos(t)$ into $\mathcal{F}$. For instance, if $t = f(g(a), b)$, $Pos(t) = \{\Lambda, 1, 11, 2\}$ and *e.g.* $t(1) = g$. The subset of maximal position of t w.r.t. $\preccurlyeq$, also called subset of *leaves positions* is denoted $Posl(t)$. If $p \in Pos(t)$, we write $t|_p$ for the subterm of t at position p and $t[s]_p$ for the term obtained by replacing $t|_p$ by s (at position p) in t.

We assume the reader familiar with (constrained) term rewriting systems (see [6] for a survey). Let us only recall that a term t is *ground reducible* by a rewrite system $\mathcal{R}$ iff all the ground instances of t are reducible by $\mathcal{R}$. The rewriting relation associated to a rewrite system $\mathcal{R}$ is denoted $\xrightarrow[\mathcal{R}]{}$ and its reflexive transitive closure is denoted $\xrightarrow[\mathcal{R}]{*}$. A term t is *ground reducible* by a rewrite system $\mathcal{R}$ iff all the ground instances of t are reducible by $\mathcal{R}$.

We use the subsumption quasi-ordering on terms: $s \geqslant t$ if there is a substitution σ such that $s\sigma = t$. Two terms are *similar* if $s \geqslant t$ and $t \geqslant s$. The set of variables occurring in a term t is denoted $Var(t)$. Finally, the *size* of a term t, which is denoted by $\|t\|$, is the cardinal $|Pos(t)|$ of its positions set, and the size of a rewrite system $\mathcal{R}$, which is denoted $\|\mathcal{R}\|$, is the sum of the sizes of its left members[1].

Definition 1 *An* automaton with disequality constraints *(or ADC for short) is a tuple $(Q, Q^{\mathrm{f}}, \Delta)$ where Q is a finite set of states, Q^{f} is the subset of Q of* final states *and Δ is a finite set of transition rules of the form: $f(q_1, \ldots, q_n) \xrightarrow{c} q$ where $f \in \mathcal{F}$ has arity n, $q_1, \ldots, q_n, q \in Q$ and c is a boolean combination without negation of constraints $\pi \neq \pi'$ where π, π' are positions.*

The empty conjunction is written $\top$.
The state q is called *target state* of the rule $f(q_1, \ldots, q_n) \xrightarrow{c} q$.
A ground term $t \in \mathcal{T}(\mathcal{F})$ *satisfies* a constraint $\pi \neq \pi'$ (which we write $t \models \pi \neq \pi'$) if both π and π' are positions of t and $t|_\pi \neq t|_{\pi'}$. This notion of satisfaction is extended to conjunctions and disjunctions as expected. (In particular $t \models \top$ for every t).

Definition 2 *A* run *of the automaton $\mathcal{A} = (Q, Q^{\mathrm{f}}, \Delta)$ on a term t is a mapping ρ from $Pos(t)$ into Δ such that, for every $p \in Pos(t)$, if $t(p) = f$ with arity n then $\rho(p)$ is a rule $f(q_1, \ldots, q_n) \xrightarrow{c} q$ and **1.** for every $1 \leq i \leq n$, $\rho(p \cdot i)$ is a rule whose target is q_i **2.** $t|_p \models c$. If only the first condition is met by ρ, ρ will be called a* weak run.

Runs of $\mathcal{A}$ can also be seen as ground terms over the alphabet Δ (terms of $\mathcal{T}(\Delta)$), the arity of a "symbol" $f(q_1, \ldots, q_n) \xrightarrow{c} q$ in Δ being n, the arity of the symbol f in $\mathcal{F}$.

A term $t \in \mathcal{T}(\mathcal{F})$ is *accepted by* $\mathcal{A}$. there is a run ρ of $\mathcal{A}$ on t such that $\rho(\Lambda)$ is a rule whose target is a final state of Q^{f}. In this case, ρ is called an *accepting run*. The language $L(\mathcal{A})$ of $\mathcal{A}$ is the subset of $\mathcal{T}(\mathcal{F})$ of its accepted terms. Equivalently, $L(\mathcal{A})$ is the set of all terms $t \in \mathcal{T}(\mathcal{F})$ which can be reduced to a final state $q \in Q^{\mathrm{f}}$ by the constrained rewrite system Δ:

$$L(\mathcal{A}) = \{t \in \mathcal{T}(\mathcal{F}) \mid \exists q \in Q^{\mathrm{f}},\ t \xrightarrow[\Delta]{*} q\}$$

A *regular language* is a language of some standard tree automata, *i.e.* of an ADC all the constraints of which are $\top$.

[1] This may be a non standard definition but the size of right hand sides of rules is not relevant for our purpose.

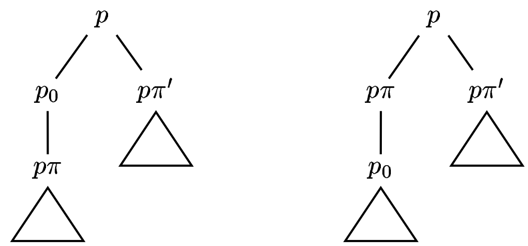

Figure 1: An equality close to p_0 An equality far from p_0.

Example 3 Let $\mathcal{F} = \{f, a, b\}$ and $Q = \{q\} = Q^{\mathrm{f}}$,

$$\Delta = \{r_1 : a \to q \quad r_2 : b \to q \quad r_3 : f(q, q) \xrightarrow{1 \neq 2} q\}.$$

This defines an automaton (which accepts the terms irreducible by the rule $f(x, x) \to a$). The term $f(a, b)$ is accepted since $\rho = r_3(r_1, r_2)$ is a run on t such that r_3 yields a final state. The term $f(a, a)$ is not accepted by $\mathcal{A}$: there is a weak run $r_3(r_1, r_1)$ on $f(a, a)$ but the disequality of r_3 is not satisfied.

Note that in general ADC can be non-deterministic (more than one run on a term) or not completely specified (no run on some term). However, given a run ρ, there is a unique term $[\rho] \in \mathcal{T}(\mathcal{F})$ *associated* to ρ.

Definition 4 *Let $\mathcal{A} = (Q, Q^{\mathrm{f}}, \Delta)$ be an ADC and ρ a weak run of $\mathcal{A}$ on t. An* equality *of ρ is a triple of positions (p, π, π') such that $p, p\pi, p\pi' \in Pos(t)$, $\pi \neq \pi'$ is in the constraint of $\rho(p)$ and $t|_{p\pi} = t|_{p\pi'}$.*

In particular, a weak run without any equality is a run. The equalities in a run are also classified according to a particular position $p_0 \in Pos(t)$:

- (p, π, π') is *close* to p_0 iff $p \preccurlyeq p_0 \prec p\pi$ or $p \preccurlyeq p_0 \prec p\pi'$
- (p, π, π') is *far* (or *remote*) from p_0 if $p\pi \preccurlyeq p_0$ or $p\pi' \preccurlyeq p_0$

These two possible situations are depicted in figure 1.

3 Reducing GR to an Emptiness Problem for ADC

In this section, we show how to construct an ADC whose emptiness is equivalent to the ground reducibility problem and we show precisely the size of such an automaton. We start with an ADC accepting the set of irreducible ground terms (*normal forms*).

3.1 Normal Forms ADC

Let $\mathcal{L}$ be the set of left hand sides of a given rewrite system $\mathcal{R}$. Let $\mathcal{L}_1$ be the subset of the linear terms in $\mathcal{L}$, let $\mathcal{L}_2$ be its complement in $\mathcal{L}$ and let $\mathcal{L}_3$ be the set of linearized versions of terms in $\mathcal{L}_2$ (*i.e.* terms obtained by replacing in some $t \in \mathcal{L}_2$ each occurrence of a variable by a new variable, yielding a linear term).

The initial set of states Q_0 consists in all strict subterms of elements in $\mathcal{L}_1 \cup \mathcal{L}_3$ plus two special states: a single variable x which will accept all terms, and q_{r} which will accept only reducible terms of $\mathcal{R}$ (hence is a failure state). We assume that all terms are considered up to variable renaming (in particular any two terms are assumed to share no variables in what follows).

The set of states $Q_{\mathrm{NF}(\mathcal{R})}$ of the normal forms automaton consists in all unifiable subsets of $Q_0 \setminus \{q_{\mathrm{r}}\}$ plus the state q_{r}. Each element of $Q_{\mathrm{NF}(\mathcal{R})}$ is denoted q_u where u is the most general unifier (*mgu*) of the state – if it is not the special symbol “r”.

The transition rules set, denoted $\Delta_{\mathrm{NF}(\mathcal{R})}$, is the set of rules of the form:

$$f(q_{u_1}, \ldots, q_{u_n}) \xrightarrow{c} q_u$$

with:

1. if one of the q_{u_i}'s is q_r or if $f(u_1, \ldots, u_n)$ is an instance of some $s \in \mathcal{L}_1$, then $q_u = q_r$ and $c = \top$
2. if $f(u_1, \ldots, u_n)$ is not an instance of any term in $\mathcal{L}_1$, then u is the mgu of all terms $v \in Q_0 \setminus \{q_r\}$ (including the variable x) such that $f(u_1, \ldots, u_n)$ is an instance of v
3. when $q_u \neq q_r$, the constraint c is defined by:

$$\bigwedge_{\substack{\ell \in \mathcal{L}_2 \\ u, \ell \text{ are unifiable}}} \quad \bigvee_{\substack{x \text{ is a var. of } \ell \\ \ell|_\pi = \ell|_{\pi'} = x \\ \pi \neq \pi'}} (\pi \neq \pi')$$

Note that the unifier in the second condition always exists because one of the states of Q_0 is q_x, $x \in \mathcal{X}$. The final states of the normal forms automaton are all states, except q_r.

$$Q^{\mathrm{f}}_{\mathrm{NF}(\mathcal{R})} := Q_{\mathrm{NF}(\mathcal{R})} \setminus \{q_r\}$$

The normal forms automaton $\mathcal{A}_{\mathrm{NF}(\mathcal{R})}$ is defined by the above constructed sets:

$$\mathcal{A}_{\mathrm{NF}(\mathcal{R})} := (Q_{\mathrm{NF}(\mathcal{R})}, Q^{\mathrm{f}}_{\mathrm{NF}(\mathcal{R})}, \Delta_{\mathrm{NF}(\mathcal{R})})$$

This automaton $\mathcal{A}_{\mathrm{NF}(\mathcal{R})}$ is not necessary complete (the automaton may have no run on terms that are reducible by a non-left linear rule). It is however deterministic.

Example 5 The normal forms automaton $\mathcal{A}$ in example 3 is the normal form automaton $\mathcal{A}_{\mathrm{NF}(\mathcal{R})}$ where $\mathcal{R} = \{f(x, x) \to a\}$. Note that $\mathcal{A}$ is indeed deterministic but that there exists no run of $\mathcal{A}$ on *e.g.* the reducible term $f(a, a)$.

Proposition 6 *The automaton $\mathcal{A}_{\mathrm{NF}(\mathcal{R})}$ accepts the set of terms of $\mathcal{T}(\mathcal{F})$ that are irreducible by $\mathcal{R}$. Its number of states is an exponential in the size of $\mathcal{R}$. Each constraint occurring in a rule of $\mathcal{A}_{\mathrm{NF}(\mathcal{R})}$ has a size bounded by $O(\|\mathcal{R}\|^3)$.*

Proof. The constraints in the rules of $\Delta_{\mathrm{NF}(\mathcal{R})}$ are conjunctions of disjunctions of disequality atoms. The size of each of these constraints can be bounded according to the respective sizes of conjunctions, disjunctions and atoms.

$$\begin{array}{rll}
 & & \text{\# of } \wedge \qquad \text{\# of } \vee \\
\text{size} & \leq & \displaystyle\sum_{\ell \in \mathcal{L}_2} \quad \sum_{\pi, \pi' \in Pos(\ell)} (|\pi| + |\pi'|) \\
 & \leq & \displaystyle\sum_{\ell \to r \in \mathcal{R}} \quad \sum_{\pi \in Pos(\ell)} \Big(|\pi| + \sum_{\pi' \in Pos(\ell)} |\pi'|\Big) \\
 & \leq & \displaystyle\sum_{\ell \to r \in \mathcal{R}} \quad \sum_{\pi \in Pos(\ell)} (|\pi| + \|\ell\|^2) \\
 & \leq & \displaystyle\sum_{\ell \to r \in \mathcal{R}} (\|\ell\|^3 + \|\ell\|^2) \\
 & \leq & \|R\|^3
\end{array}$$

Concerning the number of states, if is sufficient to remark that it is of the same magnitude as the cardinal of the closure of $Q_0 \setminus \{q_r\}$ by mgu.

We prove the first part of proposition 6 in the two following paragraphs.

Correctness. The lemmas 7 and 8 show that $\mathcal{A}_{\mathrm{NF}(\mathcal{R})}$ recognizes normal forms of $\mathcal{R}$ only.

Lemma 7 *Let $s \in \mathcal{T}(\mathcal{F})$. If $s \xrightarrow[\Delta_{\mathrm{NF}(\mathcal{R})}]{*} q_u$ for some $q_u \in Q^{\mathrm{f}}_{\mathrm{NF}(\mathcal{R})}$, then s is an instance of u and $u =$* $\sup\{v \mid q_v \in Q_{\mathrm{NF}(\mathcal{R})}$ and s is an instance of $v\}$ *(*sup *is considered w.r.t. $\geqslant$).*

Proof. By induction on the length of the derivation $s \xrightarrow[\Delta_{\mathrm{NF}(\mathcal{R})}]{*} q_u$. □

Lemma 8 *Let $s \in \mathcal{T}(\mathcal{F})$. If $s \xrightarrow[\Delta_{\mathrm{NF}(\mathcal{R})}]{*} q_u$ for some $q_u \in Q^{\mathrm{f}}_{\mathrm{NF}(\mathcal{R})}$, then s is a normal form of $\mathcal{R}$.*

Proof. By induction on the length of the derivation $s \xrightarrow[\Delta_{\mathrm{NF}(\mathcal{R})}]{*} q_u$.
If the length is 1, s is a constant of $\mathcal{F}$. In that case there exists a rule $s \to q_u \in \Delta_{\mathrm{NF}(\mathcal{R})}$ and thus $s \notin \mathcal{L}_1$ by the first construction condition, which means that s is a normal form.
Assume $s \xrightarrow[\Delta_{\mathrm{NF}(\mathcal{R})}]{*} f(q_{u_1}, \ldots, q_{u_n}) \xrightarrow{c} q_u \in Q^{\mathrm{f}}_{\mathrm{NF}(\mathcal{R})}$, and let $s = f(s_1, \ldots, s_n)$. For each $1 \le i \le n$, we have $s_i \xrightarrow[\Delta_{\mathrm{NF}(\mathcal{R})}]{*} q_{u_i}$ and $q_{u_i} \neq q_{\mathrm{r}}$ by the first condition of the construction. Thus each s_i is a normal form by induction hypothesis. Assume now that s is reducible by $\mathcal{R}$. This means that it must be an instance of some term in $\mathcal{L}$, say $f(l_1, \ldots, l_n)$. We have two cases for $f(l_1, \ldots, l_n)$:

1. if $f(l_1, \ldots, l_n) \in \mathcal{L}_1$, for each $1 \le i \le n$, s_i is an instance of l_i. By lemma 7, this implies that for each $1 \le i \le n$, u_i is also an instance of l_i, thus $f(u_1, \ldots, u_n)$ is an instance of $f(l_1, \ldots, l_n) \in \mathcal{L}_1$ which contradicts the existence of the rule $f(q_{u_1}, \ldots, q_{u_n}) \xrightarrow{c} q_u$ ($q_u \neq q_{\mathrm{r}}$) in $\Delta_{\mathrm{NF}(\mathcal{R})}$, by the first construction condition

2. if $f(l_1, \ldots, l_n) \in \mathcal{L}_2$, s is an instance of $f(l_1, \ldots, l_n) = l$ iff s is an instance of the linearized version of l, and, for every distinct positions π, π' of l such that $l|_\pi \equiv l|_{\pi'}$, we have $s|_\pi \equiv s|_{\pi'}$. This last condition implies $s \models \neg c$ by construction of c. Hence $s \not\models c$, which of course contradicts the application of the rule $f(q_{u_1}, \ldots, q_{u_n}) \xrightarrow{c} q_u$ in the last step of the derivation $s \xrightarrow[\Delta_{\mathrm{NF}(\mathcal{R})}]{*} q_u$. (end of proof of lemma 8) □

Completeness. $\mathcal{A}_{\mathrm{NF}(\mathcal{R})}$ recognizes every ground normal forms of $\mathcal{R}$.

Lemma 9 *Let s be a term of $\mathcal{T}(\mathcal{F})$ which is a normal form of $\mathcal{R}$. There exists $q_u \in Q^{\mathrm{f}}_{\mathrm{NF}(\mathcal{R})}$ such that $s \xrightarrow[\Delta_{\mathrm{NF}(\mathcal{R})}]{*} q_u$ and s is an instance of u.*

Proof. By induction on s.
If s is a constant, then it is not (an instance of) any term in $\mathcal{L}_1$ thus we have $s \to q_x \in \Delta_{\mathrm{NF}(\mathcal{R})}$ by construction. For the induction step, let $s = f(s_1, \ldots, s_n)$. The subterms $s_1, \ldots, s_n$ are normal forms of $\mathcal{R}$. Thus by induction hypothesis, we have states $q_{u_1}, \ldots, q_{u_n} \in Q^{\mathrm{f}}_{\mathrm{NF}(\mathcal{R})} = Q_{\mathrm{NF}(\mathcal{R})} \setminus \{q_{\mathrm{r}}\}$ such that $s_i \xrightarrow[\Delta_{\mathrm{NF}(\mathcal{R})}]{*} q_{u_i}$ and s_i is an instance of u_i for all $i \le n$. Thus s is an instance of $f(u_1, \ldots, u_n)$. We proceed by contradiction. Assume that no rule in $\Delta_{\mathrm{NF}(\mathcal{R})}$ with a target in $Q^{\mathrm{f}}_{\mathrm{NF}(\mathcal{R})}$ is applicable to $f(q_{u_1}, \ldots, q_{u_n})$. Then we are in one of the following cases.

1. One of the q_{u_i}'s is q_{r} or $f(u_1, \ldots, u_n)$ is an instance of some term in $\mathcal{L}_1$ (first condition in the construction of $\Delta_{\mathrm{NF}(\mathcal{R})}$). This would contradict respectively the induction hypothesis and the irreducibility of s by lemma 7.

2. There exists $f(q_{u_1}, \ldots, q_{u_n}) \xrightarrow{c} q_u$ in $\Delta_{\mathrm{NF}(\mathcal{R})}$ for some c and $t \neq \mathrm{r}$ but $s \not\models c$. This contradicts the irreducibility of s again, by construction of the constraint c.

Hence, there exists a term u, such that $s \xrightarrow[\Delta_{\mathrm{NF}(\mathcal{R})}]{*} q_u$. Moreover, by construction, u is the most general unifier of the terms $v \in Q_0 \setminus \{q_{\mathrm{r}}\}$ such that $f(u_1, \ldots, u_n)$ is an instance of v. Thus s is an instance of u. □ (end of proof of proposition 6) □

3.2 Ground Reducibility and ADC

If t is a linear term, then its ground reducibility is equivalent to the emptiness of the intersection of $L(\mathcal{A}_{\mathrm{NF}(\mathcal{R})})$ with the (regular) set of instances of t. Since the class ADC is closed by intersection with a regular language (it can be computed in time the product of the sizes of both automata), deciding ground reducibility amounts to decide emptiness of an ADC whose number of states is $\mathrm{O}(2^{\|\mathcal{R}\|} \times \|t\|)$ and constraints have a size $\mathrm{O}(\|\mathcal{R}\|^3)$.

It is a bit more difficult when t is not linear since, in such a situation, the set of irreducible instances of t is not necessarily recognized by an ADC. For this reason, we have to compute directly an automaton whose language is empty iff t is ground reducible by $\mathcal{R}$. This ADC is denoted:

$$\mathcal{A}_{\mathrm{NF}(\mathcal{R}),t} = (Q_{\mathrm{NF}(\mathcal{R}),t}, Q^{\mathrm{f}}_{\mathrm{NF}(\mathcal{R}),t}, \Delta_{\mathrm{NF}(\mathcal{R}),t})$$

We start with the above constructed normal forms ADC $\mathcal{A}_{\mathrm{NF}(\mathcal{R})} := (Q_{\mathrm{NF}(\mathcal{R})}, Q^{\mathrm{f}}_{\mathrm{NF}(\mathcal{R})}, \Delta_{\mathrm{NF}(\mathcal{R})})$. Let $\mathcal{S}_t = \{t\sigma|_p \mid p \in Pos(t)\}$ where σ ranges over substitutions whose domain is the set of of variables occurring at least twice in t into $Q^{\mathrm{f}}_{\mathrm{NF}(\mathcal{R})}$. The cardinal of $\mathcal{S}_t$ is thus exponential in the size $\|t\|$ of t.

1. $Q_{\mathrm{NF}(\mathcal{R}),t} := \mathcal{S}_t \times Q_{\mathrm{NF}(\mathcal{R})}$

2. the final states set $Q^{\mathrm{f}}_{\mathrm{NF}(\mathcal{R}),t} := \{[u,q] \mid q \in Q_{\mathrm{NF}(\mathcal{R})},\ u \text{ is an instance of } t\}$

3. For all $f(q_1,\dots,q_n) \xrightarrow{c} q_{n+1} \in \Delta_{\mathrm{NF}(\mathcal{R})}$ and all $u_1,\dots,u_n \in \mathcal{S}_t$, $\Delta_{\mathrm{NF}(\mathcal{R}),t}$ contains the following rules:

   (a) $f([u_1,q_1],\dots,[u_n,q_n]) \xrightarrow{c\wedge c'} [f(u_1,\dots,u_n),q_{n+1}]$ if $f(u_1,\dots,u_n)$ is an instance of t and c' is defined below.

   (b) $f([u_1,q_1],\dots,[u_n,q_n]) \xrightarrow{c} [f(u_1,\dots,u_n),q_{n+1}]$ if $[f(u_1,\dots,u_n),q_{n+1}] \in Q_{\mathrm{NF}(\mathcal{R}),t}$ and we are not in the first case.

   (c) $f([u_1,q_1],\dots,[u_n,q_n]) \xrightarrow{c} [q_{n+1},q_{n+1}]$ in all other cases.

Remark that $[q_{n+1},q_{n+1}]$ is indeed one of the states of $Q_{\mathrm{NF}(\mathcal{R}),t}$
The constraint c' is constructed in three steps.

Lifting. First, all disequality constraints which are checked "in the t part" of $f(u_1,\dots,u_n) = t\sigma$, are "lifted" to the root position; this is explained in the following construction. From $f(u_1,\dots,u_n)$ we can retrieve the rules of $\mathcal{A}_{NF(\mathcal{R})}$ which are applied at any position $p \in Pos(t)$ in a run on $f(u_1,\dots,u_n)$ ($\mathcal{A}_{NF(\mathcal{R})}$ is deterministic). Let c_p be the constraint of the rule applied at position p.
We write $p\cdot c$ the constraint defined by induction on c as below and let c'_1 be $\bigwedge\limits_{p\in Pos(t)} p\cdot c_p$.

$$p\cdot\top := \top,\ p\cdot\bot := \bot \qquad p\cdot(c_1\wedge c_2) := p\cdot c_1 \wedge p\cdot c_2$$
$$p\cdot(\pi \neq \pi') := p\pi \neq p\pi' \qquad p\cdot(c_1\vee c_2) := p\cdot c_1 \vee p\cdot c_2$$

Extension. The second step consists in ensuring that all disequality constraints are "deep enough", *i.e.* below the positions of t: for each constraint $p \neq p'$ in c'_1, such that p or p' is a strict prefix of some position of t, we apply the following rule. We get then a constraint c'_2.

$$p \neq p' \rightarrow \bigvee_{\substack{p\pi\, \in\, Posl(t)\, \wedge\, p'\pi\, \notin\, Pos(t)\\ \text{or } p'\pi\, \in\, Posl(t)\, \wedge\, p\pi\, \notin\, Pos(t)}} p\pi \neq p'\pi$$

Variables. After this preparation, we take into account non-linearities of t: they imply equality constraints at the root, hence, by equational deduction, new disequality constraints can be inferred: We let c' be the constraints obtained by saturation of c'_2 using the following deduction rule for each distinct positions p_1 and p_2 in $Pos(t)$ such that $t|_{p_1} \equiv t|_{p_2}$ is a variable:

$$p_1\pi \neq p' \vdash p_2\pi \neq p'$$

Example 10 Let $\mathcal{F} = \{f,a,b\}$, $t = f(x, f(x,y))$ and $\mathcal{R} = f(x,x) \rightarrow a\}$. The automaton $\mathcal{A}_{\mathrm{NF}(\mathcal{R})}$ is (see examples 3 and 5): $\mathcal{A}_{\mathrm{NF}(\mathcal{R})} = \Big(\{q\},\{q\},\{a\rightarrow q; b\rightarrow q; f(q,q) \xrightarrow{1\neq 2} q\}\Big)$. Then the automaton $\mathcal{A}_{\mathrm{NF}(\mathcal{R}),t}$ will contain additionally the rule:

$$f\big([q_q,q],[q_{f(q,q)},q]\big) \xrightarrow{1\neq 2\wedge 1\neq 22} q$$

Proposition 11 *The term t is ground reducible by $\mathcal{R}$ iff the language accepted by $\mathcal{A}_{\mathrm{NF}(\mathcal{R}),t}$ is empty. The number of states of this automaton is $\mathrm{O}(2^{c\times\|t\|\times\|\mathcal{R}\|})$ where c is a constant. The global size of the constraints of transition rules is $\mathrm{O}(\|t\|^4 \times \|\mathcal{R}\|^3)$.*

Moreover, the number of rules of the automaton is $\mathrm{O}(2^{c\times\|t\|\times\|\mathcal{R}\|\times\alpha} \times |\mathcal{F}|)$ where α is the maximal arity of a function symbol of $\mathcal{F}$ and $|\mathcal{F}|$ is the number of function symbols.
Proof. To bound $|Q_{\mathrm{NF}(\mathcal{R}),t}|$, let us recall that $Q_{\mathrm{NF}(\mathcal{R}),t} := \mathcal{S}_t \times Q_{\mathrm{NF}(\mathcal{R})}$, and that $|Q_{\mathrm{NF}(\mathcal{R})}| = 2^{d\times\|\mathcal{R}\|}$ where d is a constraint. Moreover by construction, $|\mathcal{S}_t| \leq |Q^{\mathrm{f}}_{\mathrm{NF}(\mathcal{R})}|^{\|t\|} = \mathrm{O}(2^{d\times\|\mathcal{R}\|\times\|t\|})$ which give upper bound $\mathrm{O}(2^{c\times\|t\|\times\|\mathcal{R}\|})$ for the number of states of $\mathcal{A}_{\mathrm{NF}(\mathcal{R}),t}$.
The constraints in the rules of $\Delta_{\mathrm{NF}(\mathcal{R}),t}$ are constraints of $\Delta_{\mathrm{NF}(\mathcal{R})}$ or are conjunction of such constraints and a c' constructed as above. The c''s are conjunctions of disjunctions of conjunctions of disequality atoms

and there size after each construction step is bounded below.

$$\begin{array}{llllll}
\textit{Lifting.} & \bigwedge & \bigwedge & \bigvee & & \\
& \sum_{p \in Pos(t)} & \sum_{\ell \to r \in \mathcal{R}} & \sum_{\pi,\pi' \in Pos(\ell)} & (|\pi| + |\pi'| + 2|p|) & \\
\textit{Extention.} & \bigwedge & \bigwedge & \bigvee & \bigvee & \\
& \sum_{p \in Pos(t)} & \sum_{\ell \to r \in \mathcal{R}} & \sum_{\pi,\pi' \in Pos(\ell)} & \sum_{p' \in Pos(t)} & (|\pi| + |\pi'| + 2\|t\|) \\
\textit{Variables.} & \bigwedge & \bigwedge & \bigvee & \bigvee & \bigwedge \\
& \|t\| \times & \sum_{\ell \to r \in \mathcal{R}} & \sum_{\pi,\pi' \in Pos(\ell)} & \|t\| \times & 2\|t\| \times \;\; (|\pi| + |\pi'| + 2\|t\|) \\
\leq & 2\|t\|^3 \times \sum_{\ell \to r \in \mathcal{R}} \sum_{\pi,\pi' \in Pos(\ell)} (|\pi| + |\pi'| + 2\|t\|) & & & & \\
\leq & 2\|t\|^3 \times \|\mathcal{R}\|^3 + 2\|t\|^3 \times 2\|t\| \times \|R\|^2 & & & & \\
\leq & 2\|t\|^3 \times \|\mathcal{R}\|^3 + 4\|t\|^4 \times \|R\|^2 & & & &
\end{array}$$

The if direction of the first part of proposition 11 follows from the following lemma.

Lemma 12 *Every ground instance of t which is irreducible by $\mathcal{R}$ is accepted by $\mathcal{A}_{\mathrm{NF}(\mathcal{R}),t}$.*

Proof. Let τ be a substitution from the set of variables of t to the ground terms of $\mathcal{T}(\mathcal{F})$ such that $t\tau$ is irreducible by $\mathcal{R}$. $t \in L(\mathcal{A}_{\mathrm{NF}(\mathcal{R}),t})$ is a consequence of the following fact, proved by multiset induction.

Lemma 13 *Let τ be a substitution such that $t\tau$ is ground and irreducible by $\mathcal{R}$. For all multiset $\{\{u_1, \ldots, u_m\}\}$ of subterms of t, there exists: a substitution σ from the variables of t to $Q_{\mathrm{NF}(\mathcal{R})}$ and final states $q_1^{\mathrm{f}}, \ldots, q_m^{\mathrm{f}} \in Q^{\mathrm{f}}_{\mathrm{NF}(\mathcal{R})}$ such that for all $1 \leq i \leq m$, $u_i\tau$ is reduced by $\Delta_{\mathcal{R},t}$ into the state $[u_i\sigma, q_i^{\mathrm{f}}]$.*

Note that the substitution σ is the same for every u_i.
Proof. Let $\{\{x_1, \ldots, x_m\}\}$ be a multiset of variables of t. By hypothesis, for all $1 \leq i \leq m$, $x_i\tau$ is a normal form of $\mathcal{R}$. Thus, $x_i\tau \in L(\mathcal{A}_{\mathrm{NF}(\mathcal{R})})$ by proposition 6. This means that there exists final states $q_1^{\mathrm{f}}, \ldots, q_m^{\mathrm{f}} \in Q^{\mathrm{f}}_{\mathrm{NF}(\mathcal{R})}$ such that for each $i \leq m$, $x_i\tau \xrightarrow[\Delta_{\mathrm{NF}(\mathcal{R})}]{*} q_i^{\mathrm{f}}$.
Thus each $x_i\tau$ is reduced by $\Delta_{\mathrm{NF}(\mathcal{R}),t}$ into the state $[q_i^{\mathrm{f}}, q_i^{\mathrm{f}}]$. We can moreover assume that for all $1 \leq i_1, i_2 \leq m$ such that $x_{i_1} = x_{i_2}$, we have $q_{i_1}^{\mathrm{f}} = q_{i_2}^{\mathrm{f}}$. This give the substitution σ from $\{x_1, \ldots, x_m\}$ to $\{q_1^{\mathrm{f}}, \ldots, q_m^{\mathrm{f}}\}$.
Let $\{\{u_1, \ldots, u_m\}\}$ be a multiset of subterms of t, such that one u_j $(1 \leq j \leq m)$ at least is not a variable. We let $u_j = f(v_1, \ldots, v_n)$. By induction hypothesis for the multiset:

$$\{\{u_1, \ldots, u_{j-1}, v_1, \ldots, v_n, u_{j+1}, \ldots, u_m\}\}$$

there exists final states $q_1^{\mathrm{f}}, \ldots, q_{m+n-1}^{\mathrm{f}} \in Q^{\mathrm{f}}_{\mathrm{NF}(\mathcal{R})}$ and a substitution τ from the set of variable of t to $Q_{\mathrm{NF}(\mathcal{R})}$ such that for all $1 \leq i < j$ and all $j < i \leq m$, $u_i\tau$ is reduced by $\mathcal{A}_{\mathrm{NF}(\mathcal{R}),t}$ into the state $[u_i\sigma, q_i^{\mathrm{f}}]$.
Moreover, by hypothesis, $u_j\tau$ is irreducible by $\mathcal{R}$, thus accepted by $\mathcal{A}_{\mathrm{NF}(\mathcal{R})}$ which is deterministic (see proposition 6). Thus, $\Delta_{\mathrm{NF}(\mathcal{R})}$ contains a transition rule of the form: $f(q_j^{\mathrm{f}}, \ldots, q_{j+n-1}^{\mathrm{f}}) \xrightarrow{c} q^{\mathrm{f}}$ with $q^{\mathrm{f}} \in Q^{\mathrm{f}}_{\mathrm{NF}(\mathcal{R})}$ and $u_j\tau \models c$.
And $\Delta_{\mathrm{NF}(\mathcal{R}),t}$ contains a transition rule:

$$f([v_1\sigma, q_j^{\mathrm{f}}], \ldots, [v_n\sigma, q_{j+n-1}^{\mathrm{f}}]) \xrightarrow{c''} [f(v_1, \ldots, v_n)\sigma, q^{\mathrm{f}}]$$

where $c'' = c \wedge c'$ if $f(v_1, \ldots, v_n)\sigma$ is an instance of t and $c'' = c$ otherwise.
This gives the final states: $q_1^{\mathrm{f}}, \ldots, q_{i-1}^{\mathrm{f}}, q^{\mathrm{f}}, q_{i+1}^{\mathrm{f}}, \ldots, q_m^{\mathrm{f}}$ and the substitution σ we wanted for the multiset $\{\{u_1, \ldots, u_{i-1}, u_i, u_{i+1}, \ldots, u_m\}\}$ □

The only if direction of the first part of proposition 11 is now proved with the help of three intermediate lemmas. The automaton $\mathcal{A}_{\mathrm{NF}(\mathcal{R}),t}$ does not recognize only irreducible ground instances of t. However, we are going to show that if u is accepted then we can construct a term u' which is an irreducible instance of t and which is still accepted (u' is thus a witness for non ground reducibility of t by $\mathcal{R}$).

Lemma 14 *Each term of $L(\mathcal{A}_{\mathrm{NF}(\mathcal{R}),t})$ is a normal form of $\mathcal{R}$.*

Proof. By construction, if we transform $\mathcal{A}_{\mathrm{NF}(\mathcal{R})}$ with a projection on the first component of the states of $Q_{\mathrm{NF}(\mathcal{R}),t}$, we obtain exactly the normal form ADC $\mathcal{A}_{\mathrm{NF}(\mathcal{R})}$ of proposition 6. □

Lemma 15 *Each term of $L(\mathcal{A}_{\mathrm{NF}(\mathcal{R}),t})$ is a ground instance of the linearized version of t.*

Proof. We may show by induction the more general fact that each term of $\mathcal{T}(\mathcal{F})$ recognized by $\mathcal{A}_{\mathrm{NF}(\mathcal{R}),t}$ in the state $[u\sigma, q] \in Q_{\mathrm{NF}(\mathcal{R}),t}$ ($u\sigma \in \mathcal{S}_t$ and $u = t|_p$ for some $p \in Pos(t)$) is a ground instance of the linearized version of u, by induction on u. □

Lemma 16 *Let ρ be a run of $\mathcal{A}_{\mathrm{NF}(\mathcal{R}),t}$ and a position $p \in Pos(\rho)$ such the target state of $\rho(p)$ is $[u, q]$. Then for all position $p' \in Pos(u)$, if $u(p') = q' \in Q_{\mathrm{NF}(\mathcal{R})}$, then the target state of $\rho(pp')$ is $[q', q']$.*

Proof. by induction on ρ. □

Now, we can terminate the proof of the only if direction of the first part of proposition 11. Assume we have $s \in L(\mathcal{A}_{\mathrm{NF}(\mathcal{R}),t})$. Let ρ be a run of $\mathcal{A}_{\mathrm{NF}(\mathcal{R}),t}$ on this s. By lemma 15, s is a ground instance of the linearized version of t. It could happen though that s is a actually not an instance of t itself, because we have in s two distinct subterms at positions $p_1, p_2 \in Pos(t)$ corresponding to the same variable in t ($t(p_1) = t(p_2) \in \mathcal{X}$).
The idea is then to construct s', a ground instance of t by replacing $s|_{p_1}$ with $s|_{p_2}$ in s. Lets associate to each variable $x \in Var(t)$ the set $\mathrm{occ}_t(x)$ of positions of x in t,

$$\mathrm{occ}_t(x) := \{p \in Pos(t) \mid t(p) = x\}$$

And let $[t\sigma, q]$ be the (final) target state of $\rho(\Lambda)$.
For each variable x in t, we note $q_x := x\sigma$, which is by construction a state of $Q_{\mathrm{NF}(\mathcal{R})}$. From the lemma 16, for each $p \in \mathrm{occ}_t(x)$, the target state of $\rho(p)$ is $[q_x, q_x]$.

We then construct a weak run ρ' as follows: for each $x \in Var(t)$, if $|\mathrm{occ}_t(x)| \geq 1$, we choose $p \in \mathrm{occ}_t(x)$ and do the replacement $\rho[\rho|_p]_{p'}$ for all other $p' \in \mathrm{occ}_t(x) \setminus \{p\}$. To show that ρ' is indeed a run of $\mathcal{A}_{\mathrm{NF}(\mathcal{R}),t}$, we have to check that the constraints in ρ' are still valid after the replacements, in other words, that ρ' contains no equalities. This follows from the fact that the constraint of $\rho(\Lambda)$ is satisfied by s, in particular the subconstraint c' constructed as above.
We shall first remark that c' contains every constraints which are valid in ρ and may not be valid in ρ', because of the first part, *lifting*, of the construction of c'. There are only two kind of equalities which may occur in ρ' (see lemma 18 in section 4):

1. (p, π, π') such that there is α such that $p\pi\alpha \in \mathrm{occ}_t(x)$ for some variable x, we performed the replacement $\rho[\rho|_{p\pi\alpha}]_\beta$ (for some other $\beta \in \mathrm{occ}_t(x)$), and $s|_{p\pi'\alpha} = s|_\beta$, which is the cause of the equality $s|_{p\pi} = s|_{p\pi'}$. This situation is not possible because of the transformations performed in the part *extension* in the construction of c'.

2. $(p, \alpha\pi, \pi')$ such that $p\alpha \in \mathrm{occ}_t(x)$ for some variable x, we performed the replacement $\rho[\rho|_{p\alpha}]_\beta$ (for some other $\beta \in \mathrm{occ}_t(x)$), and $s|_{p\pi'} = s|_{\beta\pi}$. This situation is made impossible by the completion in the part called *variables* in the construction of c'.

The term s' is the term of $\mathcal{T}(\mathcal{F})$ associated to ρ' and it is a ground instance of t. Moreover, by lemma 14, s' is irreducible by $\mathcal{R}$. □

(end of proof of proposition 11) □

4 Generalized Pumping Lemmas

This is the crux part of our proof. We assume here a given ADC $\mathcal{A} = (Q, Q^{\mathrm{f}}, \Delta)$ and a well founded ordering $\gg$, total on runs[2] of $\mathcal{A}$ and monotonic (*i.e.* $\rho \gg \rho'$ implies that for every ground term s and any position $p \in Pos(s)$, $s[\rho]_p \gg s[\rho']_p$).

Definition 17 *A* pumping *(w.r.t. $\gg$) is a replacement $\rho[\rho']_p$ where ρ, ρ' are runs such that the target state of $\rho'(\Lambda)$ is the same as the target state of $\rho(p)$ and $\rho \gg \rho[\rho']_p$.*

This definition generalizes the usual pumping definition: we get the usual pumping if we choose for $\gg$ the *embedding ordering.*

[2] Let us recall that runs of $\mathcal{A} = (Q, Q^{\mathrm{f}}, \Delta)$ can be seen as terms of $\mathcal{T}(\Delta)$.

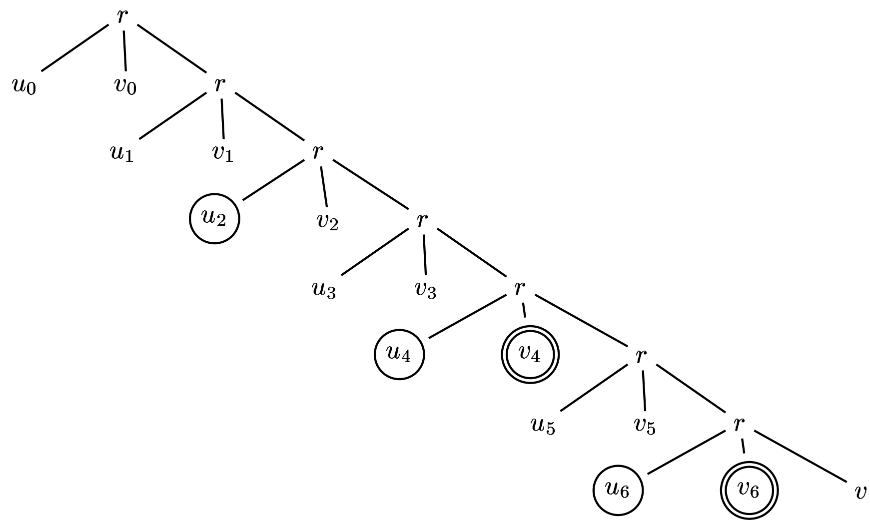

Figure 2: A run with a possible pumping.

Lemma 18 *Every pumping $\rho[\rho']_p$ is a weak run and every equality in $\rho[\rho']_p$ is either far from p or close to p.*

Proof. $\rho[\rho']_p$ is a weak run because the target states of $\rho(p)$ and $\rho'(\Lambda)$ are the same. Let (p', π, π') be an equality of $\rho[\rho']_p$. By definition, ρ and ρ' are runs, thus they do not contain equalities, thus $p' \not\geq p$ and $p' \not\parallel p$. The only remaining possibilities are equalities close to p or far from p. □
Hence, in the following, we may refer to close and far equalities in pumpings, forgetting the position p. Given a large enough run ρ, we will successively show how to construct a weak run by pumping which does not contain any close equality (this uses combinatorial arguments only) then we show how to remove far equalities by further successive pumpings.

4.1 Pumping without Creating Close Equalities

Given an ADC $\mathcal{A} = (Q, Q^{\mathrm{f}}, \Delta)$ and an integer k, we let:

$$g(\mathcal{A}, k) := (e \times k + 1) \times |Q| \times 2^{c(\mathcal{A})} \times c(\mathcal{A})!$$

where e is the exponential basis ($e := \sum_{n=0}^{+\infty} \frac{1}{n!}$) and $c(\mathcal{A})$ is the number of distinct suffixes of positions π, π' in an atom $\pi \neq \pi'$ occurring in a constraint of transition rule of $\mathcal{A}$. Then we have a pumping lemma which generalizes those of [5, 4].

Lemma 19 *Let k be an integer. If ρ is a run of $\mathcal{A}$ and $p_1, \ldots, p_{g(\mathcal{A},k)}$ are positions of ρ such that $\rho|_{p_1} \gg \ldots \gg \rho|_{p_{g(\mathcal{A},k)}}$ then there are indices $i_0 < i_1 < \ldots < i_k$ such that the pumping $\rho[\rho|_{p_{i_j}}]_{p_{i_0}}$ does not contain any close equality.*

Example 20 This example illustrates the principle of the proof of lemma 19. Let $\mathcal{F}$ contain a ternary symbol f and let the ADC $\mathcal{A}$ contain the following transition rule:

$$r: \quad f(q_1, q_2, q_3) \xrightarrow{1 \neq 31 \wedge 1 \neq 32} q$$

Consider moreover the following run (which is also depicted on figure 2):

$$\rho = r(u_0, v_0, r(u_1, v_1, r(u_2, v_2, r(u_3, v_3, r(u_4, v_4, r(u_5, v_5, r(u_6, v_6, v)))))))$$

We show that ρ is large enough so as to be able to find a pumping which does not create any close equality. Assume first that the replacement of the subtree at position 3 in ρ by any other subtree rooted by r (except ρ itself) creates a close equality. This means that, for all $i = 2, \ldots, 6$, $u_i = u_0$ or $v_i = v_0$. Then it is possible

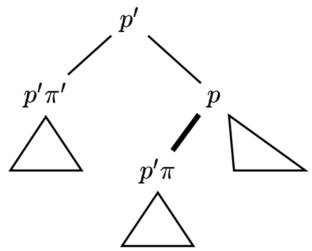

Figure 3: The bold branch is in $\mathrm{cr}(p)$.

to extract a subsequence of three indices i_1, i_2, i_3 such that $(u_0 = u_{i_1} = u_{i_2} = u_{i_3}) \vee (v_0 = v_{i_1} = v_{i_2} = v_{i_3})$. Assume we are in the first case of the disjunction and that, for instance $u_0 = u_2 = u_4 = u_6$. Now, we try to replace the subterm $r(u_2, v_2, \ldots)$ with $r(u_4, v_4, \ldots)$ and $r(u_6, v_6, v)$ respectively. Since $u_2 = u_4 = u_6 \neq u_1$, if each of these replacements creates a close equality, we must have $v_1 = v_4 = v_6$. Finally, replacing $r(u_4, v_4, \ldots)$ by $r(u_6, v_6, v)$, we cannot create a close equality since $u_6 = u_4 \neq u_3$ and $v_6 = v_4 \neq v_3$.

Proof.(lemma 19) We can first extract from $p_1 \ldots p_{g(\mathcal{A},k)}$ a subsequence $p_{l_0} \ldots p_{l_{k_1}}$ such that $\rho(p_{l_0}) \ldots \rho(p_{l_{k_1}})$ have all the same target state, with:

$$k_1 := \frac{g(\mathcal{A}, k)}{|Q|} = (e \times k + 1) \times 2^{c(\mathcal{A})} \times c(\mathcal{A})!$$

Let us define $u_0 := p_{l_0}, \ldots, u_{k_1} := p_{l_{k_1}}$ To extract a second subsequence we use a function $\mathrm{test}(p)$ defined on the positions of $Pos(\rho)$ and such that for all $p \in Pos(\rho)$:

$$\mathrm{test}(p) = \left\{ (p', \pi) \;\middle|\; \begin{array}{l} p' \prec p \preccurlyeq p'\pi \\ \exists \pi' \text{ s.t. } (\pi \neq \pi') \text{ or } (\pi' \neq \pi) \text{ is a constraint of } \rho(p') \end{array} \right\}$$

With this function $\mathrm{test}(p)$, we associate to each position $p \in Pos(\rho)$ a set of positions $\mathrm{cr}(p)$ defined by:

$$\mathrm{cr}(p) := \{(p'\pi)/p \mid (p', \pi) \in \mathrm{test}(p)\}$$

The quotient $(p'\pi)/p$ of two positions is defined by: $pp'/p := p'$. The figure 3 illustrates the definition of $\mathrm{cr}(p)$. Note that if $(p', \pi) \in \mathrm{test}(p)$, then $(p'\pi)/p$ is well defined.

Fact 1. For all $p \in Pos(\rho)$, $\mathrm{cr}(p)$, $|\mathrm{cr}(p)| \leq c(\mathcal{A})$.

We see that for all $p \in Pos(\rho)$, $\mathrm{cr}(p)$ is included in the set of suffixes of positions π and π' such that $(\pi \neq \pi')$ is a atomic constraint occurring in one of the rules of Δ. Thus the number of distinct sets $\mathrm{cr}(p)$ for $p \in Pos(\rho)$ is smaller than $2^{c(\mathcal{A})}$. We can extract a subsequence $u_{l'_0} \ldots u_{l'_{k_2}}$ of positions from $u_0 \ldots u_{k_1}$ such that $\mathrm{cr}(u_{l'_0}) = \ldots = \mathrm{cr}(u_{l'_{k_2}})$, with:

$$k_2 := \frac{k_1}{2^{c(\mathcal{A})}} = (e \times k + 1) \times c(\mathcal{A})!$$

We note $v_0 := u_{l'_0}, \ldots, v_{k_2} := u_{l'_{k_2}}$. Then we are going to show that we can finally extract from $v_0 \ldots v_{k_2}$ another subsequence corresponding to the one in lemma 19. This is a consequence of the following intermediate lemma 21. Some additional definitions and notations are used in this lemma 21. The *dependency degree* of a subsequence $v_{i_0} \ldots v_{i_m}$ of $v_0 \ldots v_{k_2}$ is:

$$\mathrm{dep}(v_{i_0} \ldots v_{i_m}) := |\{\beta \in \mathrm{cr}(v_0) \mid t|_{v_{i_0}\beta} = \ldots = t|_{v_{i_m}\beta}\}|$$

where $t \in \mathcal{T}(\mathcal{F})$ is the term associated[3] to ρ.
Let $f(n)$ be an integer function recursively defined on the interval $[0 \ldots c(\mathcal{A})]$ by:

$$\begin{array}{rcll} f(c(\mathcal{A})) & = & k & \\ f(n) & = & (c(\mathcal{A}) - n) \times (f(n+1) + 1) + k - 1 & \text{for } n < c(\mathcal{A}) \end{array}$$

[3]the term associated to the run ρ is the term on which the ADC $\mathcal{A}$ makes the run ρ.

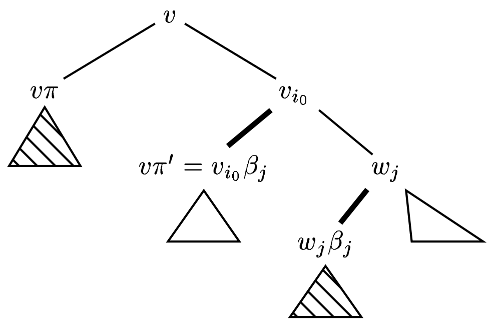

Figure 4: Definition of β_j, proof of lemma 21.

Lemma 21 *If for all* $0 \leq j \leq k_2$*, the cardinal* $|\{j' \mid k_2 \geq j' > j,\ \rho[\rho|_{v_{j'}}]_{v_j}$ *has no close equality*$\}|$ *is smaller than* k *then for all* $0 \leq n \leq c(\mathcal{A})$*, there exists a subsequence* $v_{i_0} \dots v_{i_{f(n)}}$ *of* $v_0 \dots v_{k_2}$ *such that* $\mathrm{dep}(v_{i_0} \dots v_{i_{f(n)}}) \geq n$*.*

Proof. We assume that the hypothesis of lemma 21 is true and we prove the conclusion by induction on n. For $n = 0$, by definition of the function dep, for every subsequence $v_{i_0} \dots v_{i_m}$ of $v_0 \dots v_{k_2}$, we have $\mathrm{dep}(v_{i_0} \dots v_{i_m}) \geq 0$. Thus on this case, it is sufficient to show that $f(0) \leq k_2$. Let $F(n) = f(c(\mathcal{A}) - n)$ for all $0 \leq n \leq c(\mathcal{A})$.

$$\begin{array}{rcll} F(0) & = & k & \\ F(n) & = & n(F(n-1)+1) + k - 1 & \text{for } 1 \leq n \leq c(\mathcal{A}) \end{array}$$

Developing,

$$\begin{array}{rcl} F(n) & = & n! \times (F(0)+1) + k \times n! \displaystyle\sum_{i=1}^{n} \frac{1}{i!} - 1 \\ & \leq & k \times n! + n! + k \times n! \times (e-1) - 1 \\ & \leq & n! \times (k \times e + 1) \end{array}$$

Thus,

$$\begin{array}{rcl} f(0) & = & F(c(\mathcal{A})) \\ & \leq & c(\mathcal{A})! \times (k \times e + 1) \\ & \leq & (e \times k + 1) \times c(\mathcal{A})! \\ & \leq & k_2 \end{array}$$

For $n+1$, assume that the property is true for $n < c(\mathcal{A})$. By induction hypothesis, we have a subsequence $v_{i_0} \dots v_{i_{f(n)}}$ extracted from $v_0 \dots v_{k_2}$ such that $\mathrm{dep}(v_{i_0} \dots v_{i_{f(n)}}) \geq n$. Moreover, by the hypothesis of lemma 21, for at least $f(n) - (k-1) = (c(\mathcal{A}) - n) \times (f(n+1)+1)$ positions w among $v_{i_1} \dots v_{i_{f(n)}}$, $\rho[\rho|_w]_{v_{i_0}}$ has a close equality. We let:

$$k_3 = (c(\mathcal{A}) - n) \times (f(n+1)+1)$$

and we let $w_1 \dots w_{k_3}$ be the above positions w, assuming that $w_1 \dots w_{k_3}$ is a subsequence of $v_{i_1} \dots v_{i_{f(n)}}$. By definition of close equalities, for all j, $1 \leq j \leq k_3$, there exists $\beta_j \in \mathrm{cr}(v_{i_0}) = \mathrm{cr}(v_0)$, there exists $v \prec v_{i_0}$ and $(\pi \neq \pi')$ an atomic constraint in $\rho(v)$ such that (we only consider one case because of the symmetry):

$$\begin{array}{rcl} v_{i_0}\beta_j & = & v\pi' \\ t|_{v_{i_0}\beta_j} & \neq & t|_{v\pi} \\ t|_{w_j\beta_j} & = & t|_{v\pi} \end{array} \qquad \begin{array}{r} (1) \\ (2) \\ (3) \end{array}$$

Lets recall that $t \in \mathcal{T}(\mathcal{F})$ is the term associated to the run ρ. The construction of β_j is depicted on figure 4. By definition of $\mathrm{dep}(v_{i_0} \dots v_{i_{f(n)}}) \geq n$, there exists a subset $E \subseteq \mathrm{cr}(v_0) = \mathrm{cr}(v_{i_0})$ such that:

$$|E| = n \qquad (4)$$

$$\text{for all } \beta \in E,\ t|_{v_{i_0}\beta} = \dots = t|_{v_{i_{f(n)}}\beta} \qquad (5)$$

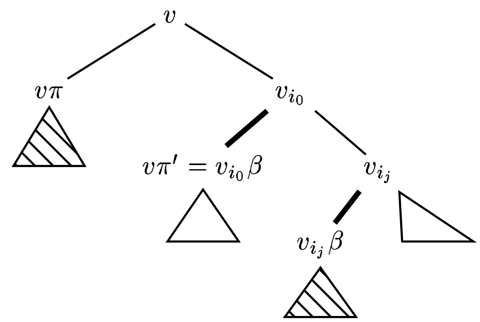

Figure 5: Proof of lemma 19.

In particular, for all $\beta \in E, t|_{v_{i_0}\beta} = t|_{w_1\beta} \ldots = t|_{w_{k_3}\beta}$. Hence, $\{\beta_1 \ldots \beta_{k_3}\} \cap E = \emptyset$ by (2) and (3). Moreover, according to the above fact 1, $|\mathrm{cr}(v_0) \leq c(\mathcal{A})$. This implies that there are at most $c(\mathcal{A}) - n$ distinct positions among $\beta_1 \ldots \beta_{k_3}$. Thus there exists: $1 \leq j_0 < \ldots < j_{f(n+1)} \leq k_3$ such that $\beta_{j_0} = \ldots = \beta_{j_{f(n+1)}}$, because $\frac{k_3}{c(\mathcal{A})-n} = f(n+1)+1$. Let β' this unique position. By construction:

$$t|_{w_{j_0}\beta'} = \ldots = t|_{w_{j_{f(n+1)}}\beta'} \tag{6}$$

Let us recall that by definition of E, $\beta' \notin E$, hence:

$$\mathrm{dep}(w_{j_0} \ldots w_{j_{f(n+1)}}) > \mathrm{dep}(v_{i_0} \ldots v_{i_{f(n)}}) \geq n \tag{7}$$

This completes the proof of lemma 21 because $w_{j_0} \ldots w_{j_{f(n+1)}}$ is a subsequence of $v_0 \ldots v_{k_2}$.

(end of the proof of lemma 21) □

Now, we have to finish the proof of lemma 19. We will show that the hypothesis of lemma 21 cannot be true. Assume it is true. Thus, for $n = c(\mathcal{A})$ and $f(n) = k$, there exists a subsequence $v_{i_0} \ldots v_{i_k}$ of $v_0 \ldots v_{k_2}$ such that $\mathrm{dep}(v_{i_0} \ldots v_{i_{f(n)}}) \geq c(\mathcal{A})$. But, by the above fact, $|\mathrm{cr}(v_0)| \leq c(\mathcal{A})$, thus by definition of $\mathrm{dep}(v_{i_0} \ldots v_{i_{f(n)}})$ we have:

$$\text{for all } \beta \in \mathrm{cr}(v_0),\ t|_{v_{i_0}\beta} = \ldots = t|_{v_{i_k}\beta} \tag{8}$$

Assume now that one of the pumping $\rho[\rho|_{v_{i_j}}]_{v_{i_0}}$ for $1 \leq j \leq k$ has a close equality. This means that there exists $v \prec v_{i_0}$ and $(\pi \neq \pi')$, an atomic constraint in $\rho(v)$ such that, $v \prec v_{i_0} \preccurlyeq v\pi'$ and $t|_{v\pi} = t|_{v_{i_j}\beta}$. The position $\beta := (v\pi')/v_{i_0} \in \mathrm{cr}(v_{i_0})$ is such that $t|_{v\pi} \neq t|_{v_{i_0}\beta}$ and with $t|_{v\pi} = t|_{v_{i_j}\beta}$ this contradicts $t|_{v_{i_0}\beta} = t|_{v_{i_j}\beta}$ (see figure 5).

Thus for all $1 \leq j \leq k$, the pumping $\rho[\rho|_{v_{i_j}}]_{v_0}$ does not have any close equality. This completes the proof of lemma 19. □

4.2 Pumping without Creating Equalities

Definition 22 *$\mathcal{M}$ is the predicate (defined relatively to an ADC $\mathcal{A}$ and an ordering $\gg$) which holds true on a run ρ of $\mathcal{A}$, a position p of ρ and an integer k iff there exists k runs $\rho|_p \gg \rho_k \gg \ldots \gg \rho_1$ such that $\rho(p), \rho_1(\Lambda), \ldots, \rho_k(\Lambda)$ have the same target state and for every $1 \leq i \leq k$ the pumping $\rho[\rho_i]_p$ does not contain any close equality.*

We list without proof two obvious consequences of definition 22.

Lemma 23 *If $k \geq k'$ then $\mathcal{M}(\rho, p, k)$ implies $\mathcal{M}(\rho, p, k')$.*

Lemma 24 *If a run ρ is such that $\mathcal{M}(\rho, \Lambda, k)$ for some $k \geq 1$, then there exists a run $\rho' \ll \rho$ such that the target states of $\rho'(\Lambda)$ and $\rho(\Lambda)$ are the same.*

Let

$$h(\mathcal{A}, k) := \big(d(\mathcal{A}) + 1\big) \times n(\mathcal{A}) \times \Big[k + g\big(\mathcal{A}, k + 2d(\mathcal{A}) \times n(\mathcal{A})\big)\Big]$$

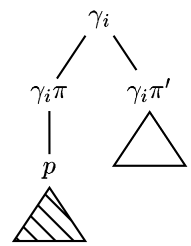

Figure 6: The far equality (γ_j, π, π') in $\rho[\rho_j]_p$.

where $n(\mathcal{A})$ is the maximal number of atomic constraints occurring in a rule of $\mathcal{A}$ and $d(\mathcal{A})$ is the maximal length $|\pi|$ or $|\pi'|$ among every atomic constraints $\pi \neq \pi'$ in the transitions rule of $\mathcal{A}$.

The following propagation lemma is the crux part of our proof. (It is also very technical to prove). It explains how to get rid of remote equalities, if we have enough pumpings which do not create close equalities. The underlying intuitive idea behind lemma 25 is the following. If we assume $h(\mathcal{A}, k)$ pumpings below p, which do not create close equalities (it will be possible to construct such pumpings thanks to lemma 19), either one of them yields a run, and we completed our goal, or each of them contains a remote equality. However, all these remote equalities give us some information on the structure of the original run, and we are going to take advantage of this to design new other pumpings, which, combined with the original ones, ensure again $h(\mathcal{A}, k)$ pumpings below $p' < p$ each of them not containing equalities below p'. This allows to prime an induction: we can construct pumpings such that $\rho|_{p'}[\rho_i]$ is a run, provided that $\rho|_p[\rho_j]$ is a run. Eventually, we will have $p' = \Lambda$ and hence a pumping which is a run.

Lemma 25 (Propagation lemma) *Let ρ be a run of $\mathcal{A}$, $p \in Pos(\rho)$ and k be an integer such that $k^2 \geq h(\mathcal{A}, k)$. If $\mathcal{M}(\rho, p, h(\mathcal{A}, k))$ is true, then one of the following properties holds:*

1. *there is a run ρ' such that $\rho|_p \gg \rho'$ and $\rho[\rho']_p$ is a run,*
2. *there exists a position p' such that $|p'| < |p|$ and $\mathcal{M}\big(\rho, p', h(\mathcal{A}, k)\big)$ is true.*

We shall show below (page 18) that such an integer k exists, and depends on $\mathcal{A}$.

Proof. Assume $\mathcal{M}(\rho, p, h(\mathcal{A}, k))$ is true. This means that we have $h(\mathcal{A}, k)$ runs $\rho_1, \ldots, \rho_{h(\mathcal{A},k)} \in \mathcal{T}(\Delta)$ such that $\rho|_p \gg \rho_i$ and $\rho[\rho_i]_p$ does not create a close equality, for $1 \leq i \leq h(\mathcal{A}, k)$, following the definition 22 of $\mathcal{M}()$. If we are not in the first case of the lemma, then for each $1 \leq i \leq h(\mathcal{A}, k)$, $\rho[\rho_i]_p$ contains far equalities.

For each index $j \leq h(\mathcal{A}, k)$, let γ_j be a maximal position w.r.t. prefix ordering such that (γ_j, π, π') is a (far) equality of $\rho[\rho_j]_p$, see figure 6. Let E be the set of triples (γ_j, π, π'). We have $|E| = h(\mathcal{A}, k)$.
Moreover, the number of distinct first components of elements of the E is:

$$\begin{array}{rcl} \Big|\{\gamma \mid \exists \pi, \pi' \ (\gamma, \pi, \pi') \in E\}\Big| & \geq & \frac{|E|}{n(\mathcal{A})} \\ & \geq & \big(d(\mathcal{A}) + 1\big) \times \Big[k + g\big(\mathcal{A}, k + 2d(\mathcal{A}) \times n(\mathcal{A})\big)\Big] \end{array} \tag{9}$$

Note that every position γ_j is a prefix of p (since ρ and ρ_i are runs) hence the set of first components of E can be totally ordered by the prefix ordering.
Let u_i, $1 \leq i \leq \big(d(\mathcal{A}) + 1\big) \times \big[k + g\big(\mathcal{A}, k + 2d(\mathcal{A}) \times n(\mathcal{A})\big)\big]$, be a strictly decreasing sequence (w.r.t. the prefix ordering) of first components of elements in E.

We are going to show that $\mathcal{M}(\rho, u_i, k^2)$ is true for some i, which implies the second case of the lemma 25 by hypothesis and by lemma 23.
First, we extract from the sequence (u_i) a subsequence (p_i) of length $k_1 := k + g\big(\mathcal{A}, k + 2d(\mathcal{A}) \times n(\mathcal{A})\big)$ defined by:

$$p_i = u_{(d(\mathcal{A})+1) \times i} \text{ for all } 1 \leq i \leq k_1 \tag{10}$$

This ensures that two positions u_i and u_j are distant enough.
To each integer $1 \leq i \leq k_1$, we can associate a unique index $\nu(i) \leq h(\mathcal{A}, k)$ defined by $p_i = \gamma_{\nu(i)}$. By construction, for every equality (γ, π, π') of the pumping $\rho[\rho_{\nu(i)}]_p$ one has $\gamma \preccurlyeq p_i$ (p_i is itself one of these γ).

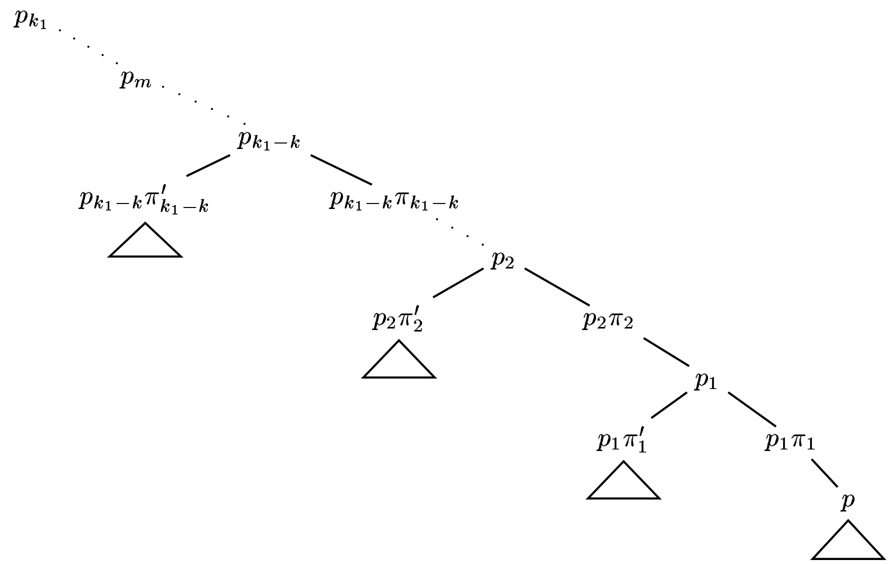

Figure 7: Proof of lemma 25.

Now, we consider any given pumping $\rho[\rho_{\nu(m)}]_p$ for $k_1 - k + 1 \leq m \leq k_1$ (*i.e.* p_m is one of the k smallest positions p_i) and we show that there is one position $p_{i_{m,0}}$, $i_{m,0} \leq k_1 - k$, and k other pumpings on $\rho[\rho_{\nu(m)}]_p$ whose equalities are far from p_m. For sake of simplicity, we note $\rho'_m := \rho[\rho_{\nu(m)}]_p$.
By construction, to each $1 \leq i \leq k_1$ we can associate some positions π_i and π'_i such that $(p_i, \pi_i, \pi'_i) \in E$ (*i.e.* it is a far equality of some $\rho[\rho_{\nu(i)}]_p$). Moreover, with this construction, by definition of far equalities, for each i, we have either $p \prec p_i\pi_i$ or $p \prec p_i\pi'_i$ (p is from the hypotheses of lemma 25). By symmetry, we assume that we are in the first case for all i. Note that by construction of (p_i), and by definition of $d(\mathcal{A})$, we have:

$$p_1 \succ p_1\pi_1 \succ p_2 \succ p_2\pi_2 \succ \ldots \succ p_{k_1-k} \succ p_m \tag{11}$$

The situation is depicted in figure 7. The equality (p_i, π_i, π'_i) is a far equality, hence, for every i, p_i, $p_i\pi_i$ and $p_i\pi'_i$ are indeed positions of ρ'_i and $\rho'_i|_{p_i\pi_i} = \rho|_{p_i\pi'_i}$. Following (11), $\rho|_{p_l\pi'_l}$ is a subterm of $\rho'_i|_{p_i\pi'_i}$ for $l > i$. It follows that the terms $\rho|_{p_i\pi'_i}$ are pairwise distinct.

Hence, we can apply the lemma 19 to the run ρ'_m and the positions $p_1\pi'_1, \ldots, p_{k_1-k}\pi'_{k_1-k}$ (remember that $k_1 - k = g\big(\mathcal{A}, k + 2d(\mathcal{A}) \times n(\mathcal{A})\big)$). This yields a subsequence $(p_{i_{m,j}}\pi'_{i_{m,j}})$, with $0 \leq j \leq k + 2d(\mathcal{A}) \times n(\mathcal{A})$, such that every pumping $\rho'_m[\rho'_m|_{p_{i_{m,j}}\pi'_{i_{m,j}}}]_{p_{i_{m,0}}\pi'_{i_{m,0}}}$ does not contain close equalities. The pumpings $\rho'_m[\rho'_m|_{p_{i_{m,j}}\pi'_{i_{m,j}}}]_{p_{i_{m,0}}\pi'_{i_{m,0}}}$ may though contain some far equalities. In the following array, we give (upper bounds for) the number of these far equalities, w.r.t. the positions $p_{i_{m,0}}$ and $p_{i_{m,0}}\pi'_{i_{m,0}}$. See also figure 8 for a picture of the 3 situations.

Equalities	Max. number
(γ, π, π') far from $p_{i_{m,0}}\pi'_{i_{m,0}}$ and $p_{i_{m,0}} \prec \gamma$	$d(\mathcal{A}) \times n(\mathcal{A})$
(γ, π, π') far from $p_{i_{m,0}}\pi'_{i_{m,0}}$ and close to $p_{i_{m,0}}$	$d(\mathcal{A}) \times n(\mathcal{A})$
(γ, π, π') far from $p_{i_{m,0}}\pi'_{i_{m,0}}$ and far from $p_{i_{m,0}}$	$\lvert p_{i_{m,0}}\rvert \times n(\mathcal{A})$

For the first two lines of this array, there are at most $d(\mathcal{A})$ possible positions for γ and at most $n(\mathcal{A})$ possible equalities for each of these positions. For the last line, the maximal number of positions is $|p_{i_{m,0}}|$. Note that every equality in one of the pumpings $\rho'_m[\rho'_m|_{p_{i_{m,j}}\pi'_{i_{m,j}}}]_{p_{i_{m,0}}\pi'_{i_{m,0}}}$ is registered in this array, by lemma 18. Thus, there exists at least k pumpings of the form $\rho'_m[\rho'_m|_{p_{i_{m,j}}\pi'_{i_{m,j}}}]_{p_{i_{m,0}}\pi'_{i_{m,0}}}$ every equality of which is far from $p_{i_{m,0}}$.

Every equality in ρ'_m itself is also far for $p_{i_{m,0}}$ since:

1. the first component of such each an equality is one of $p_{k_1-k+1}, \ldots, p_{k_1}$,

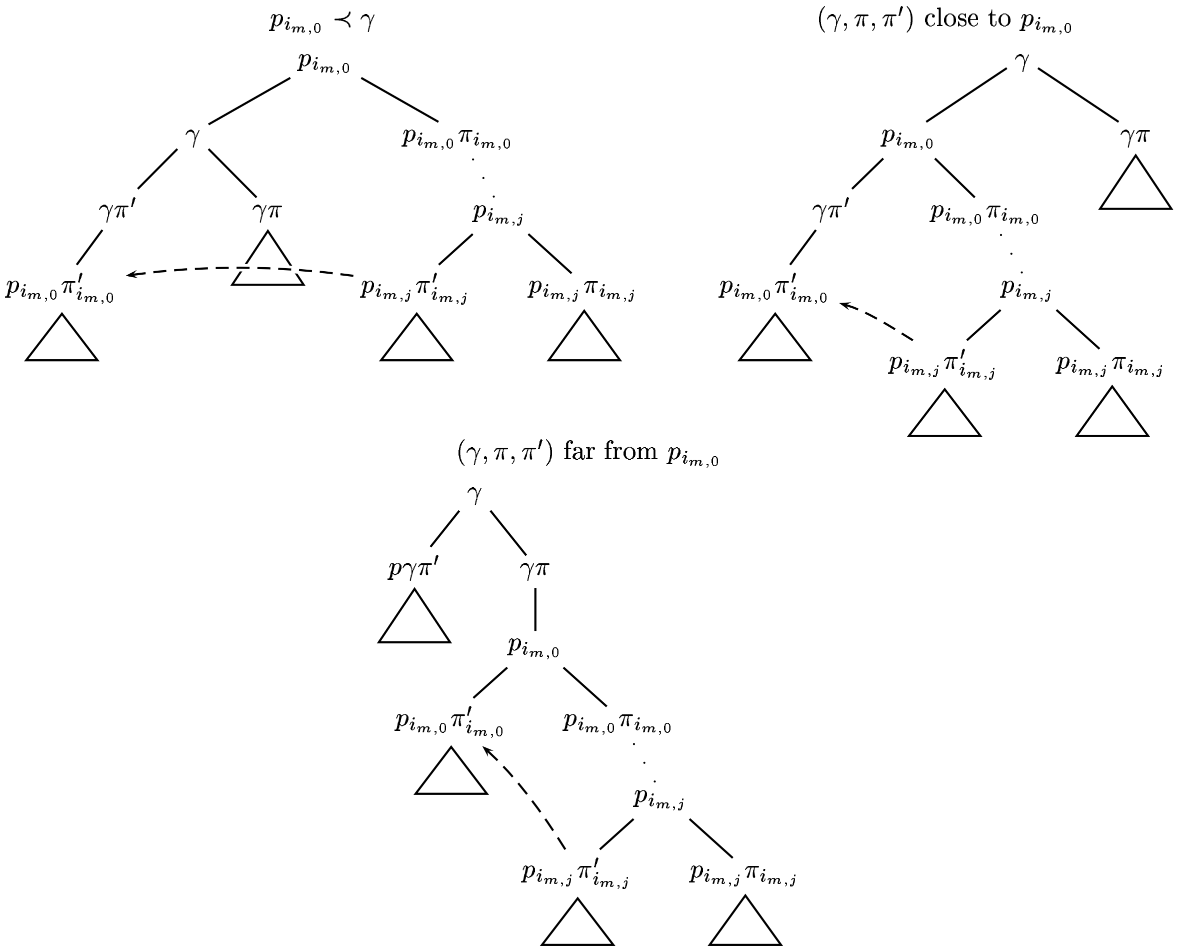

Figure 8: (γ, π, π') far from $p_{i_{m,0}}\pi'_{i_{m,0}}$.

2. $p_{k_1-k+1} \prec \ldots \prec p_{k_1} \prec p_{i_{m,0}}$,
3. the distance between $\prec p_{k_1}\ p_{i_{m,0}}$ is at least $d(\mathcal{A})+1$ ($|p_{i_{m,0}}| - |p_{k_1}| \leq d(\mathcal{A})+1$).

To summarize, we have k possible pumpings $\rho'_m \ll \rho$ and for each of them we have k other pumpings $\rho''_{m,i} \ll \rho'_m$ ($i \leq k$) such that every equality in a $\rho''_{m,i}$ is far from some position $p_{i_{m,0}} \succ p_{k_1-k+1} \succ p$ ($p_{k_1-k+1} \succ p$ is independent from m and i). Let p' be the largest of the above $p_{i_{m,0}}$. With the remark that all of these pumping are only replacement at some positions bigger than p', we proved $\mathcal{M}(\rho, p', k^2)$ thus $\mathcal{M}(\rho, p', h(\mathcal{A}, k))$, by lemma 23.

(end of proof of lemma 25) □

We initiate the process with lemma 19 and use the propagation lemma 25 to push the position under which no equality is created, up to the root of the tree. With simple sufficient conditions for the inequality $k^2 \geq h(\mathcal{A}, k)$, this yields:

Lemma 26 *Let $\mathcal{A} = (Q, Q^{\mathrm{f}}, \Delta)$ be an ADC . There exists two constants γ and δ independent from $\mathcal{A}$ such that if $\mathcal{M}(\rho, p, \gamma \times |Q|^2 \times 2^{\delta . c(\mathcal{A})^2} \times d(\mathcal{A})^2 \times n(\mathcal{A})^2)$ is true for some position p of a run ρ of $\mathcal{A}$ then there is a run of $\mathcal{A}$ $\rho' \ll \rho$ such that $\rho(\Lambda)$ and $\rho'(\Lambda)$ have the same target state.*

Proof. We shall use lemma 25. Hence, we need an integer k such that:

$$k^2 \quad \geq \quad h(\mathcal{A}, k) \tag{12}$$

For sake of simplicity, we write c, n and d respectively for $c(\mathcal{A})$, $n(\mathcal{A})$ and $d(\mathcal{A})$. We assume that $d, n \geq 1$.

$$\begin{array}{rcl} h(\mathcal{A}, k) & = & (d+1).n \times \big[k + g(\mathcal{A}, k+2dn)\big] \\ & = & (d+1).n \times \big[k + |Q|.2^c.c! \times (ek + 2edn + 1)\big] \\ & = & (d+1).n.(e.|Q|.2^c.c! + 1).k + (d+1).n.|Q|.2^c.c! \times (2edn+1) \\ & \leq & \alpha.|Q|.2^{\beta . c^2}.dn.k + \alpha.|Q|.2^{\beta . c^2}.d^2.n^2 \quad \text{where } \alpha \text{ and } \beta \text{ are constants independent of } \mathcal{A} \\ & \leq & m.k + m.dn \quad \text{where } m := \alpha.|Q|.2^{\beta . c^2}.dn \end{array}$$

One can check that for (12), it is sufficient to have $k = m + dn$. Therefore,

$$\begin{array}{rcl} h(\mathcal{A}, k) = h(\mathcal{A}, m+dn) & = & m^2 + 2mdn \\ & = & \alpha^2.|Q|^2.2^{2\beta . c^2}.d^2n^2 + 2\alpha.|Q|.2^{\beta . c^2}.d^2n^2 \\ & \leq & \gamma|Q|^2.2^{\delta . c^2}.d^2n^2 \quad \text{where } \gamma = 2\alpha^2 \text{ and } \delta = 2\beta \end{array}$$

The rest follows by induction on the depth of ρ, using the propagation lemma 25 and the lemmas 23 and 24. □

Moreover, note that $c(\mathcal{A}) \leq d(\mathcal{A}) \times n(\mathcal{A})$ and that both $d(\mathcal{A})$ and $n(\mathcal{A})$ are smaller than the size of constraints of $\mathcal{A}$. Thus, to summarize, the sufficient condition given in lemma 26 is roughly:

$$\mathcal{M}\big(\rho, p, \gamma \times |Q|^3 \times 2^{(\delta+1) \times (\text{size of constraints of } \mathcal{A})^2}\big)$$

5 Emptiness Decision for ADC

In this section we present the following result:

Theorem 27 *There is an algorithm which decides the emptiness of an ADC $\mathcal{A} = (Q, Q^{\mathrm{f}}, \Delta)$ and which runs in time $\mathrm{O}\Big(P_1(|Q|, |\Delta|) \times 2^{P_2(c(\mathcal{A}))}\Big)$ where P_1, P_2 are polynomials.*

5.1 Algorithm

We use a marking algorithm in which each state is marked with some successful runs yielding the state. This generalizes the usual marking algorithm for finite bottom-up tree automata: we do not keep only the information that a state is inhabited but also keep witnesses of this fact. The witnesses are used to check the disequality constraints higher up in the run.

To choose the witnesses runs which mark the states and ensure the termination of the algorithm, we use a sufficient condition for the above $\mathcal{M}$ predicate.

Definition 28 *$\mathcal{M}'$ is the predicate (defined relatively to an ADC $\mathcal{A}$ and an ordering $\gg$) which holds true on a run ρ of $\mathcal{A}$, a position p and an integer k iff p is a position of ρ and there exist k runs $\rho|_p \gg \rho_k \gg \ldots \gg \rho_1$ such that $\rho(p), \rho_1(\Lambda), \ldots, \rho_k(\Lambda)$ have the same target state, for all $1 \leq i \leq k$, for all $p' \in Pos(\rho_i)$, $\neg\mathcal{M}'(\rho_i, p', k)$ holds and for all $1 \leq i \leq k$ the pumping $\rho[\rho_i]_p$ does not contain any close equality.*

Note that this definition is well founded since the ordering $\gg$ is. The following lemma is an obvious consequence of definitions 22 and 28.

Lemma 29 *For every run ρ, every position $p \in Pos(\rho)$ and every integer k, $\mathcal{M}'(\rho_i, p, k)$ implies $\mathcal{M}(\rho, p, k)$.*

$\neg\mathcal{M}'$ can be seen as a *minimality predicate*: $\neg\mathcal{M}'(\rho, p, k)$ means that it is not possible to find k replacements below p with smaller runs, without creating equalities. Hence, the marking algorithm can be simply stated as follows, using a bound evaluated in the proof of lemma 26:

$$b(\mathcal{A}) := \gamma \times |Q| \times 2^{\delta.c(\mathcal{A})} \times d(\mathcal{A}) \times n(\mathcal{A}) \text{ for some fixed natural numbers } \gamma, \delta$$

Emptiness decision algorithm.

Start with a mapping which associates each state q with an empty set E_q^0
then saturate the states E_q^0 using the rule: $\dfrac{\{\rho_1, \ldots, \rho_n\} \in \bigcup_{i=0}^{n} \bigcup_{q \in Q} E_q^i}{r(\rho_1, \ldots, \rho_n) \in E_{q_0}^{n+1}}$
under the conditions:
1. $r(\rho_1, \ldots, \rho_n)$ is a run,
2. the target state of r is q_0,
3. $\neg\mathcal{M}'(r(\rho_1, \ldots, \rho_n), p, b(\mathcal{A}))$ for every p which is a prefix of some position checked by the constraint of r.

By lemmas 26 and 29 the condition in the emptiness decision algorithm is indeed a minimality condition.

We have to prove on one hand that the saturated set $E^* := \bigcup_{n \geq 0} \bigcup_{q \in Q} E_q^n$ contains an accepting run iff $\mathcal{A}$ accepts at least one tree (correctness, completeness) and on the other hand that E^* is computed with the expected complexity.

Lemma 30 (Correctness) *If E^* contains an accepting run then $L(\mathcal{A})$ is not empty.*

Proof. Immediate by the condition "$r(\rho_1, \ldots, \rho_n)$ is a run" in the emptiness decision algorithm. □

Lemma 31 (Completeness) *If E^* does not contain any accepting run then $L(\mathcal{A})$ is empty.*

Proof. Assume that $\mathcal{A}$ accepts a ground term t. We may moreover assume that the successful run $\rho = r(\rho_1, \ldots, \rho_n)$ of $\mathcal{A}$ on t is minimal w.r.t. $\gg$. Then, according to lemma 26, $\mathcal{M}(\rho, p, b(\mathcal{A}))$ is not true for any position p of ρ. Thus, by lemma 29, $\mathcal{M}'(\rho, p, b(\mathcal{A}))$ is also not true for any position p of ρ. By construction, this means that $\rho \in E^*$ whenever $\rho_1, ..., \rho_n \in E^*$. We conclude by an induction on the number of states:

- If there is only one state, then, by minimality, $n = 0$.
- If there is more than one state, consider the automata $\mathcal{A}_i$ which is a copy of $\mathcal{A}$ without the final states of $\mathcal{A}$ (and the corresponding rules) and whose final state is the target of ρ_i. By induction hypothesis, $\rho_i \in E^*$, hence $\rho \in E^*$. □

5.2 Bound on the Number of Steps

In order to give a complexity bound, we need an additional argument (a generalization of König's theorem for bipartite graphs to hypergraphs). Let us first define a notion of dependency in hypergraphs:

Definition 32 *Let S be a set and n, k be integers. The n-uples $\overline{s_1}, \ldots, \overline{s_k}$ of elements in S are* independent *iff there is a set $I \subseteq \{1, \ldots, n\}$ such that*

- $\forall i \in I, s_{1,i} = \ldots = s_{k,i}$
- $\forall i \notin I, \forall j \neq j', s_{j,i} \neq s_{j',i}$

Let $\mathcal{A}$ be an ADC and $\{\pi_1, \ldots, \pi_{c(\mathcal{A})}\}$ be the set of suffixes of positions which are checked by some rule of $\mathcal{A}$. Let ρ be a run of $\mathcal{A}$. Then Check(ρ) is the tuple $(t_1, \ldots, t_{c(\mathcal{A})}) \in (\mathcal{T}(\mathcal{F}) \cup \{\bot\})^{c(\mathcal{A})}$ such that $\rho_i = \bot$ if π_i is not checked by ρ and t_i is the term of $\mathcal{T}(\mathcal{F})$ associated to $\rho|_{\pi_i}$ otherwise.

Lemma 33 *Let ρ be a run of the ADC $\mathcal{A}$ and $p \in Pos(\rho)$, $k > b(\mathcal{A})$ and $\gg$ be a total ordering. If there are $k+1$ runs $\rho|_p = \rho_{k+1} \gg \ldots \gg \rho_1$ such that $\rho_{k+1}(\Lambda), \ldots, \rho_1(\Lambda)$ have the same target state and for all $1 \le i \le k$, for all $p' \in Pos(\rho_i)$, $\neg\mathcal{M}'(\rho_i, p', k)$ holds and* Check(ρ_1),..., Check(ρ_{k+1}) *are independent, then $\mathcal{M}'(\rho, p, k - c(\mathcal{A}))$ is true.*

Note that for this lemma this is enough that $\gg$ is total.
Proof. Assume that $\mathcal{M}'(\rho, p, k-c(\mathcal{A}))$ is false. It means that for at least $k' = k-c(\mathcal{A})+1$ runs among $\rho_1 \ldots \rho_{k+1}$, such that for all $1 \le i \le k'$, the pumping $\rho[\rho'_i]_p$ has a close equality. Lets call these terms $\rho'_1 \ldots \rho'_{k'}$ and $t'_1 \ldots t'_{k'}$ the associated terms in $\mathcal{T}(\mathcal{F})$. Let $(p_1, \pi_1, \pi'_1), \ldots, (p_{k'}, \pi_{k'}, \pi'_{k'})$ be close equalities of resp. $\rho[\rho'_1]_p \ldots \rho[\rho'_{k'}]_p$.

By construction, for each $1 \le i \le k'$, we have $p_i\pi_i \not\geq p$ or $p_i\pi'_i \not\geq p$. By symmetry, we may assume that $p_i\pi_i \not\geq p$ for all $1 \le i \le k'$ and we define:

$$\beta_i := p_i\pi_i/p \tag{13}$$

By definition of close equalities, we have:

1. for all $1 \le i \le k'$, $t|_{p_i\pi_i} \neq t|_{p_i\pi'_i}$, and $t'_i|_{\beta_i} = t|_{p_i\pi'_i}$, where t is the term of $\mathcal{T}(\mathcal{F})$ associated to ρ. Thus by transitivity, $t'_i|_{\beta_i} = t|_{p_i\pi_i} = t''|_{\beta_i}$, where $t'' \in \mathcal{T}(\mathcal{F})$ is associated to ρ_{k+1}.

2. by a pigeon hole principle, there exists $i, j_1, j_2 \le k'$ such that $\rho_{j_1}|_{\beta_i} = \rho_{j_2}|_{\beta_i}$.

Together, 1 and 2 contradict the independence of Check(ρ_1),..., Check(ρ_{k+1}). □
Now, we have the analog of König's theorem:

Theorem 34 (B. Reed, private communication) *Let S be a set and K, n be integers. Let $G \subseteq S^n$. If every subset $G_1 \subseteq G$ of independents elements has a cardinal $|G_1| \le K$, then $|G| \le K^n \times n!$*

Proof. We prove the result by induction on n.
For $n = 1$, G itself is a set of independent elements, hence $|G| \le K$.
Assume now that the property holds for $n-1$. Consider the graph $H(G)$ whose vertices are the elements of G and such that there is an edge (g, g') iff there is a component $i \in [1..n]$ such that $g_i = g'_i$. Any stable subset G_1 of G is independent, hence $|G_1| \le K$.

Now, let V be the maximal number of edges sharing some vertex v_0 in $H(G)$ (maximal neighborhood). We construct a stable set G_1 whose cardinal is $|G_1| \ge \frac{|G|}{V+1}$ as follows:

Initially, $G_2 = G$ and $G_1 = \emptyset$.

Repeat the following:

Choose a vertex v in G_2 and put it in G_1

remove v from G_2 as well as all vertices w such that $(v, w) \in H(G)$

until G_2 is empty.

Since, at each step, we remove at most $V+1$ elements from G_2, we have at most $\frac{|G|}{V+1}$ steps, and, at each step, we add an element in G_1. Hence $|G_1| \ge \frac{|G|}{V+1}$. Moreover, G_1 is stable, by construction.

Now, $K \ge |G_1| \ge \frac{|G|}{V+1}$, hence $V \ge \frac{|G|}{K} - 1$. Assume, by contradiction, that $|G| > K^n \times n!$, then $V > \frac{K^n \times n!}{K} - 1 = K^{n-1} \times n! - 1$. Now, there are at least $K^{n-1} \times n!$ edges departing from v_0 in $H(G)$. Hence there is an index $j \in [1..n]$ such that, for at least $K^{n-1} \times (n-1)!$ vertices v in $H(G)$, $v_0^j = v^j$. Let $G' = \{(v_1, ..., v_{j-1}, v_{j+1}, \ldots, v_n) | (v_1, \ldots, v_{j-1}, v_0^j, v_{j+1}, \ldots, v_n) \in G\}$ every subset G'_1 of G' of independent elements has a cardinal smaller or equal to K and $|G'| \ge K^{n-1} \times (n-1)! + 1$, which contradicts the induction hypothesis.
□
We can remark that the run generated by the emptiness decision algorithm are such that if $n > n'$ and $\rho \in E_q^n$, $\rho' \in E_q^{n'}$, then $d(\rho) > d(\rho')$.
Thus with this theorem 34 and the additional requirement that $\gg$ is such that:

$$\rho \gg \rho' \text{ if } d(\rho) > d(\rho') \tag{14}$$

where $d(\rho)$ is the depth of ρ, and with the contraposition of lemma 33, we can give an upper bound to the cardinal of sets of runs E_q^n constructed by the emptiness decision algorithm (for each $n \geq 0$, $q \in Q$):

$$\bigcup_{i=0}^{n} \bigcup_{q \in Q} E_q^i \leq |Q| \times (b(\mathcal{A}) + c(\mathcal{A}))^{c(\mathcal{A})} \times c(\mathcal{A})!$$

Which proves its termination with a number of inference steps in $\mathrm{O}\left(|Q|^{3 \times c(\mathcal{A})} \times 2^{P_0(c(\mathcal{A}))}\right)$ where P_0 is a polynomial.

5.3 Cost of one inference step.

Now, we have to estimate the cost of each inference step. The choice of one transition rule r in the algorithm is done among the set Δ, thus this (deterministic) choice is performed in time at most $|\Delta|$. Of course, we assume that identical subterms are shared. More precisely, each new run is a rule which sons are already in the set. Hence checking that $r(\rho_1, \ldots, \rho_n)$ is a run can be performed in time $c(\mathcal{A})$ since disequalities are checked in constant time.

Finally, estimate the cost of checking the predicate $\mathcal{M}'$. By the condition 14 above on the ordering $\gg$, only the runs which are already in $\bigcup_{n \geq 0} E_q^n$ are possible candidates for $\rho_1, \ldots, \rho_k$ in the definition 28 of $\mathcal{M}'$. For the test during step $n+1$, there are at most $\sum_{i=0}^{n} |E_q^i|$ possible runs to consider and $c(\mathcal{A})$ positions p. Finally, for each pumping, verifying whether or not it creates a close equality requires time $c(\mathcal{A})$. Hence minimality condition can be checked in time at most:

$$|Q| \times \left(b(\mathcal{A}) + c(\mathcal{A})\right)^{c(\mathcal{A})} \times c(\mathcal{A})! \times c(\mathcal{A})^2$$

Together with the bound on the number of steps, we get the desired property.

5.4 Ordering

It still remains to exhibit an ordering $\gg$ which satisfies all our requirements:

Lemma 35 *There is an ordering $\gg$ which is monotonic, well-founded, total on $T(\Delta)$ and such that, if $d(\rho) > d(\rho')$ then $\rho \gg \rho'$.*

Proof. Consider the following interpretation of a term t: $I(t)$ is the triple $(d(t), M(t), t)$ where $M(t)$ is the multiset of strict subterms of t.
Triples are ordered with the lexicographic composition of:

1. the ordering on natural numbers,
2. the multiset extension of $\gg$,
3. a lexicographic path ordering extending a total precedence.

$\gg$ itself is defined as $u \gg t$ iff $I(u) > I(t)$.

First, we should explain why the definition of the ordering itself is well-founded: $\gg$ is defined recursively, using its multiset extension. However, while defining $\gg$ on t, we use the multiset extension of $\gg$ on strict subterms of t and the subterm ordering is well-founded.

If $d(\rho) > d(\rho')$, then $\rho \gg \rho'$, simply because $d(\rho)$ is the first component of $I(\rho)$.

$\gg$ is monotonic. Assume $\rho_1 \gg \rho_2$ *i.e.* $I(\rho_1) > I(\rho_2)$. Then $d(\rho_1) \geq d(\rho_2)$ by definition of the lexicographic composition. Next,

$$d(\delta(t_1, \ldots, t_i, \rho_j, t_{i+1}, \ldots, t_n)) = \begin{cases} d(\rho_j) + 1 & \text{If } d(\rho_j) \geq \max(d(t_k)) \\ 1 + d(t_k) > d(\rho_j) & \text{for some } k \text{, otherwise} \end{cases}$$

In any case, $d(\delta(t_1, \ldots, t_i, \rho_1, t_{i+1}, \ldots, t_n)) \geq d(\delta(t_1, \ldots, t_i, \rho_2, t_{i+1}, \ldots, t_n))$.

$I(\rho_1) > I(\rho_2)$ implies either $d(\rho_1) > d(\rho_2)$ or $M(\rho_1) \geq M(\rho_2)$. Moreover,

$$M(\delta(t_1, \ldots, t_i, \rho_j, t_{i+1}, \ldots, t_n)) = \bigcup_{i=1}^{n} M(t_i) \cup \{\{t_1, \ldots, t_n\}\} \cup \{\{\rho_j\}\} \cup M(\rho_j)$$

If $d(\rho_1) > d(\rho_2)$, then there is a strict subterm ρ_3 of ρ_1 such that, for every strict subterm ρ_4 of ρ_2, $d(\rho_3) > d(\rho_4)$, which implies $\rho_3 \gg \rho_4$. Then, by definition of the multiset ordering, $\{\{\rho_3\}\} > M(\rho_2)$ and hence $M(\rho_1) > M(\rho_2)$. It follows that, in any case,

$$\rho_1 \gg \rho_2 \Rightarrow M(\rho_1) \geq M(\rho_2)$$

then
$$\begin{array}{rcl} M(\delta(\ldots, \rho_1, \ldots)) & \geq & \bigcup_{i=1}^{n} M(t_i) \cup M(\rho_2) \cup \{\{t_1, \ldots, t_n\}\} \cup \{\{\rho_1\}\} \\ & > & \bigcup_{i=1}^{n} M(t_i) \cup M(\rho_2) \cup \{\{t_1, \ldots, t_n\}\} \cup \{\{\rho_2\}\} \\ & = & M(\delta(\ldots, \rho_2, \ldots)) \end{array}$$

This suffices to guarantee the monotonicity.

$\gg$ is well-founded. By structural induction on t, there is no infinite strictly decreasing sequence starting with t.
If t is a constant, then $I(t) = (1, \emptyset, t)$ and t is minimal.
If this is true for the strict subterms of t, then let $\mathcal{E}$ be the set of terms smaller (w.r.t. $\gg$) than some strict subterm of t. $\gg$ is well-founded on $\mathcal{E}$ by induction hypothesis. Then its multiset extension is well-founded on multisets whose elements are in $\mathcal{E}$.

Then there is no infinite strictly decreasing sequence starting with t by well-foundedness of the lexicographic path ordering and since the lexicographic combination of well-founded orderings is itself well-founded.

$\gg$ is total. That is the purpose of the last component of the interpretation: the lexicographic path ordering extending a total precedence is total on ground terms. Hence, for any distinct terms ρ_1, ρ_2, either $I(\rho_1) \gg I(\rho_2)$ or $I(\rho_2) \gg I(\rho_1)$. $\square$

As a consequence of theorem 27 and proposition 11, the decision of ground reducibility is in DEXPTIME.

Theorem 36 *Ground reducibility of a term t w.r.t. a rewrite system $\mathcal{R}$ can be decided in deterministic time $O(2^{P(\|t\|,\|\mathcal{R}\|)})$ where P is a polynomial.*

6 Lower Bound

Theorem 37 *Ground reducibility is EXPTIME-hard, for linear rewrite systems $\mathcal{R}$ and linear terms t, with PTIME reductions.*

The proof is a reduction of the emptiness problem for the intersection of (languages recognized by) k tree automata.The latter is know to be EXPTIME-complete ([7], [14]).
We encode several (parallel) computations (runs) of k given tree automata on the same ground term $t \in \mathcal{T}(\mathcal{F})$ as a term of $s \in \mathcal{T}(\mathcal{F}')$ where $\mathcal{F}'$ is a new alphabet built from $\mathcal{F}$ and the tree automata. This encoding is polynomial. Then, we build a rewrite system $\mathcal{R}$ whose every ground normal form as the form $g(s)$ where g is a new function symbol and $s \in \mathcal{T}(\mathcal{F}')$ represents successful runs of the k automata $\mathcal{A}_1 \ldots \mathcal{A}_k$.
Thus, $L(\mathcal{A}_1 \cap \ldots \cap \mathcal{A}_k)$ is not empty iff the term $g(x)$ ($x \in \mathcal{X}$) is not ground reducible w.r.t. $\mathcal{R}$. Finally, we can conclude the proof of theorem 37 by checking that the system $\mathcal{R}$ is built in polynomial time w.r.t. the size of the size of the tree automata.

6.1 An EXPTIME-complete Problem

The formal definition of the EXPTIME-hard problem we consider is the following:

Proposition 38 *T. Frühwirth et al [7], H. Seidl [14] The following problem is EXPTIME hard: "given k tree automata $\mathcal{A}_1, \ldots, \mathcal{A}_k$, is $L(\mathcal{A}_1) \cap \ldots \cap L(\mathcal{A}_k)$ empty ?"*

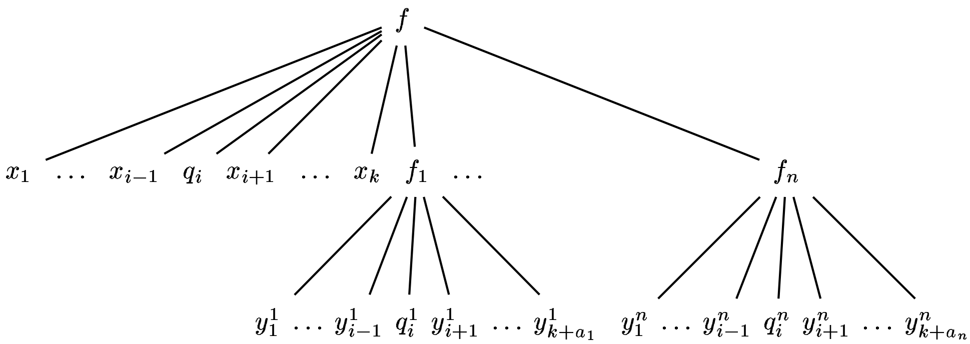

Figure 9: Rules of category 3 (left members).

6.2 Representation of Runs

Let $\mathcal{A}_1 = (Q_1, Q_1^{\mathrm{f}}, \Delta_1) \dots \mathcal{A}_k = (Q_k, Q_k^{\mathrm{f}}, \Delta_k)$. We can assume without loss of generality that the sets $Q_1 \dots Q_k$ are pairwise disjoint.

The alphabet $\mathcal{F}'$ is defined as follows:

- $\mathcal{F}' := \mathcal{F} \uplus \{g\} \uplus Q_1 \uplus \dots \uplus Q_k$,
- $g \notin \mathcal{F}$ and g is unary in $\mathcal{F}'$,
- The arity (in $\mathcal{F}'$) of each symbol of $Q_1 \uplus \dots \uplus Q_k$ is zero,
- The arity (in $\mathcal{F}'$) of each symbol of $f \in \mathcal{F}$ is the arity of f in $\mathcal{F}$ plus k.

We distinguish a subset $\mathcal{S} \subseteq \mathcal{T}(\mathcal{F}')$ which is recursively defined as follows:

- For each constant a in $\mathcal{F}$, each states $q_1 \in Q_1, \dots, q_k \in Q_k$, $a(q_1, \dots, q_k) \in \mathcal{S}$,
- For each symbol $f \in \mathcal{F}$, f having arity n in $\mathcal{F}$, each states $q_1 \in Q_1, \dots, q_k \in Q_k$, and each $t_1, \dots, t_n \in \mathcal{S}$, $f(q_1, \dots, q_k, t_1, \dots, t_n) \in \mathcal{S}$.

Note that the above $a(q_1, \dots, q_k)$ and $f(q_1, \dots, q_k, t_1, \dots, t_n)$ are indeed terms of $\mathcal{T}(\mathcal{F}')$. The terms of $\mathcal{S}$ will be used to represent parallel computations of $\mathcal{A}_1, \dots, \mathcal{A}_k$ on a term $t \in \mathcal{T}(\mathcal{F})$.

6.3 The Rewrite System $\mathcal{R}$

The system $\mathcal{R}$ is expected to reduce any ground term $t \in \mathcal{T}(\mathcal{F}')$ which is not of the form $t = g(s)$ where s represents successful runs of $\mathcal{A}_1, \dots, \mathcal{A}_n$ on a term $t \in \mathcal{T}(\mathcal{F})$. There can be four (mutually exclusive) reasons for that:

1. g occurs in t at a position which is not Λ,
2. $t = g(s)$ and s contains no g symbols ($s \in \mathcal{T}(\mathcal{F}' \setminus \{g\})$) but $s \notin \mathcal{S}$,
3. $t = g(s)$ and $s \in \mathcal{S}$ but s contains a transition which is not conform (this means, s does not code runs),
4. $t = g(s)$, $s \in \mathcal{S}$ and s codes n runs but at least one is not successful.

In the following, we enumerate the rules of $\mathcal{R}$ which reduce the ground terms falling in one of the categories. We are only interested in reducibility, which means that the right members of rules of $\mathcal{R}$ are irrelevant for our purpose. Thus, every right member of rule of $\mathcal{R}$ will be one arbitrary constant $q \in \mathcal{F}'$.

1. In this category, we have the following rules: [*g cannot occur inside a term*]

   $f(x_1, \dots, x_{i-1}, g(x), x_{i+1}, \dots, x_{k+n}) \to q$ such that:

   - $f \in \mathcal{F}$ and f as arity n in $\mathcal{F}$,

- $x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_{k+n}$ are distinct variables of $\mathcal{X}$.

2. Every rule of the second category has one of the forms:

   (a) [*no state can occur after the first kth positions below an $f \in \mathcal{F}$*]
   $f(x_1, \ldots, x_{i-1}, q', x_{i+1}, \ldots, x_{k+n}) \to q$ such that:
   - $f \in \mathcal{F}$ and f as arity n in $\mathcal{F}$,
   - $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{k+n}$ are distinct variables of $\mathcal{X}$,
   - $i > k$,
   - $q' \in Q_1 \uplus \ldots \uplus Q_k$.

   (b) [*no symbol of the otiginal signature $\mathcal{F}$ can occur in the first kth positions*]
   $f(x_1, \ldots, x_{i-1}, f'(y_1, \ldots, y_{k+n'}), x_{i+1}, \ldots, x_{k+n}) \to q$ such that:
   - $f, f' \in \mathcal{F}$ and their respective arity are n and n' in $\mathcal{F}$,
   - $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{k+n}, y_1, \ldots, y_{k+n'}$ are distinct variables of $\mathcal{X}$,
   - $i \le k$.

   (c) [*at position i, one must have a state of Q_i*]
   $f(x_1, \ldots, x_{i-1}, q', x_{i+1}, \ldots, x_{k+n}) \to q$ such that:
   - $f \in \mathcal{F}$ and its arity in $\mathcal{F}$ is n,
   - $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{k+n}$ are distinct variables of $\mathcal{X}$,
   - $i \le k$,
   - $q' \in Q_1 \uplus \ldots \uplus Q_{i-1} \uplus Q_{i+1} \uplus \ldots \uplus Q_k$.

3. The rules for this category are (see also figure 9):

   [*the subterms must describe transitions of the automata*]

$$f\left(\begin{array}{ll} x_1 \ldots x_{i-1}, q_i, x_{i+1} \ldots x_k, & f_1(y_1^1 \ldots y_{i-1}^1, q_i^1, y_{i+1}^1 \ldots y_{k+a_1}^1), \\ & \vdots \\ & f_n(y_1^n \ldots y_{i-1}^n, q_i^n, y_{i+1}^n \ldots y_{k+a_n}^n) \end{array}\right) \to q$$

   such that:

   - $f, f_1, \ldots, f_n \in \mathcal{F}$ and their respective arities in $\mathcal{F}$ are n and $a_1, \ldots, a_n$,
   - $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k, y_1^1, \ldots, y_{i-1}^1, y_{i+1}^1, \ldots, y_{k+a_1}^1,$ $y_1^n, \ldots, y_{i-1}^n, y_{i+1}^n, \ldots, y_{k+a_n}^n$ are distinct variables of $\mathcal{X}$,
   - $i \le k$,
   - $q_i, q_i^1, \ldots, q_i^n \in Q_i$,
   - $f(q_i^1, \ldots, q_i^n) \to q_i \notin \Delta_i$.

4. Finally, in the last category, we have: [*at the top of the term, we want final states*]

   $g(f(x_1, \ldots, x_{i-1}, q', x_{i+1}, \ldots, x_{k+n})) \to q$ such that:

   - $f \in \mathcal{F}$ and f as arity n in $\mathcal{F}$,
   - $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{k+n}$ are distinct variables of $\mathcal{X}$,
   - $i \le k$,
   - $q' \in (Q_1 \setminus Q_1^{\mathrm{f}}) \uplus \ldots \uplus (Q_k \setminus Q_k^{\mathrm{f}})$.

Size of $\mathcal{R}$. First of all, note that the system $\mathcal{R}$ is linear.

Now, we need to evaluate its size. It will be expressed in term of k, of the number of states of the automata $\mathcal{A}_1, \ldots, \mathcal{A}_k$, of the initial number of function symbols $|\mathcal{F}|$ and finally of the maximal arity α of a function symbol in $\mathcal{F}$: $\alpha := \max\{\text{arity of } f \mid f \in \mathcal{F}\}$. The biggest rule of $\mathcal{R}$ belongs to the category 3 and its size is: $k+2\times\alpha+2$. The number of rules in each category is summarized below:

Cat.	Number of rules
1	$\lvert\mathcal{F}\rvert \times (k+\alpha)$
2	$\lvert\mathcal{F}\rvert \times \alpha \times \sum_{i=1}^{k} \lvert Q_i\rvert + \lvert\mathcal{F}\rvert \times k \times (\lvert\mathcal{F}\rvert + \alpha) + \lvert\mathcal{F}\rvert \times k \times \sum_{i=1}^{k} \lvert Q_i\rvert$
3	$\lvert\mathcal{F}\rvert \times k \times \sum_{i=1}^{k} \lvert Q_i\rvert \times (\lvert\mathcal{F}\rvert \times (k+\alpha) \times \sum_{i=1}^{k} \lvert Q_i\rvert)^{\alpha}$
4	$\lvert\mathcal{F}\rvert \times k \times \sum_{i=1}^{k} \lvert Q_i\rvert$

Thus, the size of $\mathcal{R}$ is polynomial in the (sum of) sizes of the given tree automata. On the other hand, it is clear that the construction of $\mathcal{R}$ does not require a time bigger than the size of this system. Altogether, this proves the theorem 37.

7 Conclusion

We proved that ground reducibility is EXPTIME-complete for both the linear and the non-linear case. This closes a pending question. However, we do not claim that this result in itself gives any hint on how to implement a ground reducibility test. As we have seen, it is not tractable in general. A possible way to implement these techniques as efficiently as possible was suggested in [1]. In the average, some algorithms may behave well. In any case, we claim that tree automata help both in theory and in practice.

8 Acknowledgments

The authors wish to thank Bruce Reed for his proof of the theorem 34.

References

[1] A.-C. Caron, H. Comon, J.-L. Coquidé, M. Dauchet, and F. Jacquemard. Pumping, cleaning and symbolic constraints solving. In *Proc. Int. Conference on Algorithms, Languages and Programming*, Lecture Notes in Computer Science, vol. 820, Jerusalem, July 1994. Springer-Verlag.

[2] A.-C. Caron, J.-L. Coquidé, and M. Dauchet. Encompassment properties and automata with constraints. In C. Kirchner, editor, *5th International Conference on Rewriting Techniques and Applications*, volume 690 of *Lecture Notes in Computer Science*, Montreal, Canada, June 1993. Springer-Verlag.

[3] H. Comon. An effective method for handling initial algebras. In *Proc. 1st Workshop on Algebraic and Logic Programming, Gaussig, LNCS 343*. Springer-Verlag, Nov. 1988.

[4] H. Comon and F. Jacquemard. Ground reducibility and automata with disequality constraints. In P. Enjalbert, editor, *Proc. 11th Symp. on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science, vol. 775, Caen, 1994. Springer-Verlag.

[5] M. Dauchet, A.-C. Caron, and J.-L. Coquidé. Automata for reduction properties solving. *JSCOMP: Journal of Symbolic Computation*, 20, 1995.

[6] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 243–309. North-Holland, 1990.

[7] T. Frühwirth, E. Shapiro, M. Y. Vardi, and E. Yardemi. Logic programs as types for logic programs. In *6th IEEE Symp. Logic In Computer Science*, pages 300–309, 1991.

[8] J. Gallier and R. Book. Reductions in tree replacement systems. *Theorical Computer Science*, 37:123–150, 1985.

[9] J.-P. Jouannaud and E. Kounalis. Automatic proofs by induction in theories without constructors. *Information and Computation*, 82(1), July 1989.

[10] D. Kapur, P. Narendran, D. Rosenkrantz, and H. Zhang. Sufficient completeness, ground reducibility and their complexity. *Acta Inf.*, 28:311–350, 1991.

[11] D. Kapur, P. Narendran, and H. Zhang. On sufficient completeness and related properties of term rewriting systems. *Acta Inf.*, 24(4):395–415, 1987.

[12] E. Kounalis. Testing for the ground (co)-reducibility in term rewriting systems. *Theoretical Computer Science*, 106(1):87–117, 1992.

[13] D. Plaisted. Semantic confluence tests and completion methods. *Information and Control*, 65:182–215, 1985.

[14] H. Seidl. Haskell overloading is dexptime-complete. *Inf. Process. Letters*, 52:57–60, 1994.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399