

# First-Order Constraint Systems With Multiple Congruence Relations

Florent Jacquemard<sup>1</sup>, Étienne Lozes<sup>1</sup>, Ralf Treinen<sup>2</sup>, and Jules Villard<sup>1</sup>

<sup>1</sup> LSV, ENS Cachan, CNRS UMR 8643 and INRIA, France

<sup>2</sup> PPS, Université Paris Diderot, CNRS UMR 7126, France

**Abstract.** We investigate the problem of deciding first-order theories of finite trees with several distinguished congruence relations, each of them given by some equational axioms. We give an automata-based solution for the case where the different equational axiom systems are linear and variable-disjoint (this includes the case where all axioms are ground), and where the logic does not permit to express tree relations  $x=f(y,z)$ . We show that the problem is undecidable when these restrictions are relaxed.

As motivation and application, we show how to translate the model checking problem of ApiL, a spatial equational logic for the applied pi-calculus, to the validity of first-order formulas in term algebras with multiple congruence relations.

## 1 Introduction

There exist a number of decidability results for the first-order theory of finite trees, or also called term algebras. The decidability of the first-order theory of finite trees over a finite signature, with syntactic equality as the only predicate, was first shown by Malcev [1], this result was later rediscovered and extended independently by Maher [2] and Comon and Lescanne [3]. Encouraged by this result, several researchers started in the late 80s the program to show decidability of the first-order theory of term algebras with different predicates than just syntactic equality. Research basically went into several directions: one direction was to add relations other than equality to the theory, in particular ordering relations that were useful for ordered rewrite calculi [4, 5], or for typing of programming languages [6, 7]. Another direction was the addition of predicates that can be recognized by various classes of tree automata [8, 9]. A third direction was to replace the syntactic equality relation in the original result by an equality relation modulo a set of equational axioms. The initial optimism was fueled by the fact that for quantifier-free positive constraints, so-called unification systems, the extension of syntactic unification to unification modulo equational theories has led to a rich theory and many useful results (see, for instance, [10] for a survey). The probably strongest result in this direction is the decidability of the theory of term algebras modulo so-called *shallow* equational theories [11]. However, it also turned out that the limits of decidability are met much earlier

with first-order theories than with unification problems, and undecidability of the theory of term algebras modulo some important equational theories were shown, among them AC [12, 13].

Most of these decidability results were obtained by quantifier elimination. The reason why automata techniques were not used here is that one typically wants to express relations like  $x = f(y, z)$ , which cannot be done using automata techniques. All results concerned structures with only one congruence relation, which hence could be seen as an equality in some quotient algebra. The technical reason for this limitation was that quantifier elimination procedures typically contain a rule for eliminating positive occurrences of equations, where one side is reduced to a variable, like this:

$$\frac{\exists x. (x = t \wedge \phi)}{\phi[x \leftarrow t]} \quad \text{if } x \notin \text{Vars}(t)$$

In this rule,  $\phi$  is an arbitrary conjunction of literals, and  $\phi[x \leftarrow t]$  denotes the formula obtained by replacing every occurrence of the variable  $x$  by the term  $t$ . The hypothesis and the conclusion of this rule are obviously logically equivalent when  $=$  is interpreted as syntactic equality, and more generally when  $=$  is interpreted as a congruence relation with respect to all predicates of the structure.

Recently, work on a spatial equational logic for the applied  $\pi$ -calculus [14] demanded the decidability of a first-order logic of a term-algebra with *multiple* congruence relations, each of them defined by a set of ground equational axioms. Is it possible to obtain the needed result by extending existing techniques? An investigation of typically used quantifier elimination procedures showed that this would be very difficult at best. The reason is that, faced with a formula like

$$\exists x_1, x_2. (x_1 =_1 t_1 \wedge x_2 =_2 t_2 \wedge \phi)$$

where  $=_1$  and  $=_2$  are two different congruence relations of our structure, one could not simply eliminate  $x_1$  or  $x_2$  as before. The problem is that in general,  $=_1$  would not be a congruence with respect to  $=_2$ , and vice versa, since in general the equational axioms used for defining these equivalence relations would be independent.

Looking again back at unification one might hope that there could be a solution to this problem. If  $\{\theta_1, \dots, \theta_n\}$  is a complete set of unifiers of  $x =_1 t_1$ , that is for the equational theory  $=_1$ , then the above formula would be equivalent to

$$\exists x_2, \bar{y}_1. (x_2 =_2 t_2 \wedge \phi)\theta_1 \vee \dots \vee \exists x_2, \bar{y}_2. (x_2 =_2 t_2 \wedge \phi)\theta_n$$

where  $\bar{y}_i$  is the set of extra variables introduced by the unifier  $\theta_i$ . This paper is about the question whether this can indeed be achieved for the full first-order theory.

**Contents of this paper:** In Section 3 we show the decidability of the first-order theory of term algebras with several congruence relations. The predicates of our structure are of the form  $x =_i y$ , where each  $=_i$  is given by a set of linear and variable-disjoint equational axioms. The structure does *not* contain

function symbols, and hence does not allow to express a relation like  $x = f(y, z)$ . This restriction makes the structure accessible to automata-theoretic techniques, which is a key to our decidability result. We will also show that decidability does no longer hold when we allow to express term relations like  $x = f(y, z)$ , or when one generalizes to flat axiom systems. Then, in Section 4, we show how to extend the decidability result to a certain class of “background” rewrite systems. An application to a spatial equational logic for the applied  $\pi$ -calculus is given in Section 5.

## 2 Preliminaries

We assume the usual notions of rewriting. A signature is called *monadic* if it contains only unary and constant function symbols. The set of variables is  $V$ ; given a signature  $\Sigma$ , we denote by  $T(\Sigma, V)$  the set of terms over  $\Sigma$ , and by  $T(\Sigma)$  the set of *ground* terms (terms without variables). A term  $t \in T(\Sigma, V)$  can be conveniently seen as a function from its set of positions  $Pos(t)$  (non-empty subset of  $\mathbb{N}^*$  that is closed under prefix and left brother) into  $\Sigma \cup V$ . Let  $Vars(t)$  denote the set of variables of  $t$ ,  $depth(t)$  its depth,  $t|_p$  the subterm of  $t$  at position  $p$ , and  $t[s]_p$  the replacement in  $t$  of the subterm at position  $p$  by  $s$ . The term  $t$  is called *linear* if every variable of  $Vars(t)$  occurs exactly once in  $t$ .

Equations are considered non-oriented, that is  $\ell = r$  is considered the same as  $r = \ell$ . We call an equation  $E = (\ell = r)$  *ground* when  $Vars(\ell) = Vars(r) = \emptyset$ , *variable-disjoint* when  $Vars(\ell)$  is disjoint with  $Vars(r)$ , *flat* when  $depth(\ell), depth(r) \leq 1$  and *shallow* when every variable of  $Vars(\ell) \cap Vars(r)$  occurs at depth at most 1 in  $\ell$  and  $r$ . A set of equations is variable-disjoint (resp. ground, flat, shallow) when each of its equations is. Any flat equation is shallow, and any ground equation is both shallow and variable-disjoint, while in general flat or shallow equations are not necessarily variable-disjoint.

Let  $R$  be a rewrite system, and  $E$  a set of equational axioms. We write  $s \rightarrow_R t$  when  $s$  rewrites to  $t$  in one step by  $R$ , and  $s \leftrightarrow_E t$  when  $s$  transforms to  $t$  in one equational proof step by  $E$ . The relations  $\xrightarrow{*}_R$  and  $=_E$  are the reflexive and transitive closures of respectively  $\rightarrow_R$  and  $\leftrightarrow_E$ , that is, in the latter case,  $s =_E t$  when  $s$  and  $t$  are equal modulo the set  $E$  of equations. We write  $=_{E,R}$  for the reflexive, symmetric and transitive closure of  $\leftrightarrow_E \cup \rightarrow_R$ .

Given a finite signature  $\Sigma$ , a (bottom-up) *tree automaton*  $A$  is given by  $(Q, F, \Delta)$  where

- $Q$  is a finite set of *states*.
- $F \subseteq Q$  is called the set of *accepting states*.
- $\Delta$  is a set of rewrite rules  $f(q_1, \dots, q_n) \rightarrow q$  with  $f \in \Sigma_n$ ,  $q_1, \dots, q_n, q \in Q$ .

The automaton  $A$  *accepts* a tree  $t$  iff  $t \xrightarrow{*} q \in F$  by the transition rules  $\Delta$ . The *language*  $L_A$  is the set of all trees accepted by  $A$ .

Tree automata enjoy (almost) all the nice properties of word automata, in particular closure under Boolean operations, decidability of the emptiness problem, determinization, minimization [15].

The *convolution* operation defined below allows to code  $n$ -tuples of trees as trees over a signature of  $n$ -tuples. Let  $\Sigma$  be a signature with  $\square \notin \Sigma$ . We define the signature  $\Sigma^{[n]}$ , for  $n \geq 1$ , as

$$\Sigma^{[n]} = \{[f_1, \dots, f_n] \mid f_i \in \Sigma \cup \{\square\}, f_i \neq \square \text{ for some } i\}$$

The arity of  $[f_1, \dots, f_n]$  in  $\Sigma^{[n]}$  is the maximum of the arities of those  $f_i$  that are in  $\Sigma$ .

For  $t_1, \dots, t_n \in T(\Sigma)$ , the convolution  $t_1 \otimes \dots \otimes t_n$  is the tree  $t \in T(\Sigma^{[n]})$  defined by  $Pos(t) = Pos(t_1) \cup \dots \cup Pos(t_n)$ , and for all  $\pi \in Pos(t)$ ,  $t(\pi) = [f_1, \dots, f_n]$  where  $f_i = t_i(\pi)$  if  $\pi \in Pos(t_i)$ , and  $f_i = \square$  otherwise. Projection is defined by  $\pi_i(t_1 \otimes \dots \otimes t_n) = t_i$ .

For example, let  $\Sigma = \{h, f, a\}$ , where  $a$  is a constant,  $f$  unary, and  $h$  binary. Then we have that  $f(a) \otimes h(a, f(a)) = [f, h]([a, a], [\square, f]([\square, a]))$ .

Now, one can define *tree-automatic representations* and *tree-automatic structures* analogously to the definition given in [16] for automata over finite words. This definition applies only to so-called *relational* structures, that is structures that have only predicates in their logical language and no constants or function symbols. This is not a restriction as constants or functions can always be expressed by predicates.

Let  $A$  be a structure over a relational signature with relation symbols  $R_1, \dots, R_n$ . A *tree-automatic representation* of  $A$  is given by

1. a finite signature  $\Sigma$ ,
2. a recognizable tree language  $L_\delta \subseteq T(\Sigma)$ ,
3. an onto function  $\nu: L_\delta \rightarrow A$ ,
4. a recognizable tree language  $L_R \subseteq T(\Sigma^{[n]})$  for each relation symbol  $R$  of the signature of  $A$ , such that for all  $x_1, \dots, x_n \in L_\delta$  :

$$x_1 \otimes \dots \otimes x_n \in L_R \text{ iff } (\nu(x_1), \dots, \nu(x_n)) \in R^A$$

In this case we say that the relation  $R^A$  is *recognizable*.

A structure is *tree-automatic* if it has a tree-automatic representation. The first-order theory of any tree-automatic structure is decidable.

Ground Tree Transducers (GTT) have been introduced in [17]. A GTT is defined by two tree automata  $A_1$  and  $A_2$  over the same signature, and possibly with shared states. The GTT defined by  $A_1$  and  $A_2$  recognizes the pair  $(t, t')$  iff there exists a context  $C$ , terms  $t_i, t'_i \in T(\Sigma)$ , and states  $q_i$  for  $1 \leq i \leq n$ , such that  $t = C[t_1, \dots, t_n]$ ,  $t' = C[t'_1, \dots, t'_n]$ ,  $t_i \xrightarrow{*} q_i$  by  $A_1$  and  $t'_i \xrightarrow{*} q_i$  by  $A_2$ . Any relation defined by a GTT is recognizable, and the set of GTT-definable relations is closed under iteration (Kleene star) [18].

### 3 The Case of Several Congruence Relations

**Definition 1.** Let  $\Sigma$  be a countable signature with an upper bound on the arities of the function symbols,  $(E_i)_{i \in I}$  be a (possibly infinite) family of sets of equations

over  $\Sigma$ , and  $(L_j)_{j \in J}$  a (possibly infinite) family of recognizable tree languages over the signature  $\Sigma$ . The first-order structure  $A(\Sigma, (E_i)_{i \in I}, (L_j)_{j \in J})$  is defined as follows:

- The universe is the set of all ground  $\Sigma$ -terms.
- There are no constant or function symbols.
- For every  $i \in I$  we have a binary relation  $=_i$ , interpreted as  $t_1 =_i t_2$  iff  $t_1 =_{E_i} t_2$ .
- For every  $j \in J$  we have a unary relation  $L_j$ , interpreted as  $L_j(t)$  iff  $t \in L_j$ .

The structure  $H(\Sigma, (E_i)_{i \in I}, (L_j)_{j \in J})$  contains in addition to  $A(\Sigma, (E_i)_{i \in I}, (L_j)_{j \in J})$  all symbols from  $\Sigma$  as function symbols, interpreted as free constructor symbols. In addition we assume that the empty set of equational axioms is part of  $(E_i)_{i \in I}$ .

Note that this definition allows to consider a structure in which every ground term  $t \in T(\Sigma)$  exists as a syntactic constant. This would be represented by having in the family of recognizable tree languages, for every  $t \in T(\Sigma)$ , the language consisting of the single term  $t$  only (each such language is of course recognizable). Also, note that the logical language of  $H(\Sigma, (E_i)_{i \in I}, (L_j)_{j \in J})$  allows to express unification problems like  $x = f(y, z)$ ; however this is not possible in  $A(\Sigma, (E_i)_{i \in I}, (L_j)_{j \in J})$ .

We will show that  $A(\Sigma, (E_i)_{i \in I}, (L_j)_{j \in J})$  is an automatic structure in case all equation systems are variable-disjoint. The first step is to define the encoding of the algebra as trees over a finite signature, the (minor) difficulty here being that the algebra contains trees over a possibly infinite alphabet but with a bounded arity. The details of this are given in Appendix A.

The languages  $L_j$ ,  $j \in J$ , are recognizable by definition. In order to show that every  $=_i$ ,  $i \in I$ , is recognizable we construct a Ground Tree Transducer as follows:

Given the ground equational theory  $E = \{s_1 = t_1, \dots, s_n = t_n\}$ , let  $A_1$  be the tree automaton that recognizes the set of instances of  $s_i$  in state  $q_i$ , for any  $i$ , and the set of instances of  $t_i$  in state  $p_i$ , for any  $i$ . Symmetrically, let  $A_2$  be the tree automaton that recognizes the set of instances of  $s_i$  in state  $p_i$ , for any  $i$ , and the set of instances of  $t_i$  in state  $q_i$ , for any  $i$ . These automata can be constructed exactly because each equational axiom is linear. Since the axioms are variable-disjoint, the GTT defined by  $A_1$  and  $A_2$  recognizes a pair of terms  $(t, t')$  iff  $t$  is obtained from  $t'$  by a parallel equational replacement with respect to  $E$ . The transitive closure of this relation is exactly the equality relation modulo  $E$ , which is again a GTT [18], and hence recognizable. Hence:

**Theorem 1.** *For any arity-bounded countable signature  $\Sigma$ , family  $(E_i)_{i \in I}$  of sets of linear variable-disjoint equations over  $\Sigma$ , and family  $(L_j)_{j \in J}$  of recognizable tree languages over the signature  $\Sigma$ , the first-order theory of  $A(\Sigma, (E_i)_{i \in I}, (L_j)_{j \in J})$  is decidable.*

The following theorem states that Theorem 1 no longer holds if generalized to the structure  $H(\Sigma, (E_i)_{i \in I}, (L_j)_{j \in J})$ , that is if one also allows relations like  $x = f(y, z)$ .

**Theorem 2.** *The problem of deciding the first-order theory of  $H(\Sigma, (E_i)_{1 \leq i \leq 3}, (L_j)_{1 \leq j \leq 3})$  is undecidable even when  $\Sigma$  is finite,  $E_1$  and  $E_2$  are systems of ground equations, and  $E_3 = \emptyset$ .*

**Proof** (sketch, see Appendix B for details). We use a technique called *shifted pairing* [19] in order to encode the acceptance problem of the empty tape for Turing machines.

Let  $M$  be a Turing machine computing on a tape bounded on the left and unbounded on the right, with input alphabet  $\Gamma$  and state set  $S$ . We represent a configuration of  $M$  as a word  $c$  in  $\Gamma^*S\Gamma^*\flat^*$ , where the unique state symbol  $s \in S$  in  $c$  indicates the current position of the head of  $M$ . The blank symbol  $\flat$  is used for padding the right of the tape: for commodity, we consider finite computations of  $M$  such that successive configurations all have the same length (the length of a word  $c$  is denoted by  $|c|$ ).

We encode the configurations of  $M$  as right-combs built with a binary function symbol  $g$ , using a unary operator  ${}^t$  defined by  $c_i^t := g(c_{i,1}, \dots, g(c_{i,k}, \flat))$  for a configuration  $c_i = c_{i,1} \dots c_{i,k}$ . A computation  $c_0, \dots, c_n$  of  $M$  (sequence of configurations) is also encoded as a right comb  $f(c_0^t, \dots, f(c_n^t, \flat))$ , where  $f$  is another binary function symbol.

Let us define the following three recognizable languages

- $L_0$  which contains the term representations  $c_0^t$  of initial configurations  $c_0$  of  $M$  (*i.e.* an initial state followed by an arbitrarily long sequence of  $\flat$ ),
- $L_c$  which contains the term representations  $f(c_0^t, \dots, f(c_n^t, \flat))$  of sequences of configurations  $c_0, \dots, c_n$  of  $M$  (possibly not successive), such that  $c_n$  is a final configuration,
- $L_{sp}$  which contains terms  $f(c_0^t \otimes d_1^t, \dots, f(c_{n-1}^t \otimes d_n^t, f(c_n^t \otimes \flat, \flat)))$  such that for all  $i < n$ ,  $|c_i| = |d_{i+1}|$  and  $d_{i+1}$  is a successor of  $c_i$  using a transition of  $M$ .

Here,  $\otimes$  is a simplified version of the convolution product (see Section 2) defined recursively by  $g(a, s) \otimes g(b, t) = g([a, b], s \otimes t)$ ,  $\flat \otimes \flat = \flat$  and  $g(a, s) \otimes \flat = g([a, \square], s \otimes \flat)$ .

Then we propose then three sets of equational axioms

- $E_1$  which defines the left projection over the signatures of pairs, with equations of the form  $[a, b] = a$ ,
- $E_2$  which defines the right projection, with equations of the form  $[a, b] = b$ , and moreover will delete the last element  $c_n^t \otimes \flat$  of terms of  $L_{sp}$ , providing that the configuration  $c_n$  is final, with the equations  $g([b, \square], \flat) = \#$ ,  $g([b, \square], \#) = \#$  and  $f(g([s^f, \square], \#), \flat) = \flat$  for all final state  $s^f$
- $E_3$  which is empty.

Regarding the definition of  $E_2$ , note that we assume *wlog* that entering a final state terminates the computation of  $M$  and that before entering a final state,  $M$  deletes the whole tape (all the symbols of  $\Gamma$  are replaced by  $\flat$ ).

With these constructions, it holds that the first order formula

$$\phi := \exists y, y_1, y_2, x_0. L_{sp}(y) \wedge y =_{E_1} y_1 \wedge L_c(y_1) \wedge y =_{E_2} y_2 \wedge L_c(y_2) \wedge L_0(x_0) \wedge y_1 = f(x_0, y_2)$$

is satisfiable in  $H(\Sigma, (E_i)_{i \in \{1,2\}}, (L_{sp}, L_c, L_0))$  iff  $M$  admits a successful computation starting with a blank tape.

Note that we could get rid of the recognizable languages  $L_{sp}$  and  $L_c$  in the structure, and of the atoms  $L_{sp}(y)$  and  $L_c(x_1)$  in  $\phi$  by adding two ground equational theories which describe the bottom-up transitions of two tree automata recognizing respectively  $L_{sp}$  and  $L_c$ .  $\square$

The next result shows that Theorem 1 no longer holds when one replaces variable-disjoint equational systems by flat equational systems, even if one has only two different flat systems and unification equations  $x = f(y, z)$  are not allowed. Moreover, the signature considered below is monadic, hence the results holds already when considering a domain of words. Note that the first-order theory of *one* congruence relation defined by a shallow equational system (hence in particular, by one flat equational system), without unitary recognizable predicates but with unification equations  $x = f(y, z)$ , is decidable [11].

**Theorem 3.** *The first-order theory of  $A(\Sigma, (E_i)_{1 \leq i \leq 3}, (L_j)_{1 \leq j \leq 2})$  is undecidable when  $\Sigma$  is finite and monadic and  $E_1, E_2$  are two sets of flat equations over  $\Sigma$  and  $E_3 = \emptyset$ .*

**Proof** We reduce the Post correspondence problem (PCP). The principle of the reduction presented here follows an idea used in [20] for showing undecidability of another problem (termination of shallow term rewriting systems). Let us consider the following instance of PCP without empty words given by a finite set of pairs of words:

$$\mathcal{P} := \{(u_i, v_i) \mid u_i, v_i \in \{a, b\}^+, 1 \leq i \leq N\}$$

A solution of  $\mathcal{P}$  is a finite sequence  $(i_j)_{0 \leq j \leq k}$  with  $1 \leq i_0, \dots, i_k \leq N$ , such that  $u_{i_0} u_{i_1} \dots u_{i_k} = v_{i_0} v_{i_1} \dots v_{i_k}$ . The problem of the existence of a solution is undecidable [21]. For all  $1 \leq i \leq N$ , let  $u_i = u_{i,1} \dots u_{i,|u_i|}$  and  $v_i = v_{i,1} \dots v_{i,|v_i|}$ . Let  $L := \max(|u_i|, |v_i| \mid i \leq N)$ , and let us define the signature

$$\Sigma := \{a : 1, b : 1, b : 0\} \cup \{P_{i,j} : 1 \mid 1 \leq i \leq N, 1 \leq j \leq L\}.$$

For the sake of readability, we shall write the terms of  $T(\Sigma)$  as words of  $\Sigma_1^* \Sigma_0$ . For all  $1 \leq i \leq N$ , let  $P_i$  be the word  $P_{i,1} \dots P_{i,L}$ . Let us consider two tree automata:

- $L_\alpha$  recognizing  $\{a, b\}^+ b$ ,
- $L_P$  recognizing  $\{P_i \mid 1 \leq i \leq N\}^* b$ .

The purpose of the symbols  $P_{i,j}$  in the words  $P_i$  is to represent a “skeleton” of solution of  $\mathcal{P}$ , *i.e.* a sequence of indexes that will be replaced by letters of the  $u_i$ 's or  $v_i$ 's by the following two sets of flat equations

$$E_1 = \left\{ P_{i,j}(x) = u_{i,j}(x) \mid \left\{ \begin{array}{l} 1 \leq i \leq N, \\ 1 \leq j \leq |u_i| \end{array} \right\} \right\} \cup \left\{ P_{i,j}(x) = x \mid \left\{ \begin{array}{l} 1 \leq i \leq N, \\ |u_i| < j \leq L \end{array} \right\} \right\}$$

$$E_2 = \left\{ P_{i,j}(x) = v_{i,j}(x) \mid \left\{ \begin{array}{l} 1 \leq i \leq N, \\ 1 \leq j \leq |v_i| \end{array} \right\} \right\} \cup \left\{ P_{i,j}(x) = x \mid \left\{ \begin{array}{l} 1 \leq i \leq N, \\ |v_i| < j \leq L \end{array} \right\} \right\}$$

Let  $E_3 = \emptyset$ . Finally, let  $\phi$  be the following closed first formula over  $A(\Sigma, (E_i)_{1 \leq i \leq 3}, (L_\alpha, L_P))$

$$\phi := \exists x, u, v. L_P(x) \wedge x =_{E_1} u \wedge x =_{E_2} v \wedge L_\alpha(u) \wedge L_\alpha(v) \wedge u = v.$$

We show in Appendix C that  $\phi$  is satisfiable in  $A(\Sigma, (E_i)_{1 \leq i \leq 3}, (L_\alpha, L_P))$  iff  $\mathcal{P}$  has a solution.  $\square$

## 4 Adding a Background Term Rewrite System

In this section we show that Theorem 1 can be extended to the case where all equations are taken modulo an additional term rewrite system with some particular properties. The first property is that the system is *canonical*, that is normalizing and confluent, such that each term has a unique normal form. This allows us to restrict the universe of the logic structure to contain only terms in normal form, and each ground term would be interpreted in that structure as its normal form.

**Definition 2.** Let  $\Sigma$ ,  $(E_i)_{i \in I}$ , and  $(L_j)_{j \in J}$  be as in Definition 1, and  $R$  a canonical and left-linear term rewrite system. The first-order structure  $A(\Sigma, (E_i)_{i \in I}, (L_j)_{j \in J}, R)$  is defined as  $A(\Sigma, (E_i)_{i \in I}, (L_j)_{j \in J})$  in Definition 1, except that the universe is restricted to  $R$ -normal forms, and that  $t_1 =_i t_2$  is interpreted as  $t_1 =_{E_i, R} t_2$ .

Note that the term rewrite system may indeed intervene even when the structure contains only terms in normal form, and when all equational systems are normalized with respect to the rewrite system. Take, for example, a rewrite system  $R$  consisting of the rule

$$\text{left}(\text{pair}(x, y)) \rightarrow x$$

and the equational system  $E = \{c = \text{pair}(a, b)\}$ . The system  $E$  is normalized w.r.t.  $R$ , and so are the terms  $a$  and  $\text{left}(c)$ . However,  $\text{left}(c) = a$  is a consequence of  $E \cup R$  but not of  $E$  alone.

For the decidability result below we require in addition the rewrite system to be *orthogonal*, that is left-linear and without critical pairs. The set of terms in normal forms is then recognizable as a consequence of left-linearity [15]; absence of critical pairs will be useful in the proof of Theorem 4. Orthogonality implies confluence [22].

The idea is to “complete” any of the given equational systems w.r.t.  $R$ . If this process terminates for each single of these (probably infinitely many) systems then we can conclude. If  $l = r$  is an equational axiom and  $g \rightarrow d$  a rewrite rule then a *critical pair* is given in the following two cases:

- there is a substitution  $\sigma$  and a non-variable position  $p$  of  $g$  such that  $g\sigma \upharpoonright_p = l\sigma$ , in that case the critical pair is  $g\sigma[r\sigma]_p = d\sigma$ .
- there is a substitution  $\sigma$  and a non-variable position  $p$  of  $l\sigma$  such that  $g\sigma = l\sigma \upharpoonright_p$ , in that case the critical pair is  $r\sigma = l\sigma[d\sigma]_p$ .

In order to meet the hypotheses of Theorem 1, we have to assure that the critical pair is again linear and variable disjoint. Linearity may be violated only by a non-linearity of  $d$  (since all other terms are linear), and variable disjointness may be violated in the first case when  $g\sigma[\bullet]_p$  is not ground. We say that  $E'$  is the *completion* of  $E$  by  $R$  when  $E'$  is the smallest set containing  $E$  and that contains all its own critical pairs with  $R$ . If this set is finite then it can be calculated from  $E$  by successive addition of critical pairs.

**Lemma 1.** *If  $E'$  is the completion of  $E$  by  $R$  then  $s =_{E,R} t$  iff  $s =_{E'} t$  for all terms  $s, t$  in  $R$ -normal form.*

**Proof** Any  $E'$  proof step can be simulated by several  $E, R$  proof steps, so the back direction is obvious. For the other direction, first note that when  $s =_{E,R} t$  then  $s =_{E',R} t$  since  $E \subseteq E'$ . Any proof of  $s =_{E',R} t$  can be transformed into a proof such that any  $R$ -rewrite step is either preceded by an  $R$  rewrite-step, or by an  $E$ -step such that the redex of the rewrite step has a non-trivial overlap with the previous equational step. This is a consequence of the orthogonality of  $R$  and the fact that  $s$  and  $t$  are in  $R$ -normal form, since a rewrite step can be commuted with a non-overlapping equational step. If the shortest such proof used an  $R$ -step then we could replace the preceding equational step and that rewrite step by one single equational step (their critical pair), which would yield a contradiction.  $\square$

Hence, we obtain together with Theorem 1:

**Theorem 4.** *If  $R$  is orthogonal and terminating, and if every  $E_i$  has a finite completion w.r.t.  $R$  that is linear and variable-disjoint, then the first-order theory of the structure  $A(\Sigma, (E_i)_{i \in I}, (L_j)_{j \in J}, R)$  is decidable.*

*Example.* Let  $R$  be the following term rewrite system:

$$\text{left}(\text{pair}(x, y)) \rightarrow x \qquad \text{right}(\text{pair}(x, y)) \rightarrow y$$

and the following equational theory:

$$E = \{\text{pair}(a, \text{pair}(b, c)) = d\}$$

Completion terminates successfully with the following equational system:

$$E = \{\text{pair}(a, \text{pair}(b, c)) = d, a = \text{left}(d), \\ \text{pair}(b, c) = \text{right}(d), b = \text{left}(\text{right}(d)), c = \text{right}(\text{right}(d))\}$$

We can characterize a simple case in which completion always succeeds: We call a term a *jack* when it is either shallow and linear, or  $f(t_1, \dots, t_i, \dots, t_n)$  such that some  $t_i$  is shallow and linear, and each  $t_j$  with  $j \neq i$  is a constant.

**Lemma 2.** *When  $R$  is a non-overlapping rewrite system of rules  $g \rightarrow x$  where each  $g$  is a jack,  $x \in \text{Vars}(g)$ , and  $E$  a ground equational system such that no constant occurring on a left-hand side of  $R$  is a side of  $E$ , then completion of any variable-disjoint and linear equation system succeeds.*

**Proof** The rewrite system is, as an easy consequence of the hypotheses, terminating and orthogonal. Since any right-hand side is subterm of a left-hand-side, which in turn is linear, all terms involved and hence all critical pairs are linear. If  $l = r$  is an equation and  $g\sigma_p = l\sigma$  an overlap, then due to the definition of jacks and the third condition in the lemma,  $g\sigma[\bullet]_p$  is ground, and hence the critical pair is variable-disjoint. See Appendix D for a termination proof of completion.  $\square$

Here is an example of a term rewrite system that satisfies the condition of Lemma 2. This system describes the cryptographic operators of pairing and projection, and asymmetric encryption and decryption for *fixed* keys.

$$\begin{array}{ll} \text{left}(\text{pair}(x, y)) \rightarrow x & \text{dec}(\text{inv}(a), \text{enc}(a, x)) \rightarrow x \\ \text{right}(\text{pair}(x, y)) \rightarrow y & \text{dec}(\text{inv}(b), \text{enc}(b, x)) \rightarrow x \\ \text{enc}(a, \text{dec}(\text{inv}(a), x)) \rightarrow x & \text{enc}(b, \text{dec}(\text{inv}(b), x)) \rightarrow x \end{array}$$

In this case, equational axioms may not contain  $a$  or  $b$  (the ground subterms of the left-hand sides of  $R$ ). The generalization of these axioms to arbitrary keys represented by a variable, i.e.  $\text{dec}(\text{inv}(y), \text{enc}(y, x)) \rightarrow x$  would lead to a left-hand side that is not a jack.

Note that there might in general be an infinity of equational systems, and the completion algorithm has to terminate successfully on each of them. In order to integrate this into a practical algorithm one would have to perform the completion on demand for the only finitely many congruence relations that may occur in any given first-order formula.

In the following we will show that Theorem 4 does not hold when the completion is no longer variable disjoint:

**Theorem 5.** *The first-order theory of  $A(\Sigma, (E_i)_{1 \leq i \leq 3}, (L_j)_{1 \leq j \leq 2})$  is undecidable when  $\Sigma$  is finite,  $R$  is a rewrite system whose rules have the form  $f(x, c) = x$  for  $f \in \Sigma_2$ ,  $c \in \Sigma_0$  and  $x \in V$ ,  $E_1, E_2$  are two sets of ground equations over  $\Sigma$ ,  $E_3 = \emptyset$ , and when the congruence relation  $=_i$  is interpreted as  $=_{E_i, R}$ .*

**Proof** The proof is very similar to the proof of Theorem 3, with a reduction of the Post correspondence problem (PCP). The main difference is that we consider binary terms instead of unary terms (words).

Let us consider an instance of PCP without empty words  $\mathcal{P} := \{(u_i, v_i) \mid u_i, v_i \in \{a, b\}^+, 1 \leq i \leq N\}$  with  $u_i = u_{i,1} \dots u_{i,|u_i|}$  and  $v_i = v_{i,1} \dots v_{i,|v_i|}$  for all  $1 \leq i \leq N$  and let  $L := \max(|u_i|, |v_i| \mid i \leq N)$ . The signature is now

$$\Sigma := \{f : 2, a : 0, b : 0, \square : 2\} \cup \{P_{i,j} : 0 \mid 1 \leq i \leq N, 1 \leq j \leq L\}.$$

Given a word  $w = w_1 \dots w_n \in \Sigma_0^*$ , we denote  $w^t$  the term  $f(w_1, \dots, f(w_n, b)) \in T(\Sigma)$ . We extend this notation to sets of words in the natural way.

The tree automata are defined the same way as in the proof of Theorem 3

- $L_\alpha$  recognizes  $(\{a, b\}^+)^t$  and
- $L_P$  recognizes  $(P_i \mid 1 \leq i \leq L)^t$ .

The transformation of  $P_{i,1} \dots P_{i,L}$  into  $u_i$  and  $v_i$  is now done using ground equations for replacement of constant symbols and collapsing rewrite rules for erasing the superfluous  $\square$ 's.

$$\begin{aligned}
E_1 &= \{P_{i,j} = u_{i,j} \mid 1 \leq i \leq N, 1 \leq j \leq |u_i|\} \\
&\cup \{P_{i,j} = \square \mid 1 \leq i \leq N, |u_i| < j \leq L\} \\
E_2 &= \{P_{i,j} = v_{i,j} \mid 1 \leq i \leq N, 1 \leq j \leq |v_i|\} \\
&\cup \{P_{i,j} = \square \mid 1 \leq i \leq N, |v_i| < j \leq L\} \\
E_3 &= \emptyset \\
R &= \{f(\square, x) \rightarrow x\}
\end{aligned}$$

Finally, the closed formula  $\phi$  over  $A(\Sigma, (E_i)_{1 \leq i \leq 3}, (L_\alpha, L_P))$  is the same as for Theorem 3

$$\phi := \exists x, u, v. L_P(x) \wedge x =_{E_1} u \wedge x =_{E_2} v \wedge L_\alpha(u) \wedge L_\alpha(v) \wedge u = v.$$

We can show similarly as for Theorem 3 (see Appendix C) that  $\mathcal{P}$  has a solution iff  $\phi$  is satisfiable in  $A(\Sigma, (E_i)_{1 \leq i \leq 3}, R, (L_\alpha, L_P))$ .  $\square$

Note that in the case of the proof of Theorem 5, completion of for instance an equation  $P_{i,j} = \square$  by the rule  $f(\square, x) \rightarrow x$  yields the non variable-disjoint equation  $f(P_{i,j}, x) = x$ .

## 5 Application to a Spatial Logic for the Applied $\pi$ -Calculus

We investigate the model checking problem for a spatial equational logic for the applied  $\pi$ -calculus [23] ( $A\pi$  for short), an extension of the  $\pi$ -calculus [24] where processes may communicate terms through channels. These terms are tested for equality using an equational theory  $\mathcal{E}$ , which is global and defined by the user so as to allow her to choose what cryptographic primitives are used, as well as local axioms placed in a *frame*, which act as a record of what has been sent to the environment so far. The calculus is parametric in  $\mathcal{E}$ , so as to be flexible w.r.t. the properties that the user wants to model about her cryptographic primitives.

For example, assuming  $\mathcal{E}$  is the equational theory of pairs, the frame  $\{u = \text{pair}(s, w), v = \text{left}(u)\}$  augments the equational theory with two equations on  $u$  and  $v$ . We can declare  $s$  to be a secret name by making it hidden, using the following notation:  $F = \nu s. \{u = \text{pair}(s, w), v = \text{left}(u)\}$ . Here,  $s$  denotes a *name* in applied  $\pi$ -calculus, whereas  $u, v$  denote a *frame variable*. Names are used to refer to channels, nonces, and secret keys, while each  $A\pi$  variable identifies a message whose content is the (ground) term associated with it in the frame. We will represent both frame variables and names as constants of our signature, keeping them distinct from the first-order variables. More formally, we suppose given a signature  $\Sigma$ , along with an equational theory  $\mathcal{E}$ .  $\Sigma$  contains the sets  $\mathcal{V}^\pi$  and  $\mathcal{N}^\pi$ , which are disjoint, infinite, and represent respectively  $A\pi$  variables and names.  $V$  is the usual set of (first-order) variables, distinct from  $\mathcal{V}^\pi$ .

*Notations.* We will use the letters  $h, n, m, s$  to refer to elements of  $\mathcal{N}^\pi$ ,  $u, v, w$  for elements of  $\mathcal{V}^\pi$ ,  $a, b, c$  for elements of  $\mathcal{N}^\pi \cup \mathcal{V}^\pi$  and  $x, y, z$  for elements of  $V$ . We write  $t$  for arbitrary terms in  $T(\Sigma, V)$ , and  $r$  for ground terms in  $T(\Sigma)$ .  $fn(t)$  and  $fav(t)$  respectively denote the sets of free names and free  $\lambda$ -variables of  $t$ , defined as usual, and  $fnav(t) := fn(t) \cup fav(t)$ .  $\uplus$  is used to denote the union of *disjoint* sets.

A *frame*  $F$  is a record of the current knowledge of the environment, in the form of *active substitutions*, each accounting for a message that has been sent over the network. For simplicity, we define a frame as a pair  $(H, S)$ , where  $H$  is a set of hidden names and  $S$  is a set of ground equations of the form  $u = r$ . We only consider frames  $F = (H, \{u_1 = r_1, \dots, u_k = r_k\})$  where the  $u_i$ 's are pairwise disjoint, the  $r_i$ 's are ground, and there is no cycle in the  $\lambda$ -variables (*i.e.* there is an ordering  $(i_1, \dots, i_k)$  of the set of substitutions such that  $u_{i_j} \notin fav(r_{i_{j'}})$  for  $j \leq j'$ ). The  $u_i$ 's (resp.  $r_i$ ) form the *domain* (resp. *codomain*) of  $F$ , written  $dom(F)$  (resp.  $codom(F)$ ). Frames are considered up to the following structural congruence relation:

**Definition 1 (Structural congruence)** *Structural congruence*  $\equiv$  is the smallest equivalence relation on frames that is stable by  $\alpha$ -conversion on hidden names and satisfies the following rules:

$$\begin{array}{ll}
\alpha\text{-CONV} & (H, S) \equiv (H[n \leftarrow n'], S[n \leftarrow n']) \\
& \text{if } n \in H \text{ and } n' \notin H \cup fn(S) \\
\text{NEW-0} & (H, \emptyset) \equiv (\emptyset, \emptyset) \\
\text{REWRITE} & (H, \{u_1 = r_1, \dots, u_k = r_k\}) \equiv (H, \{u_1 = r'_1, \dots, u_k = r'_k\}) \\
& \text{if } \forall i \in \{1, \dots, k\}. \mathcal{E} \vdash r_i = r'_i
\end{array}$$

Hence names in  $H$  are bound, and for  $F = (H, S)$ , we define  $fn(F) = fn(S) - H$ . In the full  $\lambda$ -setting, frames have a more complicated presentation, essentially due to the fact that they are coupled with processes. One significant simplification (in that our method depends on it) is that we assume that substitutions of a frame cannot apply to other substitutions of this frame. In all other respects, our presentation is equivalent to the standard one. In particular, we consider the standard parallel composition, restricted to frames:

**Definition 2 (Frame Composition)** *Two frames*  $F_1 = (H_1, S_1)$  *and*  $F_2 = (H_2, S_2)$  *are orthogonal if*  $H_1 \cap H_2 = \emptyset$ ,  $dom(F_1) \cap dom(F_2) = \emptyset$  *and*  $fn(codom(S_1)) \cap H_2 = fn(codom(S_2)) \cap H_1 = \emptyset$ . *The composition*  $F = F_1 * F_2$  *of orthogonal frames*  $F_1, F_2$  *is the frame*  $(H_1 \uplus H_2, S_1 \uplus S_2)$ .

As usual, we will write  $F_1 \equiv F_2 * F_3$  if there are  $F'_1, F'_2, F'_3$  such that  $F'_1 = F'_2 * F'_3$  and  $F_i \equiv F'_i$ . Finally, let us formalize the intuition that hidden names cannot be tested for equality:

**Definition 3** *Two ground terms*  $r_1$  *and*  $r_2$  *are equal in the frame*  $F$ , *written*  $F \vdash r_1 = r_2$  *when there exists a frame*  $F' = (H', S') \equiv F$  *such that*  $fn(t_1, t_2) \cap H' = \emptyset$  *and*  $\mathcal{E}, S' \vdash r_1 = r_2$ .

An important notion in  $A\pi$  is the deducibility of terms:

**Definition 4 (Deducibility)** *A ground term  $r$  is said to be deducible from the frame  $F = (H, S)$  if there exists a term  $r'$  such that  $fn(r') \cap H = \emptyset$  and  $(\emptyset, S) \vdash r = r'$ .*

For instance, the term  $s$  is deducible from the frame  $F = (\{s\}, \{u = pair(s, w)\})$  by taking  $r' = left(u)$ . In this case, the deducibility of  $s$  might correspond to a leak in the frame  $F$ , as  $s$  is supposed to be secret, yet it can be retrieved using only the publicly available piece of information  $u$ .

### 5.1 A Fragment of $A\pi\mathcal{L}$ for Frames

Spatial logics have been proposed to reason locally and modularly on algebraic models of distributed systems such as ambients [25] or  $\pi$ -calculus [26].  $A\pi\mathcal{L}$  [14] is a spatial logic for the applied  $\pi$ -calculus. In all generality, and even in the case of an empty equational theory, the model-checking for the whole logic is undecidable [27]. We present here a fragment for which it becomes decidable when the global equational theory  $\mathcal{E}$  can be represented as a rewrite system satisfying the conditions of Lemma 2. Consider the logic  $\mathcal{L}$  formed by the formulas  $\Phi$  of the following grammar, where  $r, r_1, r_2$  denote *ground* terms.

$$\Phi ::= r_1 = r_2 \mid x = r \mid x = x' \mid \emptyset \mid \textcircled{c}a \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \exists x. \Phi \mid \mathcal{I}a. \Phi \mid \text{H}n. \Phi \mid \Phi_1 * \Phi_2 \mid \Phi \odot a$$

$$\begin{aligned} F, \sigma \models t_1 = t_2 &\Leftrightarrow F \vdash t_1 \sigma = t_2 \sigma \\ F, \sigma \models \emptyset &\Leftrightarrow F \equiv (\emptyset, \emptyset) \\ F, \sigma \models \textcircled{c}a &\Leftrightarrow \forall F' \equiv F. a \in fnav(F') \\ F, \sigma \models \neg\Phi &\Leftrightarrow F, \sigma \not\models \Phi \\ F, \sigma \models \Phi_1 \wedge \Phi_2 &\Leftrightarrow F, \sigma \models \Phi_1 \text{ and } F, \sigma \models \Phi_2 \\ F, \sigma \models \exists x. \Phi &\Leftrightarrow \exists r \in T(\Sigma). F, (\sigma \cup \{x \mapsto r\}) \models \Phi \\ F, \sigma \models \mathcal{I}a. \Phi &\Leftrightarrow \exists a' \notin fnav(F, \sigma, \Phi). F, \sigma \models \Phi[a \leftarrow a'] \\ F, \sigma \models \text{H}n. \Phi &\Leftrightarrow \exists n' \notin fn(F, \sigma, \Phi). \exists (H', S'). F \equiv (\{n'\} \uplus H', S') \text{ and } (H', S'), \sigma \models \Phi[n \leftarrow n'] \\ F, \sigma \models \Phi_1 * \Phi_2 &\Leftrightarrow \exists F_1, F_2. F \equiv F_1 * F_2, F_1, \sigma \models \Phi_1 \text{ and } F_2, \sigma \models \Phi_2 \\ F, \sigma \models \Phi \odot n &\Leftrightarrow (\{n\} \cup H, S), \sigma \models \Phi \\ F, \sigma \models \Phi \odot u &\Leftrightarrow S = S_1 \uplus \{u = r\} \text{ and } (H, S_1[u \leftarrow r]), \sigma \models \Phi \end{aligned}$$

**Fig. 1.** Satisfaction relation of  $\mathcal{L}$  for a frame  $F = (H, S)$

The semantics of a formula  $\Phi$  is given by the satisfaction relation of Fig. 1 for a frame  $F$  and a valuation  $\sigma$  mapping term variables to ground terms. Intuitively,  $t_1 = t_2$  (which captures the first three predicates of the grammar  $r_1 = r_2$ ,  $x = r$  and  $x = x'$ ) is an equality test under  $F$ ,  $\emptyset$  describes the empty frame,  $\textcircled{c}a$  is true whenever the name or  $A\pi$  variable  $a$  appears free in all the frames equivalent to  $F$ ,  $\neg$ ,  $\wedge$  and  $\exists$  is the classical first-order fragment,  $\mathcal{I}a$  is the Gabbay-Pitts

quantifier over fresh names or  $\Lambda\pi$  variables,  $\mathbf{H}n$  is a quantifier over hidden names of the frame,  $*$  is the spatial conjunction that decomposes  $F$  into two disjoint parts, and  $\Phi \odot a$  hides the name or  $\Lambda\pi$  variable  $a$  in  $F$  and proceeds with  $\Phi$ . In the last case, as  $\Lambda\pi$  variable hiding is not part of our frame syntax, we simulate the effect it has in the usual setting of the  $\Lambda\pi$ , namely to apply the corresponding substitution and discard it.

This logic can express many properties about frames, and in particular deducibility; the formula below is true in any frame if and only if one of the hidden names is revealed by the frame:

$$\exists x. \mathbf{H}s. x = s$$

As the term quantification is placed first, the guessed term cannot contain the revealed names. The general deducibility problem can also be expressed in this fragment, although the formula depends on the frame  $(\{h_1, \dots, h_l\}, \{u_1 = t_1, \dots, u_k = t_k\})$  due to  $\alpha$ -conversion issues:

$$\exists x. \mathbf{H}h_1, \dots, h_l. (x = t \wedge u_1 = t_1 \wedge \dots \wedge u_k = t_k)$$

## 5.2 From Spatial to Equational

In this section, we reduce the model-checking problem for  $\mathcal{L}$  to the evaluation of an equational formula over a term algebra. We assume given a signature  $\Sigma$  and an equational theory  $\mathcal{E}$  defined by a term rewriting system  $R_{\mathcal{E}}$  satisfying all hypothesis of Section 4. For every finite set  $S$  of ground equations of the form  $u = r$ ,  $=_S$  denotes the congruence axiomatized by  $S \cup \mathcal{E}$ . We write  $\mathcal{S}$  to denote the set of all such  $S$ . Let us moreover write  $A$  to denote the structure  $A(\Sigma, (=_S)_{S \in \mathcal{S}}, (C_a)_{a \in \mathcal{N}^\pi \cup \mathcal{V}^\pi}, R_{\mathcal{E}})$  where  $C_a := \{t \mid a \in \text{fn}(t)\}$ . The grammar of the target logic  $\mathcal{L}_{\text{eq}}$  is described below, and its semantics is given by the satisfaction relation between valuations and formulas of Figure 2.

$$\phi ::= r_1 =_S r_2 \mid x =_S r \mid x_1 =_S x_2 \mid n \in \text{fn}(t) \mid u \in \text{fav}(t) \mid \phi_1 \wedge \phi_2 \mid \neg \phi \mid \exists x. \phi \mid \forall n. \phi$$

$$A, \sigma \models t_1 =_S t_2 \Leftrightarrow \mathcal{E}, S \vdash t_1 \sigma = t_2 \sigma$$

$$A, \sigma \models n \in \text{fn}(t) \Leftrightarrow n \in \text{fn}(t\sigma)$$

$$A, \sigma \models u \in \text{fav}(t) \Leftrightarrow u \in \text{fav}(t\sigma)$$

$$A, \sigma \models \phi_1 \wedge \phi_2 \Leftrightarrow A, \sigma \models \phi_1 \text{ and } \sigma \models \phi_2$$

$$A, \sigma \models \neg \phi \Leftrightarrow A, \sigma \not\models \phi$$

$$A, \sigma \models \exists x. \phi \Leftrightarrow \exists r \in T(\Sigma). A, \sigma \cup \{x \rightarrow r\} \models \phi$$

$$A, \sigma \models \forall n. \phi \Leftrightarrow \exists n' \notin \text{fn}(\sigma, \phi). A, \sigma \models \phi[n \leftarrow n']$$

**Fig. 2.** Satisfaction relation of  $\mathcal{L}_{\text{eq}}$

Let us observe that, for  $\phi \in \mathcal{L}_{\text{eq}}$ , it is decidable whether  $A, \sigma \models \phi$ : the Gabbay-Pitts quantifiers can be eliminated by first rewriting the formula in prenex form (the only non-homomorphic case being  $\exists x. \forall n. \phi \Leftrightarrow \forall n. \exists x. (\neg n \in \text{fn}(x)) \wedge \phi$ ), and then dropping them. The remaining formula is a standard first-order formula over the structure  $A$ , which can be decided according to Section 4.

Let us now detail the reduction. We will present a translation  $H, S, \Phi \mapsto \langle H, S, \Phi \rangle$ , that associates an equational formula in  $\mathcal{L}_{\text{eq}}$  to a frame  $(H, S)$  and a spatial formula  $\Phi$ . The inductive property we want to prove on the translation is as follows:

**Lemma 5 (Inductive hypothesis)** *For all  $\sigma, S, \Phi, H: A, \sigma \models \langle H, S, \Phi \rangle$  iff  $(H, S), \sigma \models \Phi$ .*

*Notations.* We write  $\mathbf{t} = \mathbf{t}'$  for  $\bigwedge_{i=1}^n t_i = t'_i$  (and similarly when the right-hand side is a set of terms). Arities are implicitly supposed to match: for instance, in  $\exists \mathbf{t}. \mathbf{t} = \text{codom}(S)$ ,  $\mathbf{t}$ 's size is implicitly chosen to match the size of  $\text{codom}(S)$ . Finally,  $\top$  (resp.  $\perp$ ) is a formula that is always true (resp. always false), for instance  $u = u$  for some  $u$  (resp.  $\neg u = u$ ).

The translation of  $\odot u$  follows its semantics and thus is quite straightforward. It is defined as  $\perp$  when  $u \in H$ ,  $\top$  when  $u \in \text{dom}(S)$ , and as shown below otherwise.

$$\langle H, S, \odot u \rangle := \forall \mathbf{t}. \mathbf{t} = \text{codom}(S) \Rightarrow u \in \text{fn}(\mathbf{t})$$

When hiding a variable  $u \in \text{dom}(S)$ , we need to apply the corresponding substitution to the rest of the frame and then throw the substitution on  $u$  away. Thus, if  $u \notin \text{dom}(S)$ , then  $\langle H, S, \Phi \odot u \rangle := \perp$ , and otherwise we let:

$$\langle H, S \uplus \{u = r\}, \Phi \odot u \rangle := \langle H, S[u \leftarrow r], \Phi \rangle$$

Hiding a name consists merely of adding  $h$  to the set of hidden names, and term quantification is left as-is, since the semantics of  $\exists x$  for  $\mathcal{L}$  and  $\mathcal{L}_{\text{eq}}$  are the same. As we know all the hidden names of the frame, we can treat name revelation as a disjunction over those names, plus one fresh extra name to model the fact that we can reveal “fake” hidden names:

$$\langle H, S, \Phi \odot n \rangle := \langle H \cup \{n\}, S, \Phi \rangle \quad \langle H, S, \exists x. \Phi \rangle := \exists x. \langle H, S, \Phi \rangle$$

$$\langle H, S, \text{Hn}. \Phi \rangle := \forall n'. \bigvee_{h \in H \uplus \{n'\}} \langle H \setminus \{h\}, S[h \leftarrow n'], \Phi[n \leftarrow n'] \rangle$$

To translate an equality  $t_1 = t_2$ , one has to take care of the hidden names of  $S$ , as  $\mathcal{L}_{\text{eq}}$  only allows public frames in its grammar. To overcome this situation, we first replace the names of  $S$  in  $H$  that should be restricted with fresh names  $H'$  such that  $H' \cap \text{fn}(S, t_1, t_2) = \emptyset$ . It is easy to check that these fresh names behave like hidden names for the equality test.

$$\langle \{h_1, \dots, h_k\}, S, t_1 = t_2 \rangle := \forall h'_1, \dots, h'_k. t_1 =_{S[h_1, \dots, h_k \leftarrow h'_1, \dots, h'_k]} t_2$$

To translate  $*$ , we need to be able to state that the set of hidden names appearing in two subframes are disjoint one from another up to rewriting of terms using the equational theory. This is achieved by defining the operator  $\mathbf{t}_1 \perp^H \mathbf{t}_2$  below which states that two sets of names  $\mathbf{t}_1$  and  $\mathbf{t}_2$  may be rewritten not to share names in  $H$ :

$$\mathbf{t}_1 \perp^H \mathbf{t}_2 := \exists \mathbf{x}_1, \mathbf{x}_2. \mathbf{x}_1 = \mathbf{t}_1 \wedge \mathbf{x}_2 = \mathbf{t}_2 \wedge \bigwedge_{h \in H} (h \in \text{fn}(\mathbf{x}_1) \Rightarrow h \notin \text{fn}(\mathbf{x}_2))$$

The translation of frame composition then only needs to quantify over all 2-partitions of the set of active substitutions that yield orthogonal subframes:

$$\langle H, S, \Phi_1 * \Phi_2 \rangle := \bigvee_{S_1 \uplus S_2 = S} (\text{codom}(S_1) \perp^H \text{codom}(S_2) \wedge \langle H, S_1, \Phi_1 \rangle \wedge \langle H, S_2, \Phi_2 \rangle)$$

This particular step of our translation would be unsound if substitutions of the frame could be applied to other substitutions of the frame, like in the original  $A\pi$ .

Finally,  $\langle H, S, \neg\Phi \rangle := \neg\langle H, S, \Phi \rangle$ ,  $\langle H, S, \Phi_1 \wedge \Phi_2 \rangle := \langle H, S, \Phi_1 \rangle \wedge \langle H, S, \Phi_2 \rangle$ ,  $\langle H, S, \emptyset \rangle := \top$  if  $S = \emptyset$  and  $\perp$  otherwise.

This translation gives us the following theorem and corollary:

**Theorem 6** *For all frame  $F$  and formula  $\Phi$  of  $\mathcal{L}$  there is a formula  $\phi$  of  $\mathcal{L}_{eq}$  such that:*

$$F \models \Phi \text{ if and only if } A \models \phi .$$

**Corollary 7** *If the global equational theory can be expressed using a rewriting system for which the procedure described in Section 4 terminates then the model-checking problem for  $\mathcal{L}$  is decidable.*

## 6 Conclusion

Classically used decision procedures for first-order theories seem not be applicable when faced with multiple congruence relations defined by independent equational axioms. Automata-based methods, on the other hand, have the advantage that the combination of different predicates, each of them recognizable for the same encoding of the elements of the algebra, comes for free. However, automata-based methods can handle only restricted classes of equational axioms. Whether it is possible to push the method to, for instance, non left-linear background equational theories like  $\text{dec}(x, \text{enc}(x, y)) \rightarrow y$  is up to future work.

In the case of the application considered here, we also simplified the structural congruence for frames, although we would have liked to consider the standard structural congruence. One possible avenue would be to make substitutions explicit in the background equational theory, with axioms like  $\text{subst}(u, u, x) = x$ ,  $\text{subst}(v, u, x) = v$ ,  $\text{subst}(f(t), x, y) = f(\text{subst}(t, x, y))$ , etc. However, we do not yet know how to handle such axioms. Considering a larger fragment of  $A\pi\mathcal{L}$  would also be challenging. In particular, we do not know yet how to treat  $A\pi$

variable revelation  $Hu.\Phi$  in our setting, as it amounts to quantifying over a new, unknown substitution  $u = r$  against which terms can be tested. In other words, we would have not only to consider multiple congruence relations, but also to quantify over them.

## References

1. Malc'ev, A.I.: Axiomatizable classes of locally free algebras of various type. In Benjamin Franklin Wells, I., ed.: *The Metamathematics of Algebraic Systems: Collected Papers 1936–1967*. North Holland (1971) 262–281
2. Maher, M.J.: Complete axiomatizations of the algebras of finite, rational and infinite trees. In: *LICS, Edinburgh, Scotland, UK (July 1988)* 348–357
3. Comon, H., Lescanne, P.: Equational problems and disunification. *Journal of Symbolic Computation* **7** (1989) 371–425
4. Comon, H.: Solving symbolic ordering constraints. *Int. J. Found. Comput. Sci.* **1**(4) (1990) 387–412
5. Jouannaud, J.P., Okada, M.: Satisfiability of systems of ordinal notation with the subterm property is decidable. In Albert, J.L., Monien, B., Artalejo, M.R., eds.: *International Colloquium on Automata, Languages and Programming. Volume 510 of LNCS.*, Madrid, Spain, Springer Verlag (1991) 455–468
6. Su, Z., Aiken, A., Niehren, J., Priesnitz, T., Treinen, R.: The first-order theory of subtyping constraints. In: *POPL'02, Portland, OR, USA, ACM (January 2002)* 203–216
7. Kuncak, V., Rinard, M.C.: Structural subtyping of non-recursive types is decidable. In: *Logic in Computer Science, Ottawa, Canada (June 2003)* 96–107
8. Caron, A.C., Coquide, J.L., Dauchet, M.: Encompassment properties and automata with constraints. In Kirchner, C., ed.: *Rewriting Techniques and Applications. Volume 690 of LNCS.*, Montreal, Canada, Springer-Verlag (June 1993) 328–342
9. Comon, H., Delor, C.: Equational formulae with membership constraints. *Information and Computation* **112**(2) (August 1994) 167–216
10. Baader, F., Snyder, W.: Unification theory. In Robinson, J.A., Voronkov, A., eds.: *Handbook of Automated Reasoning. Volume I.* Elsevier and MIT Press (2001) 445–532
11. Comon, H., Haberstrau, M., Jouannaud, J.P.: Syntacticness, cycle-syntacticness and shallow theories. *Information and Computation* **111**(1) (May 1994) 154–191
12. Treinen, R.: A new method for undecidability proofs of first order theories. *Journal of Symbolic Computation* **14**(5) (November 1992) 437–457
13. Marcinkowski, J.: Undecidability of the  $\exists^*\forall^*$  part of the theory of ground term algebra modulo an AC symbol. In Narendran, P., Rusinowitch, M., eds.: *Rewriting Techniques and Applications. Volume 1631 of LNCS.*, Trento, Italy, Springer-Verlag (July 1999) 92–102
14. Lozes, É., Villard, J.: A spatial equational logic for the applied  $\pi$ -calculus. In van Breugel, F., Chechik, M., eds.: *19th International Conference on Concurrency Theory. Volume 5201 of LNCS.*, Toronto, Canada, Springer-Verlag (August 2008) 387–401
15. Comon, H., Dauchet, M., Gilleron, R., Löding, C., Jacquemard, F., Lugiez, D., Tison, S., Tommasi, M.: Tree automata techniques and applications. Available on: <http://www.grappa.univ-lille3.fr/tata> (2007) release October, 12th 2007.

16. Blumensath, A., Grädel, E.: Automatic structures. In: Logic in Computer Science, Santa Barbara, CA (June 2000) 51–62
17. Dauchet, M., Tison, S., Heuillard, T., Lescanne, P.: Decidability of the confluence of ground term rewriting systems. In: Logic in Computer Science, Ithaca, NY, USA (June 1987) 353–359
18. Dauchet, M., Tison, S., Heuillard, T., Lescanne, P.: Decidability of the confluence of finite ground term rewrite systems and of other related term rewrite systems. *Information and Computation* **88**(2) (October 1990) 187–201
19. Gurevich, Y., Veanes, M.: Logic with equality: Partisan corroboration and shifted pairing. *Information and Computation* **152**(2) (1999) 205 – 235
20. Godoy, G., Huntingford, E., Tiwari, A.: Termination of rewriting with right-flat rules. In Baader, F., ed.: *Term Rewriting and Applications (RTA)*. Volume 4533 of LNCS., Springer (2007) 200–213
21. Post, E.L.: A variant of a recursively unsolvable problem. *Bulletin of the AMS* **52** (1946) 264–268
22. Huet, G.: Confluent reductions: Abstract properties and applications to term rewriting systems. *Journal of the ACM* **27**(4) (October 1980) 797–821
23. Abadi, M., Fournet, C.: Mobile values, new names, and secure communication. In: *POPL'01*. (2001) 104–115
24. Milner, R., Parrow, J., Walker, D.: A calculus of mobile processes, i. *Inf. Comput.* **100**(1) (1992) 1–40
25. Gordon, A., Cardelli, L.: Anytime, anywhere: Modal logics for mobile ambients. In Press, A., ed.: *POPL 2000*. (2000) 365–377
26. Caires, L., Cardelli, L.: A spatial logic for concurrency (part I). *Journal of Information and Computation* **186**(2) **186**(2) (2003)
27. Villard, J., Lozes, É., Treinen, R.: A spatial equational logic for the applied pi-calculus. Research Report LSV-08-10, LSV, ENS Cachan, France (March 2008) 44 pages.

## A Coping With Infinite Signatures

In order to show that  $A(\Sigma, (E_i)_{i \in I}, (L_j)_{j \in J})$  is an automatic structure we first have to define the encoding of the algebra as trees over a finite signature, the (minor) difficulty here being that the algebra contains trees over a possibly infinite alphabet but with a bounded arity.

If  $n$  is the maximal arity of a function symbol in  $\Sigma$  then we can arrange all function symbols of arity  $n$  into a (finite or infinite) enumeration. The signature of the automatic representation would consist of a constant  $0$ , a unary function  $s$ , and function symbols  $f_0, \dots, f_i$ , each  $f_i$  being of arity  $i + 1$ . A function symbol  $f(x_1, \dots, x_n) \in \Sigma$  of arity  $n$ , being number  $i$  in the enumeration, would be represented for the automatic representation as  $f_n(s^i(0), x_1, \dots, x_n)$ . The interpretation function  $\nu$  is straightforward to define, and the automaton for  $L_\delta$  would just have to ensure that  $0$  and  $s$  only occur as first argument of the  $f_i$ , and that the first argument of any  $f_i$  is of the form  $s^j(0)$ , possibly with a bound on  $j$  in case there are only finitely many function symbols of the corresponding arity.

More exactly, let  $n$  be the maximal arity occurring in  $\Sigma$ , and let for any  $i$ ,  $0 \leq i \leq n$ ,  $m_i \in \mathbf{N} \cup \infty$  be the number of symbols in  $\Sigma$  of arity  $i$ . The tree automaton that recognizes all terms that are encoding of a ground  $\Sigma$ -term is :

$$f_i(q_i, q \dots, q) \rightarrow q$$

for any  $i$ . Here, the state  $q$  is the only accepting state that recognizes all encodings of terms. For any  $i$ , the state  $q_i$  recognizes all encoding of natural numbers that are not larger than  $m_i$ : If  $m_i = \infty$  then we define

$$\begin{aligned} 0 &\rightarrow q_i \\ s(q_i) &\rightarrow q_i \end{aligned}$$

and for  $m_i \in \mathbf{N}$  we have that

$$\begin{aligned} 0 &\rightarrow p_i^0 \\ s(p_i^j) &\rightarrow p_i^{j+1} & j < m_i \\ p_i^{m_i} &\rightarrow q_i \end{aligned}$$

## B Proof of Theorem 2

In order to prove that the first-order theory of  $H(\Sigma, (E_i)_{1 \leq i \leq 3}, (L_j)_{1 \leq j \leq 3})$  is undecidable for ground equations (Theorem 2), we propose a reduction of the acceptance problem of the empty tape for deterministic Turing machines using a technique of *shifted pairing* [19].

Let  $M$  be a deterministic Turing machine computing on a tape bounded on the left and unbounded on the right, with input alphabet  $T \cup \{b\}$  ( $b$  is a special

blank symbol), state set  $S$ , initial state  $s_0$ , final state set  $S_f$ , and transition function  $\delta : (S \setminus S_f) \times \Gamma \cup \{b\} \rightarrow S \times \Gamma \cup \{b\} \times \{\text{left}, \text{right}, \text{stay}\}$ . Note that it is assumed *wlog* that entering a final state terminates the computation. Moreover, we also assume *wlog* that before entering a final state,  $M$  deletes the whole tape (all the symbols of  $\Gamma$  are replaced by  $b$ ).

We represent a *configuration* of  $M$  as a word  $c$  in  $\Gamma^* S \Gamma^* b^*$ , where the unique state symbol  $s \in S$  in  $c$  indicates the current position of the head of  $M$  in the configuration, in the sense that the head of  $M$  is on the symbol of  $\Gamma \cup \{b\}$  immediately following  $s$  in  $c$ . The length of a word  $c$  is denoted  $|c|$ . The languages of initial and final configurations of  $M$  are respectively  $C_0 := s_0 b^*$  and  $C_f := S_f b^*$ . The transition relation of  $M$ , written  $\vdash_M$ , is the binary relation on configurations such that  $c \vdash_M c'$  iff  $c'$  is obtained from  $c$  according to  $\delta$ . For instance, if  $\delta(s, a) = \langle s', a', \text{left} \rangle$ , then  $c = \alpha b s a \beta b^m$  with  $\alpha, \beta \in \Gamma^*$ ,  $b \in \Gamma$  and  $c' = \alpha s' b a' \beta b^m$ , if  $\delta(s, a) = \langle s', a', \text{right} \rangle$ , then  $c = \alpha s a \beta b^m$  and  $c' = \alpha a' s' \beta b^m$ , and if  $\delta(s, a) = \langle s', a', \text{stay} \rangle$ , then  $c = \alpha s a \beta b^m$  and  $c' = \alpha s' a' \beta b^m$ . A *computation* of  $M$  is a finite sequence  $c_0, c_1, \dots, c_n$  of configurations of  $M$  such that  $c_0 \in C_0$  and for all  $0 \leq i < n$ ,  $c_i \vdash_M c_{i+1}$ . It is *successful* if the state of  $c_n$  is final, *i.e.*  $c_n \in S_f b^*$  by hypothesis.

We shall encode the configurations and computations of  $M$  as right-combs built on the signature  $\Upsilon := \{f : 2, g : 2, b : 0, \# : 0\} \cup \{b : 0 \mid b \in \Gamma \cup S\}$ . Let us moreover extend  $\Upsilon$  into the signature  $\Sigma := \Upsilon \cup \Upsilon_0^{[2]}$  (*i.e.*  $\Sigma$  extends  $\Upsilon$  with the set of constant symbols of the form  $[a, b]$  with  $a, b \in \Sigma_0 \cup \{\square\}$  such that  $a$  or  $b$  is not  $\square$ ).

A computation  $c_0, \dots, c_n$  is encoded as a term  $f(c_0^t, \dots, f(c_n^t, b))$  of  $T(\Sigma)$ , where for all  $0 \leq i \leq n$ ,  $c_i^t$  is the term encoding of the configuration  $c_i = c_{i,1} \dots c_{i,k}$  defined as  $c_i^t := g(c_{i,1}, \dots, g(c_{i,k}, b))$ .

Let  $L_0 := \{c_0^t \mid c_0 \in C_0\}$  be the recognizable language of term encodings of initial configurations of  $M$ . Let  $L_c$  be the recognizable language of terms of the form  $f(c_0^t, \dots, f(c_n^t, b))$ , with  $n \geq 0$ , such that for all  $0 \leq i \leq n$ ,  $c_i$  is a configuration of  $M$  (*i.e.*  $c_i \in \Gamma^* S \Gamma^* b^*$ ) and  $c_n$  is a final configuration of  $C_f$ .

For technical convenience, we shall use below a simplified convolution product  $\otimes$  defined only on configurations of same length or a configuration and  $b$ :

$$\begin{aligned} g(a, s) \otimes g(b, t) &= g([a, b], s \otimes t) \\ b \otimes b &= b \\ g(a, s) \otimes b &= g([a, \square], s \otimes b) \end{aligned}$$

It is easy to verify that the set  $\{c^t \otimes d^t \mid c \vdash_M d, |c| = |d|\}$  is a recognizable tree language of  $T(\Sigma)$ . Hence, the following set is also a recognizable tree language (called shifted pairing language):

$$L_{sp} := \{f(c_0^t \otimes d_1^t, \dots, f(c_{n-1}^t \otimes d_n^t, f(c_n^t \otimes b, b))) \mid n \geq 0, \forall 0 \leq i < n. |c_i| = |d_{i+1}| \text{ and } c_i \vdash_M d_{i+1}\}.$$

Note that in the definitions of  $L_{sp}$  and  $L_c$ , the configurations  $c_0, \dots, c_n$  and  $d_1, \dots, d_n$  are arbitrary. In particular it is not required that the sequence  $c_1, \dots, c_n$  is a computation of  $M$  (otherwise the languages would not be recognizable!).

We define two ground equational theories describing roughly the left and right projections on terms of  $L_{sp}$ . More precisely, these theories  $E_1$  and  $E_2$  are defined by

$$\begin{aligned} E_1 &:= \{[a_1, a_2] = a_1 \mid [a_1, a_2] \in \Upsilon_0^{[2]}\} & E_2 &:= \{[a_1, a_2] = a_2 \mid a_1, a_2 \in \Sigma_0\} \\ E_3 &:= \emptyset & & \cup \{g([b, \square], b) = \#, \quad g([b, \square], \#) = \#\} \\ & & & \cup \{f(g([s^f, \square], \#), b) = b\} \quad (s^f \in S_f) \end{aligned}$$

Let us now consider the following closed first-order formula over  $H(\Sigma, (E_i)_{1 \leq i \leq 3}, (L_\otimes, L_c))$ :

$$\phi := \exists y, y_1, y_2, x_0. L_{sp}(y) \wedge y =_{E_1} y_1 \wedge L_c(y_1) \wedge y =_{E_2} y_2 \wedge L_c(y_2) \wedge L_0(x_0) \wedge y_1 = f(x_0, y_2).$$

Let us establish now the correctness of the reduction.

**Lemma 3.**  *$\phi$  is satisfiable in  $H(\Sigma, (E_i)_{i \in 1,2}, (L_{sp}, L_c, L_0))$  iff  $M$  admits a successful computation starting with a blank tape.*

**Proof** For the *if* direction, assume that there exists a finite computation  $c_0, \dots, c_n$  of  $M$  with  $c_0 \in C_0 = s_0 b^*$  and  $c_n \in S_f b^*$ . We can assume moreover that the configurations  $c_0, \dots, c_n$  have all the same length, using if necessary some padding with  $b$ 's at the right.

Let  $y = f(c_0^t \otimes c_1^t, \dots, f(c_{n-1}^t \otimes c_n^t, f(c_n^t \otimes b, b)))$ . By definition,  $y \in L_{sp}$ .

Let  $y_1 = f(c_0^t, \dots, f(c_n^t, b))$  and  $y_2 = f(c_1^t, \dots, f(c_n^t, b))$ . We can observe easily that  $y_1 \in L_c$ ,  $y =_{E_1} y_1$  and  $y =_{E_2} y_2$ . Moreover, with  $x_0 = c_0^t$ , we have  $x_0 \in L_0$  and  $y_1 = f(x_0, y_2)$ . Hence  $\phi$  is satisfiable in  $H(\Sigma, (E_i)_{i \in 1,2}, (L_{sp}, L_c, L_0))$ .

For the *only if* direction, assume that  $\phi$  is satisfiable, and let  $y, y_1, y_2, x_0$  be terms such that  $y \in L_{sp}$ ,  $y =_{E_1} y_1$ ,  $y =_{E_2} y_2$ ,  $y_1 \in L_c$ ,  $y_2 \in L_c$ ,  $x_0 \in L_0$  and  $y_1 = f(x_0, y_2)$ .

Let  $y = f(c_0^t \otimes d_1^t, \dots, f(c_{n-1}^t \otimes d_n^t, f(c_n^t \otimes b, b)))$  with  $n \geq 0$ , and for all  $0 \leq i < n$ ,  $|c_i| = |d_{i+1}|$  and  $c_i \vdash_M d_{i+1}$  (\*). Since  $y =_{E_1} y_1$  and  $y_1 \in L_c$ , it holds that  $y_1 = f(c_0^t, \dots, f(c_n^t, b))$ . Moreover  $c_n \in C_f$  (set of final configurations) by definition of  $L_c$ . Since  $y =_{E_2} y_2$  and  $y_2 \in L_c$ , we have necessarily  $y_2 = f(d_1^t, \dots, f(d_n^t, b))$  (the terms of  $L_c$  do not contain the symbols  $\square$  or  $\#$ ).

Finally,  $y_1 = f(x_0, y_2)$  implies that  $x_0 = c_0^t$  and  $d_i^t = c_i^t$  for all  $1 \leq i \leq n$ . From (\*), it follows that  $c_i \vdash_M c_{i+1}$  for all  $0 \leq i < n$ . Hence  $c_0, \dots, c_n$  is a successful computation of  $M$  starting with a blank tape since  $x_0 \in L_0$ .  $\square$

## C Proof of Theorem 3

In Section 3, we have presented a reduction of PCP into satisfiability of a formula. The ingredients of the reduction are the following, given an instance of PCP  $\mathcal{P} = \{(u_i, v_i) \mid u_i, v_i \in \{a, b\}^*, 1 \leq i \leq N\}$ ,

- the signature  $\Sigma = \{a : 1, b : 1, b : 0\} \cup \{P_{i,j} : 1 \mid 1 \leq i \leq N, 1 \leq j \leq L\}$  where  $L = \max(|u_i|, |v_i| \mid i \leq N)$ , and For all  $1 \leq i \leq N$ ,  $u_i = u_{i,1} \dots u_{i,|u_i|}$  and  $v_i = v_{i,1} \dots v_{i,|v_i|}$ ,

- one recognizable language  $L_\alpha = \{a, b\}^+b$   
(recall that a term of  $T(\Sigma)$  is written as a word of  $\Sigma_1^*\Sigma_0$ ),
- a second recognizable language  $L_P = \{P_i \mid 1 \leq i \leq N\}^*b$   
(recall that  $P_i$  denotes the word  $P_{i,1} \cdots P_{i,L}$ ),
- one flat equational theory  $E_1 = \{P_{i,j}(x) = u_{i,j}(x) \mid 1 \leq i \leq N, 1 \leq j \leq |u_i|\} \cup \{P_{i,j}(x) = x \mid 1 \leq i \leq N, |u_i| < j \leq L\}$ ,
- a second flat equational theory  $E_2 = \{P_{i,j}(x) = v_{i,j}(x) \mid 1 \leq i \leq N, 1 \leq j \leq |v_i|\} \cup \{P_{i,j}(x) = x \mid 1 \leq i \leq N, |v_i| < j \leq L\}$ ,
- the empty theory  $E_3 = \emptyset$ ,
- a closed first formula  $\phi = \exists x, u, v. L_P(x) \wedge x =_{E_1} u \wedge x =_{E_2} v \wedge L_\alpha(u) \wedge L_\alpha(v) \wedge u = v$ .

Let us show that the reduction above is correct.

**Lemma 4.**  $\phi$  is satisfiable in  $A(\Sigma, (E_i)_{1 \leq i \leq 3}, (L_\alpha, L_P))$  iff  $\mathcal{P}$  has a solution.

**Proof** For the *if* direction, assume that  $\mathcal{P}$  admits a solution  $(i_j)_{0 \leq j \leq k}$  with  $1 \leq i_0, \dots, i_k \leq N$ , and  $u_{i_0}u_{i_1} \dots u_{i_k} = v_{i_0}v_{i_1} \dots v_{i_k}$ . Let  $x = P_{i_0} \cdots P_{i_k}b$  and let  $u = u_{i_0}u_{i_1} \dots u_{i_k}b$  and  $v = v_{i_0}v_{i_1} \dots v_{i_k}b$ . Hence  $u = v$ . Moreover,  $x \in L_P$ ,  $u, v \in L_\alpha$ , and  $x =_{E_1} u$ ,  $x =_{E_2} v$ . Therefore  $\phi$  is satisfiable.

For the *only if* direction, assume that  $\phi$  is satisfiable, and let  $x, u, v$  be terms such that  $x \in L_P$ ,  $x =_{E_1} u$ ,  $x =_{E_2} v$ ,  $u \in L_\alpha$ ,  $v \in L_\alpha$ , and  $u = v$ . Let  $x = P_{i_0} \cdots P_{i_k}b$  for some  $1 \leq i_0, \dots, i_k \leq N$ . From  $x =_{E_1} u$  and  $u \in L_\alpha$ , it follows that necessarily  $u = u_{i_0}u_{i_1} \dots u_{i_k}b$ . Note that the equations of  $E_1$  can be applied in both direction, *i.e.*  $P_{i,j}$  can be replaced by  $u_{i,j}$  (or deleted) but also  $u_{i,j}$  can be replaced by another  $P_{i',j'}$  when  $u_{i,j} = u_{i',j'}$ . But this  $P_{i',j'}$  will eventually be placed by  $u_{i,j}$  in order to get  $u \in L_\alpha$  (there are no other replacement possible). Similarly,  $v = v_{i_0}v_{i_1} \dots v_{i_k}b$ . From  $u = v$ , it follows that  $(i_j)_{0 \leq j \leq k}$  is a solution of  $\mathcal{P}$ .  $\square$

## D Termination of Completion

**Lemma 5.** Let  $R$  be a non-overlapping rewrite system of rules  $g \rightarrow x$  where each  $g$  is a jack,  $x \in \text{Vars}(g)$ , and  $E$  a ground equational system such that no constant occurring on a left-hand side of  $R$  is a side of  $E$ , then completion of any variable-disjoint and linear equation system terminates.

**Proof** In the special situation of this lemma, critical pairs are formed as follows:

1. there is a substitution  $\sigma$  and a non-variable position  $p$  of  $g$  such that  $g\sigma|_p = l$ , in that case the critical pair is  $g[r]_p = d\sigma$ .
2. there is a substitution  $\sigma$  and a position  $p$  of  $l$  such that  $g\sigma = l|_p$ , in that case the critical pair is  $r = l[d\sigma]_p$ .

First note that addition of critical pairs maintains the invariant that no constant occurring on a left-hand side of  $R$  is a side of  $E$ . This is due to the fact that, in the first case,  $g$  cannot be a constant.

Let  $G$  denote the set of ground subterms of the left-hand sides of  $R$ . We define, for any term  $t$ ,  $\phi(t)$  as the size of  $t$ , where all terms in  $G$  are understood to have size 0. More precisely,

$$\phi(t) = \begin{cases} 0 & \text{if } t \in G \\ 1 + \sum_{i=1}^n \phi(t_i) & \text{if } t = f(t_1, \dots, t_n) \notin G \end{cases}$$

For any  $n$  there exist only finitely many terms  $t$  with  $\phi(t) \leq n$  since  $G$  is finite. We will show that when superposition of  $l = r$  with the rewrite rule  $g \rightarrow d$  leads to addition of the critical pair  $l' = r'$  then  $\phi(l') + \phi(r') \leq \phi(l) + \phi(r)$ . As a consequence, only finitely many critical pairs can be added. We consider the two cases above:

1. In that case we have, by the form of the rewrite system, that  $p$  is of length at most 1. Hence,  $g[\bullet]_p$  is of the form  $f(t_1, \dots, t_{i-1}, \bullet, t_{i+1}, \dots, t_n)$  where  $t_i \in G$ . As a consequence,  $\phi(r') = \phi(g[r]_p) \leq \phi(r) + 1$ .  
On the other hand,  $l$  cannot be an element of  $G$  since no side of the equational system is a ground subterm of a left-hand side of  $R$ , and hence  $\phi(l) > 0$ . We have that  $l' = d\sigma$  is a proper subterm of  $l$ , and hence that  $\phi(l') < \phi(l)$ .
2. First note that  $g\sigma$  cannot be an element of  $G$  since the rewrite system is orthogonal. Hence,  $\phi(l') < \phi(l)$ , and we conclude in this case since  $r = r'$ . □