Constrained rewriting in recognizable theories

Tony Bourdier, Horatiu Cirstea

Constrained rewriting in recognizable theories

Tony Bourdier and Horatiu Cirstea

INRIA & LORIA & Nancy Université
BP 101, 54602 Villers-les-Nancy Cedex, France
first.last@loria.fr *

Abstract. Rewriting has long been shown useful for equational reasoning but its expressive power is not always appropriate for certain situations, as for instance when dealing with relations over terms. That is why some generalizations of rewriting, such as strategic rewriting, conditional or constrained rewriting, have emerged. In particular, constraints over terms are very suitable to define sets of terms thanks to logic formulae. Works on constrained rewriting mainly focus on term algebra constraints (equality, disequality, matching, etc.) with a fixed predicate interpretation. We propose in this paper a notion of constrained rewriting whose constraints are first order formulae and we we concentrate on formulae whose predicates are freely interpreted as recognizables relations on tuples. We then then characterize a class of first order formulae for which we can decide the step of constrained rewriting.

1 Introduction

It is a common claim that rewriting is ubiquitous in computer science and mathematical logic. The rewriting concept appears from the very theoretical settings to the very practical implementations. Rewriting is used in semantics in order to describe the meaning of programming languages but also to perform deduction when describing by inference rules a theorem prover or a constraint solver. It is of course central in systems making the notion of rule an explicit object, like expert systems, programming languages based on equational logic, algebraic specifications, functional programming and transition systems. It is hopeless to try to be exhaustive and the cases we have just mentioned show part of the huge diversity of the rewriting concept. The central idea of rewriting is to use rewriting to mechanize as much as possible equational reasoning by using oriented axioms called rewrite rules. Several extensions of the basic rewriting has been proposed. For instance, conditional rewrite systems provide a way to handle partial operations and case analysis while class rewriting [13, 18] can be used to handle semantic data structures which can be modeled using equational axioms (that cannot be oriented without loosing the termination property of reduction). Rewriting with constraints [16] emerged as a unified way to cover the previous extensions but goes beyond this since it is very suitable to define sets of terms thanks to logic

* Thanks: ANR SSURF and Region Lorraine.

formulae. Most of the works on constrained rewriting focus on order, equality, disequality and membership constraints. Anti-pattern rewriting [15] and rewriting in many-sorted algebra [5] are two instances of the general constrained rewriting.

Constrained rewrite systems have been used to model systems restricted by security policies [3] for which the basic rewriting has shown its limitations. Rewriting is performed in this case *w.r.t.* an algebra representing the current state of the system and thus, even if the specification of the policy remains unchanged during the evolution of the system, its semantics (*i.e.* the associated rewriting relation) changes as the system evolves. The formalism defined in [3] specifies the conditions for solving the underlying constraints in algebras with a finite carrier. We propose here a more general formalism that can deal with infinite structures. The rewriting and consequently the underlying constrained matching are performed *w.r.t.* theories whose formulae are recognizable using tree automata. In this formalism, we characterize the constrained patterns for which a solution can be computed and recognized. In this paper, we propose a general definition of constrained rewriting and introduce the notion of recognizable theory *w.r.t* which the rewriting is performed. We characterize the class of corresponding matching problems for which the solutions can be computed and based on this, we identify a class rewrite systems for which the one-step constrained rewriting decidable.

2 Preamble

We assume the reader familiar with the standard notions of term rewriting, universal algebra, first order logic and tree automata. We refer to [10] for logic considerations, to [1] for rewriting considerations and to [7] for more details on tree automata. This section presents notations used throughout this paper.

We call *alphabet* any set E fitted with an application ar from E to $\mathbb{N}$ called *arity*. We say that $u \in E$ is n-ary and we write $u_{/n}$ if $ar(u) = n$. A *constant* is a symbol whose arity is 0. For any $m > 1$, E^m denotes the alphabet containing all $\langle u_1, \ldots, u_m \rangle$ with $u_i \in E \cup \{\Lambda\}$ such that $ar(\langle u_1, \ldots, u_m \rangle) = \max_{i \in [1,m]}(ar(u_i) \mid u_i \neq \Lambda)$. A signature will be denoted by $\Sigma = (\mathcal{F}, \mathcal{P})$ where $\mathcal{F}$ is an alphabet of function symbols and $\mathcal{P}$ an alphabet of predicate symbols. $\mathcal{T}_{\mathcal{F},\mathcal{X}}$ is the set of *terms* built from $\mathcal{F}$ and a set of variables $\mathcal{X}$. We write $\mathcal{T}_{\mathcal{F}}$ instead of $\mathcal{T}_{\mathcal{F},\varnothing}$ for the set of ground terms. For any set of symbols E, $\mathcal{T}_{\mathcal{F}[E]}$ denotes the set of ground terms over $\mathcal{F} \cup E$ where elements of E are considered as constants. Given a term $t \in \mathcal{T}_{\mathcal{F},\mathcal{X}}$ we denote by $\mathcal{V}ar(t)$ the set of variables occurring in t. $\mathcal{P}os(t)$ denotes the set of positions of t, ε the root position, $t_{|\omega}$, resp. $t(\omega)$, the subterm of t, resp. the symbol of t, at position ω, and $t\,[s]_\omega$ the term t with the subterm at position ω replaced by s. We denote by $>_{pref}$ the prefix order over positions. A *substitution* σ is a mapping from $\mathcal{X}$ to $\mathcal{T}_{\mathcal{F},\mathcal{X}}$ which is the identity except over a finite set of variables $\mathcal{D}om(\sigma)$, called the *domain* of σ, extended to an endomorphism of $\mathcal{T}_{\mathcal{F},\mathcal{X}}$. A substitution σ is often denoted by $\{x \mapsto \sigma(x) \mid x \in \mathcal{D}om(\sigma)\}$. If $\mathcal{C}odom(\sigma) \subseteq \mathcal{T}_{\mathcal{F}}$, σ is said to be ground. A context, denoted by $C[\Box, \ldots, \Box]$, is a ground term of $\mathcal{T}_{\mathcal{F}[\Box]}$, and $C[t_1, \ldots, t_n]$ denotes the term $C[\Box, \ldots, \Box]$ in which

the $\Box$ have been pairwise replaced by t_i. Any linear term can be seen as a context by replacing each of its variables by $\Box$.

Given a signature $\Sigma = (\mathcal{F}, \mathcal{P})$, atoms, literals and formulae over Σ are defined as usual. $\mathcal{F}or_{\Sigma,\mathcal{X}}$ denotes the set of formulae built from symbols in Σ and a set of variables $\mathcal{X}$. Free and bound variables of a formula φ are denoted by $\mathcal{FV}ar(\varphi)$ and $\mathcal{BV}ar(\varphi)$, respectively. We write $\varphi[x_1, \ldots, x_n]$ to specify that $\{x_1, \ldots, x_n\}$ are the free variables of φ and in this case $\varphi[t_1, \ldots, t_n]$ denotes the formula $\sigma(\varphi)$ with $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$. We call Σ-theory any set of Σ-formulae. Given a Σ-*interpretation* $\Im$, the carrier of $\Im$ is denoted by $|\Im|$. For any symbols $f \in \mathcal{F}$, $p \in \mathcal{P}$ and term $t \in \mathcal{T}_{\mathcal{F}}$, the interpretations of f, p and t are denoted by $f_{\Im}$, $p_{\Im}$ and $t_{\Im}$. The interpretation of the equality in $\Im$ is denoted by $=_{\Im}$. Given a set of variables $\mathcal{X}$, an $\Im$-*valuation* α is a mapping from $\mathcal{X}$ to $|\Im|$ and induces together with $\Im$ a mapping from $\mathcal{T}_{\mathcal{F},\mathcal{X}}$ to $|\Im|$. The semantics of a Σ-formula φ in $\Im$ according to the $\Im$-valuation α is denoted by $[\![\varphi]\!]^{\alpha}_{\Im}$. We write $\Im \models \varphi$ iff $[\![\varphi]\!]^{\alpha}_{\Im}$ is *true* for any valuation α. The set of all models of φ is denoted by $\mathcal{M}od(\varphi)$ and for any theory Γ, $\mathcal{M}od(\Gamma)$ is the set of interpretations which are models of all formulae in Γ. We write $\Gamma \models \varphi$ iff for any $\Im \in \mathcal{M}od(\Gamma)$, $\Im \models \varphi$.

We call *n-ary tree automaton* (over $\mathcal{F}$) any quadruple $\mathbb{A} = \langle\, \mathcal{F}, Q, F, \Delta \,\rangle$ such that $\mathcal{F}$ is a set of function symbols, Q is a finite set of *states*, F is a subset of Q whose elements are called *final states* and Δ is a relation over $\mathcal{T}_{\mathcal{F}^n[Q]} \times Q$ whose elements are called *transitions*. An element of $\mathcal{T}_{\mathcal{F}^n[Q]}$ is called a *configuration*. A transition $lhs \rightarrow rhs$ of Δ is *normalized* iff for any $\omega \neq \varepsilon$, $lhs(\omega) \in Q$. An automaton whose transitions are normalized is said normalized. A tree automaton is said *deterministic* iff all its transitions have a different left-hand side. Without loss of generality, we can consider that all automata are normalized and deterministic. The rewriting relation induced by Δ over $\mathcal{T}_{\mathcal{F}^n[Q]}$ is denoted by $\rightarrow_{\mathbb{A}}$ and the language recognized by $\mathbb{A}$ is $\mathcal{L}(\mathbb{A}) = \{\langle\, t_1, \ldots, t_n \,\rangle \in \mathcal{T}_{\mathcal{F}} \mid \exists q_f \in F, t_1 \otimes \ldots \otimes t_n \xrightarrow{*}_{\mathbb{A}} q_f\}$ where $t = t_1 \otimes \ldots \otimes t_n$ is the configuration such that: $\forall \omega \in \bigcup_{i=1}^{n} \mathcal{P}os(t_i)$, $t(\omega) = \langle\, t_1[\omega), \ldots, t_n[\omega) \,\rangle$ where $u[\omega) = u(\omega)$ if $\omega \in \mathcal{P}os(t)$ and Λ otherwise. A set E of n-tuples of terms (or equivalently n-ary relation) is said *recognizable* iff there exists an n-ary tree automaton $\mathbb{A}$ such that $E = \mathcal{L}(\mathbb{A})$. The following table recalls usual tree automata (operations) together with their semantics:

Notation	Language recognized by the automaton
$\mathbb{T}$ (0-ary)	the 0-tuple $\langle\, \rangle$
$\mathbb{F}$ (0-ary)	$\varnothing$
$\mathbb{A} \oplus \mathbb{A}'$	$\mathcal{L}(\mathbb{A}) \oplus \mathcal{L}(\mathbb{A}')$ where $\oplus$ is $\cap$, $\cup$, or $\times$
$\overline{\mathbb{A}}$	$(\mathcal{T}_{\mathcal{F}})^n \setminus \mathcal{L}(\mathbb{A})$
$\mathbb{U}niv^n_{\mathcal{F}}$	n-tuples $\langle\, t_1, \ldots, t_n \,\rangle$ of $\mathcal{T}_{\mathcal{F}}$
$\mathbb{I}dentity^n_{\mathcal{F}}$	n-tuples $\langle\, t, \ldots, t \,\rangle$ of $\mathcal{T}_{\mathcal{F}}$
$rec(t)$	- if $t = C[\mathbb{A}_1, \ldots, \mathbb{A}_n] \in \mathcal{T}_{\mathcal{F}[\mathbb{S}]}$ then $C[t_1, \ldots, t_n]$, $t_i \in \mathcal{L}(\mathbb{A}_i)$ - if $t = C[x_1, \ldots, x_n] \in \mathcal{T}_{\mathcal{F},\mathcal{X}}$ then $C[t_1, \ldots, t_n]$, $t_i \in \mathcal{T}_{\mathcal{F}}$
$\mathbb{S}wap_{i,j}(\mathbb{A})$	n-tuples $\langle\, t_1, \ldots, t_j, \ldots, t_i, \ldots, t_n \,\rangle$ *s.t.* $\langle\, t_1, \ldots, t_i, \ldots, t_j, \ldots, t_n \,\rangle \in \mathcal{L}(\mathbb{A})$
$\mathbb{C}yl_i(\mathbb{A})$	$(n+1)$-tuples $\langle\, t_1, \ldots, t_{i-1}, t, t_i, \ldots, t_n \,\rangle$ *s.t.* $\langle\, t_1, \ldots, t_n \,\rangle \in \mathcal{L}(\mathbb{A})$
$\mathbb{P}roj_i(\mathbb{A})$	$(n-1)$-tuples $\langle\, t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n \,\rangle$ *s.t.* there exists a $t \in \mathcal{T}_{\mathcal{F}}$: $\langle\, t_1, \ldots, t_{i-1}, t, t_{i+1}, \ldots, t_n \,\rangle \in \mathcal{L}(\mathbb{A})$
$\mathbb{P}roj_{t/i}(\mathbb{A})$	$\langle\, t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n \,\rangle$ *s.t.* $\langle\, t_1, \ldots, t_{i-1}, t, t_{i+1}, \ldots, t_n \,\rangle \in \mathcal{L}(\mathbb{A})$, $t \in \mathcal{T}_{\mathcal{F}}$

where $\mathbb{A}$ and $\mathbb{A}'$ are n-ary tree automata, n, i are integers and $\mathbb{S}$ is the set of symbols denoting an automaton.

3 Constrained rewriting

In [16] the authors give the foundations for constrained equational reasoning and propose a definition of constrained rewriting in which constraints are equational problems, *i.e.* a constrained rewriting dealing only with equality. Frameworks for rewriting with membership, equality and disequality constraints have been also proposed [11, 5, 6] and this can be seen as a first step towards an approach integrating more general predicates in equational reasoning. We go a step further and we consider in this paper more general constraints, namely, first order formulae. We introduce in this section the notion of constrained term rewrite rule and define the relation induced by a set of such rules.

Definition 1 (Constrained rewrite system, constrained rewriting). *A* constrained rule *over a signature $\Sigma = (\mathcal{F}, \mathcal{P})$ is a 3-tuple $(l, \varphi, r) \in \mathcal{T}_{\mathcal{F},\mathcal{X}} \times \mathcal{F}or_{\Sigma,\mathcal{X}} \times \mathcal{T}_{\mathcal{F},\mathcal{X}}$, denoted by $l \xrightarrow{\varphi} r$, such that $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l) \cup \mathcal{F}\mathcal{V}ar(\varphi)$. l (resp. r) is the left-hand side (resp. right-hand side) and φ the constraint of the rule (omitted when it is $\top$). A* constrained term rewrite system (CTRS) *over Σ is a set $\Re$ of constrained rules over Σ. Given a Σ-theory Γ, we say that $u \in \mathcal{T}_{\mathcal{F}}$* rewrites into *a term $v \in \mathcal{T}_{\mathcal{F}}$ w.r.t. $\Re$ and Γ, which is also denoted by $u \rightarrow_{\Re}^{\Gamma} v$ iff there exist (i) two terms t and t' in $\mathcal{T}_{\mathcal{F}}$, (ii) a position $p \in \mathcal{P}os(t)$, (iii) a rewrite rule $l \xrightarrow{\varphi} r \in \Re$, and (iv) a ground substitution σ with $\mathcal{D}om(\sigma) = \mathcal{V}ar(l) \cup \mathcal{F}\mathcal{V}ar(\varphi)$ such that $t_{|p} = \sigma(l)$, $t' = t\,[\sigma(r)]_p$ and $\Gamma \models \{u = t\ ;\ \sigma(\varphi)\ ;\ v = t'\}$.*

The intermediate rewriting step from t to t' is denoted by $t \rightarrow_{\mathcal{R}\|\Gamma} t'$. A constrained rewriting step can thus be written $\rightarrow_{\mathcal{R}}^{\Gamma} \triangleq =_{\Gamma} \cdot \rightarrow_{\mathcal{R}\|\Gamma} \cdot =_{\Gamma}$. The terms t' such that $t =_{\Gamma} t'' \rightarrow_{\mathcal{R}\|\Gamma} t'$ are called the $\Re_{\Gamma}$-reducts of t. A term is said $\Re_{\Gamma}$-irreducible if it has no $\Re_{\Gamma}$-reducts and t' is an $\Re_{\Gamma}$ normal form of t if $t \rightarrow_{\Re}^{\Gamma} \ldots \rightarrow_{\Re}^{\Gamma} t'$ with t' $\Re_{\Gamma}$-irreducible.

Note that we can assume, without loss of generality, that all left- and right-hand sides of CTRSs are linear (any non-linear constrained rewrite rule can be transformed into an equivalent linear one by simply adding to the rule's constraint the equalities between the corresponding variables).

Example 1. Given $\Sigma = (\mathcal{F}, \mathcal{P})$ with $\mathcal{F} = \{zero_{/0}, succ_{/1}, f_{/2}, g_{/2}\}$ and $\mathcal{P} = \{inf_{/2}\}$, $\Re = \{g(zero, v) \longrightarrow v; g(x, v) \xrightarrow{\psi(x,y)} succ(g(y, v))\}$ with $\psi(x, y) \triangleq inf(y, x) \wedge \forall z.inf(z, x) \Rightarrow (z = y \vee inf(z, y))$ is a CTRS over Σ. Let $\mathcal{N}$ be the interpretation whose carrier is $\mathbb{N}$ and such that $zero_{\mathcal{N}} = 0$, $succ_{\mathcal{N}} = n \mapsto n + 1$ and inf is interpreted by $<_{\mathbb{N}}$. Then, we have $g(g(succ(zero), zero), zero) \rightarrow_{\Re}^{\mathcal{N}} g(succ(g(zero, zero)), zero) \rightarrow_{\Re}^{\mathcal{N}} g(succ(zero), zero) \rightarrow_{\Re}^{\mathcal{N}} succ(g(zero, zero)) \rightarrow_{\Re}^{\mathcal{N}} succ(zero)$. This reduction is the only one starting from $g(g(succ(zero), zero), zero)$ since the interpretation of inf in $\mathcal{N}$ is not specified for terms containing g.

It is not surprising that when restricting the general constrained rewriting we obtain some well-known formalisms. For example, when restricting to signatures of the form $\Sigma = (\mathcal{F}, \varnothing)$ containing only the equality predicate and to constrained rewrite systems $\Re$ over Σ such that all constraints are $\top$, the $\Re$-rewriting relation over a Σ-theory Γ is exactly the relation $\rightarrow_{\Re/\Gamma}$ from rewriting modulo [14, 2].

Rewriting a ground term *w.r.t.* a (constrained) rewrite rule consists in finding the potential redexes, *i.e.* the terms that can be rewritten by the rule and then transforming them accordingly. The search for a redex strongly relies on syntactic matching in plain term rewriting and on matching modulo in some versions of rewriting modulo [18]. In the formalism presented here the search for a redex can be decomposed in several steps: (i) search for all terms equivalent to the initial one, (ii) search for the (positions of the) subterms in these terms which (syntactically) match the left-hand side of the rewrite rule and finally (iii) check if the rule's constraint correspondingly instantiated is satisfiable. We propose thus a definition of matching that takes into account the steps mentioned above.

Definition 2. *Given a signature Σ and a Σ-theory Γ, a* context-sensitive constrained matching problem *or CMP is a formula of the form $l \parallel \varphi \ll^?_\Gamma t$ with $l \in \mathcal{T}_{\mathcal{F},\mathcal{X}}$, $\varphi \in \mathcal{F}or_{\Sigma,\mathcal{X}}$ and $t \in \mathcal{T}_{\mathcal{F}}$. A pair $\langle\, C[\Box], \sigma \,\rangle$ where σ is a ground substitution such that $\mathcal{D}om(\sigma) = \mathcal{V}ar(l) \cup \mathcal{FV}ar(\varphi)$ is a* solution *of the context-sensitive matching problem $l \parallel \varphi \ll^?_\Gamma t$ iff $\Gamma \models C[\sigma(l)] = t$ and $\Gamma \models \sigma(\varphi)$. The set of solutions of a problem $l \parallel \varphi \ll^?_\Gamma t$ is denoted by $\mathcal{S}ol(l \parallel \varphi \ll^?_\Gamma t)$.*

We will show in Section 5 that the problem of solving matching problems whose formula is $\top$ is decidable when the underlying theory can be described by tree automata. We propose then a method for solving constraints in such theories and combine the two approaches to solve the global problem. We call these theories recognizable and we define them formally in what follows.

4 Recognizable theories

In [8] it was shown that recognizable sets play an important role in the computation in Horn clause theories. Indeed, they proposed a restriction of Horn clauses (called uniform clauses) insuring that the least model consists of recognizable sets of terms and thus can be represented by a tree automaton [17]. Moreover, it has been shown [6] that order-sorted signatures could be viewed as regular tree languages and in [9] a correspondence between Prolog monadic programs and tree automata was formalized. Starting from theses observations, we propose to describe some theories by tree automata. The description of a Σ-theory Γ could be obtained, when it is possible, by characterizing for each predicate p of Σ the set of tuples for which p is completely interpreted, *i.e.* the set $\mathcal{S}(p) \subseteq \mathcal{T}^n_{\mathcal{F}}$ such that for any $\langle\, t_1, \ldots, t_n \,\rangle \in \mathcal{S}(p)$ either $\Gamma \models p(t_1, \ldots, t_n)$ or $\Gamma \models \neg p(t_1, \ldots, t_n)$.

Definition 3 (Recognizable presentation, recognizable theory). *Given a signature $\Sigma = (\mathcal{F}, \mathcal{P})$ a* recognizable presentation *$\wp$ of a Σ-theory Γ consists of*

two applications $\mathcal{S}_\wp$ *and* $\mathcal{A}_\wp$ *associating to every* $p \in \mathcal{P}$ *a tree automaton respectively called* support *and* axiomatization *of* p *and a binary automaton* $\mathcal{E}_\wp$ *called* equivalence *such that for any* $t, t' \in \mathcal{T}_\mathcal{F}$, $\Gamma \models t = t'$ *iff* $\langle t, t' \rangle \in \mathcal{L}(\mathcal{E}_\wp)$ *and for any* $p_{/n} \in \mathcal{P}$, $n > 0$, $\langle t_1, \ldots, t_n \rangle \in \mathcal{T}_\mathcal{F}^n$: $\Gamma \models p(t_1, \ldots, t_n)$ *iff* $\langle t_1, \ldots, t_n \rangle \in [\![p]\!]_\wp$ *and* $\Gamma \models \neg p(t_1, \ldots, t_n)$ *iff* $\langle t_1, \ldots, t_n \rangle \in [\![\overline{p}]\!]_\wp$ *where* $[\![p]\!]_\wp$ *denotes the set* $\mathcal{L}(\mathcal{S}_\wp(p)) \cap \mathcal{L}(\mathcal{A}_\wp(p))$ *and* $[\![\overline{p}]\!]_\wp$ *the set* $\mathcal{L}(\mathcal{S}_\wp(p)) \setminus \mathcal{L}(\mathcal{A}_\wp(p))$. *The pair* $p_\wp = \langle \mathcal{S}_\wp(p), \mathcal{A}_\wp(p) \rangle$ *is called* presentation *of* p *in* $\wp$. *A* Σ*-theory admitting a recognizable presentation is said* recognizable.

For readability reasons, we use in what follows the same symbol to denote a predicate and its presentation. We also usually assume that $\mathcal{L}(\mathcal{A}_\wp(p)) \subseteq \mathcal{L}(\mathcal{S}_\wp(p))$. For an overview of recognizable relations we refer to [7] and [19] for example.

Example 2. We define an extension of the theory $\mathcal{N}$ defined in Example 1 by specifying that *zero* is an absorbing element for f. A presentation $\wp$ of $\mathcal{N}$ is given by an automaton $\mathcal{S}_\wp(inf)$ denoting all the pairs of terms equivalent to the ones containing no f, together with an automaton $\mathcal{A}_\wp(inf)$ recognizing pairs of terms pairwise equivalent to a pair $\langle t, t' \rangle$ of terms built without f such that t' contains more symbols *succ* than t, and an automaton $\mathcal{E}_\wp$ denoting pairs $\langle t, t' \rangle$ such that for any ω such that $t(\omega) = zero$, $t'(\omega)$ is either *zero* or of the form $f(zero, u)$ or $f(u, zero)$ and reciprocally (details of these automata are given in Appendix A). We have $\mathcal{N} \models inf(f(succ(zero), zero), succ(zero))$ but if $t = f(succ(zero), succ(zero))$ and $t' = succ(zero)$ then we have neither $\mathcal{N} \models inf(t, t')$ nor $\mathcal{N} \models \neg inf(t, t')$. Indeed, $\langle f(succ(zero), zero), succ(zero) \rangle \in [\![inf]\!]_\wp$ but $\langle t, t' \rangle \notin \mathcal{L}(\mathcal{S}_\wp(inf))$.

5 CMP resolution

We propose in what follows an approach for solving context-sensitive constrained matching problems *w.r.t.* a recognizable underlying theory. We first propose an algorithm for unconstrained problems and then we show how constraints can be solved in the same formalism.

The solution of a classical syntactic matching problem, when it exists, is unique and is computed by a simple recursive algorithm [12]. As opposed to syntactic matching, matching modulo an equational theory is undecidable as well as not unitary in general [4]. Since we are concerned here with recognizable theories, we can take advantage of the properties of recognizable relations in order to obtain the decidability of the corresponding matching problems for a large class of constraints. When the matching problems are not constrained, *i.e.* when the constraint is $\top$, we can build automata that allow the representation of all the solutions of a given problem. In particular, starting from the equivalence automaton of the presentation of the corresponding theory, we build (couples of) automata that characterize the context and the corresponding substitution for all the solutions of the matching problem. As a consequence we obtain a finite presentation of all the solutions of an (unconstrained) context-sensitive matching problem.

Proposition 1. *Given a ground term* t*, a pattern* l *and a recognizable theory* Γ*, the set* $\mathcal{S}ol\left(l \ll^?_\Gamma t\right)$ *is decidable.*

More precisely, we can give an algorithm (detailed in Appendix B) that produces a set denoted by $\mathcal{R}ec\left(l \ll^?_\Gamma t\right)$ of pairs consisting of an automaton and a mapping from variables to automata such that: $\langle\, C[\Box], \sigma \,\rangle \in \mathcal{S}ol\left(l \ll^?_\Gamma t\right)$ iff there exists $\langle\, \mathbb{A}_C, \mu \,\rangle \in \mathcal{R}ec\left(l \ll^?_\Gamma t\right)$ such that $C[\Box] \in \mathcal{L}(\mathbb{A}_C)$, and for any $x \in \mathcal{D}om(\sigma)$ we have $\sigma(x) \in \mathcal{L}(\mu(x))$.

We address now the problem of computing the solutions of a *constraint resolution problem* (CRP) of the form $\Gamma \models^? \varphi[x_1, \ldots, x_n]$ with $\varphi[x_1, \ldots, x_n] \in \mathcal{F}or_{\Sigma,\mathcal{X}}$ and more precisely we propose an algorithm that computes the tuples of ground terms $\langle\, t_1, \ldots, t_n \,\rangle$ such that $\Gamma \models \varphi[t_1, \ldots, t_n]$. In order to find the solutions of a constraint *w.r.t.* a recognizable theory we will compute, when possible, a presentation of the respective constraint, *i.e.* a pair consisting of a support and an axiomatization. These presentations are denoted by expressions built using the predicate symbols of the theory and some constructors and operators whose semantics are defined using the automata operations introduced in Section 2. We introduce first an additional operation characterized by the following lemma:

Lemma 1. *Given an n-ary tree automaton $\mathbb{A}$ over $\mathcal{F}$, for any n-tuple of function symbols $\langle\, f_1, \ldots, f_n \,\rangle$ of $\mathcal{F}^n$ and $0 < k \leq \max_{i\in[1,n]}(ar(f_i))$ there exists an automaton denoted by $\mathbb{C}hild_{\langle\, f_1,\ldots,f_n \,\rangle/k}(\mathbb{A})$ and recognizing the set of tuples $\langle\, t_1, \ldots, t_n \,\rangle$ such that there exists a tuple $\langle f_1(x_1^1, \ldots, x_1^{k-1}, t_1, x_1^{k+1}, \ldots, x_1^m), \ldots, f_n(x_n^1, \ldots, x_n^{k-1}, t_n, x_n^{k+1}, \ldots, x_n^m)\rangle \in \mathcal{L}(\mathbb{A})$.*

Proof. The automaton $\mathbb{C}hild_{\langle\, f_1,\ldots,f_n \,\rangle/k}(\mathbb{A})$ is obtained from $\mathbb{A}$ by replacing the set of final states Q_F of $\mathbb{A}$ by the set of states q_k such that $\langle\, f_1, \ldots, f_n \,\rangle(q_1, \ldots, q_n) \rightarrow q$ is a rule of $\mathbb{A}$ and $q \in Q_F$ and by subsequently reducing the obtained automaton (by removing the rules concerning useless states).

Definition 4 (Formula presentation). *Given a presentation $\wp$ of a recognizable theory, a formula presentation over $\wp$ is an expression built out of the predicate presentation symbols from $\mathcal{P}$ and the operators in Figure 1,which denotes a pair of automata (a support and an axiomatisation). Moreover, we say that a formula presentation ϑ encodes a Σ-formula φ w.r.t. Γ iff:*

$$\begin{cases} \Gamma \models \varphi[t_1, \ldots, t_n] \text{ iff } \langle\, t_1, \ldots, t_n \,\rangle \in [\![\vartheta]\!]_\wp \\ \Gamma \models \neg\varphi[t_1, \ldots, t_n] \text{ iff } \langle\, t_1, \ldots, t_n \,\rangle \in [\![\bar{\vartheta}]\!]_\wp \end{cases}$$

where $[\![\vartheta]\!]_\wp = \mathcal{L}(\mathcal{S}_\wp(\vartheta)) \cap \mathcal{L}(\mathcal{A}_\wp(\vartheta))$ and $[\![\bar{\vartheta}]\!]_\wp = \mathcal{L}(\mathcal{S}_\wp(\vartheta)) \setminus \mathcal{L}(\mathcal{A}_\wp(\vartheta))$. In this case, we write $\vartheta \approx_\Gamma \varphi$.

Notation	Support	Axiomatisation
$\top$	$\mathbb{T}$	$\mathbb{T}$
$\bot$	$\mathbb{T}$	$\mathbb{F}$
$\bowtie$	$\mathbb{F}$	$\mathbb{F}$
$p \sqcap q$	$(\mathcal{S}_\wp(p) \cap \mathcal{S}_\wp(q))$ $\cup(\mathcal{S}_\wp(p) \cap \overline{\mathcal{A}_\wp(p)}) \cup (\mathcal{S}_\wp(q) \cap \overline{\mathcal{A}_\wp(q)})$	$\mathcal{A}_\wp(p) \cap \mathcal{A}_\wp(q)$
$p \sqcup q$	$(\mathcal{S}_\wp(p) \cap \mathcal{S}_\wp(q))$ $\cup(\mathcal{S}_\wp(p) \cap \mathcal{A}_\wp(p)) \cup (\mathcal{S}_\wp(q) \cap \mathcal{A}_\wp(q))$	$\mathcal{A}_\wp(p) \cup \mathcal{A}_\wp(q)$

$\widetilde{p}$	$\mathcal{S}_\wp(p)$	$\overline{\mathcal{A}_\wp(p)}$
$\Omega^n_{\mathcal{F}}$	$\mathbb{U}niv^n_{\mathcal{F}}$	$\mathbb{U}niv^n_{\mathcal{F}}$
$Id^n_{\mathcal{F}}$	$\mathbb{U}niv^n_{\mathcal{F}}$	$\mathbb{I}dentity^n_{\mathcal{F}}$
$\overrightarrow{t}$	$\mathbb{U}niv^n_{\mathcal{F}}$	$rec(t)$
where t a term of $\mathcal{T}_{\mathcal{F}[\Theta]}$ where Θ is the formula presentations of arity 1.		
$\mathfrak{S}_{i,j}(p)$	$\mathbb{S}wap_{i,j}(\mathcal{S}_\wp(p))$	$\mathbb{S}wap_{i,j}(\mathcal{A}_\wp(p))$
$\Pi_i(p)$	$\mathbb{C}yl_i(\mathcal{S}_\wp(p))$	$\mathbb{C}yl_i(\mathcal{A}_\wp(p))$
$\Pi_i(p)$	$\mathbb{P}roj_i(\mathcal{S}_\wp(p))$	$\mathbb{P}roj_i(\mathcal{A}_\wp(p))$
$\Pi_{t/i}(p)$	$\mathbb{P}roj_{t/i}(\mathcal{S}_\wp(p))$	$\mathbb{P}roj_{t/i}(\mathcal{A}_\wp(p))$
$\partial_{\langle f_1,\ldots,f_n\rangle/i}(p)$	$\mathbb{C}hild_{\langle f_1,\ldots,f_n\rangle/i}(\mathcal{S}_\wp(p))$	$\mathbb{C}hild_{\langle f_1,\ldots,f_n\rangle/i}(\mathcal{A}_\wp(p))$
$Equiv$	$\mathbb{U}niv^2_{\mathcal{F}}$	$\mathcal{E}_\wp$

Fig. 1. Operators over presentations

Theorem 1. *Given a recognizable theory Γ and a formula φ containing no function symbols, we can compute a formula presentation encoding φ w.r.t. Γ.*

Proof. We can give a terminating set of rewrite rules (detailed in Appendix C) that, based on a presentation $\wp$ of a recognizable Σ-theory Γ, transforms a Σ-formula φ into a formula presentation ϑ which encodes φ *w.r.t.* Γ. Similarly to automata that manipulate "configurations" mixing terms and states, the rewrite rules handle expressions which combine formula and formula presentations. The transformation strongly relies on the semantics of the Π operator which allows the transformation of configurations of the form $\exists x.\vartheta(x_1,\ldots,x,\ldots,x_n)$ into presentations of the form $\Pi_i(\vartheta)(x_1,\ldots,x_n)$. The overall goal of the rules is to isolate this kind of formulae and to transform them accordingly. We should point out that the equality is handled as any binary predicate and that any non linear atom can be seen as a conjunction of a linear one and an equality. The transformation can yield different formula presentations (depending on the order the rules are applied) but they all have the same semantics.

Example 3. Let φ be the formula $inf(y,t) \wedge \forall z.(inf(z,t) \Rightarrow (z = y \vee inf(z,y)))$ with $t = succ(succ(succ(zero)))$. One can check that:

$$\vartheta = \left(\Pi_{2/t}(\inf) \sqcap \widetilde{\Pi}_1\left(\left(\Pi_2\ (\Pi_{2/t}(\inf)) \sqcap \widetilde{Id^2_{\mathcal{F}}} \sqcap \widetilde{\inf}\right)\right)\right) \ \approx_{\mathcal{N}} \varphi[y]$$

for $\mathcal{N}$ given in Example 2. Moreover, we can see that $[\![\vartheta]\!]_\wp = \{succ(succ(zero))\}$.

An immediate consequence is the fact that the set of solutions (*w.r.t.* a recognizable theory) of any first order formula containing no function symbols is a recognizable set. The algorithm computing a presentation formula which encodes a formula containing no function symbols cannot be extended to deal with any first order formula. This comes from the fact that recognizable relations are not stable by context, that is to say, if R is an n-ary recognizable relation then, the set $\{f(t_1,\ldots,t_n) \mid R(t_1,\ldots,t_n)\}$ is not recognizable in general. However given n recognizable sets R_i, the set $\{f(t_1,\ldots,t_n) \mid R_1(t_1) \wedge \ldots R_n(t_n)\}$ is recognizable. This remark together with Lemma 1 allows us to to identify a class of formulae for which we are able to compute a corresponding presentation formula.

Definition 5. *A* critical pair of variables *in an atom $p(t_1, \ldots, t_n)$ is a pair of variables $\langle x, y \rangle$ such that there exist two positions ω and ω' and two terms t_i and t_j such that $t_i(\omega) = x$, $t_j(\omega) = y$ and $\omega' >_{pref} \omega$.*

Example 4. The pair of variables $\langle x, y \rangle$ is critical in $p(f(x, b), g(g(y)))$ since x occurs at position 1 and y at position 1.1 while $q(f(x, b), g(z), f(a, g(y)))$ has no critical pair of variables.

Theorem 2. *If for any critical pair of variables $\langle x, y \rangle$ in an atom of a formula φ one of the variables x or y is quantified by a quantifier which has no other quantifier in its scope, and it occurs in no other non-monadic atom in φ, then we can compute a formula presentation encoding φ. A formula satisfying the conditions of the theorem is said* REC*-preserving.*

Proof. Without loss of generality, we assume that formulae are written in an antiprenex form, *i.e.* quantifiers are distributed *w.r.t.* logical connectors and their ranges are minimized, and follow the Barendregt's convention, *i.e.* there is no pair of quantified variables with the same name and no variable out of the scope of its quantifier. If φ contains no critical pair of variables, then we can apply Lemma 1 and replace the atoms containing terms by formula presentations containing no function symbols. In this case we can conclude as for Theorem 1. Otherwise, if any variable x in a critical pair of variables of an atom $p(t_1, \ldots, t_n)$ occurs in a subformula of the form $Qx.p(t_1, \ldots, t_n) \oplus_1 q_1(x) \oplus_2 \ldots \oplus_m q_m(x)$ where Q is a quantifier and $\oplus_j$ are boolean operators, then x represents a recognizable set and since it is not in relation with other variables, its occurrence in p can be replaced by the corresponding formula presentation. The rules in Figures 2 in Appendix C implement these transformations and together with the other rules allow the computation of a corresponding formula presentation.

In particular, from the above theorem follows the well-known result saying that any monadic formula has a recognizable set of solutions.

Example 5. The formula $\forall z \exists y.((p(f(x), y) \vee q(f(y))) \Rightarrow p(f(y), z))$ has for antiprenex form the formula $\exists y.(p(f(x), y) \vee q(f(y))) \Rightarrow \forall z \exists y'.p(f(y'), z)$. Since this latter formula is REC-preserving, we can compute the formula presentation $\partial_{f/1}\left(\Pi_2\left(\widetilde{p} \sqcap \Pi_1\left(\partial_{f/1}(\widetilde{q})\right)\right)\right) \sqcup \widetilde{\Pi_2}\widetilde{\Pi_1}\left(\Pi_2\left(\overrightarrow{f(\Omega^1_{\mathcal{F}})} \sqcap p\right)\right)$ which encodes it.

Since there exist algorithms resolving the context-sensitive matching and computing a presentation formula ϑ encoding a formula under certain conditions, we can now address the problem of computing $\mathcal{S}ol(l \,\|\, \varphi \ll^?_\Gamma t)$ given a recognizable theory Γ and a CMP $l \,\|\, \varphi \ll^?_\Gamma t$.

Proposition 2. *Given a ground term t, a pattern l, a recognizable theory Γ, and a* REC*-preserving formula φ, the set $\mathcal{S}ol(l \,\|\, \varphi \ll^?_\Gamma t)$ is decidable.*

More precisely, we start from the set $\mathcal{R}ec\left(l \ll^?_\Gamma t\right)$ denoting the decidable set $\mathcal{S}ol\left(l \ll^?_\Gamma t\right)$ and for each $\langle \mathbb{A}_C, \mu \rangle \in \mathcal{R}ec\left(l \ll^?_\Gamma t\right)$ we build the automaton $\mathbb{A}_\sigma$

recognizing n-tuples $\langle\, t_1, \ldots, t_n \,\rangle$ (where each position i corresponds to a variable $x_i \in \mathcal{V}ar(l) \cup \mathcal{FV}ar(\varphi)$) such that $t_k \in \mathcal{L}(\mu(x_k))$ for all positions k corresponding to the variables of $\mathcal{V}ar(l)$ and $\langle\, t_{j_1}, \ldots, t_{j_m} \,\rangle \in [\![\vartheta]\!]_{\wp}$ with $\langle\, j_1, \ldots, j_m \,\rangle$ the positions corresponding to the variables of $\mathcal{FV}ar(\varphi)$ and $\vartheta \not\approx_{\wp} \varphi$. Schematically, $\mathbb{A}_\sigma = \mathbb{C}yl_I\,(\mu(x_1) \times \ldots \times \mu(x_n)) \cap \mathbb{C}yl_J(\mathbb{A}_\vartheta)$ where I and J are such that the parameters of both automata are the same and in the same order and $\mathbb{A}_\vartheta$ the automaton recognizing $[\![\vartheta]\!]_{\wp}$. We obtain thus a set $\mathcal{R}ec\left(l \,\|\, \varphi \ll^?_\Gamma t\right)$ such that $\langle\, C[\Box], \sigma \,\rangle \in \mathcal{S}ol(l \,\|\, \varphi \ll^?_\Gamma t)$ iff for some $\langle\, \mathbb{A}_C, \mathbb{A}_\sigma \,\rangle \in \mathcal{R}ec\left(l \,\|\, \varphi \ll^?_\Gamma t\right)$ we have $C \in \mathcal{L}(\mathbb{A}_C)$ and $\sigma : x_i \mapsto t_i$ for $\langle\, t_1, \ldots, t_n \,\rangle \in \mathbb{A}_\sigma$.

6 One-step constrained rewriting

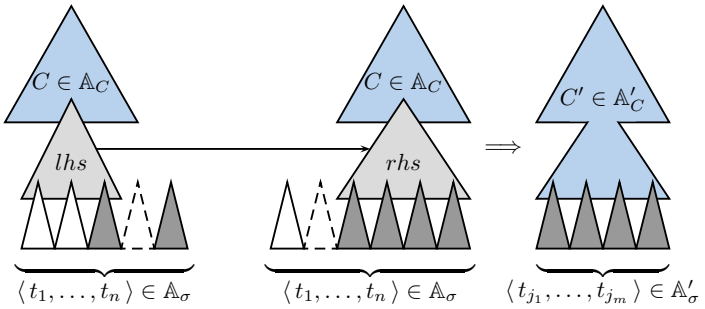
In this section we address the problem of computing the set of $\Re_\Gamma$-reducts of a ground term t, *i.e.* we want to compute the set of ground terms t' such that $t =_\Gamma t'' \to_{\mathcal{R}\|\Gamma} t'$, denoted by $\Re_\Gamma(t)$ in what follows. We need a representation of this set, allowing us to decide the membership but also the finiteness, the enumeration and the emptiness of the set of $\Re_\Gamma$-reducts of a term and thus, to decide if a term is in $\Re_\Gamma$-normal form *w.r.t* Γ and to be able to enumerate the reducts of a term. We show in what follows that the set of reducts is decidable and, in particular, we propose an appropriate representation for this set.

Definition 6 (Bi-recognizable reducts). *Given a CTRS $\Re$, a recognizable theory Γ and a ground term t, the set $\Re_\Gamma(t)$ is* bi-recognizable *iff there exists a finite set $Rec_\Gamma(t, \Re)$ of pairs $\langle\, \mathbb{A}_C, \mathbb{A}_\sigma \,\rangle$ consisting of two tree automata such that $t' \in \Re_\Gamma(t)$ iff there exists $\langle\, \mathbb{A}_C, \mathbb{A}_\sigma \,\rangle \in Rec_\Gamma(t, \Re)$ such that $t' = C[t_1, \ldots, t_n]$ for some $C[\Box, \ldots, \Box] \in \mathcal{L}(\mathbb{A}_C)$ and $\langle\, t_1, \ldots, t_n \,\rangle \in \mathcal{L}(\mathbb{A}_\sigma)$.*

The bi-recognizable representation of reducts corresponds to our expectations since all necessary properties are decidable.

Proposition 3. *The set of reducts of a term t w.r.t. a CTRS $\Re$ and a recognizable theory is bi-recognizable if for any rule $lhs \xrightarrow{\varphi} rhs \in \Re$, either φ is* REC*-preserving or the set $\{t' \mid t' =_\Gamma t\}$ is finite and $\sigma(\varphi)$ is* REC*-preserving for any ground substitution σ whose domain is $\mathcal{V}ar(lhs)$.*

Proof. Note that, as previously, since any argument position of the automata we handle corresponds to a variable, we assume that for any n-ary tree automaton $\mathbb{A}$ we dispose of a mapping from $[1, n]$ to $\mathcal{X}$ associating to any argument index of $\mathbb{A}$ the corresponding variable. We make a similar assumption for the holes of a context. If φ is REC-preserving, then $Rec_\Gamma(t, \Re)$ contains a pair $\langle\, \mathbb{A}'_C, \mathbb{A}'_\sigma \,\rangle$ for each $lhs \xrightarrow{\varphi} rhs \in \Re$ and $\langle\, \mathbb{A}_C, \mathbb{A}_\sigma \,\rangle \in \mathcal{R}ec\left(lhs \,\|\, \varphi \ll^?_\Gamma t\right)$ where $\mathbb{A}'_C$ is built from $\mathbb{A}_C$ by replacing each rule $\Box \to q$ by the rules of the automaton recognizing the context corresponding to rhs (that is to say in which variables are replaced by $\Box$) in which final states are renamed into q, and $\mathbb{A}'_\sigma$ is the projection of $\mathbb{A}_\sigma$ over its components associated to variables of rhs. The following picture describes the way we build $\langle\, \mathbb{A}'_C, \mathbb{A}'_\sigma \,\rangle$ from $\langle\, \mathbb{A}_C, \mathbb{A}_\sigma \,\rangle$:



We consider now the case when $\{t' \mid t' =_\Gamma t\}$ is finite. It is easy to see that, given a term t and a rule $lhs \xrightarrow{\varphi} rhs \in \Re$, we can find the terms u, the substitutions σ, and the positions $\omega \in \mathcal{P}os(t)$ such that $t = u[\sigma(lhs)]_\omega$. We can thus build the terms $C = u[\sigma(rhs)]_\omega$ with $Var(C) = \mathcal{FV}(\varphi) \setminus \mathcal{V}ar(rhs)$ and we can build the automata $rec(C)$ where the variables have been replaced by $\Box$ (and the corresponding mapping memorized). Since φ is REC-preserving, we can build the automaton $\mathbb{A}_\vartheta$ recognizing $[\![\vartheta]\!]_\wp$ with $\vartheta \approx_\wp \sigma(\varphi)$. If $Rec_\Gamma(t, lhs \xrightarrow{\varphi} rhs) = \{\langle rec(C), \mathbb{A}_\vartheta \rangle \mid u, \sigma, \omega$ s.t $t = u[\sigma(lhs)]_\omega$ and $C = u[\sigma(rhs)]_\omega$ and $\vartheta \approx_\wp \sigma(\varphi)\}$ then the set of $\Re_\Gamma$-reducts of t is represented by $Rec_\Gamma(t, \Re) = \bigcup_{\substack{t' =_\Gamma t \\ lhs \xrightarrow{\varphi} rhs \in R}} Rec_\Gamma(t', lhs \xrightarrow{\varphi} rhs)$.

In practice, we want to "perform" rewriting, that is to say to built a sequence of ground terms $t_0 \rightarrow \ \ldots \rightarrow t_i \rightarrow t_{i+1} \in \Re_\Gamma(t_i) \rightarrow \ldots$ until possibly obtaining a $\Re_\Gamma$-normal form. The bi-recognizable representation allows us to build such sequences by defining a strategy for choosing a term among the set of terms $\Re_\Gamma(t)$ for some t. We could use a strategy based on an order on terms like, for example, considering the number of symbols of the term or an order over symbols. This kind of strategies are easily implementable with a bi-recognizable representation and extend classical rewriting strategies which are usually based on the choice of the rewrite rule and on the redex-position at each step of the rewriting process.

Example 6. Let us consider again the CTRS $\Re$ defined in Example 1 together with the theory $\mathcal{N}_p$ whose carrier is $\mathbb{Z}/p\mathbb{Z}$ for some p, the equality relation begin the classical congruence $\equiv$ $(mod\ p)$ (which is recognizable). Given the term $t = g(zero, zero)$, the first rule is obviously applicable but also the second one since $\psi(zero, y)$ holds in $\mathcal{N}_p$ for any $y \equiv p - 1$ $(mod\ p)$. Thus, the set $Rec_{\mathcal{N}_p}(t, \Re)$ contains $\langle rec(\Box), \mathbb{A}_1 \rangle$ and $\langle rec(succ(g(\Box, \Box))), \mathbb{A}_2 \rangle$ where $\mathbb{A}_1$ recognizes terms of the form $succ^{k \cdot p}(zero)$ for some k and $\mathbb{A}_2$ recognizes pairs of the form $\left\langle succ^{(k+1)\cdot p-1}(zero), succ^{k' \cdot p}(zero) \right\rangle$ for some k and k' where $succ^m(zero)$ denotes the term built with m symbols $succ$ followed by $zero$.

7 Conclusion

We have proposed in this paper a general definition of constrained term rewriting by considering all first order formulae and we studied its decidability. We introduced an original characterization of a large class of theories using tree automata

and showed how constrained rewriting can be performed *w.r.t.* these so-called recognizable theories. We have shown that the potentially infinite set of solutions of a matching problem can be computed and finitely represented using automata. We have also characterized a large class of formulae for which the solutions can be finitely represented and showed how these results can be combined to obtain a finite representation of the corresponding constrained matching problems. This automata based representation of the matching solutions allows us to compute and finitely characterize the reducts *w.r.t.* constrained rewriting rules whose underlying matching problems fit in the computable class we have identified.

A library implementing the various tree automata operations, the matching algorithms and constrained rewriting is under development. The next step in this direction is the integration in the security policy analyser implementing the framework described in [3]. Furthermore, we investigate an extension of the notion of rewriting strategy compatible with the representation of reducts by tree automata.

References

1. F. Baader and T. Nipkow. *Term Rewriting and All That.* C.U. Press, 1998.
2. L. Bachmair and N. Dershowitz. Completion for rewriting modulo a congruence. *Theor. Comput. Sci. 67(2&3):173-201*, 1989.
3. T. Bourdier, H. Cirstea, M. Jaume, and H. Kirchner. On Formal Specification and Analysis of Security Policies. `http://hal.inria.fr/inria-00429240/en/`, 2009.
4. H.-J. Bürckert. Matching - A special case of unification? *J. Symb. Comput. 8(5):523-536*, 1989.
5. H. Comon. Completion of rewrite systems with membership constraints. In *ICALP*, LNCS 623:392-403, 1992.
6. H. Comon and C. Delor. Equational formulae with membership constraints. *Inf. and Comput. 112(2):167-216*, 1994.
7. H. Comon et al. Tree automata techniques and applications. November 2008.
8. T. Fruhwirth et al. Logic programs as types for logic programs. In *IEEE Symp. on Logic In Computer Science*, pages 300–309, 1991.
9. J. Goubault-Larrecq. The `h1` tool suite documentation. release Dec., 6th 2005.
10. P. Hinman. *Fundamentals of mathematical logic.* A.K. Peters, Ltd., 2005.
11. C. Hoot. Completion for constrained term rewriting systems. In *CTRS*, LNCS 656:408-423, 1992.
12. G. Huet. *Résolution d'equations dans les langages d'ordre 1,2, ..., ω.* PhD thesis, Université de Paris 7, 1976.
13. G. Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. In *J. ACM 27(4):797-821*, 1980.
14. J.-P. Jouannaud and H. Kirchner. Completion of a set of rules modulo a set of equations. *SIAM J. Comput. 15(4):1155-1194*, 1986.
15. C. Kirchner et al. Anti-pattern matching. In *ESOP*, LNCS 4421:110-124, 2007.
16. C. Kirchner and H. Kirchner. Constrained equational reasoning. In *ISSAC*, 1989.
17. F. Nieslon et al. Normalizable horn clauses, strongly recognizable relations and spi. In *SAS*, LNCS 2477:20-35, 2002.
18. G. Peterson and M. Stickel. Complete sets of reductions for some equational theories. *J. ACM 28(2):233-264*, 1981.
19. S. Tison. Tree automata and term rewrite systems. In *RTA*, LNCS 1833:27-30, 2000.

A Extended examples

Example 1 (contn'd). Given the signature $\Sigma = (\mathcal{F}, \mathcal{P})$ with $\mathcal{F} = \{zero_{/0}, succ_{/1}, f_{/2}, g_{/2}\}$ and $\mathcal{P} = \{inf_{/2}\}$, the following set of rules is a CTRS over Σ:

$$\mathcal{R}' = \left\{ f(x,y) \xrightarrow{\varphi(x,y)} x \qquad f(x,y) \xrightarrow{\varphi(y,x)} y \qquad f(x,x) \longrightarrow x \right\}$$

with $\varphi(x,y) \triangleq inf(x,y)$

The formula $\varphi(x,y)$ is obviously used to check that the first argument is smaller than the second one *w.r.t. inf*. Valid instantiations of the free variable y in the formula $\psi(x,y)$ represent the "supremums" of *inf w.r.t.* x (*i.e.* the bigger elements smaller than x). Notice that while all the variables of the constraints of the rules in $\mathcal{R}'$ are bound by the left-hand sides, this is not the case for $\mathcal{R}$ (page 6) where variables in constrains are not only free but also used in the right-hand side of the rule.

We denote, as before, by $\mathcal{N}$ the interpretation whose carrier is $\mathbb{N}$ and such that $zero_{\mathcal{N}} = 0$, $succ_{\mathcal{N}} = n \mapsto n+1$ and *inf* is interpreted by $<_{\mathbb{N}}$. We denote by $\mathcal{N}^{-1}$ the interpretation which differs from $\mathcal{N}$ only by the interpretation of *inf* by $>_{\mathbb{N}}$. Then, the term $f(succ(zero), zero)$ rewrites by $\mathcal{R}'$ into *zero w.r.t.* $\mathcal{N}$ and into $succ(zero)$ *w.r.t.* $\mathcal{N}^{-1}$.

One can see that g (page 6) behaves as the addition operator *w.r.t.* $\rightarrow_{\mathcal{R}}^{\mathcal{N}}$ and that we can easily obtain infinite $\rightarrow_{\mathcal{R}}^{\mathcal{N}^{-1}}$ reductions.

Example 2 (contn'd). We define an extension of the theory $\mathcal{N}$ defined in Example 1 by specifying that *zero* is an absorbing element for f. A (partial) presentation $\wp$ of $\mathcal{N}$ is given by: $\mathcal{S}_\wp(inf) = (\mathcal{F}^2, \{q_1, q_2, q_\mathcal{N}\}, \{q_\mathcal{N}\}, \Delta_1)$ where Δ_1 consists of:

$$\begin{array}{ccc} \langle zero, \Lambda \rangle \to q_1 & \langle succ, \Lambda \rangle (q_1) \to q_1 & \langle zero, zero \rangle \to q_\mathcal{N} \\ \langle \Lambda, zero \rangle \to q_2 & \langle \Lambda, succ \rangle (q_2) \to q_2 & \\ \langle succ, zero \rangle (q_1) \to q_\mathcal{N} & \langle zero, succ \rangle (q_2) \to q_\mathcal{N} & \langle succ, succ \rangle (q_\mathcal{N}) \to q_\mathcal{N} \end{array}$$

together with $\mathcal{A}_\wp(inf) = ((\mathcal{F} \cup \{\Lambda\})^2, \{q, q_{inf}\}, \{q_{inf}\}, \Delta_2)$ where Δ_2 consists of

$$\begin{array}{cc} \langle succ, succ \rangle (q_{inf}) \to q_{inf} & \langle zero, succ \rangle (q) \to q_{inf} \\ \langle \Lambda, succ \rangle (q) \to q & \langle \Lambda, zero \rangle \to q \end{array}$$

and $\mathcal{E}_\wp = ((\mathcal{F} \cup \{\Lambda\})^2, \{q_0, q, q_F, \}, \{q_F\}, \Delta_=)$ where $\Delta_=$ is given by:

$$\begin{array}{ccc} \langle f, zero \rangle (q_0, q) \to q_F & \langle f, zero \rangle (q, q_0) \to q_F & \langle zero, \Lambda \rangle \to q_0 \\ \langle f, \Lambda \rangle (q, q) \to q & \langle succ, \Lambda \rangle (q) \to q & \langle zero, \Lambda \rangle \to q \\ \langle f, f \rangle (q_F, q_F) \to q_F & \langle succ, succ \rangle (q_F) \to q_F & \langle zero, zero \rangle \to q_F \end{array}$$

For readability and conciseness reasons, $\mathcal{E}_\wp$ is not symmetric and $\mathcal{A}_\wp(inf)$ and $\mathcal{S}_\wp(inf)$ are not saturated *w.r.t.* $\mathcal{E}_\wp$ but both properties can be enforced, for example, by using the tree automata operations introduced in the Section 2. One can see that we have $\mathcal{N} \models inf(f(succ(zero), zero), succ(zero))$ but neither $\mathcal{N} \models inf(t, t')$ nor $\mathcal{N} \models \neg inf(t, t')$ with $t = f(succ(zero), succ(zero))$ and $t' = succ(zero)$. Indeed, $\langle f(succ(zero), zero), succ(zero) \rangle \in [\![inf]\!]_\wp$ but $\langle t, t' \rangle \notin \mathcal{L}(\mathcal{S}_\wp(inf))$.

B Unconstrained context-sensitive matching

The Algorithm 1 given below computes the solutions of an unconstrained CMP with a recognizable underlying theory. More precisely, given a ground term t, a pattern p and a recognizable theory Γ, the algorithm produces a set $\mathcal{R}ec\left(p \ll^?_\Gamma t\right)$ of pairs consisting of an automaton and a mapping from variables to automata such that: $\langle\, C[\Box], \sigma \,\rangle \in \mathcal{S}ol\left(p \ll^?_\Gamma t\right)$ iff there exists $\langle\, \mathbb{A}_C, \mu \,\rangle \in \mathcal{R}ec\left(p \ll^?_\Gamma t\right)$ such that $C[\Box] \in \mathcal{L}(\mathbb{A}_C)$, and for any $x \in \mathcal{D}om(\sigma)$ we have $\sigma(x) \in \mathcal{L}(\mu(x))$.

Algorithm 1 Matching modulo a recognizable theory

Input: a ground term t; a linear pattern p;
a recognizable theory Γ (and a presentation $\wp$ of Γ)
Output: a set $\mathcal{R}ec\left(p \ll^?_\Gamma t\right)$ of pairs $\langle\, \mathbb{A}_C, \mu \,\rangle$ where $\mathbb{A}_C$ is a tree automaton over $\mathcal{F} \cup \{\Box\}$ and μ a mapping from $\mathcal{V}ar(p)$ to tree automata

1: $\{\mathcal{F}, Q, Q_F, \Delta\} \leftarrow \mathbb{P}roj_{t/1}(\mathcal{E}_\wp)$
2: **for** $q \in Q$ **do**
3: $\quad\zeta \leftarrow visit(\varepsilon, q)$
4: $\quad$**if** $\zeta \neq \varnothing$ **then**
5: $\quad\quad\mathbb{A}_C \leftarrow$ automaton over $\mathcal{F} \cup \{\Box\}$ recognizing the intersection between $\{\mathcal{F} \cup \{\Box\}, Q, Q_F, \Delta \cup \{\Box \to q\}\}$ and the automaton recognizing terms containing exactly one $\Box$
6: $\quad\quad$**for each** $\mu \in \zeta$ **do**
7: $\quad\quad\quad\mathcal{R}ec\left(p \ll^?_\Gamma t\right) \leftarrow \mathcal{R}ec\left(p \ll^?_\Gamma t\right) \cup \{\langle\, \mathbb{A}_C, \mu \,\rangle\}$
8: $\quad\quad$**end for**
9: $\quad$**end if**
10: **end for**

11: **function** visit(position ω, state $prod$) **do**
12: $\quad symbol \leftarrow p(\omega)$
13: $\quad$**if** $symbol \in \mathcal{X}$ **then**
14: $\quad\quad$**return** $\{\{x \mapsto \{\mathcal{F}, Q, \{prod\}, \Delta\}\}\}$
15: $\quad$**else**
16: $\quad\quad\zeta \leftarrow \varnothing$
17: $\quad\quad$**for each** $symbol(q_1, \ldots, q_n) \to prod \ \in \Delta$ **do**
18: $\quad\quad\quad$**if** $\forall k \in [1, n].visit(\omega.k, q_k) \neq \varnothing$ **then**
19: $\quad\quad\quad\quad\zeta \leftarrow \zeta \cup \bigcup_{\mu_k \in visit(\omega.k, q_k)} (\mu_1 \uplus \ldots \uplus \mu_n)$
20: $\quad\quad\quad$**end if**
21: $\quad\quad$**end for**
22: $\quad\quad$**return** ζ
23: $\quad$**end if**
24: **end function**

where $\mu_1 \uplus \ldots \uplus \mu_n : x \mapsto \mu_i(x)$ *if* $x \in \mathcal{D}om(\mu_i)$ (in particular $\mathcal{D}om(\mu_1 \uplus \ldots \uplus \mu_n) = \varnothing$ if $n = 0$). Note that $\mathcal{D}om(\mu_i)$ and $\mathcal{D}om(\mu_j)$ are supposed to be disjoint for any $i \neq j$.

Example 3. Given the problem $\mathcal{S}ol\left(f(x, y) \ll^?_\Gamma succ(zero)\right)$, the first line of the algorithm computes the set of terms equivalent to $succ(zero)$ in Γ recognized by

$\{\mathcal{F}, \{q_0, q_1, q, q_F\}, \{q_F\}, \Delta\}$ with

$$\Delta = \begin{cases} succ(q_F) \to q_1 & succ(q_1) \to q_1 & succ(q_0) \to q_F & f(q_F, q_F) \to q_1 \\ f(q_0, q_1) \to q_0 & f(q_1, q_1) \to q_1 & f(q_1, q_0) \to q_0 & f(q_1, q_1) \to q_1 \\ f(q_0, q_0) \to q_0 & f(q_1, q_F) \to q_1 & f(q_F, q_0) \to q_0 & f(q_0, q_F) \to q_0 \\ zero \to q_0 & & & \end{cases}$$

The output of the algorithm is the set

$$\left\{ \left\langle \mathbb{A}_C, \begin{cases} x \mapsto \mathbb{A}_{q_0} \\ y \mapsto \mathbb{A}_{q_0} \cup \mathbb{A}_{q_1} \end{cases} \right\rangle, \left\langle \mathbb{A}_C, \begin{cases} y \mapsto \mathbb{A}_{q_0} \\ x \mapsto \mathbb{A}_{q_0} \cup \mathbb{A}_{q_1} \end{cases} \right\rangle, \left\langle \mathbb{A}_{C_0}, \begin{cases} x \mapsto \mathbb{A}_{q_0} \cup \mathbb{A}_{q_1} \\ y \mapsto \mathbb{A}_{q_0} \cup \mathbb{A}_{q_1} \end{cases} \right\rangle \right\}$$

where $\mathbb{A}_{q_i}$ is $\{\mathcal{F}, \{q_0, q_1, q, q_F\}, \{q_i\}, \Delta\}$, $\mathbb{A}_C$ denotes the set of terms whose head is *succ* and such that any symbol over the path between the root and the $\Box$ is an f and $\mathbb{A}_{C_0}$ denotes the set of terms whose head is *succ* and such that there is a path between the root and a *zero* composed only by f and for which $\Box$ can be at any position. For example, a solution is $\langle C, \sigma \rangle$ with $C = succ(f(succ(zero), \Box))$ and $\sigma : \begin{cases} x \mapsto f(zero, succ(zero)) \\ y \mapsto succ(zero) \end{cases}$.

C Constraint transformation

We propose in what follows a set of rewrite rules that, based on a presentation $\wp$ of a recognizable Σ-theory Γ, transforms a Σ-formula φ into a formula presentation ϑ which encodes φ *w.r.t.* Γ. Similarly to automata that manipulate "configurations" mixing terms and states, the rewrite rules handle expressions which combine formula and formula presentations. The transformation strongly relies on the semantics of the Π operator which allows the transformation of configurations of the form $\exists x.\vartheta(x_1, \ldots, x, \ldots, x_n)$ into presentations of the form $\Pi_i(\vartheta)(x_1, \ldots, x_n)$. The overall goal of the rules is to isolate this kind of formulae and to transform them accordingly.

First of all, since any first order formula has an equivalent prenex normal form [10], we presuppose that any formula of a CRP has been recast in prenex normal form.

The system is split into two parts, the first one handling formulas containing no function symbol (Figure 1) and the second one that handles formulas that may contain terms (Figure 2). Recall that, for readability reasons, we use the same symbol to denote a predicate and its presentation. We write $\alpha \to \alpha'$ to denote the one step transformation of the configuration α into α' and $\alpha \twoheadrightarrow \alpha'$ to denote that α' is obtained after an arbitrary number of steps. We split the presentation according to the intended use of each of the corresponding sets of rules and we show that each of the rules preserves the semantics.

We first isolate existentially quantified subformulae and "solve" them according to the semantics of the operator Π. Since existential quantification is distributive over disjunction but not under conjunction, the real scope of an existentially quantified variable can always be expressed with a conjunction of literals. The

(N_1) $(Q_i x_i)_{i=1}^q.\exists y. \bigwedge_{i=1}^{n} \bigvee_{j=1}^{m_i} a_{i,j} \to_{\mathcal{X}} (Q_i x_i)_{i=1}^q.\exists y. \bigvee_{(i_1,\ldots,i_n)\in\prod_{k=1}^n [1,m_k]} \bigwedge_{j=1}^{n} a_{j,i_j}$

(N_2) $(Q_i x_i)_{i=1}^q.\forall y. \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} a_{i,j} \to_{\mathcal{X}} (Q_i x_i)_{i=1}^q.\forall y. \bigwedge_{(i_1,\ldots,i_n)\in\prod_{k=1}^n [1,m_k]} \bigvee_{j=1}^{n} a_{j,i_j}$

(Q_1) $(Q_i x_i)_{i=1}^q.\exists y. \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} a_{i,j} \to_{\mathcal{X}} (Q_i x_i)_{i=1}^q. \bigvee_{i=1}^{n} \exists y. \bigwedge_{j=1}^{m_i} a_{i,j}$

(Q_2) $(Q_i x_i)_{i=1}^q.\forall y. \bigwedge_{i=1}^{n} \bigvee_{j=1}^{m_i} a_{i,j} \to_{\mathcal{X}} (Q_i x_i)_{i=1}^q. \bigwedge_{i=1}^{n} \forall y. \bigvee_{j=1}^{m_i} a_{i,j}$

(U_1) $\forall y. \bigvee_{j=1}^{m} a_j \to_{\mathcal{X}} \neg\exists y. \bigwedge_{j=1}^{m} \neg a_j$ *if y occurs in at least one a_j*

(U_2) $\forall y. \bigvee_{j=1}^{m} a_j \to_{\mathcal{X}} \bigvee_{j=1}^{m} a_j$ *if y occurs in no a_j*

(E) $\exists y. \bigwedge_{j=1}^{m} a_j \to_{\mathcal{X}} \bigwedge_{j=1}^{m} a_j$ *if y occurs in no a_j*

(P) $\exists y.(\exists y_k)_{k=1}^m.p(x_1,\ldots,x_{i-1},y,x_{i+1},\ldots,x_n) \wedge \psi \to_{\mathcal{X}} (\exists y_k)_{k=1}^m.\Pi_i(p)(x_1,\ldots,x_n) \wedge \psi$
if $y \notin \{x_1,\ldots,x_n\} \cup \mathcal{FV}ar(\psi)$

(C) $\neg p(t_1,\ldots,t_n) \to_{\mathcal{X}} \widetilde{p}(t_1,\ldots,t_n)$

(M_1) $p(x_1,\ldots,x_n) \wedge q(x_1,\ldots,x_n) \to_{\mathcal{X}} (p \sqcap q)(x_1,\ldots,x_n)$
(M_2) $p(x_1,\ldots,x_n) \vee q(x_1,\ldots,x_n) \to_{\mathcal{X}} (p \sqcup q)(x_1,\ldots,x_n)$

(O_1) $p(x_1,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_n) \to_{\mathcal{X}} \mathfrak{S}_{i,i+1}(p)(x_1,\ldots,x_{i-1},x_{i+1},x_i,\ldots,x_n)$
if $x_i > x_{i+1}$

(G_1) $p(x_1,\ldots,x_n) \wedge q(y_1,\ldots,y_m) \to_{\mathcal{X}} \Pi^k(p)(x_1,\ldots,x_{k-1},y,x_k,\ldots,x_n) \wedge q(y_1,\ldots,y_m)$
if $\{x_1,\ldots,x_n\} \cap \{y_1,\ldots,y_m\} \neq \varnothing$
and $y \in \{y_1,\ldots,y_m\} \setminus \{x_1,\ldots,x_n\}$ and $x_{k-1} < y < x_k$

(G_2) $p(x_1,\ldots,x_n) \wedge \psi \wedge q(y_1,\ldots,y_m) \to_{\mathcal{X}} \Pi^k(p)(x_1,\ldots,x_{k-1},y,x_k,\ldots,x_n) \wedge q(y_1,\ldots,y_m) \wedge \psi$
if $\{x_1,\ldots,x_n\} \cap \{y_1,\ldots,y_m\} = \varnothing$
and $y \in \{y_1,\ldots,y_m\}$ and $x_{k-1} < y < x_k$
and ψ contains no equality

(G_3) $p(x_1,\ldots,x_n) \vee q(y_1,\ldots,y_m) \to_{\mathcal{X}} \Pi^k(p)(x_1,\ldots,x_{k-1},y,x_k,\ldots,x_n) \vee q(y_1,\ldots,y_m)$
if $y \in \{y_1,\ldots,y_m\} \setminus \{x_1,\ldots,x_n\}$ and $x_{k-1} < y < x_k$

(L_1) $p(y_1,\ldots,y,\ldots,y,\ldots,y_n) \to_{\mathcal{X}} \Pi^j(p \sqcap \Pi_{[1,n]\setminus\{i,j\}}(Id^2_{\mathcal{F}}))(y_1,\ldots,y,\ldots,y_n)$
if y occurs in positions i and j in $p(\ldots,y,\ldots,y,\ldots)$

(T_1) $t = t' \to_{\mathcal{X}} Equiv(t,t')$

Fig. 1. Transformation rules for function-free formulae.

normalization of a quantified formula *w.r.t.* the sets of rules **N**, **Q** and **U** isolates subformulae of the form $\exists y.\bigwedge_{j=1}^{m} a_j$.

A configuration $\exists y.\varphi$ can be solved if either φ contains no occurrence of y or if φ is a formula presentation $\vartheta(x_1,\ldots,y,\ldots,x_n)$; the corresponding transformations are performed by the sets of rules **E** and **P**.

The existentially quantified configurations that do not have this solvable shape can be transformed by the sets of rules **C** and **M** which correspond to the semantics of the operators associated to boolean operations (conjunction, disjunction and complementation).

The previous transformations are performed on configurations which have the same parameters but this condition is not always satisfied in arbitrary formulae. We use thus the set of rules **O** to rearrange the parameters of atoms so that all variables of the atoms which must be combined respect a certain order, and the set of rules **G** to add parameters to atoms so that their number is the same for all atoms which must be combined.

When an atom contains several occurrences of the same variable, the rule (P) cannot be applied. This kind of atoms are decomposed using the linearization rule (L). We should point out that, since the presentation of any predicate is saturated *w.r.t.* the appropriate equivalence, in this latter rule we could use the identity instead of the equivalence operator.

Equalities are solved by the rule (T) which transforms them using the appropriate equivalence. Since there are no function symbols in formulae so far, we could have used x, x' instead of t, t' but the same rule will be used in its full generality later on.

Proposition 4. *Given a Σ-formula φ and a recognizable theory Γ together with a presentation $\wp$ of Γ, for any formula $\varphi[x_1,\ldots,x_n]$ and formula presentation ϑ such that $\varphi \twoheadrightarrow_{\mathcal{X}} \vartheta$, we have $\vartheta \models\!\!\!\approx_{\Gamma} \varphi$.*

Proof. Rules **N**, **Q**, **U** and **E** describe classical equivalences over first order formulae and $\vartheta \models\!\!\!\approx_{\Gamma} \varphi$ iff $\vartheta \models\!\!\!\approx_{\Gamma} \varphi'$ and $\varphi \twoheadrightarrow_{\mathbf{N},\mathbf{Q},\mathbf{U},\mathbf{E}} \varphi'$. The semantics preservation by rules **P**, **C**, **M**, **O**, **G**, **L** and **T** follows from the definitions of the operators over presentations used in right-hand sides of these rules. Indeed, if $\vartheta_1 \models\!\!\!\approx_{\Gamma} \varphi_1[x_1,\ldots,x_n]$ and $\vartheta_2 \models\!\!\!\approx_{\Gamma} \varphi_2[x_1,\ldots,x_n]$, then:

- $\vartheta_1 \sqcap \vartheta_2 \models\!\!\!\approx_{\Gamma} \varphi_1[x_1,\ldots,x_n] \wedge \varphi_1[x_1,\ldots,x_n]$
- $\vartheta_1 \sqcup \vartheta_2 \models\!\!\!\approx_{\Gamma} \varphi_1[x_1,\ldots,x_n] \vee \varphi_1[x_1,\ldots,x_n]$
- $\widetilde{\vartheta}_1 \models\!\!\!\approx_{\Gamma} \neg\varphi_1[x_1,\ldots,x_n]$
- $\mathfrak{S}_{i,j}(\vartheta_1) \models\!\!\!\approx_{\Gamma} \varphi_1[x_1,\ldots,x_j,\ldots,x_i,\ldots,x_n]$
- $\mathit{\Pi}_i(\vartheta_1) \models\!\!\!\approx_{\Gamma} \exists x_i.\varphi_1[x_1,\ldots,x_j,\ldots,x_i,\ldots,x_n]$ knowing that $\exists y_1 \ldots \exists y_n.\varphi$ is logicaly equivalent to $\exists y_{\kappa(1)} \ldots \exists y_{\kappa(n)}.\varphi$ for any permutation κ and that $\exists y.\varphi_1 \wedge \varphi_2$ is not equivalent to $\exists y.\varphi_1 \wedge \exists y.\varphi_2$ (which justifies the $y \notin \psi$ in **P**)
- $\mathit{\Pi}_i(\vartheta)[x_1,\ldots,y,\ldots,x_n] \models\!\!\!\approx_{\Gamma} \varphi_1[x_1,\ldots,x_n] \wedge y = y$

The transformation $\twoheadrightarrow_{\mathcal{X}}$ not only preserves the semantics of formulae but also eventually leads to a solved form:

Lemma 2. *Any first order formula containing no function symbols is rewritten w.r.t.* $\twoheadrightarrow_{\mathcal{X}}$ *into a formula presentation.*

Proof. We can prove that the relation induced by the transformation rules is strongly terminating by using the lexicographic product of two orders. The first order is a measure (the number of predicate symbols) on terms such that the size of the right-hand side is smaller than that of the left-hand side for all the rules except **N**, **Q** and **U**, for which is equal. The second order is based on a polynomial interpretation which decreases on all the other rules.

Moreover, the normal forms are formula presentations. First, note that any formula in prenex form φ of the general shape: $(Q_i x_i)_{i=1}^{m}.\psi$ where $(Q_i x_i)_{i=1}^{m}$ is seen as a "waiting line" in which any quantifier will be successively "resolved" (*i.e.* removed). Whatever the last quantifier Q_m is, the formula can be rewritten by the rules **N**, **Q** and **U** into a formula $(Q_i x_i)_{i=1}^{m-1}.((\exists x_m.\bigwedge_j a_j^1)\bigvee\ldots\bigvee(\exists x_m.\bigwedge_j a_j^q))$ or in one of the form $(Q_i x_i)_{i=1}^{m-1}.((\neg\exists x_m.\bigwedge_j \neg a_j^1)\bigwedge\ldots\bigwedge(\neg\exists x_m.\bigwedge_j \neg a_j^q))$ where a_j^k are literals. Each of the existentially quantified subformulae can be resolved independently of the others as follows:

1. any non-linear literal can be transformed into a linear one (containing no equalities) using the rules **L** and **T**;
2. any literal can be transformed into an atom using **C** and we obtain thus a conjunction containing only linear atoms;
3. this kind of conjunctions can be transformed by the rules **O** and **G** into a conjunction whose atoms have all the same parameters and in the same order: $\exists x_m.\bigwedge_j p_j^k(y_1,\ldots,y_n)$;
4. this conjunction can be transformed by **M** into an atom: $\exists x_m.p^k(y_1,\ldots,y_n)$;
5. the rules in **P** or **E** can then be repeatedly applied to obtain an atom without quantifier: $p_k^*(z_1,\ldots,z_{n-1})$.

Proposition 4 together with the Lemma 2 lead to the following theorem:

Theorem 1. *Given a recognizable theory* Γ *and a formula* φ *containing no function symbols, we can compute a formula presentation encoding* φ *w.r.t.* Γ.

We lift now the restriction on the presence of function symbols and show that we can we can obtain a result similar to Theorem 1 but only when some conditions are imposed on the corresponding formulae. In particular, if for all critical pairs of variables $\langle x, y\rangle$ in an atom of a formula φ, one of the variables x or y is quantified by a quantifier which has no other quantifier in its scope and it occurs in no other non-monadic atom in φ, then the formula is said REC-preserving.

We consider the rules presented in Figure 2 together with the rules in Figure 1 with the rule (T) applied now on terms instead of simple variables. We denote by $\rightarrow_{\mathcal{FX}}$ the relation $\rightarrow_{\mathcal{X}}\cup\rightarrow_{\mathcal{F}}$.

The rule (L_2) performs term linearization. The decomposition rule (D) takes advantage of Lemma 1 and reduces a relation over n terms into a conjunction of relations over their direct subterms. Recognizable relations allow us to deal only with configurations of the shape $\exists x.p(t_1,\ldots,x,\ldots,t_n)$ where each t_i is either a

$$(L_2)\ p(\ldots, f(\ldots, y, \ldots, y, \ldots), \ldots) \rightarrow_{\mathcal{F}} \quad \exists z.p(\ldots, f(\ldots, y, \ldots, z, \ldots), \ldots) \wedge Id^2_{\mathcal{F}}(y, z)$$

$$(D)\ p(f_1(t^1_1, \ldots, t^{m_1}_1), \ldots, f_n(t^1_n, \ldots, t^{m_n}_n)) \rightarrow_{\mathcal{F}} \quad \bigwedge_{i=1}^{\max_k(m_k)} \partial_{\langle f_1, \ldots, f_n \rangle / i}(p)(t^1_i, \ldots, t^m_i)$$

$$(R)\ p(t_1, \ldots, t_{i-1}, t, t_{i+1}, \ldots, t_n) \rightarrow_{\mathcal{F}} \quad \Pi_{t/i}(p)(t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n) \quad \text{if } t \in \mathcal{T}_{\mathcal{F}}$$

$$\begin{array}{lll}
(T_2)\ p(\ldots, t_{i-1}, t, t_{i+1}, \ldots) & \rightarrow_{\mathcal{F}} & \Pi_i(p \sqcap \Pi_{[1,n]\setminus\{i\}}(\overrightarrow{t}))(\ldots, t_{i-1}, t_{i+1}, \ldots) \quad \text{if } t \in \mathcal{T}_{\mathcal{F}[\Theta]} \\
(T_3)\ \exists y.p(\ldots, t, \ldots) \wedge \psi & \rightarrow_{\mathcal{F}} & p(\ldots, \sigma(t), \ldots) \wedge \psi \\
 & & \text{if } y \notin \mathcal{FV}ar(\psi),\ \sigma = \{y \mapsto \Omega_{\mathcal{F}}\} \\
 & & \quad \text{and } y \text{ occurs exactly once in } t \\
(T_4)\ \exists y.p(\ldots, t, \ldots) \wedge q(y) \wedge \psi & \rightarrow_{\mathcal{F}} & p(\ldots, \sigma(t), \ldots) \wedge \psi \\
 & & \text{if } y \notin \mathcal{FV}ar(\psi),\ \sigma = \{y \mapsto p\} \\
 & & \quad \text{and } y \text{ occurs exactly once in } t
\end{array}$$

Fig. 2. Transformation rules dealing with function symbols.

variable (rule (P)) or a ground term (rule (R)). A a configuration of the form $p(t_1, \ldots, t_n)$ where at least one t_i is a term containing a variable must be reduced until we obtain a conjunction of configurations having the shape we mentioned just above. The **T** rules capture the cases of the variables which represent a regular set of terms and use the fact that regular sets can be integrated into relational constraints.

Proposition 5. *Given a Σ-formula φ and a recognizable theory Γ together with a presentation $\wp$ of Γ, for any formula $\varphi[x_1, \ldots, x_n]$ and formula presentation ϑ such that $\varphi \twoheadrightarrow_{\mathcal{FX}} \vartheta$, we have $\vartheta \models\!\!\approx_{\Gamma} \varphi$.*

Proof. We can use the same arguments as in the proof Proposition 4 and the characterization given by Lemma 1.

As before, the transformation $\twoheadrightarrow_{\mathcal{FX}}$ eventually leads to a solved form:

Lemma 3. *Any first order* REC*-preserving formula is rewritten w.r.t. $\twoheadrightarrow_{\mathcal{FX}}$ into a formula presentation.*

Proof. For the termination proof we adapt the measure used in Lemma 2 to take into account the decreasing number of existential quantifiers and non-linear terms (with a bigger weight for the latter). The REC-preserving preserving formula are exactly the ones that can be handled by the rules in Figure 2 when the previous rules are not applicable and thus, we eventually obtain formula presentations.

Proposition 5 together with the Lemma 3 lead to the following theorem:

Theorem 2. *Given a recognizable theory Γ and a* REC*-preserving formula φ, we can compute a formula presentation encoding φ w.r.t. Γ.*