On Unification of Terms with Integer Exponents

Hubert Comon

CNRS and LRI, Bat. 490,
Université de Paris Sud,
91405 ORSAY cedex, France.
E-mail: comon@lri.lri.fr

November 27, 1992

1 Introduction

H. Chen and J. Hsiang proposed in [1] a unification algorithm for what they called ω-terms, and later ρ-terms (we keep this last terminology), with intended applications to logic programming. These terms allow to iterate terms with one hole along fixed paths. The number of iterations is part of the syntax of the terms and may include integer variables.

Example 1.1
Let the alphabet of functions symbols be f (binary), g (unary) and a (constant). Let N be an integer variable. Then, in Chen and Hsiang's formalism, $\Phi(f(\diamond, a), N, g(a))$ is a typical ρ-term whose instances (obtained by replacing N with an actual non-negative integer) are $g(a), f(g(a), a), f(f(g(a), a), a), \ldots$. The term $f(\diamond, a)$ has one "hole" (the $\diamond$) and its iteration along the path 1 (which is the position of the hole) is allowed.

Such constructions can be useful in expressing infinite sets of terms which occur in logic programming (see [1]) or in the Knuth-Bendix completion procedure (see [2]). In both cases, the ability to express iterated terms can prevent the non termination of the deduction process. Other applications are currently under investigation, for example in unification theory (see [4]).

Constructions involving more than one occurrence of an integer variable such as

$$f(\Phi(f(\diamond, a), N, g(a)), \Phi(f(\diamond, a), N, f(a, a)))$$

are also allowed in [1] which shows that the ρ-terms can schematize non regular sets of ground terms and hence they are not encompassed by the study of terms with context variables investigated in [3]. On the other hand, ρ-terms do not have the power of regular languages; there are two restrictions in these ρ-terms: nested ρ-terms are forbidden and the iterated part should not itself contain ρ-terms. Finally, in [1], the iterated parts and the terms in the holes should not contain variables.

Example 1.2 (see also figure 1)

$\Phi(f(\diamond, a), N, \Phi(g(\diamond), M, a))$ is not a ρ-term. (Nested ρ-terms are forbidden)

$\Phi(f(\diamond, \Phi(f(\diamond, a), N, a)), M, g(a))$ is not a ρ-term (the iterated part itself should not contain a ρ-term).

$\Phi(f(x,\diamond),N,g(a))$ is not a ρ-term (variables are not allowed in the iterated part)

$\Phi(f(a,\diamond),N,g(x))$ is not a ρ-term (variables are not allowed "below" the iterated part, i.e. in the third position of the Φ construction).

In this note we give another unification algorithm where these restrictions are dropped. We keep however the condition that, along an iterated path there should not occur any ρ-term. (i.e., for example, $\Phi(\Phi(f(\diamond,a),N,g(\diamond)),M,g(a))$ is still not allowed. This means that, if the iterated part contains itself iterated parts, the two iterated paths (or if one prefers, the two positions of the hole holder) should be uncomparable w.r.t. the prefix ordering (as in the right-up example of figure 1). Indeed, without this restriction, unification becomes undecidable.

2 Syntax and Interpretation of Formulae

2.1 Terms

Missing definitions can be found in [5]. We assume that F is a (finite or infinite) set of function symbols together with the arity function a. F is assumed to contain at least one constant (i.e. a symbol of arity 0). X is an infinite set of constants (disjoint from F) called *variables.* $\Diamond$ is a special symbol of arity 0 (the hole holder). $V_{\mathbf{N}}$ is a fixed set of symbols denoting integer variables. The set of *I-terms* T (also called "terms" for short) and the set of *terms with one hole* T_1 (also called "contexts") are the least sets that satisfy:

$$\begin{cases} f(\vec{s}) \in T & \Leftarrow & \vec{s} \in T^{a(f)} \\ x \in T & \Leftarrow & x \in X \\ \Diamond \in T_1 & & \\ f(\vec{s_1}, s, \vec{s_2}) \in T_1 & \Leftarrow & (\vec{s_1}, s, \vec{s_2}) \in T^{n_1} \times T_1 \times T^{n_2}, n_1 + 1 + n_2 = a(f) \\ s^N \cdot t \in T & \Leftarrow & s \in T_1, t \in T, N \in V_{\mathbf{N}}, s \neq \Diamond \end{cases}$$

The construction $s^N \cdot t \in T_1 \Leftarrow s \in T_1, t \in T_1, N \in V_N$ is not allowed here. But, as we will see in section 7, this additional construction does not increase the expressive power. We also define $Pos(t)$, the set of *positions* of a term $t \in T \cup T_1$ as follows:

$$\begin{cases} Pos(x) & \stackrel{\text{def}}{=} & \{\Lambda\} \\ Pos(\Diamond) & \stackrel{\text{def}}{=} & \{\Lambda\} \\ Pos(f(\vec{s})) & \stackrel{\text{def}}{=} & \{\Lambda\} \cup 1 \cdot Pos(s_1) \cup \ldots \cup a(f) \cdot Pos(s_{a(f)}) \\ Pos(s^N \cdot t) & \stackrel{\text{def}}{=} & \{\Lambda\} \end{cases}$$

We often omit the $\cdot$ when concatenating the positions ($\cdot$ is overloaded). Positions are (partially) ordered with the prefix ordering $\geq_{pref}$. If p and q are uncomparable positions, we write $p \parallel q$. In the above definition, $i \cdot Pos(s_i)$ stands for the set $\{i \cdot p \mid p \in Pos(s_i)\}$. If $p \in Pos(t)$, $t|_p$ is (as usual) the subterm at position p. $t(p)$ is the label (function symbol) at position p and $t[u]_p$ the term obtained by replacing $t|_p$ with u at position p. We will often use the notations $s[\Diamond]_p, s[\Diamond]_p^N \cdot t$ to indicate the position at which $\Diamond$ occurs in s: it follows from the definitions of T_1 and Pos that, for every term $s \in T_1$ which is not $\Diamond$ itself, there is exactly one $p \in Pos(s)$ ($p \neq \Lambda$) such that $s|_p = \Diamond$.

Extensions of this syntax (keeping the same expressive power) are described in section 7. In particular, contexts with multiple holes will be considered.

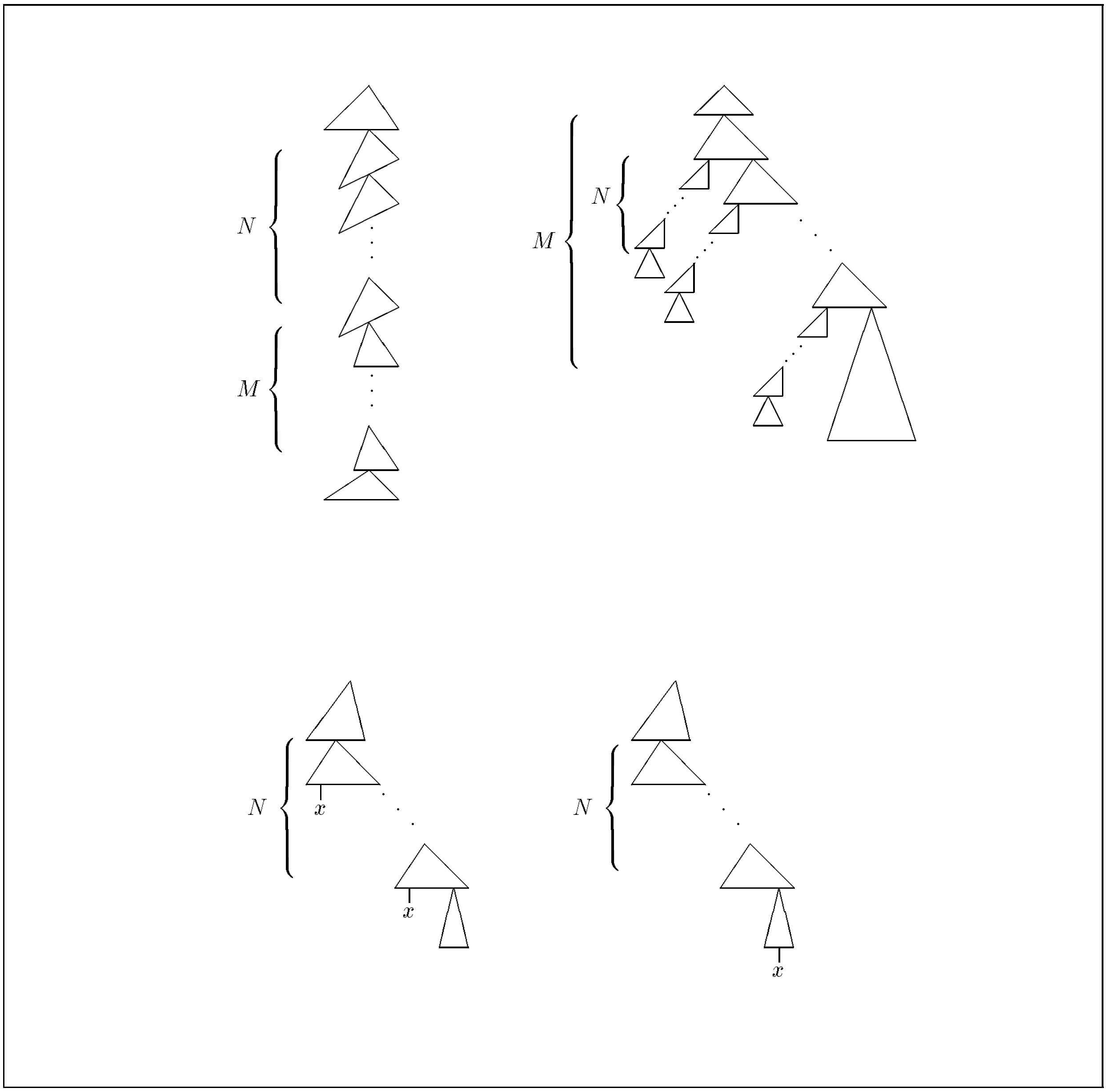


Figure 1: Examples of terms in T which are not ρ-terms

Example 2.1
We use the same alphabet as in example 1.1. Assuming $x, y \in X$, $N, M, Q \in V_{\mathbf{N}}$, the following expressions are I-terms:

$t_1 \equiv f(x, f(x, \diamond))^N \cdot f(\diamond, a)^M \cdot g(x)$ (this corresponds to I-terms with nested exponents)

$t_2 \equiv f(f(y, \diamond)^Q \cdot a, \diamond)^Q \cdot y$ (this corresponds to I-terms where the iterated part itself contains integer exponents).

Of course the intended meaning of the construction $s[\diamond]_p^N$ (which is defined precisely below) is to iterate the context $s[]_p$ (i.e. the term s in which the subterm at position p has been erased) N times.

The identity of I-terms is denoted $\equiv$ (in order to avoid confusion with $=$ which is used for equations).

2.2 Equations and Formulae

We consider two kinds of equations: ordinary equations of the form $s = t$ where $s, t \in T$ ($=$ is assumed symmetric: there is no difference between $s = t$ and $t = s$) and linear diophantine equations over a set of variables $V_{\mathbf{N}}$. If ψ is a conjunction of ordinary equations, then $IV(\psi)$ (*integer variables* of ψ) is the set of symbols of $V_{\mathbf{N}}$ which occur in some I-term of ψ. On the other hand $Var(\psi)$ is the set of free variables $x \in X$ which occur in ψ.

Unification formulae are disjunctions of formulae of the form $\exists \vec{n} : \phi \wedge \psi$ where $\vec{n}$ is a finite subset of $V_{\mathbf{N}}$, ψ is a conjunction of ordinary equations (called the *pure part* of the conjunction) and ϕ is a conjunction of linear diophantine equations whose variables is a subset of $\vec{n} \cup IV(\psi)$ (ϕ is called the *diophantine part* of the conjunction).

2.3 Substitutions and Solutions

Substitutions of ordinary variables (i.e. elements of X) by I-terms are defined in the usual way: a substitution σ is defined by a finite set of pairs (variable,term) written $\{x_1 \mapsto t_1;\ \ldots; x_n \mapsto t_n\}$. Its application to $t \in T$ is defined by:

- $f(s_1, \ldots, s_n)\sigma \stackrel{\text{def}}{=} f(s_1\sigma, \ldots, s_n\sigma)$
- $x_i\{x_1 \mapsto t_1;\ \ldots; x_n \mapsto t_n\} \stackrel{\text{def}}{=} t_i$ and $x\{x_1 \mapsto t_1;\ \ldots; x_n \mapsto t_n\} \stackrel{\text{def}}{=} x$ if x is not one of the x_i's.
- $s[\diamond]_p^N \cdot t\sigma \stackrel{\text{def}}{=} s\sigma[\diamond]_p^N \cdot t\sigma$

Note that this makes sense since, by induction on the structure of the terms,

$$(p \in Pos(s) \wedge s|_p \equiv \diamond) \Rightarrow (p \in Pos(s\sigma) \wedge s\sigma|_p \equiv \diamond)$$

If σ is a mapping which assigns a non-negative integer to each variable belonging to $IV(s = t)$, then $s\sigma$ and $t\sigma$ are defined by:

- $f(\vec{s})\sigma \stackrel{\text{def}}{=} f(\vec{s}\sigma)$
- $x\sigma \stackrel{\text{def}}{=} x$ (when $x \in X$),

- $(s[\diamond]_p^N \cdot u)\sigma \stackrel{\mathrm{def}}{=} \underbrace{s\sigma[\ldots s\sigma[}_{N\sigma} u\sigma \underbrace{]_p \ldots]_p}_{N\sigma}$

Similarly, we use the notation $t[\diamond]_p^n$ where n is a (constant) non-negative integer for the context obtained by unfolding $t[]_p$ n times.

A *solution* σ to an ordinary equation $s = t$ is a pair (σ_1, σ_2) of a mapping σ_1 which assigns a non-negative integer to each variable in $IV(s = t)$ and a substitution σ_2 which assigns a I-term to each variable in $Var(s = t)$ in such a way that $s\sigma_1\sigma_2 \equiv t\sigma_1\sigma_2$. This definition extends to any of our formulae in a straightforward way.

3 Outline of the unification procedure

Our goal is to transform any unification formula ψ into a formula ψ' which is equivalent (i.e. has the same set of solutions) to ψ and which is in *solved form*.

Definition 3.1 *A* solved form *is either* $\perp$, $\top$ *or a finite disjunction of formulae of the form*

$$\exists \vec{n}.\ N_1 = E_1 \wedge \ldots \wedge N_k = E_k \wedge x_1 = t_1 \wedge \ldots \wedge x_n = t_n$$

where $N_1, \ldots, N_k$ *are integer (free) variables which occur only once in the conjunction,* $E_1, \ldots, E_k$ *are linear expressions over* $\vec{n}$ *and* $x_1, \ldots, x_n$ *are ordinary variables which are solved in the conjunction (i.e. that occur only once in the conjunction).*

Such solved forms may also be regarded as substitutions.

The transformation of an unification formula into a solved form is described by means of *transformation rules*, as in [5].

We split the rules into 5 parts. First, in section 4, we use the classical unification rules (or slight extensions of them). Then, we may assume that every unsolved ordinary equation has (at least) one member of the form $s^N \cdot t$. Such terms will be called N*-terms* in the following.

The second step (again in section 4) consists in getting rid of equations $s[\diamond]_p^N \cdot t = u$ when p is a position of u. This step is very simple: we only "unfold" $s[\diamond]_p^N$, after which me may again apply the unification elementary rules. Now, we have only to consider equations of the form $s[t_1[\diamond]_{q_1}^{N_1} \cdot u_1]_p = t_2[\diamond]_{q_2}^{N_2} \cdot u_2$ (see figure 2).

The third step consists roughly in taking a "common multiple" of q_1 and q_2, reducing the general case to the case where q_1 and q_2 have the same length. For, we have to consider separately the cases where $N_2 \leq |q_1|$ and $N_1 \leq |q_2|$. Then, dividing N_2 by $|q_1|$ and N_1 by $|q_2|$, it is possible to replace q_1 with $q_1^{|q_2|}$ and q_2 with $q_2^{|q_1|}$ which have the same size.

The fourth part consists in reducing again the general case to the case where $q_2 \leq_{pref} p \cdot q_1 \leq_{pref} q_2 \cdot q_2$: it is sufficient to unfold once on each side. Now, if this last condition is satisfied and if $|q_1| = |q_2|$, there must exist q' such that $q_2 = p \cdot q'$ and $q_1 = q' \cdot p$. (That is the main trick).

Now, if there is no clash, we are left to solve equations of the form $s[v_1[w_1]_{q'}[\diamond]_{q'p}^{N_1} \cdot u_1]_p = v_2[w_2]_p[\diamond]_{pq'}^{N_2} \cdot u_2$. This situation is depicted on figure 3. We have first to ensure the equality of subterms which are not located along the path $p \cdot (q' \cdot p)^{N_1}$; this is ensured by the equalities $s = v_2$, $v_1 = w_2$ and $w_1 = v_2$. Then, either $N_1 = N_2$, $N_1 > N_2$ or $N_1 < N_2$. In each case, we simplify both sides: the path $(p \cdot q')^{\min(N_2, N_1)}$ is shared by the two members of the equation.

The correctness of all the rules (i.e. the preservation of the set of solutions) is not difficult in general. The termination is neither very difficult because we never introduce new ordinary variables. Moreover, the number of integer variables which occur in the I-terms of the problem is never increasing. When these two numbers are constant, then the rules aim at "separating the integer variables", which is a bit technical to explain formally.

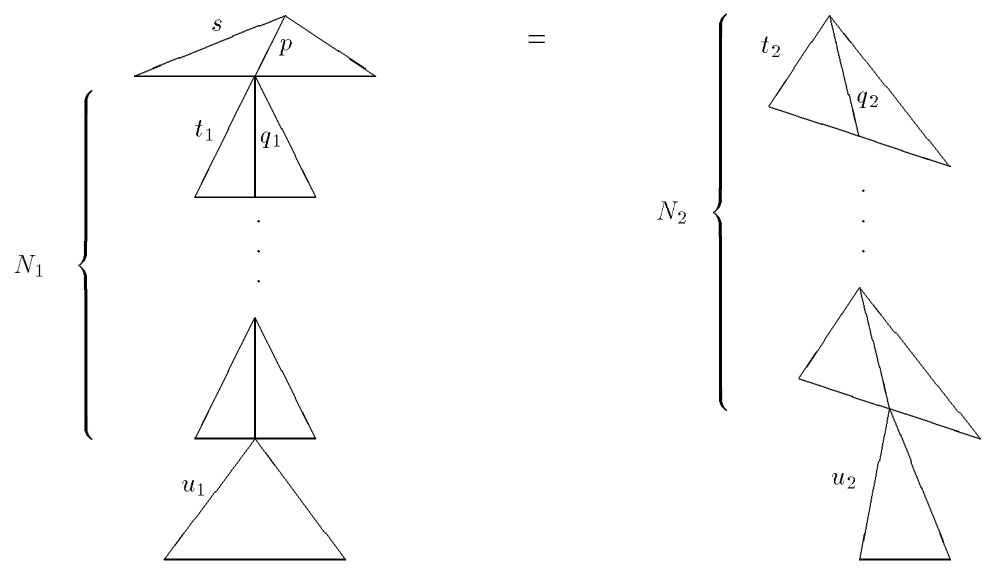


Figure 2: Equations that remain to be considered after step 2

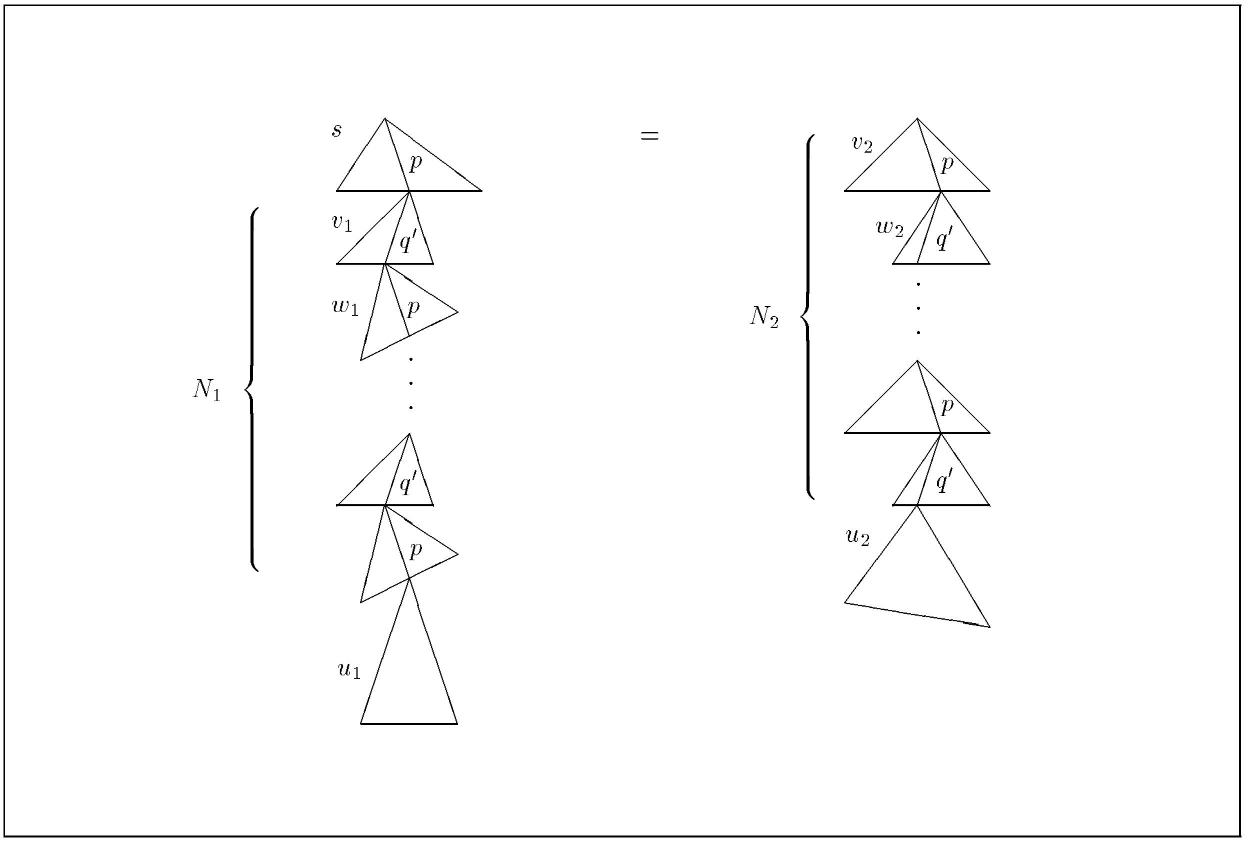


Figure 3: The equations that remain to be solved after step 4

4 Elementary Rules

First of all, we restrict our attention to solutions which assign non-null integers to the integer variables of the problem. Indeed, we may at once non-deterministically choose the integer variables which are mapped to 0. Hence, from now on, we only consider such interpretations. [1]

Trivial	$s = s$	$\rightarrow$	$\top$	
Decompose	$f(\vec{s}) = f(\vec{t})$	$\rightarrow$	$\vec{s} = \vec{t}$	
Clash	$f(\vec{s}) = g(\vec{t})$	$\rightarrow$	$\bot$	if $f \neq g$
Variable Elimination	$x = s \wedge P$	$\rightarrow$	$x = s \wedge P\{x \mapsto s\}$	
				if $x \notin Var(s), x \in Var(P)$ and $(s \in X \Rightarrow s \in Var(P))$
Occur-Check	$x = s$	$\rightarrow$	$\bot$	
				if $s \not\equiv x$ and $x \in Var(s)$

Let R_1 be the above system of rules.

Lemma 4.1 *All rules in R_1 are correct. Moreover R_1 terminates on any unification formula.*

Proof

The correctness is quite straightforward. The termination is classical: it is similar to the termination proof of the usual unification rules (see [5]). □

Let us call an equation *solved* in a unification formula ϕ if one of its member is a variable $x \in X$ which has only one occurrence in ϕ. (In such a case x itself is also called a *solved variable*).

Lemma 4.2 *If ϕ is a unification formula which is irreducible w.r.t. R_1, then equations between I-terms which occur in ϕ are either solved or of the form $s^N \cdot t = u$ where u is not a variable.*

Proof

Every equation which is not of the form $s^N \cdot t = u$ where u is not a variable can be written either $x = s$ where x is a variable, or $f(\vec{s}) = g(\vec{t})$. In the first case, either $x = s$ is solved or else $x \in Var(s)$ (in which case, the Occur-Check Rule or the Trivial rule applies) or the Variable Elimination rule applies. In the second case, either Clash or Decompose applies. □

We first consider the case of equations $s = t[\diamond]_p^N \cdot u$ where no prefix q of p is such that $s|_q$ is an N-term. In such a case, we may "unfold" $t[\diamond]_p^N$ once: this will lead by decomposition along the path p to equations which have either less integer exponents or the same integer exponents and smaller terms.

Example 4.3
Consider the equation $f(\diamond, a)^N \cdot x = f(y, f(a, \diamond)^M \cdot z)$. We are going to unfold $f(\diamond, a)^N$: either $N = 1$ which leads to the equation $f(x, a) = f(y, f(a, \diamond)^M \cdot z)$ which contains less integer variables, or else N can be written $N'+1$ (with $N' \geq 1$) and $f(f(\diamond, a)^{N'} \cdot x, a) = f(y, f(a, \diamond)^M \cdot z)$. This last equation can be simplified using R_1, which yields

$$f(\diamond, a)^{N'} \cdot x = y \wedge a = f(a, \diamond)^M \cdot z$$

a conjunction of simpler equations.

[1] The non-deterministic choice at once is very inefficient and is not really necessary. We choose this presentation for sake of simplicity (for example, we can use occur-checks without considering the null case separately) but this should not be followed in an implementation.

$$\text{(Unfold 1)} \quad s = t[\diamond]_p^N \cdot u \quad \rightarrow \quad (N = 1 \wedge s = t[u]_p) \vee (\exists M. N = M + 1 \wedge s = t[t^M \cdot u]_p$$

If

- there is no prefix q of p such that $s|_q$ is an N-term.
- R_1 does not apply

Lemma 4.4 *The system R_2 obtained by adding Unfold 1 to R_1 is correct and terminating.*

Proof

The correctness is straightforward: only termination needs a proof. We give an interpretation of formulae which are irreducible w.r.t. R_1.

Any unification formula can be written

$$(\exists \vec{n_1}.\phi_1 \wedge \psi_1) \vee \ldots \vee (\exists \vec{n_k}.\phi_k \wedge \psi_k)$$

where each ϕ_i is a system of linear diophantine equations and ψ_i is a pure conjunction of equations. We interpret this unification formula as the multiset

$$\{I(\psi_1), \ldots I(\psi_k)\}$$

of interpretations of each pure part. The multisets are ordered using the multiset extension of the ordering on the components.

Now, $I(s_1 = t_1 \wedge \ldots \wedge s_n = t_n)$ is a pair $(a, \{I(s_1 = t_1), \ldots, I(s_n = t_n)\})$ which consists of the number a of unsolved (ordinary) variables in the conjunction and the multiset of interpretations of each equation. The pairs are ordered lexicographically. $I(\perp)$ is assumed to be minimal.

Before to define $I(s = t)$ we need to define some measures on the I-terms. These measures will be re-used in the following. Roughly, $E - size(s)$ will give a measure of the number of *nested N-terms*. But actually, there are two notions of nestedness: in the expression $s^N \cdot t^M \cdot u$, t^M is nested below s^N. In the expression $s[t^M \cdot u, \diamond]^N \cdot v$, t^M is nested below s^N in another way because, for any instance n, m of N, M respectively, t will be repeated $n \times m$ times in the latter case whereas t will be repeated m times in the former case. That is why E-size(s) is a pair of natural numbers, recursively defined as follows:

- E-size(x) = E-size($\diamond$) = E-size(a) = $(0, 0)$ for every variable x and every constant symbol a
- E-size($f(s_1, \ldots, s_n)$) = $\max\{$E-size$(s_i) \mid 1 \leq i \leq n\}$
- E-size($s^N \cdot t$) = $\max\{(n_1 + 1, 0), (m_1, m_2 + 1)\}$ if E-size(s) = (n_1, n_2) and E-size(t) = (m_1, m_2) (The maximum is considered w.r.t. the lexicographic extension of the ordering on natural numbers)

$I(s = t)$ is the multiset of pairs $\{(\text{E-size}(s), |Pos(s)|), (\text{E-size}(t), |Pos(t)|)\}$, ordered using multiset and lexicographic extensions of the orderings.

We are going to show that the interpretation is strictly decreasing on the normal forms w.r.t. R_1 by application of (Unfold 1).

Assume that

$$\begin{aligned}\Phi \vee (\exists \vec{n}.\phi \wedge \psi \wedge s = t[\diamond]_p^N \cdot u) \rightarrow\ & \Phi \vee (\exists \vec{n}.\phi \wedge N = 1 \wedge \psi \wedge s = t[u]_p) \\ & \vee (\exists \vec{n}, M.\phi \wedge N = M + 1 \wedge \psi \wedge s = t[t[\diamond]_p^M \cdot u]_p)\end{aligned}$$

R_1 does not introduce any disjunction. Hence, by definition of the multiset extension of an ordering, it is sufficient to prove that the normal forms (w.r.t. R_1) of $\psi \wedge s = t[u]_p$ and $\psi \wedge s = t[t[\diamond]_p^M \cdot u]_p$ are both strictly smaller than $\psi \wedge s = t[\diamond]_p^N \cdot u$ in the interpretation. If there is at least one occurrence of application of the Variable Elimination rule along the normalization by R_1, then the interpretation is decreasing because there are strictly less unsolved variables in the normal form. Similarly, if a Clash or an Occur-Check is applied during the normalization, the decreasingness is obvious. These situations can be excluded. Now, since it only remains (Decompose) and (Trivial), we only have to compare $s = t[\diamond]_p^N \cdot u$ with the decomposed forms of $s = t[u]_p$ and $s = t[t[\diamond]_p^M \cdot u]_p$ respectively. The decomposed forms of these equations are conjunctions of equations $s|_q = t|_q[u]_r$ (resp. $s|_q = t|_q[t[\diamond]_p^M \cdot u]_r$) where q is a position of s and t and $q \cdot r = p$ and equations of the form $s|_q = t|_q$ where $p \parallel q$ and q is a position of s, t. It is quite straightforward to see that E-size$(s|_q) \leq$ E-size(s). We also have

- E-size$(t|_q[u]_r) <$ E-size$(t[\diamond]_p^N \cdot u)$ since E-size$(t|_q[u]_r) = \max\{$E-size$(t|_q),$ E-size$(u)\}$ and both E-size(t) and E-size(u) are strictly smaller than E-size$(t[\diamond]_p^N \cdot u)$.
- E-size$(t|_q) <$ E-size$(t[\diamond]_p^N \cdot u)$ when $p \parallel q$ since E-size$(t|_q) \leq$ E-size$(t) <$ E-size$(t[\diamond]_p^N \cdot u)$
- $|Pos(s|_q)| < |Pos(s)|$.

Hence, we only have to consider the case of irreducible equations $s|_q = t|_q[t[\diamond]_p^M \cdot u]_r$ such that $q \cdot r = p$. In these cases, by hypothesis on the rule (Unfold 1) the head of $s|_q$ is either a function symbol or a variable. Then it cannot be a variable because we assumed that there is no (Variable Elimination) and no (Occur Check). This means that $s|_q \equiv f(s_1, \ldots, s_n)$. Then, by irreducibility of $s|_q = t|_q[t[\diamond]_p^M \cdot u]_r$, we must have $q = p$: the equation can be written $s|_p = t[\diamond]_p^M \cdot u$. Now the interpretation of this equation is strictly smaller than the interpretation of $s = t[\diamond]_p^N \cdot u$.

In order to complete the proof, we only have to associate with each unification formula ϕ the pair consisting of the above interpretation applied on a normal form of ϕ w.r.t. R_1 and the formula itself. The pairs being ordered lexicographically and the second components being ordered by the reduction relation $\rightarrow_{R_1}^*$, we get a well founded interpretation which is decreasing by any application of a rule. □

5 Solving some particular equations

It only remains to consider the equations of the form $s[t_1[\diamond]_{q_1}^{N_1} \cdot u_1]_p = t_2[\diamond]_{q_2}^{N_2} \cdot u_2$ where p is a prefix of q_2 (p may equal Λ, but q_1 and q_2 may not, by definition).

5.1 Reduction to $|q_1| = |q_2|$

The first step consists in reducing the above problem to the case $|q_1| = |q_2|$. This is done by unfolding $|q_2|$ times $t_1[\diamond]_{q_1}^{N_1}$ and unfolding $|q_1|$ times $t_2[\diamond]_{q_2}^{N_2}$. Let us give an example.

Example 5.1
Consider the equation $g(f(a, g(\diamond))^{N_1} \cdot z) = g(f(x, g(\diamond)))^{N_2} \cdot y$. The first $\diamond$ position $q_1 = 21$ has a length 2 and the second $\diamond$ position $q_2 = 121$ has a length 3. We unfold the I-terms so as to have the same lengths; the above equation is equivalent to the disjunction of 9 formulae. The first three formulae correspond to "small" values of N_1, N_2:

- $N_2 = 1 \wedge g(f(a, g(\diamond))^{N_1} \cdot z) = g(f(x, g(y)))$

- $N_1 = 1 \wedge g(f(a, g(z))) = g(f(x, g(\diamond)))^{N_2} \cdot y$
- $N_1 = 2 \wedge g(f(a, g(f(a, g(z))))) = g(f(x, g(\diamond)))^{N_2} \cdot y$

The next 6 formulae correspond to all possible remainders of the divisions of N_1 by 3 and N_2 by 2 respectively; we gather together 3 iterations of $g(f(a, \diamond))$ and 2 iterations of $g(f(x, g(\diamond)))$.

- $\exists N_1', N_2'. N_2 = 2 \times N_2' \wedge N_1 = 3 \times N_1'$
  $\wedge g(f(a, g(f(a, g(f(a, g(\diamond)))))))^{N_1'} \cdot z) = g(f(x, g(g(f(x, g(\diamond))))))^{N_2'} \cdot y$
- $\exists N_1', N_2'. N_2 = 2 \times N_2' \wedge N_1 = 3 \times N_1' + 1$
  $\wedge g(f(a, g(f(a, g(f(a, g(\diamond)))))))^{N_1'} \cdot f(a, g(z))) = g(f(x, g(g(f(x, g(\diamond))))))^{N_2'} \cdot y$
- $\exists N_1', N_2'. N_2 = 2 \times N_2' \wedge N_1 = 3 \times N_1' + 2$
  $\wedge g(f(a, g(f(a, g(f(a, g(\diamond)))))))^{N_1'} \cdot f(a, g(f(a, g(z))))) = g(f(x, g(g(f(x, g(\diamond))))))^{N_2'} \cdot y$
- $\exists N_1', N_2'. N_2 = 2 \times N_2' + 1 \wedge N_1 = 3 \times N_1'$
  $\wedge g(f(a, g(f(a, g(f(a, g(\diamond)))))))^{N_1'} \cdot z) = g(f(x, g(g(f(x, g(\diamond))))))^{N_2'} \cdot g(f(x, g(y)))$
- $\exists N_1', N_2'. N_2 = 2 \times N_2' + 1 \wedge N_1 = 3 \times N_1' + 1$
  $\wedge g(f(a, g(f(a, g(f(a, g(\diamond)))))))^{N_2'} \cdot f(a, g(z))) = g(f(x, g(g(f(x, g(\diamond))))))^{N_2'} \cdot g(f(x, g(y)))$
- $\exists N_1', N_2'. N_2 = 2 \times N_2' + 1 \wedge N_1 = 3 \times N_1' + 2$
  $\wedge g(f(a, g(f(a, g(f(a, g(\diamond)))))))^{N_1'} \cdot f(a, g(f(a, g(z))))) = g(f(x, g(g(f(x, g(\diamond))))))^{N_2'} \cdot g(f(x, g(y)))$.

Now, the two $\diamond$ positions 121121 and 121212 have the same length.

$$
\begin{array}{ll}
\text{(Unfold 2)} & s[t_1[\diamond]_{q_1}^{N_1} \cdot u_1]_p = t_2[\diamond]_{q_2}^{N_2} \cdot u_2 \rightarrow \\
& \displaystyle\bigvee_{1 \le r_1 < \alpha_2} N_1 = r_1 \wedge s[t_1[\diamond]_{q_1}^{r_1} \cdot u_1]_p = t_2[\diamond]_{q_2}^{N_2} \cdot u_2 \\
& \displaystyle\bigvee_{1 \le r_2 < \alpha_1} N_2 = r_2 \wedge s[t_1[\diamond]_{q_1}^{N_1} \cdot u_1]_p = t_2[\diamond]_{q_2}^{r_2} \cdot u_2 \\
& \displaystyle\bigvee_{\substack{0 \le r_1 < \alpha_2 \\ 0 \le r_2 < \alpha_1}} \exists M_1, M_2. N_1 = \alpha_2 \times M_1 + r_1 \wedge N_2 = \alpha_1 \times M_2 + r_2 \\
& \qquad \wedge s[(t_1^{\alpha_2})^{M_1} \cdot t_1^{r_1} \cdot u_1]_p = (t_2^{\alpha_1})^{M_2} \cdot t_2^{r_2} \cdot u_2
\end{array}
$$

if $|q_1| \neq |q_2|$, R_2 cannot be applied and p is a prefix of q_2. d is the gcd of $|q_1|$ and $|q_2|$, $\alpha_1 = \frac{|q_1|}{d}$, $\alpha_2 = \frac{|q_2|}{d}$ and $m = \alpha_1 \times \alpha_2 \times d$ is the lcm of $|q_1|$ and $|q_2|$.

Lemma 5.2 *(Unfold 2) is correct.*

Proof

Concerning correctness, we only have to notice that all integers $n_1, n_2 \geq 1$ (i.e. all possible assignments to N_1, N_2) have to satisfy $n_1 \in \{1, \ldots, \alpha_2 - 1\}$ or $n_2 \in \{1, \ldots, \alpha_1 - 1\}$ or else $n_1 = \alpha_2 \times m_1 + r_1$ and $n_2 = \alpha_1 \times m_2 + r_2$ for some $m_1, m_2 \geq 1$ and some $r_1 \in \{0, \ldots, \alpha_2 - 1\}, r_2 \in \{0, \ldots, \alpha_1 - 1\}$. $\square$

$R_2 \cup \{(Unfold2)\}$ is also terminating, but we will prove this later for all rules together.

5.2 Ensuring some prefix conditions on the positions

The second step consists in applying rules similar to Unfold 1 in order to eliminate uncomparable positions. More precisely, we want to ensure that $p \leq_{pref} q_2 \leq_{pref} p \cdot q_1 \leq_{pref} q_2 \cdot q_2$ in equations $s[t_1[\diamond]_{q_1}^{N_1} \cdot u_1]_p = t_2[\diamond]_{q_2}^{N_2} \cdot u_2$. (The first inequality is already known from (Unfold 1)).

Example 5.3
Consider the equation $f(a, f(f(x, \diamond), y)^{N_1} \cdot z) = f(a, f(x', \diamond))^{N_2} \cdot y'$. We have $p = 2$, $q_1 = 12$ and $q_2 = 22$. None of the rules which have been shown up to now can be applied. But, unfolding once on both sides, we get:

$$\begin{array}{l} N_1 = 1 \wedge f(a, f(f(x, z), y) = f(a, f(x', \diamond))^{N_2} \cdot y' \\ \vee\ N_2 = 1 \wedge f(a, f(f(x, \diamond), y)^{N_1} \cdot z) = f(a, f(x', y')) \\ \vee\ \exists M_1, M_2. N_1 = M_1 + 1 \wedge N_2 = M_2 + 1 \\ \qquad\qquad\qquad \wedge f(a, f(f(x, f(f(x, \diamond), y)^{M_1} \cdot z), y) = f(a, f(x', f(a, f(x', \diamond))^{M_2} \cdot y')) \end{array}$$

By decomposition the last equation reduces the exponent sizes of the members, because q_2 is not a prefix of $p \cdot q_1$; we get:

$$x' = f(x, f(f(x, \diamond), y))^{M_1} \cdot z \wedge y = f(a, f(x', \diamond))^{M_2} \cdot y'$$

$$\begin{array}{lll} \text{(Unfold 3)} & s[t_1[\diamond]_{q_1}^{N_1} \cdot u_1]_p = t_2[\diamond]_{q_2}^{N_2} \cdot u_2 \quad \rightarrow & (N_2 = 1 \wedge t_2[u_2]_{q_2} = s[t_1^{N_1} \cdot u_1]_p) \\ & & \vee(N_1 = 1 \wedge s[t_1[u_1]_{q_1}]_p = t_2^{N_2} \cdot u_2) \\ & & \vee(N_2 = 2 \wedge t_2[t_2[u_2]_{q_2}]_{q_2} = s[t_1^{N_1} \cdot u_1]_p) \\ & & \vee(\exists M_1, M_2. N_1 = M_1 + 1 \wedge N_2 = M_2 + 2 \\ & & \qquad \wedge s[t_1[t_1^{M_1} \cdot u_1]_{q_1}]_p = t_2[t_2[t_2^{M_2} \cdot u_2]_{q_2}]_{q_2}) \end{array}$$

If p is a prefix of q_2, but either q_2 is not a prefix of $p \cdot q_1$ or $p \cdot q_1$ is not a prefix of $q_2 \cdot q_2$.

Lemma 5.4 *(Unfold 3) is correct.*

Proof
It is sufficient to consider that any solution either assigns N_2 to 1 or 2, or assigns N_1 to 1, or else assigns N_1 to $n_1 \geq 2$ and N_2 to $n_2 \geq 3$. □

5.3 The crux decomposition rule

The key property is given in the following lemma. It shows that all equations that remain to be considered do have some commutation properties on their paths (actually they are of the form depicted in figure 3).

Lemma 5.5 *If $s[t_1[\diamond]_{q_1}^{N_1} \cdot u_1]_p = t_2[\diamond]_{q_2}^{N_2} \cdot u_2$ is irreducible by (Unfold 1), (Unfold 2) and (Unfold 3), then there is a q' such that $q_2 = p \cdot q'$ and $q_1 = q' \cdot p$.*

Proof
By irreducibility w.r.t. (Unfold i) (i=1,2,3), we know that

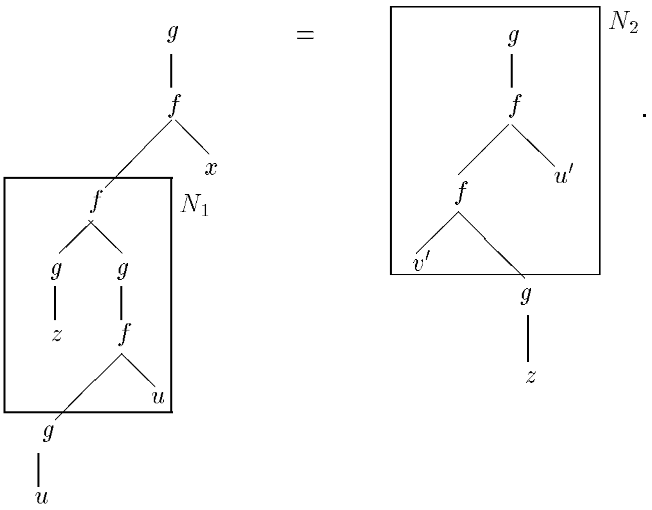
$$p \leq_{pref} q_2 \leq_{pref} p \cdot q_1 \leq_{pref} q_2 \cdot q_2$$

We also know that $|q_1| = |q_2|$. Let then $q_2 = p \cdot q'$ and $p \cdot q_1 = q_2 \cdot q''$. Then $p \cdot q_1 = p \cdot q' \cdot q''$, hence $q_1 = q' \cdot q''$. From $q_2 = p \cdot q'$, $q_1 = q' \cdot q''$ and $|q_1| = |q_2|$, we derive that $|p| = |q''|$.

Now, $p \cdot q_1$ is a prefix of $q_2 \cdot q_2$, hence, for some r, $q_2 \cdot q_2 = p \cdot q_1 \cdot r$. Which means that $p \cdot q' = q'' \cdot r$. Using now $|p| = |q''|$ we derive $p = q''$. It follows that $q_2 = p \cdot q'$ and $q_1 = q' \cdot p$. □

In order to have an intuition of the next decomposition rule, have a look at figure 3: in such equations, we must have $s = v_2 = w_1$ and $v_1 = w_2$. Then, rearranging the parentheses, we lift up the N_1 exponent and guess whether $N_1 \geq N_2$ or $N_2 \geq N_1$. Assume for example $N_1 \geq N_2$, then we can remove $t_2[\diamond]_{q_2}^{N_2}$ on both sides.

Example 5.6
Consider the equation $g(f(f(g(z), g(f(\diamond, u)))^{N_1} \cdot g(u), x)) = g(f(f(v', \diamond), u'))^{N_2} \cdot g(z)$ which can be depicted as:



We have $s \equiv g(f(a, x))$, $v_1 \equiv f(g(z), a)$, $w_1 \equiv g(f(a, u))$, $v_2 \equiv g(f(a, u'))$, $w_2 \equiv f(v', a)$, $p = 11$ and $q' = 2$. The equation can be decomposed into the following problems (the a can be replaced by any term. we only replace the $\diamond$ in order to keep terms in T):

$$
\begin{array}{ll}
\ g(f(a,x)) = g(f(a,u')) & \text{(i.e. } s = v_2) \\
\wedge\ g(f(a,x)) = g(f(a,u) & \text{(i.e. } s = w_1) \\
\wedge\ f(g(z),a) = f(v',a) & \text{(i.e. } v_1 = w_2) \\
\wedge\ (\quad (\ N_1 = N_2 \wedge g(f(g(u),u)) = g(z)\ ) & \\
\quad \vee\ (\ \exists M_1.N_1 = N_2 + M_1 \wedge g(f(f(g(z), g(f(\diamond,u)))^{M_1} \cdot g(u), x)) = g(z) & \\
\quad \vee\ (\ \exists M_2.N_2 = N_1 + M_2 \wedge g(f(g(u),u)) = g(f(f(v',\diamond),u'))^{M_2} \cdot g(z)\ )\ ) &
\end{array}
$$

More generally, we get the following rule (where the right side has actually to be put in disjunctive normal form in order to get a unification formula):

$$
\begin{array}{lll}
\text{(Decompose 2)} & s[(v_1[w_1[\diamond]_p]_{q'})^{N_1} \cdot u_1]_p = (v_2[w_2[\diamond]_{q'}]_p)^{N_2} \cdot u_2 \rightarrow & \\
& \quad s[a]_p = w_1[a]_p \wedge s[a]_p = v_2[a]_p \wedge v_1[a]_{q'} = w_2[a]_{q'} & \\
& \wedge\ (\quad (\ N_1 = N_2 \wedge s[u_1]_p = u_2) & \\
& \quad \vee\ (\ \exists M_1.N_1 = N_2 + M_1 \wedge s[(v_1[w_1[\diamond]_p]_{q'})^{M_1} \cdot u_1]_p = u_2) & \\
& \quad \vee\ (\ \exists M_2.N_2 = N_1 + M_2 \wedge w_1[u_1]_p = (v_2[w_2[\diamond]_{q'}]_p)^{M_2} \cdot u_2)) &
\end{array}
$$

Lemma 5.7 *The rule (Decompose 2) is correct. (a being any constant).*

Proof

This is again quite easy: the first three equations are obtained by considering the top part of both sides for an arbitrary assignment (remember that N_1 and N_2 must be assigned to integers larger or equal to 1). The next 3 disjunctions are obtained by guessing how N_1 and N_2 compares. Assume for example that N_1 is assigned to a strictly larger integer than N_2. Writing N_1 as $N_2 + M_1$, and using the equations $s = v_2$ and $v_1 = w_2$, the equation becomes

$$v_2[(w_2[v_2[\Diamond]_p]_{q'})^{N_2} \cdot (v_1[w_1[\Diamond]_p]_{q'})^{M_1} \cdot u_1]_p = (v_2[w_2[\Diamond]_{q'}]_p)^{N_2} \cdot u_2$$

which becomes, after reorganizing the terms and simplifying by $(v_2[w_2[\Diamond]_{q'}]_p)^{N_2}$ on both sides:

$$v_2[(v_1[w_1[\Diamond]_p]_{q'})^{M_1} \cdot u_1]_p = u_2$$

Replacing again v_2 with s, we get the corresponding equation . □

6 Unification of Terms with Integer exponents

Now, we have the whole set of rules. Let us prove first termination. Then, solving the diophantine equations part, leads to solved forms.

6.1 Termination

Lemma 6.1 *The system R_3 which consists of applying R_2 as long as possible and (Unfold 2), (Unfold 3), (Decompose 2) on irreducible problems w.r.t. R_2 is correct and terminating.*

Proof

The correctness has been shown in lemmas 4.4, 5.2, 5.4, 5.7

Regarding termination, we use the same interpretation as in lemma 4.4, except for the definition of $I(s = t)$ which is modified as follows:

$$I(s = t) = (\{\text{E-size}(s), \text{E-size}(t)\}, \{|Pos(s)|, |Pos(t)|\}, \text{status}(s = t))$$

where $\text{status}(s = t)$ is equal to 1 if (Unfold 2) can be applied on $s = t$. Otherwise, $\text{status}(s = t) = 0$.

As in the proof of lemma 4.4, we only have to prove that, for each rule, each equation in the pure part of a normal form w.r.t. R_1 of the right hand side is strictly smaller than the left hand side.

In the following table, we summarize, for each rule, and each equation in a normal form w.r.t. R_1 of the right hand side of this rule, how it compares with the left side of the rule. We assume here (as in the proof of lemma 4.4) that there is no occurrence of a (Variable Elimination) rule along the simplification. (Because, otherwise, the interpretation is obviously decreasing). A pair $\{<, =\}$ on the line $s = t$ for the interpretation i means for example that $s = t$ has one member for which i is strictly decreasing and one member for which i is constant. In some places, we use the notation $u|_q$ without any further explanation. In such cases, q is assumed to be a position of u, such that the displayed equation is irreducible w.r.t. (Decompose 1). Finally, most of the results reported in the table are quite easy. How look the irreducible forms of equations in the right side of (Unfold 1) has been investigated in the proof of lemma 4.4. We only give further explanation for the two results which are quoted (1), (2), (3), (4) and (5) in the table.

		E-size	size	status
(Unfold 1)	$s\|_q = t[u]_p\|_q$	$\{\leq, <\}$		
	$s\|_q = t\|_q$	$\{\leq, <\}$		
	$s\|_p = t[\diamond]_p^M \cdot u$	$\{\leq, =\}$	$\{<, =\}$	
(Unfold 2)	$s[t_1[\diamond]_{q_1}^{r_1} \cdot u_1]_p = t_2[\diamond]_{q_2}^{N_2} \cdot u_2$	$\{<, =\}$ (1)		
	$s[t_1[\diamond]_{q_1}^{N_1} \cdot u_1]_p\|_q = (t_2[\diamond]_{q_2}^{r_2} \cdot u_2)\|_q$	$\{\leq, <\}$ (1)		
	$s[(t_1^{\alpha_2})^{M_1} \cdot t_1^{r_1} \cdot u_1]_p = (t_2^{\alpha_1})^{M_2} \cdot t_2^{r_2} \cdot u_2$	$\{=, =\}$ (2)	$\{=,=\}$ (3)	$<$ (4)
(Unfold 3)	$t_2[u_2]_{q_2}\|_q = s[t_1^{N_1} \cdot u_1]_p\|_q$	$\{<, \leq\}$		
	$s[t_1[u_1]_{q_1}]_p = t_2^{N_2} \cdot u_2$	$\{<, =\}$		
	$t_2[t_2[u_2]_{q_2}]_{q_2}\|_q = s[t_1^{N_1} \cdot u_1]_p\|_q$	$\{<, \leq\}$		
	$s[t_1[t_1^{M_1} \cdot u_1]_{q_1}]_p\|_q = t_2[t_2[t_2^{M_2} \cdot u_2]_{q_2}]_{q_2}\|_q$	$\{<, \leq\}$ (5)		
(Decompose 2)	$s\|_q = w_1\|_q$	$\{<, <\}$		
	$s\|_q = v_2\|_q$	$\{<, <\}$		
	$v_1\|_q = w_2\|_q$	$\{<, <\}$		
	$s[u_1]_p\|_q = u_2\|_q$	$\{<, <\}$		
	$s[(v_1[w_1[\diamond]_p]_{q'})^{M_1} \cdot u_1]_p\|_q = u_2\|_q$	$\{\leq, <\}$		
	$w_1[u_1]_p\|_q = ((v_2[w_2[\diamond]_{q'}]_p)^{M_2} \cdot u_2)\|_q$	$\{<, \leq\}$		

(1) It should be noted that, in these situations, the number of occurrences of interger variables may increase. However, E-size is still strictly decreasing. Indeed,

$$\text{E-size}(s[t_1[\diamond]_{q_1}^{r_1} \cdot u_1]_p) = \max\{\text{E-size}(s), \text{E-size}(t_1), \text{E-size}(u_1)\}$$

since, by definition, for all terms u, $\text{E-size}(u) = \max\{\text{E-size}(u|_q) \mid q \in Pos(u)\}$ and $p \cdot q_1^{r_1} \in Pos(s[t_1[\diamond]_{q_1}^{r_1} \cdot u_1]_p)$. Moreover, as already noticed in the proof of lemma 4.4, for every term $t^N \cdot u$ E-size(t) and E-size(u) are both strictly smaller than E-size$(t^N \cdot u)$.

(2) As above, $\text{E-size}(t_1^{\alpha_2}) = \text{E-size}(t_1)$ and $\text{E-size}(t_1^{r_1} \cdot u_1) = \max\{\text{E-size}(t_1), \text{E-size}(u_1)\}$. Now, assume that $\text{E-size}(t_1) = (n_1, n_2)$ and $\text{E-size}(u_1) = (m_1, m_2)$. There are two situations: either $n_1 \geq m_1$ or $n_1 < m_1$. In the first case, $\text{E-size}(t_1^{r_1} \cdot u_1) \leq (n_1, \max\{m_1, m_2\})$, hence

$$\text{E-size}((t_1^{\alpha_2})^{M_1} \cdot t_1^{r_1} \cdot u_1) = (n_1 + 1, 0) = \text{E-size}(t_1^{N_1} \cdot u_1)$$

In the second case, $\text{E-size}(t_1^{r_1} \cdot u_1) = (m_1, m_2)$, hence

$$\text{E-size}((t_1^{\alpha_2})^{M_1} \cdot t_1^{r_1} \cdot u_1) = (m_1, m_2 + 1) = \text{E-size}(t_1^{N_1} \cdot u_1)$$

(3) By definition of Pos, $|Pos(s[(t_1^{\alpha_2})^{M_1} \cdot t_1^{r_1} \cdot u_1]_p)| = |Pos(s)| = |Pos(s[t_1^{N_1} \cdot u_1]_p)|$

(4) By definition, the equation on which (Unfold 2) is applied has a status 1. We have to check that (unfold 2) cannot be applied again on the equation we consider. In which case, the status is 0 and hence strictly decreasing. Note first that p is fixed (because it must be a prefix of q_2). Then note that $q_1^{\alpha_2}$ has the same length as $q_2^{\alpha_1}$, which prevents the application of (Unfold 2).

(5) Several cases have to be considered:

- if $q \parallel p \cdot q_1$, then

$$\text{E-size}(s[t_1[t_1^{M_1} \cdot u_1]_{q_1}]_p|_q) = \text{E-size}(s[t_1]_p|_q) < \text{E-size}(s[t_1[\diamond]_{q_1}^{N_1} \cdot u_1]_p)$$

  and the result follows.

- if $q \parallel q_2 \cdot q_2$, we get a similar result for the other member. (Note that, in any case, E-size is never increasing).
- Now assume that q is a prefix of both $p \cdot q_1$ and $q_2 \cdot q_2$. By irreducibility w.r.t. (Decompose 1), we must have $q = p \cdot q_1$ or $q = q_2 \cdot q_2$. Which means that either $p \cdot q_1$ is a prefix of $q_2 \cdot q_2$ or else $q_2 \cdot q_2$ is a prefix of $p \cdot q_1$. But both situations are prevented by the condition on the application of (Unfold 3). (See the rule). Hence, this last case cannot occur.

As in the proof of lemma 4.4, these decreasing properties complete the termination proof. □

6.2 Irreducible formulae

Lemma 6.2 *If a formula is irreducible w.r.t. R_3, then it is a disjunction of conjunctions of the form*

$$\exists \vec{n}.\ \phi \wedge x_1 = t_1 \wedge \ldots \wedge x_n = t_n$$

where ϕ is a conjunction of linear diophantine equations, $\vec{n}$ is a set of integer variables, $x_1, \ldots, x_n$ are ordinary variables which are solved in the conjunction.

Proof

We only have to check that each unification formula which is not in the form of the lemma can be reduced using one of the rules in R_3. As already noticed at the beginning of section 5, every irreducible unsolved equation (w.r.t. R_2) must be of the form $s[t_1[\diamond]_{q_1}^{N_1} \cdot u_1]_p = t_2[\diamond]_{q_2}^{N_2}$ with $p \leq_{pref} q_2$. By irreducibility w.r.t. (Unfold 2), q_1 and q_2 must have the same size. By irreducibility w.r.t (Unfold 3), we know that $q_2 \leq_{pref} p \cdot q_1 \leq_{pref} q_2 \cdot q_2$. And, in this case, by lemma 5.5, we can apply (Decompose 2). □

6.3 The last step

Now, we solve the linear diophantine equations parts. This means that the diophantine parts are replaced with finite disjunctions of systems of the form

$$\exists \vec{n}.N_1 = E_1 \wedge \ldots \wedge N_k = E_k$$

where $E_1, \ldots, E_k$ are linear expressions and $N_1, \ldots, N_k$ are integer variables occurring only once in the problem.

The values of $N_1, \ldots N_k$ can be replaced in the corresponding pure part, leading to equivalent solved forms. (t^E is expanded according to the rule $t^{N+M} \rightarrow t^N \cdot t^M$; then replacing integer variables with linear expressions yields a unification formula).

Theorem 6.3 *Unification of I-terms is decidable and finitary: there is a correct and terminating algorithm which computes a finite set of solved forms for $s = t$.*

6.4 Minimality

It should be noticed that the solved forms of our unification algorithms do not contain redundancies: if $d_1 \vee d_2$ is a solved form of an equation $s = t$, then d_1 and d_2 do not share any solution. This can be checked on each rule: each disjunction introduced by a rule splits the solutions in dijoint sets. However, we cannot claim that the solved form are *minimal*, because there are solved forms $d_1 \vee d_2$ where d_1, d_2 are conjunctions of equations, that can be "folded" into a single conjunction of equations. For example, consider the equation $f(x) = f(f(\diamond))^N \cdot f(a)$. By (Unfold 1) and further reductions, we get

$$(N = 1 \wedge x = f(f(a))) \vee \exists N'.N = N' + 1 \wedge x = f(f(f(\diamond))^{N'} \cdot f(a))$$

Which is obviously not minimal since $x = f(f(\diamond))^N \cdot a$ is equivalent to the original problem.

7 Extensions of the syntax

We may allow a number of new syntactic constructions while keeping the result of theorem 6.3. Let T_n be the set of terms with n "holes" ($T' = T_0$) defined as the least sets which satisfy:

$$\left\{ \begin{array}{rcl} x \in T_0 & \Leftarrow & x \in X \text{ or } x \in F, a(x) = 0 \\ f(s_1, \ldots, s_n) \in T_{m_1 + \ldots + m_n} & \Leftarrow & s_1 \in T_{m_1}, \ldots, s_n \in T_{m_n}, 1 \leq n \\ s_{p_1 \ldots p_k}^{N_1 \ldots N_k} \in T_k & \Leftarrow & s \in T_0, 1 \leq k, p_1, \ldots p_k \in Pos(s) - \{\Lambda\} \\ & & \qquad \forall i \neq j. p_i \parallel p_j, N_1, \ldots, N_k \in V_{\mathbf{N}} \\ s \cdot t \in T_{n+k-1} & \Leftarrow & s \in T_n, 1 \leq n,\ t \in T_k \end{array} \right.$$

Positions are defined as before; in particular, $Pos(s \cdot t) = \{\Lambda\}$. We do not use the special symbol $\diamond$ in this syntax for the reason shown in the following example.

Example 7.1 Let $s \equiv (f(a,b)_{1,2}^{N_1,N_2} \cdot c) \cdot d$. Assume that N_1 and N_2 are both assigned to 2. then, according to the semantics defined below, $s\sigma \equiv f(f(c,b), f(a,d))$. Now, using the special place holder $\diamond$, what should be $(f(\diamond, \diamond)^{2,2} \cdot c) \cdot d$? There are several possibilities for unfolding $f(\diamond, \diamond)_{1,2}^{2,2}$, but all of them lead to terms with more than two occurrences of $\diamond$. (This could be $f(f(\diamond, \diamond), f(\diamond, \diamond))$ or $f(f(\diamond, \diamond), f(f(\diamond, \diamond), \diamond)))$. Hence, we would have to decide which occurrences of $\diamond$ have to be replaced with c and which have to be replaced with d. This lead to another interpretation of the terms which is more complicated. And even if we do, the expressive power is not increased. In order to be able to express other sets of terms we have either to allow s to be $u \cdot v$ (which is investigated in the next section) or, e.g., unfold simultaneously w.r.t. all holes (as in [6]).

Now, given an assignment to the integer variables of $s \in T_m$, $s\sigma$ is defined by

$$\begin{array}{rcll} x\sigma & \stackrel{\text{def}}{=} & x & \text{if } x \in X, \text{ or } x \in F \\ f(s_1, \ldots, s_n)\sigma & \stackrel{\text{def}}{=} & f(s_1\sigma, \ldots, s_n\sigma) & \\ s_{p_1, \ldots, p_k}^{N_1, \ldots, N_k}\sigma & \stackrel{\text{def}}{=} & s_{p_1, \ldots, p_{k-1}}^{N_1, \ldots, N_{k-1}}\sigma[\underbrace{s\sigma[\ldots[s\sigma[\diamond]_{p_k}]\ldots]_{p_k}}_{N_k\sigma - 1})]_{p_k} & \\ (s \cdot t)\sigma & \stackrel{\text{def}}{=} & s\sigma[t\sigma]_p & \text{if } p \text{ is the smallest position of } \diamond \text{ in } s\sigma \end{array}$$

Assignments to 0 are excluded (because this would complicate a lot the definitions). And positions are compared lexicographically in the above definition.

The above additional constructions actually do not increase the expressive power of terms with exponents. That is what is shown in the next proposition. $t \in T_n$ is said to be *equivalent* to $t' \in T_n$ if $IV(t) = IV(t')$ and, for every assignment to these integer variables $t\sigma \equiv t'\sigma$. A term $s \in T$ is *weakly equivalent* to a term $t \in T_0$ if $IV(s) = IV(t)$ and,

- for every assignement σ of non-null integers to these variables, there is an assignment θ such that $s\theta \equiv t\sigma$

- for every assignment θ to these variables, there is an assignment σ of non-null integers such that $s\theta \equiv t\sigma$.

Proposition 7.2 *For every $t' \in T_0$ there is a weakly equivalent $t \in T$.*

Sketch of the Proof
We show the proposition by structural induction on t'.

- if $t' \equiv f(\vec{s'})$ with $\vec{s'} \in T_0^{a(f)}$, then it is sufficient to choose $t \equiv f(\vec{s})$ where each term $s_i \in \vec{s}$ satisfies, for all σ, $s_i\sigma \equiv s'_i\sigma$. ($\vec{s}$ exists by induction hypothesis).

- If $t' \in X$, then let $t \equiv t'$

- If we are not in one of the two above cases, then $t' \equiv s \cdot u$ with $s \in T_1$ and $u \in T_0$. Note that $\cdot$ is "associative" in the following sense: when $s \in T_n, t \in T_m, u \in T_k$ with $n, m \geq 1$, then $(s \cdot t) \cdot u$ and $s \cdot (t \cdot u)$ are equivalent. Hence, we may assume that t' has the form $(s \cdot u_1) \cdot \ldots \cdot u_k$ where $u_1, \ldots, u_k \in T_0$ and $s \in T_{k+1}$ cannot be decomposed as $s_1 \cdot s_2$. There are still two possible cases:

  - $s \equiv f(s_1, \ldots, s_n)$. In this case, t' is equivalent to a term $f(v_1, \ldots, v_n)$, where $v_i \equiv s_i \cdot u_{m_i} \cdot \ldots \cdot u_{m_{i+1}-1}$ and $m_1 = 1, m_{n+1} = k+1$. Therefore, this case reduces to the case $t' \equiv f(\vec{s'})$ which has already been considered.
  - $s \equiv (s_0)_{p_1,\ldots,p_m}^{N_1,\ldots,N_m}$ where $s_0 \equiv f(s_1, \ldots, s_n)$. Then we have necessary $k = m$. If $m = 1$, then we are done: let $v_0 \in T$ be equivalent to s_0 and $v_1 \in T$ be equivalent to u_1. Then $v_0[\diamond]_{p_1}^{N_1} \cdot v_1$ is equivalent to $(s_0)_{p_1}^{N_1} \cdot u_1$ Assume that $p_1, \ldots, p_m$ are listed in increasing order. This is possible since $s_{p,q}^{N,M}$ is equivalent to $s_{q,p}^{M,N}$. We claim that $(s_{p_1,\ldots,p_k}^{N_1,\ldots,N_k} \cdot u_1) \cdot \ldots \cdot u_k$ is weakly equivalent to the term $(s[s_{p_1}^{M_1} \cdot u_1]_{p_1}) \ldots [s_{p_k}^{M_k} \cdot u_k]_{p_k}$ in whcih every occurrence of N_i has been replaced with M_i. For, consider an assignment σ to the integer variables of the terms. We have

$$(s_{p_1,\ldots,p_k}^{N_1,\ldots,N_k} \cdot u_1)\sigma \quad \equiv \quad s_{p_2,\ldots,p_k}^{N_2,\ldots,N_k}\sigma[\underbrace{s\sigma[\ldots[s\sigma[u_1]_{p_1}]\ldots]_{p_1}}_{N_1\sigma-1}]_{p_1}$$

  since $p <_{lex} q$ implies $p^n <_{lex} q^n$ for all positive integers n. And, since $p_1, \ldots, p_k$ are disjoint positions, $p_1^{N_1\sigma}, \ldots p_k^{N_k\sigma}$ are also disjoint positions. Hence

$$(s_{p_1,\ldots,p_k}^{N_1,\ldots,N_k} \cdot u_1 \cdot \ldots \cdot u_k)\sigma \quad \equiv \quad (s_{p_2,\ldots,p_k}^{N_2,\ldots,N_k} \cdot u_2 \ldots \cdot u_k)\sigma[\underbrace{s\sigma[\ldots[s\sigma[u_1]_{p_1}]\ldots]_{p_1}}_{N_1\sigma-1}]_{p_1}$$

  which shows the desired equivalence, by induction on k: it is sufficient to choose $M_i\theta = N_i\sigma - 1$, whatever substitution is considered first. $\square$

8 An unsound Generalisation

Let now T_U be defined by

$$\begin{cases} f(\vec{s}) \in T_U & \Leftarrow \quad \vec{s} \in T_U^{a(f)} \\ x \in T_U & \Leftarrow \quad x \in X \\ \diamond \in T_1 & \\ f(\vec{s_1}, s, \vec{s_2}) \in T_1 & \Leftarrow \quad (\vec{s_1}, s, \vec{s_2}) \in T_U^{n_1} \times T_1 \times T_U^{n_2}, n_1 + n_2 + 1 = a(f) \\ s \cdot t \in T_1 & \Leftarrow \quad s, t \in T_1 \\ s^N \in T_1 & \Leftarrow \quad s \in T_1,\ N \in V_{\mathbf{N}} \\ s \cdot t \in T_U & \Leftarrow \quad s \in T_1,\ t \in T_U \end{cases}$$

And Pos is defined as previously. This definition is close to the definition of T. Te main difference is that we allow now a construction such as $(s^N \cdot s')^M$ which enables to encode the multiplication of integer variables and hence diophantine equations.

Proposition 8.1 *Unification of terms in T_U is undecidable.*

Proof

Encoding diophantine equations is straightforward. We only need one unary symbol f and one constant symbol a in the signature. $code(e)$ is defined on every monomial by:

- $code(1) \stackrel{\text{def}}{=} f(\diamond)$.
- $code(N) \stackrel{\text{def}}{=} f(\diamond)^N$ for a single variable N
- $code(e \times N) \stackrel{\text{def}}{=} code(e)^N$ for every non empty product of variables e and every variable N.

This coding extends to every integer polynomial, setting $code(e + e') \stackrel{\text{def}}{=} code(e) \cdot code(e')$. Then, the integer polynomial equation $P = Q$ has a solution iff $code(P) \cdot a = code(Q) \cdot a$ has a solution. Indeed f can be seen as the successor function: $code(P)$ can be seen as $f(\diamond)^P$. (Although this is not allowed by the above syntax).

For example, $x^2 + 2 \times x \times y + 1$ is encoded by $(f(\diamond)^x)^x \cdot (f(\diamond)^x)^y \cdot (f(\diamond)^x)^y \cdot f(a)$. □

9 Further Remarks

9.1 Complexity issues

We do not know the exact complexity of the problem of unifying two I-terms. However, as in the previous section, it is easy to see that, if we restrict the alphabet to a single unary function symbol and a constant, then the problem of unifying two I-terms is equivalent to solving an arbitrary linear diophantine equation. Hence, for arbitrary alphabets, the problem is at least as hard as solving systems of linear diophantine equations[2], which may have an exponential number of minimal solutions. In the same way, checking the existence of a solution is NP-hard.

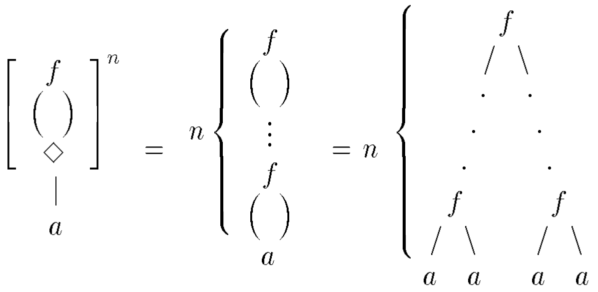
On the other hand, if we only consider one disjunct when applying a rule (i.e. choosing "carefully" which disjunct will lead to a solution), and if we use an appropriate representation of terms (which prevents the exponential behaviour of variable elimination), we conjecture that our algorithms decide the existence of a solution in NP-time. This would mean that the unifiability of I-terms is NP-complete.

9.2 Related work

Since the first version of this paper, we got informed of the work of G. Salzer, which has been done independently from ours [6]. He uses a quite different formalism, but he shows essentially the same results as ours, except that his formalism is slightly more powerful since he can express for example the set of all complete finite binary trees with internal nodes labeled with f. Indeed, his syntax allows for multilple holes in the terms, in which case, the semantics is different from what we considered in section 7. In order to express it shortly, you can assume that the holes

[2] Note that this has nothing to do with the linear diophantine equations which are solved in our unification procedure. Indeed, the diophantine equations that we have to consider in our procedure are not arbitrary linear diophantine systems, and their solution might be simpler to solve than the general case.

are shared, leading to a DAG on which we apply the semantics defined in section 2. For example the instances of $f(\diamond, \diamond)^N \cdot a$ are all complete binary trees:



We believe that there is no important difference if we use DAGs instead of terms. Hence, the technique of our paper should apply to this interpretation as well.

Ackowledgments

I would like to the thank the referees who gave valuable remarks and suggestions.

References

[1] H. Chen and J. Hsiang. Logic programming with recurrence domains. In *Proc. 18th Int. Coll. on Automata, Languages and Programming, Madrid, LNCS 510*, 1991.

[2] H. Chen, J. Hsiang, and H.-C. Kong. On finite representations of infinite sequences of terms. In S. Kaplan and M. Okada, editors, *Proc. CTRS 90*, Montreal, 1990.

[3] H. Comon. Completion of rewrite systems with membership constraints. In W. Kuich, editor, *Proc. 19th Int. Coll. on Automata, Languages and Programming, LNCS 623*, Vienna, 1992. Springer-Verlag. An extended version is available as LRI Research Report number 699, Sept. 1991.

[4] E. Contejean. Éléments pour la décidabilité de l'unification modulo la distributivité. Thèse de Doctorat, Université de Paris-Sud, France, Avril 1992.

[5] J.-P. Jouannaud and C. Kirchner. Solving equations in abstract algebras: A rule-based survey of unification. In J.-L. Lassez and G. Plotkin, editors, *Computational Logic: Essays in Honor of Alan Robinson.* MIT-Press, 1991.

[6] G. Salzer. On unification of infinite sets of terms and its applications. In Voronkov, editor, *Proc. LPAR 92, LNCS 624*, pages 409–421, 1992.