Unification of Terms With Exponents

Rolf Socher-Ambrosius

MPI–I–93–217 May 1993

Unification Of Terms With Exponents

Rolf Socher-Ambrosius*
Max-Planck-Institut für Informatik, Im Stadtwald
W-6600 *Saarbrücken, Germany*
e-mail: socher@mpi-sb.mpg.de

Abstract. In an ICALP (1991) paper, H. Chen and J. Hsiang introduced a notion that allows for a finite representation of certain infinite sets of terms. These so called ω-terms find an application in logic programming, where they can serve to represent finitely an infinite number of answers or to avoid nontermination in certain cases. Another application is in the field of equational logic. Using ω-terms, it is possible to avoid a certain type of divergence of ordered completion. In all cases, unification is the basic computational aspect of this notation. Chen and Hsiang give a complete and terminating unification algorithm for ω-terms. Recently, H. Comon introduced *terms with exponents*, thus significantly extending Chen and Hsiang's notion of ω-terms. He provides a fairly complicated unification algorithm. This paper introduces a further syntactic generalization of Comon's notion together with a comparatively simple inference system for unification.

* This research was funded by the German Ministery for Research and Technology (BMFT) under grant ITS 9103.

1 Introduction

Infinite computations in first order logic often come together with a certain type of recursive data structure. For instance, in logic programming, recursive clauses usually account for looping derivations.

1 Example (*simplified from* [4]) *Consider the following logic program:*

(1) $P(x) \leftarrow R(x), Q(x).$

(2) $R(a).$

(3) $R(f(x)) \leftarrow R(x).$

(4) $Q(d).$

If we ask for the goal $\leftarrow P(x)$*, then the program does not terminate due to infinite backtracking. If the goal is* $\leftarrow R(x)$*, then we obtain an infinite set of answers* $x = a$*,* $x = f(a)$*,* $x = f(f(a))$*,.... Clearly, this case of nontermination is due to the recursive clause* (3).

As another case in point, unification in non-regular order-sorted signatures does not terminate in general [13]. Consider the following example:

2 Example *Assume we are given an order-sorted signature with the sorts* S *and* T *and the following declarations:*

$$a \in S \qquad\qquad a \in T$$
$$f : S \to S \qquad\qquad f : T \to T$$

As the term a *has no unique least sort, this signature is not regular. Accordingly, the unification problem* $x_S = x_T$ *yields the infinite set of solutions* $x_S = x_T = a$*,* $x_S = x_T = f(a)$*,* $x_S = x_T = f(f(a))$*, ... This kind of non-termination is due to the same reasons as in the first example, which can be seen by the relativized* [13] *formulation*

$$P_S(a). \qquad\qquad P_T(b).$$
$$P_S(f(x)) \leftarrow P_S(x). \qquad\qquad P_T(f(x)) \leftarrow P_T(x).$$

In equational logic, one is usually interested in having a convergent rewrite system for a given equational specification. This allows, for instance, to compute values of ground terms, to prove equational consequences, or to solve equations by narrowing techniques. There are well known techniques [2,7] to complete such specifications in order to obtain convergent systems. In many cases, however, this process diverges producing an infinite set of rewrite rules. The following example originates from [1] and [7].

3 Example *We consider the specification of the natural numbers with the gcd-function:*

$x + 0 = x$	$g(x,0) = x$	$g(x+y,y) = g(x,y)$
$x + s(y) = s(x+y)$	$g(0,y) = y$	$g(x,y+x) = g(x,y)$

In this example, ordered completion using a recursive path ordering with precedence + > s diverges. Among others, the following equations are derived:

$$g(s(x+y),s(y)) = g(x,s(y)),\ \ g(s(s(x+y)),s(s(y))) = g(x,s(s(y))),\ \ldots$$

In all these examples, it is the unability to express recursive data structures like $f^n(x)$ in the language of terms that accounts for nontermination. In [1], the problem of diverging completion is tackled by embedding techniques. The base specification is enriched by a copy of the natural numbers. New function symbols are introduced, in our example the two-place function S, in order to express the object $s^n(x)$ by the term $S(n,x)$. Appropriate equations are added to the base specification, such that ordered completion eventually yields a convergent system.

Ohlbach [10] uses a data structure called *abstraction trees with continuations* to represent recursive clauses. The emphasis in this paper lies more in the issue of efficient representation of terms. This data structure does not yield a terminating and complete unification algorithm.

Chen and Hsiang [3,4] take a different approach. They extend the usual notion of terms in order to allow for objects such as $f^n(x)$, or $f(a, g(\cdot))^m(x)$. The computational aspect of those so called *ω-terms* is mirrored by an appropriate unification algorithm. They show that using ω-terms avoids nontermination in cases such as example 1. In this example, the second and third clause would be replaced by the fact $R(f^n(a))$, thus discarding the recursive clause that gives rise to the looping behaviour of the program. However, their notion of ω-terms is unnecessarily restricted in that neither variables in the repeated pattern nor nested repetitions like $f^n(g^m(x))$ are allowed.

Recently, H. Comon [5] introduced *terms with exponents*, thus significantly extending Chen and Hsiang's notion of ω-terms. There is no variable restriction for those terms with exponents, and also nested iterations are allowed. Comon presents a fairly complicated unification algorithm.

Salzer's so called R-terms [11] provide a further generalization of terms with exponents. He can handle term sequences "growing in two dimensions" such as

$$f(x,x),\ f(f(x,x),f(x,x)),\ f(f(f(x,x),f(x,x)),f(f(x,x),f(x,x))),\ldots.$$

In this paper, we adopt Comon's notions for terms with exponents with two slight generalizations: First, we allow *integer expressions* to occur as exponents. For instance, we allow terms like $f^{2n}(x)$, $g^{n+m}(x)$, or $h^{n-m}(x)$. The first two examples actually do not increase the expressive power of the syntax, they can easily be seen equivalent to $(ff)^n(x)$ and $g^n(g^m(x))$, respectively. The term $h^{n-m}(x)$, however, can not appropriately be expressed using Comon's notions. As an example, consider the following recursive logic program:

(1) $P(a, g(g(g(a)))) \leftarrow$

(2) $P(f(x), y) \leftarrow P(x, g(y))$

The original notion of terms with exponents is not adequate to replace the two clauses with a single fact. Using the extended notion, however, the clause $P(f^n(a), g^{3-n}(a))$ does the job. As a second generalization of Comon's terms with exponents, we allow a very restricted form of term sequences "growing in two dimensions". The notion of terms with exponents presented in this paper thus stands between Comon's and Salzer's notions with respect to its expressive power.

We shall present an inference system for unification of terms with exponents, which is based on the ideas presented in [5]. The presentation of the rules, however, emphasizes the similarity to the well-known system for unification of ordinary terms. This results in a straightforward termination proof. Finally, we shall discuss possible extension and limitations of this approach.

2 Notation

We assume given a signature $\mathcal{F}$ of *function symbols*, each function symbol f coming with an arity $arity(f) \geq 0$, a denumerable set $\mathcal{V}$ of *variables*, and a denumerable set $\mathcal{VN}$ of *integer variables* (also called *parameters*), and a symbol $*$. The sets $\mathcal{F}$, $\mathcal{V}$, and $\mathcal{VN}$ are assumed to be mutually disjoint, and the symbol $*$ is not contained in any of these sets. The set $\mathcal{L}$ of (*linear*) *integer expressions* is defined by

$$\mathcal{L} = \{k_0 + k_1N_1 + \ldots + k_nN_n \mid k_j \neq 0 \text{ for } j = 1,\ldots,n\}$$

where the N_i are pairwise different parameters and the k_j are integer constants. Using the common rewrite rules for linear integer arithmetic, it is clear that $\mathcal{L}$ is closed under addition, subtraction and linear multiplication.

We shall use the letters x,y,z for variables, N,M,K for integer variables, and α, β, γ for integer expressions.

4 Definition *The set* $\mathcal{T}$ *of* **terms with exponents** *is the least set that satisfies:*

a) $\mathcal{V} \subseteq \mathcal{T}$

b) *If* $f \in \mathcal{F}$, $arity(f) = n$ *and* $t_1,\ldots,t_n \in \mathcal{T}$, *then* $ft_1\ldots t_n \in \mathcal{T}$

c) *If* $C \in \mathcal{C}-\{*\}$, $u \in \mathcal{T}$, *and* $\alpha \in \mathcal{L}$, *then* $C^\alpha u \in \mathcal{T}$

d) *If* $s \in \mathcal{T}$ *and* $P \subseteq \mathcal{P}os(s)$ *is a nonempty set of positions such that* $s|_{p'} = s|_{q'}$ *holds for any* $p, q \in P$ *and for any* $p' \leq p$, $q' \leq q$ *with* $|p'| = |q'|$, *then* $s[*]_P \in \mathcal{C}$

e) $\mathcal{P}os(x) = \{\Lambda\}$

f) $\mathcal{P}os(ft_1 \ldots t_n) = \{\Lambda\} \cup 1\cdot\mathcal{P}os(t_1) \cup \ldots \cup n\cdot\mathcal{P}os(t_n)$

g) $\mathcal{P}os(C^{\alpha}\, u) = \emptyset$

In part d, $s[*]_P$ means the term obtained from s by replacing $s|_p$ by $*$ for each $p \in P$. The elements of $\mathcal{C}$ are called **contexts**. *The* **characteristic** *of the context C is defined by* char$(C) = P$. *Two contexts C and D of the same characteristic are called similar, written $C \sim D$. The context $C = *$ is called* **trivial**. *If $C = s[*]_P$, and u is a term or context, then Cu denotes the term or context $s[u]_P$. We call a term of the form $C^{\alpha}\, u$ an* **N-term**.

From the definition, it is clear the set $\mathcal{C}$ of contexts together with the concatenation operation and the trivial context as identity is isomorphic to the free semigroup generated by the set $\mathcal{C}^1$ of contexts of length 1. Each context C of length n, $n \geq 0$, can thus uniquely be represented in the form $c_1 \ldots c_n$, $n \geq 0$, with $c_i \in \mathcal{C}^1$ for $i = 1,\ldots,n$. The number n is called the *length* of the context C, written $|C|$. As an example, the context $C = f(g(h(*,a)),g(h(*,a)))$ can be represented in the form $f(*,*)g(*)h(*,a)$. Thus $|C| = 3$. If $C = c_1 \ldots c_n$ is a nontrivial context, then $\mathrm{char}_1(C)$ is defined as $\mathrm{char}(c_1)$.

A context C is called a *prefix* of the term or context t, if $t = Ct'$ for some t'.

5 Example *a) The expression $t = f(*)^N\, x$ is an element of $\mathcal{T}$. A more common way to write such a term is the form $t = f^N(x)$.*

b) The expression $g(f(*),f(*))^N x$ is a term.

A construction such as $(f(*)^N g(*))^M\, x$ cannot be represented using the syntax above, because the set of positions of the term $f(*)^N g(y)$ is empty.

For technical reasons, we define extensions $\mathcal{L}^*$ and $\mathcal{T}^*$ of $\mathcal{L}$ and $\mathcal{T}$, respectively, as follows: $\mathcal{L}^* = \mathcal{L} \cup \{\alpha + q \mid \alpha \in \mathcal{L}, q \in \mathbb{Q}\}$ and $\mathcal{T}^* = \mathcal{T} \cup \{C^{\alpha - k/c}\, u \mid C \in \mathcal{C}, u \in \mathcal{T}, c = |C|, 1 \leq k < c\}$. Obviously, any exponent $\alpha \in \mathcal{L}^*$ can be written uniquely in the form $\alpha = \alpha' + \alpha''$ with $\alpha' \in \mathcal{L}$ and $\alpha'' \in \mathbb{Q}$, $0 \leq \alpha'' < 1$.

We shall usually refer to the elements of $\mathcal{T}^*$ simply as terms. Usual terms, that is, elements of $\mathcal{T}$ having no exponents, will also be called *simple* terms.

We shall often work with examples where only monadic functions occur. In this case we use a more convenient notation that resembles the well-known notation $f^n(x)$. We shall, for instance, write $(fg)^N gh^M x$ rather than $(f(g(*)))^N g(h(*))^M\, x$.

We define the *set of variables* of a term t by

a) $\mathcal{V}ar(x) = \{x\}$

b) $\mathcal{V}ar(f(t_1,\ldots,t_n)) = \mathcal{V}ar(t_1) \cup \ldots \cup \mathcal{V}ar(t_n)$

c) $\mathcal{V}ar(C^{\alpha}u) = \mathcal{V}ar(C) \cup \mathcal{V}ar(u)$

We define the *leading function symbol* $\mathscr{Head}(t)$ of a nonvariable term t by $\mathscr{Head}(ft_1...t_n) = f$ and $\mathscr{Head}(C^{\alpha}u) = \mathscr{Head}(C)$.

The intended meaning of the construction C^{α} is an α-fold iteration of the context C. The following definition captures this meaning.

6 Definition *a) For any nontrivial context C, we define*

$$C^0 = *$$
$$C^a = CC^{a'} \text{ where } a \in \mathbb{Q},\ a \geq 1 \text{ and } a' = a-1$$
$$C^{k/c} = C' \text{ where } c = |C| \text{ and } C = C'D \text{ with } |C'| = k$$

A term t is called *$\mathscr{VA}$-ground*, iff t contains no parameters, and *positive*, iff additionally it is equivalent to a simple term. A $\mathscr{VA}$-ground term that is not positive is called *negative*. As a matter of simplicity, we shall not distinguish between a positive $\mathscr{VA}$-ground term and its simple counterpart. We shall thus assume in the following that each exponent α is an element of $\mathscr{L} - \mathbb{N}$.

An *equation* is an unordered pair (s,t) of terms, written in the form $s = t$, or an unordered pair $\alpha = \beta$ of integer expressions. An *equational system E* is a finite conjunction of equations.

A *substitution* σ on $\mathscr{T}$ is a pair (σ_1,τ) of a mapping $\sigma_1: \mathscr{V} \rightarrow \mathscr{T}$ and an endomorphism $\tau: \mathscr{L} \rightarrow \mathscr{L}$. The application of σ to terms is defined by

a) $x\sigma = x\sigma_1$ if $x \in \mathscr{V}$

b) $(ft_1...t_n)\sigma = ft_1\sigma...t_n\sigma$

c) $(C^{\alpha} u)\sigma = (C\sigma)^{\alpha\sigma} u\sigma$

d) $\alpha\sigma = \alpha\tau$ for $\alpha \in \mathscr{L}$.

Of course, we are interested only in the initial model of $\mathscr{L}$. Therefore, we introduce a congruence $\cong$ on non-negative terms by $s \cong t$ iff $s\sigma = t\sigma$ holds for all positive $\mathscr{VA}$-ground instances $s\sigma$ of s and for all positive $\mathscr{VA}$-ground instances $t\sigma$ of t. In [4], the relation $\cong$ is called *strong equivalence*.

From Definition 6, we immediately obtain the relations $C^{a+b} = C^a\, C^b$ for all a, $b \in \mathbb{N}$, $C^{na}\, u = (C^n)^a\, u$ for all a, $n \in \mathbb{N}$, and $C(DC)^a = (CD)^a\, C$ for all $a \in \mathbb{N}$, and henceforth the following lemma.

7 Lemma *For all contexts C, D and terms u, for all $n \in \mathbb{N}$, and for all α, $\beta \in \mathscr{L}$, the following holds:*

a) $C^{\alpha+\beta}\, u \cong C^{\alpha}\, C^{\beta}\, u$ *and* $C^{n\alpha}\, u \cong (C^n)^{\alpha}\, u$

b) $C^{n\alpha}\, u \cong (C^n)^{\alpha}\, u$

c) $C(DC)^{\alpha}\, u \cong (CD)^{\alpha}\, Cu$

The following lemma follows easily from properties of semigroups, too.

8 Lemma *Let C be a context, let u and v be contexts or terms, and let n,m be natural numbers such that $n \geq m$. Then*

$$C^m\ u = C^n\ v \text{ iff } u = C^{n-m}\ v$$

The substitution σ is a *unifier* of the equation $s = t$, iff $s\sigma$ and $t\sigma$ are $\mathcal{VN}$-ground terms with $s\sigma = t\sigma$. An equational system

$$E \equiv x_1 = t_1 \wedge \ldots \wedge x_n = t_n \wedge N_1 = \alpha_1 \wedge \ldots \wedge N_k = \alpha_k \quad (1)$$

is in *solved form*, iff each integer variable N_i and each variable x_j occurs only once in E. The system E is a (particular) *solution* of the system E', iff additionally each unifier of E is also a unifier of E'. The set of all solutions of a system E' will be denoted by Sol(E'). The set $\mathcal{E}$ of equational systems in solved form is a *general solution* of E, iff $\mathrm{Sol}(E) = \bigcup_{E' \in \mathcal{E}} \mathrm{Sol}(E')$.

3 An Inference System for Unification

For unification of ordinary terms, the decomposition rule plays an important role. Decomposition is a transformation of an equation $\{fs_1 \ldots s_n = ft_1 \ldots t_n\}$ into a system $\{s_1 = t_1, \ldots, s_n = t_n\}$ of equations between the immediate subterms. In order to introduce decomposition for terms with exponents, we first have to define an appropriate notion of immediate subterms. Consider the following example: Let

$$t = (fg)^N\ x\ = (fg)(fg)\ldots(fg)\ x$$

Then the immediate subterm of t (at position 1) can be written

$$(gf)(gf)\ldots(gf)\ gx = (gf)^{N-1}\ gx = (gf)^{N-1/2}\ x$$

This leads to the following definitions:

9 Definition (Immediate Subterm) *a) We define a mapping ρ on contexts by*

$$*\rho = *$$
$$(c_1 \ldots c_n)\rho = c_2 \ldots c_n c_1$$

b) Let t be a term with $\mathcal{H}ead(t) = f$, and let $1 \leq i \leq$ arity(f). If t is not an N-term, we define $t{\downarrow_i} = t|_i$. If $t = C^\alpha u$ is an N-term, we define

$$t\downarrow_i = \begin{cases} (C\rho)^{\alpha - 1/c}\, u & \text{if } i \in char_1(C) \\ C\,|_i & \text{otherwise} \end{cases}$$

where $c = |C|$.

10 Example *Consider the term $t = (f(g(*),b))^N x$. (See fig. 1a). According to definition 9, the immediate subterm of t is (See fig. 1b)*

$$t{\downarrow_1} = g(f(*,b))^{N-1/2}(x)$$

11 Lemma *Let t be a term with leading function symbol f. Then*

$$t \cong f(t{\downarrow}_1,\ldots,t{\downarrow}_n)$$

Proof. The assertion of the lemma is obvious, if t is not an N-term. So let $t = C^{\alpha}u$ with $C \neq *$. First, we prove that $t{\downarrow}_i \cong (CC^{\alpha-1}u)\big|_i$ holds for $i = 1,\ldots,n$. This will readily imply the assertion of the lemma, since

$$f((CC^{\alpha-1}u)\big|_1,\ldots,(CC^{\alpha-1}u)\big|_n) = CC^{\alpha-1}u \cong C^{\alpha}u = t$$

So let $i \in \{1,\ldots,n\}$, and let $C = c_1D$ with $c_1 \in \mathcal{C}^1$. If $i \notin \mathrm{char}_1(C)$, then

$$t{\downarrow}_i = C\big|_i = (CC^{\alpha-1}u)\big|_i$$

If $i \in \mathrm{char}_1(C)$, then $C\big|_i = D$, and we obtain from definition 9,

$$t{\downarrow}_i = (C\rho)^{\alpha-1/c}\ u = (Dc_1)^{\alpha-1/c}\ u \cong (Dc_1)^{\alpha-1}\ (Dc_1)^{(c-1)/c}\ u \cong$$
$$(Dc_1)^{\alpha-1}\ D\ u \cong D(c_1D)^{\alpha-1}\ u = C\big|_i\ C^{\alpha-1}\ u = (CC^{\alpha-1}u)\big|_i$$

which concludes the proof of the lemma. □

With this notion of an immediate subterm, the usual inference system for uni-

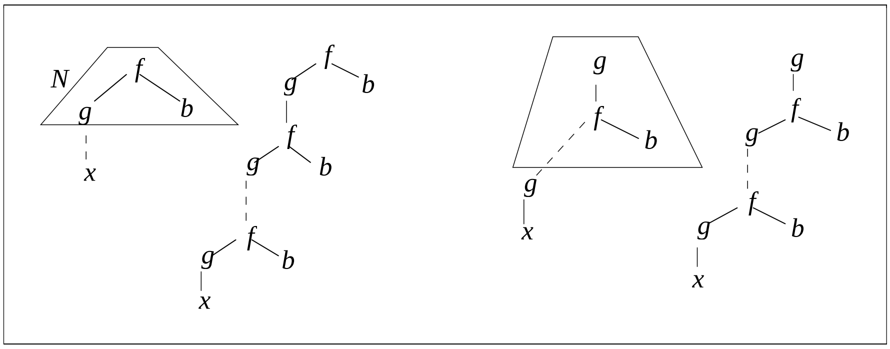

Figure 1a *Figure 1b*

fication, consisting of the rules Trivial, Clash, Merge, Occur-Check and Decomposition, is almost sufficient for unification of terms with exponents. We only have to add an inference rule (Eliminate) that eliminates integer variables. We thus obtain a system that enumerates all solutions of a given equation. This system, however, is in general not terminating.

12 Definition (Inference System I_1 for Unification) *The system I_1 consists of the rules shown in fig.* 2. *We write $\Rightarrow$ for the derivation relation w.r.t. I_1.*

Trivial	$\dfrac{t=t}{\top}$
Clash	$\dfrac{s=t}{\bot}$
if $\mathcal{H}ead(s) \neq \mathcal{H}ead(t)$	
Merge	$\dfrac{x=t \wedge P}{x=t \wedge P\{x \leftarrow t\}}$
if $x \notin \mathcal{V}ar(t)$, $x \in \mathcal{V}ar(P)$ *and* $t \in \mathcal{V} \Rightarrow t \in \mathcal{V}ar(P)$	
Occur-Check	$\dfrac{x=t}{\bot}$
if $x \in \mathcal{V}ar(t)$	
Decompose	$\dfrac{s=t}{s{\downarrow}_1=t{\downarrow}_1 \wedge \ldots \wedge s{\downarrow}_n=t{\downarrow}_n}$
if $\mathcal{H}ead(s) = \mathcal{H}ead(t)$ *has arity* n	
Eliminate	$\dfrac{C^{\alpha}u=t}{\alpha=0 \wedge u=t}$
if $\alpha \in \mathcal{L}$	

Figure 2

The rules of system I_1 are to be understood don't-know-nondeterministic, that is, all alternative rule applications have to be explored in order to find all solutions. The symbols $\bot$ and $\top$ denote the logical values false and true, respectively. Moreover, it should be noted that equations are defined as unordered pairs of terms, and consequently the inference rules Merge and Occur Check also apply to equations of the form $t = x$, and Eliminate also applies to equations of the form $t = C^{\alpha}\, u$.

13 Example *a) Consider the equation*

$$f^N g f^M x = (fg)^K y \tag{1}$$

The following successful derivations compute all solutions:

$$f^N g f^M x = (fg)^K y \quad \Rightarrow_E \quad K = 0 \wedge y = f^N g f^M x$$

$$\Downarrow_D$$

$$f^{N-1} g f^M\, x = (gf)^{K-1/2}\, y$$

$$\Downarrow_E$$

$$N{=}1 \wedge g f^M\, x = (gf)^{K-1/2}\, y$$

$$\Downarrow_D$$

$$N{=}1 \wedge f^M x = (fg)^{K-1} y \quad \Rightarrow_E \quad N{=}1 \wedge M{=}0 \wedge x = (fg)^{K-1} y$$

$$\Downarrow_{\mathrm{D}} \qquad\qquad \Downarrow_{\mathrm{D}}$$

$$N{=}1 \wedge f^{M-1} x = (gf)^{K-1-1/2} y \qquad N{=}1 \wedge K{=}1 \wedge f^M x = y$$

$$\Downarrow_{\mathrm{E}}$$

$$N{=}1 \wedge M = 1 \wedge x = (gf)^{K-1-1/2} y$$

There are other derivations, such as

$$f^N g f^M x = (fg)^K y \qquad \Rightarrow_{\mathrm{D}} f^{N-1} g f^M x = (gf)^{K-1/2} y \qquad \Rightarrow_{\mathrm{C}} \bot$$

which, however, do not yield solutions. Moreover, all I_1-derivations are terminating.

b) *Consider the equation*

$$(ff)^N x = f^M y \tag{2}$$

There is an infinite derivation

$$\begin{aligned} (ff)^N x = f^M y \quad & \Rightarrow_{\mathrm{D}} (ff)^{N-1/2} x = f^{M-1} y \Rightarrow_{\mathrm{D}} (ff)^{N-1} x = f^{M-2} y \Rightarrow_{\mathrm{D}} \dots \\ & \Rightarrow_{\mathrm{D}} (ff)^{N-k} x = f^{M-2k} y \Rightarrow_{\mathrm{D}} \dots \end{aligned}$$

The solutions of (2) *are simply*

$$x = f^{M-2N} y \text{ and } y = f^{2N-M} x$$

The following lemma describes the properties of inference system I_1.

14 Lemma *If $E \Rightarrow E'$, then $Sol(E') \subseteq Sol(E)$.*

Proof. It is obviously sufficient to prove the assertion for $E' \neq \bot$ and $E' \neq \top$. If $E \Rightarrow E'$ by application of the Merge rule, then the assertion holds trivially, too. Now let $E \Rightarrow E'$ by an application of Decompose to some equation $s = t$ with $\mathcal{Head}(s) = \mathcal{Head}(t) = f$, and let $\sigma \in \mathrm{Sol}(E')$. Then, according to lemma 11

$$s\sigma = f(s{\downarrow}_1,\dots,s{\downarrow}_n)\sigma = f(s{\downarrow}_1\sigma,\dots,s{\downarrow}_n\sigma) = f(t{\downarrow}_1\sigma,\dots,t{\downarrow}_n\sigma) = f(t{\downarrow}_1,\dots,t{\downarrow}_n)\sigma = t\sigma$$

holds, which implies $\sigma \in Sol(E)$. Finally, let $E \Rightarrow E'$ by an application of Eliminate to some equation $C^\alpha u = t$ yielding $\alpha = 0 \wedge u = t$. Let $\sigma \in Sol(E')$, that is, $\alpha\sigma = 0$ and $u\sigma = t\sigma$. Then

$$(C^\alpha u)\sigma = (C\sigma)^{\alpha\sigma} u\sigma = (C\sigma)^0 u\sigma = u\sigma = t\sigma$$

which implies $\sigma \in Sol(E)$. □

15 Lemma *The system I_1 is correct, that is, if $\mathcal{E}$ is the set of immediate successors of E w.r.t. $\Rightarrow$, then $Sol(E) = \bigcup_{E' \in \mathcal{E}} \mathrm{Sol}(E')$.*

Proof. We show that the assertion of the lemma holds, if $\mathcal{E}$ is the set of immediate successors of E w.r.t. application of a rule to one equation $s = t$ in E. We can obviously assume that $s \neq t$.

Case 1: One of s, t, say s, is a variable, say $s = x$. If $x \in \mathcal{Var}(t)$, then only the Occur-Check rule applies and $Sol(E) = Sol(\bot) = \emptyset$. If $x \notin \mathcal{Var}(t)$, then only Merge applies and $Sol(E) = Sol(E')$, where $\mathcal{E} = \{E'\}$.

Case 2: None of s and t is a variable. If none of s and t is an N-term, then either decomposition or Clash applies, both of which is correct. So suppose at least one of s, t, is an N-term, say $s = C^\alpha u$. If $\mathcal{Head}(s) \neq \mathcal{Head}(t)$, then Clash and Eliminate apply, and $\mathcal{E} = \{\bot, E'\}$, where $E' = E - \{s = t\} \cup \{\alpha = 0 \wedge u = t\}$. Let $\sigma \in Sol(E)$, and suppose $\alpha\sigma \neq 0$. Then

$$t\sigma = (C^\alpha u)\sigma = (C\sigma)^{\alpha\sigma} u\sigma = C\sigma(C\sigma)^{\alpha\sigma-1} u\sigma$$

But this is impossible, since $\mathcal{Head}(C) = \mathcal{Head}(s) \neq \mathcal{Head}(t)$ by assumption. Hence $\alpha\sigma = 0$ must hold. This implies $u\sigma = t\sigma$ and σ is a unifier of $\{\alpha = 0 \wedge u = t\}$, that is, $\sigma \in Sol(E')$.

Finally, let $\mathcal{Head}(s) = \mathcal{Head}(t)$. Then Decompose and Eliminate apply, and $\mathcal{E} = \{E_1, E_2\}$, where $E_1 = E - \{s = t\} \cup \{s{\downarrow}_1 = t{\downarrow}_1 \wedge \ldots \wedge s{\downarrow}_n = t{\downarrow}_n\}$, and $E_2 = E - \{s = t\} \cup \{\alpha = 0 \wedge u = t\}$. Let $\sigma \in Sol(E)$. If $\alpha\sigma = 0$, then $t\sigma = (C^\alpha u)\sigma = (C\sigma)^{\alpha\sigma} u\sigma = u\sigma$ and σ thus solves E_2. So let $\alpha\sigma \neq 0$. Then $(\alpha-1)\sigma \geq 0$ and, according to lemma 11, σ solves E_1. □

Having solved equations between terms, it remains to solve the parameter part. Since we are interested only in solutions having nonnegative exponents, we have to add to the parameter part of a solved system E a set of inequations of the form $\{\alpha \geq 0 \mid \alpha$ occurs as exponent in $E\}$. We assume given an inference system $\mathcal{D}$ for solving linear diophantine equations and inequations. By $\Rightarrow_1$, we denote the inference relation given by the system $I_1 \cup \mathcal{D}$. Now, since irreducible systems w.r.t. $\Rightarrow_1$ are obviously in solved form, we have the following

16 Corollary *If $E \Rightarrow_1^* E^*$, such that E^* is irreducible by I_1, then E^* is a solution of E.*

It can even be shown that whenever E^* is a solution of E, then there is a derivation $E \Rightarrow_1^* E^*$.

Obviously, decomposition is the only rule causing nontermination of the inference system I_1. Moreover, each nonterminating derivation contains a cyclic subderivation of the form

$$P \wedge C^\alpha u = D^\beta v \Rightarrow \ldots \Rightarrow P' \wedge C^{\alpha-n} u = D^{\beta-m} v$$

consisting only of Decompose steps and such that $|C| \cdot n = |D| \cdot m = \mathrm{lcm}(|C|,|D|)$ and $C^n \sim D^m$. Such a decomposition issuing an infinite derivation can be restricted to the length $\mathrm{lcm}(|C|,|D|)$.

In the following, whenever $C, D \in \mathcal{C}$, we use $[C,D]$ to denote the number n such that $|C|^n = \mathrm{lcm}(|C|,|D|)$.

17 Definition (Inference System I_2 for Unification) *The system I_2 consists of* Trivial, Clash, Merge, Occur-Check, Eliminate *plus the rules shown in fig.* 3. *In the Induce rule, B is the prefix of D of length* $|B| = \mathrm{gcd}(|C|,|D|)$. *In the first three rules* $\alpha' \in \mathcal{L}$, $\alpha'' \in \mathbb{Q}$, $0 \leq \alpha'' < 1$.

Before proving termination and correctness of the inference system I_2, we give some examples to show how the system works.

18 Examples *a) Consider the unification problem*

$$(ff)^N\, x = (fff)^M\, y$$

As $(ff)^3 \sim (fff)^2$, *the Induce rule applies. The problem thus has the solutions*

$$\{x = f^{3M-2N}\, y\},\ \{y = f^{2N-3M}\, x\}$$

b) Consider the problem $f(*,*)^N\, x = f(*,y)^M\, a$. *Then* $f(*,*) \not\sim f(*,y)$ *and* $1 \in \mathrm{char}(C) \cap \mathrm{char}(D)$. *Eliminate and Eliminate2 thus yield the derivations*

$$N = 0 \wedge x = f(*,y)^M\, a$$

$$N = 1 \wedge f(x,x) = f(*,y)^M\, a \Rightarrow N = 1 \wedge x = y = f(*,y)^{M-1}\, a$$

$$N = 2 \wedge f(f(x,x),f(x,x)) = f(*,y)^M\, z \Rightarrow x = f(*,y)^{M-2}\, z \wedge x = y \wedge f(x,x) = y \Rightarrow \bot$$

The set $\{\{N = 0 \wedge x = f(*,y)^M\, a\},\{N = 1 \wedge x = y = f(*,y)^{M-1}\, a\}\}$ *is a general solution for the problem.*

19 Lemma *If there exist* $n,m \geq 1$, *such that* $C^n \sim D^m$, *then* $n' := [C,D]$ *and* $m' := [D,C]$ *are the smallest such numbers.*

20 Lemma *The system I_2 is terminating.*

Proof. We define a complexity measure for a given equational system E. For any equational system E, let

$$u(E) = \{s = t \in E \mid s,t \in \mathcal{T} - \mathcal{V}\}$$

that is, $u(E)$ is the unsolved pure part of E. We define $v(E) := |\mathcal{V}ar(u(E))|$

For each term t, $\chi(t)$ is the total number of exponents occurring in t, that is:

$\chi(x) = 0$ if x is a variable

$\chi(C^\alpha\, u) = 1 + \chi(C) + \chi(u)$

$\chi(ft_1 \ldots t_n) = \sum_{i=1}^{n} \chi(t_i)$

Let $\chi(s = t) := \chi(s) + \chi(t)$. The *size* of a term t is defined to be the number of its positions, that is, $\mathrm{size}(t) = |\mathcal{P}os(t)|$, and $\mathrm{size}(s = t) := \mathrm{size}(s) + \mathrm{size}(t)$.

The following relations obviously hold for any context C and any term t:

Eliminate1	$\dfrac{C^{\alpha}u{=}D^{\beta}v}{\alpha'{=}i \wedge C^{i+\alpha''}u{=}D^{\beta}v}$

if $\alpha = \alpha' + \alpha''$, $\exists n,m \geq 1$: $C^n \sim D^m$ *and* $0 < i < n$

Eliminate2	$\dfrac{C^{\alpha}u{=}D^{\beta}v}{\alpha'{=}i \wedge C^{i+\alpha''}u{=}D^{\beta}v}$

if $\alpha = \alpha' + \alpha''$, $\exists n,m \geq 1$: $C^n \not\sim D^m$, $\mathrm{char}(C^n) \cap \mathrm{char}(D^m) \neq \emptyset$ *and* $0 < i \leq 2n$

Induce	$\dfrac{C^{\alpha}u{=}D^{\beta}v}{C^n{=}D^m \wedge C^{\alpha''}u{=}B^{n\beta-m\alpha'}v}$

if $\alpha = \alpha' + \alpha''$, $\exists n,m \geq 1$: $C^n \sim D^m$ *and* $n\beta - m\alpha$ *is not a negative number*

Decompose1	$\dfrac{s{=}t}{s{\downarrow_1}{=}t{\downarrow_1} \wedge \ldots \wedge s{\downarrow_n}{=}t{\downarrow_n}}$

if $\mathcal{H}ead(s) = \mathcal{H}ead(t)$, *and Eliminate*1 *or Induce does not apply*

Figure 3

$\chi(Ct) = \chi(C) + \chi(t)$
$\chi(t) \geq \chi(t{\downarrow_j})$, whenever $t{\downarrow_j}$ is defined.
$\chi(t) > \chi(t{\downarrow_j})$, if $t = C^{\alpha}\, u$ and $j \notin \mathrm{char}_1(C)$

If $C = c_1 \ldots c_n$ and $D = d_1 \ldots d_m$ are contexts, then the length of the maximal similar prefix, $msp(C,D)$, is defined to be the maximal number r such that $c_1 \ldots c_r \sim d_1 \ldots d_r$, and $\pi(s = t)$ is defined to be $r(C^{[C,D]}, D^{[D,C]})$, if $s = C^{\alpha}\, u$ and $t = D^{\beta}\, v$ both are N-terms and $\pi(s = t) = 0$ otherwise.

For any equational system E, we define the multiset

$$X(E) = \{(\chi(e),\ size(e),\ \pi(e)) \mid e \in u(E)\}$$

We define $>_3$ to be the threefold lexicographic combination of the ordering $>$ on natural numbers. Finally, we define an ordering $>_e$ on equational systems by $E >_e E'$, iff $(v(E),X(E)) >_2 (v(E'),X(E'))$, where $>_2$ is the lexicographic combination of $>$ with the multiset extension of $>_3$. By construction, it is clear that the ordering $>_e$ is well-founded. We show that $E \Rightarrow_2 E'$ implies $E' = \perp$ or $E >_e E'$, which in turn will prove the assertion of the lemma.

First of all, we remark that application of a rule of I_2 to E cannot increase $v(E)$, that is, $E \Rightarrow_2 E'$ implies $v(E) \geq v(E')$. Therefore, it is sufficient to show that $E \Rightarrow_2 E'$ implies $v(E) > v(E')$ or $X(E)\ (>_3)_{mul}\ X(E')$. If $E \Rightarrow_2 E'$ by an application of *Merge*, then obviously $v(E) > v(E')$. In all other cases, we show that $X(E)\ (>_3)_{mul}\ X(E')$ holds. First, if $E \Rightarrow_2 E'$ by an application of *Trivial*, then $u(E') \subset u(E)$, which implies $X(E)\ (>_3)_{mul}\ X(E')$.

	χ	size	lcp
1 Eliminate/Eliminate1	<		
2 Induce	<		
3 Decompose1 (s,t)	≤	<	
4a Decompose1 $(C^\alpha u, t)$	≤	<	
4b Decompose1 $(C^\alpha u, t)$	<		
5a Decompose1 $(C^\alpha u, D^\beta v)$	<		
5b Decompose1 $(C^\alpha u, D^\beta v)$	≤	≤	<

Table 1

The remaining results are summarized in table 1. An entry ≤ (<) at column i, row j, means that the measure i is not increased (is decreased) by rule j. The different cases for application of Decompose1 (row 3 to 5b) are explained below.

To justify the entries in the table, we first note that obviously no application of an inference rule of I_2 to a system E can increase $\chi(E)$.

Case 1: $E \Rightarrow_2 E'$ by an application of *Eliminate*, *Eliminate*1, or *Eliminate*2 to an equation $C^\alpha u = t$, yielding $\alpha = 0 \wedge u = t$. Then

$$\chi(C^\alpha u) = 1 + \chi(C) + \chi(u) > \chi(u)$$

Case 2: $E \Rightarrow_2 E'$ by an application of *Induce*. Similar to case 2.

Case 3: $E \Rightarrow_2 E'$ by an application of *Decompose*1 to some equation $s = t$, where none of s and t is an N-term. This implies $\mathrm{size}(s{\downarrow_i}) < \mathrm{size}(s)$ and $\mathrm{size}(t{\downarrow_i}) < \mathrm{size}(t)$ for i=1,…,n.

Case 4: $s = C^\alpha u$ is an N-term and t is not an N-term. a) For $i \in \mathrm{char}_1(C)$, we have $\mathrm{size}(s) = \mathrm{size}(s{\downarrow_i}) = 0$ and

$$\mathrm{size}(s{\downarrow_j} = t{\downarrow_j}) = \mathrm{size}(t{\downarrow_j}) < \mathrm{size}(t) = \mathrm{size}(s = t)$$

b) If $i \notin \mathrm{char}_1(C)$, then $\chi(s{\downarrow_i}) < \chi(s)$, which implies

$$\chi(s{\downarrow_i} = t{\downarrow_i}) < \chi(s = t)$$

Case 5: Both $s = C^\alpha u$ and $t = D^\beta v$ are N-terms. Since Eliminate1 or Induce do not apply, we can assume that $C^n \not\leftarrow D^m$ for all $m,n \geq 1$. Let $C^n = c_1 \ldots c_k$, $D^m = d_1 \ldots d_k$ with $c_1,\ldots,c_k, d_1,\ldots,d_k \in \mathcal{C}$.

a) $i \notin \mathrm{char}(c_1)$ or $i \notin \mathrm{char}(d_1)$. We obtain

$$\chi(s{\downarrow_i}) < \chi(s), \text{ hence } \chi(s{\downarrow_i} = t{\downarrow_i}) < \chi(s = t) \text{ for } i \notin \mathrm{char}(c_1)$$

$\chi(t{\downarrow_i}) < \chi(t)$, hence $\chi(s{\downarrow_i} = t{\downarrow_i}) < \chi(s = t)$ for $i \notin \mathrm{char}(d_1)$

b) $i \in \mathrm{char}(c_1) \cap \mathrm{char}(d_1)$. Then $c_1 \sim d_1$ and $s{\downarrow_i} = (C\rho)^{\alpha-1/c}\, u$ and $t{\downarrow_i} = (D\rho)^{\beta-1/d}\, v$, where $c = |C|$ and $d = |D|$.

First, we remark that $\mathrm{size}(s{\downarrow_i} = t{\downarrow_i}) = 0 = \mathrm{size}(s = t)$. Let $r = \pi(s = t) = \mathrm{msp}(C^n, D^m)$. From $c_1 \sim d_1$ and $C^n \not\sim D^m$ follows $1 \le r < k$. Then

$$\pi(s{\downarrow_i} = t{\downarrow_i}) = \mathrm{msp}((C\rho)^n, (D\rho)^m) = \mathrm{msp}(C^n\rho, (D\rho)^m)$$
$$= \mathrm{msp}(c_2 \ldots c_k c_1,\ d_2 \ldots d_k d_1) = r-1 < r = \pi(s = t)$$ □

21 Lemma *Let C and D be simple contexts, such that $C^n = D^m$ holds for some $n,m \ge 1$. Then there exists a common prefix B of C and D, such that*

$$C = B^m \text{ and } D = B^n$$

and $|B| = \gcd(|C|,|D|)$.

Proof. Since contexts can be regarded as words, the assertion of the lemma follows from a well known lemma on semigroups (see, e.g., [9]) □

22 Lemma *Let C and D be nontrivial contexts. If $C \not\sim D$ and $\mathrm{char}(C) \cap \mathrm{char}(D) \neq \emptyset$, then $C^2\, u \neq D^2\, u$ for all terms u.*

Proof. Assume to the contrary that $C^2\, u = D^2\, u$, and let $p \in \mathrm{char}(C) \cap \mathrm{char}(D)$. This implies $C^2|_p = C$ and $D^2|_p = D$. Moreover, we can assume without loss of generality that there exists some $q \in \mathrm{char}(C) \setminus \mathrm{char}(D)$. Then (see fig. 4)

$$u = C|_q = C^2|_{pq} = D^2|_{pq} = D|_q$$

$$Cu = C^2|_q = D^2|_q = D|_q$$

which is a contradiction, because $C \neq *$ by assumption.

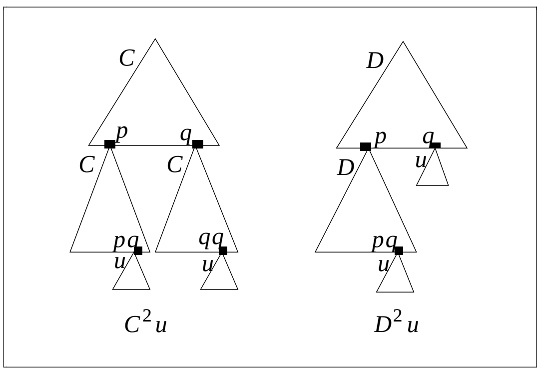

Figure 4

23 Lemma *If $E \Rightarrow_2 E'$, then* $Sol(E') \subseteq Sol(E)$.

Proof. According to lemma 14, it is sufficient to prove the assertion for the Eliminate1, Eliminate2 and the Induce rule.

In view of the relation $C^\alpha\, u = C^{\alpha' + \alpha''}\, u \cong C^{\alpha'}\, C^{\alpha''}\, u$, we can assume without loss of generality in the following that $\alpha \in \mathscr{L}$, hence $\alpha = \alpha'$ and $\alpha'' = 0$ holds.

Case 1: $E \Rightarrow E'$ by application of Eliminate1 or Eliminate2. Let $\sigma \in \mathrm{Sol}(\alpha = i \wedge C^i\, u = D^\beta\, v)$. Then $\alpha\sigma = i$ and

$$(C^\alpha\, u)\sigma = (C\sigma)^{\alpha\sigma}\, u\sigma = (C\sigma)^i\, u\sigma = (D\sigma)^{\beta\sigma}\, v\sigma = (D^\beta\, v)\sigma$$

which implies $\sigma \in \mathrm{Sol}(C^\alpha\, u = D^\beta\, v)$.

Case 2: $E \Rightarrow E'$ by application of Induce. Let $\sigma \in \mathrm{Sol}(C^n = D^m \wedge u = B^{n\beta - m\alpha})$, that is

$$(C\sigma)^n = (D\sigma)^m \text{ and} \tag{1}$$

$$u\sigma = (B\sigma)^{n\beta\sigma - m\alpha\sigma}\, v\sigma \tag{2}$$

From (2) and lemma 8 follows

$$(B\sigma)^{m\alpha\sigma}\, u\sigma = (B\sigma)^{n\beta\sigma}\, v\sigma \tag{3}$$

By lemma 8, we can infer

$$((B\sigma)^m)^{\alpha\sigma}\, u\sigma \cong ((B\sigma)^n)^{\beta\sigma}\, v\sigma \tag{4}$$

Since $B\sigma$ is a prefix of $D\sigma$ of length $\gcd(|C|,|D|)$, we can conclude by (1) and lemma 21, that $C\sigma \cong (B\sigma)^m$ *and* $D\sigma \cong (B\sigma)^n$, hence

$$(C\sigma)^{\alpha\sigma}\, u\sigma \cong ((B\sigma)^m)^{\alpha\sigma}\, u\sigma \cong ((B\sigma)^n)^{\beta\sigma}\, v\sigma \cong (D\sigma)^{\beta\sigma}\, v\sigma$$

which finally yields $\sigma \in \mathrm{Sol}(C^\alpha\, u = D^\beta\, v)$. □

24 Theorem *The system I_2 is correct, that is, if $\mathscr{E}$ is the set of immediate successors of E w.r.t. $\Rightarrow$, then $Sol(E) = \bigcup_{E' \in \mathscr{E}} \mathrm{Sol}(E')$.*

Proof. We show that the assertion of the lemma holds, if $\mathscr{E}$ is the set of immediate successors of E w.r.t. application of a rule to one nontrivial equation $s = t$ in E.

The cases where not both s and t are N-terms, are already dealt with in the proof of lemma 15. So let $s = C^\alpha\, u$, $t = D^\beta\, v$ and let $\sigma \in Sol(E)$, which in particular implies $s\sigma = t\sigma$. Moreover, let $n := [C,D]$ and let $m := [D,C]$.

Again, we can assume without loss of generality that $\alpha \in \mathscr{L}$, $\alpha = \alpha'$ and $\alpha'' = 0$ hold.

If $\mathrm{char}(C^n) \cap \mathrm{char}(D^m) = \emptyset$, the Decomposition and the Eliminate rule apply, which is already dealt with in the proof of lemma 15. So let us assume that $\mathrm{char}(C^n) \cap \mathrm{char}(D^m) \neq \emptyset$.

If $C^n \not\approx D^m$, then only rule Eliminate1 applies. We show that $0 \le \alpha\sigma < 2n$ or $0 \le \beta\sigma < 2m$. Suppose to the contrary that $\alpha\sigma = 2n + a$, $a > 0$ and $\beta\sigma = 2m + b$, $b > 0$ holds. From $s\sigma = t\sigma$ follows

$$(C\sigma)^{2n}\, (C\sigma)^a\, u\sigma = (C\sigma)^{2n+a}\, u\sigma = (C\sigma)^{\alpha\sigma}\, u\sigma =$$
$$(D\sigma)^{\beta\sigma}\, v\sigma = (D\sigma)^{2m+b}\, v\sigma = (D\sigma)^{2m}\, (D\sigma)^b\, v\sigma$$

Since $|C^{2n}| = |D^{2m}|$, this implies

$$u' := (C\sigma)^a\, u\sigma = (D\sigma)^b\, v\sigma$$

$$((C\sigma)^n)^2\ u' = (C\sigma)^{2n}\ u' = (D\sigma)^{2m}\ u' = ((D\sigma)^m)^2\ u',$$

thus contradicting lemma 22. This shows that $0 \le \alpha\sigma < 2n$ or $0 \le \beta\sigma < 2m$ holds. Without loss of generality, we assume $i := \alpha\sigma < 2n$. Then $\sigma \in \mathrm{Sol}(\alpha = i \wedge u = D^\beta v)$.

Now suppose $C^n \sim D^m$. Then the rules Eliminate, Eliminate1, and Induce apply. We distinguish three cases:

Case 1: $\alpha\sigma = i < n$. In this case $\sigma \in Sol(\alpha = i \wedge C^i\ u = D^\beta\ v)$.

Case 2: $\beta\sigma = j < m$. In this case $\sigma \in Sol(\beta = j \wedge C^\alpha\ u = D^j\ v)$.

Case 3: $\alpha\sigma \ge n$ and $\beta\sigma \ge m$. We assume without loss of generality that $n\beta\sigma - m\alpha\sigma \ge 0$. From $(C\sigma)^{\alpha\sigma}\ u\sigma = (D\sigma)^{\beta\sigma}\ v\sigma$ now follows

$$(C\sigma)^n = (D\sigma)^m \tag{1}$$

Let B be the prefix of D of length $\gcd(|C|,|D|)$. From (1) follows by lemma 21 that

$$C\sigma = (B\sigma)^m \text{ and } D\sigma = (B\sigma)^n$$

Hence we obtain

$$((B\sigma)^m)^{\alpha\sigma}\ u\sigma = (C\sigma)^{\alpha\sigma}\ u\sigma = (D\sigma)^{\beta\sigma}\ v\sigma = ((B\sigma)^n)^{\beta\sigma}\ v\sigma$$

and finally

$$u\sigma = (B\sigma)^{n\beta\sigma - m\alpha\sigma}\ v\sigma$$

which implies $\sigma \in \mathrm{Sol}(u = B^{n\beta - m\alpha}\ v)$. □

4 Extensions and Limitations of the Approach

It is clear that the notion of terms with exponents can subsume only a small part of the nontermination and divergence phenomena that occur in applications like logic programming or equational logic. In logic programming, for instance, the limits are set by undecidability results for fairly simple classes of Horn-clauses. Schmidt-Schauß [12], for instance, showed that satisfiability is an undecidable property for the class of Horn clauses consisting of two clauses with two atoms and an arbitrary number of ground unit clauses. Recently, Hanschke and Würtz [6] showed undecidability for the class of Horn clauses consisting of one clause with two atoms and an arbitrary number of unit clauses. Consequently, there can be no algorithm that transforms any given recursive clause with two atoms into a nonrecursive clause using terms with exponents, since this would yield a decision procedure for unsatisfiability.

Yet, there are possibilities to extend the syntax of terms with exponents in order to cope with particular divergence phenomena. As an example, consider the equational specification consisting of the axioms of associativity and idempotence

$$(xy)z = x(yz)$$
$$xx = x$$

Completing this specification yields two infinite sets of equations

$$x_1(x_2(x_1x_2)) = x_1x_2,\ x_1(x_2(x_3(x_1(x_2x_3)))) = x_1(x_2x_3),\ \ldots \tag{1}$$

and

$$x_1(x_1x_2) = x_1x_2,\ x_1(x_2(x_1(x_2x_3))) = x_1(x_2x_3),\ \ldots \tag{2}$$

Our notion of terms with exponents is too weak to cope with such sequences of terms with an increasing number of variables. However, the syntax can be extended by introducing an additional sort of variables, called *flexible variables*. Roughly spoken, the flexible variables are renamed with each unfolding, an idea that originates with [3]. To be more precise, we need to extend the signature by a set

$$\mathcal{FV} = \bigcup_{n \geq 0} V_n$$

of *flexible variables*, such that $V_i \cap V_j = \emptyset$ for $i \neq j$, and $|V_i| = |V_j|$ for all i, j. Moreover, let $\pi : FV \to FV$ be a bijective mapping with $\pi(V_i) = V_{i+1}$ for all $i \geq 0$. Now the relation = on $\mathcal{T}$ is defined by

$$C^0 = *$$
$$C^n = C((C^{n'})\pi) \text{ where } n \in \mathbb{N},\ n' = n-1$$

Additionally, we have to change the definition of the mapping ρ, which now reads as follows: $*\rho = *$, $(c_1\ldots c_n)\rho = c_2\ldots c_n(c_1\pi)$.

As an example, the sequence (2) can now be represented in the form

$$(x_0*)^N((x_0*)^N y) = (x_0*)^N y$$

where x_0 is a flexible variable.

The inference system for unification remains unchanged under this extension.

Other directions for further research include equational computation using terms with exponents. For instance, it is not clear how to extend the concept of simplification orderings to this new kind of terms. As an example, consider the ground terms $s = f^N(a)$ and $t = a$. The term t is both a subterm of s and an instance of s, and so any naive application of the concept of simplification ordering fails.

Moreover, it is not clear which notion of subterm replacement should be used for equational computations. For instance, consider the rewrite system $\mathcal{R}$ consisting of the rule $f(x) \to g(x)$, and let $t = f^N(a)$. Rewriting the term t at the root position yields an infinite sequence of rewritings

$$f^N a \Rightarrow gf^{N-1}a \Rightarrow g^2 f^{N-2}a \Rightarrow \ldots$$

Instead, one would like to be able to rewrite t into the term $g^N(a)$.

Acknowledgment

I would like to thank Andrea Sattler-Klein and Thomas Deiß for contributing some valuable suggestions to this paper.

References

1. Avenhaus, J. Proving Equational and Inductive Theorems by Completion and Embedding Techniques. In: Book, R.V. (Ed.), *Proc. of 4th International Conference on Rewriting Techniques and Applications,* Springer LNCS 488, Como, 1991, 361-373.

2. Bachmair, L., Dershowitz, N., and Plaisted, D. Completion without Failure. In: *Coll. On the Resolution of Equations in Algebraic Structures.* Academic Press, 1987.

3. Chen, H., Hsiang, J., and Kong, H.C. On Finite Representations of Infinite Sequences of Terms. In: Kaplan, S. and Okada, M. (Eds.), *Proc. of 2nd Int. Workshop on Conditional and Typed Rewriting Systems,* Springer LNCS 516, Montreal, 1990, 206-232.

4. Chen, H. and Hsiang, J. Logic Programming with Recurrence Domains. In: Albert, J.L., Monien, B., and Artalejo, M.R. (Eds.), *Proc. of 18th Int. Coll. on Automata, Languages, and Programming,* Springer LNCS 510, 1991, 20-34.

5. Comon, H., "Unification of Terms with Exponents," 1992, To appear in: *Mathematical Systems Theory.*

6. Hanschke, P. and Wuertz, J. Satisfiability of the smallest binary program. Information Processing Letters 45/5(1993), 237-241.

7. Hermann, M., "Vademecum of divergent term rewriting systems," CRIN, Research Report, no. 88-R-082, 1988.

8. Knuth, D.E. and Bendix, P.B. Simple Word Problems in Universal Algebra. In: Leech, J. (Ed.), *Computational Problems in Universal Algebra.* Pergamon Press, 1970.

9. Lothaire, M. *Combinatorics and Words,* Addison-Wesley (1983).

10. Ohlbach, H.J. Compilation of Recursive Two-Literal Clauses into Unification Algorithms. In: Jorrand, P. and Sgurev, V. (Eds.), *Artficial Intelligence IV, Methodology, Systems, Applications.* North-Holland, 1990.

11. Salzer, G. The Unification of Infinite Sets of Terms and its Applications. In: A. Voronkov (Ed.): *Proc. of 3rd Conference on Logic Programming and Automated Reasoning.* Springer LNCS 624, 1991, 409-420.

12. Schmidt-Schauß, M. Implication of Clauses is Undecidable. *Theoretical Computer Science* 59(1988), 287-296.

13. Schmidt-Schauß, M. *Computational Aspects of an Order-Sorted Logic With Term Declarations,* Springer LNAI 395, New York (1989).