Theorem Proving with Sequence Variables and Flexible Arity Symbols*

Temur Kutsia

Research Institute for Symbolic Computation
Johannes Kepler University Linz
A-4040, Linz, Austria
kutsia@risc.uni-linz.ac.at

Software Competence Center Hagenberg
Hauptstrasse, 99
A-4232, Hagenberg, Autsria
teimuraz.kutsia@scch.at

Abstract. An ordering for terms with sequence variables and flexible arity symbols is presented. The ordering coincides with the lexicographic extension of multiset path ordering on terms without sequence variables. It is shown that the classical strict superposition calculus with ordering and equality constraints can be used as a refutationally complete proving method for well-constrained sets of clauses with sequence variables and flexible arity symbols.

1 Introduction

Sequence variables are variables which can be instantiated by an arbitrary finite, possibly empty, sequence of terms. Flexible arity symbols are not assigned unique arity. Sequence variables and flexible arity symbols add flexibility and expressiveness into a language[1], which makes them a useful tool in many applications: knowledge engineering and artificial intelligence (Knowledge Interchange Format KIF [GF92] and its version SKIF [HM01]), databases (Sequence Datalog [MB95], Sequence Logic [GW92]), programming (programming language of Mathematica [Wol99]), term rewriting (rewriting with sequences [Ham97], [WB01]). However, theorem proving with sequence variables and flexible arity symbols is not well-studied. The simplifier prover [BM97] of the Theorema system [BDJ+00] and the Epilog [Gen95] package are probably the only provers with (restricted) features for sequence variables.

Buchberger ([Buc96], [Buc01]) proposed to study usage of sequence variables in proving, solving and rewriting context, which, among the other results, lead to development of unification procedure for equational theories with sequence variables and flexible arity symbols ([Kut02c], [Kut02a], [Kut02b]). It was shown that although (general) unification is decidable, its type is infinitary, even for the free theory with sequence variables and flexible arity symbols. It suggests to use

* Supported by the Austrian Science Foundation (FWF) under Project SFB F1302 and by Software Competence Center Hagenberg (Austria) under ForMI project.

[1] In fact, unrestricted quantification over sequence variables takes the language even beyond first-order expressiveness, but there is an useful sublanguage which is strictly first-order, see [HM01]. In this paper we stick to this sublanguage.

"proving with constraints" approach ([NR92], [NR95], [Rub95], [NR01]): instead of unifying the terms, keeping the unification problem in an equality constraint. One can detect unsatisfiability of a constraint using efficient incomplete methods, which would allow to remove clauses with unsatisfiable constraints. Only the constraint of the empty clause should be checked on solvability, to know whether inconsistency was derived or not.

In order to use this approach efficiently, another ingredient – a term ordering with sequence variables and flexible arity symbols – is needed. In this paper we present an ordering which is total on ground terms, is stable under substitutions and for terms without sequence variables coincides with the lexicographic extension of multiset path ordering [Der82]. It is, to our knowledge, the first such ordering on terms with sequence variables and flexible arity symbols. Moreover, it can be shown that the ordering is, in fact, a reduction ordering. However, for our purposes the stability property is sufficient.

Finally, it is shown that the classical strict superposition calculus with ordering and equality constraints (see e.g. [NR95] or [NR01]) is a refutationally complete proving method for theories with sequence variables and flexible arity symbols.

2 Preliminaries

2.1 Syntax

We consider an alphabet $\mathfrak{A}$ consisting of the following pairwise disjoint sets of symbols: the set of individual variables $\mathcal{V}_{\mathrm{Ind}}$, the set of sequence variables $\mathcal{V}_{\mathrm{Seq}}$, the set of fixed arity function constants $\mathcal{F}_{\mathrm{Fix}}$ and the set of flexible arity function constants $\mathcal{F}_{\mathrm{Flex}}$.

The set of terms (over $\mathfrak{A}$) is the smallest set of strings over $\mathfrak{A}$ that satisfies the following conditions:

- If $t \in \mathcal{V}_{\mathrm{Ind}} \cup \mathcal{V}_{\mathrm{Seq}}$ then t is a term.
- If $f \in \mathcal{F}_{\mathrm{Fix}}$, f is n-ary, $n \geq 0$ and $t_1, \dots, t_n$ are terms such that for all $1 \leq i \leq n$, $t_i \notin \mathcal{V}_{\mathrm{Seq}}$, then $f(t_1, \dots, t_n)$ is a term.
- If $f \in \mathcal{F}_{\mathrm{Flex}}$ and $t_1, \dots, t_n$ $(n \geq 0)$ are terms, then so is $f(t_1, \dots, t_n)$.

f is called the head of $f(t_1, \dots, t_n)$.

An equation (over $\mathfrak{A}$) is a multiset $\{s, t\}$, denoted $s \simeq t$, where s and t are terms (over $\mathfrak{A}$) such that $s \notin \mathcal{V}_{\mathrm{Seq}}$ and $t \notin \mathcal{V}_{\mathrm{Seq}}$.

A clause (over $\mathfrak{A}$) is a pair of finite multisets of equations (over $\mathfrak{A}$) Γ (the *antecedent*) and Δ (the *succedent*), denoted by $\Gamma \rightarrow \Delta$. The empty clause $\Box$ is a clause $\Gamma \rightarrow \Delta$ where both Γ and Δ are empty.

If not otherwise stated, the following symbols, with or without indices, are used as metavariables: x and y – over individual variables, $\overline{x}$, $\overline{y}$, $\overline{z}$ and $\overline{u}$ – over sequence variables, a and b – over constants, f and g – over (fixed or flexible arity) function symbols, s, t and r – over terms.

We generalize standard notions of unification theory ([BS01]) for a theory with sequence variables and flexible arity symbols.

Definition 1 (Substitution). *A substitution is a finite set* $\{x_1 \leftarrow s_1, \ldots, x_n \leftarrow s_n, \overline{x}_1 \leftarrow t_1^1, \ldots, t_{k_1}^1, \ldots, \overline{x}_m \leftarrow t_1^m, \ldots, t_{k_m}^m\}$ *where*

- $n \geq 0$, $m \geq 0$ *and for all* $1 \leq i \leq m$, $k_i \geq 0$,
- $x_1, \ldots, x_n$ *are distinct individual variables,*
- $\overline{x}_1, \ldots, \overline{x}_m$ *are distinct sequence variables,*
- *for all* $1 \leq i \leq n$, s_i *is a term,* $s_i \notin \mathcal{V}_{\mathrm{Seq}}$ *and* $s_i \neq x_i$,
- *for all* $1 \leq i \leq m$, $t_1^i, \ldots, t_{k_i}^i$ *is a sequence of terms and if* k_i=1 *then* $t_{k_i}^i \neq \overline{x}_i$.

Greek letters are used to denote substitutions. The empty substitution is denoted by ε.

Given a substitution θ, the notion of an instance of a term t with respect to θ, denoted $t\sigma$ is defined recursively as follows:

- $x\theta = \begin{cases} s \text{ if } x \leftarrow s \in \theta, \\ x \text{ otherwise} \end{cases}$
- $\overline{x}\theta = \begin{cases} s_1, \ldots, s_m \text{ if } \overline{x} \leftarrow s_1, \ldots, s_m \in \theta,\ m \geq 0, \\ \overline{x} \qquad\qquad \text{otherwise} \end{cases}$
- $f(s_1, \ldots, s_n)\theta = f(s_1\theta, \ldots, s_n\theta)$.

Instances of an equation and a clause are defined as usual. By $\mathcal{L}$ we denote the equational language with the alphabet $\mathfrak{A}$ and terms, equations and clauses defined as above. A substitution σ is called a grounding substitution for an expression Q of $\mathcal{L}$ iff $Q\sigma$ contains no variables.

2.2 Semantics

We define semantics of $\mathcal{L}$. To interpret sequence variables and flexible arity symbols we choose an approach similar to the semantics of SKIF language [HM01]. First we adopt some notions from [HM01] with a slight modification:

For any set A, let A^n be the set of sequences of length n of members of A, i.e. functions from the set $\{0, 1, \ldots, n-1\}$ of ordinals less then n into A (it follows that $A^0 = \{\emptyset\}$ since the set of ordinals less than 0 is empty). We call the members of A^n n-tuples over A. We will write the n-tuple $\{\langle 0, a_0\rangle, \langle 1, a_1\rangle, \ldots \langle n-1, a_{n-1}\rangle\}$ as $\langle a_0, a_1, \ldots, a_{n-1}\rangle$. In particular, we will sometimes refer to $\emptyset$ as $'\langle\rangle'$ when thinking of it as a 0-tuple. Let A^* be the set of all n-tuples over A, for all n, i.e. $A^* = \cup_{n<\omega} A^n$. We will call the members of A^* *tuples over* A. We want to consider n-tuples in which there is no distinction between 1-tuple and its (sole) member, so we define an equivalence relation $\approx$ on $A \cup A^*$ by $\langle x\rangle \approx x$. Let $[a]_\approx$ denote an equivalence class of a on $A \cup A^*$ with respect to $\approx$. Then we define $A^{**} = \{[t]_\approx \mid t \in A \cup A^*\}$. We call the members of A^{**} rows over A or A-rows. Notice that any element of A constitutes a singleton row.

The operation of concatenation cc on A^{**} is defined as follows:

$cc() = [\langle\rangle]_\approx$.
$cc([a]_\approx) = [\langle a\rangle]_\approx,\ a \in A$.
$cc([\langle t_1, \ldots, t_n\rangle]_\approx) = [\langle t_1, \ldots, t_n\rangle]_\approx,\ \langle t_1, \ldots, t_n\rangle \in A^*$.
$cc([a]_\approx, [b]_\approx) = [\langle a, b\rangle]_\approx,\ a \in A,\ b \in A$.
$cc([a]_\approx, [\langle t_1, \ldots, t_n\rangle]_\approx) = [\langle a, t_1, \ldots, t_n\rangle]_\approx,\ a \in A,\ \langle t_1, \ldots, t_n\rangle \in A^*$.
$cc([\langle t_1, \ldots, t_n\rangle]_\approx, [a]_\approx) = [\langle t_1, \ldots, t_n, a\rangle]_\approx,\ \langle t_1, \ldots, t_n\rangle \in A^*, a \in A$.
$cc([\langle t_1, \ldots, t_n\rangle]_\approx, [\langle s_1, \ldots, s_m\rangle]_\approx) = [\langle t_1, \ldots, t_n, s_1, \ldots, s_m\rangle]_\approx,$
$\qquad \langle t_1, \ldots, t_n\rangle \in A^*, \langle s_1, \ldots, s_m\rangle \in A^*$.

It can be easily shown that cc is associative. Therefore we will use cc in the flattened form.

We say that a set of rows S satisfies functional condition iff for any two rows $[\langle e_1, \ldots, e_n, s\rangle]_\approx \in S$ and $[\langle e_1, \ldots, e_n, t\rangle]_\approx \in S$, $n \geq 0$, we have $s = t$. It implies that if a non-empty set of rows satisfies functional condition then it contains at most one singleton row.

An interpretation I for the language $\mathcal{L}$ consists of

- a non-empty set D called a domain of I,
- an assignment for each constant c in $\mathcal{L}$ of an element $c_I \in D^{**}$ such that $c_I = [\langle d\rangle]_\approx$, where $d \in D$.
- an assignment for each n-ary function symbol f in $\mathcal{L}$ a set $f_I \subseteq D^{**}$ of the form $\{[\langle d_1, \ldots, d_n, d_{n+1}\rangle]_\approx \mid d_1, \ldots, d_n, d_{n+1} \in D\}$, satisfying functional condition,
- an assignment for each flexible arity function symbol f in $\mathcal{L}$ a set $f_I \subseteq D^{**}$ of the form $\{[\langle d_1, \ldots, d_k\rangle]_\approx \mid k \geq 0,\ d_1, \ldots, d_k \in D\}$, satisfying functional condition.

A state σ^I (over I) is defined as follows:

- $\sigma^I(x) = [\langle d\rangle]_\approx \in D^{**}$, for an individual variable x,
- $\sigma^I(\overline{x}) = [\langle d_1, \ldots, d_n\rangle]_\approx \in D^{**}$, for a sequence variable $\overline{x}$,
- $\sigma^I(c) = c_I$, for a constant c,
- $\sigma^I(f(t_1, \ldots, t_n)) = [\langle s\rangle]_\approx \in D^{**}$ for a term $f(t_1, \ldots, t_n)$ with fixed or flexible arity head f, where s is the unique element of D such that $cc(\sigma^I(t_1), \ldots, \sigma^I(t_n), [\langle s\rangle]_\approx) \in f_I$.

Let $true$ and $false$ be truth values. Truth value in a state σ^I over I of an equation $t_1 \simeq t_2$ in $\mathcal{L}$, written $I_\sigma(t_1 \simeq t_2)$, is defined as follows: $I_\sigma(t_1 \simeq t_2) = true$ iff $\sigma^I(t_1) = \sigma^I(t_2)$, otherwise $I_\sigma(t_1 \simeq t_2) = false$.

Truth value in I of an equation $t_1 \simeq t_2$ in $\mathcal{L}$, written $I(t_1 \simeq t_2)$, is defined as follows: $I(t_1 \simeq t_2) = true$ iff $I_\sigma(t_1 \simeq t_2) = true$ for all σ^I over I, otherwise $I(t_1 \simeq t_2) = false$.

We say that I is a model of $t_1 \simeq t_2$, or I satisfies $t_1 \simeq t_2$, and write $I \models t_1 \simeq t_2$ iff $I(t_1 \simeq t_2) = true$. An interpretation I is a model for a set of equations E (written $I \models E$) iff I is a model for each equation in E. An equation $t_1 \simeq t_2$ is a semantic consequence of a set of equations E, written $E \models t_1 \simeq t_2$, iff every model of E is a model of $t_1 \simeq t_2$.

Truth value in I of a clause $\Gamma \to \Delta$ in $\mathcal{L}$, written $I(\Gamma \to \Delta)$, is defined as follows: $I(\Gamma \to \Delta) = true$ iff $I(\Gamma) = false$ or $I(\Delta) = true$, otherwise $I(\Gamma \to \Delta) = false$. We write $I \models \Gamma \to \Delta$ iff $I(\Gamma \to \Delta) = true$.

2.3 Rewriting and Orderings

We assume that the reader is familiar with the basics of rewriting [DJ90], [BN98]. In this section we give some notions adapted for terms with sequence variables and flexible arity symbols.

Let $\mathcal{F} \subseteq \mathcal{F}_{\mathrm{Fix}} \cup \mathcal{F}_{\mathrm{Flex}}$ and $\mathcal{V} \subseteq \mathcal{V}_{\mathrm{Ind}} \cup \mathcal{V}_{\mathrm{Seq}}$. The set of terms over $\mathcal{F}$ and $\mathcal{V}$ is denoted by $\mathcal{T}(\mathcal{F},\mathcal{V})$. The set of ground terms over $\mathcal{F}$ is denoted by $\mathcal{T}(\mathcal{F})$. We call $\mathcal{F}$ a *signature* and assume that it contains at least one constant.

A rewrite rule is an ordered pair (s,t), written $s \Rightarrow t$, where $s,t \in \mathcal{T}(\mathcal{F},\mathcal{V}) \setminus \mathcal{V}_{\mathrm{Seq}}$. A set of rewrite rules R is a rewrite system. The rewrite relation with R on $\mathcal{T}(\mathcal{F},\mathcal{V}) \setminus \mathcal{V}_{\mathrm{Seq}}$, denoted $\Rightarrow_R$, is the smallest monotonic relation such that $l\sigma \Rightarrow_R r\sigma$ for all $l \Rightarrow r \in R$ and all σ. A term s is called *reducible* by R if there is t such that $s \Rightarrow_R t$, otherwise s is *irreducible* by R.

A (strict partial) ordering $\succ$ is a transitive and irreflexive binary relation.

We say that an ordering $\succ$ on $\mathcal{T}(\mathcal{F},\mathcal{V}) \setminus \mathcal{V}_{\mathrm{Seq}}$

- is *stable* under (grounding) substitutions if $s \succ t$ implies $s\sigma \succ t\sigma$ for all $s,t \in \mathcal{T}(\mathcal{F},\mathcal{V}) \setminus \mathcal{V}_{\mathrm{Seq}}$ and σ (grounding for s and t).
- fulfills the *subterm property* if $u[s]_p \succ s$ for all $s,u \in \mathcal{T}(\mathcal{F},\mathcal{V}) \setminus \mathcal{V}_{\mathrm{Seq}}$ and p is not the top position.
- fulfills the *deletion property* if $f(\dots s \dots) \succ f(\dots\dots)$ for $f \in \mathcal{F} \cap \mathcal{F}_{\mathrm{Flex}}$ and all $s \in \mathcal{T}(\mathcal{F},\mathcal{V})$.
- is *total* up to a congruence $\cong$ on $\mathcal{T}(\mathcal{F})$ if for all $s,t \in \mathcal{T}(\mathcal{F})$, either $s \cong t$, or $s \succ t$ or $t \succ s$.

A *rewrite ordering* is a monotonic ordering stable under substitutions. A *reduction ordering* is a well-founded rewrite ordering. A *simplification ordering* is a rewrite ordering with the subterm property and the deletion property.

2.4 Constraints and Constrained Clauses

An (ordering and equality) constraint is a quantifier-free first-order formula over the binary predicate symbols $\succ$ and $\doteq$ relating terms in $\mathcal{T}(\mathcal{F},\mathcal{V})$. The constraints are interpreted in $\mathcal{T}(\mathcal{F})$, $\doteq$ is interpreted as a syntactic equality $\simeq$ on $\mathcal{T}(\mathcal{F})$ and $\succ$ is interpreted as a given reduction ordering on $\mathcal{T}(\mathcal{F})$ that is total up to $\simeq$. Logical connectives are interpreted in the usual way.

We denote by

- $ivars(Q)$ - the set of all individual variables occurring in Q;
- $svars(Q)$ - the set of all sequence variables occurring in Q;
- $vars(Q)$ - $ivars(Q) \cup svars(Q)$;

where Q can be a term, an equation, a clause or a constraint.

A solution of a constraint T is a ground substitution σ with domain $vars(T)$ and such that $T\sigma$ evaluates to *true* for $\simeq$ and a given reduction ordering. If a solution of T exists, then T is called satisfiable. If every ground substitution with domain $vars(T)$ is a solution of T then T is a tautology.

A constrained clause is a pair $(C \parallel T)$ where C is a clause and T is a constraint. A ground instance of $(C \parallel T)$ is a ground clause $C\sigma$ where σ is a solution of T. We say that an interpretation I satisfies $(C \parallel T)$ if $I \models C\sigma$ for every ground instance $C\sigma$ of $(C \parallel T)$. Therefore, clauses with unsatisfiable constraints are tautologies.

3 Ordering for Terms with Sequence Variables and Flexible Arity Symbols

The goal of this section is to define an ordering on $\mathcal{T}(\mathcal{F},\mathcal{V}) \setminus \mathcal{V}_{\text{Seq}}$. Let us assume that a well-founded ordering $\succ_{\mathcal{F}}$ is given on the set $\mathcal{F}$. Here $\succ_{\mathcal{F}}$ is called the *precedence*. Furthermore, let $\simeq_{mul}$ denote the equality on $\mathcal{T}(\mathcal{F},\mathcal{V}) \setminus \mathcal{V}_{\text{Seq}}$ defined as follows: $s \simeq_{mul} t$ iff

- $s, t \in \mathcal{V}_{\text{Ind}}$ and $s \simeq t$, or
- $s = f(s_1, \ldots, s_n)$, $t = f(t_1, \ldots, t_n)$, $f \in \mathcal{F}_{\text{Fix}} \cup \mathcal{F}_{\text{Flex}}$ and for all i, $1 \le i \le n$, $s_{\pi(i)} \simeq_{mul} t_i$ or $s_{\pi(i)}$ and t_i are the same sequence variables, where π is a permutation of $1, \ldots, n$.

For example, $f(\overline{x}, g(a, \overline{y}, y), b) \simeq_{mul} f(b, \overline{x}, g(\overline{y}, a, y))$.

In the subsections below, first we define an ordering $\succ_{mposvm}$ on $\mathcal{T}(\mathcal{F},\mathcal{V}) \setminus \mathcal{V}_{\text{Seq}}$, which is stable, and total on ground terms up to $\simeq_{mul}$, when the precedence is total. For the terms without sequence variables (i.e. on $\mathcal{T}(\mathcal{F},\mathcal{V} \setminus \mathcal{V}_{\text{Seq}})$) $\succ_{mposvm}$ coincides with the multiset path ordering $\succ_{mpo}$ (also called the recursive path ordering without status [KL80], [Der82]). After that, we extend $\succ_{mposvm}$ to a stable ordering $\succ_{mposv}$ which is total on ground terms up to $\simeq$, when the precedence is total.

3.1 The Ordering $\succ_{mposvm}$

Let $\succ$ be an ordering on $\mathcal{T}(\mathcal{F},\mathcal{V}) \setminus \mathcal{V}_{\text{Seq}}$ and $\cong$ be a congruence on $\mathcal{T}(\mathcal{F},\mathcal{V}) \setminus \mathcal{V}_{\text{Seq}}$ compatible with $\succ$. We will need two extensions of $\succ$ with respect to $\cong$: the *lexicographic extension* and the *multiset extension*.

The *lexicographic extension* of $\succ$ with respect to $\cong$ is the relation $\succ^{lex,\cong}$ on n-tuples over $\mathcal{T}(\mathcal{F},\mathcal{V})$ defined by $\langle s_1, \ldots, s_n\rangle \succ^{lex,\cong} \langle t_1, \ldots, t_n\rangle$, if there exists k, $1 \le k \le n$, such that

1. for all i, $1 \le i \le k$, $s_i \cong t_i$ or s_i and t_i are the same sequence variables and
2. $s_k \succ t_k$, or $t_k \in svars(s_k)$ and s_k and t_k are not the same sequence variables.

The *multiset extension* of $\succ$ with respect to $\cong$ is the relation $\succ^{mul,\cong}$ on the set $\mathcal{M}(\mathcal{T}(\mathcal{F},\mathcal{V}))$ of all multisets over $\mathcal{T}(\mathcal{F},\mathcal{V})$ such that for all $M, N \in \mathcal{M}(\mathcal{T}(\mathcal{F},\mathcal{V}))$ $M \succ^{mul,\cong} N$ iff

1. M and N are not equal up to $\cong$ and
2. for all $n \in N \setminus M$ there exists $m \in M \setminus N$ such that $m \succ n$, or $n \in svars(m)$ and n and m are not the same sequence variables (the operation $\setminus$ is performed modulo $\cong$).

First, we define a binary relation $\succ_{mposvm}$ ($mposvm$ stands for <u>m</u>ultiset <u>p</u>ath <u>o</u>rdering with <u>s</u>equence <u>v</u>ariables ground total up to $\simeq_{\underline{m}ul}$).

Definition 2. *Let $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V}) \setminus \mathcal{V}_{\text{Seq}}$ and $\succ_{\mathcal{F}}$ be a precedence. Then $s \succ_{mposvm} t$ iff*

1. *$t \in ivars(s)$ and $s \neq t$, or*
2. *$s = f(s_1, \ldots, s_m)$, $m \geq 0$, $t = g(t_1, \ldots, t_n)$, $n \geq 0$ and*
   (a) *$s_i \succ_{mposvm} t$ or $s_i \simeq_{mul} t$ for some i with $1 \leq i \leq m$ or*
   (b) *$f \succ_{\mathcal{F}} g$, and for all j with $1 \leq j \leq n$, either $s \succ_{mposvm} t_j$ or $t_j \in svars(s)$ or*
   (c) *$f = g$, and*
   $\{s_1, \ldots, s_m\} \succ_{mposvm}^{mul,\simeq_{mul}} \{t_1, \ldots, t_n\}$ *and*
   $\{s_1, \ldots, s_m\} \setminus \mathcal{V}_{\text{Seq}} \succ_{mposvm}^{mul,\simeq_{mul}} \{t_1, \ldots, t_n\} \setminus \mathcal{V}_{\text{Seq}}$.

For better readibility we omit $\simeq_{mul}$ from the superscript of $\succ_{mposvm}^{mul,\simeq_{mul}}$ and write $\succ_{mposvm}^{mul}$.

It is easy to see that on $\mathcal{T}(\mathcal{F}, \mathcal{V} \setminus \mathcal{V}_{\text{Seq}})$ the relation $\succ_{mposvm}$ coincides with $\succ_{mpo}$: since on $\mathcal{T}(\mathcal{F}, \mathcal{V} \setminus \mathcal{V}_{\text{Seq}})$ we have $\{s_1, \ldots, s_m\} \setminus \mathcal{V}_{\text{Seq}} = \{s_1, \ldots, s_m\}$ and $\{t_1, \ldots, t_n\} \setminus \mathcal{V}_{\text{Seq}} = \{t_1, \ldots, t_n\}$, the case 2c) of the Definition 2 can be formulated as "$f = g$, and $\{s_1, \ldots, s_m\} \succ_{mposvm}^{mul,\simeq_{mul}} \{t_1, \ldots, t_n\}$", which gives exactly the definition of $\succ_{mpo}$ (see, e.g. [Der82]).

In the next sections we will use the following property of $\succ_{mposvm}$:

Theorem 1. *$\succ_{mposvm}$ is stable under grounding substitutions.*

Proof. We have to show that $s \succ_{mposvm} t$ implies $s\sigma \succ_{mposvm} t\sigma$ for all $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V}) \setminus \mathcal{V}_{\text{Seq}}$ and a substitution σ grounding for s and t. We use well-founded induction on $|s| + |t|$.

1. $s \succ_{mposvm} t$ by the case 1) of the Definition 2. Then $t \in ivars(s)$ and $t \neq s$. Since σ is grounding for t, $t\sigma$ is a strict ground subterm of $s\sigma$. By the subterm property of $\succ_{mpo}$ we have $s\sigma \succ_{mpo} t\sigma$. Since on ground terms the relations $\succ_{mposvm}$ and $\succ_{mpo}$ coincide, we get $s\sigma \succ_{mposvm} t\sigma$.
2. $s \succ_{mposvm} t$ by the case 2a) of the Definition 2. Then $s = f(s_1, \ldots, s_m)$, $m \geq 0$, and for some i, $1 \leq i \leq m$, $s_i \succ_{mposvm} t$ or $s_i \simeq_{mul} t$. If $s_i \succ_{mposvm} t$, then by the induction hypothesis $s_i\sigma \succ_{mposvm} t\sigma$. Since σ is grounding for s and t, $s\sigma$ and $t\sigma$ are ground terms. On the ground terms $\succ_{mposvm}$ and $\succ_{mpo}$ coincide, therefore $s_i\sigma \succ_{mpo} t\sigma$. By the subterm property of $\succ_{mpo}$, $s\sigma \succ_{mpo} s_i\sigma$. By transitivity of $\succ_{mpo}$, $s\sigma \succ_{mpo} t\sigma$. If $s_i \simeq_{mul} t$, then $s_i\sigma \simeq_{mul} t\sigma$. Therefore, by the subterm property of $\succ_{mpo}$ and compatibility of $\succ_{mpo}$ with $\simeq_{mul}$, we get $s\sigma \succ_{mpo} t\sigma$. Since on the ground terms $\succ_{mpo}$ and $\succ_{mposvm}$ coincide, we get $s\sigma \succ_{mposvm} t\sigma$.

3. $s \succ_{mposvm} t$ by the case 2b) of the Definition 2. Then $s = f(s_1, \ldots, s_m)$, $m \geq 0$, $t = g(t_1, \ldots, t_n)$, $n \geq 0$, $f \succ_{\mathcal{F}} g$, and for all j with $1 \leq j \leq n$, either $s \succ_{mposvm} t_j$ or $t_j \in svars(s)$. Then for an arbitrary but fixed j, if $s \succ_{mposvm} t_j$, then by the induction hypothesis $s\sigma \succ_{mposvm} t_j\sigma$ and, thus, $s\sigma \succ_{mpo} t_j\sigma$. Therefore, $s\sigma \succ_{mpo} t\sigma$ and since on the ground terms $\succ_{mpo}$ and $\succ_{mposvm}$ are the same relations, $s\sigma \succ_{mposvm} t\sigma$. If $t_j \in svars(s)$ then let the sequence $r_1, \ldots, r_k$, $k \geq 0$, be $t_j\sigma$. Since σ is grounding for t, each r_l is a ground term and by the subterm property of $\succ_{mpo}$ we have $s\sigma \succ_{mpo} r_l$, $1 \leq l \leq k$. Thus, if we denote $t\sigma$ by $g(u_1, \ldots, u_k)$, then we have for all i, $1 \leq i \leq k$, $s\sigma \succ_{mpo} u_i$. Therefore, $s\sigma \succ_{mpo} t\sigma$ and, thus, $s\sigma \succ_{mposvm} t\sigma$.
4. $s \succ_{mposvm} t$ by the case 2c) of the Definition 2. Then $s = f(s_1, \ldots, s_m)$, $m \geq 0$, $t = f(t_1, \ldots, t_n)$, $n \geq 0$, $\{s_1, \ldots, s_m\} \succ^{mul}_{mposvm} \{t_1, \ldots, t_n\}$ and $\{s_1, \ldots, s_m\} \setminus \mathcal{V}_{\mathrm{Seq}} \succ^{mul}_{mposvm} \{t_1, \ldots, t_n\} \setminus \mathcal{V}_{\mathrm{Seq}}$. Let the set $\{s_1, \ldots, s_p\}$ be $\{s_1, \ldots, s_m\} \setminus \{t_1, \ldots, t_n\}$ and $\{t_1, \ldots, t_q\}$ be $\{t_1, \ldots, t_n\} \setminus \{s_1, \ldots, s_m\}$. Then for all j, $1 \leq j \leq q$, there exists i, $1 \leq i \leq p$, such that $s_i \succ_{mposvm} t_j$ or $t_j \in svars(s_i)$. From $s_i \succ_{mposvm} t_j$ by the induction hypothesis we get $s_i\sigma \succ_{mposvm} t_j\sigma$ and, thus, $s_i\sigma \succ_{mpo} t_j\sigma$. From $t_j \in svars(s_i)$ we have that $t_j\sigma$ is either the empty sequence, or a sequence $r_1, \ldots, r_k$, $k \geq 1$, of proper ground subterms of the ground term $s_i\sigma$. In the last case, by the subterm property of $\succ_{mpo}$, we have $s_i\sigma \succ_{mpo} r_l$ for all l with $1 \leq l \leq k$. Let $\{s_1\sigma, \ldots, s_{p'}\sigma\}$ be $\{s_1\sigma, \ldots, s_m\sigma\} \setminus \{t_1\sigma, \ldots, t_n\sigma\}$ and $\{t_1\sigma, \ldots, t_{q'}\sigma\}$ be $\{t_1\sigma, \ldots, t_n\sigma\} \setminus \{s_1\sigma, \ldots, s_m\sigma\}$. Since $\{s_1, \ldots, s_m\} \setminus \mathcal{V}_{\mathrm{Seq}} \succ^{mul}_{mposvm} \{t_1, \ldots, t_n\} \setminus \mathcal{V}_{\mathrm{Seq}}$, the set $\{s_1\sigma, \ldots, s_{p'}\sigma\}$ can not be empty, i.e. $\{s_1\sigma, \ldots, s_m\sigma\}$ can not be equal to $\{t_1\sigma, \ldots, t_n\sigma\}$ up to $\simeq_{mul}$. Assume without loss of generality that $s_1\sigma$ is a maximal element of $\{s_1\sigma, \ldots, s_p\sigma\}$ with respect to $\succ_{mpo}$. Then for all j, $1 \leq j \leq q'$, if $t_j\sigma$ is a single term, we have $s_1\sigma \succ_{mpo} t_j\sigma$, and if $t_j\sigma$ is a sequence of terms $r_1, \ldots, r_k$, $k > 1$, then we have $s_1\sigma \succ_{mpo} r_l$ for all l with $1 < l \leq k$. It implies $s\sigma \succ_{mpo} t\sigma$ and, thus, $s\sigma \succ_{mposvm} t\sigma$.

□

Note that the stability property would not hold if we would have extended $\succ_{mpo}$ on $\mathcal{T}(\mathcal{F}, \mathcal{V}) \setminus \mathcal{V}_{\mathrm{Seq}}$ in a straightforward way, i.e. to have the case 2c) of Definition 2 formulated as follows: $f = g$ and $\{s_1, \ldots, s_m\} \succ^{mul}_{mposvm} \{t_1, \ldots, t_n\}$. A counterexample is the following: by this definition we have $f(\overline{x}, a) \succ^{mul}_{mposvm} f(a)$, but for $\sigma = \{\overline{x} \leftarrow\}$, $f(\overline{x}, a)\sigma \not\succ^{mul}_{mposvm} f(a)\sigma$.

It can also be proved that $\succ_{mposvm}$ is a transitive, irreflexive, monotonic, and well-founded relation, which does not fulfil the deletion property (counterexample: $f(\overline{x}, a) \not\succ_{mposvm} f(a)$). It implies that $\succ_{mposvm}$ is a reduction (but not simplification) ordering on $\mathcal{T}(\mathcal{F}, \mathcal{V}) \setminus \mathcal{V}_{\mathrm{Seq}}$, which is total on $\mathcal{T}(\mathcal{F})$ up to $\simeq_{mul}$ if the precedence $\succ_{\mathcal{F}}$ is total.

3.2 The Ordering $\succ_{mposv}$

Now we extend the ordering $\succ_{mposvm}$ to the ordering $\succ_{mposv}$ which is total on ground terms up to $\simeq$, if the precedence is total.

Definition 3. *Let $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V}) \setminus \mathcal{V}_{\text{Seq}}$. Then $s \succ_{mposv} t$ iff*

- $s \succ_{mposvm} t$ *or*
- $s \simeq_{mul} t$, $s = f(s_1, \ldots, s_n)$, $t = f(t_1, \ldots, t_n)$ *and* $\langle s_1, \ldots, s_n \rangle \succ^{lex}_{mposv} \langle t_1, \ldots, t_n \rangle$.

On $\mathcal{T}(\mathcal{F}, \mathcal{V} \setminus \mathcal{V}_{\text{Seq}})$ the relation $\succ_{mposvm}$ coincides with the lexicographic extension of $\succ_{mpo}$.

The following theorem establishes the most important property of $\succ_{mposv}$:

Theorem 2. *The relation $\succ_{mposv}$ is stable under grounding substitutions.*

Proof. We have to show that $s \succ_{mposv} t$ implies $s\sigma \succ_{mposv} t\sigma$ for all $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V}) \setminus \mathcal{V}_{\text{Seq}}$ and a substitution σ grounding for s and t. We use well-founded induction on $|s| + |t|$. Suppose $s \succ_{mposv} t$. If $s \succ_{mposvm} t$ then by grounding stability of $\succ_{mposvm}$ we have $s\sigma \succ_{mposv} t\sigma$ for any grounding σ. If $s \simeq_{mul} t$, then $s = f(s_1, \ldots, s_n)$, $t = f(t_1, \ldots, t_n)$ and $\langle s_1, \ldots, s_n \rangle \succ^{lex}_{mposv} \langle t_1, \ldots, t_n \rangle$. From $s \simeq_{mul} t$, by definition of $\simeq_{mul}$ we get $s\sigma \simeq_{mul} t\sigma$ for any grounding σ. Therefore, by the definition of lexicographic extension and the induction hypothesis, $\langle s_1\sigma, \ldots, s_n\sigma \rangle \succ^{lex}_{mposv} \langle t_1\sigma, \ldots, t_n\sigma \rangle$ for any grounding σ, which implies $s\sigma \succ_{mposv} t\sigma$. □

It can be proved that $\succ_{mposv}$ is a reduction ordering on $\mathcal{T}(\mathcal{F}, \mathcal{V}) \setminus \mathcal{V}_{\text{Seq}}$, which is total on $\mathcal{T}(\mathcal{F})$ up to $\simeq$ if the precedence $\succ_{\mathcal{F}}$ is total. However, as we will see in the next section, only stability under grounding substitutions is sufficient for our purposes.

4 Inference System

The following inference system is the classical strict superposition calculus (see [NR95], [NR01]), with the only difference that in clauses and constraints sequence variables and flexible arity symbols are allowed to occur. The ordering $\succ$ is a reduction ordering for terms with sequence variables and flexible arity symbols, which is total on ground terms when the precedence is total. In particular, $\succ_{mposv}$ can be taken for $\succ$. The constraint $gr(s \simeq t, \Delta)$ expresses that for all equations $l \simeq r$ in Δ, the equation $s \simeq t$, i.e. the multiset $\{s, t\}$ is strictly greater than the multiset $\{l, r\}$ with respect to the multiset extension of $\succ$: $\{s, t\} \succ^{mul,\simeq} \{l, r\}$. The constraint $greq(s \simeq t, \Delta)$ expresses $\{s, t\} \succeq^{mul,\simeq} \{l, r\}$ for all $l \simeq r$ in Δ.

Definition 4. *The inference rules of the inference system $\mathcal{I}$ of strict superposition calculus are the following:*

1. *Strict superposition right:*

$$\frac{(\Gamma' \to \Delta', s' \simeq t' \parallel T') \qquad (\Gamma \to \Delta, s \simeq t \parallel T)}{\begin{array}{l}(\Gamma', \Gamma \to \Delta', \Delta, s[t']_p \simeq t \;\parallel T' \wedge s' \succ t' \wedge s' \succ \Gamma' \wedge gr(s' \simeq t', \Delta') \wedge \\ \qquad\qquad\qquad\qquad\qquad T \wedge s \succ t \wedge s \succ \Gamma \wedge gr(s \simeq t, \Delta) \wedge s|_p \doteq s')\end{array}}$$

where $s|_p \notin vars(s)$.

2. Strict superposition left:

$$\frac{(\Gamma' \to \Delta', s' \simeq t' \parallel T') \qquad\qquad (\Gamma, s \simeq t \to \Delta \parallel T)}{\begin{array}{ll}(\Gamma', \Gamma, s[t']_p \simeq t \to \Delta', \Delta & \parallel T' \wedge s' \succ t' \wedge s' \succ \Gamma' \wedge gr(s' \simeq t', \Delta') \wedge \\ & \;\; T \wedge s \succ t \wedge s \succ \Gamma \wedge greq(s \simeq t, \Gamma \cup \Delta) \wedge \\ & \;\; s|_p \doteq s')\end{array}}$$

where $s|_p \notin vars(s)$*.*

3. Equality resolution:

$$\frac{(\Gamma, s \simeq t \to \Delta \parallel T)}{(\Gamma \to \Delta \;\; \parallel T \wedge greq(s \simeq t, \Gamma \cup \Delta) \wedge s \doteq t)}$$

4. Factoring

$$\frac{(\Gamma \to \Delta, s \simeq t, s' \simeq t' \parallel T)}{\begin{array}{ll}(\Gamma, t \simeq t', \to \Delta, s \simeq t \;\parallel & T \wedge s \succ t \wedge s' \succ t' \wedge greq(s \simeq t, \Delta \cup \{s' \simeq t'\}) \wedge \\ & s \succ \Gamma \wedge s \doteq s')\end{array}}$$

The model generation proof method ([BG90], [BG94]) used in the completeness proof for the strict superposition calculus for general constrained clauses without sequence variables in [NR95] or [NR01] apply as well to the case with sequence variables. Completeness holds for so called *well-constrained* sets of clauses, which can be defined in the same way as in [NR01], only a slight refinement is needed for the notion of irreducible ground substitution: Let R be a ground rewrite system contained in the given ordering $\succ$. A ground substitution σ is *irreducible* by R, if for every $x \in dom(\sigma)$, $x\sigma$ is irreducible by R and for every $\overline{x} \in dom(\sigma)$ either $\overline{x}\sigma$ is the empty sequence or $\overline{x}\sigma = t_1, \dots, t_n$, such that for all i, $1 \leq i \leq n$, t_i is irreducible with respect to R. Furthermore, if S is a set of constrained clauses, then $irred_R(S)$ is its set of *irreducible instances*, that is, the set of ground instances $C\sigma$ of clauses $(C \parallel T)$ in S such that σ is a solution of T, for all $x \in vars(C)$ the term $x\sigma$ is irreducible by R and for all $\overline{x} \in vars(C)$ either $\overline{x}\sigma$ is the empty sequence or $\overline{x}\sigma = t_1, \dots, t_n$, such that for all i, $1 \leq i \leq n$, t_i is irreducible by R. Then well-constrained set is defined as follows:

Definition 5. *A set S of constrained clauses is well-constrained if either there are no clauses with equality predicates in S or else for all R contained in $\succ$ we have $irred_R(S) \cup R \models S$.*

The completeness theorem is formulated as follows:

Theorem 3. *The inference system $\mathcal{I}$ is refutationally complete for a well-constrained set of clauses.*

Proof. By the model generation method, as the completeness proof for the strict superposition calculus for general constrained clauses without sequence variables in [NR95] or [NR01]. □

Example 1. Let S be the following set of clauses: $\{(\rightarrow f(g(\overline{x}, a, \overline{y})) \simeq f(g(\overline{x}), g(\overline{y})) \parallel \mathit{true}),\ (f(g(b, a), g()) \simeq f(g(b), g(a)) \rightarrow \parallel \mathit{true})\}$. Obviously S is well-constrained. Let the ordering $\succ$ be $\succ_{mposv}$ with the precedence $f \succ_{\mathcal{F}} g \succ_{\mathcal{F}} a \succ_{\mathcal{F}} b$. Then we have the following refutation of S by the inference system $\mathcal{I}$ (the ordering constraints which evaluate to *true* are omitted):

1. $(\rightarrow f(g(\overline{x}, a, \overline{y})) \simeq f(g(\overline{x}), g(\overline{y})) \parallel \mathit{true})$.
2. $(f(g(b, a), g()) \simeq f(g(b), g(a)) \rightarrow \parallel \mathit{true})$.
3. $(\rightarrow f(g(\overline{x}), g(\overline{y})) \simeq f(g(\overline{z}), g(\overline{u})) \parallel f(g(\overline{x}, a, \overline{y})) \doteq f(g(\overline{z}, a, \overline{u})))$
   strict superposition right of the clause 1 and the (renamed copy of) clause 1.
4. $(f(g(\overline{z}), g(\overline{u})) \simeq f(g(b), g(a)) \rightarrow \parallel f(g(\overline{x}, a, \overline{y})) \doteq f(g(\overline{z}, a, \overline{u})) \wedge f(g(\overline{x}), g(\overline{y})) \succ f(g(\overline{z}), g(\overline{u})) \wedge f(g(\overline{x}), g(\overline{y})) \doteq f(g(b, a), g()))$
   strict superposition left of the clause 3 and the clause 2.
5. $(\square \parallel f(g(\overline{x}, a, \overline{y})) \doteq f(g(\overline{z}, a, \overline{u})) \wedge f(g(\overline{x}), g(\overline{y})) \succ f(g(\overline{z}), g(\overline{u})) \wedge f(g(\overline{x}), g(\overline{y})) \doteq f(g(b, a), g()) \wedge f(g(\overline{z}), g(\overline{u})) \doteq f(g(b), g(a)))$
   equality resolution in the clause 4.

The constraint in the clause 5 is satisfiable. The substitution $\{\overline{z} \leftarrow b,\ \overline{u} \leftarrow a,\ \overline{x} \leftarrow b, a,\ \overline{y} \leftarrow\}$ is a solution of it. Therefore, by Theorem 3, S is unsatisfiable.

Note that in the inference system the ordering constraints do not affect soundness and completeness. Their purpose is to prune the search space. Therefore, we do not really need to decide the satisfiability of ordering constraints, but would like to detect many cases of unsatisfiable constraints. Therefore, for instance, given an atomic constraint $s \succ_{mposv} t$, we would like to use the ordering $\succ_{mposv}$ for detecting that $s\sigma \succ_{mposv} t\sigma$ for no grounding σ (on ground terms $\succ_{mposv}$ coincides with the lexicographic extension of $\succ_{mpo}$). For this purpose the ground stability property is sufficient. This was the reason why in the previous section we concentrated on the stability under grounding substitutions.

The notions of saturation and redundancy defined in [NR01], [NR95] applies to the case with sequence variables as well, which leads to the following result:

Theorem 4. *Let S be a well-constrained set of clauses saturated with respect to $\mathcal{I}$. Then S is satisfiable iff $\square \notin S$.*

Finally, note that for a special subclass of pure equational theories with sequence variables and flexible arity symbols unfailing completion is a refutationally complete proving method. The special subclass consists of unit unconstrained equations, where sequence variables can occur only as the last arguments in the terms. It was shown in [Kut02b] that this restriction makes unification type finitary and matching type unitary. Completeness of the unfailing completion can be shown in the same way as the completeness of the classical unfailing completion [BDP89] for the theories without sequence variables.

5 Conclusion

We presented an ordering for terms with sequence variables and flexible arity symbols, which is stable under grounding substitutions and coincides with the lexicographic extension of multiset path ordering on terms without sequence variables. We showed that the classical strict superposition calculus with equality and ordering constraints can be used as a refutationally complete proving method for well-constrained sets of clauses with sequence variables and flexible arity symbols.

6 Acknowledgments

I am grateful to Bruno Buchberger for his supervision and support, to Robert Nieuwenhuis for very helpful remarks and to Mircea Marin for interesting discussions.

References

[BDJ+00] B. Buchberger, C. Dupré, T. Jebelean, F. Kriftner, K. Nakagawa, D. Vasaru, and W. Windsteiger. The Theorema project: A progress report. In M. Kerber and M. Kohlhase, editors, *Symbolic Computation and Automated Reasoning (Proceedings of CALCULEMUS 2000)*, pages 98–113, St.Andrews, UK, 6–7 August 2000.

[BDP89] L. Bachmair, N. Dershowitz, and D. Plaisted. Completion without failure. In H. Aït-Kaci and M. Nivat, editors, *Resolution of Equations in Algebraic Structures*, volume 2, pages 1–30. Elsevier Science, 1989.

[BG90] L. Bachmair and H. Ganzinger. On restrictions of ordered paramodulation with simplification. In M. E. Stickel, editor, *Proceedings of the 10th International Conference on Automated Deduction*, volume 449 of *Lecture Notes in Artificial Intelligence*, pages 427–441, Kaiserslautern, Germany, July 1990. Springer Verlag.

[BG94] L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994.

[BM97] B. Buchberger and M. Marin. Proving by simplification. In *Proceedings of the First International Theorema Workshop, RISC-Linz Technical Report 97-20*, Hagenberg, Austria, 9–10 June 1997.

[BN98] F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, New York, 1998.

[BS01] F. Baader and W. Snyder. Unification theory. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 8, pages 445–532. Elsevier Science, 2001.

[Buc96] B. Buchberger. Mathematica as a rewrite language. In T. Ida, A. Ohori, and M. Takeichi, editors, *Proceedings of the 2nd Fuji International Workshop on Functional and Logic Programming*, pages 1–13, Shonan Village Center, Japan, 1–4 November 1996. World Scientific.

[Buc01] B. Buchberger. Personal communication, 2001.

[Der82] N. Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 17(3):279–301, 1982.

[DJ90] N. Dershowitz and J.-P. Jouannaud. Rewriting systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 243–320. Elsevier Science, Amsterdam, 1990.

[Gen95] M. R. Genesereth. Epilog for Lisp 2.0 Manual. Technical report, Epistemics Inc., Palo Alto, California, US, 1995.

[GF92] M. R. Genesereth and R. E. Fikes. Knowledge Interchange Format, Version 3.0 Reference Manual. Technical Report Logic-92-1, Computer Science Department, Stanford University, Stanford, California, US, June 1992.

[GW92] S. Ginsburg and X. S. Wang. Pattern matching by Rs-operations: Toward a unified approach to querying sequenced data. In *Proceedings of the 11th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 293–300, San Diego, California, US, 2–4 June 1992.

[Ham97] M. Hamana. Term rewriting with sequences. In *Proceedings of the First International Theorema Workshop, RISC-Linz Technical Report 97-20*, Hagenberg, Austria, 9–10 June 1997.

[HM01] P. Hayes and C. Menzel. Semantics of knowledge interchange format. http://reliant.teknowledge.com/IJCAI01/HayesMenzel-SKIF-IJCAI2001.pdf, 2001.

[KL80] S. Kamin and J.-J. Lévy. Two generalizations of the recursive path ordering. Unpublished note, Department of Computer Science, University of Illinois, Urbana, Illinois, US, 1980.

[Kut02a] T. Kutsia. Pattern unification with sequence variables and flexible arity symbols. In M. Ojeda-Asiego, editor, *Proceedings of the Workshop on Unification in Non-Classical Logics*, volume 66, issue 5 of *Electronic Notes on Theoretical Computer Science*. Elsevier Science, 2002.

[Kut02b] T. Kutsia. Unification in a free theory with sequence variables and flexible arity symbols and its extensions. SFB Report 02-6, Johannes Kepler University, Linz, Austria, 2002.

[Kut02c] T. Kutsia. Unification with sequence variables and flexible arity symbols and its extension with pattern-terms. In *Artificial Intelligence, Automated Reasoning and Symbolic Computation. Proceedings of Joint AICS'2002 – Calculemus'2002 conference*, volume 2385 of *Lecture Notes in Artificial Intelligence*, Marseille, France, 1–5 July 2002. Springer Verlag.

[MB95] G. Mecca and A. J. Bonner. Sequences, Datalog and transducers. In *Proceedings of the Fourteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 23–35, San Jose, California, US, 22–25 May 1995.

[NR92] R. Nieuwenhuis and A. Rubio. Basic superposition is complete. In B. Krieg-Brückner, editor, *Proceedings of the European Symposium of Programming*, volume 582 of *Lecture Notes in Computer Science*, Rennes, France, 1992. Springer Verlag.

[NR95] R. Nieuwenhuis and A. Rubio. Theorem proving with ordering and equality constrained clauses. *Journal of Symbolic Computation*, 19:321–351, 1995.

[NR01] R. Nieuwenhuis and A. Rubio. Paramodulation-based theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 7, pages 371–443. Elsevier Science, 2001.

[Rub95] A. Rubio. Theorem proving modulo associativity. In *Proceedings of the Conference of European Association for Computer Science Logic*, Lecture Notes in Computer Science, Paderborn, Germany, 1995. Springer Verlag.

[WB01] M. Widera and C. Beierle. A term rewriting scheme for function symbols with variable arity. Technical Report No. 280, Praktische Informatik VIII, FernUniversität Hagen, Germany, 2001.

[Wol99] S. Wolfram. *The Mathematica Book*. Cambridge University Press and Wolfram Research, Inc., fourth edition, 1999.