Unification Modulo Flatness*

Temur Kutsia

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

Abstract. Flat theory with sequence variables and flexible arity symbols has infinitary matching and unification type. Decidability of general unification is shown and a unification procedure to enumerate minimal complete set of unifiers is described. The flat matching procedure is compared with the flat matching algorithm implemented in the MATHEMATICA system.

1 Introduction

Symbol attributes are an important feature of the MATHEMATICA [11] system that specify how the function symbols should be treated during evaluation and pattern matching. One of such attributes is `Orderless`, which specifies that the order of arguments of a function with this attribute does not matter and allows them to be rearranged in trying to match patterns. Another attribute is `Flat`, allowing to flatten all the nested occurrences of a symbol with this attribute. In MATHEMATICA 4.2 there are 19 possible attributes, and they can also be combined with each other.

The work described in this paper was motivated by the `Flat` attribute of MATHEMATICA. Initially, our goal was to characterize pattern matching modulo flatness for terms (possibly) involving sequence variables, as it was implemented in MATHEMATICA. However, finally it evolved into a more general framework, equational unification with sequence variables and flat and free function symbols.

Sequence variables add flexibility and expressiveness into a language. They are used together with flexible arity symbols. Sequence variables can be instantiated by an arbitrary, possibly empty, sequence of terms, and flexible arity symbols can take an arbitrary finite, possibly empty, sequence of arguments. Unification with sequence variables is a quite hard problem: a particular case can be reduced to A-unification. The minimal complete set of solutions of a unification problem with sequence variables is infinite even for the free theory [8]. Here we show that in the flat theory not only unification, but also matching is infinitary. We prove decidability of flat unification and describe a minimal complete unification procedure.

Relations between decidability of unification/matching problems and unification/matching type have been studied in the literature (see, e.g., [6]). However,

* Supported by the Austrian Science Foundation (FWF) under Project SFB F1302.

to our knowledge, no theory with infinitary matching type, decidable unification, and minimal complete unification procedure has been described so far.

Besides, we also compare the general flat matching procedure with the MATHEMATICA flat matching algorithm. To our knowledge, the algorithm itself is nowhere described, but is briefly explained on examples in [11].

We also made an experimental implementation of the general flat unification procedure in a rule-based system FUNLOG [9] built on top of MATHEMATICA.

The paper is organized as follows: in Sect. 2 we give preliminary notions related with unification theory with sequence variables. Section 3 shows decidability of flat unification. In Sect. 4 a flat unification procedure is described. Sect. 5 discusses an implementation of flat functions in MATHEMATICA.

2 Preliminaries

We consider an alphabet 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 $\mathcal{F}_{\mathrm{Flex}}$. We denote by $\mathcal{V}$ the set of variables $\mathcal{V}_{\mathrm{Ind}} \cup \mathcal{V}_{\mathrm{Seq}}$ and by $\mathcal{F}$ the set of function symbols $\mathcal{F}_{\mathrm{Fix}} \cup \mathcal{F}_{\mathrm{Flex}}$. A term (over $\mathcal{F}$ and $\mathcal{V}$) is defined recursively as follows:

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

f is called the head of $f(t_1, \ldots, t_n)$. Function symbols with the fixed arity 0 are called constants. For a fixed arity symbol f, $ar(f)$ denotes its arity. The set of all terms over $\mathcal{F}$ and $\mathcal{V}$ is denoted by $\mathcal{T}(\mathcal{F}, \mathcal{V})$. An equation (over $\mathcal{F}$ and $\mathcal{V}$) is a pair $\{s, t\}$, denoted $s \simeq t$, where $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V}) \setminus \mathcal{V}_{\mathrm{Seq}}$. If not otherwise stated, the following symbols, with or without indices, are used as metavariables: x and y – over individual variables; $\overline{x}$, $\overline{y}$ and $\overline{z}$ – over sequence variables; v – over (individual or sequence) variables; f, g and h – over function symbols; s, t, r, q – over terms. We use some other denotations as well: Let Q be a term, a sequence of terms, or a set of terms. Then 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,
$vars(Q)$ – the set $ivars(Q) \cup svars(Q)$,
$fix(Q)$ – the set of all fixed arity function symbols occurring in Q,
$flex(Q)$ – the set of all flexible arity function symbols occurring in Q.

We assume that the reader is familiar with the standard notions of unification theory [4]. Here we generalize some of them for sequence variables and flexible arity symbols.

Definition 1. *A binding is either a pair* $x \mapsto s$ *where* $x \in \mathcal{V}_{\mathrm{Ind}}$ *and* s *is a term with* $s \notin \mathcal{V}_{\mathrm{Seq}}$, $s \neq x$, *or an expression* $\overline{x} \mapsto \ulcorner s_1, \ldots, s_n \urcorner$[1] *where* $\overline{x} \in \mathcal{V}_{\mathrm{Seq}}$ *and* $s_1, \ldots, s_n$ *is a (possibly empty) sequence of terms such that* $s_1 \neq \overline{x}$ *if* $n = 1$.

A substitution is a finite set of bindings $\{x_1 \mapsto s_1, \ldots, x_n \mapsto s_n, \overline{x}_1 \mapsto \ulcorner t^1_1, \ldots, t^1_{k_1} \urcorner, \ldots, \overline{x}_m \mapsto \ulcorner t^m_1, \ldots, t^m_{k_m} \urcorner\}$ *where* $n, m \geq 0$ *and* $x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_m$ *are distinct variables.*

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 $t\theta$, is defined recursively as follows:

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

Example 1. Let $\theta = \{x \mapsto a, y \mapsto f(\overline{x}), \overline{x} \mapsto \ulcorner\urcorner, \overline{y} \mapsto \ulcorner a, f(\overline{x}), b \urcorner\}$. Then $f(x, \overline{x}, g(y, y), \overline{y})\theta = f(a, g(f(\overline{x}), f(\overline{x})), a, f(\overline{x}), b)$.

Instance of an equation $s \simeq t$ with respect to a substitution θ is defined as $(s \simeq t)\theta = s\theta \simeq t\theta$.

For a substitution σ, the domain is the set of variables $dom(\sigma) = \{v \mid v\sigma \neq v\}$, the codomain is the set of terms $cod(\sigma) = \{v\sigma \mid v \in dom(\sigma)\}$[2], and the range is the set of variables $ran(\sigma) = vars(cod(v\sigma))$.

Definition 2. *Let* $\theta = \{x_1 \mapsto s_1, \ldots, x_n \mapsto s_n, \overline{x_1} \mapsto \ulcorner t^1_1, \ldots, t^1_{k_1} \urcorner, \ldots, \overline{x_m} \mapsto \ulcorner t^m_1, \ldots, t^m_{k_m} \urcorner\}$ *and* $\lambda = \{y_1 \mapsto r_1, \ldots, y_{n'} \mapsto r_{n'}, \overline{y_1} \mapsto \ulcorner q^1_1, \ldots, q^1_{k'_1} \urcorner, \ldots, \overline{y_{m'}} \mapsto \ulcorner q^{m'}_1, \ldots, q^{m'}_{k'_{m'}} \urcorner\}$ *be two substitutions. Then the composition of* θ *and* λ *is the substitution, denoted by* $\theta \circ \lambda$, *obtained from the set*

$$\{\, x_1 \mapsto s_1\lambda,\ \ldots,\ x_n \mapsto s_n\lambda,\ \overline{x_1} \mapsto \ulcorner t^1_1\lambda, \ldots, t^1_{k_1}\lambda \urcorner,\ \ldots,\ \overline{x_m} \mapsto \ulcorner t^m_1\lambda, \ldots, t^m_{k_m}\lambda \urcorner,$$
$$y_1 \mapsto r_1,\ \ldots,\ y_{n'} \mapsto r_{n'},\ \overline{y_1} \mapsto \ulcorner q^1_1, \ldots, q^1_{k'_1} \urcorner,\ \ldots,\ \overline{y_{m'}} \mapsto \ulcorner q^{m'}_1, \ldots, q^{m'}_{k'_{m'}} \urcorner \}$$

by deleting

- *all the bindings* $x_i \mapsto s_i\lambda$ $(1 \leq i \leq n)$ *for which* $x_i = s_i\lambda$,
- *all* $\overline{x_i} \mapsto \ulcorner t^i_1\lambda, \ldots, t^i_{k_i}\lambda \urcorner$*-s* $(1 \leq i \leq m)$ *for which* $k_i = 1$ *and* $\overline{x}_i = t^i_1\lambda$,
- *all the bindings* $y_i \mapsto r_i$ $(1 \leq i \leq n')$ *such that* $y_i \in \{x_1, \ldots, x_n\}$,
- *all the bindings* $\overline{y_i} \mapsto \ulcorner q^i_1, \ldots, q^i_{k'_i} \urcorner$ $(1 \leq i \leq m')$ *such that* $\overline{y_i} \in \{\overline{x_1}, \ldots, \overline{x_m}\}$.

An *equational theory* is defined by a set of equations E (called identities) over $\mathcal{F}$ and $\mathcal{V}$. It is the least congruence relation on $\mathcal{T}(\mathcal{F}, \mathcal{V}) \setminus \mathcal{V}_{\mathrm{Seq}}$, that is closed under substitution and contains E, and it will be denoted by $\simeq_E$. If $s \simeq_E t$, then

[1] To improve the readability, we write sequences that bind sequence variables between $\ulcorner$ and $\urcorner$.

[2] Note that codomain is a set of terms, not a set of terms and sequences of terms, e.g. $cod(\{x \mapsto f(a),\ \overline{x} \mapsto \ulcorner a, a, b \urcorner\}) = \{f(a), a, b\}$.

we say that the term s is equal modulo E to the term t. In the following, we will often slightly abuse the notion of an equational theory by also calling a set E an equational theory, or E-theory. For a given set of equations E, we denote by $sig(E)$ the set of all function symbols occurring in E. Solving equations in an E-theory is called *E-unification*. The fact that the equation $s \simeq t$ has to be solved in an E-theory is written as $s \simeq^?_E t$.

Definition 3. *Let E be an equational theory and $\mathcal{F}$ be a signature containing $sig(E)$. An E-unification problem over $\mathcal{F}$ is a finite set of equations $\Gamma = \{s_1 \simeq^?_E t_1, \ldots, s_n \simeq^?_E t_n\}$. A substitution θ is called an E-unifier of Γ iff $s_i\theta \simeq_E t_i\theta$ for all $1 \leq i \leq n$. The set of all E-unifiers of Γ is denoted by $\mathcal{U}_E(\Gamma)$, and Γ is* E-unifiable *(*E-solvable*) iff $\mathcal{U}_E(\Gamma) \neq \emptyset$.*

Definition 4. *A substitution θ is more general than σ on a finite set of variables $\mathcal{X}$ modulo a theory E, denoted $\theta \trianglelefteq^{\mathcal{X}}_E \sigma$, iff there exists a substitution λ such that*

- *for all $\overline{x} \in \mathcal{X}$, $\overline{x} \mapsto \ulcorner\urcorner \notin \lambda$; there exist terms $t_1, \ldots, t_n, s_1, \ldots, s_n, n \geq 0$, such that $\overline{x}\sigma = t_1, \ldots, t_n$, $\overline{x}\theta \circ \lambda = s_1, \ldots, s_n$, and for each $1 \leq i \leq n$, either t_i and s_i are the same sequence variables, or $t_i \simeq_E s_i$;*
- *for all $x \in \mathcal{X}$, $x\sigma \simeq_E x\theta \circ \lambda$.*

Example 2. Let $\theta = \{\overline{x} \mapsto \overline{y}\}$, $\sigma = \{\overline{x} \mapsto \ulcorner a, b\urcorner, \ \overline{y} \mapsto \ulcorner a, b\urcorner\}$, $\eta = \{\overline{x} \mapsto \ulcorner\urcorner, \overline{y} \mapsto \ulcorner\urcorner\}$, $\mathcal{X} = \{\overline{x}, \overline{y}\}$, $E = \emptyset$. Then $\theta \trianglelefteq^{\mathcal{X}}_E \sigma$ and $\theta \not\trianglelefteq^{\mathcal{X}}_E \eta$.

The strict part of $\trianglelefteq^{\mathcal{X}}_E$ is denoted by $\vartriangleleft^{\mathcal{X}}_E$. The relation $\trianglelefteq^{\mathcal{X}}_E$ is a quasi-ordering.

Definition 5. *A set of substitutions Σ is called minimal with respect to a set of variables $\mathcal{X}$ modulo an equational theory E iff two distinct elements of Σ are incomparable with respect to $\trianglelefteq^{\mathcal{X}}_E$, i.e., for all σ, $\theta \in \Sigma$, $\sigma \trianglelefteq^{\mathcal{X}}_E \theta$ implies $\sigma = \theta$.*

Definition 6. *Let Γ be a E-unification problem over $\mathcal{F}$ and let $\mathcal{X} = vars(\Gamma)$. The minimal complete set of E-unifiers of Γ, denoted $mcu_E(\Gamma)$, is an minimal set of substitutions with respect to $\mathcal{X}$ modulo E, satisfying E-correctness ($mcu_E(\Gamma) \subseteq \mathcal{U}_E(\Gamma)$) and E-completeness (for each $\sigma \in \mathcal{U}_E(\Gamma)$ there exists $\theta \in mcu_E(\Gamma)$ such that $\theta \trianglelefteq^{\mathcal{X}}_E \sigma$).*

An E-unification problem Γ is called a general E-unification problem iff $sig(\Gamma) \backslash sig(E)$ contains arbitrary (fixed or flexible arity) function symbols, where $sig(\Gamma)$ is a set of all function symbols occurring in Γ.

Flat theory, or briefly F-theory, is defined as $E = \{f(\overline{x}, f(\overline{y}), \overline{z}) \simeq f(\overline{x}, \overline{y}, \overline{z})\}$, and f is called a flat flexible arity symbol.

It should be noted that although (free or flat) unification with sequence variables and flexible arity symbols looks similar to A-unification, there are essential differences illustrated by the following example (even without sequence variables). Let $\Gamma = \{f(x, f(y, z)) \simeq^?_E f(f(a, b), c)\}$ be a unification problem, where x, y, z are individual variables, and a, b, c are constants. First, assume E is an associative theory with f the associative symbol. Then $mcu_E(\Gamma) = \{\{x \mapsto$

$a, y \mapsto b, z \mapsto c\}\}$. Next, let $E = \emptyset$. Then f is a free function symbol and $mcu_E(\Gamma) = \emptyset$. Finally, assume E is a flat theory with f the flat flexible arity symbol. Then there are 23 substitutions in the minimal complete set of solutions: $mcu_E(\Gamma) = \{\{x \mapsto f(), y \mapsto f(), z \mapsto f(a,b,c)\}, \{x \mapsto f(), y \mapsto a, z \mapsto f(a,b,c)\}, \{x \mapsto f(), y \mapsto f(a), z \mapsto f(a,b,c)\} \ldots, \{x \mapsto f(a,b,c), y \mapsto f(), z \mapsto f()\}\}$.

3 Decidability

To show decidability of a general F-unification problem Γ, we first reduce it by unifiability preserving transformation to a simpler unification problem, and then show decidability of the reduced problem.

3.1 Reduction

First we define an operation on terms called flattening. It replaces a term of the form $f(\tilde{t}, f(\tilde{s}), \tilde{r})$, where f is a flat flexible arity symbol and $\tilde{t}, \tilde{s}$ and $\tilde{r}$ are (possibly empty) sequences of terms, with the term $f(\tilde{t}, \tilde{s}, \tilde{r})$. Given a term t, we denote by $flt(t)$ the term obtained from t by flattening all its subterms until impossible. Obviously $t \simeq_F flt(t)$.

Let Γ be a general F-unification problem $\{s \simeq^?_F t\}$. Then we denote by Δ the F-unification problem $\{flt(s) \simeq^?_F flt(t)\}$. It is easy to see that the following theorem holds:

Theorem 1. *Γ is solvable iff Δ is solvable.*

Now we reduce Δ to another general F-unification problem Φ by introducing a new flat symbol *seq* and a new unary symbol g_h for each free flexible arity symbol h in Δ, and replacing each term $h(r_1, \ldots, r_n)$ in Δ by $g_h(seq(r_1, \ldots, r_n))$.

Sequence variables occur in Φ only as direct arguments of terms with the head $f_1, \ldots, f_k$, or *seq*, where $f_1, \ldots, f_k$, $k \geq 1$, are all flat flexible arity symbols in Δ. There are no free flexible arity function symbols in Φ. We impose *individual variable restrictions* on Φ demanding that for a solution θ of Φ and for any individual variable x, $x\theta$ must not have *seq* as the head.

Theorem 2. *Δ is solvable iff Φ with individual variable restrictions is solvable.*

Remark. Note that solvability of Φ without individual variable restrictions does not imply solvability of Δ: Let Δ be $f(h(x)) \simeq^?_F f(h(a,b))$, with flat f and free flexible arity h. Then Φ is $f(g_h(seq(x))) \simeq^?_F f(g_h(seq(a,b)))$, where g_h is introduced to replace h. It is clear that Δ does not have a solution, while $\{x \mapsto seq(a,b)\}$ is a solution of Φ, because the flatness of *seq* implies $seq(seq(a,b)) \simeq_F seq(a,b)$.

Next, we construct a general F-unification problem that is F-unifiable (without restrictions) iff Φ with individual variable restrictions is F-unifiable.

Let T be a finite set of terms consisting of the following elements:

1. a new constant c,
2. exactly one term of the form $h(y_1, \ldots, y_{ar(h)})$ for each $h \in fix(\Phi)$ such that $y_1, \ldots, y_{ar(h)}$ are distinct individual variables occurring neither in the other terms from T nor in Φ, and
3. exactly one term of the form $h(\overline{x})$ for each $h \in flex(\Phi) \setminus \{seq\}$ such that $\overline{x}$ is a new sequence variable occurring neither in the other terms from T nor in Φ.

Theorem 3. *Let Φ be $\{s_1 \simeq_F^? s_2\}$ with $ivars(\Phi) = \{x_1, \ldots, x_n\}$ and $g \in \mathcal{F}_{\mathrm{Fix}}$ be a new $(n+1)$-ary symbol. Then Φ with individual variable restrictions is solvable iff there exist $r_1, \ldots, r_n \in T$ such that the general F-unification problem Ψ of the form $\{g(s_1, x_1, \ldots, x_n) \simeq_F^? g(s_2, r_1, \ldots, r_n)\}$ is solvable.*

3.2 Decidability of the Reduced Problem

We have to show that unifiability of an F-unification problem Ψ of the form $\{g(s_1, x_1, \ldots, x_n) \simeq_F^? g(s_2, t_1, \ldots, t_n)\}$ is decidable. Ψ has the following properties:

1. The signature of Ψ consists of fixed arity and flat flexible arity function symbols. The set of all flat flexible arity function symbols of Ψ contains *seq* and at least one other function symbol. There are no free flexible arity function symbols in the signature of Ψ.
2. $\{x_1, \ldots, x_n\} = ivars(s_1, s_2)$ and for all $1 \leq i, j \leq n$, if $i \neq j$ then $x_i \neq x_j$.
3. For all $1 \leq i \leq n$, the head of the term t_i belongs to the set $fix(s_1, s_2) \cup \{c\} \cup flex(s_1, s_2) \setminus \{seq\}$.

We will use the combination method introduced by Baader and Schulz in [3] to show that solvability of Ψ is decidable[3].

Linear constant restrictions (*lcr* in short) are induced by a linear order $<$ on the set of variables and constants, demanding that, for a unifier θ, a constant d and a variable v, d must not occur in $v\theta$ if $d > v$.

The combination method is formulated as follows:

Theorem 4 (Combination Method). *Let $E_1, \ldots, E_n$ be equational theories over disjoint signatures such that solvability of E_i-unification problems with linear constant restriction is decidable for $i = 1, \ldots, n$. Then unifiability is decidable for the combined theory $E_1 \cup \ldots \cup E_n$.*

Let $\{f_1, \ldots, f_k, seq\} = flex(\Psi)$, $k \geq 1$. Let for all $1 \leq i \leq k$, $\mathcal{F}_i$ be the set $\{f_i\}$, $\mathcal{F}_{k+1}$ be $\{seq\}$, and $\mathcal{F}_{k+2}$ be $fix(\Psi)$. Let for all $1 \leq i \leq k+1$, E_i be an equational theory with sequence variables over the signature $\mathcal{F}_i$ and E_{k+2} be a free theory (without sequence variables) over $\mathcal{F}_{k+2}$. Then we can consider Ψ as a unification problem in the combined theory $E_1 \cup \cdots \cup E_{k+2}$. Since $\mathcal{F}_1, \ldots, \mathcal{F}_{k+2}$ are pairwise disjoint, by Theorem 4, to prove decidability of Ψ in $E_1 \cup \cdots \cup E_{k+2}$

[3] In [3] the combination method was introduced for theories without sequence variables, but it remains valid for theories with sequence variables as well.

we need to show for each $1 \leq i \leq k+2$ that solvability of E_i-unification problem with lcr is decidable.

For each $1 \leq i \leq k+1$, solvability of E_i-unification with lcr is equivalent to solvability of word equations with lcr that is decidable (see [2]).

The E_{k+2}-unification problem is a Robinson unification problem. Decidability of Robinson unification with lcr is shown in [2].

Thus, solvability of Ψ is decidable, which by Theorem 1, Theorem 2 and Theorem 3, implies the following result:

Theorem 5. *Solvability of general flat unification is decidable.*

4 Unification Procedure

In this section we design a general F-unification procedure based on projection, flattening, and transformation rules. Each of the rules have one of the following forms: $\Gamma \rightsquigarrow \bot$ or $\Gamma \rightsquigarrow \langle\langle \Delta_1, \sigma_1\rangle, \ldots, \langle \Delta_n, \sigma_n\rangle\rangle$, where each of the successors Δ_i is either $\top$ or a new unification problem, and σ-s are substitutions.

4.1 Projection and Flattening Rules

The idea of projection [1] is to eliminate some sequence variables from the given problem. Let $\Pi(\Gamma)$ be a set of substitutions such that $\pi \in \Pi$ iff $dom(\pi) \subseteq svars(\Gamma)$ and $cod(\pi) = \emptyset$. Thus, Each $\pi \in \Pi$ replaces some sequence variables from Γ with the empty sequence. Flattening rule transforms a unification problem into its flattened from. The projection and flattening rules are shown in Fig. 1.

Projection:	$s \simeq^?_F t \rightsquigarrow \langle\langle s\pi_1 \simeq^?_F t\pi_1,\, \pi_1\rangle, \ldots,$ $\langle s\pi_k \simeq^?_F t\pi_k,\, \pi_k\rangle\rangle$	where $\{\pi_1, \ldots, \pi_k\} = \Pi(s \simeq^?_F t)$.
Flattening:	$s \simeq^?_F t \rightsquigarrow \langle\langle flt(s) \simeq^?_F flt(t), \varepsilon\rangle\rangle$	

Fig. 1. Projection and flattening rules.

4.2 Transformation

If Γ has a form $s \simeq^?_F t$, where s and t are either identical terms, terms with different heads, or terms with non-flat heads, then Γ is transformed by one of the transformation rules in Fig. 2 (note the usage of widening techniques similar to, e.g., [10] in the elimination rules for sequence variables).

Otherwise (i.e., if s and t have the same flat heads, or one of them is a variable and the other one has a flat head) we define transformation rules for Γ in Fig. 3, Fig. 4, and Fig. 5.

$$
\begin{array}{lll}
\textsf{Success:} & t \simeq^?_{\emptyset} t \rightsquigarrow \langle\langle \top,\ \varepsilon\rangle\rangle. & \\
 & x \simeq^?_F t \rightsquigarrow \langle\langle \top,\ \{x \mapsto t\}\rangle\rangle, & \text{if } x \notin \mathit{vars}(t). \\
 & t \simeq^?_F x \rightsquigarrow \langle\langle \top,\ \{x \mapsto t\}\rangle\rangle, & \text{if } x \notin \mathit{vars}(t). \\
 & & \\
\textsf{Failure:} & c_1 \simeq^?_F c_2 \rightsquigarrow \bot, & \text{if } c_1 \neq c_2. \\
 & x \simeq^?_F t \rightsquigarrow \bot, & \text{if } t \neq x \text{ and } x \in \mathit{vars}(t). \\
 & t \simeq^?_F x \rightsquigarrow \bot, & \text{if } t \neq x \text{ and } x \in \mathit{vars}(t). \\
 & h_1(\tilde{t}) \simeq^?_F h_2(\tilde{s}) \rightsquigarrow \bot, & \text{if } h_1 \neq h_2. \\
 & h() \simeq^?_F h(t_1, \tilde{t}) \rightsquigarrow \bot. & \\
 & h(t_1, \tilde{t}) \simeq^?_F h() \rightsquigarrow \bot. & \\
 & h(\overline{x}, \tilde{t}) \simeq^?_F h(s_1, \tilde{s}) \rightsquigarrow \bot, & \text{if } s_1 \neq \overline{x} \text{ and } \overline{x} \in \mathit{vars}(s_1). \\
 & h(s_1, \tilde{s}) \simeq^?_F h(\overline{x}, \tilde{t}) \rightsquigarrow \bot, & \text{if } s_1 \neq \overline{x} \text{ and } \overline{x} \in \mathit{vars}(s_1). \\
 & h(t_1, \tilde{t}) \simeq^?_F h(s_1, \tilde{s}) \rightsquigarrow \bot, & \text{if } t_1 \simeq^?_F s_1 \rightsquigarrow \bot. \\
 & & \\
\textsf{Eliminate:} & h(t_1, \tilde{t}) \simeq^?_F h(s_1, \tilde{s}) \rightsquigarrow \langle\langle g(\tilde{t}\sigma) \simeq^?_F g(\tilde{s}\sigma),\ \sigma\rangle\rangle, & \text{if } t_1 \simeq^?_F s_1 \rightsquigarrow \langle\langle \top,\ \sigma\rangle\rangle. \\
 & h(\overline{x}, \tilde{t}) \simeq^?_F h(\overline{x}, \tilde{s}) \rightsquigarrow \langle\langle h(\tilde{t}) \simeq^?_F h(\tilde{s}),\ \varepsilon\rangle\rangle. & \\
 & h(\overline{x}, \tilde{t}) \simeq^?_F h(s_1, \tilde{s}) \rightsquigarrow & \text{if } s_1 \notin \mathcal{V}_{\mathrm{Seq}} \text{ and } \overline{x} \notin \mathit{vars}(s_1), \\
 & \qquad \langle\langle h(\tilde{t}\sigma_1) \simeq^?_F h(\tilde{s}\sigma_1),\ \sigma_1\rangle, & \quad \text{where } \sigma_1 = \{\overline{x} \mapsto s_1\}, \\
 & \qquad \langle h(\overline{x}, \tilde{t}\sigma_2) \simeq^?_F h(\tilde{s}\sigma_2),\ \sigma_2\rangle\rangle, & \quad \sigma_2 = \{\overline{x} \mapsto \ulcorner s_1, \overline{x} \urcorner\}. \\
 & h(s_1, \tilde{s}) \simeq^?_F h(\overline{x}, \tilde{t}) \rightsquigarrow & \text{if } s_1 \notin \mathcal{V}_{\mathrm{Seq}} \text{ and } \overline{x} \notin \mathit{vars}(s_1), \\
 & \qquad \langle\langle h(\tilde{s}\sigma_1) \simeq^?_F h(\tilde{t}\sigma_1),\ \sigma_1\rangle, & \quad \text{where } \sigma_1 = \{\overline{x} \mapsto s_1\}, \\
 & \qquad \langle h(\tilde{s}\sigma_2) \simeq^?_F h(\overline{x}, \tilde{t}\sigma_2),\ \sigma_2\rangle\rangle, & \quad \sigma_2 = \{\overline{x} \mapsto \ulcorner s_1, \overline{x} \urcorner\}. \\
 & h(\overline{x}, \tilde{t}) \simeq^?_F h(\overline{y}, \tilde{s}) \rightsquigarrow & \text{where} \\
 & \qquad \langle\langle h(\tilde{t}\sigma_1) \simeq^?_F h(\tilde{s}\sigma_1),\ \sigma_1\rangle, & \quad \sigma_1 = \{\overline{x} \mapsto \overline{y}\}, \\
 & \qquad \langle h(\overline{x}, \tilde{t}\sigma_2) \simeq^?_F h(\tilde{s}\sigma_2),\ \sigma_2\rangle, & \quad \sigma_2 = \{\overline{x} \mapsto \ulcorner \overline{y}, \overline{x} \urcorner\}, \\
 & \qquad \langle h(\tilde{t}\sigma_3) \simeq^?_F h(\overline{y}, \tilde{s}\sigma_3),\ \sigma_3\rangle\ \rangle, & \quad \sigma_3 = \{\overline{y} \mapsto \ulcorner \overline{x}, \overline{y} \urcorner\}. \\
\textsf{Split:} & h(t_1, \tilde{t}) \simeq^?_F h(s_1, \tilde{s}) \rightsquigarrow & \text{if } t_1, s_1 \notin \mathcal{V} \text{ and} \\
 & \qquad \langle\langle h(r_1, \tilde{t}\sigma_1) \simeq^?_F h(q_1, \tilde{s}\sigma_1), \sigma_1\rangle, \ldots, & \quad t_1 \simeq^?_F s_1 \rightsquigarrow \langle\langle r_1 \simeq^?_F q_1,\ \sigma_1\rangle, \\
 & \qquad \langle h(r_k, \tilde{t}\sigma_k) \simeq^?_F h(q_k, \tilde{s}\sigma_k),\ \sigma_k\rangle\rangle, & \qquad \ldots, \langle r_k \simeq^?_F q_k,\ \sigma_k\rangle\rangle.
\end{array}
$$

Fig. 2. Transformation rules. $\tilde{t}$ and $\tilde{s}$ are possibly empty sequences of terms. $h \in \mathcal{F}$ is free. h_1, $h_2 \in \mathcal{F}$ can be free or flat. $g \in \mathcal{F}_{\mathrm{Flex}}$ is a new free symbol.

$$
\begin{array}{lll}
\mathsf{SuccessF}: & x \simeq^?_F f(x) \rightsquigarrow \langle\langle \top,\ \{x \mapsto f(x)\}\rangle\rangle. & \\
 & x \simeq^?_F t \rightsquigarrow \langle\langle \top,\ \{x \mapsto t\}\rangle\rangle, & \text{if } x \notin ivars(t) \text{ and } t \neq f(x). \\
 & f(x) \simeq^?_F x \rightsquigarrow \langle\langle \top,\ \{x \mapsto f(x)\}\rangle\rangle. & \\
 & t \simeq^?_F x \rightsquigarrow \langle\langle \top,\ \{x \mapsto t\}\rangle\rangle, & \text{if } x \notin ivars(t) \text{ and } t \neq f(x). \\
 & & \\
\mathsf{FailureF}: & x \simeq^?_F t \rightsquigarrow \bot, & \text{if } t \neq x,\ t \neq f(x),\ x \in ivars(t). \\
 & t \simeq^?_F x \rightsquigarrow \bot, & \text{if } t \neq x,\ t \neq f(x),\ x \in ivars(t). \\
 & f() \simeq^?_F f(t_1, \tilde{t}) \rightsquigarrow \bot, & \text{if } t_1 \notin \mathcal{V}. \\
 & f(t_1, \tilde{t}) \simeq^?_F f() \rightsquigarrow \bot, & \text{if } t_1 \notin \mathcal{V}. \\
 & f(\overline{x}, \tilde{t}) \simeq^?_F f(s_1, \tilde{s}) \rightsquigarrow \bot, & \text{if } s_1 \neq \overline{x},\ s_1 \neq g(\overline{x}),\ \overline{x} \in svars(s_1). \\
 & f(s_1, \tilde{s}) \simeq^?_F f(\overline{x}, \tilde{t}) \rightsquigarrow \bot, & \text{if } s_1 \neq \overline{x},\ s_1 \neq g(\overline{x}),\ \overline{x} \in svars(s_1). \\
 & f(t_1, \tilde{t}) \simeq^?_F f(s_1, \tilde{s}) \rightsquigarrow \bot, & \text{if } t_1 \simeq^?_F s_1 \rightsquigarrow \bot. \\
 & & \\
\mathsf{EliminateF}: & f() \simeq^?_F f(v, \tilde{t}) \rightsquigarrow & \text{where } v \in \mathcal{V} \text{ and} \\
 & \quad \langle\langle f() \simeq^?_F f(\tilde{t})\sigma, \sigma\rangle\rangle, & \quad \sigma = \{v \mapsto f()\}. \\
 & f(v, \tilde{t}) \simeq^?_F f() \rightsquigarrow & \text{where } v \in \mathcal{V} \text{ and} \\
 & \quad \langle\langle f(\tilde{t}\sigma) \simeq^?_F f(), \sigma\rangle\rangle, & \quad \sigma = \{v \mapsto f()\}. \\
 & f(t_1, \tilde{t}) \simeq^?_F f(s_1, \tilde{s}) \rightsquigarrow & \text{if } t_1, s_1 \notin \mathcal{V} \text{ and} \\
 & \quad \langle\langle f(\tilde{t}\sigma) \simeq^?_F f(\tilde{s}\sigma),\ \sigma\rangle\rangle, & \quad t_1 \simeq^?_F s_1 \rightsquigarrow \langle\langle \top,\ \sigma\rangle\rangle. \\
 & f(v, \tilde{t}) \simeq^?_F f(v, \tilde{s}) \rightsquigarrow & \text{where } v \in \mathcal{V}. \\
 & \quad \langle\langle f(\tilde{t}) \simeq^?_F f(\tilde{s}),\ \varepsilon\rangle\rangle. & \\
 & f(x, \tilde{t}) \simeq^?_F f(y, \tilde{s}) \rightsquigarrow & \text{where } x \neq y \text{ and} \\
 & \quad \langle\langle f(\tilde{t}\sigma_1) \simeq^?_F f(y, \tilde{s}\sigma_1),\ \sigma_1\rangle, & \quad \sigma_1 = \{x \mapsto f()\}, \\
 & \quad \langle f(\tilde{t}\sigma_2) \simeq^?_F f(\tilde{s}\sigma_2),\ \sigma_2\rangle, & \quad \sigma_2 = \{x \mapsto y\}, \\
 & \quad \langle f(\tilde{t}\sigma_3) \simeq^?_F f(\tilde{s}\sigma_3),\ \sigma_3\rangle, & \quad \sigma_3 = \{x \mapsto f(y)\}, \\
 & \quad \langle f(x, \tilde{t}\sigma_4) \simeq^?_F f(\tilde{s}\sigma_4),\ \sigma_4\rangle, & \quad \sigma_4 = \{x \mapsto f(y, x)\}, \\
 & \quad \langle f(x, \tilde{t}\sigma_5) \simeq^?_F f(\tilde{s}\sigma_5),\ \sigma_5\rangle, & \quad \sigma_5 = \{y \mapsto f()\}, \\
 & \quad \langle f(\tilde{t}\sigma_6) \simeq^?_F f(\tilde{s}\sigma_6),\ \sigma_6\rangle, & \quad \sigma_6 = \{y \mapsto f(x)\}, \\
 & \quad \langle f(\tilde{t}\sigma_7) \simeq^?_F f(y, \tilde{s}\sigma_7),\ \sigma_7\rangle\rangle, & \quad \sigma_7 = \{y \mapsto f(x, y)\}. \\
 & f(x, \tilde{t}) \simeq^?_F f(s_1, \tilde{s}) \rightsquigarrow & \text{where } s_1 \notin \mathcal{V} \text{ and} \\
 & \quad \langle\langle f(\tilde{t}\sigma_1) \simeq^?_F f(s_1, \tilde{s}\sigma_1),\ \sigma_1\rangle, & \quad \sigma_1 = \{x \mapsto f()\}, \\
 & \quad \langle f(\tilde{t}\sigma_2) \simeq^?_F f(\tilde{s}\sigma_2),\ \sigma_2\rangle, & \quad \sigma_2 = \{x \mapsto s_1\}, \\
 & \quad \langle f(\tilde{t}\sigma_3) \simeq^?_F f(\tilde{s}\sigma_3),\ \sigma_3\rangle, & \quad \sigma_3 = \{x \mapsto f(s_1)\}, \\
 & \quad \langle f(x, \tilde{t}\sigma_4) \simeq^?_F f(\tilde{s}\sigma_4),\ \sigma_4\rangle\rangle, & \quad \sigma_4 = \{x \mapsto f(s_1, x)\}. \\
 & f(t_1, \tilde{t}) \simeq^?_F f(x, \tilde{s}) \rightsquigarrow & \text{where } t_1 \notin \mathcal{V} \text{ and} \\
 & \quad \langle\langle f(t_1, \tilde{t}\sigma_1) \simeq^?_F f(\tilde{s}\sigma_1),\ \sigma_1\rangle, & \quad \sigma_1 = \{x \mapsto f()\}, \\
 & \quad \langle f(\tilde{t}\sigma_2) \simeq^?_F f(\tilde{s}\sigma_2),\ \sigma_2\rangle, & \quad \sigma_2 = \{x \mapsto t_1\}, \\
 & \quad \langle f(\tilde{t}\sigma_3) \simeq^?_F f(\tilde{s}\sigma_3),\ \sigma_3\rangle, & \quad \sigma_3 = \{x \mapsto f(t_1)\}, \\
 & \quad \langle f(\tilde{t}\sigma_4) \simeq^?_F f(x, \tilde{s}\sigma_4),\ \sigma_4\rangle\rangle, & \quad \sigma_4 = \{x \mapsto f(t_1, x)\}.
\end{array}
$$

Fig. 3. Transformation rules for the F-unification. $\tilde{t}$ and $\tilde{s}$ are possibly empty sequences of terms. $f, g \in \mathcal{F}_{\text{Flex}}$ are flat.

$$
\begin{array}{lll}
\textsf{EliminateF} & f(\overline{x},\tilde{t})\simeq^?_F f(y,\tilde{s}) \rightsquigarrow & \text{where} \\
\textsf{(continued):} & \quad \langle\langle f(\tilde{t}\sigma_1)\simeq^?_F f(y,\tilde{s}\sigma_1),\ \sigma_1\rangle, & \quad \sigma_1 = \{\overline{x} \mapsto f()\}, \\
& \quad \langle f(\tilde{t}\sigma_2)\simeq^?_F f(\tilde{s}\sigma_2),\ \sigma_2\rangle, & \quad \sigma_2 = \{\overline{x} \mapsto y\}, \\
& \quad \langle f(\tilde{t}\sigma_3)\simeq^?_F f(\tilde{s}\sigma_3),\ \sigma_3\rangle, & \quad \sigma_3 = \{\overline{x} \mapsto f(y)\}, \\
& \quad \langle f(\overline{x},\tilde{t}\sigma_4)\simeq^?_F f(\tilde{s}\sigma_4),\ \sigma_4\rangle, & \quad \sigma_4 = \{\overline{x} \mapsto f(y,\overline{x})\}, \\
& \quad \langle f(\overline{x},\tilde{t}\sigma_5)\simeq^?_F f(y,\tilde{s}\sigma_5),\ \sigma_5\rangle, & \quad \sigma_5 = \{\overline{x} \mapsto \ulcorner f(),\overline{x}\urcorner\}, \\
& \quad \langle f(\overline{x},\tilde{t}\sigma_6)\simeq^?_F f(\tilde{s}\sigma_6),\ \sigma_6\rangle, & \quad \sigma_6 = \{\overline{x} \mapsto \ulcorner y,\overline{x}\urcorner\}, \\
& \quad \langle f(\overline{x},\tilde{t}\sigma_7)\simeq^?_F f(\tilde{s}\sigma_7),\ \sigma_7\rangle, & \quad \sigma_7 = \{\overline{x} \mapsto \ulcorner y,f(\overline{x})\urcorner\}, \\
& \quad \langle f(\overline{x},\tilde{t}\sigma_8)\simeq^?_F f(\tilde{s}\sigma_8),\ \sigma_8\rangle, & \quad \sigma_8 = \{\overline{x} \mapsto \ulcorner f(y),\overline{x}\urcorner\}, \\
& \quad \langle f(\overline{x},\tilde{t}\sigma_9)\simeq^?_F f(\tilde{s}\sigma_9),\ \sigma_9\rangle, & \quad \sigma_9 = \{\overline{x} \mapsto \ulcorner f(y),f(\overline{x})\urcorner\}, \\
& \quad \langle f(\overline{x},\tilde{t}\sigma_{10})\simeq^?_F f(\tilde{s}\sigma_{10}),\ \sigma_{10}\rangle, & \quad \sigma_{10} = \{y \mapsto f()\}, \\
& \quad \langle f(\tilde{t}\sigma_{11})\simeq^?_F f(\tilde{s}\sigma_{11}),\ \sigma_{11}\rangle, & \quad \sigma_{11} = \{y \mapsto f(\overline{x})\}, \\
& \quad \langle f(\tilde{t}\sigma_{12})\simeq^?_F f(y,\tilde{s}\sigma_{12}),\ \sigma_{12}\rangle\rangle, & \quad \sigma_{12} = \{y \mapsto f(\overline{x},y)\}. \\
& f(y,\tilde{t})\simeq^?_F f(\overline{x},\tilde{s}) \rightsquigarrow & \text{where} \\
& \quad \langle\langle f(y,\tilde{t}\sigma_1)\simeq^?_F f(\tilde{s}\sigma_1),\ \sigma_1\rangle, & \quad \sigma_1 = \{\overline{x} \mapsto f()\}, \\
& \quad \langle f(\tilde{t}\sigma_2)\simeq^?_F f(\tilde{s}\sigma_2),\ \sigma_2\rangle, & \quad \sigma_2 = \{\overline{x} \mapsto y\}, \\
& \quad \langle f(\tilde{t}\sigma_3)\simeq^?_F f(\tilde{s}\sigma_3),\ \sigma_3\rangle, & \quad \sigma_3 = \{\overline{x} \mapsto f(y)\}, \\
& \quad \langle f(\tilde{t}\sigma_4)\simeq^?_F f(\overline{x},\tilde{s}\sigma_4),\ \sigma_4\rangle, & \quad \sigma_4 = \{\overline{x} \mapsto f(y,\overline{x})\}, \\
& \quad \langle f(y,\tilde{t}\sigma_5)\simeq^?_F f(\overline{x},\tilde{s}\sigma_5),\ \sigma_5\rangle, & \quad \sigma_5 = \{\overline{x} \mapsto \ulcorner f(),\overline{x})\urcorner\}, \\
& \quad \langle f(\tilde{t}\sigma_6)\simeq^?_F f(\overline{x},\tilde{s}\sigma_6),\ \sigma_6\rangle, & \quad \sigma_6 = \{\overline{x} \mapsto \ulcorner y,\overline{x}\urcorner\}, \\
& \quad \langle f(\tilde{t}\sigma_7)\simeq^?_F f(\overline{x},\tilde{s}\sigma_7),\ \sigma_7\rangle, & \quad \sigma_7 = \{\overline{x} \mapsto \ulcorner y,f(\overline{x})\urcorner\}, \\
& \quad \langle f(\tilde{t}\sigma_8)\simeq^?_F f(\overline{x},\tilde{s}\sigma_8),\ \sigma_8\rangle, & \quad \sigma_8 = \{\overline{x} \mapsto \ulcorner f(y),\overline{x}\urcorner\}, \\
& \quad \langle f(\tilde{t}\sigma_9)\simeq^?_F f(\overline{x},\tilde{s}\sigma_9),\ \sigma_9\rangle, & \quad \sigma_9 = \{\overline{x} \mapsto \ulcorner f(y),f(\overline{x})\urcorner\}, \\
& \quad \langle f(\tilde{t}\sigma_{10})\simeq^?_F f(\overline{x},\tilde{s}\sigma_{10}),\ \sigma_{10}\rangle, & \quad \sigma_{10} = \{y \mapsto f()\}, \\
& \quad \langle f(\tilde{t}\sigma_{11})\simeq^?_F f(\tilde{s}\sigma_{11}),\ \sigma_{11}\rangle, & \quad \sigma_{11} = \{y \mapsto f(\overline{x})\}, \\
& \quad \langle f(y,\tilde{t}\sigma_{12})\simeq^?_F f(\tilde{s}\sigma_{12}),\ \sigma_{12}\rangle\rangle, & \quad \sigma_{12} = \{y \mapsto f(\overline{x},y)\}. \\
& f(\overline{x},\tilde{t})\simeq^?_F f(\overline{y},\tilde{s}) \rightsquigarrow & \text{where } \overline{x} \neq \overline{y} \text{ and} \\
& \quad \langle\langle f(\tilde{t}\sigma_1)\simeq^?_F f(\overline{y},\tilde{s}\sigma_1),\ \sigma_1\rangle, & \quad \sigma_1 = \{\overline{x} \mapsto f()\}, \\
& \quad \langle f(\tilde{t}\sigma_2)\simeq^?_F f(\tilde{s}\sigma_2),\ \sigma_2\rangle, & \quad \sigma_2 = \{\overline{x} \mapsto \overline{y}\}, \\
& \quad \langle f(\tilde{t}\sigma_3)\simeq^?_F f(\tilde{s}\sigma_3),\ \sigma_3\rangle, & \quad \sigma_3 = \{\overline{x} \mapsto f(\overline{y})\}, \\
& \quad \langle f(\overline{x},\tilde{t}\sigma_4)\simeq^?_F f(\tilde{s}\sigma_4),\ \sigma_4\rangle, & \quad \sigma_4 = \{\overline{x} \mapsto f(\overline{y},\overline{x})\}, \\
& \quad \langle f(\overline{x},\tilde{t}\sigma_5)\simeq^?_F f(\overline{y},\tilde{s}\sigma_5),\ \sigma_5\rangle, & \quad \sigma_5 = \{\overline{x} \mapsto \ulcorner f(),\overline{x})\urcorner\}, \\
& \quad \langle f(\overline{x},\tilde{t}\sigma_6)\simeq^?_F f(\tilde{s}\sigma_6),\ \sigma_6\rangle, & \quad \sigma_6 = \{\overline{x} \mapsto \ulcorner \overline{y},\overline{x}\urcorner\}, \\
& \quad \langle f(\overline{x},\tilde{t}\sigma_7)\simeq^?_F f(\tilde{s}\sigma_7),\ \sigma_7\rangle, & \quad \sigma_7 = \{\overline{x} \mapsto \ulcorner \overline{y},f(\overline{x})\urcorner\}, \\
& \quad \langle f(\overline{x},\tilde{t}\sigma_8)\simeq^?_F f(\tilde{s}\sigma_8),\ \sigma_8\rangle, & \quad \sigma_8 = \{\overline{x} \mapsto \ulcorner f(\overline{y}),\overline{x}\urcorner\}, \\
& \quad \langle f(\overline{x},\tilde{t}\sigma_9)\simeq^?_F f(\tilde{s}\sigma_9),\ \sigma_9\rangle, & \quad \sigma_9 = \{\overline{x} \mapsto \ulcorner f(\overline{y}),f(\overline{x})\urcorner\}, \\
& \quad \langle f(\overline{x},\tilde{t}\sigma_{10})\simeq^?_F f(\tilde{s}\sigma_{10}),\ \sigma_{10}\rangle, & \quad \sigma_{10} = \{\overline{y} \mapsto f()\}, \\
& \quad \langle f(\tilde{t}\sigma_{11})\simeq^?_F f(\tilde{s}\sigma_{11}),\ \sigma_{11}\rangle, & \quad \sigma_{11} = \{\overline{y} \mapsto f(\overline{x})\}, \\
& \quad \langle f(\tilde{t}\sigma_{12})\simeq^?_F f(\overline{y},\tilde{s}\sigma_{12}),\ \sigma_{12}\rangle, & \quad \sigma_{12} = \{\overline{y} \mapsto f(\overline{x},\overline{y})\}, \\
& \quad \langle f(\overline{x},\tilde{t}\sigma_{13})\simeq^?_F f(\overline{y},\tilde{s}\sigma_{13}),\ \sigma_{13}\rangle, & \quad \sigma_{13} = \{\overline{y} \mapsto \ulcorner f(),\overline{y}\urcorner\}, \\
& \quad \langle f(\tilde{t}\sigma_{14})\simeq^?_F f(\overline{y},\tilde{s}\sigma_{14}),\ \sigma_{14}\rangle, & \quad \sigma_{14} = \{\overline{y} \mapsto \ulcorner \overline{x},\overline{y}\urcorner\}, \\
& \quad \langle f(\tilde{t}\sigma_{15})\simeq^?_F f(\overline{y},\tilde{s}\sigma_{15}),\ \sigma_{15}\rangle, & \quad \sigma_{15} = \{\overline{y} \mapsto \ulcorner \overline{x},f(\overline{y})\urcorner\}, \\
& \quad \langle f(\tilde{t}\sigma_{16})\simeq^?_F f(\overline{y},\tilde{s}\sigma_{16}),\ \sigma_{16}\rangle, & \quad \sigma_{16} = \{\overline{y} \mapsto \ulcorner f(\overline{x}),\overline{y}\urcorner\}, \\
& \quad \langle f(\tilde{t}\sigma_{17})\simeq^?_F f(\overline{y},\tilde{s}\sigma_{17}),\ \sigma_{17}\rangle\rangle, & \quad \sigma_{17} = \{\overline{y} \mapsto \ulcorner f(\overline{x}),f(\overline{y})\urcorner\}.
\end{array}
$$

Fig. 4. Transformation rules for the F-unification (continued). $\tilde{t}$ and $\tilde{s}$ are possibly empty sequences of terms. $f \in \mathcal{F}_{\mathrm{Flex}}$ is flat.

EliminateF (continued):	$f(\overline{x},\tilde{t})\simeq^?_F f(s_1,\tilde{s})$	$s_1 \notin \mathcal{V}$, $\overline{x} \notin svars(s_1)$
	$\rightsquigarrow$	or $s_1 = g(\overline{x})$, and
	$\langle\langle f(\tilde{t}\sigma_1)\simeq^?_F f(s_1,\tilde{s}\sigma_1),\ \sigma_1\rangle,$	$\sigma_1 = \{\overline{x} \mapsto f()\},$
	$\langle f(\tilde{t}\sigma_2)\simeq^?_F f(\tilde{s}\sigma_2),\ \sigma_2\rangle,$	$\sigma_2 = \{\overline{x} \mapsto s_1\},$
	$\langle f(\tilde{t}\sigma_3)\simeq^?_F f(\tilde{s}\sigma_3),\ \sigma_3\rangle,$	$\sigma_3 = \{\overline{x} \mapsto f(s_1)\},$
	$\langle f(\overline{x},\tilde{t}\sigma_4)\simeq^?_F f(\tilde{s}\sigma_4),\ \sigma_4\rangle,$	$\sigma_4 = \{\overline{x} \mapsto f(s_1,\overline{x})\},$
	$\langle f(\overline{x},\tilde{t}\sigma_5)\simeq^?_F f(s_1,\tilde{s}\sigma_5),\ \sigma_5\rangle,$	$\sigma_5 = \{\overline{x} \mapsto \ulcorner f(),\overline{x}\urcorner\},$
	$\langle f(\overline{x},\tilde{t}\sigma_6)\simeq^?_F f(\tilde{s}\sigma_6),\ \sigma_6\rangle,$	$\sigma_6 = \{\overline{x} \mapsto \ulcorner s_1,\overline{x}\urcorner\},$
	$\langle f(\overline{x},\tilde{t}\sigma_7)\simeq^?_F f(\tilde{s}\sigma_7),\ \sigma_7\rangle,$	$\sigma_7 = \{\overline{x} \mapsto \ulcorner s_1,f(\overline{x})\urcorner\},$
	$\langle f(\overline{x},\tilde{t}\sigma_8)\simeq^?_F f(\tilde{s}\sigma_8),\ \sigma_8\rangle,$	$\sigma_8 = \{\overline{x} \mapsto \ulcorner f(s_1),\overline{x}\urcorner\},$
	$\langle f(\overline{x},\tilde{t}\sigma_9)\simeq^?_F f(\tilde{s}\sigma_9),\ \sigma_9\rangle\rangle,$	$\sigma_9 = \{\overline{x} \mapsto \ulcorner f(s_1),f(\overline{x})\urcorner\}.$
	$f(t_1,\tilde{t})\simeq^?_F f(\overline{x},\tilde{s})$	$t_1 \notin \mathcal{V}$, $\overline{x} \notin svars(t_1)$
	$\rightsquigarrow$	or $t_1 = g(\overline{x})$, and
	$\langle\langle f(t_1,\tilde{t}\sigma_1)\simeq^?_F f(\tilde{s}\sigma_1),\ \sigma_1\rangle,$	$\sigma_1 = \{\overline{x} \mapsto f()\},$
	$\langle f(\tilde{t}\sigma_2)\simeq^?_F f(\tilde{s}\sigma_2),\ \sigma_2\rangle,$	$\sigma_2 = \{\overline{x} \mapsto t_1\},$
	$\langle f(\tilde{t}\sigma_3)\simeq^?_F f(\tilde{s}\sigma_3),\ \sigma_3\rangle,$	$\sigma_3 = \{\overline{x} \mapsto f(t_1)\},$
	$\langle f(\tilde{t}\sigma_4)\simeq^?_F f(\overline{x},\tilde{s}\sigma_4),\ \sigma_4\rangle,$	$\sigma_4 = \{\overline{x} \mapsto f(t_1,\overline{x})\},$
	$\langle f(t_1,\tilde{t}\sigma_5)\simeq^?_F f(\overline{x},\tilde{s}\sigma_5),\ \sigma_5\rangle,$	$\sigma_5 = \{\overline{x} \mapsto \ulcorner f(),\overline{x}\urcorner\},$
	$\langle f(\tilde{t}\sigma_6)\simeq^?_F f(\overline{x},\tilde{s}\sigma_6),\ \sigma_6\rangle,$	$\sigma_6 = \{\overline{x} \mapsto \ulcorner t_1,\overline{x}\urcorner\},$
	$\langle f(\tilde{t}\sigma_7)\simeq^?_F f(\overline{x},\tilde{s}\sigma_7),\ \sigma_7\rangle,$	$\sigma_7 = \{\overline{x} \mapsto \ulcorner t_1,f(\overline{x})\urcorner\},$
	$\langle f(\tilde{t}\sigma_8)\simeq^?_F f(\overline{x},\tilde{s}\sigma_8),\ \sigma_8\rangle,$	$\sigma_8 = \{\overline{x} \mapsto \ulcorner f(t_1),\overline{x}\urcorner\},$
	$\langle f(\tilde{t}\sigma_9)\simeq^?_F f(\overline{x},\tilde{s}\sigma_9),\ \sigma_9\rangle\rangle,$	$\sigma_9 = \{\overline{x} \mapsto \ulcorner f(t_1),f(\overline{x})\urcorner\}.$
SplitF:	$f(t_1,\tilde{t})\simeq^?_F f(s_1,\tilde{s}) \rightsquigarrow$	if $t_1, s_1 \notin \mathcal{V}$ and
	$\langle\langle f(r_1,\tilde{t}\sigma_1)\simeq^?_F f(q_1,\tilde{s}\sigma_1),\ \sigma_1\rangle,$	$t_1\simeq^?_F s_1 \rightsquigarrow$
	$\ldots,$	$\langle\langle r_1\simeq^?_F q_1,\ \sigma_1\rangle, \ldots$
	$\langle f(r_k,\tilde{t}\sigma_k)\simeq^?_F f(q_k,\tilde{s}\sigma_k),\ \sigma_k\rangle\rangle$	$\langle r_k\simeq^?_F q_k,\ \sigma_k\rangle\rangle.$

Fig. 5. Transformation rules for the F-unification (continued). $\tilde{t}$ and $\tilde{s}$ are possibly empty sequences of terms. $f, g \in \mathcal{F}_{\mathrm{Flex}}$ are flat.

4.3 Tree Generation

We design the unification procedure for the flat theory with sequence variables and flexible arity symbols as a tree generation process. The single steps in this process are projection, flattening, and transformation for the flat theory.

Each node of the tree is labeled either with a unification problem, $\top$ or $\bot$. The edges are labeled by substitutions. The nodes labeled with $\top$ or $\bot$ are terminal nodes. The nodes labeled with unification problems are non-terminal nodes. The children of a non-terminal node are constructed in the following way: Let Δ be a unification problem attached to a non-terminal node. First, we decide whether Δ is unifiable. If the answer is negative, we replace Δ with the new label $\bot$. Otherwise, we apply flattening, projection, or transformation on Δ and get $\langle\langle \Phi_1, \sigma_1\rangle, \ldots, \langle \Phi_n, \sigma_n\rangle\rangle$. Then the node Δ has n children, labeled respectively with $\Phi_1, \ldots, \Phi_n$, and the edge to the Φ_i node is labeled with σ_i for all $1 \leq i \leq n$.

We design the general unification procedure for a general flat unification problem Γ as a breadth first (level by level) tree generation process. We label the root of the tree with Γ (zero level). First level nodes of the tree are obtained from the original problem by projection. Starting from the second level, we apply only flattening and transformation steps to a unification problem of each node, first flattening it and then transforming the flattened problem, thus getting new successor nodes. The branch which ends with a node labeled by $\top$ is called a successful branch. The branch which ends with a node labeled by $\bot$ is a failed branch. For each node in the tree, we compose substitutions (top-down) displayed on the edges of the branch that leads to this node, flatten all the terms in the codomain of the composition, and attach the obtained substitution to the node together with the unification problem the node was labeled with. The empty substitution is attached to the root. For a node N, the substitution attached to N in such a way is called the associated substitution of N. Let $\Sigma(\Gamma)$ be the set of all substitutions associated with the $\top$ nodes. We call the tree a unification tree for Γ and denote it $utree(\Gamma)$.

The following lemma plays the key role in proving that $\Sigma(\Gamma)$ is a complete set of F-unifiers of Γ.

Lemma 1. *Let Γ be a general flat unification problem and let $\mathcal{X} = vars(\Gamma)$. Then for every $\gamma \in \mathcal{U}_F(\Gamma)$ there exists a branch β in $utree(\Gamma)$ with the following property: if Φ is a unification problem occurring in β with the associated substitution ϕ, then $\phi \trianglelefteq^{\mathcal{X}}_F \gamma$.*

Using this lemma, the following theorem can be easily proved:

Theorem 6 (Completeness). *$\Sigma(\Gamma)$ is a complete set of F-unifiers for Γ.*

The next example shows that the unification procedure is not minimal:

Example 3. Let Γ be $f(x) \simeq^?_F f(y)$, $f \in \mathcal{F}_{\mathrm{Flex}}$ being flat. Then, among the other solutions, the procedure returns the unifiers $\sigma = \{x \mapsto y\}$ and $\theta = \{x \mapsto f(), y \mapsto f()\}$. Obviously $\sigma \trianglelefteq^{vars(\Gamma)}_F \theta$.

The next example shows that F-matching is (at least) infinitary.

Example 4. Let Γ be $f(\overline{x}) \simeq^?_F f(a)$, $f \in \mathcal{F}_{\mathrm{Flex}}$ being flat. Then the procedure computes infinitely many unifiers of Γ:

$$\{\overline{x} \mapsto a\}, \{\overline{x} \mapsto f(a)\}, \{\overline{x} \mapsto \ulcorner a, f()\urcorner\}, \{\overline{x} \mapsto \ulcorner f(a), f()\urcorner\}, \{\overline{x} \mapsto \ulcorner f(), a\urcorner\},$$
$$\{\overline{x} \mapsto \ulcorner f(), f(a)\urcorner\}, \{\overline{x} \mapsto \ulcorner f(), a, f()\urcorner\}, \{\overline{x} \mapsto \ulcorner f(), f(a), f()\urcorner\}, \ldots.$$

The reason of such a behavior is that the term $f()$ occurs in transformation substitutions. Skipping any of the rules involving $f()$ would lead to incompleteness:

Example 5. The unique solution $\{x \mapsto f()\}$ of $f(x, a) \simeq^?_F f(a)$ ($f \in \mathcal{F}_{\mathrm{Flex}}$ being flat) can not be computed without the transformation substitution $\{x \mapsto f()\}$.

It can be proved that the minimal complete set of F-unifiers of a general flat unification problem Γ exists (see [7]). Therefore, we can refine the unification procedure to compute the minimal complete set of unifiers. Let $\mathcal{X} = vars(\Gamma)$. During the tree generation process, at each step when a new solution appears, check whether the set of already computed solutions Ξ of Γ is minimal or not (it is decidable). If Ξ is minimal, then continue the process, otherwise minimize Ξ as follows: Let $\sigma \in \Xi$ be a substitution such that for some other substitution $\theta \in \Xi$ we have $\theta \leq_F^{\mathcal{X}} \sigma$. Let β be the branch in $utree(\Gamma)$ such that σ is attached to the leaf of β. Let $\eta_1, \ldots, \eta_n$ be the edges in β, starting from the root. Let k be the number such that none of $\eta_n, \eta_{n-1}, \ldots, \eta_{n-k}$ have a sibling in $utree(\Gamma)$, but $\eta_{n-(k+1)}$ has. Then we delete $\eta_n, \eta_{n-1}, \ldots, \eta_{n-k}$ from $utree(\Gamma)$ and continue the tree generation process. Let $utree_{min}(\Gamma)$ be the tree constructed in such a manner and let $\Sigma_{min}(\Gamma)$ denote the set of all substitutions that are associated with $\top$ nodes in $utree_{min}(\Gamma)$. Then Theorem 6 and the construction of Σ_{min} imply the following result:

Theorem 7. $\Sigma_{min}(\Gamma) = mcu_F(\Gamma)$.

Bürckert et al [5] investigated properties of equational theories important for unification theory. These properties can easily be extended for theories with sequence variables and flexible arity symbols (see [7]). It can be proved that the flat theory is regular, collapse free, almost collapse free, Noetherian, and strongly complete, but neither permutative, simple, finite, nor Ω-free.

5 Flat Functions in Mathematica

The Mathematica system [11] implements matching modulo flatness. It is not hard to observe that the algorithm is not complete. It does not match, for instance, $f(x, a)$ to $f(a)$, $f(x, g(x))$ to $f(a, g(a))$, or $f(\overline{x}, g(\overline{x}))$ to $f(a, g(f(a)))$, where f is flat and g is free.

The main difference between the F-matching procedure and the Mathematica flat matching is that the latter does not consider transformation rules involving $f()$. It makes Mathematica flat matching finitary.

Another difference is in the case where an individual variable x matches a single argument s_1 in a term with a flat head f. The F-matching procedure returns four substitutions as it is shown in the sixth case of Eliminate in Fig. 3, while the Mathematica matching algorithm chooses only the last two of those four. If in the same situation we have a sequence variable $\overline{x}$, the F-matching procedure tries nine different ways to resolve the case (the first rule of Eliminate in Fig. 5), while Mathematica would choose only the second and sixth.

On the other hand, Mathematica can verify that each solution computed by the F-matching procedure is correct, e.g., it sees $f(x, g(x))\{x \mapsto a\}$ and $f(a, g(a))$ as identical expressions, although, as it was already mentioned, the Mathematica matching algorithm can not compute $\{x \mapsto a\}$.

6 Conclusion

We described a flat theory with sequence variables and flexible arity symbols. In this theory solvability of the general unification problem is decidable, unification and matching types are infinitary, and a minimal complete unification procedure exists. A practically useful restriction of the procedure can be identified, which describes the meaning of flatness implemented in the MATHEMATICA system.

7 Acknowledgments

I wish to thank Prof. Bruno Buchberger for his support, Mircea Marin for interesting discussions and Hans-Jürgen Bürckert for giving useful references.

References

1. H. Abdulrab and J.-P. Pécuchet. Solving word equations. *J. of Symbolic Computation*, 8(5):499–522, 1990.
2. F. Baader and K. U. Schulz. General A- and AX-unification via optimized combination procedures. In *Proc. of the Second International Workshop on Word Equations and Related Topics, IWWERT'91*, volume 677 of *LNCS*, pages 23–42, Rouen, France, 1992. Springer Verlag.
3. F. Baader and K. U. Schulz. Unification in the union of disjoint equational theories: Combining decision procedures. *J. of Symbolic Computation*, 21(2):211–244, 1996.
4. 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.
5. H.-J. Bürckert, A. Herold, and M. Schmidt-Schauß. On equational theories, unification and (un)decidability. In C. Kirchner, editor, *Unification*, pages 69–116. Academic Press, 1990.
6. H.-J. Bürckert and M. Schmidt-Schauß. On the solvability of equational problems. Technical Report SR-89-07, University of Kaiserslautern, Germany, 1989.
7. T. Kutsia. Solving and proving in equational theories with sequence variables and flexible arity symbols. Technical Report 02-09, Research Institute for Symbolic Computation, Johannes Kepler University, Linz, Austria, 2002.
8. T. Kutsia. Unification with sequence variables and flexible arity symbols and its extension with pattern-terms. In J. Calmet, B. Benhamou, O. Caprotti, L. Henocque, and V. Sorge, editors, *Artificial Intelligence, Automated Reasoning and Symbolic Computation. Proc. of Joint AISC'2002 and Calculemus'2002 Conferences*, volume 2385 of *LNAI*, pages 290–304, Marseille, France, 1–5 July 2002. Springer Verlag.
9. M. Marin and T. Kutsia. Programming with transformation rules. In T. Jebelean and V. Negru, editors, *Proc. of the 5th International Workshop on Symbolic and Numeric Algorithms for Scientific Computing*, Timisoara, Romania, 1–4 October 2003. This volume.
10. J. Siekmann. String unification. Research paper, Essex University, 1975.
11. S. Wolfram. *The Mathematica Book*. Cambridge University Press and Wolfram Research, Inc., fourth edition, 1999.