Unification of Stratified Second-Order Terms

Manfred Schmidt-Schauß,
Fachbereich Informatik,
Johann Wolfgang-Goethe-Universität Frankfurt
60054 Frankfurt
E-mail: schauss@informatik.uni-frankfurt.de

Abstract.
We consider the problem of unifying a set of equations between second-order terms. Terms are constructed from function symbols, constant symbols and variables, and furthermore using monadic second-order variables that may stand for a term with one hole, and parametric terms.
We consider stratified systems, where for every first-order and second-order variable, the string of second-order variables on the path from the root of a term to every occurrence of this variable is always the same.
It is shown that unification of stratified second-order terms is decidable by describing a non-deterministic decision algorithm that eventually uses Makanin's algorithm for deciding the unifiability of word equations. As a generalization, we show that the method can be used as a unification procedure for non-stratified second-order systems, and describe conditions for termination in the general case.

1 Introduction.

Finding solutions for equations between terms containing variables is a basic mechanism for several algorithms in automated deduction, term rewriting, and logic programming. This operation is commonly called unification of terms.
Most work has been devoted to unification of first order terms, though unification of higher order terms has always attracted some interest. This interest is now increasing, which is due to the popularization of higher order programming languages and higher-order extension of logic programming and automated deduction systems. A further source of higher order problems is the schematization of terms in term rewriting and automated deduction .
The general second order unification problem is undecidable [Gol81], even if only one binary function symbol is available (besides constants and unary function symbols), (cf. [Fa91]). However, the semantics used there to show undecidability is different from ours. It appears to be an open problem whether the second order unification problem under the restricted “term with one hole” semantics is decidable. (see also [Com94b]).
The “one-hole-semantics” used in this paper for second-order variables is an interesting specialization of the general semantics, and also appears in a natural way in schematizing infinite sets of terms. For example, investigating unification of first-order terms under two-sided distributivity (cf. [Sz82, Si89, BS93, Con93, Sch94] reduces to a unification problem for second-order terms with the “one-hole-semantics”.

The formulation of the unification problem for which we will show decidability in this paper is as follows. We have function symbols of arbitrary arity in the signature. Terms may contain second order variables in function positions. Such a second order variable always stands for a term with exactly one hole. We consider the restriction that for every first or second-order variable x, the second-order context is the same for every occurrence of x. I.e., for every first-order and second-order variable, the string of second-order variables on the path from the root of a term to every occurrence of this variable is always the same. For example f(x, X(h(y))) = f(X(Y(z)), x) is an admissable equation, but f(X(x),x) = f(y,y) is not, since in the last equation, there are two paths from the root to x, and the first contains a second order variable, whereas the second path doesn't. It is eventually shown, that these stratified unification problems can be decided by Makanin's decision algorithm for unification [Ma71, Sch93].

The presented algorithm is a non-deterministic one and rather involved. It is not stream-lined for stratified systems, but can also handle more general situations, which may be helpful for parametric stratified systems (see section 9) or for non stratified second order systems (see section 10). We have resisted the temptation to built-in some obvious optimizations, instead we try to obtain maximal clarity of presentation. Optimizations are left for further research. For the same reason, the algorithm is designed as a decision algorithm, not as an algorithm generating a complete set of unifiers. Nevertheless, the presented algorithm can be modified to be complete.
We will show in section 9 that a restricted form of parametric terms is also permitted in the input system. A parametric term is either $f(t_1,\ldots t_{i-1},\Omega,t_{i+1},\ldots, t_n)$, (where Ω stands for the hole,) or a concatenation of parametric terms $p_1 \bullet \ldots \bullet p_n$, or a power p^n of a parametric terms p. Basically, the restriction on the input system are that i) in parametric terms of the form p^n, p should not represent a power, ii) for two occurrences p^n and q^n, the parametric terms p and q are identical, and iii) parametric terms do not occur under a second order variable. This permits, for example, to encode terms like $(f^n g)^m(x)$ for unary function symbols.
For input systems satisfying these restrictions, we will also show that unifiability is decidable.
A further generalization is to consider systems that are not stratified, but dynamically stratified, i.e., the algorithm can always use a rule that makes the system smaller. We will show that the unification problem for these systems is decidable.

As an example for a run of the non-deterministic algorithm on a stratified system consider the second-order unification problem {X(f(x,y)) = f(a, X(z)), X(f(a,b)) = f(u,v)}. This problem satisfies the restrictions above. There is one second-order cycle for the second order variable X. Now it is possible to solve the cycle w.r.t. X in the first equation, which means to remove the X in the first equation and to replace X by the parametric term $(f(a, \Omega))^n$ in the second, where n is an (positive) integer variable. This gives the equations $\{f(x,y) = f(a,z), (f(a, \Omega))^n (f(a,b)) = f(u,v)\}$. Decomposing the first equation gives $\{x = a, y = z, (f(a, \Omega))^n (f(a,b)) = f(u,v)\}$. Splitting n into the cases n= 1 and n= 1+m gives two cases: i) $\{x = a, y = z, (f(a, \Omega)) (f(a,b)) = f(u,v)\}$ and thus $\{x = a, y = z, a = u, f(a,b) = v\}$. ii) $\{x = a, y = z, f(a, (f(a, \Omega))^m (f(a,b))) = f(u,v)\}$. This gives $\{x = a, y = z, a = u, v = (f(a, \Omega))^m (f(a,b))\}$.

It is not easy to compare the expressivity of the different versions of stratified systems with existing approaches. If compared with respect to the possibilitiy to express infinite sets of terms

in a schematic way, it appears to be impossible to encode terms like $(f^n g)^m$ with usual stratified second order systems. However, this is possible in parametric stratified systems (see section 9.), as well as in dynamically stratified systems (see section 10).
Most papers on the subject have an explicit representation mechanism for infinite sets of terms ([Gra88], [CH91], [CHK90], [Her92], [KiH91], [Sa92], [So94], [Her92], [Com95]), and permit no second order variables. B. Gramlich ([Gra88]) permits second-order variables and describes restrictions such that a first order unification algorithm can be used. In the work of H. Kirchner [KiH91]), there are meta-variables permitted that are similar to second-order variables, but the meta-variables come in general with a domain of terms that are permitted for instantiation. H. Comon [Com94a] considers completion methods of rewrite systems with membership constraints and has to solve unification problems containing second order variables. The occurrences of these second order variables are restricted to be always of the same shape, i.e. if X(s) and X(t) occur in a system, then s and t must be syntactically equal. Since the constraints in [Com94a] can also express terms like (f*g)*, the approach is not directly comparable with the approach in this paper.
W. Farmer [Fa88] has worked on second order unification problems, and has shown that unification of monadic second order terms is decidable. These terms are built from second order variables (arity may be arbitrary) and monadic function symbols. The difference to our problem is that in [Fa88], function symbols of arity ≥ 2 are not permitted, nested applications of second order variables are permitted, and that the semantics of second-order variables is different from ours: a second order variable of arity n may be instantiated with a term function of smaller arity.

The method we will use to solve the second order unification problem has its roots in the paper of W. Farmer [Fa88] in that we make extensive use of parametric terms and parametric words, and in work on unification, where the technique of non-deterministic transformation rules is elaborated (cf. Sch89, BS93), which means that the algorithm makes a lot of guessing steps.

The paper is structured as follows: After setting the scene in section 2, we present the subalgorithm in the following sections. Section 3 contains the decomposition rules, section 4 the cycle elimination method, section 5 the method to eliminate clusters of maximal second order variables. Section 6 and 7 contain a description of the transformation into string unification, section 8 summarizes and discusses the main algorithm, and section 9 and 10 present some generalizations.

2. Preliminaries

We use a fixed signature that contains an infinite number of free constants and some functions symbols with fixed arity, where there is at least one function symbols with arity greater than 1 (i.e., arity(f) ≥ 2). We also have an infinite supply of first order variables, usually denoted x,y,z, an infinite supply of second-order monadic variables, which we denote by upper-case letters like X, Y, Z, and an infinite supply of integer variables, usually denoted n.
We define second order terms and parametric terms in a mutual recursive way:

2.1 Definition.

A **parametric term** can be:

i) Ω: the trivial parametric term
ii) a **basic parametric term** $f(t_1,\ldots t_{i-1},\Omega,t_{i+1},\ldots, t_n)$, where t_i are second order terms (the **parameters**),
iii) a concatenation of two parametric terms $p \bullet q$.
iv A power p^n of a parametric term p, where n is an integer variable (representing a positive natural number).

A **term** can be:

i) a first-order variable,
ii) $f(t_1,\ldots, t_n)$ for terms t_i,
iii) X(t), where X is a second-order variable and t a term,
iv) p(t), where p is a parametric term and t a term.

At some places it is necessary to consider Ω as a trivial parametric term. In general we will assume that parametric terms are not trivial. ♦

An example for a syntactically permitted term is $g(a,\Omega)\ (f(X(x),\ f(\Omega,Y(X(y)))\ (Y(g(a)))\ )\ )$, where X,Y are SO-variables, x,y are FO-variables, g is a unary function symbol, f is a binary function symbol, and a is a constant. Note that $f(X(\Omega),x)$ is not a parametric term, whereas $f(X(x), \Omega)$ and $f(X(g(\Omega)(a)), x)$ are permitted.

We call a term **ground**, iff it does not contain variables. A term is called a **first order** term, iff it does not contain parametric terms nor second order variables. Every ground term can be reduced to a first order ground term using the following reductions:

i) $f(t_1,\ldots t_{i-1},\Omega,t_{i+1},\ldots, t_n)(s) \rightarrow f(t_1,\ldots t_{i-1},s,t_{i+1},\ldots, t_n)$
ii) $\Omega(t) \rightarrow t$
iii) $p \bullet q(t) \rightarrow p(q(t))$
iv) $p^1(t) \rightarrow p(t)$
v) $p^N(t) \rightarrow p(p^{N-1}(t))$, where N is an integer greater than 1.

We need extended substitutions as solutions to unification problems. Therefore we need ground parametric terms, which can be seen as lambda-abstracted terms, and are also called "terms with one hole". Then a term with a hole is a first-order term t (up to an occurrence of Ω), and Ω occurs exactly once in t. A (second-order ground-) substitution is a mapping that can be represented by a finite set of pairs (x_i, t_i), (X_i, p_i), (n,N) where t_i is a first-order ground term, p_i is a first-order ground parametric term, and N is a positive integer. In the following, we will use always ground substitutions, i.e. substitutions that instantiate ground terms for first order variables, and ground terms with one hole for second order variables.
The application of a substitution σ to a second-order term s is as follows: If $\sigma x = t$, then all x's in s are replaced by t; if $\sigma X = p$, then every occurrence X(u) in s is replaced by $p[\Omega \rightarrow u]$, i.e., the usual β-reduction, and all integer variables are replaced by the corresponding integer.
Below we sometimes speak of an extension of a substitution σ to new variables. This should indicate that σ has the same value on the old variables, but the value on the new ones can be freshly defined.

A ground parametric term can be considered as a string of basic ground parametric terms, where composition is viewed as concatenation. The "letters" are the basic ground parametric terms and concatentation of string corresponds to composition. Using this point of view, we can also speak of powers, i.e. strings of the form s^n, where power means n-fold concatenation.

For the purposes of the algorithm, we simplify parametric terms and applications of parametric terms to terms as follows:
Using associativity, we can flatten concatenations and applications. Thus every application of a parametric term is of the form $(p_1 \bullet \ldots \bullet p_n)(t)$, where p_i are basic parametric terms, or proper powers p^m, and t does not start with a parametric term.
We will use positions in terms for parametric terms and second-order variables:
It is sufficient to have a 0 for the function position, and to assume that a concatenation is a list of parametric terms enumerated beginning from 1, and that p^n is seen as an application pow(p,n), i.e. p is at position 1. The other positions are as usual. Thus for example in $((g(x,\Omega)) \bullet (g(\Omega,y)))\ (X(a))$, x is at position 0;1;1, y at position 0;2;2, X at position 1;0, and a at position 1;1.
We need a notion of depth of a term that is adapted to our unification algorithm. We will call this syntactic term measure p-depth of a term, and it corresponds to the length of the positions. It is defined on terms, where parametric terms are assumed to be flattened, i.e.:

p-depth(t) = 0 for constants and variables t,
$\text{p-depth}(f(t_1, \ldots, t_n)) = 1 + \max\{\text{p-depth}(t_i) \mid i=1,\ldots,n\}$,
$\text{p-depth}(p(t)) = 1 + \max\{\text{p-depth}(p), \text{p-depth}(t)\}$
$\text{p-depth}(f(t_1,\ldots,t_{j-1},\Omega,t_{j+1},\ldots,t_n) = 1 + \max\{\text{p-depth}(t_i) \mid i=1,\ldots,n \text{ and } i \neq j\}$,
$\text{p-depth}(p^n) = 1+\text{p-depth}(p)$,
$\text{p-depth}((p_1 \bullet \ldots \bullet p_n) = 1 + \max\{\text{p-depth}(p_i), i=1,\ldots,n\}$,
$\text{p-depth}(X(t)) = 1+\text{p-depth}(t)$.

Analogously, we will speak of the p-depth of a subterm in some term or of a position.
Note that the reduction to a flattened representation makes the p-depth smaller.
We will also use the common depth, called c-depth of some posititon within a term. This is the depth if ther temr is transformed into the form without flattening of parametric terms.

In the following we will work with systems of equations (and also disequations), i.e. a multi-set of pairs of terms, where the pairs of terms are interpreted as equations (or disequations, respectively). We will use Term(Γ) for the set of terms occurring in Γ and Var(Γ) for the set of variables occurring in Γ. A substitution σ is a (usual) **unifier** of Γ, if for every equation s = t , the terms s and t become equal after applying the substitution σ (and for every disequation $s \neq t$, we have $\sigma s \neq \sigma t$). We consider equations and disequations as symmetric. This makes the denotation of some unification rules simpler, but also implies some non-determinism, for example the rule Replace-Variable is non-deterministic, if the equation is x = y.

2.2 Definition.

For every position π of some term in Γ, the **second-order-prefix** of π is the string of second-order variables on a path from the root to this position. To be more rigorous, let SOP be a function computing this second-order prefix as follows:
$SOP(t,\pi) = \varepsilon$, if t is a variable or constant or Ω, and ε denotes the empty word.
$SOP(f(t_1,\ldots,t_n),k\bullet\pi) = SOP(t_k,\pi)$ for terms or basic parametric terms.
$SOP(X(t), 1\bullet\pi) = X\bullet SOP(t,\pi)$
$SOP(p(t), 0\bullet\pi) = SOP(p,\pi)$
$SOP((p_1\bullet\ldots\bullet p_n), k\bullet\pi) = SOP(p_k,\pi)$
A set of terms S is called **stratified**, iff
i) for every first and second-order variable x, the second-order-prefixes of every position of x in some term in S are the same.
ii) the second order-prefix of every Ω in some term from S is the empty string.
The system Γ is called **stratified**, iff the set of terms $Term(\Gamma)$ is stratified.
♦

Note that this property is invariant w.r.t. the flattening operation.

2.3 Unification Problem. The following problem is to be solved by the algorithm:
Input is a stratified system Γ of equations between second order terms that has no occurrences of parametric terms.
The issue is to decide, whether there exists a unifier of Γ. ♦

This does not mean that we can forget about parametric terms. The parametric terms are an important ingredient for the intermediate systems of equations and they are immediately introduced by the elimination rules for cycles between second order variables.
Furthermore we will also consider systems where parametric terms already occur in the input (see section 9)

3. Datastructure and Syntactic Decomposition Rules

In this section, we give a first step for deciding unifiability of stratified second order systems. In particular, we concentrate on eliminating cycles between second-order variables.

A ground parametric term p can be interpreted as a description of a set of words, where the alphabet is the set of basic ground parametric terms occurring in p. We say a word w is a **power**, if there is another word u, such that $w = u^k$ for some integer $k \geq 2$, (where exponentiation means k-fold concatenation of words), otherwise, we say w is a **non-power**. Furthermore, we say a string u is a **cyclic permutation** of a string w, if there are two strings u_1 and u_2, such that $u = u_1u_2$ and $w = u_2u_1$. The empty word is denoted as ε.

Now a more rigid explanation of the used datastructure for the algorithm will be given.

3.1 Definition. Datastructure and unifiers

The datastructure for the unification algorithm consists of a multi-set of equations between terms, of equations between parametric terms, and of disequations between basic parametric terms. We refer to the three different components of Γ as Γ_T for the term part, Γ_P for the parametric term part, and $\Gamma_{P,\neq}$ for the disequation part.
The substitution σ is an **unconstrained unifier** of Γ, if it solves all equations of terms as well as of the parametric terms.
An unconstrained unifier of Γ is a **unifier** of Γ, if for for every disequation $p \neq q$ in $\Gamma_{P,\neq}$, we have $\sigma p \neq \sigma q$, and for every term p^n we have that $\sigma(p)$ is a non-power. The set of unifiers of Γ is denoted as $U(\Gamma)$. ♦

Note that for the input system, the set of unifiers and unconstrained unifiers is the same, since no parametric terms are permitted in the input.

We will also use a restricted variant of second-order variables, so-called **F-variables** that can be viewed as abstracting a function symbol. An F-variable has a built-in arity n. A substitution also has to instantiate F-variables as follows: A substitution σ for a F-variable F can be represented as a lambda-expression as follows: $\sigma F = \lambda x_1,\ldots,x_n: f(s_1, \ldots s_m)$, where $\{s_1,\ldots,s_m\} = \{x_1,\ldots,x_n\} \cup M$, as multi-sets, where M is a multi-set of ground terms. The Application of σ to $F(t_1,\ldots,t_n)$ gives the term $f(s_1, \ldots s_m)$, where the x_i ´s are replaced by σt_i. Note that these F-variables can appear only in Γ_P, the only place where we have to deal with these variables is in section 5.

The sub-system Γ_P contains parametric terms. It is necessary to identify a top-layer of this parametric terms. The set of top basic parametric terms TBPTS for a parametric term p is defined as follows:

$TBPTS(p) = \{p\}$ if p is a basic parametric term.
$TBPTS(\Omega) = \emptyset$
$TBPTS(p \bullet q) = TBPT(Sp) \cup TBPTS(q)$
$TBPTS(p^n) = TBPTS(p)$

The set $TBPT(\Gamma_P)$ is defined as the union of all TBPTS(p) for all parametric terms p that occur in some equation $p = q$ (or $q = p$) in Γ_P.
This definition can be seen as distinguishing the top layer of a parametric term from the other layers.

3.2 Definition. We will show that the following syntactic **invariants** are maintained by all rules:

i) The set of $Term(\Gamma_T) \cup \{p \mid p \in TBPT(\Gamma_P)$ and $Var(p) \subseteq Var(\Gamma_T)\}$ is stratified. This includes that parametric terms are not below second-order variables.
ii) If there is some integer variable n, then for different occurrences p^n and q^n of n in Γ, the parametric terms p and q are syntactically equal.
iii) there is no power of the form $(p^n)^m$
iv) For all basic parametric terms p,q in $TBPT(\Gamma_P)$, either p is syntactically equal to q, or there is a disequation $p \neq q$ in $\Gamma_{P,\neq}$.

v) Every parametric term in TBPT(Γ_P) that has an occurrence of an F-variable, has also an occurrence of some SO-variable Z that does not occur in Γ_T. ♦

In the following we talk about sound and complete rules. A rule that transforms Γ into Γ' is **sound**, if every unconstrained unifier of Γ' is also an unconstrained unifier of Γ. A rule that can transform Γ into alternatives $\Gamma_1,\ldots,\Gamma_n$ is called **complete**, if for every unifier of Γ, there exists some Γ_i and some extension σ' of σ, such that σ and σ' coincide on the variables of Γ_T and the integer variables in Γ_P, and σ' is a unifier of Γ_i.
We will sometimes say that a rule stops with FAIL. In this case, we mean that it is detected, that the algorithm is on a wrong path. I.e., either the system is not unifiable, or the system is redundant.
In the presentation of rules and function we will sometimes use EXIT to indicate that this rule is exited. A further method to structure non-deterministic rules is to use the construct EITHER ... OR .... to indicate alternative executions.

Eliminating SO-cycles is a central method for this algorithm:
3.3 Definition. SO-cycles.
Let Γ be a system of equations.
An **SO-cycle** in Γ is a sequence of equations in Γ_T of the following form.

$$X_1(s_1) = t_1,\ X_2(s_2) = t_2,\ \ldots,X_n(s_n) = t_n,$$

For $i = 2,\ldots,n$, X_i occurs in t_{i-1} and X_1 occurs in t_n, such that at least one such occurrence of some X_i is not at the top.♦

Note that for SO-cycles in a stratified system of equations the occurrences of the X_i in t_{i-1} and X_1 in t_n are not under a further second-order variable.

3.4 Definition. Structure of the unification algorithm for stratified second order systems.
The main operations are:
1) Bring all equations into the form X(s) = t.
2) If there is an SO-cycle, eliminate it using the methods from section 4.
3) If all equations are in the form X(s) = t, and there is no SO-cycle, then use the methods of section 5) to eliminate one SO-variable.
4) If Γ_T is empty, use the method of section 7) to decide unifiability. ♦

The corresponding well-founded ordering on stratified systems Γ is based on the following measure μ.
It will be a lexicographic combination of the following well-founded orderings.
1) μ_1: number of second-order variables in Γ_T.
2) μ_2: number of first-order variables in Γ_T.
3) μ_3: number of integer variables in Γ. ♦

In this section we will need a local measure λ for proving termination of the decomposition rules.

Local measure λ: Built the multiset of the following multisets. For every equation in Γ_T take the multiset of the two p-depths of the the two terms; for every basic parametric term p in TBPT(Γ_P) take the multi-set of the p-depth of p. The ordering is the (doubly nested) multiset ordering generated by the ordering for natural numbers.

In the following we give some rules for removing all equations from Γ_T that are not of the form X(s) = t. A system of equations is called in **decomposed normal form**, if all equations are of the form X(s) = t.
As a shorthand for replacing a second order variable X by Ω, and immediate simplification, we will say that X is removed from Γ. We will do the same for integer variables n, if they are instantiated with 1.

3.5 Definition. Basic transformation rules

Rule *Replace-Variable.* If x = t is an equation in Γ, and x does not occur in t, then remove the equation x = t, and replace x by t everywhere in Γ.

Rule *Trivial-Equations*. Remove equations of the form x = x, where x is a variable.

Rule *Occur-Check.* If there is an equation x = t, such that t is not a variable, and x occurs in t, then FAIL: the system is not unifiable.

Rule *Decomposition*
If $f(s_1,\ldots,s_n) = f(t_1,\ldots,t_n)$ is an equation in Γ, then
remove the equation and add n equations $s_i = t_i$, i = 1,…,n to Γ.

Rule *Clash*. If $f(s_1,\ldots,s_n) = g(t_1,\ldots,t_m)$ is an equation in Γ, and $f \neq g$, then FAIL.
the system is not unifiable.

Rule *Constants*. If a =X(t) is in the system, then remove X from Γ.
If a = p(t) is in the system, where p is a (non-trivial) parametric term, then FAIL: the system is not unifiable.

Rule *Para-Clash*. If there is a disequation $p \neq p$ in $\Gamma_{P,\neq}$, then FAIL ♦

Note that the case x = X(x) cannot occur, since then the system would not be stratified.
The decomposition rule that replaces X(s) = X(t) by s = t is obviously correct, however, it cannot be used as a stand-alone rule, since the stratifiedness of Γ could be destroyed. These extended rules are reconsidered in section 10.

3.6 Lemma. The rules and transformations in 3.5 are sound and complete, and either fail or decrease μ or leave μ invariant and strictly decrease λ. Furthermore they preserve the invariants.
Proof. Soundness and Completeness are trivial.
Measure: Consider the rules that do not terminate with FAIL. The rule Replace-Variables makes μ_2 strictly smaller. The rules Decomposition and Trivial-Equations do not increase μ and strictly reduce λ.
Stratifiedness: The replacement of a first-order variable x leaves the property stratifiedness invariant, since for all occurrences of x the second order prefix is the empty string. The removal of a second-order variable is always done for all occurrences.

Parametric terms are not under a second-order variable, since no parametric terms are introduced, and since Variable-Replacement doesn't replace variables that are below second-order variables. ♦

The following is a modular procedure that adds equation between parametric terms to Γ_P. This procedure has to take care of identification of basic parametric terms. If these identifications are made without any precaution, then there may be a reintroduction of variables that are already eliminated from Γ, but are in Γ_P. Furthermore, the identification may reintroduce large terms from Γ_P. The following definition is designed siuch that the orderings have a chance to decrease.

3.7 Definition.

Rule "Add-Para-Equation" . The intention is to add an equation E to Γ_P, where $E = \{e_1 = e_2\}$ is an equation between parametric terms, and to adjust $\Gamma_{P,\neq}$.

If e_1 and e_2 are basic parametric terms, then

If the function symbol in e_1 and e_2 are different or the position of Ω in e_1 and e_2 is different , then FAIL

Add the equations resulting from decomposition to Γ and EXIT.

If one of the e_i is a basic parametric term, while the other one is not, then FAIL.

While there are syntactically different basic parametric terms p, q in $TBPT(E \cup \Gamma_P)$ such that $p \neq q$ is not in $\Gamma_{P,\neq}$, do:

If the top-level function symbols of p and q are different, then add $p \neq q$.

If the index of Ω in p and q is different, then add $p \neq q$.

If p is in TBPT(E), and q in $TBPT(\Gamma_P)$, and there is some SO-variable X in q such that X does not occur in Γ_T, then add $p \neq q$.

If p is in TBPT(E), and q in $TBPT(\Gamma_P)$, and there is some FO-variable x in q such that x does not occur in Γ_T, then add $p \neq q$.

Otherwise:

Either add $p \neq q$

Or for $p = f(s_1,\ldots,s_{i-1},\Omega,s_{i+1},\ldots,s_n)$ and $q = f(t_1,\ldots,t_{i-1},\Omega,t_{i+1},\ldots,t_n)$ add the n-1 equations $s_j = t_j$ for $j \neq i$,

and if p-depth(p) > p-depth(q) replace p by q everywhere in Γ_P

or if p-depth(q) ≥ p-depth(p) replace q by p everywhere in Γ_P.

Add E to Γ_P.

3.8 Lemma. The rule Add-Para-Equation is sound and complete.
Furthermore it leaves the invariants unchanged and does not increase the measure μ.

Proof.

Soundness is obvious, since disequations are ignored for soundness.

Completeness. We prove this for one identification step. We can assume that for all syntactically different basic parametric terms in $TBPT(\Gamma_P)$, there is an equation $p \neq q$.

Assume there is a unifier σ before the application.

If the top level function symbols of p and q or the indices are different, then $\sigma p \neq \sigma q$, and we can add $p \neq q$.

Let $p \in TBPT(E)$ and $q \in TBPT(\Gamma_P)$, such that some variable in q does not occur in Γ_T. Since σ is a unifier of Γ, and the variable does not occur in Γ_T, the instantiation of this variable does not influence the property of σ to unify the equations in Γ_T. The same holds for the Γ_P, since the property for σ to be a unifier of Γ_P depends only on the instantiation for the integer variables, since all different basic parametric terms are assumed to be different under σ. If $\sigma p = \sigma q$, then Lemma 3.9 below shows that there is a further substitution σ' that differs from σ only on the variable X (or x, respectively), and that is a unifier of the current system, but instantiates X (or x, respectively) in a different way, hence $\sigma' p = \sigma p \neq \sigma' q$

Measure μ is not increased, since the number of SO-variables and FO-variables in Γ_T is not increased, since it is explicitely forbidden by the rule to reintroduce variables from Γ_P into Γ_T.

Invariants. The only interesting part is that after the rule has been applied, for all $p,q \in TBPT(\Gamma_P)$, we have that either p and q are syntactically equal or the disequation $p \neq q$ is in $\Gamma_{P,\neq}$. ♦

3.9 Lemma. Let σ be a unifier of a set $\Gamma_{P,\neq}$.

i) Let X be a second order variable occurring in $\Gamma_{P,\neq}$. Then there is further unifier σ' that is equal to σ except for the variable X.

ii) Let x be a first order variable occurring in $\Gamma_{P,\neq}$. Then there is further unifier σ' that is equal to σ except for the variable x.

Proof. We can assume that σ is applied such that σ only instantiates the focussed variable X (or x). Furthermore we assume that the concatenation is rewritten as usual term application, and that there are no exponents.

i) Instead of σX we use $\sigma' X$, where the Ω in σX is replaced by $X'(\Omega)$, and X' is a new second-order-variable. Then σ' is a unifier of $\Gamma_{P,\neq}$.

The proof is by induction on the number of disequations in $\Gamma_{P,\neq}$ and then on the multiset of the depths of terms in $\Gamma_{P,\neq}$.

In the following we will also permit disequations between terms.

If there is some disequation with different top function symbols or a different first symbol of the position of Ω, remove this disequation, and then use induction. For every disequation $f(t_1,\ldots,t_n) \neq f(s_1,\ldots,s_n)$, there is some index i, such that $\sigma' s_i \neq \sigma' t_i$. We replace the disequation by this disequation, and can use induction, since the depths have decreased.

A disequation $f(\ldots) \neq f(\ldots)$ can be replaced by some disequation for subterms, since there must be subterms at the same argument position that are syntactically different after application of σ.

If there is a disequation of the form $f(\ldots) \neq g(\ldots)$, then we can eliminate this disequation.

The remaining case is that we have $X'(s) \neq X'(t)$ or $X'(s) \neq f(\ldots)$. In the first case we replace the disequation by $s \neq t$, and get a smaller problem. Finally, there are only disequations of the form $X'(s) \neq f(\ldots)$. If there are different top level function symbols on the right hand sides, then we can instantiate X' with $f(a, X''(\Omega)), \ldots)$ and get a fewer number of disequations after application of the operations for $f(\ldots) \neq g(\ldots)$. This instantiation is correct, since after the instantiation all the disequalites are satisfied.

Consider the case that there is only one function symbol f remaining on the right hand side.

If f is a constant, then instantiate X' by $g(a,\Omega)$, where a is some constant and g some function symbol of arity ≥ 1. Now we have constructed the desired instantiation.

If f is unary, then replace X' by g(a, Ω, …), where g is not unary, and all non-Ω arguments are some constant a. Such a function symbol exists by assumption on the signature. In this case we have constructed a unifier σ'.
If the function symbol f is not unary, and there is another function symbol g in the signature, then instantiate X' by g(a, Ω, …), where all non-Ω arguments are a constant a. Now we have constructed a unifier σ'.
If the function symbol f is not unary and f is the only function symbol in the signature, then we have to be careful. If there is some disequation, where X' occurs in the i^{th} argument on the right hand side, then we replace X' by f(…,a, …,X"(Ω),…), where the constant a is at the i^{th} position, and X" is a new SO-variable. Then we can again reduce the problem set by removing this disequation.
Now consider the case that X' does not occur in some right hand side. Let $X'(s_j) \neq f(t_{j,1},\ldots,t_{j,n})$ be the remaining disequations. Then we can replace X' by f(X"(Ω),a,…) and all remaining disequations by $X"(s_j) \neq t_{j,1}$ We have smaller right hand sides and can use induction.

ii) We consider the case, where the variable is a first order variable. Instead of σx we use σ'x, where some occurrence of some constant in the term σx is replaced by a new first order variable x'. Then σ' is a unifier of $\Gamma_{P,\neq}$.
If there is some disequation with different top function or a different index of the first Ω, remove this disequation, and then use induction. For every disequation $f(t_1,\ldots,t_n) \neq f(s_1,\ldots,s_n)$, there is some index i, such that $\sigma s_i \neq \sigma t_i$. We replace the disequation by this equation. Then we can use induction.
Note that we have now also disequation between terms.
If there is a disequation of the form f(…) ≠ g(…), then we can eliminate this disequation.
Disequations f(…) ≠ f(…) between terms can be replaced by some subterm disequation, where the subterms have to different.
The remaining case is that we have x' ≠ t. It is not possible that t is the variable x', since σ' is a unifier. If x' occurs in t, then we can remove this disequation. In order to construct a unifier we have to find a non-constant ground term that is different from the finitely many ground terms on the right hand sides. Since there exists an infinite number of different ground terms constructed from one constant and a function symbol of arity ≥ 1, this is a trivial task. ♦

We need some non-deterministic splitting operations on parametric terms in Γ for the case p(s) = q(t).

3.10 Definition.

i) *Split-first:* splits a non-trivial parametric term into first basic parametric term and rest using the following non-deterministic recursion equations.
split-first (p) = split-first-r(p,Ω)
split-first-r (p,r) = (p,r) if p is a basic parametric term.
split-first-r(p_1•…•p_n, r) := split-first-r(p_1, p_2• … •p_n•r))
split-first-r(p^n, r) :=split-first-r(p, r) :
and n is removed from Γ.

split-first-r(p^n, r) := split-first-r(p, p^m•r)
and p^n is replaced by p•p^m everywhere in Γ where m is a new integer variable.

ii) *Split-middle*: splits a parametric term into start, and an end, somewhere in the middle using the following non-deterministic recursion equations.
We assume that the argument is not trivial.

split-middle(p) = split-middle-r(Ω, p, Ω).
split-middle-r(s,p,r) = (s•p,r)
split-middle-r(s,p,r) = (s,p•r)
split-middle-r(s,p•q,r) = split-middle-r(s•p, q, r)
split-middle-r(s,p•q,r) = split-middle-r(s, p, q•r)
split-middle-r(s,p^n,r) = split-middle-r(s,p,r)
and n is removed from Γ.
split-middle-r(s,p^n,r) = split-middle-r(s•p^m,p,r)
and p^n is replaced by p^m•p everywhere in Γ
split-middle-r(s,p^n,r) = split-middle-r(s,p,p^m•r)
and p^n is replaced by p•p^m everywhere in Γ
split-middle-r(s,p^n,r) = split-middle-r(s•p^k,p,p^m•r)
and p^n is replaced by p^k•p•p^m everywhere in Γ

where m, k areew integer variables in the last three cases. ♦

Note that split-first and split-middle do not only provide a result, but also have a side-effect in Γ. In particular, the original inputted term may be modified. In the following, we sometimes apply these functions, but do not change the notation of the input term. This should cause no problems.

3.11 Lemma.

i) The side-effects of split-first and split-middle have the following properties:
   a) They are sound and complete and preserve the invariants.
   b) The measure μ is either decreased or left unchanged.
ii) The p-depths of the components of the result of the split functions are not greater than the p-depth of the input.
iii) Let σ be a unifier of Γ and let the function split-first or split-middle start with p, and let the result be (p_1, p_2). Assume that the side-effects transform (Γ, p) into (Γ', p').
Then there is a unifier σ' of Γ' as an extension of σ such that σp = σ'p' = (σ'p_1)•(σ'p_2).
iv) For every substitution σ and every splitting of σp = r_1•r_2, there is a possible execution of split-middle, such that for the results (p_1, p_2) of the function split-middle, there is an extension σ' of σ with σp = (σ'p_1)•(σ'p_2), such that σ'p_1 = r_1 and σ'p_2 = r_2.

Proof.

i.a) Soundness is obvious. For completeness, we have to inspect the choice points that modify Γ. There is one in split-first, which selects on whether σn = 1 or σn > 1. There are two possible choices for p•q and four such choices for p^n in split-middle. In every case it is obvious how to extend σ to the new integer variables. The choice is also complete, since it depends on whether σn is 1, 2, or greater than 2.

The invariants are preserved, since the introduction of new exponents in parametric terms is done with new integer variables.
i.b) Obvious.
ii) In order to show that the p-depth is not increased, an inspection of the single steps is sufficient. The only interesting case is that the replacement of p^n by $p^k pp^m$ doesn't increase the p-depth.
iii) The equality $\sigma p = \sigma' p' = \sigma' p_1(\sigma' p_2)$ can be verified by induction on the number of steps of the algorithm.
iv) Now let $\sigma p = r_1 \bullet r_2$ be an arbitrary splitting of σp, where σ is a substitution. Then we can control split-middle such that the occurrence of the split must be somewhere in the middle argument of split-middle. This terminates, since the size of the middle argument is strictly decreased in every step. ♦

The next definition gives non-deterministic decomposition rules for equations of the form $p(s) = f(t_1,\ldots,t_n)$ and $p(s) = q(t)$.

3.12 Definition. Decomposition rules for parametric terms.
Rule. (PT-decomposition)
This non-deterministic rule treats equations of the form $p(s) = f(t_1,\ldots,t_n)$.
Let $p(s) = f(t_1,\ldots,t_n)$ be such an equation in Γ. Then let (p_0, p_1) = split-first(p).
If p_0 does not start with function symbol f, then FAIL.
Otherwise, let $p_0 = f(s_1,\ldots,s_{h-1},\Omega,s_{h+1},\ldots,s_n)$.
Remove the equation $p(s) = f(t_1,\ldots,t_n)$ and add the equations $s_i = t_i$ for $i \neq h$, and $p_1(s) = t_h$.

Rule. (PP-decomposition)
This non-deterministic rule decomposes equations of the form $p(s) = q(t)$, where p and q are parametric terms.
Let $p(s) = q(t)$ be such an equation. Then select one of the following two alternatives.
i) Let (p_0, p_1) = split-middle(p).
If $p_0 = \Omega$ then FAIL.
Replace the equation by $p_1(s) = t$ and
perform Add-Para-Equation($q = p_0$) .
ii) Let (p_0, p_1) = split-middle(p) and (q_0, q_1) = split-middle(q).
(The side-effects of the second application should affect the results of the first application)
If $p_1 = \Omega$ or $q_1 = \Omega$, then FAIL.
perform Add-Para-Equation($p_0 = q_0$).
Then process the equation $p_1(s) = q_1(t)$:
Let (p_2, p_3) = split-first(p_1) and (q_2, q_3) = split-first(q_1).
If p_2 and q_2 have a different top function symbol, then FAIL.
Let $p_2 = f(s_1,\ldots,s_{h-1},\Omega,s_{h+1},\ldots,s_n)$ and $q_2 = f(t_1,\ldots,t_{k-1},\Omega,t_{k+1},\ldots,t_n)$.
If the arity of f is ≤ 1 or h = k, then FAIL.
Replace the inputted equation by the n equations:
$s_i = t_i$ for $i \notin \{h,k\}$, $p_3(s) = t_h$ and $q_3(t) = s_k$. ♦

Note that we exploit symmetry of the equality relation in the rules above.

3.13 Lemma. PT-decomposition is sound, complete, and either fails or strictly decreases μ or leaves μ unchanged and strictly reduces λ. Furthermore it leaves the invariants unchanged.
Proof. Soundness is obvious. For completeness, let σ be a unifier of Γ, such that $\sigma(p(s)) = \sigma(f(t_1,\ldots,t_n))$. Then split-first yields $p_0 = f(s_1,\ldots,s_{h-1},\Omega,s_{h+1},\ldots,s_n)$ and $\sigma(f(t_1,\ldots,t_n)) = \sigma(f(s_1,\ldots,s_{h-1},\Omega,s_{h+1},\ldots,s_n)(p_1(s)))$. Hence the new equations are satisfied. Furthermore the stronger condition for unifiers are also satisfied, since the instances of the basic parametric terms were not changed.
The measure μ is left invariant. The local measure λ is strictly decreased, since p-depth$(p_1(s)) \leq$ p-depth$(p(s))$ and the p-depths of all the terms t_i are strictly smaller than the p-depth of $f(t_1,\ldots,t_n)$.
The invariants are not changed, since parametric terms are not modified. ♦

3.14 Lemma. PP-decomposition is sound, complete, and either fails or strictly decreases μ or leaves μ invariant and strictly decreases λ. Furthermore it leaves the invariants unchanged.
Proof. Soundness is obvious.
For completeness, let σ be a unifier of Γ, such that $\sigma(p(s)) = \sigma(q(t))$.
Case: σq is a prefix of σp.
Then control split-middle such that split-middle(p) = (p_0, p_1), such that $\sigma' p_0 = \sigma q$ for a extension σ'. Then we can apply alternative i) of the rule and get a unifier of the resulting system.
Case: neither σp is a prefix of σq nor σq a prefix of σp.
Then we apply alternative ii). Split-middle applied to p and q can be guided, such that the results p_0, p_1, q_0, q_1 are such that with a sensible extensions σ', we have $\sigma' p_0 = \sigma' q_0$, and σp_0 is the greatest common prefix of σp and σq. Furthermore σp_1 and σq_1 are nontrivial, since otherwise, we are in first case. Then we can apply split-first to p_1 and q_1 resulting in p_2, p_3, q_2, q_3. Since we have assumed unifiability, the top level function symbols of p_2 and q_2 must be the same. Let $p_2 = f(s_1,\ldots,s_{h-1},\Omega,s_{h+1},\ldots,s_n)$ and $q_2 = f(t_1,\ldots,t_{k-1},\Omega,t_{k+1},\ldots,t_n)$. Since $\sigma' p_0$ is the greatest common prefix of σp and σq, the arity of f must be greater than 1 and $h \neq k$. Now we can decompose and add the equations. The unifier σ' is also a unifier of the resulting system, also satisfying the implicit assumptions.
Measure μ is left invariant. We have to show that the local measure λ is strictly decreased:
In alternative i) p-depth$(p_1(s)) \leq$ p-depth$(p(s))$ and p-depth$(t) <$ p-depth$(q(t))$. We have also to take into account the effects by Add-Para-Equation. Since Add-Para-Equation modifes Γ_P only inn the case that the equation is between non-basic parametric terms, we have that the p-depths of the terms in TBPT$(q = p_0)$ are strictly smaller than the maximum of the p-depths of p(s) or q(t). If the decomposition in Add-Para-Equation adds new equations to Γ_T, then there may be a basic parametric term r from TBPT(Γ_P) that has a greater p-depth than q and p_0. But then the parametric term r will be replaced by some smaller basic parametric term, and the new equations are also smaller than r, hence the measure λ is decreased.

In alternative ii), the p-depths of the t_i and s_i are all smaller than the maximum of the p-depths of p(s) and q(t). Furthermore p-depth(p_3(s)) ≤ p-depth(p(s)), p-depth(t_h) < p-depth(q(t)), p-depth(q_3(t)) ≤ p-depth(q(t)), p-depth(s_k) < p-depth(p(s)).
The argumentation for Add-Para-Equation is the same as for alternative i) Hence λ is strictly smaller in the doubly nested multiset-ordering.
The invariants are not changed, since parametric terms are neither moved nor modified. ♦

3.15 Theorem. Applying the rules in this section to a system Γ until no rule is applicable either yields a FAIL, or it terminates and yields a decomposed normal form.
Proof. The rules in this section can be applied whenever there is an equation s = t, and neither s nor t have a second-order variable as the top function symbol. The rules either strictly decrease μ, or if they leave it invariant they strictly decrease λ. Since (μ, λ) is well-founded, this process must terminate, and in case of termination, the system is in decomposed normal form, since no rule of this section is applicable. ♦

4. Eliminating SO-Cycles

In this section we concentrate on eliminating SO-cycles, and we show that the existence of some SO-cycle permits to remove at least one SO-variable.

4.1 Definition. i) Let t be a term and let X be a second-order variable that has an occurrence in t. The second order variable X **uniquely occurs in** t, if there is only one occurrence, and furthermore, the SO-variable X doesn't occur in parametric term of the form p^n.
ii) An SO-cycle $X_1(s_1) = t_1$, $X_2(s_2) = t_2$, … , $X_n(s_n) = t_n$ is called **unique**, if all the occurrences of the X_i in t_{i-1}, i = 2,…,n and X_1 in t_n are unique.
iii) Let t be a term and let X be a second-order variable that has a non-unique occurrence in t. The greatest common prefix of all the positions of non-unique occurrences of X in t is called the non-uniqueness index of X in t. ♦

Note that a unique occurrence can be of the form $p^n(X(x))$, but not of the form $(f(X(a), \Omega)^n$.

4.2 Definition. In order to show termination of the rules described in this section, we will use as a local measure λ.in this section as the lexicographic combination of the following components:

λ_1: the minimal length of an SO-cycle in Γ_T.

λ_2: the minimal p-depth of the non-trivial non-uniqueness index of the of all non-unique occurrences in a SO-cycle of minimal length.
If no non-unique occurrence is in the minimal SO-cycle, then -1.

λ_3: λ_1 – d where d is the length of the longest chain in a minimal SO-cycle of the form $X_1(s_1) = X_2(t_1)$, …, $X_k(s_k) = X_{k+1}(t_k)$.

λ_4: minimal p-depth of X_{k+2} (or X_1, respectively) in t_{k+1} in a longest chain of the form $X_1(s_1) = X_2(t_1)$, …, $X_{k-1}(s_{k-1}) = X_k(t_{k-1})$, $X_k(s_k) = t_k$ in a minimal SO-cycle.

λ_5: minimal c-depth of the same variable as in the definiition of λ_4.

4.3 Definition. The following non-deterministic subules operate on SO-cycles. We do not give any control, since the control depends on the purpose.

Subrule (Exponent-GT-1)

Let p^n be a parametric term somewhere in Γ. Then select one of the following two possibilities.

i) remove n from Γ.

ii) replace p^n by $p \bullet p^m$ everwhere in Γ, where m is a new positive integer variable. ♦

Subrule (Split-SO-Variable-Xf)

If there is an equation $X(s) = f(t_1,\ldots,t_n)$ in Γ, then select one of the two alternatives:

i) remove X from Γ.

ii) Select some i with $1 \le i \le n$.
If X occurs in some t_j for $j \neq i$, then FAIL.
Otherwise, let X' be a new second-order variable, and replace X everywhere according to $X(\Omega) := f(t_1,\ldots,t_{i-1}, X'(\Omega), t_{i+1},\ldots,t_n)$. Let s' be s and t_i' be t_i after this replacement.
Now remove the orginial equation and add $X'(s') = t_i'$

Rule (Split-SO-Variable-Xp)

The rule operates on an equation $X(s) = p(t)$ in Γ, given a position π.

Select one of the two alternatives:

i) Remove X from Γ. EXIT.

ii) Let (p_1, p_2) = split-middle(p).

If X occurs in p_1, then FAIL.

If the position of Ω in p_1 is not a prefix of π, then FAIL.

If $p_2 = \Omega$, then replace X everywhere using $X(\Omega) = p(X'(\Omega))$, where X' is a new SO-variable and replace the input equation by $X'(s') = t'$.

Otherwise, replace X everywhere using $X(\Omega) = p_1(X'(\Omega))$, where X' is a new SO-variable.
We mark the new terms after replacement with a '.
Select one of the two alternatives:

i) remove X' from Γ. EXIT.

ii) Let (p_3, p_4) = split-first(p_2'), and let $p_3 = f(t_1,\ldots,t_{i-1},\Omega,t_{i+1},\ldots t_n)$.
If X' occurs in $p_4(t')$ or arity(f) ≤ 1, then FAIL.
Now select some index $j \neq i$ with $1 \le j \le n$.
If X' occurs in some t_h with $h \neq j$, then FAIL.
Replace X' everywhere using $X'(\Omega) = f(t_1,\ldots,t_{j-1},X''(\Omega),t_{j+1},\ldots,t_{i-1}, p_4(t'), t_{i+1},\ldots,t_n)$, where X'' is a new SO-variable.
Replace the input equation by $X''(s') = t_j$.

We may refer to the last two rules also under the name Split-SO-Variable. ♦

4.4 Lemma. Consider the rule Split-SO-Variable-Xf.

i) It is a sound and complete rule.

ii) It leaves the invariants unchanged.

Proof. Soundness is trivial. For completeness, let σ be a unifier of Γ. Then $\sigma X(s) = \sigma f(t_1,\ldots,t_n)$. If $\sigma X = \Omega$, then select the first alternative. Let σX be a nontrivial parametric term. Then $\sigma X = f(\sigma t_1, \ldots, \sigma t_{i-1}, p, \sigma t_{i+1},\ldots,\sigma t_n)$ where p is a ground parametric term. Decomposition gives $p(\sigma s) = \sigma t_i$. If X would occur in some t_j for $j \neq i$, then we would get a contradiction, since σt_j would be properly contained in σt_j. Hence X is not contained in t_j for $j \neq i$. We can extend σ such that $\sigma(X') = p$ ♦

4.5 Lemma. Consider the rule Split-SO-Variable-Xp.

i) It is a sound and complete rule.

ii) It leaves the invariants unchanged.

Proof. Soundness is trivial.

For completeness let σ be a unifier of Γ before the application. If $\sigma X = \Omega$, then select alternative i).

Assume $\sigma X \neq \Omega$. Then split σX into $q_1 \bullet q_2$, such that the position of Ω in q_1 is a prefix of π, and q_1 is maximal w.r.t. this condition. Then we can apply split-middle to p, such that either $\sigma p_1 = q_1$ or σp_1 is a prefix of q_1, and σp_2 and q_2 do not have a common prefix.

The SO-variable X cannot occur in p_1, since then we get a incompatible size of instances.

If $p_2 = \Omega$, then we can replace $X(\Omega)$ by $p(X'(\Omega))$, and the equation $\sigma'(X'(s')) = \sigma't'$ holds for a unifier σ' that extends σ.

In the case $p_2 \neq \Omega$, we can now focus on the equation $X'(s) = p_2(t)$, and we have $\sigma X'(s) = \sigma p_2(t)$. Either we have $\sigma X' = \Omega$, or the occurrences of Ω in the first component of σp_2 and $\sigma X'$ are different. This means that the top function symbol must have arity greater than 1. Now we can use split-first on p_2 to split it into the first and and the remaining components. This leads to the replacements and decompositions as described in the rule.

That X' cannot occur in some t_h with $h \neq j$ can be seen by standard arguments.

ii) The invariants are unchanged, since removing SO-variables or replacing X using $X(\Omega) = p(X'(\Omega))$ consistently changes the second-order prefixes. ♦

4.6 Lemma. The Rule Exponent-GT-1 is a sound and complete rule. Furthermore. it preserves the invariants.

Proof. Obvious, since substitutions should instantiate n with a positive integer and this may be either equal to 1 or greater than 1. ♦

4.7 Definition.

Rule. Make-cycles-unique.

Let $X_1(s_1) = t_1,\ldots, X_n(s_n) = t_n$ be a non-unique SO-cycle with minimal λ.

Assume that $X_1(s_1) = t_1$ is the equation with a minimal non-unique occurrence of X_2 in t_1.

If there is only one occurrence of X_2 in t_1,

then let p^n be some subterm of t_1 such that X_2 occurs in p.

Apply Exponent-GT-1 to p^n.

If n has been removed, then EXIT.

Now there are at least two occurrences of X_2 in t_1.

Let π be the greatest common prefix of the positions of X_2 in t_1.

If π is not empty,

then apply Split-SO-Variable to the equation using the position π. EXIT.

If π is empty and the SO-cycle has length 1, then remove X_1 from Γ.

If π is empty and the SO-cycle has length > 1, then let $t_1 = f(t_{1,1},\ldots,t_{1,n})$,

perhaps after using split-first.

Apply Split-SO-Variable-Xf to the equation. ♦

4.8 Lemma. The rule "Make-cycles-unique" is sound and complete, preserves the invariants and either fails or strictly reduces (μ, λ).

Proof. Soundness is trivial.

Completeness: Let $X_1(s_1) = t_1,\ldots , X_n(s_n) = t_n$ be the minimal SO-cycle, where $X_1(s_1) = t_1$ is the equation with a non-unique occurrence of X_2 in t_1, and some of the occurrences of X_2 in t_1 is the w.r.t. p-depth minimal occurrence in the SO-cycle.

Completeness can be derived from the completeness of Exponent-GT-1, Split-SO-Variable and split-first.

There only exception is the case that the non-uniqueness index is 0 and the SO-cycle is of length 1. In this case a unifier of this equation must remove X_1, since otherwise, we get an occur-check failure.

Invariants: Are not modified, since the applied rules do not modify the invariants.

Measure: Let $X_1(s_1) = t_1,\ldots , X_n(s_n) = t_n$ be the minimal SO-cycle, where $X_1(s_1) = t_1$ is the equation with a non-unique occurrence of X_2 in t_1, and some of the occurrences of X_2 in t_1 is the w.r.t. p-depth minimal occurrence in the SO-cycle. Assume that σ is a unifier of Γ.

The stratifiedness condition enforces that t_1 is either of the form $f(t_{1,1},\ldots,t_{1,n})$, or $p(t')$.

There are two cases, either there are two occurrences of X_2 in t_1, or X_2 occurs in a power q^m.

If X_2 occurs in a power, then the application of “Exponent-GT-1” either removes an exponent, and the measure μ_3 is reduced, or it generates two occurrences of X_2 in t_1. The non-uniqueness index is not increased.

Now we can assume that there are two occurrences.

If π is empty and the SO-cycle has length 1, then μ is strictly decreased.

In the following π is empty or not.

Now we make an application of Split-SO-variable to $X_1(s_1) = t_1$.

Split-SO-Variable-Xf: If X has been removed from Γ, then μ is strictly decreased.

Otherwise, there are two cases: either i is a prefix of π or not.

If not, then X_1 will be replaced using $X_1(\Omega) = f(t_{1,1},\ldots,t_{1,i-1}, X_1'(\Omega), t_{1,i+1},\ldots,t_{1,n})$, and X_2 is contained in some $t_{1,j}$. This will shorten the SO-cycle, since the replacement introduces an occurrence of X_2 in t_n.

If i is a prefix of π, then using $X_1(\Omega) = f(t_{1,1},\ldots,t_{1,\ i-1},X_1'(\Omega), t_{1,i+1},\ldots,t_{1,n})$ and decomposing the equation leaves the SO-cycle with its length, but makes the non-uniqueness index of the first equation in the SO-cycle smaller.

Split-SO-Variable-Xp: Consider the case where there is no FAIL and no reduction of μ.

In the case $p_2 = \Omega$, the length of the SO-cycle remains invariant but the non-uniqueness index has been strictly decreased. After the decomposition of p_1, the other case is basically the same as applying Split-SO-Variable-Xf in the non-prefix case. The length of the SO-cycle will then be strictly decreased. ♦

4.9 Definition. The following function is a deterministic construction of a parametric term from a term t and a position π within t, where the second-order prefix of π must be empty and π is not wihin p^n for some paramtric term p. Hence we can assume that the considered parametric terms are basic.

uni-para(t, ε) := Ω

uni-para($f(t_1,\ldots,t_n)$, $i\bullet\pi$) := $f(t_1,\ldots,t_{i-1}, \Omega, t_{i+1},\ldots,t_n)\bullet$uni-para($t_i,\pi$)

uni-para($f(t_1,\ldots,t_{i-1}, \Omega, t_{i+1},\ldots,t_n)(s)$, $i\bullet\pi$) := $f(t_1,\ldots,t_{i-1},\Omega,t_{i+1},\ldots,t_n)\bullet$(uni-para(s),$\pi$)

uni-para($f(t_1,\ldots,t_{i-1}, \Omega, t_{i+1},\ldots,t_m)(s)$, $j\bullet\pi$) :=

$f(t_1,\ldots,t_{j-1},\Omega,t_{j+1},\ldots,t_{i-1}, s, t_{i+1},\ldots,t_n)\bullet$(uni-para($t_j$),$\pi$) if $j \neq i$.

4.10 Definition.

Rule "Break-1-cycles". This rule is to be applied to a unique SO-cycle of length 1.

Let X(s) = t be an equation, such that X uniquely occurs in t.

Then select a number $k \leq$ p-depth of X in t and apply the appropriate rule of Split-SO-Variable k times using the following specializations.

If X(s) = p(t) is the equation, and X occurs in p, where $p = p_1\bullet\ldots\bullet p_n$, then the first component p_1 of p is a basic parametric term. Apply Split-SO-Variable-Xf to $p_1(p_2\bullet\ldots\bullet p_n(t))$.

If X has been removed from Γ then EXIT .

Let us denote the cyclic equation again as X(s) = t.

Now select one of the following alternatives:

i) remove X from Ω.

ii) Let p = uni-para(t,π), where π is the position of X in t.
Perform split-middle(p) and let q be the first element of the result,
If q is a power q'^m, then FAIL.
Perform Add-Para-Equation ($q\bullet p = p\bullet q$).
Replace the equation X(s) = t with s = t', where X is removed from t' and replace X by $(q)^n$ everywhere, where n is a new positive natural integer

4.11 Lemma. The rule Break-1-Cycle is sound and complete and preserves the invariants of Γ. Furthermore, either the rule fails or the measure μ is strictly decreased.

Proof. *Soundness*: Is trivial, using the argument that addition of restrictions and equations is sound.

Completeness: Let σ be a unifier of Γ before application of Break-1-Cycle. Lemma 4.4 and Lemma 4.5 show that an application of Split-SO-Variable is complete. Since $\sigma X(s) = \sigma(t)$, and X uniquely occurs in t, σX can be represented as $u_1^k\bullet u_2$, where u_1 and u_2 are strings of basic parametric ground terms, $|u_1|$ is equal to the p-depth of X in t (after application of σ), $|u_2|$ is equal to or smaller than $|u_1|$ and u_2 is a prefix of u_1. It is easy to see that this can be rewritten to $u_3u_4^k$, where $|u_4| = |u_1|$ and $|u_2| = |u_3|$, and that this can be rewritten as $u_3u_5^h$, where u_5 is a non-power. Then we apply the rule Split-SO-Variable a number of times, where this number is not greater than the p-depth of X in t. We do this until u_3 has been worked off. The failure cases in Split-SO-Variable show that the split must directly follow the position of X. After these applications we can assume that $\sigma X = u_4^k$. If k = 0, then we apply alternative i) of Break-1-

cycle and remove X from Γ, otherwise $k \geq 1$. For p = uni-para(t,π), we have that $(\sigma p)^k = u_4^k = u_5^h$. Split-middle(p) can be guided such that for the first result, q, we have $\sigma q = u_5$. Obviously, we have $u_4u_5 = u_5u_4$, hence $\sigma(p \bullet q) = \sigma(q \bullet p)$. Furthermore, σq is a non-power. The parametric term q cannot be a syntactic power, since split-middle is able to split terms of the form r^m.
Addition of $q \bullet p = p \bullet q$ is now justifed. The replacement of X(s) = t by s = t' is correct for the chosen unifier σ, since σX is a power u_4^k, and t was rewritten as p(X(t")), and σp and σX commute, hence σX can be cancelled. Of course, the replacement of X by q^n is also justified.
Invariants: The rule preserves the invariants, i.e. stratifiedness is preserved, since parametric terms cannot occur under a second-order variable and thus identification doesn't move variables out of the scope of a second-order variable. The introduction of parametric terms is also designed such that they are not put under the scope of a second order variable.
Furthermore, it is excluded that a parametric term of the form $(p^n)^m$ is generated.
Measure: The applications of Split-SO-Variable leave a unique SO-cycle of length 1, all the exits of the rules remove one SO-variable, hence μ_1 has been strictly decreased. ♦

The aim of rule Flat-Partial-Cycle is to decrease the measure λ_3 by considering a minimal SO-cycle with a longest chain of the form $X_1(s_1) = X_2(t_1)$, …, $X_{k-1}(s_{k-1}) = X_k(t_{k-1})$, $X_k(s_k) = t_k$, and operating on the last equation using a variant of Split-SO-Variable.

The remaining case is that a minimal SO-cycle is unique and has length at least 2.

4.12 Definition.

Rule "Flat-Partial-Cycle".

Assume there is a minimal SO-cycle of the following form:
$X_1(s_1) = X_2(t_1)$, …, $X_{k-1}(s_{k-1}) = X_k(t_{k-1})$, $X_k(s_k) = t_k$,… , $X_n(s_n) = t_n$
and X_{k+1} occurs in t_k not at top level, and the SO-cycle has length ≥ 2 and $k \neq n$.
Then select one of the following three alternatives:

i) Select some i with $1 \leq i \leq k$ and remove X_i from Γ.

ii) If $t_k = f(r_1,\ldots,r_m)$.
Let j be the index, such that X_{k+1} occurs in r_j.
Select on of the following alternatives:
   i) For all a = 1,…,k replace X_a using $X_a(\Omega) = f(r_1,\ldots r_{j-1},X_a'(\Omega), r_{j+1},\ldots,r_m)$ and decompose the k equations. The X_a' are new SO-variables.
   ii) If arity(f) ≤ 1, then FAIL.
   Select an index g with $1 \leq g \leq k$, and replace X_a using $X_a(\Omega) = f(r_1,\ldots,r_{j-1}, X_a'(\Omega), r_{j+1},\ldots,r_m)$ for a = g+1,…, k, where X_a' are new SO-variables. Select some index $h \neq j$ and replace X_g everywhere using $X_g(\Omega) = f(r_1,\ldots,r_{j-1},X_{g+1}'(t_g), r_{j+1},\ldots,r_{h-1}, X_g'(\Omega), r_{h+1},\ldots,r_m)$.
   The we use decomposition for the equations of the form f(…) = f(…). This gives the equations $X_i'(s_i) = X_{i+1}'(t_i)$ for i = g+1,…,k-1. Furthermore, for example the equation $X_g'(s_g) = r_h$ is added.

iii) If $t_k = p(t_k')$ and X_{k+1} occurs in t_k'.
Let (p_1, p_2) be the result of split-middle(p).
If $p_1 = \Omega$, then FAIL.

Replace X_a using $X_a(\Omega) = p_1(X_a'(\Omega))$ for all a = 1,…,k, and decompose the equations such that the SO-cycle looks as follows:

$X_1'(s_1) = X_2'(t_1)$, …, $X_{k-1}'(s_{k-1}) = X_k'(t_{k-1})$, $X_k'(s_k) = p_2(t_k')$,… ,$X_n(s_n) = t_n'$.

Either select some i with $1 \le i \le k$ and replace $X_i'(\Omega)$ by Ω or do nothing.

If $p_2 = \Omega$, then EXIT.

If $p_2 \neq \Omega$, then replace p_2 by $p_3 \bullet p_4$, where (p_3, p_4) = split-first(p_2)

Let $p_3 = f(r_1,\ldots,r_{j-1},\Omega, r_{j+1},\ldots,r_m)$.

If arity(f) ≤ 1, then FAIL.

Select an index g with $1 \le g \le k$, and replace X_a' using $X_a'(\Omega) = f(r_1,\ldots,r_{j-1}, X_a''(\Omega), r_{j+1},\ldots,r_m)$ for a = g+1,…, k, select some index $h \neq j$ and replace X_g' everywhere using $X_g'(\Omega) = f(r_1,\ldots,r_{j-1},X_{g+1}''(t_g), r_{j+1},\ldots,r_{h-1}, X_g''(\Omega), r_{h+1},\ldots,r_m)$.

Then we use decomposition for the equations of the form f(…) = f(…). This gives the equations $X_i''(s_i) = X_{i+1}''(t_i)$ for i = g+1,…,k-1. Furthermore, for example the equations $X_g''(s_g) = r_h$ is added.

iv) If $t_k = p(t_k')$ and X_{k+1} occurs in p.

Let $p = q_1 \bullet \ldots \bullet q_n$. If X_{k+1} does not occur in q_1, then use $q_1(q_2 \bullet \ldots \bullet q_n(t_k'))$ instead of $p(t_k')$ and then perform part iii). If X_{k+1} occurs in q_1, then q_1 is not a power, and we can perform case ii) ♦

4.13 Lemma. The rule Flat-Partial-Cycle is sound and complete. The invariants remain unchanged. Furthermore an application either fails or strictly reduces the measure (μ, λ).

Proof. *Soundness* is obvious.

Completeness: let σ be a unifier of Γ. If there is some i with $1 \le i \le k$ and $\sigma X_i = \Omega$, then choose alternative i). We can assume now that $\sigma X_i \neq \Omega$ for all i = 1,…,k.

Case. t_k is of the form $f(r_1,\ldots,r_n)$. Then σX_i is a term starting with f for all $1 \le i \le k$.

If for all $1 \le i \le k$, j is a prefix of the position of Ω in $\sigma X_i(\Omega)$, then we can extend σ, such that $\sigma(X_i(\Omega)) = \sigma(f(r_1,\ldots r_{j-1}, X_i'(\Omega), r_{j+1},\ldots,r_m))$. Hence we have case ii.i) of the rule and can replace the equations $X_i(s_i) = X_{i+1}(t_i)$ by $X_i'(s_i) = X_{i+1}'(t_i)$ for i = 1,…,k-1 and the equation $X_k(s_k) = f(r_1,\ldots,r_m)$ by $X_k'(s_k) = r_j$. The form of the SO-cycle is the same for the first k elements, but the depths of the X_{k+1} in the top term in the k^{th} equation has been reduced.

In the other case, there is a maximal index g, such that for all a with $g < a \le k$, Ω occurs beneath index j in $\sigma X_a(\Omega)$, but beneath a different position prefix h in $\sigma X_g(\Omega)$. Hence f must have arity at least 2. We have $\sigma(X_a(\Omega)) = \sigma(f(r_1,\ldots,r_{j-1}, X_a'(\Omega), r_{j+1},\ldots,r_m))$ and $\sigma(X_g(\Omega)) = \sigma(f(r_1,\ldots,r_{h-1}, X_g'(\Omega), r_{h+1},\ldots,r_{j-1}, X_{g+1}'(t_g), r_{j+1},\ldots,r_m))$, which can be seen by considering the g^{th} equation: $\sigma X_g(s_g) = \sigma X_{g+1}(t_g)$ gives $\sigma(f(r_1,\ldots r_{h-1}, X_g'(s_g), r_{h+1},\ldots,r_{j-1}, X_{g+1}'(t_g), r_{j+1},\ldots,r_m)) = \sigma(f(r_1,\ldots,r_{j-1},X_{g+1}'(t_g), r_{j+1},\ldots,r_m))$. Replacement of $X_g(\Omega)$ by $f(r_1,\ldots,r_{h-1},X_g'(\Omega), r_{h+1},\ldots,r_{j-1}, X_{g+1}'(t_g), r_{j+1},\ldots, r_m)$ shortens the SO-cycle, since the equation with index g is no longer necessary.

Case t_k is of the form $p(t_k')$.

If X_{k+1} does not occur in t_k', then the syntactic transformations permit to use the other cases.

Hence we can assume that X_{k+1} occurs in t_k'.

Let π be the maximal position, such that π is a prefix of all the positions of Ω in σX_i for i = 1,…,k, and such that π is also a prefix of the position of σX_{k+1} in $\sigma(p(t_k'))$. We can control

split-middle(p), and extend σ, such the position of Ω in σp_1 is π. The replacements of $X_a(\Omega)$ by $p_1(X_a'(\Omega))$ for a = 1,…,k now produces the same situation, but the position π can be assumed to be ε. Again, we need a guessing step, if some $\sigma X_a' = \Omega$ for $1 \le a \le k$.
Now we assume that $\sigma X_a' \neq \Omega$ for $1 \le a \le k$.
Assume also $p_2 \neq \Omega$. Then we look from σX_k down to σX_1 for the first index, such that the first symbol in the position of Ω in σX_i is not a prefix of the position of X_{k+1}' in t_k'. Let this position be g. Then we can split p_2 using split-first in p_3 and p_4. The top function symbol f of p_3 must have arity greater than 1, since otherwise there is only one possible starting symbol for positions. Let us inspect the equation at the index g. With $p_3 = f(r_1,\ldots,r_{j-1},\Omega, r_{j+1},\ldots,r_m)$, where j is the first symbol of the Ω in σp, and $X_g' = X_g'(\Omega) = f(r_1,\ldots,r_{j-1},x_j, r_{j+1},\ldots,r_{h-1}, X_g''(\Omega), r_{h+1},\ldots,r_m)$, the equation is: $\sigma f(r_1,\ldots,r_{j-1},x_j, r_{j+1},\ldots,r_{h-1}, X_g''(\Omega), r_{h+1},\ldots,r_m)(s_g) = \sigma f(r_1,\ldots,r_{j-1},X_{g+1}''(\Omega), r_{j+1},\ldots,r_m)(t_g)$. This shows $\sigma x_j = \sigma(X_{g+1}''(t_g))$.Furthermore the equation $\sigma X_g''(s_g) = \sigma r_h$ holds.
Invariants: There are no new parametric terms. Furthermore, the replacements of SO-variables are such that the stratifiedness conditions remains valid.
Measure: Inspecting the algorithm, there are several places, where μ is strictly reduced. In case ii.i), the measure λ_4 has been strictly decreased. In ii.ii), the minimal length of an SO-cycle is strictly decreased, since a new SO-cycle is generated, where the equation with index g is no longer necessary, and since we have applied the rule to an SO-cycle of length at least 2 and this length was minimal. In case iii), if $p_2 = \Omega$, either λ_3 has been strictly decreased or all λ_i are invariant, and λ_4 is strictly decreased. In the last part of iii), the minimal length of a SO-cycle is strictly decreased.
In case iv), either the maesure $(\mu, \lambda_1,\ldots,\lambda_4)$ is strictly decreased. If this remains unchanged, then the c-depth is strictly decreased, hence λ_5 was strictly decreased. ♦

We describe a non-deterministic algorithm to reduce the measure (μ,λ) of Γ, if the shortest SO-cycle is unique and has length ≥ 2.

4.14 Definition. "Long-Basic-SO-Cycle".
This rule is to be applied to an SO-cycle of minimal length, where this length should be greater than 1. Furthermore the SO-cycle should be of the form $X_1(s_1) = X_2(t_1)$, $X_2(s_2) = X_3(t_2)$,… , $X_n(s_n) = t_n$, and t_n has an occurrence of X_1 not at top level.
Then:

1) Replace all X_i for i=1,…,n using $X_i(\Omega) = Z(X_i'(\Omega))$, where Z is a new SO-variable.
   Eliminate the unique SO-cycle for Z in the last equation using the rule Break-1-Cycle.
   In the Split-SO-Variable-steps in addition the equations with index i=1,…,n-1 are always decomposed, such that $X_i^{(h)}(s_i) = X_{i+1}^{(h)}(t_i)$ is replaced by $X_i^{(h+1)}(s_i) = X_{i+1}^{(h+1)}(t_i)$
   After such a step, the SO-cycle has the form
   $X_1^{(h)}(s_1)) = X_2^{(h)}(t_1)$, $X_2^{(h)}(s_2) = X_3^{(h)}(t_2)$,… , $X_n^{(h)}(s_n) = t_n^{(h)}$, and $t_n^{(h)}$ has an occurrence of $X_1^{(h)}$ not at top level.
   In the final step, the q^m-replacement for Z is also decomposed away in the SO-cycle.
2) The SO-cycle is now again of the form

$Y_1(s_1) = Y_2(t_1)$, $Y_2(s_2) = Y_3(t_2)$,… , $Y_n(s_n) = t_n'$, and t_n' has an occurrence of Y_1 not at the top level.

3) Either select some i and remove Y_i from Γ and EXIT,
   or do nothing.
4) Transform t_n' into the form $f(r_1,\ldots,r_m)$, using split-first if necessary.
   Let j be the index, such that r_j has an occurrence of Y_1.
   If the arity of f is ≤ 1, then FAIL.
5) Select some index k with $1 \le k \le n$.
   i) If $k < n$, then:
      Replace Y_a using $Y_a(\Omega) = f(r_1,\ldots,r_{j-1}, Y_a'(\Omega), r_{j+1},\ldots,r_m)$ for a = k+1,…, n.
      Select some index $h \neq j$ and replace Y_k everywhere using $Y_k(\Omega) = f(r_1,\ldots,r_{j-1},Y_{k+1}'(t_k), r_{j+1},\ldots,r_{h-1}, Y_k'(\Omega), r_{h+1},\ldots,r_m)$.
      The we use decomposition for the equations of the form f(…) = f(…). This gives the equations $Y_i'(s_i) = Y_{i+1}'(t_i)$ for i = k+1,…,n-1 and $Y_n'(s_n) = r_j$. Furthermore, for example the equations $Y_k'(s_k) = r_h$ is added. Now the SO-cycle is reflected in the sequence Y_1, … Y_{k-1}, Y_{k+1}', … Y_n'.
   ii) If k = n, then select some index $h \neq j$ and replace Y_n using $Y_n(\Omega) = f(r_1,\ldots,r_{h-1},Y_n'(\Omega), r_{h+1},\ldots,r_m)$.
      The SO-cycle is then reflected in the sequence $X_1,\ldots,X_{n-1}$. ♦

4.15 Lemma. Long-Basic-SO-Cycle used as a non-deterministic rule is sound and complete . Furthermore, it either fails or the measure (μ,λ) is strictly reduced.
The invariants remain unchanged.

Proof. *Soundness* is obvious.

Completeness. Let σ be a unifier of Γ before step 1. The only interesting case is that $\sigma X_i \neq \Omega$ for all i. We choose σZ, such that σZ is the greatest common prefix of all the σX_i, i=1,…,n. Syntactically, the rule Break-1-Cycle show that elimination of X(s) = t, where X occurs in t, produces an equation s = t', where t' is t with X removed. Hence, the form of the SO-cycle is as claimed.

Now we can assume that we have $\sigma Y_i \neq \Omega$ for all i = 1,…,n and that the first symbol of the position of Ω in the σY_i is not always the same. The term t_n' can be brought into the form $f(r_1,\ldots,r_m)$. The arity of f must be greater than 1.

Now let j be the index, such that Y_1 occurs in r_j. The choice of σZ shows that there is some index k, such that the Ω in σY_k doesn't occur in the j^{th} position. We choose k to be maximal.

If k = n, then $\sigma Y_n(\Omega) = \sigma f(r_1,\ldots,r_{h-1},Y_n'(\Omega), r_{h+1},\ldots,r_m)$ for some $h \neq j$. The replacement as in case 4.ii) can be performed.

Let k < n. Then $\sigma Y_a(\Omega) = \sigma f(r_1,\ldots,r_{j-1},Y_a'(\Omega), r_{j+1}, \ldots, r_m)$ for a = k+1,…,n for some new SO-variables Y_a and an extended σ. For a = k we have $\sigma Y_k(\Omega) = \sigma f(r_1,\ldots,r_{j-1},x_j,r_{j+1},\ldots,r_{h-1},Y_k'(\Omega), r_{h+1}, \ldots, r_m)$ for some $h \neq j$, a new FO-variable x_j and an extension of σ. The special form of the equation shows that $\sigma Y_k(s_k) = \sigma f(r_1,\ldots,r_{j-1}, x_j, r_{j+1},\ldots,r_{h-1}, Y_k'(s_k), r_{h+1}, \ldots, r_m) = \sigma f(r_1,\ldots,r_{j-1},Y_{k+1}'(t_k), r_{j+1},\ldots,r_m) = \sigma Y_{k+1}(t_k)$. Decomposition gives $\sigma Y_{k+1}'(t_k) = \sigma x_j$ and $\sigma Y_k'(s_k) = \sigma r_h$. Hence we can also use $Y_k(\Omega) = f(r_1,\ldots,r_{j-1},Y_{k+1}'(t_k),r_{j+1},\ldots,r_{h-1},Y_k'(\Omega), r_{h+1}, \ldots, r_m)$.

We have shown that the rule is complete.

Note that the generated SO-cycle is always a proper SO-cycle, i.e., there is some term with a proper occurrence of a participating second-order variable.
Measure: It is important to note that the rule is only applied to a minimal SO-cycle. Minimality implies that the SO-variables X_i do not occur elswhere in the SO-cycle, except at the mentioned positions. The addition of the SO-variable increases the measure μ_1 by 1, however the application of Break-1-Cycle decreases μ_1 by 1 , such that the effect is neutralized. Furthermore this rule does not increase the measures μ_2 and μ_3.
The measure (μ,λ) of Γ is reduced, since either some SO-variable is removed, or the SO-cycle is shortened in step 4).
The invariants remain unchanged, since the replacements of an SO-variable are made everywhere, and since the replacements occur in "the same layer". ♦

4.16 Theorem. If there is a second-order cycle, then we can transform the system such that either a FAIL occurs or the measure μ is strictly decreased.
Proof.We can assume for the purpose of this proof that no FAIL occurs. We select an SO-cycle with minimal length. If this SO-cycle is not unique, then Lemma 4.8 shows that we can transform Γ, such that (μ, λ) is strictly reduced. Finally we will either reduce μ or get a system with a minimal SO-cycle that is unique. If this SO-cycle has length ≥ 2, then we can use Flat-Partial-Cycle or Long-Basic-SO-Cycle to reduce (μ,λ). Since the measure is well-founded, after finitely many steps either μ will be strictly decreased or we will get a unique SO-cycle of length 1. Then Lemma 4.11 shows that at least one SO-variable can be removed. ♦

5 Cycle-free Stratified SO-systems.

In this section we assume that the system is in decomposed normal form and that there is no SO-cycle. In this case, we show that we can remove at least one SO-variable from Γ_T, (perhaps not from Γ).

This transitive relation $\geq_{SO}$ can be generated as follows:
i) If there is an equation $X(s) = Y(t)$, then add $X \geq_{SO} Y$ and $Y \geq_{SO} X$.
ii) If there is an equation $X(s) = t$, and the top level function symbol of t is not a second order variable, and Y occurs in t, then add $X >_{SO} Y$

Since there are no SO-cycles, the generated quasi-ordering is consistent with the generating relations. We denote the corresponding equivalence relation by $\approx_{SO}$.
An **SOV-cluster** C is a $>_{SO}$-maximal $\approx_{SO}$-equivalence class of second-order variables. Note that second-order variables in some SOV-cluster C appear only at the top-level of terms, since we assume that all terms are in decomposed normal form. Every SOV-cluster C is correlated with a subset E(C) of the equations Γ_T, namely all equations that contain an occurrence of some SO-variable from C. The syntactic form of equations in this set may be $X(s) = Y(t)$, $X(s) = f(t_1,\ldots,t_n)$, and $X(s) = p(t)$.
We need a local ordering for this section that shows that the rules terminate.
Let C be a SOV-cluster, then we measure C by
i) λ_1: the number of SO-variables that occur in C.

ii) λ_2: the multiset of all p-depths of toplevel terms in C which are not of the form X(s).
We will combine the measures lexicographically.
Le the local measure λ be be the measure (λ_1, λ_2) of the smallest SOV-cluster.

5.1 Definition This rule is intended to operate on an equation in E(C) of the form $X(s) = f(t_1,\ldots,t_n)$ within a minimal SOV-cluster C.
Rule Mixed-Cluster
Let $X_1,\ldots,X_m$ be an SOV-cluster, and assume there is an equation $X_1(s) = f(t_1,\ldots,t_n)$ in E(C). Then select one of the following alternatives.

i) Remove some X_i from Γ.
ii) For every X_i, select some index h(i) and replace X_i using $X_i(\Omega) = f(x_{i,1},\ldots,x_{i,h(i)-1}, X_i'(\Omega),x_{i,h(i)+1},\ldots,x_{i,n})$, where all the x_{ij} are new first-order variables.
Then decompose all the equations of the form $f(s_1,\ldots,s_n) = f(r_1,\ldots,r_n)$, i.e., replace them by the equations between the components, and use the rule Replace-Variables.
The equations of the form $f(s_1,\ldots,s_n) = p(r)$ are replaced by other equations by one application of PT-decomposition.

5.2 Lemma. The rule Mixed-Cluster is sound and complete, either fails or reduces the global measure μ or leaves μ invariant and reduce the local measure λ. Furthermore it leaves the invariants unchanged.
Proof. Soundness is obvious.
Completeness: Let σ be a unifier of Γ. If $\sigma X_i = \Omega$ for some X_i, then select alternative i). Otherwise, σX_i is a ground parametric term that starts with f. Hence, we can extend σ, such that there is a unifier after the replacement. The other applied rules are complete, hence the rule Mixed-Cluster is complete.
Measure We have to show that the measure is strictly decreased. For alternative i), this is obvious. Now consider alternative ii).
If all the indices h(i) are the same, then the equations in E(C') are derived from index h(i). The new first order variables can all be removed. Consider the new SOV-cluster C' consisting of X_i'. The multiset of terms in E(C') is smaller, since the terms derived from the parametric terms have p-depths not greater than the previous ones, but the terms in E(C') derived from the terms $f(t_1,\ldots,t_n)$ have smaller p-depths. Hence the measure λ_2 is properly decreased.
If the indices h(i) are different, then the equations resulting from the decomposition are split into n different classes, where n is the arity of the top level function symbol of some term in E(C). The decomposition and the following variable replacements of the new variables are able to remove all the new variables. But then the SOV-cluster splits into more than one new SOV-cluster, hence the number of SO-variables in C' is decreased. This means, λ_1 is strictly decreased. ♦

5.3 Definition. This rule is intended to operate on a minimal SOV-cluster C where some equation in E(C) is of the form X(s) = p(t), but none is of the form $X(s) = f(t_1,\ldots,t_n)$.
Rule Parametric-Cluster
Let $X_1,\ldots,X_m$ be an SOV-cluster, let E be the correlated set of equations, and assume there are some equations X(s) = p(t) in E. Select one of the following alternatives:

i) Remove some X_i from Γ.

ii) For every term $p_i(t_i)$ in E(C), apply split-middle with result $p_{i,0}$ and $p_{i,1}$.
If some $p_{i,0} = \Omega$, then FAIL.
Perform Add-Para-Equation($p_{i,0} = p_{1,0}$), replace the p_i in E(C) by $p_{1,0} \bullet p_{i,1}$, and replace all X_i by $p_{10}(X_i'(\Omega))$. Then remove $p_{1,0}$ from the top level of the equations by decomposition.
Either select some i and remove X_i and EXIT
or do nothing.
If some $p_{i1} = \Omega$, then EXIT.
Otherwise apply split-first to all $p_{i,1}$ with result $p_{i,2}$ and $p_{i,3}$ and replace the $p_{i,1}$ in E(C) accordingly.
Then apply Mixed-Cluster to the SOV-cluster C, where in the case that alternative ii) of Mixed-Cluster is chosen, it should not be the case, that there is some index j, such that all X_i' occur at position j. Otherwise the algorithm stops with FAIL. ♦

5.4 Lemma. The rule Parametric-Cluster is sound and complete, reduces the global measure μ or leaves μ invariant and reduces the local measure λ. Furthermore it leaves the invariants unchanged.

Proof. Soundness is obvious.

Completeness: Let σ be a unifier of Γ. If $\sigma X_i = \Omega$ for some X_i, then select alternative i).
Let r be the maximal common prefix of all the σX_i and σp_i for terms in E(C). Then we can guide split-middle, such that $\sigma p_{i,0} = r$, and $\sigma p_{i,1}$ is the corresponding rest. We can also do the appropriate action, if now some $\sigma X_i' = \Omega$. If $p_{i,1} \neq \Omega$ for all i, then we can apply split-first to all the parametric terms. Since the maximal common prefix was chosen for r, the last condition for the application of Mixed-Cluster can be satisfied. Hence the rule is complete.

Measure: We have to show that the measure is reduced. If alternative i) is chosen, then the measure μ is reduced. If alternative ii) is chosen, the first action using split-middle either decreases the measure or leaves it unchanged. In particular, if some $p_{i,1} = \Omega$, then the measure is reduced. The last step in applying Mixed-Cluster under some restrictions reduces the measure, since the SOV-cluster will be split into at least two, such that one has less SO-variables.

The invariants remain unchanged, since there is no creation of parametric terms. ♦

Note that it is not possible to use an unrestricted second-order decomposition replacing X(s) = X(t) by s = t, since then the stratified-property may be destroyed.

5.5 Definition.

Rule. Remove-SOV-Cluster-GT-1

Let $X_1, \ldots, X_m$ be some SOV-cluster C, such that E(C) contains only terms of the form $X_i(s)$. Assume that m > 1 and that the SOV-cluster has a minimal number of variables.
Select one of the following alternatives:

i) Remove some X_i.

ii) Replace every X_i using $X_i(\Omega) = Z(X_i'(\Omega))$, where Z is a new SO-variable.
In the equations in E(C), remove all the Z's.
Select one of the following two possibilities:

i) either remove some X_i'

ii) Select some k with $2 \le k \le m$ and select for every X_i an index h(i) with $1 \le h(i) \le k$, such that every index between 1 and k occurs at least once. Replace every X_i' by the term $F(x_{i,1},\ldots,x_{i,h(i)-1},X_i''(\Omega), x_{i,h(i)+1},\ldots,x_{i,k}))$, where all $x_{i,j}$ are new FO-variables and F is a new F-variable. Then decompose all the equations in E(C) using decomposition for F. Use replacement of variables.

Rule. Remove-SOV-Cluster-EQ-1
If there is an SOV-cluster with one SO-variable, and all terms in E(C) are of the form X(t),
then decompose all the equations in E(C), i.e., replace all the equations $X(s_i) = X(t_i)$ in Γ_T by $s_i = t_i$. ♦

Alternative ii) of the rule Remove-SOV-Cluster-GT-1 adds new equations to Γ_T and furthermore splits the SOV-cluster C into at least two SOV-cluster containing less SO-variables. The introduction of an F-variable is necessary, since otherwise we have to try all possibilities for function symbols in the signature, which would be only possible if the number of function symbols is finite.

5.6 Lemma. The rule Remove-SOV-Cluster-GT-1 is sound, complete and either fails or strictly reduces the global measure μ or leaves μ unchanged and strictly reduces λ. Furthermore it leaves the invariants unchanged.
Proof. Soundness is obvious.
Completeness: Let σ be a unifier. Either $\sigma X_i = \Omega$ for some i, then select alternative i). Otherwise, let σZ be the common prefix of all σX_i, and extend σ, such that $\sigma X_i'$ is the corresponding rest. Again either some $\sigma X_i'$ is equal to Ω or none is equal to Ω. In the first case we select ii.i) . Otherwise, the $\sigma X_i'$ all start with the same function symbol. This function symbol must have arity greater than 1. Let k be the number of different first symbols in the prefix of the position of Ω in σX_i. Obviously, $2 \le k \le m$. For every X_i, we consistently select an index, such that X_i's with the same first symbol of the position of Ω in σX_i have the same index. The same should hold for different first symbols. Now we can define σF and $\sigma X_i''$, and all the decomposed equations hold.
Measure: Either some SO-variable is removed, or the part ii.ii) produces an SO-cluster with a strictly smaller number of SO-variables.
Invariants The interesting observation is that the F-variables are only introduced after the SO-variable Z is introduced, where Z does not occur in Γ_T. ♦

5.7 Lemma. The rule Remove-SOV-Cluster-EQ-1 is sound, complete and either fails or strictly reduces the global measure μ. Furthermore it leaves the invariants unchanged.
Proof. Soundness is obvious.
Completeness: Every unifier σ of X(s) = X(t) is also a unifier of s = t.
The measure is reduced, since the number of SO-variables in Γ_T is decreased by one.
The invariants remain unchanged, since after application of the rule, the SO-variable X is no longer in Γ_T, and the SOP for variables in the new terms is consistently reduced. The SO-prefixes in Γ_P of these do not matter, since X occurs also in every such term in Γ_P.

5.8 Algorithm. The algorithm to remove SOV-cluster operates as follows:
A precondition for the algorithm is that all equations in the SOV-cluster are of the form $X(s) = X(t)$, $X(s) = f(t_1,\ldots,t_n)$ or $X(s) = p(t)$, and that there is no SO-cycle.
Use the rules Mixed-Cluster, Parametric-cluster, Minimize-SOV-Cluster-GT-1 and Minimize-SOV-Cluster-EQ-1 to proceed, if an appropriate minimal SOV-cluster is available. If the measure μ is strictly reduced, then we have a new game. We have again to look for SO-cycles.
The rules from section 3 are to be used if appropriate to replace variables or to transform the system into decomposed normal form. ♦

5.9 Theorem. If there is an SOV-cluster in Γ_T, then the non-deterministic algorithm 5.8 terminates and either fails or strictly reduces the measure μ.
Proof. If there is some SOV-cluster, then one of the rules above can be applied. It either strictly decreases μ or leaves μ invariant and reduces the local measure. Whenever we have a decomposed normal form without an SO-cycle and if the system Γ_T is not empty, we can apply some rule that operates on SOV-clusters. There are two possible cases after the application of a rule: Either there is an equation $x = X(s)$ in Γ_T, or there is a smaller SOV-cluster in Γ_T. In the first case, we can apply Replace-Variable, and reduce μ_2. In the other case, we operate on the smaller SOV-cluster. Since the ordering is well-founded, the generation of smaller SOV-clusters terminates. ♦

6. Properties of Non-Power Strings

In order to treat the case where $\Gamma_T = \emptyset$, we have to investigate properties of strings.
A string u is called a **non-power**, iff there is no other string w, such that $u = w^n$ for some positive integer $n > 1$. We call a string u a **cyclic permutation** of w, iff $u = u_1u_2$ and $w = u_2u_1$, where u_1 and u_2 are non trivial strings. Note that the cyclic permutation relation is an equivalence relation, if we add reflexivity. We use $|s|$ to denote the length of a string.
It is well-known, that $uv = vu$ for two nontrivial strings u and v implies hat there exists some non-power string w, such that $u = w^n$ and $v = w^m$ for some positive integers n and m.

6.1 Lemma. Let s and t be non-power strings such that t^2 is a substring of s^n and $|s| \le |t|$.
Then s is a cyclic permutation of t.
Proof. Without loss of generality, we can assume that tt is a prefix of s^n:
Otherwise, $s^n = s^k s_1 s_2 s^h$, and tt is a prefix of $s_2 s^h$. Then tt is a prefix of $s_2 s^h s^k s_1 = (s_2 s_1)^n$, and $s_2 s_1$ is a cyclic permutation of s. Here we use that cyclic permutativity is an equivalence relation.
If $|s| = |t|$, we are ready, since tt is a prefix of s^n. Now consider the case that $|s| < |t|$.
It is not possible, that $s^k = t$ for some $k > 1$, since t is a non-power. Hence $t = s^k s_1$, and t is a prefix of $s_2 s^h$ for some k, $h \ge 1$, where $s = s_1 s_2$. Furthermore $s_1 \neq \varepsilon$ and $s_2 \neq \varepsilon$. From $s^k s_1 = s_2 s^h = s_2 s_1 s_2 s^{h-1}$, we get that $s_1 s_2 = s_2 s_1$, which is a contradiction to the assumption that s is a non-power. ♦

6.2 Lemma. Let $(s_1 \bullet \ldots \bullet s_n) = t^m$, where s_i and t denote strings such that $s_1 \bullet \ldots \bullet s_n$ is flattened, and m and n positive integers ≥ 2. Let m ≥ 3*n, let t be a non-power, and let the s_i either be constants or powers t_i^{ni}, where t_i is a non-power.
Then for some power t_i^{ni} in $(s_1 \bullet \ldots \bullet s_n)$, t_i is a cyclic permutation of t.
Proof. There must be some power t_k^{nk} in $(s_1 \bullet \ldots \bullet s_n)$, since otherwise the lengths of $(s_1 \bullet \ldots \bullet s_n)$ and t^m disagree. Let t_k^{nk} be the power with the largest number of symbols in $(s_1 \bullet \ldots \bullet s_n)$. There are two cases: if $|t| \geq |t_k|$ and $|t| \leq |t_k|$. If $|t| \leq |t_k|$, then we have that t_k^2 is a substring of t^m and we can apply Lemma 6.1. Now consider the case $|t| \geq |t_k|$. Since t_k^{nk} has more symbols than all other s_i, we obtain the inequation $n*|t_k^{nk}| \geq |t^m| \geq n*|t^3|$, hence $|t_k^{nk}| > |t^3|$. This implies that t^2 is a substring of t_k^{nk} and we can apply the lemma above. ♦

6.3 Lemma. If the alphabet is fixed and finite, then unifiability of an associative system of equations and disequations can be decided using a decision algorithm for associative equations. We assume that the empty string cannot be used as a replacement for a variable.
Proof. Let $s \neq t$ be a disequation. Then we replace this disequation by equations. There is a nondeterministic choice:

i) s = tx, where x is a new variable
ii) t = sx, where x is a new variable
iii) s = xay, t = xbz, where $a \neq b$ are chosen non-deterministically from the alphabet and x,y,z are new variables
Non-deterministically remove some or none of the variables x,y,z. This gives 8 cases.
Since the alphabet is finite, the number of choices is also finite.

It is not hard to see that this transformation method is correct. ♦

Thus it is no restriction to add some disequations to an associative system of equations if the letters of the alphabet are fixed and finite. We can perform a non-deterministic preparating step that replaces every disequation by a finite set of equations, and then use the associative unification algorithm of Makanin.

7. Applying Associative Unification

This section is devoted to the case where Γ_T is empty. The remaining task is then to decide unifiability of Γ_P.
We will also need parametric words (without variables) (cf. [Fa88]). Let A be an alphabet. Then a parametric word is either a string of A, or w_1w_2, where w_1, w_2 are parametric words, or w^n, where w is a parametric word and n an integer variable.
Note that the parametric words in [Fa88] have slightly more general exponents, since linear polynomials are permitted.

7.1 Definition. Rule "Apply-Associative-Unification".
Let Γ_P be the subset of equations of Γ between parametric terms.

Then all parametric terms in TBPT(Γ_P) are considered as parametric words over the alphabet of the constants. "$f(t_1,\ldots,t_{i-1},\Omega,t_{i+1},\ldots,t_n)$", where $f(t_1,\ldots,t_{i-1},\Omega,t_{i+1},\ldots,t_n)$ is a basic parametric term. We interpret • as associative concatenation. Note that now also F-variables may appear in the parametric terms.
We construct a string unification problem Γ_A as follows:
First all basic parametric terms are replaced consistently be some constants that constitute the alphabet. This gives a set of equations between parametric words.
The following rule is used to eliminate powers

Let p^n be a power expression of some parametric word. Then let x,y be two new variables,
Replace p^n by x in the new system Γ_A.
Add the following equations to Γ_A: $p = y, yx = xy$.
In order to encode the property that p is a non-power, let $p = p_1 \bullet \ldots \bullet p_m$.
If m = 1, then p is a basic parametric term, since $(p^n)^k$ cannot occur in Γ.
If m > 1, we add the following disequations: $y \neq y_2y_2$, $y \neq y_3y_3y_3,\ldots$, $y \neq (y_{3*m})^{3*m}$, where the y_i are new variables. Note that m is a given positive integer.
Furthermore, for i = 1,…,m, we add the m disequations
$p_2 \bullet \ldots \bullet p_m \bullet y \neq y \bullet p_2 \bullet \ldots \bullet p_m$, $p_1 \bullet p_3 \bullet \ldots \bullet p_m \bullet y \neq y \bullet p_1 \bullet p_3 \bullet \ldots \bullet p_m$, … ,
$p_1 \bullet \ldots, p_{i-1} \bullet p_{i+1} \bullet \ldots \bullet p_m \bullet y \neq y \bullet p_1 \bullet \ldots, p_{i-1} \bullet p_{i+1} \bullet \ldots \bullet p_m$, … , $p_1 \bullet \ldots \bullet p_{m-1} \bullet y \neq y \bullet p_1 \bullet \ldots \bullet p_{m-1}$.

This rule is used until no powers p^n remain in Γ_A
Then we use the algorithm in Lemma 6.3 to transform the disequations into equations of Γ_A, where the alphabet is the set of all basic parametric terms.

If the resulting system Γ_A is unifiable w.r.t. associative unification, then the final answer is "unifiable", otherwise "FAIL". ♦

7.2 Lemma. The rule Apply-Associative-Unification" is sound and complete.
Proof. Completeness: Let σ be a unifier of Γ. This means that different basic parametric terms in TBPT(Γ_P) are different under σ, hence we can also construct an associative unifier for the initial Γ_A. The replacement of p^n by different variables that commute is justified, since $\sigma(p^n)$ and σp commute. Since we have assumed that σp is a non-power, if p^n occurs in Γ, the disequations $\sigma y \neq \sigma(y_k)^k$ are valid for $k \geq 2$. Furthermore, since σy is a non-power, for every proper, nontrivial substring u of σy, we have also $u\sigma(y) \neq \sigma(y)u$.
The hard part is soundness: Let σ be a unifier of Γ_A, where the alphabet is restricted to the letters occurring in Γ_A. We have to show that for p^n, and for the translation y of $p = p_1 \bullet \ldots \bullet p_m$ and x of p^n, we have that σx is a power of σy. Note that if m = 1, then p_1 is a basic parametric term and we are ready. Now let m > 1. Since xy = yx is an equation, we have $\sigma(xy) = \sigma(yx)$ and furthermore all the translated disequations are satisfied. Suppose $\sigma y = u^k$ for $k \geq 2$ and some non-power string u, then the first kind of disequations imply that k > 3*m. Since k > 3*m, not all of the p_i can be basic parametric terms. Then Lemma 6.2 shows that there is some i such that $\sigma p_i = v^h$ with a non-power v for some h > 1 and v is a cyclic permutation of u. But then we would have $\sigma(p_1 \bullet \ldots \bullet p_{i-1} \bullet p_{i+1} \bullet \ldots \bullet p_m \bullet y \neq y \bullet p_1 \bullet \ldots \bullet p_{i-1} \bullet p_{i+1} \bullet \ldots \bullet p_m$, which contradicts the disequation $p_1 \bullet \ldots \bullet p_{i-1} \bullet p_{i+1} \bullet \ldots \bullet p_m \bullet y \neq y \bullet p_1 \bullet \ldots \bullet p_{i-1} \bullet p_{i+1} \bullet \ldots \bullet p_m$. Hence y is a non-power, and then $\sigma(xy) = \sigma(yx)$ implies that $\sigma x = (\sigma y)^h$ for some positive h. ♦

8 The Main Algorithm for Stratified Second Ordewr Systems

In this section we want to assemble the different parts of the unification algorithm.

There are several non-deterministic subalgorithms that perform some specific transformation tasks.

1) Transform the system Γ_T into decomposed normal form using the transformations from section 3.
2) If there is some SO-cycle, then eliminate the SO-cycle from Γ_T using the transformations in section 4.
3) If there is no SO-cycle in Γ_T, but Γ_T is not empty, then there is some SOV-cluster. Use the transformations in section 5 to make the system Γ_T smaller.
4) If Γ_T is empty, then use the transformations of section 7 to generate an associative unification problem, and decide unifiability using Makanin's decision procedure.

8.1 Theorem Unifiability of stratified second-order systems is decidable.
Proof. The algorithm in this section is a decision algorithm: Every non-deterministic transformation terminates, and the number of choices is finite. This is obvious for all rules. Since all rules are sound and complete, we have that the input system is unifiable, iff there is some final system that is unifiable.
Furthermore, unifiability of the final systems is decidable [Ma71, Sch93]. ♦

A straightforward extension of this algorithm to produce a complete set of unifiers would be as follows. The instantiation of variables have to remembered and the final associative unification step has to produce a complete set of unifiers. For example, the algorithm given in [Jaf90] can be used for generating a minimal, and complete set of unifiers of an associative unification problem. However, note that there may be instantiations containing F-variables.

The complexity of the algorithm is rather bad. It is non-deterministic, has at least exponential space usage, and then uses a decision algorithm for associative equations. The theoretical complexity of the problem is thus only bounded above. It is open whether the unification problem for stratified second order systems is associative-unification-hard.

9 Parametric Stratified Second Order Systems

We want to generalize the possible input such that parametric terms are permitted in the input. Therefore, we need some notions from [Fa88]. A parametric word w is **simple**, if for all p^n that occur in w, the term p has only instances that are non-powers. For example, $(a^nb)^m$ is simple, but $(abb(ab^n))^m$ is not simple, since for n = 2, the basis of the exponent is abbabb, which is a non-power. This property of (general) parametric words is decidable, as shown in [Fa88] (see also 9.1)

The following restrictions must hold in order to permit parametric terms in the input:
i) there are no powers of the form $(p^n)^m$,
ii) for two occurrences p^n and q^n, the parametric terms p and q are identical
iii) the system is stratified
We call such systems **parametric stratified second order systems**.

It is possible with the methods developed in this paper to give a simple proof of two Corollaries in [Fa88], however, we use the decidability of associative unification, whereas Farmer describes a different algorithm, which may have a better complexity.

9.1 Corollary.
i) It is decidable, whether parametric words are simple.
ii) The unification problem for simple parametric words is decidable.
Proof. i) Let q be a given parametric word. First we check for powers of the form $(p^n)^m$ that may occur in q. Then we simply use the translation method in 7.1 for q, where m is a new integer variable, and get a set of string equations. If this system is solvable, then the word q was simple. The argumentation from Lemma 7.2 can be copied.
ii) Again we use the translation from 7.1 and 7.2. Since bases can never be powers, the translation method is complete. For the result we use the decision algorithm for associative unification. ♦

The following first non-deterministic transformations checks, whether the input is acceptable, and if the check succeeds, makes the input acceptable for our algorithm. The intuition behind this check is that the algorithm can not handle parametric terms that have instances of the form $(p^m)^n$.

9.2 Definition. Input transformation for parametric stratified second order systems Γ.
i) Make a non-deterministic identification on all the basic parametric terms in Γ, i.e., select an equivalence relation $\approx$ on all basic parametric terms, such that $p \approx q$ implies that the top function symbol and the index of Ω in p and q is the same.
For all basic parametric terms p, q with $p \approx q$, replace p by q, and then add the equation resulting from decomposition. I.e. If $p = f(t_1,\ldots,\Omega,\ldots t_n)$ and $q = f(s_1,\ldots,\Omega,\ldots,s_n)$, and Ω occurs at index j, then add $s_i = t_i$ for all $i \neq j$.
ii) Check the following:
Either for all parametric terms of the form p^n, the base p is a simple word, if different terms are considered different
Or every possible decomposition to a decomposed normal form terminates with FAIL: not unifiable.
If for all possible non-deterministic choices in i), the check in ii) succeeds, then we call the system **admissable**. ♦

9.3 Proposition. Admissability of a parametric stratified second order system is decidable.

Proof. This follows from results in [Fa88] and from Lemma 9.1 and since the number of choices for the equivalence relations is bounded by the input size of Γ, and the decomposition algorithm terminates as shown in 3.15 ♦

9.4 Lemma. Let Γ be an admissable parametric stratified second order systems. If σ is an unrestricted unifier of Γ, then it is possible to transform Γ into Γ' using 9.2, such that Γ' has a unifier, and furthermore satisfies the invariants defined in section 3.
Proof. Let σ be an unrestricted unifier of Γ. If for some r^n, σr is a power, then we show that the system is not admissable:
The equivalence relation can be chosen as $p \approx q$, if $\sigma p = \sigma q$. The replacements then enforce that syntactically different basic parametric terms are different under σ. The second condition in 9.2 ii) does not hold, since Γ has an unrestricted unifier. Now the parametric term r has as instance some power, hence r is not simple. Thus the system is not admissable, hence there is no such power. This means that σ is a unifier of Γ' in the sense of the definition in section 3.
The resulting system satisfies all the invariants. ♦

9.6 Theorem. Unifiability of admissable parametric stratified second-order systems is decidable.
Proof. This holds, since after the input transformations, we can apply the unification algorithm for stratified second order systems. ♦

Note that permitting nesting powers $(p^m)^n$ would lead to an unresolved problem during encoding the problem as associative unification problem. This would require an encoding of a set of solutions "$x = a^n$ and $y = x^m$" using a word equation. I have found no way of encoding such a solution, and conjecture that such an encoding doesn't exist.

Open Problem. A related problem that appears to be open is the decidability of the unification problem for parametric words with unique basis of exponents, i.e. systems of equations that satisfy the condition that for all p^n and q^n occurring somewhere, p and q are syntactically equal

10 Dynamically Stratified Second Order Systems

In the following we will talk about a generalization to dynamically stratified systems, and we do this in a sketchy way.

The idea for the generalization is to use the rules without taking care of stratifiedness until we get stuck, or the system is solved. If there is no non-deterministic execution that never gets stuck, then the system can be tested for unifiability using the methods in this paper.

We have to adapt some definitions to meet the purposes of this generalization.
A **flat SO-cycle** is a cycle $X_1(s_1) = t_1$, $X_2(s_2) = t_2$, … ,$X_n(s_n) = t_n$, such that for i = 2,…,n, X_i occurs in t_{i-1} and X_1 occurs in t_n, such that for every i at least one such occurrence has empty second order prefix.

A **deep SO-cycle** is such a cycle that does not satisfy the condition on the second order prefix.
An **SOV-cluster** C is a $\geq_{SO}$-maximal equivalence class, such that every SO-variable in C has an empty second-order prefix.

Now we adapt the rules of section 3 to the new situation:
The following non-deterministic rules may be used in addition
Rule : Remove X from Γ.
Rule Occur-Check**.** If there is an equation x = t, such that t is not a variable, and x occurs in t, and there is some first order function symbol on a path from the root of t to some occurrence of x, then FAIL: the system is not unifiable.
Rule X(s) = X(t) can be replaced by s = t.

In section 4 on removing SO-cycles, the rules are the same, however they have to concentrate on a flat SO-cycle. It is easy to verify that the rules produce always a new flat and smaller SO-cycle.
In the case of an SO-cycle of length 1, we have the additional rule:
Rule If there is a flat SO-cycle X(s) = t of length 1, and the occurrence of X in t is not unique, then remove X from Γ.

In section 5 on SOV-clusters, there is no modification.

10.1 Definition dynamically stratified systems.
We permit second order systems of equations that obey the following restrictions:
i) there are no directly nested powers in parametric terms, i.e. $(p^m)^n$ is forbidden
ii) for two occurrences p^n and q^n, the parametric terms p and q are identical.
Furthermore, the admissablitiy check of 9.2 should accept the system.
The (dynamic) restriction is that there is some non-deterministic execution of the algorithm, such that no specific execution generates a system with the following property:
i) It is in decomposed normal form
ii) There is no flat SO-cycle.
iii) There is no SOV-cluster.
We will call such systems **dynamically stratified systems.** ♦

This is a decidable condition, simply adding an appropriate check to the algorithm, and then trying all possibilities of branching until it stops with “no rule applicable” or with an empty set Γ_T or with “not unifiable”.

An example of dynamically stratified systems are second-order matching problems (without parametric terms), i.e., unification problems, where in every equation, one term is a ground first -order term.
Another example is a system that becomes stratified after some applications of rules, for example x = X(y), z = Y(x) is not stratified, but elimination of x makes it stratified.

10.2 Definition. Structure of the generalized algorithm for dynamically stratified systems.
The main operations of the global algorithm are:

1) If the input contains parametric terms, use the method from 9.2 to check whether the system is admissable.
2) Transform all equations into decomposed normal form.
3) If there is a flat SO-cycle, eliminate it using the methods from section 4.
4) If all equations are in the form X(s) = t, and there is some SOV-cluster, use the methods of section 5) to proceed.
5) If Γ_T is not empty, but there is no SO-cycle nor a SOV- cluster, then return "algorithm is not applicable".
6) If Γ_T is empty, use the method of section 7) to decide unifiability.

10.3 Theorem. The unification problem for dynamically stratified second-order systems is decidable.

In order to extend the algorithm to the case in 10.2, where the algorithm answers "not applicable", it appears to be necessary to permit second order variables in Γ_P. This, however, would require new methods and is beyond the scope of this paper.

11 Application to infinite term rewriting systems

Using second order variables together with constraints, it is possible to represent several infinite term rewriting systems in a finite way and to show local confluence by overlapping and computing second order unifiers. In order to be general enough, we have to use dynamically stratified systems, or equivalently, we use the unification algorithm until it signals "not applicable".
We do not formally treat this topic since this is beyond the scope of this paper. In order to apply the results, the unification algorithm in this paper must be modified to be a complete unification algorithm. This however, does not help since the final step of associative unification may generate an infinite set of unifiers. Nevertheless, we want to give some hints on the consequences, under the premise that the only finitely many unifiers are necessary.

An example of a term rewriting system that produces an infinite set of rewrite rules on Knuth Bendix completion is the system
R: $fgfx \to gfx$
On completion, the infinitely many rules $fg^nf\ x \to g^nfx$ are generated
These rules can be represented by the second order rule
R_2: $fXf(x) \to X(f(x)$ together with the constraint $X(z) = g(X(a))$.
Now we make the computation that check for local confluency in the generalized case.

We have several kinds of overlap:
1) Overlap at top level.
$fXf(x) = fYf(x')$, $X(z) = g(X(a))$, $Y(z') = g(Y(a))$.

Using the second order unification algorithm for dynamically stratified systems, we get
$Xf(x) = Yf(x')$, $X(z) = g(X(a))$ and $Y(z') = g(Y(a))$
We replace X by $g(\Omega)^n$ and Y by $g(\Omega)^m$
$g^n(\Omega)(f(x)) = g^m(\Omega)(f(x')$, $z = g(a)$ and $z' = g(a)$
Since g is unary, the only alternative that will not fail is that $g^n(\Omega) = g^m(\Omega)$ is added to the associative subsystem, which has as only solution n = m, and that x = x'. In this case the right hand sides are also identical.

2) The overlap is at the inner position of f.
The unification problem is now: $f(x) = fYf(x')$, $X(z) = g(X(a))$, $Y(z') = g(Y(a))$.
We can ignore the second equation, since it is solvable, but does not influence the other parts. The next step is decomposition
$x = Yf(x')$, $Y(z') = g(Y(a))$.
Now we can check for confluence of the critical pair:
$fXfYf\ x' \rightarrow XfYf\ x' \rightarrow XYf\ x'$ with constraint $X(z) = g(X(a))$, $Y(z') = g(Y(a))$.
$fXfYf\ x' \rightarrow fXYf(x') \rightarrow XYfx'$ with the same constraints.
The terms on the right hand side are identical, hence all the critical pairs that are represented can be joined.

3) The overlap is within the second order variable.
This gives $X_2(f(x)) = fYf(x')$, $X_1X_2(z) = g(X_1X_2(a))$, $Y(z') = g(Y(a))$.
We will get $X_1(\Omega) = g^n(\Omega)$, $X_2(\Omega) = g^m(\Omega)$ and $Y(\Omega) = g^k(\Omega)$. This will lead to FAIL.
The right alternative is to remove X_2, giving $f(x) = fYf(x')$, $X_1(z) = g(X_1(a))$, $Y(z') = g(Y(a))$, which is exactly case 2) of the overlapping.

This shows that the system R_2 is locally confluent.

This application to term rewriting systems has its problems, since we have to define reduction. The computation of a reduction step may require to solve second order unification problems, which may be undecidable in the most general case.

Conclusion.

We have presented a decision algorithm for unifiability of a new general class of second order systems of equations. There are several interesting further issues and extensions, which are worth investigating.

i) loosening some conditions to cover broader classes of problems. One direction could be to extend the semantics of possible instances of second order monadic variables to permit terms with several holes. However, the current methods appear not to be sufficient for solving these problems, since it is not clear how the enforce termination.
ii) The algorithm is rather complex and should be optimized.
iii) Applications to completion of rewrite systems, and combination with sorts and membership constraints .

Acknowledgements

I thank Klaus Schulz for his advice concerning associative unification and parametric words, and Hubert Comon for discussions concerning second-order unification problems.

References.

BS93 Baader, F, Siekmann, J., Unification Theory, in D.M. Gabbay, C.J. Hogger, J.A: Robinson (eds.), Handbook of Logic in Artifiial Intelligence and Logic Programming, Oxford University Press, (1994)

BHS89 Bürckert, H.-J., Herold, A., Schmidt-Schauß, M., On Equational Theories, Unification and (Un)Decidability, J. Symbolic Computation 8, pp. 3-49, 1989

CH91 Chen, H., Hsiang, J., Logic programming with recurrence domains, ICALP 91, LNCS 510, pp. 20-34, (1991)

CHK90 Chen, H., Hisang, J., Kong, H.C., On finite representation of infinite sequences of terms. Proc. of 2nd CTRS, Springer LNCS 516, pp. 100-114, (1990)

Con93 Contejean, E., Solving *-problems modulo Distributivity by a reduction to AC1-Unification, J. of Symbolic Computation 16, pp. 493-521. (1993)

Com91 Comon. H., Completion of rewrite systems with membership constraints, technical report 699, LRI, Orsay, France, (1991).

Com92 Comon, H., On unification of terms with exponents, Technical report 770, LRI, Orsay, France, (1992). to appear in Mathematical System Theory

Com94a Comon. H., Completion of rewrite systems with membership constraints, part I: Deduction rules and Part II: Constraint Solving, (to appear in J. Symbolic Computation), (1994)

Com94b Comon, H., presentation on the unification workshop , 1994, (Nancy), (1994)

Com95 Comon, H., On unification of terms with integer exponents, Mathematical Systems Theory, 28(1), pp 67-88, (1995)

Fa88 Farmer, W., A unification algorithm for second order monadic terms, Annals of Pure and Applied Logic, 39, pp. 131-174, (1988)

Fa91 Farmer, W., A., Simple second-order languages for which unification is undecidable. Theoretical Computer Science 87, pp. 173-214, (1991)

Gol81 Goldfarb, W. The undecidability of the second-order unification problem, J. TCS 13, pp. 225-230, (1981)

Gra88 Gramlich, B., Unification of term schemes - theory and applications, SEKI report SR-88-18, Universität Kaiserslautern, Germany, (1988)

Her92 Hermann, M. On the relation between primitive recursion, schematization, and divergence, Proc. 3rd Conf. on Algebraic and Logic programming, Volterra (Italy), LNCS 632, pp. 115-127, (1992)

Jaf90 Jaffar, Minimal and complete word unification , J. of the ACM, 37, 47-85, (1990)

KiC89 Kirchner, C. (ed.), Unification, reprint from J. of Symbolic Computation, Academic Press, (1989)

KiH89 Kirchner, H., Schematization of infinite sets of rewrite rules generated by divergent completion processes, J. TCS 67, pp. 303-332, (1989)

Ma77 Makanin G.S:, The problem of solvability od equations in a free semigroup. Math. Sbornik, 103, 147-236, (1977), english translation in Math USSR Sbornik 32, (1977)

Sa92 Salzer, G., The unification of infinite sets of terms and its applications, Proc. Logic Programming and Automated Reasoning (LPAR '92) LNAI 624, pp. 409-420, (1992)

Sch89 Schmidt-Schauß, M., Unification in a combination of arbitrary disjoint equational theories, J. Symbolic computation 8, 51-99, (1989)

Sch93 Schulz, K.U., Word unification and transformation of generalized equations, J. Automated Reasoning 11, pp. 149-184, (1993)

Sch94 Schmidt-Schauß, M., An algorithm for distributive unification, internal report 13/94, J.W.Goethe-Universität Frankfurt, (1994)

Si89 Siekmann, J., Unification theory: a survey, in C. Kirchner (ed.), Special issue on unification, Journal of Symbolic Computation 7, (1989)

So94 Socher-Ambrosius, R., Unification of terms with exponents, internal report, MPI-I-93-217, Max Planck Institut, Saarbrücken, (1994),

Sz82 Szabó, P., Unifikationstheorie erster Ordnung, Dissertation, Karlsruhe, (1982)