Universität München
Centrum für Informations- und
Sprachverarbeitung

Word Unification and Transformation of Generalized Equations

Klaus U. Schulz

In: *J. of Automated Reasoning*, 11:149–184, 1993.

Word Unification and Transformation of Generalized Equations

Klaus U. Schulz
CIS, University of Munich
Leopoldstr. 139, D-8000 Munich 40

October 11, 1994

Abstract

Makanin's algorithm [Ma77] shows that it is decidable whether a word equation has a solution. The original description was hard to understand and not designed for implementation. Since words represent a fundamental data type, various authors have given improved descriptions [Pé81, Ab87, Sc90, Ja90]. In this paper we present a version of the algorithm which probably cannot be further simplified without fundamentally new insights which exceed Makanin's original ideas. We give a transformation which is efficient, conceptually simple and applies to arbitrary generalized equations. No further subprocedure is needed for the generation of the search tree. Particular attention is then given to the proof that proper generalized equations are transformed into proper generalized equations. This point, which is important for the termination argument, was treated erroneously in other papers. We also show that a combination of the basic algorithm for string-unification (see [Pl72, Le72, Si75, Si78] and Makanin's algorithm offers a simple solution to the problem of terminating minimal and complete word unification.

Introduction

One of the simpler tasks which a text editing system has to solve again and again is the problem to determine whether a particular string W occurs in a given text T. This problem may be expressed by means of an equation $T == xWy$, where x and y are variables and T and W are strings of constants. Obviously W occurs in T if and only if there exists words X and Y in the text alphabet which solve this equation, i.e., words X and Y such that T and XWY are identical. What we have to solve is a particular *word equation*. A word equation is an expression of the form $W_1 == W_2$, where $W_i \in (\mathcal{C} \cup \mathcal{V})^+$ are non-empty words in a mixed alphabet $\mathcal{C} \cup \mathcal{V}$ of constants and variables respectively. Of course the set of constants $\mathcal{C}$ and the set of variables $\mathcal{V}$ are disjoint. A *solution* of the word equation $W_1 == W_2$ is an assignment of values $X_i \in \mathcal{C}^+$ to the variables x_i occurring in the equation such that W_1 and W_2 become identical if all variables are replaced by the corresponding values. When the values X_i are allowed to be words in the mixed alphabet $(\mathcal{C} \cup \mathcal{X})^+$, then we get the notion of a *unifier* of the word equation.

The importance of the problem to decide whether a word equation has a solution/unifier becomes apparent if other formulations are used. Word equations may be called equations in a free semigroup, equations over lists (of atomic elements) with concatenation, or associative unification problems with constants, stressing their role in mathematical logic (e.g., [Hm71, Ma77], constraint logic programming (e.g., [Col88]) or universal unification theory ([Ba90, Si89]).[1]

Historically A.A. Markov, at the end of the 1950's, was probably the first to ask whether it is decidable if a word equation has a solution. Markov noted that every word equation over a two-letter constant alphabet may be translated into a finite system of diophantine equations, preserving solvability in both directions. He hoped to obtain a proof for the unsolvability of Hilbert's tenth problem by showing that solvability of word equations is an undecidable problem (see [Ma81] for more details). Approximately at the same time Lentin and Schützenberger [LeSc67] independently considered word equations.

In the following period, in the western countries the main attention was given to the (relatively simple) problem to enumerate in a compact form the set of all solutions of a word equation. Plotkin [Pl72], in the context of resolution based theorem proving, gave a simple algorithm to generate a minimal and complete set of unifiers (see section 5 for these notions) for a given word equation (see also [Le72, Si75, Si78].[2] In the eastern countries, the much harder decision problem was addressed. Hmelevskiĭ, [Hm71] obtained a partial solution, showing that the solvability of word equations with three variables is decidable. Later G.S. Makanin showed in his epochal paper [Ma77] that the solvability of arbitrary word equations is decidable.

[1]As a matter of fact it is simple to decide solvability of *matching equations* like $T == xWy$, and efficient algorithms are known for these restricted problems (see, e.g., [Ah90]).

[2]Plotkin's algorithm was able to deal with a more general problem, namely unification of first order terms modulo associativity of a given function symbol. Plotkin [Pl72], in contrast to Siekmann [Si78] does not prove that this algorithm is correct.

The Problem of an Optimal Form

Makanin's algorithm is based on the new data type of *generalized equations*. It starts translating a word equation into a finite set of generalized equations. Then two subprocedures, "transformation" and "normalization," are used to generate a finitely branching search tree. In general, this subtree generation process does not stop, and Makanin uses an ingenious idea for termination.

The original decision procedure was not designed for implementation and is hard to understand. Several attempts have been made to find a better description [Pé81, Sc90, Ja90] and the algorithm has now been implemented [Ab87]. In this paper we shall present a version of Makanin's algorithm which probably cannot be further simplified without fundamentally new insights into the problem — insights which essentially exceed Makanin's original ideas. Our presentation will be based on J. Jaffar's [Ja90] modified notion of generalized equations which replaces Makanin's concept of a "boundary connection" by the concept of a "boundary equation." With this step, various important improvements are obtained: the normalization subprocedure which occurs in [Ma77, Pé81, Ab87, AbPe89, Sc90] is avoided and transformation of generalized equations becomes simpler since no boundary connections have to be updated. This source of complexity has been removed from the algorithm and is now situated in the correctness proof.

Jaffar's algorithm may be described as a tree generation process which is based on the iteration of two subprocedures, completion and transformation (= reduction). The former — trivial — algorithm transforms every generalized equation in an equivalent set of completed generalized equations. The latter procedure transforms a completed generalized equation into a simpler generalized equation. Jaffar, as well as [Ma77, Pé81, Ab87, AbPe89], distinguishes several types of completed generalized equations. For each type a special transformation rule is given.

In this paper we present an improved version of the transformation algorithm which has a built-in completion and consists of just one rule which applies to arbitrary completed generalized equations[3]. Thus, with the new transformation algorithm, the search tree generation process is based on one subprocedure only. From a conceptual point of view, the effect of a transformation step may be described very easily:

- At a transformation step, a *non-empty* left part of the generalized equation is *simultaneously* carried towards the right side of the generalized equation.

[3] A similar procedure was introduced in [Sc90], based on Pécuchet's notion of a position equation. But it turns out that the procedure is much simpler if based on Jaffar's representation.

With this property the new transformation is very similar to the transformation steps which are used in Plotkin's (Lentin's) procedure. Our main aim, however, was to reduce generalized equations as efficient as possible, even for the price that proofs become more complex.

- In essence, one step of our algorithm combines all those steps of Jaffar's algorithm that are made with the same "carrier". Thus, a maximal number of bases and boundaries are transported simultaneously. In comparison with sequences of case dependent transformations a lot of redundant work is avoided. The permanent encoding and decoding of information in and from boundary equations is widely avoided and new boundary equations are only introduced in specific cases.

The simplicity of the new transformation makes it very natural and easy to implement.

The Problem of a Complete and Correct Proof

The proof that Makanin's algorithm is correct and complete has its own history. It has turned out that the definition of a generalized equation has to contain two subtle conditions (see [Sc90], pg. 126) whose sense becomes only clear when technical details of the transformation algorithm are considered. Unfortunately, this point was not treated correctly in the "classical" papers on Makanin's algorithm. Already Makanin's description [Ma77] (at least in its English version) contained a misprint in the definition of a boundary connection[4]. Supported by the subsequent formulations in [Ma77], most readers were led to a real misinterpretation of this definition. In particular, Pècuchet's and Abdulrab's definition of a position equation in [Pé81, Ab87, AbPe89] follows such a misinterpretation and does not lead to a correct proof. Similar difficulties arise from Jaffar's [Ja90] notion of a proper generalized equation.

In the meantime, these points are more or less wellknown among the experts in the field. But, to my knowledge, there is no journal publication which contains a complete and error-free proof showing that transformation of generalized equations behaves as it should behave. Such a proof will (hopefully) be given in the appendix of this paper where we show that proper generalized equations are transformed into proper generalized equations with our transformation rule.

The structure of the paper is as follows. In section 1 we shall first describe the basic algorithm for word unification which goes back to [Le72, Pl72, Si75, Si78].

[4]In [Ma77], the last inequality of (3.11) should be $l_{\alpha(\lambda_k)} \geq l_{\alpha(\Delta(\lambda_k))}$. In later, less known papers [Ma80, Ma81], this misprint was eliminated but the notion of "convexity" given there was not quite correct.

The behaviour of this algorithm in a particular subcase will be helpful to understand one of the main ideas behind the definition of a generalized equation. This definition is given afterwards. In the third part of section 1 we will sketch how word equations are translated into sets of generalized equations. In section 2 we introduce some notation, and in section 3 we define the transformation algorithm and show that transformation preserves unifiability in both directions. We obtain a correct and complete procedure testing unifiability of word equations. In section 4 we discuss termination. For this purpose, the notion of a proper generalized equation is introduced. In section 5 we shall prove — following an idea of F. Baader — that a combination of Plotkin's and Makanin's algorithms gives a simple solution to the problem of *terminating minimal and complete word unification* which was first solved in [Ja90]. This is the problem to find an algorithm which generates a minimal and complete set of unifiers for a given word equation and terminates if this set is finite.

1 Word Equations and Generalized Equations

In this section we want to introduce the concept of a generalized equation and to show how word equations are translated into sets of generalized equations. As mentioned above we shall start with a description of the basic algorithm for word unification which was independently found by several authors [Le72, Pl72, Si78, Si75]. Later, in section 5, we shall prove that the algorithm computes a disjoint and complete set of unifiers, for every word equation WE_0. For the moment we are only interested in the general structure.

The Basic Algorithm

For given word equation WE_0, a finitely branching search tree $\mathcal{T}_{LP}(WE_0)$ is generated, using non-deterministic transformation rules based on "variable-splitting" techniques. With $Succ(WE)$ we will denote the set of successors of the word equation WE under transformation. If WE has the form $uW_1 == vW_2$ (where u and v are constants or variables and W_1, W_2 are possibly empty words), then we say that WE has the "head" (u, v) and the *tail* $Tail(WE) := W_1 == W_2$. Expressions $(v \mapsto W)$ denote substitutions which simultaneously replace all occurrences of v by W if applied to any word equation.

Transformation:

(i) If WE has head (a, b) with two distinct constants, then $Succ(WE)$ is empty.

(ii) If WE has head (u, u) with two identical constants or variables, then $Succ(WE) := \{Tail(WE)\}$.

(iii) If WE has head (a, x) or (x, a) with one constant and one variable, then let $S_1^{LP} = (x \mapsto a)$, $S_2^{LP} = (x \mapsto ax)$. Now
$Succ(WE) := \{Tail(S_1^{LP}(WE)), Tail(S_2^{LP}(WE))\}$.

(iv) If WE has head (x, y) with two distinct variables, then let $S_3^{LP} = (y \mapsto x)$, $S_4^{LP} = (x \mapsto yx)$ and $S_5^{LP} = (y \mapsto xy)$. Now
$Succ(WE) := \{Tail(S_3^{LP}(WE)), Tail(S_4^{LP}(WE)), Tail(S_5^{LP}(WE))\}$.

Successor elements may have one or two empty sides. Every node labelled with a "word equation" with two empty sides is a successful leaf. Every node labelled with a "word equation" with exactly one empty side is a blind leaf.

In this version the algorithm defines a semi-decision procedure: it is straightforward to see that WE_0 has a solution iff $\mathcal{T}_{LP}(WE_0)$ has a successful leaf (see section 5).

The Explosion of the Data Size

In general, the basic algorithm does not terminate since the size of the word equations which are created via transformation may grow. As an example, consider the word equation $xyxy == axyxzb$. We may apply the transformation $WE \rightarrow Tail(S_2^{LP}(WE))$. The resulting word equation $xyaxy == axyaxzb$ has again head (x, a). After k iterations we obtain the word equation $xya^kxy == axya^kxzb$. In some cases, termination arguments may be given based on splitting techniques (see [LiSi75]). In general, however, there is no simple additional technique to decide solvability by deciding when an infinite branch can be cut.

A Decidable Subcase

In a particular case it is simple to obtain a terminating algorithm by means of loop-checking methods: a trivial inspection of the transformation rules shows that the size of word equations cannot grow if no variable occurs more than twice. Thus, in this case there is only a finite number of word equations which may occur in the search tree. Suppose that we have reached — at node ν_2 — a word equation which has occurred earlier in the same path, at node ν_1. In this case we may stop with failure: if any sequence of transformations leads to a successful leaf, starting from ν_2, then we may apply the same sequence starting from ν_1 and we will again find a successful leaf. This shows that we may ignore the subtree below ν_2 for matters of decidability. More generally we may stop when we have found a word equation WE_2 which is *isomorphic* to a predecessor WE_1 in the same path. This means that WE_2 may be obtained from WE_1 by a permutation of the variable alphabet and a permutation of the alphabet of constants.

With this pruning method we obtain a *finite* subtree $\mathcal{T}_{LP}^{fin}(WE_0)$ and thus a decision procedure: WE_0 has a unifier iff $\mathcal{T}_{LP}^{fin}(WE_0)$ has a successful leaf.

Generalized Equations

The observation that unifiability of word equations is decidable if variables occur at most twice does not solve the general problem. We cannot translate an arbitrary word equation into an equivalent word equation where every variable has at most two occurrences. But, with a more complex data type, it is in fact possible to get an artificial duality of variables. This is one of the important ideas behind the following concept.

Definition 1.1: A *generalized equation* is a quadrupel $GE = (BS,BD,Col,BE)$ with four entities:

(1) A finite set of *bases*: $BS = \{bs_1, ..., bs_N\}$, $N \geq 1$.
Every base is either a variable base or a constant base. Each constant base bs_i is associated with exactly one letter a in the alphabet $\mathcal{C}$, we say that bs_i has type a. Each variable base bs_i is paired off with exactly one other variable base $bs_j \in BS$; bs_i and bs_j are called *duals* of each other. Letters $x, y, z, \ldots$ denote variable bases. We write $\bar{x}$ for the dual of the variable base x.

(2) A finite set of *boundaries*: $BD = \{1, 2, \ldots, M\}$, $M \geq 1$.
Letters $i, j, k, \ldots$ denote boundaries. A pair (i, j) of boundaries with $i \leq j$ is called a *column* of GE. Columns (i, i) are called empty, columns $(i, i+1)$ are called indecomposable. For $i < j < k$ we say that boundary j is in (i, k).

(3) A column-function Col:
A function which assigns a column of GE to every base of BS such that $Col(bs_i)$ is indecomposable for every constant base $bs_i \in BS$ and that $Col(x) \neq Col(\bar{x})$ if x is a non-empty variable base[5]. For convenience, we introduce the two functions $Left$ and $Right$: if $Col(bs_i) = (j, k)$, then $Left(bs_i) = j$ and $Right(bs_i) = k$.

[5] i.e., a variable base with non-empty column.

(4) A finite set BE of *boundary equations*:
A boundary equations is a quadrupel of the form $(i, x, j, \bar{x})$ where i and j are boundaries, x and $\bar{x}$ are dual variable bases, i is in $Col(x)$ and j is in $Col(\bar{x})$. Symbols $E, E_1, ...$ denote boundary equations.

The role of parts (1)-(3) of the definition will become clear in a moment when we have given the definition of a unifier of a generalized equation and when we show how word equations are translated into sets of generalized equations. In order to understand the role of the boundary equations (4), more background is needed. For the moment, imagine that a boundary equation $(i, x, j, \bar{x})$ expresses that the position of i in x corresponds to the position of j in $\bar{x}$.

For readers which are familiar with Jaffar's [Ja90] corresponding definition let us point out two differences: by (2), boundaries are naturally ordered and our generalized equations are completed in the sense of [Ja90]. The condition $Col(x) \neq Col(\bar{x})$ for non-empty variable bases x may be regarded as a normalization condition in the sense of [Ma77]. It will guarantee later that the "carrier" of a non-trivial generalized equation and its dual never have the same position.

Definition 1.2: Every assignment S of non-empty words $S(i, i+1) \in (\mathcal{C} \cup \mathcal{V})^+$ to the indecomposable columns of GE has — via concatenation — a unique extension which assigns a (non-) empty word to every (non-) empty column of GE. We identify S with this extension. S is a *unifier* of GE if three conditions are satisfied:

(i) $S(Col(bs_i)) = a$ for every constant base bs_i of type a ($a \in \mathcal{C}$),

(ii) $S(Col(x)) = S(Col(\bar{x}))$ for all variable bases x of GE,

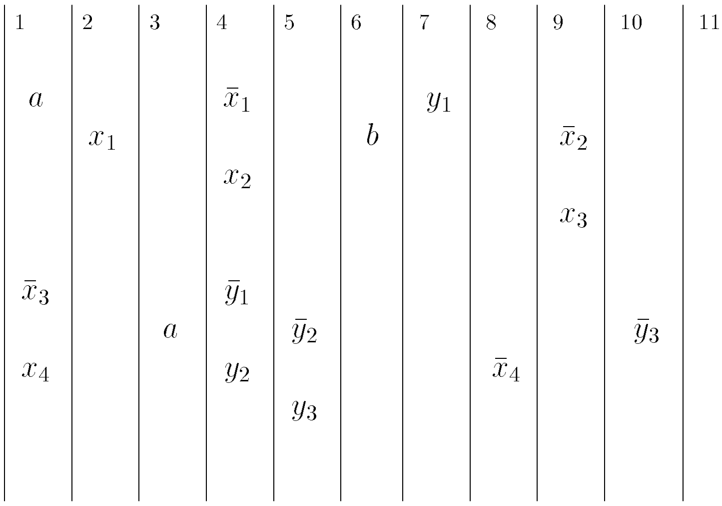
(iii) $S(Left(x), i) = S(Left(\bar{x}), j)$ for every boundary equation $(i, x, j, \bar{x})$ in GE.

The *index* of S is the number $|S(1, M)|$, where M is the maximal boundary and $|W|$ denotes the length of the word W. The *exponent of periodicity* of S is the maximal number e such that $S(Col(x))$ may be represented in the form $S(Col(x)) = UV^eW$, V non-empty, for a variable base x of GE. S is a *solution* of GE if $S(1, M) \in \mathcal{C}^+$.

Translation

The following example shows how word equations WE are translated into sets $\geq (WE)$ of generalized equations. We visualize one generalized equation which is assigned to the word equation WE_0 of the form $axxbyx == xayyxy$ with variables

x and y:



The vertical lines are the boundaries $1, 2, \ldots, 11$ of GE_0 which fix the relative extension of variables. GE_0 contains a certain "variant" of the left side $axxbyx$ of WE_0 in the upper part and a similar variant of the right side $xayyxy$ in the lower part.[6] In GE_0, multiple occurrences of the same symbol are formally distinguished. For this reason bases are introduced (horizontal lines). For the sake of simplicity we did not use distinct names for the two coefficient bases of type a in our figure. More important is the variable part. In word equations, a variable may have an arbitrary number of occurrences. In a generalized equation, every variable base has exactly two "dual" occurrences which are notationally distinguished by means of a bar "$\bar{\ }$". By 1.2 (ii), dual bases have to get the same value. With this dualism it will be possible to transform a generalized equation without any enlargement of the number of bases. Exactly for this reason the variable dualism is introduced.[7]

When we translate WE_0 we must store the information that all columns of GE_0 which correspond to the four occurrences of x in WE_0 have to get the same value. We may only use pairs of dual variable bases. But we may also identify distinct variable bases by writing them into the same column. In our example every solution of GE_0 will assign the same value to $\bar{x}_1$ and to x_2, similarly to $\bar{x}_2$ and x_3 and also to $\bar{x}_3$ and x_4 because they have the same column. In combination with the equality of dual bases this will ensure that the four "x"-columns $(2, 4)$, $(4, 6)$, $(9, 11)$ and $(1, 3)$ will get the same value under an arbitrary solution. The same holds for the "y"-columns $(7, 9)$, $(4, 5)$, $(5, 8)$ and $(10, 11)$.

The remaining elements of $\geq (WE_0)$, the set of all generalized equations corresponding to WE_0, only differ from the one given above in the relative position

[6]The vertical position is, however, irrelevant — it was just chosen for the sake of readability.

[7]Later we shall see that this does not mean that we have avoided the "explosion of the data size" – generalized equation may start growing in another part.

of bases. In order to preserve solvability the elements of $\geq (WE_0)$ must represent every possible distribution for the relative length of the bases. The formal definition of the translation algorithm is not difficult and therefore omitted. We refer to [Ja90].

The following lemma summarizes some properties of the translation which are trivial but become important later. If S is a solution of the word equation WE, the exponent of periodicity of S is the maximal number e such that $S(x)$ may be represented in the form $S(x) = UV^eW$, V non-empty, for a variable x of WE.

Lemma 1.3: *There exists an algorithm which computes for every word equation WE_0 a finite set $\geq (WE_0)$ of generalized equations with the following properties:*

(a) *WE_0 has a unifier with exponent of periodicity e if and only if some $GE \in \geq (WE_0)$ has a unifier with exponent of periodicity e.*

(b) *The elements of $\geq (WE_0)$ do not contain boundary equations. Every boundary is the right or left boundary of a base.*

(c) *For $GE \in \geq (WE_0)$, the number of bases of GE does not exceed $2l(WE_0)$, where $l(WE_0)$ is the notational length*[8] *of WE_0.*

As in [Ja90], a generalized equation GE is called *trivially true*, if all variable bases of GE are empty and if GE has a unifier. GE is *true*, if it is trivially true or if all constant bases of GE have the same type and GE has a unifier. (If all constant bases of GE have the same type, then unifiability reduces to a set of length restrictions which may be represented by an existential formula of first-order arithmetic without multiplication (Presburger arithmetic). The validity of such formulas is decidable, see [Coo72]). GE is *trivially false* if two constant bases of distinct type have the same column. GE is *false*, if it is trivially false or if the generalized equation GE^a which we get when we assign the same type $a \in \mathcal{C}$ to all constant bases has no unifier (in this case some inherent length restrictions cannot be satisfied). GE is trivial if it is either trivially true or trivially false.

Since empty variable bases cannot be involved in boundary equations it is trivial to decide the unifiability of GE if all variable bases are empty.

Lemma 1.4: *It is decidable whether a generalized equation is trivial (true, false).*

[8] i.e., the number of symbol occurrences.

2 Transformation — Notions

Suppose that *GE* is a non-trivial generalized equation. Let $l^\star$ denote the leftmost boundary among all left boundaries of non-empty variable bases. The *carrier* of *GE* is the largest[9] base among all variable bases with left boundary $l^\star$ (if there are several candidates, any may be chosen). The symbol x_c will be used to denote the carrier. The basic idea of the transformation procedure is to carry a part of the structure of $Col(x_c)$ to $Col(\bar{x}_c)$ and to erase a left part of *GE* afterwards. In general there are various ways how the structures of the two columns $Col(x_c)$ and $Col(\bar{x}_c)$ can be superposed and transformation is non-deterministic. In order to maintain unifiability downwards and upwards, all relevant information on identical subwords has to be preserved at a transformation step. As it turns out, it is possible to transport simultaneously the complete structure of $Col(x_c)$ up to a certain *critical boundary*. From now on, $(l^\star, r^\star)$ and $(l^{\star\star}, r^{\star\star})$ always denote the columns of x_c and $\bar{x}_c$ respectively.

Definition 2.1: The critical boundary of *GE* is the leftmost boundary among all left boundaries of variables bases y such $r^\star$ is in $Col(y)$, if such a base exists, and $r^\star$ in the other case. The symbol *cr* denotes the critical boundary.

Remark 2.2: In any non-trivial generalized equation *GE*, $l^\star < cr \leq l^{\star\star}$.

Up to x_c and $\bar{x}_c$, the bases and boundaries of *GE* will now be partitioned in three classes of superfluous objects, transport objects and fixed objects. The superfluous objects will be erased at a transformation step. The transport entites are carried to $Col(\bar{x}_c)$, the fixed entities keep their position. Roughly, the classification may be seen in the following figure, but details will become important later.

1	$l^\star$	cr	$r^\star$	M
superfluous bases/boundaries	transport bases/boundaries	transport or fixed bases/boundaries	fixed bases/boundaries	

Definition 2.3: The *transport bases* of *GE* are all bases $bs \neq x_c$ such that $l^\star \leq Left(bs) < cr$ and the empty bases with column (cr, cr). A base bs with $Col(bs) = (i, i)$, $i < l^\star$ is called *superfluous*. A base $bs \neq x_c, \bar{x}_c$ which is not superfluous and not a transport base is called a *fixed* base.

[9] i.e., the one with the largest right boundary.

Note that *all* bases bs with $Left(bs) < l^\star$ are necessarily empty, by definition of the carrier.

Remark 2.4: It will be frequently used that columns (i, j) of non-empty transport bases are always subcolumns of $(l^\star, r^\star)$ with $i < cr$. On the other hand, if (i, j) is the column of a fixed base, then $cr \leq i$ and $cr < j$.

Definition 2.5: A boundary i of GE is a *transport boundary* in three cases:

(i) if $l^\star < i \leq cr$,

(ii) if $cr < i < r^\star$ and $i = Right(x)$ where x is a transport base,

(iii) if $cr < i < r^\star$ and i occurs in a boundary equation $(i, x, j, \bar{x})$ or $(j, \bar{x}, i, x)$ where x is a transport base.

A boundary $i \leq l^\star$ is called *superfluous*. A boundary is *fixed* if it is neither superfluous nor a transport boundary.

When we carry the transport entities from $Col(x_c)$ to $Col(\bar{x}_c)$ we have to superpose the structures of both columns. The following definition excludes superpositions which are trivially wrong, contradicting information about equal subparts of the two columns which are encoded in boundary equations of the form $(i, x_c, j, \bar{x}_c)$ or $(j, \bar{x}_c, i, x_c)$.

Definition 2.6: Let $l^{\star tr}, (l^\star+1)^{tr}, ..., r^{\star tr}$ be a sequence of symbols not occurring in GE, representing a copy of all boundaries between $l^\star$ and $r^\star$. An *extended print* is a linear order $\preceq$ on the set[10] $BD \cup \{l^{\star tr}, (l^\star + 1)^{tr}, ..., r^{\star tr}\}$ satisfying the following conditions:

(i) $l^{\star tr} = l^{\star\star}, r^{\star tr} = r^{\star\star}$,

(ii) $\prec$ extends the natural order of BD and $k^{tr} \prec l^{tr}$ for $l^\star \leq k < l \leq r^\star$,

(iii) if $l^\star \leq i$ and $(i, j) = Col(bs_k)$ for a constant base, then i and j are consecutive with respect to $\preceq$. Similarly, if $l^\star \leq i < j \leq r^\star$ and $(i, j) = Col(bs_k)$ for a constant base, then also i^{tr} and j^{tr} are consecutive with respect to $\preceq$,

(iv) if $l^\star < i < r^\star$ and GE has a boundary equation $(i, x_c, j, \bar{x}_c)$ or $(j, \bar{x}_c, i, x_c)$, then $i^{tr} = j$,

(v) if x is a transport base, $\bar{x}$ a fixed base and if GE has a boundary equation $(i, x, j, \bar{x})$ or $(j, \bar{x}, i, x)$, then $i^{tr} = j$ iff $Left(x)^{tr} = Left(\bar{x})$ iff $Right(x)^{tr} = Right(\bar{x})$ (equalities with respect to $\preceq$).

[10] For pure formalists: since elements of $M := BD \cup \{l^{\star tr}, (l^\star+1)^{tr}, ..., r^{\star tr}\}$ may be identified with respect to $\preceq$, this linear order is formally an order on a partition of M.

A *print* is the restriction of an extended print to the set

$$BD' = \{cr, cr+1, ..., M\} \cup \{i_1^{tr}, \ldots, i_r^{tr}\}$$

where $\{i_1, ..., i_r\}$ is the set of all *transport* boundaries of *GE*.

Lemma 2.7: *The set of all prints of GE is finite and may effectively be computed.*

Definition 2.8: The boundary equations of the form $E = (i, x, j, \bar{x}), x \neq x_c, \bar{x}_c$, are called *standard equations*. The *natural image* E' of E is the quadrupel which we get from E, replacing the entry i (entry j) by i^{tr} (by j^{tr}) if x (resp. $\bar{x}$) is a transport base and leaving it unchanged in the other case. Of course both bases x and $\bar{x}$ may be transport bases in which case both i and j have to be replaced to obtain the natural image. If $\preceq$ is a print, E' is *degenerate* with respect to $\preceq$ if the two boundaries occurring in E' coincide with respect to $\preceq$.

Remark 2.9: According to 2.4 and 2.6 (ii), $i^{tr} \prec cr^{tr}$ if i is the left boundary of a non-empty transport base, for any print $\preceq$.

3 Transformation — Algorithm

The following Transformation procedure assigns a finite set *Transf*(*GE*) (of non-false generalized equations) to every non-trivial generalized equation *GE*:

• **Transformation** of *GE* = (*BS*,*BD*,*Col*,*BE*).

Step 1: Compute the set of all prints for *GE*.

Step 2: For every print $\preceq$ of *GE* let $GE_{\preceq}$ be the generalized equation (BS', BD', Col', BE') with components as defined below. $GE_{\preceq}$ is — modulo a trivial renaming of boundaries — a generalized equation. *Transf1*(*GE*) is the set of all resulting structures $GE_{\preceq}$.

3.1 BS' is the set of all non-superfluous bases of *GE*.

3.2 BD' contains
3.2.1: all boundaries i, $cr \leq i$,

3.2.2: a new boundary i^{tr} for every transport boundary i of *GE*.

3.3 Col' is defined as follows:

3.3.1: $Col'(bs) = Col(bs)$ if bs is a fixxed base of GE, with the exception described in 3.3.4,

3.3.2: $Col'(bs) = (Left(bs)^{tr}, Right(bs)^{tr})$ if bs is a transport base of GE, with the exception described in 3.3.4,

3.3.3: $Col'(x_c) = (cr, r^\star)$ and $Col'(\bar{x}_c) = (cr^{tr}, r^{\star\star})$,

3.3.4: If x is a non-empty variable base and if $Col'(x) = Col'(\bar{x})$ according to 3.3.1 and 3.3.2, then this value is corrected: $Col'(x) = Col'(\bar{x}) = (Right'(x), Right'(x))$.

3.4: BE' contains:

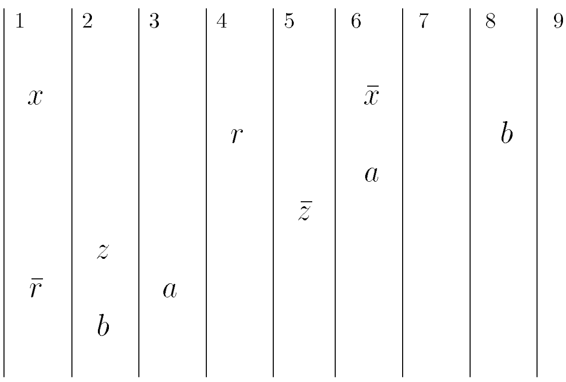
3.4.1: every non-degenerate natural image E' of a standard equation E of GE,

3.4.2: all boundary equations of GE of the form $(i, x_c, j, \bar{x}_c)$ or $(j, \bar{x}_c, i, x_c)$ for $cr < i$,

3.4.3: a new equation $E' = (i, x_c, i^{tr}, \bar{x}_c)$ for every transport boundary $i > cr$.

Step 3: Erase all false elements of *Transf1*(GE). In the remaining structures, rename boundaries using natural numbers $1, \ldots, M'$, according to their order with respect to $\prec$. The resulting set is *Transf*(GE).

Example 3.5: To get a better picture of the algorithm it is useful to distinguish three levels of growing complication. In the first situation the boundary $r^\star$ is not inside the column of another base. Then $cr = r^\star$, the complete structure of $Col(x_c)$ is transported to $Col(\bar{x}_c)$ and $x_c, \bar{x}_c$ become empty bases. The transformation does not introduce any new boundary equation. In the following example, x is the carrier of GE, $r^\star = cr = 4$.



$$(2, x, 7, \bar{x}), (3, z, 6, \bar{z})$$

The first boundary equation is used to determine the new position 7 of 2^{tr} and erased afterwards. Here is one element of *Transf1*(GE). It is the only successor in this case — the empty variable bases x and $\bar{x}$ are omitted. Boundary names

1^{tr} and 4^{tr} are only added in order to facilitate the reading.

4	5	6,1^{tr}	7,2^{tr}	3^{tr}	8,4^{tr}	9
r					b	
		a				
	$\bar{z}$					
			z			
		$\bar{r}$		a		
			b			

$$(3^{tr}, z, 6, \bar{z})$$

The corresponding element of *Transf*(GE) woud be obtained using standardized boundary names $1, \ldots, 7$.

The second typical situation occurs if there exists a base y, $Left(y) < r^\star$, which exceeds the carrier, but if there is no transport base whose right boundary falls into the column $(cr, r^\star)$. The subpart of $Col(x_c)$ up to the critical boundary is transported and cr becomes the new initial boundary. As a consequence of the second condition, no new boundary equations are introduced. The following generalized equation GE is an example — the carrier is x, $cr = 3$:

1	2	3	4	5	6	7	8	9
x					$\bar{x}$			
			r					
					a			
				$\bar{z}$			$\bar{y}$	
	z							
$\bar{r}$		a						
	b							
		y						

$$(2, x, 7, \bar{x}), (3, z, 6, \bar{z})$$

Again *Transf1*(GE) has only one element:

3	4	5	6,1^{tr}	7,2^{tr}	3^{tr}	8,4^{tr}	9
x					$\bar{x}$		
	r						
			a				
		$\bar{z}$				$\bar{y}$	
				z			
a			$\bar{r}$				
				b			
y							

$$(3^{tr}, z, 6, \bar{z})$$

In the third and most complex situation we have a transport boundary between cr and $r^\star$. In our example it is the boundary $4 = Right(s)$.

1	2	3	4	5	6	7	8	9	10
x						$\bar{x}$			
				r					
						a			
					$\bar{z}$			$\bar{y}$	
	z								
$\bar{r}$			a						
	b			$\bar{s}$					
		y							
	s								

$$(2, x, 8, \bar{x}), (3, z, 7, \bar{z})$$

In this case we need a new boundary equation $(4, x, 4^{tr}, \bar{x})$ after the transformation in order to store all information on identical subcolumns:

3	4	5	6	$7{,}1^{tr}$	$8{,}2^{tr}$	3^{tr}	4^{tr}	$9{,}5^{tr}$	10
x						$\bar{x}$			
		r							
				a					
			$\bar{z}$					$\bar{y}$	
					z				
	a			$\bar{r}$					
		$\bar{s}$			b				
y									
					s				

$$(3^{tr}, z, 7, \bar{z}), (4, x, 4^{tr}, \bar{x})$$

Remark 3.6: (a) Suppose that $GE' = GE_{\prec} \in Transf(GE)$. If the natural image of a boundary equation $E = (i, x, j, \bar{x})$ of GE is degenerate in GE', then we are in the case described in 3.3.4 and vice versa, by 2.6 (v).
(b) Note that the boundary equations of GE which are erased in GE' are exactly the degenerate boundary equations and the boundary equations of the form $(i, x_c, j, \bar{x}_c)$ or $(j, \bar{x}_c, i, x_c)$ where $i \leq cr$.

As a matter of fact, functions $Left'$ and $Right'$ may be defined in 3.3. In the following it is convenient to distinguish occurrences of the same base in GE and in $GE' := GE_{\prec}$ if their position is changed. We allow to write bs^{tr} for an occurrence of the transport base bs in GE', similarly we often write x'_c and $\bar{x}'_c$ for occurrences of x_c and $\bar{x}_c$ in GE'. We also allow to write $l^{\star tr}$ ($r^{\star tr}$) for $l^{\star\star}$ ($r^{\star\star}$).

Theorem 3.7: *For every print $\preceq$, the structure $GE_{\preceq}$ is - modulo renaming of boundaries - a generalized equation. The number N' of bases of $GE_{\preceq}$ does not exceed the number N of bases of GE.*

Proof: The only nontrivial part is to show that all elements E' of BE' are in fact boundary equations of GE'. If E' is the (non-degenerate) natural image of $E = (i, x, j, \bar{x})$, suppose, for example, that x is a transport base while $\bar{x}$ is fixed. Thus $E' = (i^{tr}, x^{tr}, j, \bar{x})$. Clearly j is in $Col'(\bar{x}) = Col(\bar{x})$, by 2.6 (ii) since E is a boundary equation. Again by condition (ii) of 2.6, i^{tr} is in $Col'(x^{tr})$ and E' is in fact a boundary equation. If E' has one of the types of 3.4.2 or 3.4.3, then it follows from $Col'(x'_c) = (cr, r^\star)$ and $Col'(\bar{x}'_c) = (cr^{tr}, r^{\star\star})$ that E' is a boundary equation, using 2.6 (ii) and, for 3.4.3, 2.6 (iv) to get $cr^{tr} \prec j$. □

Before we continue the formal analysis of the transformation procedure we have to add a general remark: it is clear that the non-false elements of *Transf1*(GE) and the elements of *Transf*(GE) are identical modulo a trivial standardization of boundary names. We want to establish various results concerning *Transf*(GE). For the proofs it is much more convenient to use the corresponding structures of *Transf1*(GE). Thus we shall henceforth ignore this notational distinction.

Theorem 3.8: (a) *If GE has a unifier S with index I and exponent of periodicity e, then Transf(GE) has an element GE' which has a unifier S' with index $I' < I$ and exponent of periodicity $e' \leq e$.*
(b) *If an element of Transf(GE) has a unifier, then GE is unifiable.*

Proof: (a) Let S be a unifier of GE. For all i in $(l^\star, r^\star)$ and all j in $(l^{\star\star}, r^{\star\star})$ define

- $i^{tr} \prec j$ iff $S(l^\star, i)$ is a proper prefix of $S(l^{\star\star}, j)$,
- $i^{tr} = j$ iff $S(l^\star, i) = S(l^{\star\star}, j)$,
- $j \prec i^{tr}$ iff $S(l^{\star\star}, j)$ is a proper prefix of $S(l^\star, i)$.

Obviously $\preceq$ determines a unique extended print for GE. We may use the same symbol $\preceq$ for the corresponding print. We show that $GE' = GE_{\preceq}$ has a unifier S'. We have to consider the indecomposable columns of GE'. All such columns of the form $(i, i+1)$ which have an empty intersection with $(l^{\star\star}, r^{\star\star})$ are columns of GE. We define $S'(i, i+1) = S(i, i+1)$. There are at most the following four types of indecomposable subcolumns of $(l^{\star\star}, r^{\star\star})$ in GE':

(i) $(j, j+1)$ where $l^{\star\star} \leq j < r^{\star\star}$,
(ii) (i^{tr}, j) where i in $(l^\star, r^\star)$ is a transport boundary and $l^{\star\star} < j \leq r^{\star\star}$,
(iii) (j, i^{tr}) where i in $(l^\star, r^\star)$ is a transport boundary and $l^{\star\star} \leq j < r^{\star\star}$,
(iv) (i^{tr}, j^{tr}) where $l^\star < i < j < r^\star$ and both i and j are transport boundaries.

We define $S'(j, j+1) = S(j, j+1)$ in case (i), $S'(i^{tr}, j) = S(l^\star, i)^{-1}S(l^{\star\star}, j)$ in case (ii), $S'(j, i^{tr}) = S(l^{\star\star}, j)^{-1}S(l^\star, i)$ in case (iii) and $S'(i^{tr}, j^{tr}) = S(i, j)$ in case (iv). The following claim may be proved by induction on the number of indecomposable subcolumns. The technical proof is omitted:

Claim: For all common columns (i, j) of GE and GE': $S'(i, j) = S(i, j)$; for all columns of the form (i^{tr}, j^{tr}) of GE': $S'(i^{tr}, j^{tr}) = S(i, j)$.

With 3.3.1 and 3.3.2 it follows immediately that $S'(Col'(bs_i)) = S(Col(bs_i)) = a$ for all constant bases bs_i of GE' of type a, and that $S'(Col'(x)) = S'(Col'(\bar{x}))$ for all fixed and transport bases x of GE. Moreover, by the claim we get

$$S'(cr, r^\star) = S(cr, r^\star) = S'(cr^{tr}, r^{\star tr}) = S'(cr^{tr}, r^{\star\star})$$

and therefore $S'(Col'(x'_c)) = S'(Col'(\bar{x}'_c))$. Thus S' satisfies conditions (i) and (ii) of definition 1.2.

Suppose now that E' is a boundary equation of GE'. Recall 3.4. We have to show that S' satisfies condition (iii) of 1.2 for E'. In the first case E' is the (non-degenerate) natural image of $E = (i, x, j, \bar{x}) \in BE$. Assume first that x is a transport base while $\bar{x}$ is fixed. Thus $E' = (i^{tr}, x^{tr}, j, \bar{x})$. By 3.3.2, the claim, 1.2 (iii) and 3.3.1 we have

$$\begin{aligned} S'(Left'(x^{tr}), i^{tr}) &= S'(Left(x)^{tr}, i^{tr}) = S(Left(x), i) \\ &= S(Left(\bar{x}), j) = S'(Left'(\bar{x}), j). \end{aligned}$$

Similarly condition (iii) of definition 1.2 may be verified for the remaining subcases.

In the second case E' has the form $(i, x'_c, j, \bar{x}'_c)$ or $(j, \bar{x}'_c, i, x'_c)$ where $cr < i$ and where $E = (i, x_c, j, \bar{x}_c)$ or $(j, \bar{x}_c, i, x_c) \in BE$. Recall 3.3.3. Here $i^{tr} = j$, by 2.6 (iv). By the claim,

$$S'(cr, i) = S(cr, i) = S'(cr^{tr}, i^{tr}) = S'(cr^{tr}, j)$$

as demanded.

In the third case, $E' = (i, x'_c, i^{tr}, \bar{x}'_c)$ is new and $cr < i$, by 3.4.3. By the claim,

$$S'(cr^{tr}, i^{tr}) = S(cr, i) = S'(cr, i).$$

Thus condition (iii) of definition 1.2 is always satisfied.

We have shown that S' is a unifier of GE'. Obviously the index of S' is strictly smaller than the index of S since $l^\star < cr$. The exponent of periodicity e' of S' does not exceed the exponent of periodicity e of S since $S'(Col'(x))$ is always a suffix of $S(Col(x))$, for any variables base x of GE'.

(b) Assume now that S' is a unifier of the structure $GE' = GE_{\preceq}$. All indecomposable columns $(i, i+1)$ of GE with $cr \leq i$ are columns of GE'. We define $S(i, i+1) = S'(i, i+1)$ in these cases. For the indecomposable columns $(i, i+1)$ of GE with $i+1 \leq l^\star$ we define $S(i, i+1) = a$ if $(i, i+1) = Col(bs_i)$ for a constant base of type a (remember that GE is nontrivial, hence a is unique). In the other cases, an arbitrary non-empty word $S(i, i+1)$ may be assigned to $(i, i+1)$ for $i < l^\star$. The indecomposable columns $(l^\star, l^\star+1)$,...,$(cr-1, cr)$ of GE have images $(l^{\star\star}, (l^\star+1)^{tr})$,...,$((cr-1)^{tr}, cr^{tr})$ in GE' (3.2.2). We define $S(l^\star, l^\star+1) = S'(l^{\star\star}, (l^\star+1)^{tr})$,...,$S(cr-1, cr) = S'((cr-1)^{tr}, cr^{tr})$. It follows immediately that $S'(i, j) = S(i, j)$ for all common columns (i, j) of GE' and GE. Furthermore, for all $l^\star \leq i < j \leq cr$ we have $S(i, j) = S'(i^{tr}, j^{tr})$. Since $S'(cr, j) = S'(cr^{tr}, j^{tr})$ for all transport boundaries $j > cr$ of GE (compare 3.4.3, 3.3.3 and 1.2 (ii)) it is possible to prove that $S(i, j) = S'(i^{tr}, j^{tr})$ for arbitrary transport boundaries $i < j$ of GE. Similarly as in (a) it is now easy to verify that S is in fact a unifier of GE. $\square$

For word equation WE_0 the tree $\mathcal{T}_{Mak}(WE_0)$ has WE_0 as top element and $\geq (WE_0)$ as first level (compare lemma 1.3). For every $GE \in \geq (WE_0)$, the downward tree is the unordered, finitely branching tree which results form iterated transformation.

Corollary 3.9: *WE_0 has a unifier if and only if $\mathcal{T}_{Mak}(WE_0)$ has a node which is labelled with a trivially true generalized equation.*

Proof: "*only if*": Suppose that WE_0 has a unifier. By 1.3 some element of $\geq (GE)$ has a unifier, of index I, say. By 3.8 there is a downward branch in $\mathcal{T}_{Mak}(WE_0)$ labelled with unifiable generalized equations where the index decreases at every step. Since the index is non-negative, the length of this branch cannot exceed I. The generalized equation GE which labels the last node cannot have a non-empty variable base since otherwise the transformation algorithm would apply again. The rest is obvious.
"*if*": by Lemma 1.3 (a) and Theorem 3.8, with a trivial induction. $\square$

4 Proper Generalized Equations and Termination

In most cases $\mathcal{T}_{Mak}(WE_0)$ will be an infinite tree. The previous results show that it offers a correct and complete semi-decision procedure. In order to obtain termination we need two additional arguments. The first is simple: similarly as for the basic algorithm we may eliminate some branches by means of loop checking methods. Let us call two generalized equations GE and GE' *isomorphic* if the latter equation differs from the former only by means of a "consistent renaming of bases", i.e., if GE' may be obtained from GE by a bijection between the sets of bases which maps coefficient bases of the same type again into coefficient bases of the same type. Obviously we may stop a branch if we have found a generalized equation GE' which is isomorphic to a predecessor GE in the same branch.

Even with this pruning method we will get an infinite tree in general: generalized equations may have an arbitrary number of boundaries and boundary equations, thus there is an infinite number of non-isomorphic generalized equations even for a fixed number of bases. The second argument is much more difficult and may be regarded as the main idea behind Makanin's decidability result. It is based on the following theorem which was first proved in Bulitko [Bul]. A second version occurs in [Mak], recently Kościelski and Pacholski [KoP] found the bound which is used in the following formulation:

Theorem: *Let WE_0 be a word equation with notational length l. If WE_0 has any solution, then it has a solution where the exponent of periodicity e satisfies*

$$e \leq epp^{max}(l) = 2^{1.07l}.$$

Recall that translation into generalized equations and subsequent transformation steps preserve solvability under a given upper bound for the exponent of periodicity in the downward direction (1.3 and 3.8). Thus, for a mere decision procedure it suffices to consider in $\mathcal{T}_{Mak}(WE_0)$ the generalized equations which possibly have a "tame" solution, i.e. a solution where the exponent of periodicity does not exceed $epp^{max}(l)$.

Makanin's main technical result — now adapted to the present terminology — was the proof that for the generalized equations GE *which are generated via transformation* the number of boundaries determines a lower bound for the exponent of periodicity of an arbitrary solution. If the number of boundaries is very large, GE cannot have a tame solution and may be treated as a failure leaf.

A similar result for arbitrary generalized equations can *not* be proved. The relevant properties which guarantee that a lower bound for the exponent of periodicity in terms of the number of boundaries may be given are captured by the concept of a *proper* generalized equation. This notion will be introduced below. We will also show that all generalized equations in $\mathcal{T}_{Mak}(WE_0)$ are proper.

Let us continue with the decidability argument. Note that for all generalized equations in $\mathcal{T}_{Mak}(WE_0)$ the number N' of bases does not exceed the number $N = 2l(WE_0)$ where $l(WE_0)$ is the notational length of WE_0 (1.3 (c) and 3.7).

Theorem: *There exists a recursive function $NBD^{max}(N, b)$ such that every* *proper generalized equation with $N' \leq N$ bases and $M' \geq NBD^{max}(N, b)$ boundaries has only solutions S where the exponent of periodicity exceeds b.*

This is essentially Jaffar's Main Lemma ([Ja90], pg. 75). Now obviously the number of nonisomorphic proper generalized equations with N' bases and M' boundaries, where $N' \leq N$ and $M' \leq NBD^{max}(N, epp^{max}(l))$, is finite. Thus there is only a finite number of generalized equations to consider. Summarizing we arrive at the following

First Decision Procedure

Suppose the word equation WE_0 of length l is given. Let $epp^{max}(l)$ be the bound given in the theorem of Bulitko, Makanin and Kościelski-Pacholski. Translate WE_0 into $\geq (WE_0)$, erasing false generalized equations. Iterate transformation. A node labelled with the generalized equation GE is a leaf in the following cases:

- GE is trivial: since GE is non-false, it is trivially true and solvable (success).
- GE is isomorphic to a predecessor equation (failure).
- If GE has $M' \geq NBD^{max}(N, epp^{max}(l))$ boundaries, where $N = 2l$ (failure).

Let $\mathcal{T}^{fin}_{Mak}(WE_0)$ be the resulting tree. It is finitely branching (2.7) and every path is of finite length. Thus $\mathcal{T}^{fin}_{Mak}(WE_0)$ is finite. WE_0 is unifiable if and only if $\mathcal{T}^{fin}_{Mak}(WE_0)$ has a leaf which is labelled with a trivially true generalized equation GE.

Proper Generalized Equations

If $E = (i, x, j, \bar{x})$ is a boundary equation of the generalized equation GE, then it is obviously possible to replace it by $E^{-1} = (j, \bar{x}, i, x)$ without affecting solvability. We shall say that *both* E and E^{-1} are *oriented versions* of E and *both* are oriented boundary equations of GE in this case. Throughout this subsection $GE = (BS, BD, Col, BE)$ denotes a nontrivial generalized equation. If not mentioned otherwise, all bases, boundaries and boundary equations are always from GE.

Definition 4.1: A *chain* of GE is a sequence $\pi = E_1, E_2, \ldots, E_m, w$ $(m > 0)$ where the E_l are oriented boundary equations of the form $(i_l, x_l, i_{l+1}, \bar{x}_l)$[11] $(1 \leq$

[11] Thus, the second boundary of E_l and the first boundary of E_{l+1} must always be identical.

Figure 1: A "domino-tower".

$l \leq m$) and w is a *witness*, i.e. a base $w = bs_j$ with $Right(w) = i_{m+1}$. The *leading base* of π is x_1.

Definition 4.2: Let $\pi = E_1, \ldots, E_m, w$ be a chain, suppose that E_l has the form $(i_l, x_l, i_{l+1}, \bar{x}_l)$ $(1 \leq l \leq m)$. To π we assign the word $\chi(\pi)$ of m symbols s_j in the alphabet $\{>, =, <\}$ defined by $Left(\bar{x}_j)\ s_j\ Left(x_{j+1})$ $(1 \leq j < m)$ and $Left(\bar{x}_m)\ s_m\ Left(w)$. The chain π is called *convex* if $\chi(\pi) \in \{>, =\}^\star \circ \{<, =\}^\star$. A convex chain π is *adaptive* if $\chi(\pi) \in \{<, =\}^\star$.

For every unifier of *GE*, the values of the variable bases occurring in a convex chain π may be arranged to a "domino-tower" of the form indicated in figure 1. Here parts which have vertical contact are identical. If π is adaptive, then the upper part — the part which has a frontier growing leftwards — is empty. This means that we may put any additional "pair of stones" on top of the given tower without destroying the convex form.

Later we shall consider sequences $\pi' = E'_1, ..., E'_m, w'$ where the E'_l are boundary equations of $GE' = GE_{\preceq} \in Transf(GE)$ and w' is a base of GE'. Then $\chi'(\pi') \in \{\succ, =, \prec\}^m$ is defined accordingly.

Definition 4.3: The *left* (*right*) boundaries of *GE* are the boundaries i such that $Left(bs_j) = i$ ($Right(bs_j) = i$) for a base bs_j of *GE*. The *involved* boundaries

Figure 2:

are the boundaries i which occur in an oriented boundary equation $(i, x, j, \bar{x})$ of *GE*. A boundary i of *GE* is *abandoned* if it is neither left nor right nor involved. In the following $LB(GE)$ denotes the number of left boundaries of non-empty bases of *GE* and $AB(GE)$ the number of abandoned boundaries of *GE*.

Definition 4.4: The generalized equation *GE* with N bases and M boundaries is *proper*, if the following two conditions are satisfied:
(i) $LB(GE) + AB(GE) \leq N$.
(ii) For every boundary equation E of *GE* there exists a convex chain $E_1, E_2, ..., E_m, w$ where E_1 is an oriented version of E.

Let us briefly sketch the argument which shows that for proper generalized equations with $N' \leq N$ bases the number of boundaries determines a lower bound for the exponent of periodicity of an arbitrary solution. Condition (i) trivially implies that proper generalized equations with a large number of boundaries have a large number of boundary equations, for fixed number of bases. Condition (ii) implies that these boundary equations may be ordered to *long* convex chains π such that $\chi(\pi)$ contains a large number of symbols ">" or "<" (this is non-trivial). Such chains show that some solution component $S(Col(x_l))$ (here x_l is a variables base occurring in π) may be arranged to a high "domino-tower", as indicated in figure 2. But such arrangements are only possible if $S(Col(x_l))$ has a large number of periodical, consecutive repetitions of the same nonempty subword. For details we refer to [Ma77, Ja90].

By 1.3 (b), all generalized equations in $\geq (WE_0)$ are proper, for any word equation WE_0. We shall prove the following theorem:

Theorem 4.5: *If GE is a non-trivial proper generalized equation, then all* *generalized equations in Transf(GE) are proper.*

It follows that all generalized equations occurring in $\mathcal{T}_{Mak}(WE_0)$ are proper. The proof of theorem 4.5 will be divided two parts. We shall first show that transformation preserves condition (i) of definition 4.4. In the appendix it will be shown that transformation also preserves condition (ii).

Definition 4.6: A convex chain $E_1, ..., E_m, w$ of GE is *clean*, if it does not contain a subsequence E_i, E_{i+1} of the form $(i, x, j, \bar{x}), (j, \bar{x}, i, x)$.

Lemma 4.7: *Suppose that GE is a proper. If $E \in BE$, then there exists a clean convex chain $E_1, E_2, ..., E_m, w$ such that E_1 is an oriented version of E.*

Proof: Since GE is proper there exists a convex chain $E_1, ..., E_m, w$ starting with E or E^{-1}. We show that every subsequence E_l, E_{l+1} of the form $(i, x, j, \bar{x}), (j, \bar{x}, i, x)$ may be replaced by a shorter sequence. The result follows then by induction.

Case 1, $l = 1$: Note that $E_2 = E_1^{-1}$. The chain $E_2, E_3, ..., E_m, w$ is again convex and starts with E or E^{-1}.

Case 2: $l + 1 = m$: The case $m = 2$ reduces to case 1, thus let $m > 2$ and $E_{m-2} = (k, y, i, \bar{y})$. Now $E_1, ..., E_{m-2}, w$ is a chain since $Right(w) = i$ and thus w is a witness for $E_1, ..., E_{m-2}$. Convexity is trivial for $m = 3$, or if $\chi(E_1, \ldots, E_{m-3}, E_{m-2})$ does not contain a symbol "$<$". If $m > 3$ and if $\chi(E_1, \ldots, E_{m-3}, E_{m-2})$ contains a symbol "$<$", then $Left(\bar{y}) \leq Left(x) \leq Left(w)$, by convexity of π.

Case 3: $1 < l < m - 1$: In this case $E_1, ..., E_m$ has a subsequence

$$\pi = (k, y, i, \bar{y})(i, x, j, \bar{x})(j, \bar{x}, i, x)(i, z, h, \bar{z}).$$

We replace it by $\pi' = (k, y, i, \bar{y})(i, z, h, \bar{z})$. The new chain is convex (it is trivial to see that in a convex "domino-tower" some "stones" may be omitted without destroying the convex form). $\square$

Corollary 4.8: *A clean convex chain π of GE does not have a subsequence of the form $(i, x_c, j, \bar{x}_c), (j, \bar{x}_c, k, x_c)$ or $(j, \bar{x}_c, i, x_c), (i, x_c, k, \bar{x}_c)$.*

Proof: Since GE is non-false this would imply that $k = i$ resp. $k = j$. $\square$

Of course this property does not only hold for the carrier x_c, but also for arbitrary bases. But we need this corollary only in the present form.

Lemma 4.9: *Suppose that GE is proper. Then every $GE' \in Transf(GE)$ satisfies condition (i) of definition 4.4.*

Proof: Remember that the boundaries of GE' are $cr, cr+1, \ldots, M$ and the boundaries of the form i^{tr}, where $l^\star < i < r^\star$ is a transport boundary of GE. Assume first, for simplicity, that the situation of 3.3.4 does not occur. We show that $LB(GE') \leq LB(GE)$: let $Cl(i)$ ($Cl'(i)$) be the set of all non-empty bases bs with left boundary i in GE (GE'). Thus we want to show that the number of non-empty classes of GE' does not exceed the number of non-empty classes of GE. If $x \in Cl(i)$ is a fixed (transport) base, then all elements of this class are fixed (transport) bases, with the possible exception $\bar{x}_c$ (x_c). Thus distinct non-empty classes of GE' have distinct non-empty parent classes in GE with the possible exception $Cl'(cr)$, $Cl'(cr^{tr})$ and the inequality is trivial if $cr = r^\star$ and x'_c, $\bar{x}'_c$ are empty. In the other case, $Cl(cr) \neq \emptyset$ and the members of $Cl(l^\star)$ are distributed over $Cl'(cr)$ (element x'_c) and $Cl'(l^{\star\star})$ (all others). Thus three non-empty classes of GE — $Cl(l^\star)$, $Cl(cr)$ and $Cl(l^{\star\star})$ — become two non-empty classes $Cl'(cr)$, $Cl'(l^{\star\star})$ of GE'. On the other hand, we have at most one new non-empty class, namely $Cl'(cr^{tr})$. Thus in fact $LB(GE') \leq LB(GE)$.
Let us now consider the abandoned boundaries. We shall prove the following claims:

Claim 1: If $cr < i$ is a left or right boundary of GE, $i \neq l^{\star\star}$, then i is left, right or involved in GE'.

Claim 2: If i is a transport boundary of GE and i is left, right or involved in GE, then i^{tr} is left, right or involved in GE'.

Claim 3: If $cr < i$ is neither left nor right in GE, but involved in a boundary equation E of GE, then i is left, right or involved in GE'.

Proofs are given below. Since cr is left in GE' claims 1-3 show that GE' may only have the following abandoned boundaries beside $l^{\star\star}$: some boundaries $i \in BD$, $cr < i$, but then i was already abandoned in GE, moreover some boundaries of the form i^{tr}, but then $l^\star < i < cr$, i was abandoned in GE and $i \notin BD'$. Thus, beside $l^{\star\star}$, distinct abandoned boundaries of GE' have distinct abandoned parents in GE. This shows that $AB(GE') \leq AB(GE)+1$, and if $AB(GE') = AB(GE)+1$, then $l^{\star\star}$ is abandoned in GE'. But in the latter case $cr \neq l^{\star\star}$, $Cl(l^\star) = \{x_c\}$ and $Cl(l^{\star\star}) = \{\bar{x}_c\}$. Then $Cl'(l^{\star\star})$ is empty and $LB(GE') \leq LB(GE) - 1$. Thus $LB(GE') + AB(GE') \leq LB(GE) + AB(GE) \leq N$.

Proof of claim 1: If $l^{\star\star} \neq i = Left(x)$, then x is fixed and $Left'(x) = i$ holds in GE'. If $i = Right(x)$ and x is fixed, then $i = Right'(x)$ holds in GE'. If $i = Right(x)$ and x is a transport base, then i is involved in GE', by 2.5 (ii) and 3.4.3. If $i = Right(x_c)$, then $i = Right'(x'_c)$ in GE', if $i = Right(\bar{x}_c)$, then

$i = Right'(\bar{x}'_c)$ in GE'.

Proof of claim 2: This is clear if $i = cr$ (3.3.3). If $cr < i$, then i^{tr} is involved in GE' by 3.4.3. Suppose now that $l^\star < i < cr$. If i is the left or right boundary of a base x, then x is a transport base and i^{tr} is a left or right boundary in GE'. Thus assume that i is involved in GE. If i occurs in $(i, x, j, \bar{x})$ or $(j, \bar{x}, i, x)$, then x is a transport base or $x = x_c$. In the first case i^{tr} is involved in GE', by 2.8 and 3.4.1. Thus assume that $(i, x_c, j, \bar{x}_c)$ or $(j, \bar{x}_c, i, x_c) \in BE$. Note that $cr < j \neq l^{\star\star}$. If j is left or right in GE, then j is left, right or involved in GE', by claim 1. But $i^{tr} = j$, by 2.6 (iv). Thus it remains to consider the case where both i and j are neither left nor right. Since GE is proper we have a clean convex chain $(i, x_c, j, \bar{x}_c), (j, y, k, \bar{y}), \ldots$, (case 1) or $(j, \bar{x}_c, i, x_c), (i, y, k, \bar{y}), \ldots$ (case 2) of length $m > 1$ in GE (for length 1, i or j would be the right boundary of the witness base). In case 1, $y \neq \bar{x}_c$, by 4.8. If y is fixed or $y = x_c$, then $i^{tr} = j$ (2.6 (iv)) is involved in GE', by 2.8, 3.4.1 or 3.4.2. If y is a transport base, then j is a transport boundary, by 2.5 (iii) and $(j, x'_c, j^{tr}, \bar{x}'_c) \in BE'$, by 3.4.3, thus i^{tr} is involved in GE'. In case 2, the subcases $y = x_c$, y fixed and $y = \bar{x}_c$ can be excluded, by 4.8 or since $i < cr$. Thus y is a transport base and i^{tr} is involved in GE', by 2.8 and 3.4.1.

Proof of claim 3: Assume that i occurs in $(i, x, j, \bar{x})$ or in $(j, \bar{x}, i, x) \in BE$. If x is fixed, then i is involved in GE' by 2.8 and 3.4.1. If x is a transport base, then i is involved in GE' by 2.5 (iii) and 3.4.3. If i occurs in $(i, x_c, j, \bar{x}_c) \in BE$, then i is involved in GE' by 3.4.2. If i occurs in $(i, \bar{x}_c, j, x_c) \in BE$ for $cr < j$, then $(i, \bar{x}'_c, j, x'_c) \in BE'$, by 3.4.2. If $j \leq cr$, then j is a transport boundary, by 2.5 (i) and involved. By claim 2, j^{tr} is left, right or involved in GE'. But $j^{tr} = i$, by 2.6 (iv).

Assume now that the situation of 3.3.4 happens to be true for the base x of GE. Then exactly one of the bases x and $\bar{x}$ — x, say — is a transport base. The transformation of x and $\bar{x}$ may be described in two steps. First x is transported to the position (i, j) of $\bar{x}$. This step does not leave the simplified situation described above. Then both x and $\bar{x}$ are compressed to the column (j, j). If i becomes abandoned, then there is no other base bs_l such that $i = Left'(bs_l)$. Thus the number of left boundaries of non-empty variable bases decreases at this step. This compensates the enlargement of the number of abandoned boundaries, we still have $LB'(GE') + AB'(GE') \leq LB(GE) + AB(GE) \leq N$. □

5 Terminating Minimal and Complete Word Unification

The decision procedure of the preceding section could in principle be turned into an algorithm which computes a minimal and complete set of unifiers for a given word equation WE_0, terminating if this set is finite, using the same construction as in Jaffar [Jaf]. A detailed proof is however rather tedious. Following an idea of F. Baader we shall now describe an algorithm which does the same and is conceptually simpler. The basic idea is the following: we shall use the basic algorithm for word unification described in section 1 in order to generate a minimal and complete set of unifiers for a given word equation WE_0. With some additional amount of work these substitutions may be displayed at the successful leaves of $\mathcal{T}_{LP}(WE_0)$ (see below). The tree $\mathcal{T}_{LP}(WE_0)$ will be generated in breadth first manner, i.e., level for level, and we shall just collect all substitutions associated with successful leaves. As a matter of fact, each level contains only a finite number of word equations. Thus Makanin's decision procedure may be used as a subprocedure which decides for every level whether it contains a solvable equation or not. As soon as we have found a level where all equations are unsolvable we shall stop. We shall now show that the sequence of all substitutions which are displayed at the successful leaves of $\mathcal{T}_{LP}(WE_0)$ is in fact a minimal and complete set of unifiers for WE_0. Thus it is trivial that our algorithm generates such a set and terminates if this set is finite.

To be precise, let us recall several definitions. If S is a substitution and WE is a word equation of the form $W_1 == W_2$, then $S(WE)$ denotes the word equation $S(W_1) == S(W_2)$. Let $\mathcal{W} \subseteq \mathcal{V}$, let S, T be substitutions. Then S is more general than T with respect to $\mathcal{W}$, $S \leq T$ $(\mathcal{W})$, iff there exists a substitution R such that $R(S(x)) = T(x)$ for all $x \in \mathcal{W}$. A set Σ of unifiers for a word equation WE_0 is *complete* if for every unifier T of WE_0 there exists an $S \in \Sigma$ such that $S \leq T$ $(\mathcal{W}_0)$ where $\mathcal{W}_0$ is the set of variables occurring in WE_0. A complete set Σ of unifiers for WE_0 is *minimal* if it does not contain two elements $S_1 \neq S_2$ such that $S_1 \leq S_2$ $(\mathcal{W}_0)$. A stricter condition than minimality is *disjointness*. A complete set Σ of unifiers for WE_0 is disjoint with respect to $\mathcal{W}_0$ if two distinct unifiers in Σ cannot be brought together: for all $S_1 \neq S_2 \in \Sigma$: there are no substitutions T_1, T_2 such that $T_1(S_1(x)) = T_2(S_2(x))$ for all $x \in \mathcal{W}_0$. $S \circ R$ denotes the product of two substitutions, S being applied first. $Var(WE)$ denotes the set of variables occurring in WE.

We shall now describe how unifiers may be displayed at successful leaves of $\mathcal{T}_{LP}(WE_0)$. For this purpose, let us associate with every word equation WE in $\mathcal{T}_{LP}(WE_0)$ the substitution S which is the product of all the substitutions S_i^{LP} which were applied at the transformation steps which led to WE. A convenient way to compute this substitution is to enrich WE_0 with a list $\langle x_1, ..., x_k\rangle$ of its variables, representing the trivial substitution. At every transformation step the respective substitution S_i^{LP} is not only applied to the word equation, but also to the actual substitution list. Let Σ denote the set of all substitutions which are associated with successful leaves of $\mathcal{T}_{LP}(WE_0)$ in this way.

Theorem 6.1: *Σ is a disjoint and complete set of unifiers for WE_0.*

A proof was given in Siekmann's thesis [Si78], but the notation used there is more complicated. For the convenience of the reader we include a rather compact argument.
It is trivial that the elements of Σ are unifiers for WE_0. Let $\mathcal{W}_0$ denote the set of variables occurring in WE_0. The following lemma immediately implies that Σ is a complete set of unifiers for WE_0.

Lemma 6.2: *Every unifier T of WE_0 recursively defines a path π through $\mathcal{T}_{LP}(WE_0)$ with the following property: if WE is a word equation occurring in π with associated substitution S, then $S \leq T$ $(\mathcal{W}_0)$.*

Proof: The topmost node of π contains WE_0, and it is clear that Id (identity) satisfies $Id \leq T$ $(\mathcal{W}_0)$. Suppose now for the induction hypothesis that WE is in π with associated substitution S satisfying the condition of the lemma. Thus there exists a substitution R such that $R(S(z)) = T(z)$ for all $z \in \mathcal{W}_0$. For the induction step let us consider the case where WE has the head (a, x). To find the successor of WE with respect to T
(1) we apply S_1^{LP} if $R(x) = a$,
(2) we apply S_2^{LP} if $R(x) = aV$, with $V \in (\mathcal{V} \cup \mathcal{C})^+$.

This subcase analysis is complete: R unifies WE since $S \circ R$ is a unifier for WE_0 and WE is a suffix of $S(WE_0)$ which may be reached by iterated deletion of identical head-symbols form both sides. In the first case we define $T_1' : y \mapsto R(y)$ for $x \neq y \in Var(WE)$, in the second case we define $T_2' : x \mapsto V$ and $y \mapsto R(y)$ for $x \neq y \in Var(WE)$. Now T_1' and T_2' show that $S_i^{LP} \leq R$ (i = 1,2).
The substitution associated with the successor is $S \circ S_i^{LP}$ in case (i), i = 1,2. Let $z \in \mathcal{W}_0$. We have $T_i'(S_i^{LP}(S(z))) = R(S(z)) = T(z)$. Thus $S \circ S_i^{LP} \leq T$ $(\mathcal{W}_0)$.
The proof for the situation where WE has head of type (x, y) is completely analogous. $\square$

If WE is any word equation in $\mathcal{T}_{LP}(WE_0)$ it is straightforward to show that the substitutions S_i^{LP} which may be applied are disjoint with respect to the variables occurring in the actual equation WE. However, in order to prove that Σ is a disjoint set of unifiers for WE_0 we have to show that the substitutions associated with word equations in distinct paths are disjoint with respect to the variables $\mathcal{W}_0$ of WE_0. For this purpose we shall introduce the notion of a D-preserving (disjointness-preserving) substitution:

Definition 6.3: A substitution S is *D-preserving* with respect to a set $\mathcal{W}$ of variables iff the following holds: for any two substitution T_1 and T_2 which are disjoint with respect to $\bigcup\{Var(S(x)); x \in \mathcal{W}\}$ the substitutions $S \circ T_1$ and $S \circ T_2$ are disjoint with respect to $\mathcal{W}$.

Lemma 6.4: *The transformation substitutions $S_i^{(LP)}$ are D-preserving with respect to the variables occurring in the actual word equation WE to be transformed ($i = 1, ..., 5$).*

Proof: Let us treat the situation where WE has head of type (x, y). Let us consider S_4^{LP}. Let T_1 and T_2 be two substitutions. Assume that $S_4^{LP} \circ T_1$ and $S_4^{LP} \circ T_2$ are *not* disjoint with respect to $Var(WE)$. Then there exist substitutions R_1 and R_2 such that $R_1(T_1(S_4^{LP}(z))) = R_2(T_2(S_4^{LP}(z)))$ for all $z \in Var(WE)$. In particular,

$$\begin{aligned} R_1(T_1(S_4^{LP}(x))) &= R_2(T_2(S_4^{LP}(x))) \\ R_1(T_1(S_4^{LP}(y))) &= R_2(T_2(S_4^{LP}(y))). \end{aligned}$$

Thus

$$\begin{aligned} R_1(T_1(yx))) &= R_2(T_2(yx)) \\ R_1(T_1(y))) &= R_2(T_2(y)) \end{aligned}$$

and thus $R_1(T_1(x))) = R_2(T_2(x))$. Since for $z \neq x$ always $S_4^{LP}(z) = z$, this shows that T_1 and T_2 are not disjoint with respect to $Var(S_4^{LP}(WE)$. Thus S_4^{LP} is in fact D-preserving. The proof for S_5^{LP} is symmetric, the proof for S_3^{LP} is trivial, the proof for head (a, x) is analogous. □

Lemma 6.5: *The set Σ is disjoint with respect to $\mathcal{W}_0$.*

Proof: The transformation substitutions S_i^{LP} which may be applied are disjoint with respect to the actual variable set. Suppose that S_1 and S_2 are substitutions associated with distinct successful leaves of $\mathcal{T}_{LP}(WE_0)$. A simple induction on the length of the common part of the paths leading to the respective leaves based on the preceding lemma shows that S_1 and S_2 are disjoint with respect to $\mathcal{W}_0$. □

Improved Decision Procedure

Let us conclude with an improved decision procedure where some ideas from the basic algorithm are used for a pre-analysis of word equations which always simplifies the decision procedure described in section 4 and makes it even dispensable in some cases. We have to choose a slightly modified representation of a word equation.

Example 6.6: The word equation $axbzx == zczyyy$ is translated into the following matrix

$$\left| \begin{array}{c} ax_1bz_1x_2 \\ z_2cz_3y_1y_2y_3 \end{array} \right| \left| \begin{array}{c} x_1 \\ x_2 \end{array} \right| \left| \begin{array}{c} y_1 \\ y_2 \\ y_3 \end{array} \right| \left| \begin{array}{c} z_1 \\ z_2 \\ z_3 \end{array} \right|$$

representing the four multi-equations $ax_1bz_1x_2 == z_2cz_3y_1y_2y_3$ (principal multi-equation), $x_1 == x_2$, $y_1 == y_2 == y_3$ and $z_1 == z_2 == z_3$.

Note that every new indexed variable has exactly two occurrences. Thus the new structure lies half on the way between word equations and generalized equations. We may now apply the Lentin/Plotkin transformation strategy in order to resolve all columns with two lines only — it is simple to see that the number of symbols cannot grow! The first two successors are the following systems (simplifying the first system in the straightforward way):

$$\left| \begin{array}{c} x_1bax_2 \\ cay_1y_2y_3 \end{array} \right| \left| \begin{array}{c} x_1 \\ x_2 \end{array} \right| \left| \begin{array}{c} y_1 \\ y_2 \\ y_3 \end{array} \right| \qquad \left| \begin{array}{c} x_1bz_1x_2 \\ z_2cz_3y_1y_2y_3 \end{array} \right| \left| \begin{array}{c} x_1 \\ x_2 \end{array} \right| \left| \begin{array}{c} y_1 \\ y_2 \\ y_3 \end{array} \right| \left| \begin{array}{c} z_1 \\ az_2 \\ z_3 \end{array} \right|.$$

Similar transformation steps are applied as long as there is any column with two lines left. We stop if a system is reached which is isomorphic to a predecessor in the same path. Eventually, when we reach a system where all columns have at least three lines, the matrix is translated into an equivalent set of generalized equations, introducing boundaries between all symbols occurring in a line of a column and choosing a linear order between the boundaries of the same column.

The use of such multi-equation systems has various advantages (see [Sc90] for a detailed discussion). For all word equations where every variable occurs at most twice the translation into generalized equations is completely avoided. Perhaps the most important point is the following: when the principal multi-equation is resolved, the number of additional transformation steps which lead to a unifier S cannot exceed the number $|X_1X_2\ldots X_n|$ where $X_i = S(x_i)$ and $x_1, \ldots, x_n$ are the variables occurring in WE_0, due to the vertical orientation of non-principal columns. Thus the maximal number of such transformation steps is *independent from the number of occurrences of each variable*, in contrast to the situation in the first decision procedure. The resolution of principal multi-equation corresponds to the computation of the possible linear orders between boundaries for the structures in $\geq (WE_0)$ in the standard approach. This amount of work cannot be avoided in either case.

References

[Ab87] H. Abdulrab, “Résolution d’équations sur les mots: Etude et implémentation LISP de l'algorithme de MAKANIN," Thèse de doctorat - Laboratoire d'informatique, Rouen 1987.

[AbPe89] H. Abdulrab, J.-P. Pécuchet, "Solving Word Equations," *J. Symbolic Computation* **8** (1989), pp. 499-521.

[Ah90] A.V. Aho, "Algorithms for Finding Patterns in Strings," in *Handbook of Theoretical Computer Schience* (J. van Leeuwen, Ed.) Elsevier Science Publishers 1990, pp. 256-300.

[Ba90] F. Baader, "Unification Theory," *Proceedings of the First International Workshop on Word Equations and Related Topics IWWERT '90*, (K.U. Schulz, Ed.) Tübingen 1990, Springer LNCS 572.

[Bu70] V.K. Bulitko, "Equations and Inequalities in a Free Group and a Free Semigroup," *Tul. Gos. Ped. Inst. Učen. Zap. Mat. Kafedr Vyp.2, Geometr. i Algebra*, (1970), pp. 242-252 (Russian).

[Col88] A. Colmerauer, "Final Specification for Prolog III," ESPRIT Ref. number P1219(1106), 1988.

[Coo72] D.C. Cooper, "Theorem Proving in Arithmetic without Multiplication," Machine Intelligence **7** (1972), pp. 82-95.

[Hm71] J.I. Hmelevskiĭ, "Equations in Free Semigroups", *Trudy Mat. Inst. Steklov*, Vol. 107, 1971; English translation: *Proc. Steklov Inst. of Mathematics*, Vol 107, 1971.

[Ja90] J. Jaffar, "Minimal and Complete Word Unification", *Journal of the ACM* **37**, No. 1, 1990, pp. 47-85.

[KoPa91] A. Kościelski, L. Pacholski, "Complexity of Makanin's Algorithms", Research Report, University of Wroclaw (1991); preliminary version: Complexity of Unification in Free Groups and Free Semigroups, *Proceedings 31st annual IEEE Symposium on Foundations of Computer Science*, Los Alamitos 1990, pp.824-829.

[Le72] A. Lentin, "Equations in Free Monoids", in *Automata Languages and Programming*, (M.Nivat, Ed.) North Holland Publishers, Amsterdam 1972, pp. 67-85.

[LeSc67] A. Lentin, M.P. Schützenberger, "A Combinatorial Problem in the Theory of Free Monoids", in *Proceedings of the University of North Carolina*, (1967) pp. 67-85.

[LiSi75] M. Livesey, J. Siekmann, "Termination and Decidability Results for String Unification", Essex University, 1975.

[Ma77] G.S. Makanin, “The problem of solvability of equations in a free semigroup”, *Math. USSR Sbornik* **32**, 2 (1977), pp. 129-198.

[Ma80] G.S. Makanin, “Recognition of the Rank of Equations in a free semigroup”, *Math. USSR Izvestija* Vol. **14** (1980) No. 3, pp. 499-545.

[Ma81] G.S. Makanin, “Equations in a free semigroup”, *Amer. Math. Soc. Transl.* (2) Vol. **117** (1981).

[Pé81] J.P. Pécuchet, “Equations avec constantes et algorithme de Makanin”, Thèse de doctorat, Laboratoire d' informatique, Rouen 1981.

[Pl72] G. Plotkin,“Building-in Equational Theories”, *Machine Intelligence* **7** (1972), pp. 73-90.

[Sc90] K.U. Schulz, “Makanin's Algorithm - Two Improvements and a Generalization”, (Habilitationsschrift), CIS-Report 91-39, University of Munich, also in *Proceedings of the First International Workshop on Word Equations and Related Topics IWWERT '90*, Tübingen 1990, Springer LNCS 572.

[Si75] J. Siekmann,“String-Unification”, Research Paper, Essex University, March 1975.

[Si78] J. Siekmann,“Unification and Matching Problems”, Ph.D. Thesis, Essex University, Memo CSA-4-78, 1978.

[Si89] J. Siekmann,“Unification Theory: A Survey”, in C. Kirchner (Ed.), *Special Issue on Unification, Journal of Symbolic Computation* **7**, 1989.

Appendix

We prove that transformation of proper non-false generalized equations preserves condition (ii) of the definition of properness (4.4). Let us introduce a notational convention: in *GE* = (*BS*,*BD*,*Col*,*BE*), three types of oriented boundary equations may be distinguished:

(i) the standard equations (see 2.8),

(ii) the boundary equations of the form $(i, x_c, j, \bar{x}_c)$ or

(iii) of the reverse form $(j, \bar{x}_c, i, x_c)$.

Equations of type (ii) will be called print equations, equations of type (iii) reverse equations. In the following, these types will play a distinct role. Thus, instead of using symbols $E, E_1, \ldots$ for arbitrary boundary equations of GE we shall sometimes use symbols $S, S_1, \ldots$ for standard equations, symbols $P, P_1, \ldots$ for print equations and symbols $R, R_1, \ldots$ for reverse equations.

When we consider a successor $GE' = GE_{\preceq} \in \mathit{Transf}(GE)$, then S' denotes the natural image of S with respect to $\preceq$ (see Def. 2.8). If $P = (i, x_c, j, \bar{x}_c) \in BE$ $(R = (j, \bar{x}_c, i, x_c) \in BE)$, then it is convenient to call $P' = (i, x'_c, j, \bar{x}'_c)$ $(R' = (j, \bar{x}'_c, i, x'_c))$ the natural image of P (R) with respect to $\preceq$. Note that $S' \in BE'$ iff S' is non-degenerate (3.4.1) and that P' (R') is a boundary equation of GE' iff $cr < i$ (3.4.2).

Lemma 1: *Assume that GE is proper. Let $P_1, ..., P_k, \pi$ $(R_1, ..., R_k, \pi)$ be a convex chain and $GE' = (BS', BD', Col', BE') \in \mathit{Transf}(GE)$. Then $P'_2, ..., P'_k \in BE'$ $(R'_1, ..., R'_{k-1} \in BE')$.*

Proof: Let $P_1, P_2 = (h, x_c, i, \bar{x}_c), (i, x_c, j, \bar{x}_c)$. Then $cr < i$ since i is in $Col(\bar{x}_c)$. Thus $P'_2 \in BE'$, by 3.4.2. Similarly it is clear that $P'_3, ..., P'_k$ and $R'_1, ..., R'_{k-1}$ are in BE'. $\square$

Lemma 2: *Let GE be a proper, non-trivial generalized equation and let $GE' \in \mathit{Transf}(GE)$. Assume that no natural image S' of a standard boundary equation $S \in BE$ is degenerate in $GE' = (BS', BD', Col', BE')$. There exists a recursive translation which assigns to every clean convex chain $\pi = E_1, ..., E_m, w$ of GE such that $E'_1 \in BE'$ a convex chain π' of GE' starting with E'_1. Moreover, if π is adaptive, then the following properties hold:*

(i) if $E_1 = S_1$ is a standard equation, then π' is adaptive,

(ii) if $\pi = P_1, \rho$ (where ρ is a chain of GE), then $\pi' = P'_1, \rho'$ where ρ' is adaptive and the leading base z of ρ' satisfies $l^{\star\star} \preceq Left'(z)$.

(iii) if $\pi = R_1, \rho$ (where ρ is a chain of GE) then π' is adaptive or has the form R'_1, R'^{-1}_1, γ where γ is an adaptive chain of GE' with leading base z satisfying $l^{\star\star} \preceq Left'(z)$.

Proof: For the translation we shall use the following notational convention: if $S_1 = (i, x, j, \bar{x})$ is an oriented boundary equation of GE (if w is a base of GE), then an index ${}^{(f)}S_1$ $({}^{(f)}w)$ indicates that x (w) is a fixed base, similar indices ${}^{(t)}$ indicate that the respective bases are transport bases. A right upper index $S_1^{(f)}$ $(S_1^{(t)})$ indicates that $\bar{x}$ is a fixed (transport) base. The proof is now by induction on the length m of π.

(I) For $m = 1$, we distinguish three subcases, depending on the type of the equation E_1.

Case 1: $E_1 = S_1 = (i, x, j, \bar{x})$ is a standard equation.

(a) $(S_1^{(f)}, {}^{(f)}w)' = S_1', w$

(b) $(S_1^{(f)}, {}^{(t)}w)' = S_1', (j, x_c', j^{tr}, \bar{x}_c'), w^{tr}$

(c) $(S_1^{(f)}, x_c)' = S_1', x_c'$

(d) $(S_1^{(f)}, \bar{x}_c)' = S_1', \bar{x}_c'$

(e) $(S_1^{(t)}, {}^{(f)}w)' = S_1', (j^{tr}, \bar{x}_c', j, x_c'), w$

(f) $(S_1^{(t)}, {}^{(t)}w)' = S_1', w^{tr}$.

If π has the form $\pi = S_1^{(t)}, w$, then the cases $w = x_c$ and $w = \bar{x}_c$ cannot occur since j is in $Col(x_c)$, compare 2.4. Let us show that the translation satisfies the conditions of the lemma. First note that $(j, x_c', j^{tr}, \bar{x}_c') \in BE'$ in cases (b) and (e), by 3.4.3: in case (b), $cr < j$ since j is in $Col(\bar{x})$ and $\bar{x}$ is fixed, compare 2.4. Moreover, $j = Right(w)$ is a transport base, by 2.5 (ii). Case (e) is similar, using 2.5 (iii). In all cases it is now trivial to verify that π' is a chain of GE'. For the cases (a), (c), (d) and (f) the convexity of π' is trivial. In case (b), $\chi'(\pi')$ is the word $\succeq\succ$, by 2.9[12]. In case (e), $\chi'(\pi')$ is the word $\prec\preceq$, by 2.9. Now assume that π is adaptive. In cases (a) and (f) it is trivial that π' is adaptive, following directly from 2.6 (ii), 2.8 and 3.3. In cases (b) and (c) π cannot be adaptive, by 2.4. In case (d) $Left(\bar{x}) \leq l^{\star\star}$ (by assumption) and $l^{\star\star} \prec Left'(\bar{x}_c')$ imply that π' is adaptive, by 2.2, 2.6 (ii) and 3.3.3. In case (e) we saw that π' is adaptive.

Case 2: $E_1 = P_1 = (i, x_c, j, \bar{x}_c)$ is a print equation.

(a) $(P_1, {}^{(f)}w)' = P_1', w$

(b) $(P_1, {}^{(t)}w)' = P_1', (j, x_c', j^{tr}, \bar{x}_c'), w^{tr}$

(c) $(P_1, x_c)' = P_1', x_c'$

The case $w = \bar{x}_c$ cannot occur since $Right(w) = j$ is in $Col(\bar{x}_c)$. Recall that $P_1' \in BE'$, by the assumption of the lemma. In case (b), $(j, x_c', j^{tr}, \bar{x}_c') \in BE'$ since

[12] A symbol "$\succeq$" indicates that the respective letter may be $=$ or $\succ$.

$cr < j$ (j is in $Col(\bar{x}_c)$) and $j = Right(w)$ is a transport boundary, by 2.5 (ii). It is trivial to show that π' is always a chain of GE'. The convexity of π' is trivial for (a) and (c). In case (b), $\chi'(\pi')$ is the word $\succ\succeq$, by 2.9. Note that π does not have the form $\pi = P_1, \rho$ presupposed in (ii), since every chain has at least one boundary equation. Thus there is nothing to show with respect to (ii) here.

Case 3: $E_1 = R_1 = (j, \bar{x}_c, i, x_c)$ is a reverse equation.

Here $cr < i$ since $R_1' \in BE'$, by assumption:

(a) $(R_1,^{(f)} w)' = R_1', w$

(b) $(R_1,^{(t)} w)' = R_1', (i, x_c', i^{tr}, \bar{x}_c'), w^{tr}$

The cases $w = x_c$ or $w = \bar{x}_c$ cannot occur since $Right(w) = i$ is in $Col(x_c)$. In case (b), $cr < i = Right(w)$ is a transport boundary, by 2.5 (ii). Thus $(i, x_c', i^{tr}, \bar{x}_c') \in BE'$, by 3.4.3. In both cases it is now trivial that π' is a chain of GE'. In case (b), $\chi'(\pi')$ is the word $=\succeq$ (2.9), thus π' is always convex. Note that π does not have the form $\pi = R_1, \rho$ presupposed in (iii). Thus there is nothing to show with respect to (iii). But, for later purposes note that π' is adaptive in case (a) since $Left'(x_c') = cr \preceq Left'(w)$ (2.4, w is fixed), and that $l^{\star\star} \preceq Left'(w^{tr})$ in case (b).

(II) For $m > 1$, $\pi = E_1, \rho$ where ρ is a chain of GE and we distinguish again the three subcases where E_1 is a standard equation, a print equation or a reverse equation. For the rest of the proof, greek letters $\rho, \theta, \ldots$ always denote chains.

Case 1: $\pi = S_1, \rho$. Let $S_1 = (h, \bar{x}, i, x)$. We consider the two subcases where x is a fixed base (1.1) or a transport base (1.2).

Case 1.1, $\pi = S_1^{(f)}, \rho$.

Here $cr < i$, by 2.4, since i is in $Col(x)$. Chain ρ may start with a standard equation, with a print equation or with a reverse equation. Accordingly we treat subcases 1.1.1, 1.1.2 and 1.1.3.

Case 1.1.1: $\pi = S_1^{(f)}, S_2, \theta$ or $\pi = S_1^{(f)}, S_2, w$. Let $S_2 = (i, y, j, \bar{y})$.

(a) $(S_1^{(f)},^{(f)} S_2, \theta)' = S_1', (S_2, \theta)'$ resp. $(S_1^{(f)},^{(f)} S_2, w)' = S_1', (S_2, w)'$

(b) $(S_1^{(f)},^{(t)} S_2, \theta)' = S_1', (i, x_c', i^{tr}, \bar{x}_c'), (S_2, \theta)'$
resp. $(S_1^{(f)},^{(t)} S_2, w)' = S_1', (i, x_c', i^{tr}, \bar{x}_c'), (S_2, w)'$

Case (a): By induction hypothesis, π' is a chain and $(S_2,\theta)'$ or $(S_2,w)'$ are convex. This implies that π' is convex: if $\mathit{Left}'(x) \prec \mathit{Left}'(y)$, then $\mathit{Left}(x) < \mathit{Left}(y)$ and π is adaptive. Thus S_2,θ (resp. S_2,w) is adaptive. By induction hypothesis (i) also $(S_2,\theta)'$ (resp. $(S_2,w)'$) is adaptive. Therefore π' is convex and adaptive, in this case. If $\mathit{Left}'(x) \succeq \mathit{Left}'(y)$, then π' is convex by induction hypothesis.
Case (b): Note that $(i,x_c',i^{tr},\bar{x}_c') \in \mathit{BE}'$, by 3.4.3 since $cr < i$ is a transport boundary, by 2.5 (iii). Thus π' is a chain of GE', by induction hypothesis. Now 2.4 and 2.9 show that $\chi'(\pi')$ has the form $\succeq\succ\ldots$ in both cases. Thus π' is convex by induction hypothesis. Since π cannot be adaptive (see 2.4) there is nothing to show with respect to (i).

Case 1.1.2: $(S_1^{(f)},P_1,\theta)' = S_1',(P_1,\theta)'$ resp. $(S_1^{(f)},P_1,w)' = S_1',(P_1,w)'$.

Let $P_1 = (i,x_c,j,\bar{x}_c)$. Here $P_1' = (i,x_c',j,\bar{x}_c') \in \mathit{BE}'$ since $cr < i$. The induction hypothesis shows that π' is a chain of GE' in both cases. Since $\mathit{Left}'(x_c') = cr \leq \mathit{Left}(x)$ it is clear that π' is convex, by induction hypothesis. Since π cannot be adaptive there is nothing to show with respect to (i).

Case 1.1.3: $\pi = S_1^{(f)},R_1,\theta$ or $\pi = S_1^{(f)},R_1,w$. Let $R_1 = (i,\bar{x}_c,j,x_c)$.

(a) If $\rho = R_1,w$ and $j \leq cr$ ($R_1' \notin \mathit{BE}'$), then $\pi' = S_1',w^{tr}$.

In this subcase witness w is necessarily a transport base since $\mathit{Right}(w) = j$ is in $\mathit{Col}(x_c)$ and $j \leq cr$ (2.3). By 2.6 (iv), $j^{tr} = i$ and π' is a chain of GE'. Clearly π' is convex. If π is adaptive, then $\mathit{Left}(x) \leq l^{\star\star}$. Thus $\mathit{Left}'(x) \preceq \mathit{Left}'(w^{tr})$ (see 3.3.1, 3.3.2, 2.6 (i) and (ii)) and π' is adaptive.

(b) If $\rho = R_1,w$ and $cr < j$ ($R_1' \in \mathit{BE}'$), then we have two subcases:

(b-1) If $\rho = R_1,{}^{(f)}\,w$, then $\pi' = S_1',R_1',w$.
(b-2) If $\rho = R_1,{}^{(t)}\,w$, then $\pi' = S_1',w^{tr}$.

The cases $w = x_c$ and $w = \bar{x}_c$ cannot occur in (b) since $\mathit{Right}(w) = j$ is in $\mathit{Col}(x_c)$. By 2.6 (iv), $j^{tr} = i$ and it follows that π' is a chain of GE'. Convexity is trivial for (b-2) and follows for (b-1) since $\chi'(\pi')$ has last symbol "$\preceq$". Now suppose that π is adaptive. Then $\mathit{Left}(x) \leq l^{\star\star}$. In case (b-1) $\mathit{Left}'(x) \prec cr^{tr}$, in case (b-2) $\mathit{Left}'(x) \preceq \mathit{Left}'(w^{tr})$. Thus π' is adaptive.

(c) If $\rho = R_1,\theta$ and $j \leq cr$ ($R_1' \notin \mathit{BE}'$), then ρ has necessarily the form $\rho = R_1,{}^{(t)}\,S_2,\sigma$ (or $\rho = R_1,{}^{(t)}\,S_2,w$): θ cannot start with a reverse equation since j is not in $\mathit{Col}(\bar{x}_c)$; by 4.8, θ cannot start with a print equation since π is clean; θ cannot start with ${}^{(f)}S_2$ since $j \leq cr$ (2.4). We define $\pi' = S_1',(S_2,\sigma)'$ (resp. $\pi' = S_1',(S_2,w)'$).

By 2.6 (iv), $j^{tr} = i$ and it is clear that π' is a chain of GE'. To show that π' is convex, we distinguish two subcases: if R_1, S_2, σ (resp. R_1, S_2, w) is adaptive, then $(S_2, \sigma)'$ (resp. $(S_2, w)'$) is adaptive, by induction hypothesis, and π' is convex. In the other case, if R_1, S_2, σ (resp. R_1, S_2, w) is not adaptive, let $S_2 = (j, y, k, \bar{y})$. Then $\mathit{Left}(y) = l^\star$ and $\mathit{Left}(x) \geq l^{\star\star}$. Thus $\mathit{Left}'(x) \succeq l^{\star\star} = \mathit{Left}'(y^{tr})$ and π' is convex. Now suppose that π is adaptive. Then $\mathit{Left}(x) \leq l^{\star\star}$. By induction hypothesis, $(S_2, \sigma)'$ (resp. $(S_2, w)'$) is adaptive. Since $\mathit{Left}'(x) \preceq l^{\star\star} \preceq \mathit{Left}'(y^{tr})$ also π' is adaptive.

(d) If $\rho = R_1, \theta$ and $cr < j$ ($R_1' \in BE'$), then we distinguish two subcases:

(d-1) R_1, θ is adaptive. According to the induction hypothesis, two subcases may occur:

If $(R_1, \theta)'$ is adaptive, then $\pi' = S_1', (R_1, \theta)'$.

If $(R_1, \theta)' = R_1', R_1'^{-1}, \gamma$ where γ is adaptive with leading base z satisfying $l^{\star\star} \preceq \mathit{Left}'(z)$, then $\pi' = S_1', \gamma$.

(d-2) If R_1, θ is not adaptive, then $l^{\star\star} \leq \mathit{Left}(x)$, by convexity of π. In this subcase, θ has necessarily the form $\theta = {}^{(t)}S_2, \sigma$ or $\theta = {}^{(t)}S_2, w$: if θ would start with a reverse equation, then R_1, θ would be adaptive, θ cannot start with a print equation since π is clean and θ cannot start with ${}^{(f)}S_2$ since R_1, θ is not adaptive. We define $\pi' = S_1', (S_2, \sigma)'$ or $\pi' = S_1', (S_2, w)'$.

Subcases (d-1): It is clear for both subcases that π' is a convex chain of GE'. If π is adaptive, then $\mathit{Left}(x) \leq l^{\star\star}$ and $\mathit{Left}'(x) \preceq l^{\star\star}$. In both cases, π' is adaptive by induction hypothesis.
Subcase (d-2): Let $S_2 = (j, y, k, \bar{y})$. Here $\mathit{Left}(y) = l^\star$ since R_1, θ is not adaptive. Thus $\mathit{Left}'(y^{tr}) = l^{\star\star} \preceq \mathit{Left}'(x)$ and π' is convex by induction hypothesis. There is nothing to show with respect to (i) here.

Case 1.2, $\pi = S_1^{(t)}, \rho$. Let $S_1 = (h, \bar{x}, i, x)$.

Since ρ may start with a standard equation, with a print equation or with a reverse equation we treat three subcases.

Case 1.2.1, $\pi = S_1^{(t)}, S_2, \theta$ or $\pi = S_1^{(t)}, S_2, w$. Let $S_2 = (i, y, k, \bar{y})$.

(a) $(S_1^{(t)}, {}^{(f)}S_2, \theta)' = S_1', (i^{tr}, \bar{x}_c', i, x_c'), (S_2, \theta)'$
resp. $(S_1^{(t)}, {}^{(f)}S_2, w)' = S_1', (i^{tr}, \bar{x}_c', i, x_c'), (S_2, w)'$

(b) $(S_1^{(t)}, {}^{(t)}S_2, \theta)' = S_1', (S_2, \theta)'$ resp. $(S_1^{(t)}, {}^{(t)}S_2, w)' = S_1', (S_2, w)'$

Case (a): By 3.4.3, $(i^{tr}, \bar{x}_c, i, x_c) \in BE'$ since $cr < i$ (i is in $Col(y)$, y is fixed, 2.4) and i is a transport boundary, by 2.5 (iii). Thus it is clear that π' is a chain of GE'. Here π is adaptive, by 2.4. Thus $(S_2, \theta)'$ (resp. $(S_2, w)'$) is adaptive, by induction hypothesis. Now 2.4 and 2.9 show that π' is adaptive.
Case (b): It is clear that π' is a chain of GE'. $(S_2, \theta)'$ (resp. $(S_2, w)'$) is convex, by induction hypothesis. If $Left'(x) \prec Left'(y)$, then $Left(x) < Left(y)$ (2.6 (ii)) and π is adaptive. Thus π' is convex and adaptive, by induction hypothesis. If $Left'(x) \succeq Left'(y)$, then π' is convex, by induction hypothesis. If π is adaptive clearly π' is adaptive, by 2.6 (ii) and induction hypothesis.

Case 1.2.2, $\pi = S_1^{(t)}, P_1, \theta$ (resp. $\pi = S_1^{(t)}, P_1, w$). Let $P_1 = (i, x_c, j, \bar{x}_c)$.

Note that $i^{tr} = j$, by 2.6 (iv). We distinguish several subcases:

(a) $(S_1^{(t)}, P_1, P_2, \sigma)' = S_1', (P_2, \sigma)'$ resp. $(S_1^{(t)}, P_1, P_2, w)' = S_1', (P_2, w)'$ (note that $P_2' \in BE'$, by Lemma 1).

(b) $(S_1^{(t)}, P_1, {}^{(f)}S_2, \sigma)' = S_1', (S_2, \sigma)'$ resp. $(S_1^{(t)}, P_1, {}^{(f)}S_2, w)' = S_1', (S_2, w)'$

(c) $(S_1^{(t)}, P_1, {}^{(t)}S_2, \sigma)' = S_1', (j, x_c', j^{tr}, \bar{x}_c'), (S_2, \sigma)'$ resp. $(S_1^{(t)}, P_1, {}^{(t)}S_2, w)' = S_1', (j, x_c', j^{tr}, \bar{x}_c'), (S_2, w)'$

(d) $(S_1^{(t)}, P_1, {}^{(f)}w)' = S_1', w$

(e) $(S_1^{(t)}, P_1, {}^{(t)}w)' = S_1', (j, x_c', j^{tr}, \bar{x}_c'), w^{tr}$

(f) $(S_1^{(t)}, P_1, x_c)' = S_1', x_c'$.

In cases (a), (b), (d) and (f) it is clear that π' is a chain of GE' since $i^{tr} = j$. In cases (c) and (e) j is in $Col(\bar{x}_c)$, thus $cr < j$. Moreover, j is a transport boundary, by 2.5 (iii) and (ii), thus $(j, x_c', j^{tr}, \bar{x}_c') \in BE'$, by 3.4.3, and π' is a chain of GE'. In case (a), π' is convex by induction hypothesis since x_c' is the leading base of $(P_2, \sigma)'$ (resp. $(P_2, w)'$). Let $S_2 = (j, y, k, \bar{y})$ in cases (b) and (c). For (b), we distinguish two cases: if P_1, S_2, σ (or P_1, S_2, w) is adaptive, then π' is convex, by induction hypothesis. If P_1, S_2, σ (or P_1, S_2, w) is not adaptive, then $Left(y) \leq l^{\star\star}$, thus $Left'(y) \preceq l^{\star\star} \preceq Left'(x^{tr})$ and π' is convex. In case (c) it follows from 2.2 and 2.9 and the induction hypothesis that π' is convex. In cases (d) and (f) it is trivial that π' is convex. In case (e) π' is convex, by 2.9.
Suppose that π is adaptive. This may only occur in case (b) and (d). If π is adaptive in case (b), then $Left(x) = l^\star$ and $l^{\star\star} \leq Left(y)$. Thus $Left'(x^{tr}) \preceq Left'(y)$ and π' is adaptive by induction hypothesis. If π is adaptive in case (d), then $Left(x) = l^\star$ and $l^{\star\star} \leq Left(w)$ and it follows that π' is adaptive.

Case 1.2.3, $\pi = S_1^{(t)}, R_1, \theta$ (or $\pi = S_1^{(t)}, R_1, w$). Let $R_1 = (i, \bar{x}_c, j, x_c)$.

Note that π is adaptive in this case. We have $cr < i$ since i is in $Col(\bar{x}_c)$ and i is a transport boundary, by 2.5 (iii). Thus $(i^{tr}, \bar{x}'_c, i, x'_c) \in BE'$, by 3.4.3. Moreover, $j^{tr} = i$, by 2.6 (iv).

(a) If $cr < j$ ($R'_1 \in BE'$) and $\rho = R_1, \theta$, then
$\pi' = S'_1, (i^{tr}, \bar{x}'_c, i, x'_c), (R_1, \theta)'$ if $(R_1, \theta)'$ is adaptive and
$\pi' = S'_1, (i^{tr}, \bar{x}'_c, i, x'_c), \gamma$ if $(R_1, \theta)' = R'_1, R'^{-1}_1, \gamma$ where γ is adaptive
(note that one of these cases must occur, by induction hypothesis).

It is clear in both cases that π' is a chain of GE'. We have $Left(x) < cr$ and $Left(x^{tr}) \prec cr^{tr}$. This implies that π' is adaptive in both cases since cr is minimal with respect to $\preceq$.

(b) If $cr < j$ ($R'_1 \in BE'$) and $\rho = R_1, w$, then
$\pi' = S'_1, (i^{tr}, \bar{x}'_c, i, x'_c), (i, \bar{x}'_c, j, x'_c), w$ if w is a fixed base and
$\pi' = S'_1, (i^{tr}, \bar{x}'_c, i, x'_c), w^{tr}$ if w is a transport base.

Note that the cases $w = x_c$ and $w = \bar{x}_c$ cannot occur here since $j = Right(w)$ is in $Col(x_c)$. Since $(i^{tr}, \bar{x}'_c, i, x'_c) \in BE'$ clearly π' is a chain of GE' in both cases. In the first case, $\chi'(\pi')$ is the word $\prec\prec\preceq$ and in the second case $\chi'(\pi')$ is $\prec\preceq$, by 2.9 and since $Left'(x'_c) = cr$. It follows that π' is convex and adaptive.

(c) If $j \leq cr$ ($R'_1 \notin BE'$), then π necessarily has one of the following forms:

(c-1) $\pi = S_1^{(t)}, R_1,^{(t)} w$, and we define $\pi' = S'_1, (i^{tr}, \bar{x}'_c, i, x'_c), w^{tr}$

(c-2) $\pi = S_1^{(t)}, R_1,^{(t)} S_2, \sigma$ (or $\pi = S_1^{(t)}, R_1,^{(t)} S_2, w$), and we define
$\pi' = S'_1, (i^{tr}, \bar{x}'_c, i, x'_c), (S_2, \sigma)'$ (resp. $\pi' = S'_1, (i^{tr}, \bar{x}'_c, i, x'_c), (S_2, w)'$).

In fact π cannot have the form $S_1^{(t)}, R_1,^{(f)} w$ or $S_1^{(t)}, R_1, x_c$ or $S_1^{(t)}, R_1, \bar{x}_c$ since $Right(w) = j \leq cr$. Similarly π cannot have the form $S_1^{(t)}, R_1,^{(f)} S_2, \sigma$ or $\pi = S_1^{(t)}, R_1,^{(f)} S_2, w$ since $j \leq cr$, π cannot have the form $S_1^{(t)}, R_1, R_2, \ldots$ by Lemma 1 and π cannot have the form $S_1^{(t)}, R_1, P, \ldots$ since it is clean. In both cases (c-1) and (c-2) it is clear that π' is a chain of GE' since $j^{tr} = i$, by 2.6 (iv). In case (c-1) we have $Left'(x'_c) = cr \preceq l^{\star\star} \preceq Left'(w^{tr})$. Now π' is convex and adaptive, by 2.9. Similarly, 2.9 shows that π' is adaptive in case (c-2).

Case 2: $\pi = P_1, \rho$, where ρ is a chain of GE. Let $P_1 = (i, x_c, j, \bar{x}_c)$.

Note that $cr < i$, by assumption.

Case 2.1: $(P_1,^{(f)} S_1, \theta)' = P'_1, (S_1, \theta)'$ resp. $(P_1,^{(f)} S_1, w)' = P'_1, (S_1, w)'$

It is clear that π' is a chain of GE'. Let $S_1 = (j, z, k, \bar{z})$. By induction hypothesis, $(S_1, \theta)'$ is convex. If $\chi'(\pi')$ starts with "$\prec$", then $Left'(\bar{x}'_c) \prec Left'(z)$ and $l^{\star\star} < Left(z)$. Thus π is adaptive, $(S_1, \theta)'$ (resp. $(S_1, w)'$) is adaptive by induction hypothesis and π' is convex. We prove (ii): if π is adaptive, then $l^{\star\star} \leq Left(z)$. Thus $l^{\star\star} \preceq Left'(z)$. By induction hypothesis, $(S_1, \theta)'$ (resp. $(S_1, w)'$) is adaptive.

Case 2.2: $(P_1,^{(t)} S_1, \theta)' = P'_1, (j, x'_c, j^{tr}, \bar{x}'_c), (S_1, \theta)'$
resp. $(P_1,^{(t)} S_1, w)' = P'_1, (j, x'_c, j^{tr}, \bar{x}'_c), (S_1, w)')$.

Here $cr < j$ since j is in $Col(\bar{x}_c)$, and j is a transport boundary, by 2.5 (iii) or (ii). Thus $(j, x'_c, j^{tr}, \bar{x}'_c) \in BE'$, by 3.4.3. It is clear that π' is a chain of GE'. It follows from 2.9 that π' is convex. Since π is not adaptive, by 2.2 and 2.4, there is nothing to show with respect to (ii).

Case 2.3: $(P_1, P_2, \theta)' = P'_1, (P_2, \theta)'$ resp. $(P_1, P_2, w)' = P'_1, (P_2, w)')$.

Note that $P'_2 \in BE'$, by Lemma 1. Thus π' is a chain of GE', by induction hypothesis. It is clear that π' is convex, by induction hypothesis. Since π is not adaptive, there is nothing to show with respect to (ii).

Remark: Since π is clean, the subcase analysis for case 2 is complete.

Case 3: $\pi = R_1, \rho$ where ρ is a chain of GE. Let $R_1 = (j, \bar{x}_c, i, x_c)$.

Note that $cr < i$, by assumption.

Case 3.1: $(R_1,^{(f)} S_1, \theta)' = R'_1, (S_1, \theta)'$ resp. $(R_1,^{(f)} S_1, w)' = R'_1, (S_1, w)'$.

It is clear that π' is a chain of GE'. Since π is necessarily adaptive, $(S_1, \theta)'$ (resp. $(S_1, w)'$) is adaptive, by induction hypothesis. It follows that π' is adaptive. Thus condition (iii) is satisfied.

Case 3.2: $(R_1,^{(t)} S_1, \theta)' = R'_1, R'^{-1}_1, (S_1, \theta)'$ resp.
$(R_1,^{(t)} S_1, w)' = R'_1, R'^{-1}_1, (S_1, w)'$.

Note that $R'^{-1}_1 = (i, x'_c, j, \bar{x}'_c) = (i, x'_c, i^{tr}, \bar{x}'_c) \in BE'$ (2.5 (iii), 3.4.3) since $i^{tr} = j$, by 2.6 (iv). It is clear that π' is a chain of GE'. It follows from 2.9 that π' is convex. Suppose now that π is adaptive. Then $\gamma = (S_1, \theta)'$ (resp. $\gamma = (S_1, w)'$) is adaptive, by induction hypothesis. If $S_1 = (i, y, k, \bar{y})$, then the leading base of γ is $z = y^{tr}$. Thus $l^{\star\star} \preceq Left'(z)$ and condition (iii) is verified.

Case 3.3: For $\pi = R_1, R_2, \theta$ or $\pi = R_1, R_2, w$ we distinguish several cases. Let $R_2 = (i, \bar{x}_c, k, x_c)$.

(a) If $k \leq cr$ ($R'_2 \notin BE'$) and $\pi = R_1, R_2, w$, then w is a transport base since $Right(w) = k$. We define $\pi' = R'_1, w^{tr}$.

(b) If $k \leq cr$ ($R'_2 \notin BE'$) and $\pi = R_1, R_2,^{(t)} S_1, \sigma$ (or $\pi = R_1, R_2,^{(t)} S_1, w$), then $\pi' = R'_1, (S_1, \sigma)'$ (resp. $\pi' = R'_1, (S_1, w)'$).

Remark: For $k \leq cr$ all other cases may be excluded: the leading base of θ cannot be a fixed base, by 2.4, it cannot be $\bar{x}_c$, it cannot be x_c since π is clean (4.8).

(c) If $cr < k$ ($R'_2 \in BE'$), $\pi = R_1, R_2, \theta$ and $(R_2, \theta)'$ is adaptive, then $\pi' = R'_1, (R_2, \theta)'$.

(d) If $cr < k$ ($R'_2 \in BE'$), $\pi = R_1, R_2, \theta$ and $(R_2, \theta)'$ has the form R'_2, R'^{-1}_2, γ where γ is adaptive with leading base z satisfying $l^{\star\star} \preceq Left'(z)$, then $\pi' = R'_1, \gamma$.

Remark: chains of the form $R_1, R_2, \ldots$ are adaptive. Thus the case analysis for $cr < k$ ($R'_2 \in BE'$), $\pi = R_1, R_2, \theta$ is complete, by induction hypothesis (iii).

(e) If $cr < k$ ($R'_2 \in BE'$), $\pi = R_1, R_2,^{(f)} w$, then $\pi' = R'_1, (R_2, w)'$.

(f) If $cr < k$ ($R'_2 \in BE'$), $\pi = R_1, R_2,^{(t)} w$, then $\pi' = R'_1, w^{tr}$.

It is trivial to show that π' is a chain of GE' in all cases. For (a), (b) and (f) note that $k^{tr} = i$, by 2.6 (iv). In all cases π is adaptive. Let us show that π' is adaptive, too. In case (a), π' is adaptive since $Left'(x'_c) = cr \preceq l^{\star\star} \preceq Left'(w^{tr})$. In case (b) $(S_1, \sigma)'$ (or $(S_1, w)'$) is adaptive, by induction hypthesis, and it follows that π' is adaptive. Cases (c) and (d) are similar. In case (e), $(R_2, w)'$ is adaptive, as mentioned in case 3 of (I). It follows that π' is adaptive. In case (f) we see as in case (a) that π' is adaptive.
Remark: since π is clean, the case $\pi = R_1, P_1, \ldots$ does not occur. $\square$

Lemma 3: *Suppose that GE is proper, let $GE' = GE_{\preceq} \in Transf(GE)$. If all natural images S' of standard equations $S \in BE$ are non-degenerate with respect to $\preceq$, then GE' satisfies condition (ii) of definition 4.4.*

Proof: We have to show that for any boundary equation E' of GE' there exists a convex chain of GE' with first element E' or E'^{-1}. By Lemma 2 this is clear for all boundary equations E' of GE which are natural images of boundary equations E of GE. Now let E' be a new boundary equation of GE', i.e. an equation of the form $E' = (i, x'_c, i^{tr}, \bar{x}'_c)$ where $cr < i$ is a transport boundary of GE. If $i = Right(w)$ where w is a transport base of GE (see 2.5 (ii)), then $(i, x'_c, i^{tr}, \bar{x}'_c), w^{tr}$ is a convex chain of GE'. In the other case, i occurs in a boundary equation

$S = (i, x, j, \bar{x})$ or $S^{-1} = (j, \bar{x}, i, x)$ of *GE* where x is a transport base (2.5 (iii)). Since *GE* is proper, there exists a clean convex chain π of *GE* which starts with S (case 1) or with S^{-1} (case 2). In case 1, *GE'* has a convex chain π' with first element S' according to Lemma 2. Now $(i, x'_c, i^{tr}, \bar{x}'_c), \pi'$ is a convex chain of *GE'*, by 2.9. In case 2, we distinguish several subcases. In every subcase we shall give a convex chain ρ of *GE'* starting with $(i, x'_c, i^{tr}, \bar{x}'_c)$ or $(i^{tr}, \bar{x}'_c, i, x'_c)$. We use notation and result of Lemma 2:

(a) If $\pi = (j, \bar{x}, i, x),^{(t)} w$, then $\rho = (i, x'_c, i^{tr}, \bar{x}'_c), w^{tr}$.

(b) If $\pi = (j, \bar{x}, i, x),^{(t)} S_2, \ldots$, then we replace S^{-1} by S_2 and are in case 1 described above.

(c) If $\pi = (j, \bar{x}, i, x),^{(f)} S_2, \ldots$ (or $\pi = (j, \bar{x}, i, x),^{(f)} w$), then $\rho = (i^{tr}, \bar{x}'_c, i, x'_c), (S_2, \ldots)'$ (resp. $\rho = (i^{tr}, \bar{x}'_c, i, x'_c), w$).

Remark: Since i is a transport boundary it is in $Col(x_c)$ and convex chains $(j, \bar{x}, i, x), x_c$ or $(j, \bar{x}, i, x), \bar{x}_c$ cannot occur.

(d) the case where $\pi = (j, \bar{x}, i, x), (i, x_c, k, \bar{x}_c), \ldots$ need not to be considered since here $k = i^{tr}$. Thus $(i, x'_c, i^{tr}, \bar{x}'_c)$ is a natural image of $(i, x_c, k, \bar{x}_c) \in BE$.

(e) If $\pi = (j, \bar{x}, i, x), (i, \bar{x}_c, k, x_c),^{(t)} S_2, \ldots$
(or $\pi = (j, \bar{x}, i, x), (i, \bar{x}_c, k, x_c),^{(t)} w$), then
$\rho = (i^{tr}, \bar{x}'_c, i, x'_c), (S_2, \ldots)'$ (resp. $\rho = (i^{tr}, \bar{x}'_c, i, x'_c), w^{tr}$).

(f) If $\pi = (j, \bar{x}, i, x), (i, \bar{x}_c, k, x_c),^{(f)} S_2, \ldots$
(or $\pi = (j, \bar{x}, i, x), (i, \bar{x}_c, k, x_c),^{(f)} w$), then
$cr < k$ and $(i, \bar{x}'_c, k, x'_c) \in BE'$. We define
$\rho = (i^{tr}, \bar{x}'_c, i, x'_c), (i, \bar{x}'_c, k, x'_c), (S_2, \ldots)'$ or
$\rho = (i^{tr}, \bar{x}'_c, i, x'_c), (i, \bar{x}'_c, k, x'_c), w$.

Remark: Since k is in $Col(x_c)$ convex chains $(j, \bar{x}, i, x), (i, \bar{x}_c, k, x_c), x_c$ or $(j, \bar{x}, i, x), (i, \bar{x}_c, k, x_c), \bar{x}_c$ cannot exist.

(g) If $\pi = (j, \bar{x}, i, x), R_1, R_2, \ldots$, then π is adaptive, $R'_1 \in BE'$ (Lemma 1) and we define
$\rho = (i^{tr}, \bar{x}'_c, i, x'_c), (R_1, R_2, \ldots)'$ if $(R_1, R_2, \ldots)'$ is adaptive and
$\rho = (i^{tr}, \bar{x}'_c, i, x'_c), \gamma$ if $(R_1, R_2, \ldots)'$ has the form R'_1, R'^{-1}_1, γ, where γ is adaptive (see Lemma 2 (iii)).

It is not difficult to see that this subcase analysis is complete. In case (a) and (c) it is trivial that ρ is a chain of *GE'*. In case (e) this follows from $k^{tr} = i$, see

2.6 (iv). The convexity of ρ is trivial in case (a). In cases (c), (e), (f) and (g) the convexity of ρ follows from Lemma 2 since π is adaptive, by 2.4. $\square$

Theorem 4: *If GE is non-trivial and proper and if* $GE' = GE_{\preceq} \in$ *Transf*(GE)*, then* GE' *satisfies condition (ii) of definition 4.4.*

Proof: If all natural images of standard equations S of GE are non-degenerate with respect to $\preceq$, then the result follows from Lemma 3. If some standard equations of GE of have degenerate natural images with respect to $\preceq$, then the transformation may be divided into two steps: first, all bases get columns as described in 3.3.1-3.3.3, ignoring the exceptional situation (degenerate boundary equations are not yet erased). We may introduce a column-function $Col'_{(0)}$ and functions $Left'_{(0)}$ and $Right'_{(0)}$ to describe this situation. Then the bases are contracted, as described in 3.3.4, and degenerate boundary equations are erased. Consider now any boundary equation E' of GE'. We may apply the construction of Lemma 2 and Lemma 3 in order to construct a chain $\pi'_{(0)}$ with witness w which starts with E' or E'^{-1} and is convex with respect to $Col'_{(0)}$. Now, in a second step, we erase all degenerate boundary equations of $\pi'_{(0)}$ (note that the first element is not erased). We still have a chain since the two boundaries of degenerate boundary equations coincide and since $Right'(w) = Right'_{(0)}(w)$. The resulting chain π' is again convex: when we ignore last entries, $\chi'(\pi')$ is obtained from $\chi'(\pi'_{(0)})$ by erasing some letters or contracting two letters $\succ\prec$ to $=$. This follows from the fact that in a degenerate boundary equation both boundaries are identical and both bases have the same position with respect to $Col'_{(0)}$, by 2.6 (v). Since $Left'_{(0)}(w) \prec Left'(w)$ still $\chi'(\pi') \in \{\succ, =\}^\star \circ \{=, \prec\}^\star$. $\square$