UNIVERSITY AT ALBANY
State University of New York

College of Computing and Information
Computer Science Department

Unification modulo Synchronous Distributivity

SUNYA-CS-12-01

Siva Anantharaman
LIFO - Université d'Orléans

Serdar Erbatur
Dept. of Computer Science
University at Albany—SUNY

Christopher Lynch
Dept. of Mathematics and Computer Science
Clarkson University

Paliath Narendran
Dept. of Computer Science
University at Albany—SUNY

Michael Rusinowitch
Loria-INRIA Lorraine

Unification modulo Synchronous Distributivity

February 6, 2012

Abstract

Unification modulo the theory defined by a single equation which specifies that a binary operator distributes synchronously over another binary operator is shown to be undecidable. It is the simplest known theory, to our knowledge, for which unification is undecidable: it has only one defining axiom and moreover, every congruence class is finite (so, the matching problem is decidable).

Keywords: Equational unification, Intercell Turing machine, Decidability.

1 Preliminaries

It is well known that unification plays a very major role in all formal deduction mechanisms. Syntactic unification – also known as unification modulo the empty theory – is known to be decidable from around 1930, and optimized algorithms for it are well-known as well [1]. Semantic (or equational) unification is an extension of syntactic unification, to meet the situation where terms in the underlying signature are bound by some given equational theory. Several such theories are of great practical interest, in particular the theories of commutativity, associativity, associativity-commutativity; and decision procedures for unification modulo these theories are well-known from around 1970-1980 [2]. Another equational theory of practical interest is distributivity, which specifies that a binary operator distributes over another binary operator – a typical example being that of multiplication over addition on integers. Unification modulo such a distributivity is known to be decidable [12, 14]. Note that the distributivity of multiplication over addition on integers is 'asynchronous' when used two-sided, in the sense that it then works 'argument-wise' below addition. There are other instances of distributivity for which a different theory is needed; for instance, if B stands for the division operation on nonzero rational numbers and $*$ for multiplication, then the following property is satisfied:

$$\mathcal{E}: \qquad B(u,x) * B(v,y) \;=\; B(u*v, x*y)$$

In contrast with the example mentioned earlier, here the binary operator B distributes over $*$ synchronously, i.e., in parallel on its arguments. Note that the property $\mathcal{E}$ is also satisfied by the RSA-based implementation of the *blind signature scheme* for cryptography [3] (B stands in this case for the product of an integer m with a random number r raised to a given key e, and $*$ is the usual product on integers). Yet another model for $\mathcal{E}$, of practical interest, is the '*Exchange Law for concurrent processes*' as defined in [5]. The equation $\mathcal{E}$ can be turned easily into a terminating rewrite rule, oriented either way; it forms a convergent rewrite system in both cases. The theory defined by this equation $\mathcal{E}$ will also be referred to as $\mathcal{E}$ in the sequel.

Our objective in this paper is two-fold. We first present a semi-decision procedure for the $\mathcal{E}$-unification problem. Section 2 presents an inference procedure for this purpose. A dependency graph is

associated in a natural manner with any $\mathcal{E}$-unification problem $\mathcal{P}$, given in a 'standard form' (see definition below); and it is shown that the problem admits a solution if and only if the dependency graph remains bounded under the inferences. We then show that the $\mathcal{E}$-unification problem is undecidable in general, by reduction from the boundedness problem for deterministic Intercell Turing Machines (ITM), which is known to be undecidable [10]; this is done in Section 5. Such a reduction is rendered possible by suitably encoding the relations between the nodes and the paths on the dependency graph of $\mathcal{P}$ as string rewrite relations (string equations), which can be subsequently interpreted as the transition rules of an ITM[1]; the technical developments needed for this are presented in Sections 3 and 4.

2 A Semi-Decision Procedure for Elementary $\mathcal{E}$-unification

Our signature consists of a (countably infinite) set of variables $\mathcal{X}$ and the two binary symbols B and '$*$'; the variables of $\mathcal{X}$ will be denoted by lower or upper case letters from u or U, to z or Z, with or without suffixes and primes. Note that B and '$*$' are cancellative: by this, we mean that if s_1, t_1, s_2, t_2 are ground terms in normal form, then $B(s_1, t_1) =_{\mathcal{E}} B(s_2, t_2)$ if and only if $s_1 =_{\mathcal{E}} s_2$ and $s_2 =_{\mathcal{E}} t_2$; similarly for '$*$'. (One easy way to show this is to use $\mathcal{E}$ as a rewrite rule $B(u * v, x * y) \rightarrow B(u, x) * B(v, y)$.)

Without loss of generality, the equations of the given unification problem $\mathcal{P}$ are assumed to be in a *standard form,* i.e., in one of the following forms:

$$X =^? V,\ X =^? B(V, Y),\ X =^? V * Y$$

where X, Y, V, Z, are variables. A set of equations is said to be in *dag-solved form* (or *d-solved form*) if and only if they can be arranged as a list

$$x_1 =^? t_1,\ \ldots,\ x_n =^? t_n$$

where (a) each left-hand side x_i is a distinct variable, and (b) $\forall\, 1 \leq i \leq j \leq n$: x_i does not occur in t_j ([8]). A substitution σ' is an *extension* of a substitution σ iff there is a substitution δ such that $\sigma' = \sigma \circ \delta = \sigma \uplus \delta$. The following relations on the variables of $\mathcal{P}$ will be needed in the sequel:

- $U \succ_{r_*} V$ iff there is an equation $U = T * V$
- $U \succ_{l_*} V$ iff there is an equation $U = V * T$
- $U \succ_{r_B} V$ iff there is an equation $U = B(T, V)$
- $U \succ_{l_B} V$ iff there is an equation $U = B(V, T)$
- $U \succ_* V$ iff $U \succ_{r_*} V$ or $U \succ_{l_*} V$
- $U \succ_B V$ iff $U \succ_{r_B} V$ or $U \succ_{l_B} V$

We also define $\succ$ as the union of the four relations above; i.e., $\succ = \succ_* \cup \succ_B$.

The semi-decision procedure for $\mathcal{E}$-unification is given by the following transformation (inference) rules, where $\mathcal{EQ}$ stands for a set of equations in the problem $\mathcal{P}$, the symbol $\uplus$ stands for disjoint set union, and $\cup$ is usual set union.

(1) *Variable Elimination*:

[1]The reader can see that our undecidability proof is influenced by the techniques in [10] and [7].

$$\frac{\{X =^? V\} \uplus \mathcal{EQ}}{\{X =^? V\} \cup [V/X](\mathcal{EQ})} \qquad \text{if } X \text{ occurs in } \mathcal{EQ}$$

(2) *Cancellation on B*:

$$\frac{\mathcal{EQ} \uplus \{X =^? B(V,Y),\ X =^? B(W,T)\}}{\mathcal{EQ} \cup \{X =^? B(W,T),\ V =^? W,\ Y =^? T\}}$$

(3) *Cancellation on '$*$'*:

$$\frac{\mathcal{EQ} \uplus \{X =^? V * Y,\ X =^? W * T\}}{\mathcal{EQ} \cup \{X =^? W * T,\ V =^? W,\ Y =^? T\}}$$

(4) *Splitting*:

$$\frac{\mathcal{EQ} \uplus \{X =^? B(V,Y),\ X =^? W * Z)\}}{\mathcal{EQ} \cup \{X =^? W * Z,\ W =^? B(V_0,Y_0),\ Z =^? B(V_1,Y_1),\ V =^? V_0 * V_1, Y =^? Y_0 * Y_1\}}$$

(5) *Occur-Check*:

$$\frac{\mathcal{EQ}}{FAIL} \qquad \text{if } X \succ^+ X \text{ for some } X$$

An outline of the algorithm is as follows: As long as the rules are applicable, rule (5) ("Occur-Check"), and rule (1) ("Variable Elimination"), are to be applied most eagerly; the cancellation rules (2) and (3) come next. The splitting rule (4) is applied with the lowest priority, i.e., only when no other rule is applicable.

The variable X in the specification of the splitting rule is referred to as a *peak.* In other words, a peak is any variable Z such that $Z \succ_{l_*} U$, $Z \succ_{r_*} V$, $Z \succ_{l_B} X$ and $Z \succ_{r_B} Y$ for some variables U, V, X, Y. Note that rule (4) introduces fresh variables; it also moves some variables from the right side to the left. This may give rise to further applications of (2) and (3). Furthermore, splitting may not terminate. For instance, $U =^? B(M,X)$ and $U =^? M * Z$ will cause an infinite loop, by using rule (4) forever. It is easy to conclude that there will be no solution in such a situation. The proof of correctness for this algorithm is similar to the one in Tiden-Arnborg [14].

We define a relation $\Rightarrow$ between sets of equations $\mathcal{S}$ and $\mathcal{S}'$ as follows: $\mathcal{S} \Rightarrow \mathcal{S}'$ if and only if $\mathcal{S}'$ can be obtained from $\mathcal{S}$ by applying one of the rules (1) – (5).

Lemma 2.1. *Rules* (1) – (5) *are sound and complete for unification modulo* $\mathcal{E}$.

Proof. The soundness of rules (1), (2) and (3) is easily seen[2]. Now, if there is an occur-check cycle of any length for a variable X, then clearly there is no solution for $\mathcal{EQ}$; and it is obvious that rule (5) catches such cycles in the problem if they exist. Hence rule (5) is also sound. We now show, explicitly, that the splitting rule (4) is sound.

Let $\mathcal{S} = \mathcal{EQ} \uplus \{X =^? B(V,Y),\ X =^? W * Z)\}$ where $\mathcal{EQ}$ is a set of equations and V, W, X, Y, Z are variables. And let $\mathcal{S}'$ be

$$\mathcal{EQ} \cup \{X =^? W * Z,\ W =^? B(V_0,Y_0),\ Z =^? B(V_1,Y_1),\ V =^? V_0 * V_1,\ Y =^? Y_0 * Y_1\}$$

where V_0, V_1, Y_0, Y_1 are new variables not occurring in $Var(\mathcal{S})$. Thus $\mathcal{S} \Rightarrow \mathcal{S}'$ by rule (4). We have then:

Claim (i) Any unifier of $\mathcal{S}'$ is a unifier of $\mathcal{S}$: Indeed, suppose θ is a unifier of $\mathcal{S}'$. It is easy to check then that $\theta(X) =_{\mathcal{E}} \theta(B(V,Y))$.

[2]A general proof for soundness of these rules can be found in [13].

Claim (ii) Let σ be a unifier of $\mathcal{S}$. Then there is a substitution σ' such that σ' is an extension of σ and σ' is a unifier of $\mathcal{S}'$: For proving this we reason on terms in normal form under the convergent rewrite system:

$$B(u * v,\, x * y) \;\rightarrow\; B(u, x) * B(v * y).$$

Since σ is a unifier of $\mathcal{S}$, the normal forms of $\sigma(V)$ and $\sigma(Y)$ must be product terms, i.e., terms of the form $s_0 * s_1$ and $t_0 * t_1$ respectively. Then $\sigma(W) = B(s_0, t_0)$ and $\sigma(Z) = B(s_1, t_1)$. Thus $\sigma \circ \{V_0 := s_0, V_1 := s_1, Y_0 := t_0, Y_1 := t_1\}$ is a unifier of $\mathcal{S}'$; this proves (ii).

To show completeness, first note that if the algorithm terminates on $\mathcal{S}$ without failure, the resulting system is in d-solved form. On the other hand, if the algorithm does not terminate, then it has to be because there is infinite splitting — i.e., the splitting rule (4) is applied infinitely often.

Claim (iii) If there is infinite splitting there is no unifier: Assume the contrary. Let θ be a ground unifier of $\mathcal{S}$. If there is infinite splitting, then there is an infinite sequence of variables $V_i = V_{i_1} \succ V_{i_2} \succ \ldots$ where $V_i \in Dom(\theta)$. From what was shown above, there must also be an infinite sequence of unifiers $\theta = \theta_1, \theta_2, \ldots$ where each unifier is an extension of the previous one. But this leads to a contradiction, since if γ is a unifier, $V, V' \in Dom(\gamma)$ and $V \succ V'$, then $|\gamma(V)| > |\gamma(V')|$. □

A sufficient condition for an $\mathcal{E}$-unification problem to be unsatisfiable can be formulated as cycle checking, on suitably defined relations on the variables of $\mathcal{P}$ over some of the models for the equation $\mathcal{E}$, see **Appendix**.

3 From $\mathcal{E}$-Unification Problems to Thue Systems

Before we give the reduction proving the undecidability of this unification problem, we need a few preliminaries.

As explained in the previous section, the set of variables in a problem could get larger since fresh variables may be created when splitting occurs. If a variable X is split, than we add the equation $X = X_0 * X_1$ to the problem. In general, new variables may be split further and starting from a variable X we may obtain a variable X_β where β is a string of 0s and 1s. We shall agree that the general discipline for creating new variables is specified as: $X_\beta = X_{\beta 0} * X_{\beta 1}$, where $\beta \in \{0,1\}^*$. Note that if $\beta = \lambda$, the empty string, then $X_\beta = X$, an original variable in the problem. For a set of variables V, we define $\overline{V} = \{X_\beta \mid X \in V,\ \beta \in \{0,1\}^*\}$ to denote the set of all variables which may originate from V through splitting. In the next section we define our dependency graph notation and describe splitting and variable elimination in a graph setting.

3.1 The Dependency Graph

It is common to represent the problems by dependency graphs induced by the relations ($\succ_{l_*}$ etc.) among variables. Each node corresponds to a variable and each directed edge is labeled w.r.t. the relation among the variables in the nodes. Interpretation of the unification problems through the relations among variables in a graph setting was used in [14]. Here we have four types of edges in the dependency graph: l_*, r_*, l_B and r_B. If two variables X and Y are related through $\succ_{l_*}$, then nodes induced by X and Y are connected by a directed edge labeled as l_*; similarly for the other relations. No edge on the graph corresponds to an equality. For instance, for the problem given as:

$U =^? V,\ U =^? U_0 * U_1,\ U =^? B(X,Y)$, the dependency graph is given in Figure 1; note that V does not appear as a node on this graph.

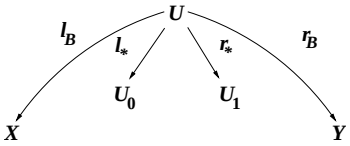

Figure 1: Graph for: $U =^? V,\ U =^? U_0 * U_1,\ U =^? B(X,Y)$

Let $\mathcal{S}$ be the initial set of equations of the problem, and let G_0 be its dependency graph. Dependency graphs are not stable; they get updated each time an inference rule applies; we thus we get a sequence of graphs G_0, G_1, G_2, ... Recall, in particular that rule (1) is applied eagerly, e.g., when there is an equation of the form $U =^? V$. The consequence of its application is that U then merges into V, more precisely V replaces U in the problem as well as on the dependency graph. The problem just considered thus becomes:

$$U =^? V,\ V =^? U_0 * U_1,\ V =^? B(X,Y).$$

Its dependency graph is obtained from the one in Figure 1, by changing the label of the node U to V. It is important to note that the variable U has not been deleted from the problem, which still contains the equation $U =^? V$; the only change is that V now represents U on the graph. In intuitive terms, we shall say: on applying rule (1), two (or more) nodes merge on the dependency graph, and two (or more) paths merge by merging their end nodes; and any variable of the problem has a unique representative node on the dependency graph, up to variable equality. Alternatively, one could label the nodes of the graph with the equivalence classes of the variables of the problem, the equivalence being defined up to variable equality.

To suit our purposes in the sequel, we agree to shorten the labels of the edges of the dependency graph as follows: We replace l_* by 0 and r_* by 1; and we also replace l_B by L and r_B by R.

The splitting rule (4) is the only rule that adds nodes to the dependency graph, and new edges joining these new nodes. When a variable is split, 0- and 1-edges are added; the other equations introduced by rule (4) cause L- and R-edges to be added too. Thus, for the problem just mentioned above, (after having applied rule (1)) we apply splitting, and the problem thus derived is:

$$U =^? V,\ V =^? U_0 * U_1,\ X =^? X_0 * X_1,\ Y =^? Y_0 * Y_1,\ U_0 =^? B(X_0,Y_0),\ U_1 =^? B(X_1,Y_1)$$

The dependency graph for the problem thus derived is given in Figure 2, using the short labels for its edges. Note that the edges from (U or) V to X and Y have been dropped out from the earlier graph (as well as the equations to which they corresponded).

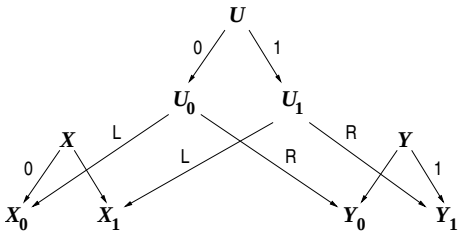

Figure 2: Graph for the problem of Figure 1 after applying rules (1) and (4)

It is in general necessary to use other rules again after applying rule (4). For instance, consider the problem given as: $\{X =^? B(Y,Z),\ X =^? X_0 * X_1,\ X_0 =^? B(U,V)\}$. Variables Y and Z split and we

obtain $X_0 =^? B(Y_0, Z_0)$ as one of the resulting equations. Therefore it is now necessary to apply first rule (2), followed by rule (1) to the equations $U =^? Y_0$ and $V =^? Z_0$ thereby derived.

For the purpose of proving the undecidability of $\mathcal{E}$-unification, we slightly modify our view of the dependency graph representation. Mainly, we don't explicitly delete any nodes or edges from the graph – all we do is to merge nodes. This leads to a more general vision of the dependency graph, that could be said to be the *relation graph*. We formalize these ideas as follows. Since nodes are merged, we assume that a node may have several labels. (See Figure 3.)

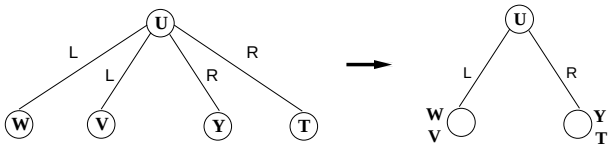

Figure 3: Applying the cancellation rule (2)

Thus there is an *onto* function ϕ defined from $\overline{Var(\mathcal{S})}$ to $V(G)$, the set of vertices of the graph. That is, each variable points to exactly one node in the graph but a node can be pointed to by more than one variable. Now using ϕ define a relation as a tuple (X, μ, Y) where X and Y are variables and μ is either L or R. In other words, (X, μ, Y) holds if and only if $\phi(X)$ and $\phi(Y)$ are connected with an edge labelled as μ.

Given a problem $\mathcal{P}$, let $G = G_\infty$ be the set of *persistent* nodes and edges (i.e., those not dropped out) in the sequence $G_0 \cup G_1 \cup \ldots$, of graphs, updated along the inferences. We now characterize this graph in terms of string relations (equalities) over $\overline{Var(\mathcal{S})} \cup \{L, R\}$. If X and Y are two distinct variables such that $\phi(X) = \phi(Y)$, then we write $X =_G Y$. If there is a directed path $\Gamma \in \{L, R\}^*$ from $\phi(X)$ to $\phi(Y)$ on G, then we express this 'path relation' between X and Y as $\Gamma X =_G Y$. For paths of 0- and 1-edges, we define a similar relation: if there is a directed path $\beta \in \{0, 1\}^*$ from $\phi(X)$ to $\phi(Y)$ on G, then we write $X_\beta =_G Y$.

We denote the length of any string $\Pi \in \{L, R\}^*$ (resp. $\beta \in \{0, 1\}^*$) as $|\Pi|$ (resp. as $|\beta|$). We say that a variable $X \in \mathcal{X}$ "exists" (in G) if there is a node in G_∞ with X as one of its labels.

Lemma 3.1. *(i) Let $\nu \in \{L, R\}$, $U, Y \in \mathcal{X}$ such that $\nu U =_G Y$. If U_0 exists (and also U_1, i.e., U splits), then $\nu U_0 =_G Y_0$ and $\nu U_1 =_G Y_1$.*

(ii) Let $\Pi \in \{L, R\}^+$, $U, Y \in \mathcal{X}$ such that $\Pi U =_G Y$. If U_0 exists (and also U_1, i.e., U splits), then $\Pi U_0 =_G Y_0$ and $\Pi U_1 =_G Y_1$.

(iii) Let $\Pi \in \{L, R\}^+$, $\beta \in \{0, 1\}^*$ such that $\Pi X =_G Y_\beta$. If X_0 exists (and also X_1, i.e., X splits), then $\Pi X_0 =_G Y_{\beta 0}$ and $\Pi X_1 =_G Y_{\beta 1}$.

Proof. Assertion (i): Without loss of generality, we assume $\nu = L$. There are two cases to consider. First, U itself splits as $U = U_0 * U_1$ and the result follows immediately. Otherwise (from the notion of relation graph defined above) there must be a variable V of $\mathcal{P}$ such that $\phi(U) = \phi(V)$ and V was a peak at some point. Thus V_0 exists, so $V_0 =_G U_0$ by cancellativity and we get $LV_0 =_G Y_0$. □

Assertion (ii): By induction on the length of Π. Let $n = |\Pi|$. The case $n = 1$ follows from Assertion (i). So suppose that $\Pi U = Y$ where $|\Pi| \geq 2$; and suppose U_0 exists. Without loss of generality, let $\Pi = \Pi' L$ and let $X = LU$. Then $X_0 = LU_0$ and $X_1 = LU_1$. Thus, we have $\Pi' X = Y$, and X_0 exists: by induction hypothesis, we get $\Pi' X_0 = Y_0$ and $\Pi' X_1 = Y_1$. □

Assertion (iii): By induction on the length of β. Let $n = |\beta|$. For $n = 0$ the result follows from Assertion (ii). Suppose $n \geq 1$, i.e., $\Pi X = Y_\beta$ where $|\beta| \geq 1$, and suppose X_0 exists (i.e., X splits). Without loss of generality, let $\Pi = \Pi' L$ and let $Z = LX$. Thus $\Pi X = \Pi' Z = Y_\beta$. By Assertion (ii) we know that Z_0 exists. Since $|\Pi'| < |\Pi|$, we have, by induction hypothesis, that $\Pi' Z_0 = Y_{\beta 0}$. Therefore $\Pi' L X_0 = \Pi X_0 = Y_{\beta 0}$; we are done. □

Lemma 3.2. *Let $\Pi \in \{L, R\}^+$, $\alpha, \beta \in \{0, 1\}^*$ and X, Y such that $\Pi X =_G Y_\beta$. If X_α exists then $\Pi X_\alpha =_G Y_{\beta\alpha}$.*

Proof. By induction on the length of α, and Lemma 3.1 □

3.2 Thue systems associated with $\mathcal{E}$-unification problems

We henceforth speak of any $\mathcal{E}$-unification problem as a set of equations in standard form, often denoted as $\mathcal{S}$. With any given $\mathcal{E}$-unification problem $\mathcal{S}$, we shall associate a Thue system (i.e., string rewrite system), and subsequently relate the Thue congruence thus obtained, to the path relations on the dependency graph of $\mathcal{S}$, as defined in the previous section.

Let $V = Var(\mathcal{S})$ be the set of variables of the given problem $\mathcal{S}$. The alphabet over which the Thue system is defined is $\Sigma = V \cup \{L, R\} \cup \{0, 1\}$. We obtain string equations from an $\mathcal{E}$–unification problem as follows. For equations of the form $X =^? B(Y, Z)$ we create string equations $LX = Y$ and $RX = Z$. For $X =^? U * Y$, we form $X0 = U$ and $X1 = Y$. (Notice the connection between these and the binary relations defined in Section 3 - e.g. if $X \succ_{l_B} Y$, then $LX = Y$.) Let S_{Th} denote the set of string equations (the Thue system) thus associated with $\mathcal{S}$. Every such string equation is either of the form $\mu X = Y$ for $\mu \in \{L, R\}$, or of the form $X\nu = Y$ for $\nu \in \{0, 1\}$, with $X, Y \in V$. There is a close connection between the congruence on strings over Σ, modulo these string equations, denoted by $=_{S_{Th}}$, and the congruence in the graph context, denoted by $=_G$, that was introduced in the previous section, on the dependency graph G of the problem $\mathcal{S}$.

The next couple of results show the relation between $=_{S_{Th}}$ and $=_G$; it is assumed in their statements that $X, Y \in Var(\mathcal{S})$, $\Pi \in \{L, R\}^*$, $\alpha, \beta \in \{0, 1\}^*$:

Proposition 3.3. For every X, Y, Π, α, β, $\Pi X\alpha =_{S_{Th}} Y\beta$ if and only if there exists $\alpha', \beta', \gamma \in \{0, 1\}^*$ such that $\alpha = \alpha'\gamma$, $\beta = \beta'\gamma$ and $\Pi X_{\alpha'} =_G Y_{\beta'}$.

Proof. The "if" part is easy to show. For the other direction, we use induction on n, which is the number of derivation steps to get $\Pi X\alpha =_{S_{Th}} Y\beta$. If $n = 1$, i.e., the equation $\Pi X\alpha =_{S_{Th}} Y\beta$ in a single S_{Th}-rewrite step, then there are three cases:

case (i) $\Pi \in \{L, R\}$, $\Pi X = Y$ is an equation in S_{Th} and $\alpha = \beta$; here, $\alpha' = \beta' = \gamma = \lambda$, where λ is the empty string.

case (ii) $\Pi = \lambda$, $X\nu = Y$ is an equation in S_{Th} for some $\nu \in \{0, 1\}$, and $\alpha = \nu\beta$; here, $\alpha' = \nu, \gamma = \beta$, and $\beta' = \lambda$.

case (iii) $\Pi = \lambda$, $X = Y\nu$ is an equation in S_{Th} for some $\nu \in \{0, 1\}$, and $\beta = \nu\alpha$; here, $\alpha' = \lambda, \beta' = \nu$, and $\gamma = \alpha$.

Now suppose $n > 1$ and let $\Pi X\alpha =_{S_{Th}} Y\beta$ be derived in n S_{Th}-steps. We consider the leftmost S_{Th}-step. As before, three cases have to be considered (with $\mu \in \{L, R\}$ and $\nu \in \{0, 1\}$):

(i) The equation used is of the form $\mu X = Z$;

(ii) The equation used is of the form $X\nu = Z$;

(iii) The equation used is of the form $X = Z\nu$.

In case (i), $\Pi = \Pi'\mu$ for some $\Pi' \in \{L,R\}^*$. Then: $\Pi X\alpha = \Pi'\mu X\alpha =_{S_{Th}} \Pi' Z\alpha =_{S_{Th}} Y\beta$. Thus $\Pi' Z\alpha =_{S_{Th}} Y\beta$ in $(n-1)$ steps and by the induction hypothesis there exists γ such that $\alpha = \alpha'\gamma$, $\beta = \beta'\gamma$ and $\Pi' Z_{\alpha'} =_G Y_{\beta'}$; we are through in this case.

In case (ii), $\alpha = \nu\omega$ for some $\omega \in \{0,1\}^*$. Then: $\Pi X\alpha = \Pi X\nu\omega =_{S_{Th}} \Pi Z\omega =_{S_{Th}} Y\beta$. Thus $\Pi Z\omega =_{S_{Th}} Y\beta$ in $n-1$ steps and by the induction hypothesis there exists γ such that $\omega = \alpha''\gamma$, $\beta = \beta'\gamma$ and $\Pi' Z_{\alpha''} =_G Y_{\beta'}$. Since $Z =_G X_\nu$, $Z_{\alpha''} =_G X_{\nu\alpha''}$ and thus $\Pi X_{\nu\alpha''} =_G Y_{\beta'}$, where $\alpha = \nu\alpha''\gamma$; we are through in this case.

In case (iii), $\Pi X\alpha =_{S_{Th}} \Pi Z\nu\alpha =_{S_{Th}} Y\beta$. Thus $\Pi Z\nu\alpha =_{S_{Th}} Y\beta$ in $n-1$ steps and by the induction hypothesis there exists γ such that $\nu\alpha = \alpha'\gamma$, $\beta = \beta'\gamma$ and $\Pi Z_{\alpha'} =_G Y_{\beta'}$. Now we need to consider two subcases, namely $\alpha' = \lambda$ and $\alpha' \neq \lambda$.

Subcase where $\alpha' = \lambda$: then $\Pi Z =_G Y_{\beta'}$ and $\beta = \beta'\nu\alpha$. By assertion (iii) of Lemma 3.1 we get $\Pi Z_\nu =_G Y_{\beta'\nu}$. Since $X =_G Z_\nu$ we are done.

Subcase where $\alpha' \neq \lambda$; here $\alpha' = \nu\omega$ for some $\omega \in \{0,1\}^*$. Thus we have $\nu\alpha = \alpha'\gamma = \nu\omega\gamma$ and $\alpha = \omega\gamma$. Therefore we get $\Pi Z_{\nu\omega} =_G \Pi X_\omega =_G Y_{\beta'}$ and hence the result follows. □

Corollary 3.4. *For every X, Y, Π, β, $\Pi X =_{S_{Th}} Y\beta$ if and only if $\Pi X =_G Y_\beta$.*

Proof. We prove this again by induction on the number of S_{Th}-steps in deriving the proof for $\Pi X =_{S_{Th}} Y\beta$. We consider two cases, depending on the *first* (leftmost) step:

Case (a) There is an equation $X = LU$ in S_{Th}: In this case, the derivation sequence is: $\Pi X =_{S_{Th}} \Pi LU =_{S_{Th}} Y\beta$. Since $\Pi LU =_{S_{Th}} Y\beta$ has a shorter derivation, we have $\Pi LU =_G Y_\beta$ by the induction hypothesis, and the result follows.

Case (b) $\Pi = \Pi' L$ and there is an equation $LX = U$ in S_{Th}: In this case, the derivation sequence is $\Pi X = \Pi' LX =_{S_{Th}} \Pi' U =_{S_{Th}} Y\beta$. Since the derivation $\Pi' U =_{S_{Th}} Y\beta$ is shorter, we get $\Pi' U =_G Y_\beta$. Since $U =_G LX$ we are done. □

Let $\mathcal{S}$ be an $\mathcal{E}$-unification problem and S_{Th} its associated Thue system. We get back to the variables which originate from $Var(\mathcal{S})$ through splitting, i.e., the set $\overline{Var(\mathcal{S})}$. We now relate these with the path relation on the graph of $\mathcal{S}$ and the Thue congruence associated with $\mathcal{S}$. For a variable $X \in Var(\mathcal{S})$, we define its extent[3] $ext(X)$ as follows:

$$ext(X) = \{\, \Pi \in \{L,R\}^* \mid \exists\, Y \in Var(\mathcal{S}) \wedge \beta \in \{0,1\}^* \text{ such that } S_{Th} \vdash \Pi X = Y\beta \,\}$$

The finiteness of $ext(X)$ for every X is closely connected to the unifiability of the problem. If $ext(X)$ is infinite for X, it obviously means that X splits infinitely. The following result is given without proof since it is (now) obvious:

Proposition 3.5. An $\mathcal{E}$-unification problem $\mathcal{S}$ is solvable if and only if no failure rule applies and $ext(X)$ is finite for every X in $Var(\mathcal{S})$.

We define this as a general concept for Thue systems. Let T be a Thue a system with the signature Σ. Let Δ be a nonempty subset of Σ. T is said to have *finite Δ-span* if and only if $\forall q \in \Delta$, $ext(q)$ is finite where

[3]This follows the definition of *extent* by Jahama and Kfoury [7].

$$ext(q) = \{\, \Pi \in (\Sigma \setminus \Delta)^* \mid \exists\, q' \in \Delta \ \wedge\ \beta \in (\Sigma \setminus \Delta)^* \text{ such that } \Pi q \leftrightarrow_T^* q'\beta \,\}$$

4 Thue Systems and Intercell Turing Machines

We give a review of relevant literature, mainly based on the notation used in [10]. An Intercell Turing Machine (ITM) is defined as a triple $M = \langle Q, \Sigma, \delta \rangle$, where Q is a set of states, Σ is a finite tape alphabet, and δ is a transition relation defined as $\delta \subseteq Q \times D \times \Sigma \times \Sigma \times Q$. Here D points to the direction of the move of the tape head (assumed placed between two tape cells) and is one of $\{-1, +1\}$. An instantaneous description (ID) of M is defined as a quadruple $\langle w_1, q, m, w_2 \rangle$ where q is the current state of the machine, $w_1 w_2$ is the string over Σ that forms the current tape content, m is an integer, and the header is between the cells $m-1$ and m, and it also separates w_1 and w_2[4] . A move of M, from one ID to another, is denoted as a relation $\vdash_M$ formally defined as follows, where $s, t \in \Sigma,\ w_1, w_2 \in \Sigma^*$ and $q_1, q_2 \in Q$, and q_1 is the current state:

- **left-move**: For $\langle q_1, -1, s, t, q_2 \rangle \in \delta, \quad \langle w_1 s, q_1, m, w_2 \rangle \vdash_M \langle w_1, q_2, m-1, t w_2 \rangle$
- **right-move**: For $\langle q_1, +1, s, t, q_2 \rangle \in \delta, \quad \langle w_1, q_1, m, s w_2 \rangle \vdash_M \langle w_1 t, q_2, m+1, w_2 \rangle$

An ITM is said to be *deterministic* if and only if:

(i) the set of states Q splits as left-move and right-move states Q_l and Q_r so that $\delta \subseteq (Q_l \times \{-1\} \times \Sigma \times \Sigma \times Q) \cup (Q_r \times \{+1\} \times \Sigma \times \Sigma \times Q)$; and

(ii) δ is partial function from $Q \times D \times \Sigma$ to $\Sigma \times Q$.

This implies that there is at most one possible move from any given ID of a deterministic ITM: a left-move or a right-move.

The symmetric closure of an ITM $M = \langle Q, \Sigma, \delta \rangle$ is defined as the ITM $M_s = \langle Q, \Sigma, \delta_s \rangle$ where:

$$\delta_s = \delta \cup \{\langle q_1, -x, a, b, q_2 \rangle \mid \langle q_2, x, b, a, q_1 \rangle \in \delta\}$$

An ITM M is said to be *symmetric* iff $M = M_s$ holds. An ITM M is said to be *bounded* iff there exists a positive integer n such that for any arbitrary ID I of M, the number of different IDs reachable by M from I is at most n.

The following results on the boundedness problem are shown in [10] by using the ideas from [6].

Lemma 4.1. *It is undecidable to check whether a deterministic ITM is bounded.*

Lemma 4.2. *A deterministic ITM M is bounded if and only if its symmetric closure M_s is bounded.*

Corollary 4.3. *Given a deterministic ITM M, it is undecidable to check whether its symmetric closure M_s is bounded.*

Let $M = \langle Q, \Sigma, \delta \rangle$ be a deterministic ITM with tape alphabet $\Sigma = \{0, 1\}$. We shall use L, R to represent tape symbols $0, 1$ respectively, to the left of the tape head. Under such a vision, any transition of M can be expressed as a string rewrite rule of the form:

$$q_1 a \ \sim\ b q_2$$

[4]We refer the reader ro Turing's work [15] for similar ideas of 'left-facing' and 'right-facing' internal configurations, i.e., IDs here. Following Turing, Gurevich and Lewis use the same idea, defining 'left-looking' and 'right-looking' states [4].

where $a \in \{0, 1\}$, $b \in \{L, R\}$, and $\sim \in \{\leftarrow, \rightarrow\}$. For instance, $q_2\, t \leftarrow L\, q_1$ represents the left-move $\langle q_1, -1, 0, t, q_2\rangle$; and $q_1\, s \rightarrow R\, q_2$ represents the right-move $\langle q_1, +1, s, 1, q_2\rangle$ of the ITM.

Let R_M be the string rewrite system consisting of all these rules and let S_M be the Thue system obtained by symmetrizing the rewrite rules, i.e., making them bidirectional[5]. Observe that one can also get S_M by first getting the symmetric closure of M and then getting the string rewrite rules.

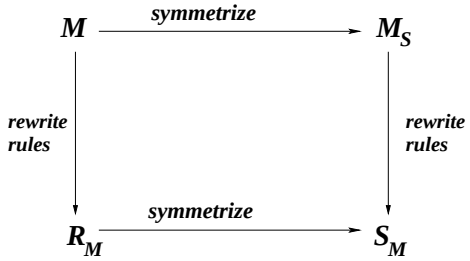

Figure 4: Commuting diagram for S_M

The extent of a state q in the ITM M is defined in terms of the system S_M:

$$ext(q) = \{\, \Pi \in \{L, R\}^* \mid \exists\, q' \in Q \ \wedge\ \beta \in \{0,1\}^* \text{ such that } S_M \vdash \Pi q = q'\beta \,\}$$

Lemma 4.4. *Let $M = \langle Q, \Sigma, \delta\rangle$ be a deterministic ITM. Then M is bounded if and only if $ext(q)$ is finite for every state q in M.*

Proof. If $ext(q)$ is infinite for some state q in M, then clearly M is unbounded. On the other hand, suppose M is unbounded, and assume that $ext(q)$ is finite for every $q \in Q$. Let k be the length of the longest string that appears in any $ext(q)$. (Recall that Q is finite.) Since M is unbounded, there are configurations C and C' such that $C = \langle \Pi,\, q,\, m,\, w\beta\rangle$ and $C' = \langle \Pi\mathcal{B},\, q',\, m+p,\, \beta\rangle$, such that C' is reachable from C, with $p = |w| = |\mathcal{B}| > k$, and the header never moves left past the m^{th} cell, nor right past the $(m+p)^{th}$ cell. Then the configurations $\langle \epsilon,\, q,\, m,\, w\rangle$ and $\langle \mathcal{B},\, q',\, m+p,\, \epsilon\rangle$ are reachable from each other as well by definition, and thus $qw \leftrightarrow^*_{S_M} \mathcal{B}q'$. Since $|\mathcal{B}| > k$ this contradicts the assumption that k is the length of the longest string in any $ext(q)$. □

Corollary 4.5. *Let $M = \langle Q, \Sigma, \delta\rangle$ be a deterministic ITM. Then M is bounded if and only if S_M has a finite Q-span.*

5 The Undecidability of Elementary $\mathcal{E}$-Unification

The undecidability result is by a reduction of the boundedness problem for deterministic ITMs. We shall proceed as follows: for each transition t of M we add two new (dummy) states along with their transitions — the reason for this will be clear later — but making sure that the resulting system, denoted M', is still deterministic, and furthermore, M' is bounded if and only if M is bounded. We then symmetrize M' to obtain M'_s. Thus M'_s is bounded if and only if M' is bounded (by Lemma 4.2) if and only if M is bounded. We shall finally show how to construct an $\mathcal{E}$-unification problem such that the problem is solvable if and only if M'_s is bounded.

[5]Such a Thue system is sometimes called a "symmetric semi-Thue system"!

For any deterministic ITM $M = \langle Q, \Sigma, \delta \rangle$ with tape alphabet $\Sigma = \{L, R\} \cup \{0, 1\}$, we construct another deterministic ITM M' based on M. For that, we first introduce the following notation: for $a \in \{0, 1\}$, we set $1 - a = 1$ if $a = 0$, and $1 - a = 0$ if $a = 1$. Analogously, for $b \in \{L, R\}$ we let $\overline{b} = R$ if $b = L$ and $\overline{b} = L$ if $b = R$. Recall that in Section 4 a move of M was specified as $q_1 a \sim b q_2$ where $\sim$ is either $\leftarrow$ or $\rightarrow$. Let us consider first the (rightward) move $q_1 a \rightarrow b q_2$; we then add a left-move state and a right-move state, denoted as w' and w respectively, for each transition of M along with the following transitions:

$$\begin{aligned} q_1\,(1-a) &\leftarrow b w' \\ w a &\rightarrow \overline{b} q_2 \\ w\,(1-a) &\rightarrow \overline{b} w' \\ w\,(1-a) &\leftarrow \overline{b} w' \end{aligned}$$

The construction is the same for the leftward move $q_1 a \leftarrow b q_2$: the same set of states and transitions get added, i.e., the direction of the move does not affect the modifications. In all other cases, different state pairs are added to M' for different transitions of M; one could thus adopt the notation w_t and w'_t corresponding to every transition t in M. So the extension M' is defined as $\langle Q', \Sigma, \delta' \rangle$, where:

$$Q' = Q \cup \{\, w_t, w'_t \mid \text{ for each corresponding transition } t \text{ of } M \,\}$$

and δ' consisting of δ plus new transitions induced by the extra moves above. M' is also deterministic because every state in Q' has only one possible move. Note that M and M' have the same tape alphabet Σ with four symbols $L, R, 0, 1$. (See illustrative Figure 5 for the move $q_1 0 \rightarrow L q_2$ of M, and the corresponding part in the extended ITM M'.)

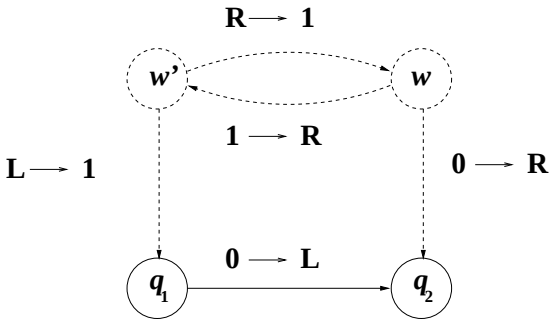

Figure 5: Extension M' of a deterministic ITM M. Edges/Nodes not in M are dashed.

Lemma 5.1. *M is bounded if and only if M' is bounded.*

Proof. The "if" part is trivial, due to the fact that M' includes all transitions of M. For the "only if" part, we need to show that any ID involving w or w' can reach finitely many different IDs. Note that the move $w\,(1-a) \leftrightarrow \overline{b} w'$ will cause only one different ID for both types of IDs. The moves back and forth between IDs corresponding to $w\,(1-a)$ and $\overline{b} w'$ don't affect the number of different IDs reachable from them. In the remaining transitions w reaches an ID with q_2 and w' to an ID with q_1. Then we are done since we assume M is bounded and q_1 and q_2 are from M. □

Both M and M' are deterministic but not symmetric. For our purpose we symmetrize M'. This is done by first finding symmetric closure M'_s of M' along with the transition set δ'_s as defined in

Section 4. Thus we get $M'_s = \langle Q', \Sigma, \delta'_s \rangle$. By Lemma 4.2 a deterministic ITM is bounded if and only if its symmetric closure is also bounded. Thus

Lemma 5.2. *M is bounded if and only if M'_s is bounded.*

Let $S_{M'}$ be the Thue system for the symmetric ITM M'. For each transition $q_1 a \sim b q_2$ of M, $S_{M'}$ will contain string equalities of the following forms:

$$\begin{aligned} q_1 a &\leftrightarrow b q_2 \\ q_1 (1 - a) &\leftrightarrow b w' \\ w a &\leftrightarrow \overline{b} q_2 \\ w (1 - a) &\leftrightarrow \overline{b} w' \end{aligned}$$

Obviously $S_{M'}$ is an extension of S_M, the Thue system of M.

We now show how each string equality can be simulated using unification problems. Let $\mathcal{S}$ be the set of equations that we create. We first look at original transitions in M. Note that depending on what a and b are in a transition t we have four possible types of moves. Then, we construct M' as described above. The variables of the unification problem are exactly the *states* of M'. Therefore for each possible pair (a, b) that t involves a unique equation modulo $\mathcal{E}$ with variables q_1, q_2, w and w' is constructed as shown below:

Case 1: If the transition of M is $q_1 0 \leftrightarrow L q_2$, then we (effectively) add the following equation to $\mathcal{S}$:

$$B(q_1, w) =^?_{\mathcal{E}} q_2 * w' \tag{1}$$

Since we assume that $\mathcal{S}$ is in standard form in our unification procedure, we transform this into standard form by using another new variable. Hence we create the equations

$$u =^?_{\mathcal{E}} B(q_1, w),\ u =^?_{\mathcal{E}} q_2 * w'$$

where u is fresh. Note that if we apply the splitting rule to (1), we get

$$u = q_2 * w',\ q_1 = q_{10} * q_{11},\ w = w_0 * w_1,\ q_2 = B(q_{10}, w_0),\ w' = B(q_{11}, w_1) \tag{2}$$

Therefore we see that the string equation (and hence the move of M) indeed corresponds to one of the relations among variables in (2). Note that $q_1 0 = q_{10}$ and $L q_2 = q_{10}$. This is the motivation behind the reduction. In addition there are three other similar relations which can be observed in (2).

$$\begin{aligned} q_1 1 &= L w' \\ w 0 &= R q_2 \\ w 1 &= R w' \end{aligned}$$

The motivation for the construction of M' and M'_s should now be clear. In fact, the equalities above are included in $S_{M'}$. Recall that in Section 3.2 we defined the Thue system for a $\mathcal{E}$-unification problem $\mathcal{S}$ and denoted it as S_{Th}. Hence the congruence induced by $S_{M'}$ is a subset of that induced by S_{Th}, or, $\leftrightarrow^*_{S_{M'}} \subset \leftrightarrow^*_{S_{Th}}$.

Case 2: If the move of M is $q_1 1 \leftrightarrow L q_2$, then we add:

$$u =^?_{\mathcal{E}} B(q_1, w),\ u =^?_{\mathcal{E}} w' * q_2$$

Case 3: If the move of M is $q_1 1 \leftrightarrow Rq_2$, then add:

$$u =^?_{\mathcal{E}} B(w, q_1),\ u =^?_{\mathcal{E}} w' * q_2$$

Case 4: If the move of M is $q_1 0 \leftrightarrow Rq_2$, then add:

$$u =^?_{\mathcal{E}} B(w, q_1),\ u =^?_{\mathcal{E}} q_2 * w'$$

We proceed now to show the following in several steps: S_{Th} has finite $Var(\mathcal{S})$-span if and only if $S_{M'}$ has finite Q-span; the "if" part is easy to prove:

Lemma 5.3. *If S_{Th} has finite $Var(\mathcal{S})$-span then $S_{M'}$ has finite Q-span.*

Proof. Trivial since the congruence induced by $S_{M'}$ is a subset of the congruence induced by S_{Th} and $Q \subseteq Var(\mathcal{S}) = Q \cup \{u_t \mid \text{ for each transition } t \text{ in } M\}$. □

We next prove the converse of Lemma 5.3, i.e., that finite Q-span of $S_{M'}$ implies finite $Var(\mathcal{S})$-span of S_{Th}. For that we consider the string rewrite systems which are obtained by orienting equations in S_{Th} and $S_{M'}$ according to the order defined below. We then apply Knuth-Bendix completion to those systems as specified in [9]. It was shown there that those final string rewrite systems are possibly infinite and *lex-confluent*[6]. Hence for any Thue system T there exists a lex-confluent system equivalent to it.

Let Σ' be the alphabet of $S_{M'}$. Note that $\Sigma' = \Sigma \cup Q$. Let $\mathcal{U}$ be the set of variables u_t that are added to ensure that the unification problem $\mathcal{S}$ is in standard form. Then the alphabet of S_{Th} is $\Sigma'' = \Sigma' \cup \mathcal{U}$. Equations in $S_{M'}$ can be oriented with the help of a length+lexicographical ordering on strings in Σ'^* induced by a total ordering $\succ$ on Σ'. One such ordering can be defined as $x > y$ if and only if:

(1) Either $|x| > |y|$;

(2) Or $|x| = |y|$, $x = ax'$, $y = by'$ where $a, b \in \Sigma'$; and,

either $a \succ b$ or ($a = b$ and $x' > y'$).

with the following assumptions on the symbol ordering $\succ$:

Symbols in Σ are ordered as $L \succ R \succ 0 \succ 1$

Symbols in Q are ordered as $q_1 \succ q_2 \succ \ldots$

For any $X \in \Sigma$ and $Y \in Q$, we assume $X \succ Y$.

Let $C^*_{M'}$ be the resulting confluent system. A total length+lexicographic ordering on Σ'' can be defined similarly with an additional assumption such that the variables u_t are greater than the symbols in Q. Thus every equation in S_{Th} will be oriented in such a way that the variables u_t are on the left. We denote by $\overrightarrow{S_{Th}}$ this new rewrite system.

Lemma 5.4. *$C^*_{M'} \cup \overrightarrow{S_{Th}}$ is confluent.*

Proof. It is not hard to see the forms of the rules in both subsystems. $\overrightarrow{S_{Th}}$ has rules of the form $\Pi u_t \to q'$ or $u_t \beta \to q$ for u_t and $\Pi \in \{L, R\}$ and $\beta \in \{0, 1\}$. On the other hand, $C^*_{M'}$ (possibly infinite) includes the rules like $\Pi q \to q'\beta$ with Π and β defined as before. We already know that $C^*_{M'}$ is confluent, i.e., any critical pair in $C^*_{M'}$ is joinable. Note that left-hand sides in $\overrightarrow{S_{Th}}$ do overlap and

[6]This is abbreviation for length+lexicographic confluence, which was introduced in [9].

thus give rise to critical pairs. But these critical pairs are joinable by the rules of $C^*_{M'}$. For instance suppose $\overrightarrow{S_{Th}}$ has $Lu_t \to q$ and $u_t 0 \to q'$. Here $Lu_t 0$ is an overlap which gives rise to the critical pair $(Lq', q0)$. But we have $Lq' \to q0$ already in $C^*_{M'}$ by definition. □

To prove the converse of Lemma 5.3, we assume that $S_{M'}$ has finite Q-span in the sense we explained earlier; we begin by showing that that the state variables have finite extent in both S_{Th} and $S_{M'}$.

Lemma 5.5. *Let $\Pi \in \{L, R\}^*$ and $\beta \in \{0,1\}^*$ and $q_1, q_2 \in Q$ be two states in M', where $q_1, q_2 \in Var(\mathcal{S})$. Then $\Pi q_1 \leftrightarrow^*_{S_{M'}} q_2\beta$ if and only if $\Pi q_1 \leftrightarrow^*_{S_{Th}} q_2\beta$.*

Proof. The "only if" part is obvious since $\leftrightarrow^*_{S_{M'}}$ is subsumed by $\leftrightarrow^*_{S_{Th}}$. Conversely, suppose $\Pi q_1 \leftrightarrow^*_{S_{M'}} q_2\beta$ holds. In this case note that the only applicable rules in $C^*_{M'} \cup \overrightarrow{S_{Th}}$ are the rules in $C^*_{M'}$. By Lemma 5.4 $C^*_{M'} \cup \overrightarrow{S_{Th}}$ is confluent and then by assumption, Πq_1 and $q_2\beta$ will rewrite to the same term w.r.t. $S_{M'}$. Then the rules involving the non-state variables u_t do not affect the derivation and hence $\overrightarrow{S_{Th}}$ does not affect the rewrite steps. As a result Πq_1 and $q_2\beta$ will have the same rewrite proof w.r.t. S_{Th}. □

The next couple of results are easy consequences of the above lemmas:

Lemma 5.6. *Let $\Pi \in \{L, R\}^*$, $\beta \in \{0,1\}^*$, $q \in Q$, $u_t \in \mathcal{U}$. Then $\Pi q \leftrightarrow^*_{S_{Th}} u_t\beta$ if and only if there exist $b \in \{0,1\}$, $q' \in Q$, $\beta' \in \{0,1\}^*$ and a rule $u_t b \to q'$ in $\overrightarrow{S_{Th}}$ such that: (i) $\beta = b\beta'$, and (ii) $\Pi q \leftrightarrow^*_{S_{M'}} q'\beta'$*

Proof. The "if" part follows from Lemma 5.5, thanks to conditions (i) and (ii). For the "only if" part, the assumption implies the existence of $b \in \{0,1\}$, such that $\beta = b\beta'$. By the construction of $\overrightarrow{S_{Th}}$, there exists a state q' and a rule $u_t b \to q'$; it is not hard then to show that $\Pi q \leftrightarrow^*_{S_{M'}} q'\beta'$. □

Corollary 5.7. *Let $q \in Q$ be a state of M'. If $ext(q)$ is finite with respect to $S_{M'}$ then it is also finite with respect to S_{Th}.*

It follows then that the new variables u_t in $Var(\mathcal{S})$ have finite extent in S_{Th}, under the assumption that S'_M has finite Q-span:

Lemma 5.8. *Let $\Pi \in \{L, R\}^*$ and $\beta \in \{0,1\}^*$, and $u_t \in Var(\mathcal{S})$. If $ext(u_t)$ is infinite then there is a state $q \in Q$ such that $ext(q)$ is infinite.*

Proof. Note that $\overrightarrow{S_{Th}}$ has rules of the form $u_t b \to q'$ with $b \in \{0,1\}$. Thus the result follows from the definition of ext. □

Corollary 5.9. *If $S_{M'}$ has finite Q-span, then S_{Th} has finite $Var(\mathcal{S})$-span.*

Lemma 5.10. *$S_{M'}$ has finite Q-span if and only if S_{Th} has finite $Var(\mathcal{S})$-span.*

Proof. The "only if" assertion is the preceding Corollary; and the "if" assertion is Lemma 5.3. □

Lemma 5.11. *M' is bounded if and only if $\mathcal{S}$ is unifiable.*

Proof. In Sections 3.1 and 3.2 we showed that the unification problem $\mathcal{S}$ is solvable if and only if S_{Th} has finite $Var(\mathcal{S})$-span. By Lemma 5.10 S_{Th} has finite $Var(\mathcal{S})$-span if and only if $S_{M'}$ has finite Q-span. Finally the result follows since M' is bounded iff $S_{M'}$ has finite Q-span by Corollary 4.5. □

From the lemmas established in this section, we finally get our main result:

Theorem 5.12. *Unifiability modulo $\mathcal{E}$ is undecidable.*

6 Conclusion

The equational theory $\mathcal{E}$ studied in this paper is defined by a single equation, which is orientable either way to give a convergent term rewrite system, and for which every congruence class is finite. It is surprising that the unification problem could be undecidable for such a "weak" theory.

Since elementary unification modulo $\mathcal{E}$ is undecidable, so are unification with free constants and general unification. The semi-decision procedure that we have given in this paper for elementary unification can be easily extended to these cases. Finally, matching modulo $\mathcal{E}$ appears actually to be tractable (decidable in polynomial time); and this could be of some interest for the possible applications mentioned in the introduction.

References

[1] F. Baader, W. Snyder. "Unification Theory". In: *Handbook of Automated Reasoning*, pp. 440–526, Elsevier Sc. Publishers B.V., 2001.

[2] F. Baader, T. Nipkow. *Rewriting and all that.* Cambridge University Press, New York NY, USA, 1998.

[3] D. Chaum. "Security without Identification: Transaction System to Make Big Brother Obsolete". *Communications of the ACM* 28(2):1030–1044, 1985.

[4] Y. Gurevich, H. R. Lewis. "The Word Problem for Cancellation Semigroups with Zero". *J. Symbolic Logic* 49(1):184–191, 1984.

[5] C.A.R. Hoare, A. Hussain, B. Möller, P.W. O'Hearn, R.L. Petersen, G. Struth. "On Locality and the Exchange Law for Concurrent Processes". In *Proc. of CONCUR 2011*, LNCS 6901, Springer-Verlag, September 2011, pp. 250–264.

[6] P. K. Hooper. "The Undecidability of the Turing Machine Immortality Problem". *J. Symbolic Logic* 31(2):219-234, 1966.

[7] S. Jahama, A. J. Kfoury. "A General Theory of Semi-Unification". Technical Report 1993-018, Dept. of Computer Science, Boston University, December 1993.

[8] J.-P. Jouannaud, C. Kirchner. "Solving equations in abstract algebras: a rule-based survey of unification." In: *Computational Logic: Essays in Honor of Alan Robinson*, pp. 360–394, MIT Press, Boston (1991).

[9] D. Kapur, P. Narendran. "The Knuth-Bendix Completion Procedure and Thue Systems". *SIAM Journal on Computing* 14(4):1052–1072, 1985.

[10] A. J. Kfoury, J. Tiuryn, P. Urzyczyn. "The Undecidability of the Semi-Unification Problem". *Information and Computation* 102(1): 83-101, 1993.

[11] P. Narendran, F. Pfenning, R. Statman. "On the Unification Problem for Cartesian Closed Categories". *J. Symbolic Logic* 62(2): 636–647, 1997.

[12] M. Schmidt-Schauss. "A Decision Algorithm for Distributive Unification". *Theor. Comput. Science* 208 (1-2): 111–148, 1998.

[13] W. Snyder. *A Proof Theory for General Unification.* pp. 25–26, Birkhäuser, Boston (1991).

[14] E. Tiden, S. Arnborg. "Unification Problems with One-sided Distributivity". *Journal of Symbolic Computation* 3(1–2): 491–505, 1987.

[15] A. M. Turing. "The Word Problem in Semi-groups with Cancellation". *Annals of Mathematics* 52(2): 183–202, 1950.

Appendix: A sufficient condition for non-unifiability

One can define four different interpretations for $\mathcal{E}$, by interpreting B and $*$ as left or right projections on pairs. We denote these interpretations by function symbols $lp(*)$, $rp(*)$, $lp(B)$ and $rp(B)$. For instance, if we interpret $*$ as left projection via $lp(*)$, then the axiom $B(m,x)*B(n,r) = B(m*n, x*r)$ is trivially satisfied. The same holds if we take $*$ as right projection; and similarly for B. Now, if a problem is solvable modulo $\mathcal{E}$, it is also solvable on any of its models; we use this fact to prove the following lemma.

Lemma 6.1. *Let $\sim_{rp(*)}, \sim_{lp(*)}, \sim_{rp(B)}, \sim_{lp(B)}$ denote the reflexive, symmetric and transitive closures of $\succ_{r_*}, \succ_{l_*}, \succ_{r_B}, \succ_{l_B}$, respectively, on the set of variables of any $\mathcal{E}$-unification problem. and let β_1, β_2, β_3, β_4 be relations on these variables defined as follows:*

- $\beta_1 = \sim_{lp(*)} \circ \succ_B \circ \sim_{lp(*)}$
- $\beta_2 = \sim_{rp(*)} \circ \succ_B \circ \sim_{rp(*)}$
- $\beta_3 = \sim_{lp(B)} \circ \succ_* \circ \sim_{lp(B)}$
- $\beta_4 = \sim_{rp(B)} \circ \succ_* \circ \sim_{rp(B)}$

If any of these relations is not well-founded, then the problem is not solvable.

Proof. Suppose for instance that β_1 is not well-founded. If we interpret '$*$' as $lp(*)$, it is not hard to see that all variables in the same $\sim_{lp(*)}$-equivalence class become equal to each other. The relation $\succ_B$ then becomes not well-founded on the set of variables. This implies that there is a cycle w.r.t. $\succ_B$ in the interpreted problem – which is a standard unification problem –, hence there is no solution. So the result follows, since the interpreted problem is solvable if the original problem is solvable. Similar arguments hold for β_2, β_3 and β_4. □

We could therefore introduce the following additional failure rule:

(6) (Failure rule-2)

$$\frac{\mathcal{EQ}}{FAIL} \qquad \text{if any of the } \beta_i,\ i \in \{1,2,3,4\} \text{ is cyclic}$$