Constraint Solving for Term Orderings Compatible with Abelian Semigroups, Monoids and Groups *

Guillem Godoy (ggodoy@lsi.upc.es) and Robert Nieuwenhuis (roberto@lsi.upc.es)
Technical University of Catalonia, Dept. LSI,
Jordi Girona 1, 08034 Barcelona, Spain

Abstract. It is crucial for the performance of ordered resolution or paramodulation-based deduction systems that they incorporate specialized techniques to work efficiently with standard algebraic theories E. Essential ingredients for this purpose are term orderings that are *E-compatible*, for the given E, and algorithms deciding constraint satisfiability for such orderings.

Here we introduce a uniform technique providing the first such algorithms for some orderings for abelian semigroups, abelian monoids and abelian groups, which we believe will lead to reasonably efficient techniques for practice.

Our algorithms are in NP, and hence optimal, since in addition we show that, for *any* well-founded E-compatible ordering for these E, the constraint satisfiability problem is NP-hard even for conjunctions of inequations.

Keywords: symbolic constraints, term orderings, automated deduction, built-in theories.

1. Introduction

For the performance of ordered resolution or paramodulation-based deduction systems it is essential to use specialized techniques dealing efficiently with standard algebraic theories E, like abelian semigroups (AC, for associative and commutative) abelian monoids (AC0), or abelian groups (AG).

Essential ingredients for this purpose are *reduction* (i.e., well-founded and monotonic) orderings $\succ$ on ground terms that are *E-compatible* for the given E, i.e., $s =_E s' \succ t' =_E t$ implies $s \succ t$, and algorithms deciding the satisfiability of *ordering constraints* for such orderings. Such ordering constraints are used to express ordered strategies in automated deduction at the formula level (Kirchner et al., 1990). This allows one

* Submission to Constraints. Contact author: Robert Nieuwenhuis, www.lsi.upc.es/~roberto, phone/fax: +34-934017330/7014. Both authors are partially supported by the ESPRIT Basic Research Action CCL-II, ref. WG # 22457. and the Spanish CICYT project HEMOSS ref. TIC98-0949-C02-01. The first author is supported by Departament d'Universitats, Recerca i Societat de la Informació de la Generalitat de Catalunya. A preliminary version of this work was presented at the LICS 2001 conference.

© 2001 *Kluwer Academic Publishers. Printed in the Netherlands.*

to reduce the search space by inheriting the ordering restrictions while keeping completeness (Nieuwenhuis and Rubio, 1995; Nieuwenhuis and Rubio, 2000).

An ordering constraint is a quantifier-free first-order formula built over terms in $T(\mathcal{F},\mathcal{X})$ and over the binary predicate symbols '$=$' and '$>$'. These constraints are interpreted over the domain of ground terms, where $=$ and $>$ are interpreted, respectively, as a congruence $\approx$ and a reduction ordering $\succ$ such that $\succ$ is total up to $\approx$, i.e., for all ground terms s and t either $s \succ t$ or $t \succ s$ or $t \approx s$. Hence a *solution* of a constraint C is a substitution σ with range $T(\mathcal{F})$ and whose domain is the set of variables of C such that $C\sigma$ evaluates to true when interpreting $=$ as $\approx$ and $>$ as $\succ$. Then we say that σ *satisfies* C.

The first practical applications of ordering constraints gave rise to the distinction between *fixed signature* semantics (solutions are built over a given signature $\mathcal{F}$), and *extended signature* semantics (new symbols are allowed to appear in solutions). The latter semantics is in some cases easier to check, and is used in applications like the computation of saturated sets of ordering constrained clauses that can be used for deduction with other clauses containing arbitrary new (e.g., Skolem) symbols, but it is less restrictive and hence less powerful for refutational theorem proving. The satisfiability problem for ordering constraints was first shown decidable for the well-known *recursive path orderings* (RPO) introduced by N. Dershowitz (Dershowitz, 1982), for fixed signatures (Comon, 1990; Jouannaud and Okada, 1991) and extended ones (Nieuwenhuis and Rubio, 1995; Nieuwenhuis, 1993). NP algorithms (fixed and extended signatures) were given in (Nieuwenhuis, 1993; Narendran et al., 1999). For the Knuth-Bendix ordering (KBO) these results have only recently been obtained in (Korovin and Voronkov, 2000; Korovin and Voronkov, 2001).

Ordered strategies and ordering constraint inheritance can be used without loosing completeness with built-in algebraic theories E, like AC (Nieuwenhuis and Rubio, 1997; Vigneron, 1994) or AG (Godoy and Nieuwenhuis, 2000). An additional advantage of constraints in this context is that in each inference only one conclusion is generated, instead of one conclusion for each E-unifier. This can have dramatic consequences. For example, there are more than a million unifiers in $mgu_{AC}(f(x,x,x), f(y_1,y_2,y_3,y_4))$. But, probably due to the lack of adequate orderings and constraint solving algorithms, these ideas have not been put into practice yet. For example, McCune found his well-known AC-paramodulation proof of the Robbins conjecture (McCune, 1997) by still computing complete sets of AC-unifiers, and adding one new equation for each one of them (although heuristics were used to discard some of the unifiers).

Indeed, of the many, rather complex, AC-compatible reduction orderings that have been defined in the literature, only for the AC-RPO ordering of (Rubio and Nieuwenhuis, 1995) a constraint solving algorithm exists (Comon et al., 1995). But, unfortunately, this algorithm is far from practical due to its conceptual and computational complexity, and moreover, it only deals with extended signature semantics.

However, in many practical cases one has to deal with only one single associative and commutative symbol, and then a simple version of the RPO on *flattened* terms, which we will call FRPO, fulfills all requirements. The same FRPO can be used as an ingredient for an AG-compatible reduction ordering AG-RPO that satisfies all requirements of (Godoy and Nieuwenhuis, 2000), by using it to compare AG-normal forms of ground terms. Finally, it turns out that an AC0-compatible ordering AC0-RPO is obtained in a similar way by considering normal forms w.r.t. the rule $x + 0 \to x$.

Here we introduce a uniform technique providing the first constraint solving algorithms for fixed signature semantics for AC compatible orderings. More precisely, we give NP algorithms for FRPO-based orderings for abelian semigroups (FRPO itself), abelian monoids (AC0-RPO) and abelian groups (AG-RPO). We believe that the new techniques will lead to reasonably efficient practical algorithms for these orderings, and give new insights for the development of constraint solving methods over fixed signatures for other E-compatible orderings.

This paper is structured as follows. After the basic definitions of Section 2, in Section 3 we give some initial assumptions on the constraints. These assumptions simplify matters and are easy to enforce. Then we introduce the crucial notion of *segments*. Section 4 is on pure FRPO constraints. Then, after explaining the relatively simple extension to AC0-RPO in Section 5, in Section 6 we deal with the technically more complex part of the paper, namely the techniques for AG-RPO. With respect to the preliminary version of this work presented at the LICS 2001 conference (Godoy and Nieuwenhuis, 2001), here we have importantly restructured the article and all proofs in order to enhance legibility and more clearly separate the proofs from the algorithms.

It is obvious that the satisfiability problems we deal with are NP-hard, because as subcases they include the AC, AC0 and AG-unifiability problems which are all NP-hard. As a consequence, since our algorithms are in NP, they are optimal, and the problems are NP-complete. But one may wonder whether there exists any ordering at all for these E such that at least the satisfiability problem for positive conjunctions of inequations (by which one cannot always encode unification) is in P. In Section 7, we answer this question negatively: we show that for *any*

well-founded total E-compatible ordering for each one of these E, the problem is NP-hard even for conjunctions of positive inequations.

Finally, in Section 9 we give some conclusions and directions for further work.

2. Basic Definitions

We use the standard notation and terminology for terms and constraints of (Dershowitz and Jouannaud, 1990) and (Nieuwenhuis and Rubio, 2000).

In this paper we consider terms built over variables and the symbols of $\mathcal{F} \cup \{+, -, 0\}$, where $+$, $-$ and 0 are not in $\mathcal{F}$. The symbols of $\mathcal{F}$ will be called *free symbols*. In the following, (possibly sub- or super-indexed) symbols x, y, and z will always denote variables, symbols s, t and u will denote terms, and f and g will denote symbols of $\mathcal{F}$, i.e., free symbols. A term u is called a *summand* if it is headed with a free symbol. It is a *top-level summand* of all terms of the form u or $u + t'$ or $-u + t'$.

The rewrite system R_{AG} consists of the following five rules:

$$\begin{array}{rcl} x + 0 & \to & x \\ -x + x & \to & 0 \\ -(-x) & \to & x \\ -0 & \to & 0 \\ -(x + y) & \to & (-x) + (-y) \end{array}$$

By AG we denote the set of seven equations consisting of these five rules (seen as equations) plus AC, the associativity and commutativity axioms for $+$. By R_0 we mean the set $\{x + 0 \to x\}$ of only the first rule, and by AC0 we mean AC $\cup\, R_0$.

By $=_E$ we denote the congruence on terms generated by a set of equations E. In this paper, rewriting with a set of rules R is always considered *modulo* AC. For instance, when writing $\to_{R_{AG}}$, we mean the relation $=_{\mathrm{AC}} \to_{R_{AG}} =_{\mathrm{AC}}$. By $nf_R(s)$ we denote a normal form w.r.t. R of a term s. Each term s has a unique (up to $=_{\mathrm{AC}}$) normal form w.r.t. R_{AG}; see, e.g., (Marché, 1996).

Furthermore, as usual, terms will always be (eagerly) considered in *flattened form* w.r.t. AC. This flattening consists of removing all operators $+$ that are immediately below another $+$. For example, the term $+(\, a, +(\, f(\, +(a, +(b, c))\,), c)\,)$ becomes $+(a, f(+(a, b, c)), c)$. Note that in the flattened form of a term t, denoted by $\mathit{flat}(t)$, different occurrences of $+$ can have different arities (but all greater than 1). Usually, the symbol $+$ will be written in infix notation: $a + b + c$, and

terms like $(-a) + (-b)$ are written as $-a - b$. A term of the form $x_1 + \ldots + x_n - y_1 - \ldots - y_m$, with $n + m > 0$ will be called a *sum of variables.*

If $\succ$ and $=$ are an ordering and a congruence relation on terms, respectively, then the *lexicographic (left to right) extension of* $\succ$ *with respect to* $=$ is the relation $\succ^{lex}$ on n-tuples of terms defined by:

$$\langle s_1, \ldots, s_n \rangle \succ^{lex} \langle t_1, \ldots, t_n \rangle \quad \text{if} \quad s_1 = t_1, \ldots, s_{k-1} = t_{k-1} \text{ and } s_k \succ t_k$$

for some k in $1 \ldots n$. The *multiset extension of* $=$ is defined as the smallest relation $==$ on multisets of terms such that $\emptyset == \emptyset$ and

$$S \cup \{s\} == S' \cup \{t\} \text{ if } s = t \wedge S == S'$$

The *multiset extension of* $\succ$ *with respect to* $=$ is defined as the smallest ordering $\succcurlyeq$ on multisets of terms such that

$$M \cup \{s\} \succcurlyeq N \cup \{t_1, \ldots, t_n\} \text{ if } M == N \text{ and } s \succ t_i \text{ for all } i \in 1 \ldots n$$

Sometimes the notation $\succcurlyeq$ is used without explicitly indicating which is the congruence $=$. In these cases $=$ is assumed to be the syntactic equality relation on terms.

We now recall the definition of RPO, which allows for variadic symbols (hence we can cope with flattened terms). Let each function symbol be in one of the two disjoint sets lex or mul. If $f \in mul$ ($f \in lex$), it is said to have multiset (lexicographic) *status.* Let $>$ be a total *precedence* ordering $>$ on $lex \cup mul$. Let $=^{\text{mul}}$ be the congruence of equality modulo permutations of the direct subterms of symbols in mul; more precisely, it is the smallest congruence such that $f(t_1, \ldots, t_n) =^{\text{mul}} f(t_{\pi(1)}, \ldots t_{\pi(n)})$ whenever $f \in mul$ and π is a permutation of $(1, \ldots, n)$.

DEFINITION 1. *The* recursive path ordering *(RPO) (*Dershowitz, 1982*) is defined as follows:* $s = f(s_1, \ldots, s_n) \succ_{rpo} g(t_1, \ldots, t_m) = t$ *iff*

1. $\exists\ i \in \{1, \ldots, n\}\ s_i \succ_{rpo} t$ *or* $s_i =^{mul} t$, *or*
2. $f > g$ *and* $s \succ_{rpo} t_i$ *for all* $i = 1, \ldots, m$ *or*
3. $f = g$, $f \in mul$, *and* $\{s_1, \ldots, s_n\} \succcurlyeq_{rpo} \{t_1, \ldots, t_m\}$ *where* $\succcurlyeq_{rpo}$ *is the multiset extension of* $\succ_{rpo}$ *w.r.t.* $=^{mul}$, *or*
4. $f = g$, $f \in lex$, *and* $(s_1, \ldots, s_n) \succ^{lex}_{rpo} (t_1, \ldots, t_n)$ *where* $\succ^{lex}_{rpo}$ *is the lexicographic extension of* $\succ_{rpo}$ *w.r.t.* $=^{mul}$.

DEFINITION 2. *Let $>$ be a precedence and assume $mul = \{+\}$. The* RPO on flattened terms, *denoted by FRPO, is defined as follows:*
$s \succ_{frpo} t$ *if* $\mathrm{flat}(s) \succ_{rpo} \mathrm{flat}(t)$.

EXAMPLE 3. *FRPO is not monotonic in general. If $+ > a > b$ then $b + b \succ_{frpo} a$ but $a + a \succ_{frpo} b + b + a$. Also, if $a > + > f$ then $f(a) + f(a) \succ_{frpo} f(f(a))$ but $f(a) + f(f(a)) \succ_{frpo} f(a) + f(a) + f(a)$. Similar non-monotonicities occur in the presence of more than one AC symbol.* □

However, we have the following result. It is not used elsewhere in this article, but we give it here for showing the applicability of FRPO in practice:

PROPERTY 4. *(*Bachmair and Plaisted, 1985*) If $+$ is the only AC symbol and either $+$ is the smallest symbol in the precedence, or else only the smallest constant is smaller than $+$, then FRPO is an AC-compatible reduction (i.e., monotonic and well-founded) ordering on ground terms that is total up to $=_{\mathrm{AC}}$.*

DEFINITION 5. *The AC0-RPO and AG-RPO orderings are defined as follows. Let s and t be two ground terms. We define:*

$$s \succ_{ac0rpo} t \quad \text{if} \quad \mathrm{nf}_{R_0}(s) \succ_{frpo} \mathrm{nf}_{R_0}(t)$$
$$\text{and}$$
$$s \succ_{agrpo} t \quad \text{if} \quad \mathrm{nf}_{R_{AG}}(s) \succ_{frpo} \mathrm{nf}_{R_{AG}}(t)$$

The following results are stated here again only for showing that AC0-RPO and AG-RPO are also useful for practical applications like (Godoy and Nieuwenhuis, 2000). They are not difficult to prove (see also (Godoy and Nieuwenhuis, 2000)).

PROPERTY 6. *AC0-RPO is a total AC0-compatible reduction ordering on ground terms in normal form w.r.t. $\rightarrow_{R_0}$ if $+$ is the only AC symbol and the precedence is of the form $\ldots > + > 0$.*

PROPERTY 7. *AG-RPO is a total AG-compatible reduction ordering on ground terms in normal form w.r.t. $\rightarrow_{R_{AG}}$ if $+$ is the only AC symbol and the precedence is of the form $\ldots > - > + > 0$.*

In the following, we will consider these precedences, and also in the FRPO case 0 denotes the smallest constant symbol.

3. Constraint Solving

We now present a first transformation of our initial constraint C into a (a disjunction of) so-called *linear* systems S. After this, we will see that such S can be assumed to satisfy certain further useful assumptions. This transformation and the assumptions are independent of which one of the three orderings we deal with is considered. In later sections further ordering-specific assumptions will be introduced.

DEFINITION 8. *A* linear system S *is a constraint of the form*

$$x_1 = t_{1,1} = \ldots = t_{1,k_1} > \ldots > x_n = t_{n,1} = \ldots = t_{n,k_n}$$

where $\{x_1, \ldots, x_n\} = \mathrm{vars}(S)$, *and all* $t_{i,j}$ *are distinct non-variable terms.*

We denote by $=_S$ *the equivalence relation on terms with variables generated by the equalities in* S. *Each subconstraint of the form* $x_i = t_{i,1} = \ldots = t_{i,k_i}$ *is called an equivalence class. By* $>_S$ *we denote the smallest strict ordering relation on terms with variables that is compatible with* $=_S$ *and containing the inequalities of* S.

LEMMA 9. *(*Comon, 1990*;* Nieuwenhuis, 1993*)*

Independently of whether FRPO, AC0-RPO or AG-RPO is considered, each constraint C *can be transformed into a finite disjunction of linear systems such that* C *is satisfiable if and only if one of the linear systems is.*

The proof of the previous lemma is an easy consequence of the fact that one can consider all different linear orderings with $=$ and $>$ on all terms that are sides of relations of C. It is easy to see that for each such a linear ordering S, either C follows from the relations of S (i.e., all solutions of S are solutions of C), or else C is incompatible with it (i.e., no solution of S is a solution of C).

In order to obtain exactly one variable in each equivalence class, it suffices to insert a new (existentially quantified) variable in each equivalence class without any variables, or to merge two equal variables into one if necessary (merging of equal variables, which will be done more often in this paper, can be recorded separately if one wants to reconstruct a solution for the original constraint rather than to decide its satisfiability).

Similarly, in what follows we will make some more assumptions.

DEFINITION 10. *The* global *assumptions about a linear system S*

$$x_1 = t_{1,1} = \ldots = t_{1,k_1} > \ldots > x_n = t_{n,1} = \ldots = t_{n,k_n}$$

are as follows:

A1. 0 *is in the rightmost class, i.e.,* 0 *is* $t_{n,i}$ *for some* i.

A2. Each $t_{i,j}$ *is either a sum of variables, or* 0*, or of the form* $f(y_1, \ldots, y_m)$.

A3. S *is of the form* $x = t > S'$*, where* t *is headed with a free symbol or* 0.

A4. Each equivalence class contains at most one term headed with a free symbol.

A5. If some $t_{i,j}$ *is headed with a free symbol, then* $t_{i,j} >_S x$ *for all* x *in* $t_{i,j}$.

A6. If $f(y_1, \ldots, y_m) >_S g(z_1, \ldots, z_k)$ *and* $g > f$*, then* $y_i \geq_S g(z_1, \ldots, z_k)$ *for some* i *in* $1 \ldots m$.

A7. If $f(y_1, \ldots, y_m) >_S f(z_1, \ldots, z_m)$ *for then either* $y_i \geq_S f(z_1, \ldots, z_m)$ *for some* i *in* $1 \ldots m$ *or else* $(y_1, \ldots, y_m) >_S^{lex} (z_1, \ldots, z_m)$.

A8. If $s \geq_S t$ *where* s *is a sum of variables and* t *is headed with some free symbol, then* $x \geq_S t$ *for some variable* x *in* s.

LEMMA 11. *Independently of whether FRPO, AC0-RPO or AG-RPO is considered, each linear system* S *can be transformed into a finite disjunction of linear systems* $S_1, \ldots, S_n$ *satisfying the global assumptions and such that* S *is satisfiable if and only if one of the* S_i *is.*

Similarly to Lemma 9, the previous lemma is based on the fact that some additional terms can be inserted in S as well (while keeping a linear system).

For $A1$, one can guess that the smallest variable is 0 or that 0 is below the smallest variable. For $A2$, one can insert as well all subterms headed with free symbols and the direct subterms of such terms. Then one can replace the non-variable arguments t of all terms by the variable x with $x =_S t$. For $A3$, any constraint $x_1 = t_{1,1} = \ldots = t_{1,k_1} > S$ can be transformed, by adding an additional leftmost equivalence class, into $x_0 = f(0, \ldots, 0, x_1) > x_1 = t_{1,1} = \ldots = t_{1,k_1} > S$, by taking f as the minimal non-constant function symbol in $\mathcal{F}$. Once assumptions $A1$-$A3$

have been imposed, by the definitions of the orderings, assumptions *A*4-*A*8 either hold or else the constraint is necessarily unsatisfiable.

We now introduce the notion of *segment*, which is again common to the algorithms for FRPO, AC0-RPO, and AG-RPO constraints.

DEFINITION 12. *A* segment *T of a linear system S is a subsequence of S of the form*

$$x_0 = t_{0,1} = \ldots = t_{0,k_0} > x_1 = t_{1,1} = \ldots = t_{1,k_1} > \ldots > x_n = t_{n,1} = \ldots = t_{n,k_n}$$

where t_{0,k_0} is headed by a free symbol, and t_{n,k_n} is 0 or headed by a free symbol, and all other $t_{i,j}$ are sums of variables.

The variables $x_1, \ldots, x_n$ (note: not *x_0) are said to be the variables* defined *in T, and their occurrences as single variables in their equivalence classes are called their* definitions.

4. FRPO constraints

We now treat the case of the AC-compatible FRPO-ordering. Some additional specific assumptions are needed that do hold for FRPO, but not for all three orderings dealt with in this article.

DEFINITION 13. *The* FRPO assumptions *about a linear system S*

$$x_1 = t_{1,1} = \ldots = t_{1,k_1} > \ldots > x_n = t_{n,1} = \ldots = t_{n,k_n}$$

are the global assumptions plus the following two additional ones:

*AC*1. *If some $t_{i,j}$ is $y_1 + \ldots + y_m$ then $t_{i,j} >_S y_i$ for all i in $1 \ldots m$.*

*AC*2. *In no equivalence class there is a term headed with $+$ and another one headed with a free symbol.*

As before, for all linear systems satisfying the global assumptions, the FRPO assumptions either hold or else the constraint is necessarily unsatisfiable w.r.t. FRPO:

LEMMA 14. *Any linear system S that fulfills the global assumptions and that is satisfiable w.r.t. FRPO also fulfills the FRPO assumptions.*

EXAMPLE 15. *Let the constraint C be $f(x+z) > y \ \wedge \ z > f(x)$. One way of linearly ordering its terms with $>$ and $=$ is $y = f(x+z) > f(x) > x+z > x = z = 0$. Enforcing the FRPO assumptions, by adding new variables w_1 and w_2 for the classes of $f(x)$ and $x+z$ respectively,*

and merging x and z, it becomes $y = f(w_2) > w_1 = f(x) > w_2 = x+x > x = 0$. However, it is in contradiction with our initial constraint C. Another linear ordering is $f(x+z) > x+z > z = y > f(x) > x = 0$, which becomes $w_1 = f(w_2) > w_2 = x + y > y > w_3 = f(x) > x = 0$. This linear system satisfies all FRPO assumptions and it is not in contradiction with C. □

4.1. The splitting transformation

Due to the FRPO assumptions, each segment is of the form

$$x_0{=}s \;>\; x_1{=}t_{1,1}{=}\ldots{=}t_{1,k_1} \;>\ldots>\; x_i{=}t_{i,1}{=}\ldots{=}t_{i,k_i} \;>\; x_{i+1}{=}t$$

In such a segment T, every variable occurring in some $t_{i,j}$ is defined either in T itself or in some other segment to the right of T. Now our aim is to transform S in such a way that the latter kind of variables are removed from T, i.e., such that all variables occurring in some $t_{i,j}$ of such a segment are defined in the segment itself. This transformation preserves satisfiability, but we will not give the proof here because Lemma 24 below gives a simpler and more direct proof of the main result needed, namely that the original system is satisfiable if, and only if, the *diophantine system* for some of its split systems is satisfiable.

As a result of this splitting transformation, terms $f(v_1, \ldots, v_n)$ where the v_i are sums of variables may appear in S, and hence assumption $A2$ will not hold anymore after the transformation.

The idea of the transformation is as follows. Let σ be some arbitrary solution of S, let x and y be variables defined in T, such that y is the variable defined in the equivalence class immediately below the one where x is defined. Then $x\sigma \succ_{frpo} y\sigma \succeq_{frpo} t\sigma$. Therefore, for at least one of the top-level summands u of $x\sigma$ we have $u \succeq t\sigma$. Hence, if U_x is the sum of all top-level summands u of $x\sigma$ with $u \succeq t\sigma$, and u_x is the (possibly empty) sum of the smaller ones, then $x\sigma$ is of the form $U_x + u_x$ or of the form U_x. Similarly, $y\sigma$ can be of the form $U_y + u_y$ or U_y. Furthermore, either (i) $U_x \succ U_y$, or else (ii) $U_x = U_y$, u_x is non-empty, and either u_y is empty or $u_x \succ u_y$. In case (i), we say that $x\sigma \succ_{frpo} y\sigma$ due to the "large" summands, and in the case (ii) due to the "small" summands. According to these ideas, S will be transformed by the following transformation.

DEFINITION 16. *Let S be a linear system satisfying the FRPO assumptions. The following* splitting transformation *for S treats one segment T at the time, segment by segment from left to right. The last segment (i.e., the rightmost one, which is of the form $x_n = 0$) needs*

no treatment. One can assume, inductively, that in every segment T' to the left of T, all variables that appear not below a free symbol in T' are defined in T' itself, and moreover, due to assumptions $A5$ and $AC1$, no variable defined in T' appears in a segment to the right of T'. After treating each segment T, the FRPO assumptions (as well as the definition of linear constraints) are assumed to be imposed eagerly in the segments to the right of T (otherwise, some of the steps of the transformation may not make sense).

Let T be a segment in S of the form:

$$x_0{=}s \ > \ x_1{=}t_{1,1}{=}\ldots{=}t_{1,k_1} \ > \ldots > \ x_i{=}t_{i,1}{=}\ldots{=}t_{i,k_i} \ > \ x_{i+1}{=}t$$

and assume the transformation has already been applied to all segments left of T in S. Then the splitting transformation *for T consists of the following steps:*

1. *Guess a subset of* split *variables of $\{x_1 \ldots x_i\}$ such that whenever $x =_S y_1 + \ldots + y_k$, then x is split if, and only if, at least one of the y_i is split or defined in a segment to the right of T (intuitively, x is split if it is guessed to have at least one "small" summand).*

2. *If x is a split variable, then introduce two new variables X and x', and everywhere in S replace x by $X + x'$. In this case we say that x is* split *into $X + x'$ (intuitively, the X is for the large summands and the x' for the small ones). If x is a non-split variable of $\{x_0 \ldots x_{i+1}\}$, replace x everywhere in S by a new variable X. These replacements are needed not only in this segment, but also to the left of it, since these variables may appear in some term of the form $f(v_1, \ldots, v_n)$ where the v_i are sums of variables.*

3. *After this, the equivalence classes e in T are either of the form $V_1{+}v_1 = \ldots = V_k{+}v_k$ or of the form $V_1{=}\ldots{=}V_k$, where the V_i are either sums of upper case variables or terms headed with free symbols, and the v_i are sums of lower case variables and variables defined in segments to the right of T (all summands contain at least one upper case variable thanks to assumptions $A8$ and $AC1$). If e is such an equivalence class, we denote by E the equivalence class $V_1 = \ldots = V_k$ and by e' the class $v_1 = \ldots = v_k$ (if it exists for e). Then we can write T as $e_0 > e_1 > \ldots > e_{i+1}$ and we can guess, for each relation $e_j > e_{j+1}$ whether (i) it is due to the large summands or (ii) to the small ones (note that case (ii) applies only if e'_j is non-empty).*

   According to these guesses, replace T by the new segment T':

$$E_0 \ > \ E_1 \ \# \ \ldots \ \# \ E_{i+1}$$

where the relations # stand for $>$ or $=$ depending on the guesses made, and insert each e'_j in some segment to the right of T, adding it to an existing equivalence class or creating a new one, in such a way that, whenever $E_j =_{T'} E_{j+1}$, either $e'_j > e'_{j+1}$ or e'_{j+1} does not exist.

Note that the previous splitting transformation does not increase the number of segments of S and only a polynomial number of variables are split: each variable can only lead to k splittings, where k is the number of segments.

EXAMPLE 17. *(Example 15 continued) Let us apply the splitting transformation to the result $w_1 = f(w_2) > w_2 = x + y > y > w_3 = f(x) > x = 0$ of Example 15. First we treat the leftmost segment $w_1 = f(w_2) > w_2 = x + y > y > w_3 = f(x)$. The possible variables to be split are w_2 and y. We guess to split only w_2 into $W_2 + w'_2$, obtaining $w_1 = f(W_2 + w'_2) > W_2 + w'_2 = x + y > y > w_3 = f(x)$. Now, for the relation $W_2 + w'_2 > y$ we guess $W_2 = y$. After removing w'_2 from this segment and inserting it, for example, in the equivalence class of 0, we obtain $w_1 = f(y + x) > y > w_3 = f(x) > x = 0$. For the segment $w_3 = f(x) > x = 0$ no splitting is needed.* □

Let us now analyze the assumptions that can be made about the transformed systems.

DEFINITION 18. *Let S be a linear system. Two sums of variables $X_1 + \ldots + X_n$ and $Y_1 + \ldots + Y_m$ are* compared by segments *in S, denoted $X_1 + \ldots + X_n >_{segs(S)} Y_1 + \ldots + Y_m$, if:*

1. *All X_i are defined in different segments of S*
2. *All Y_i are defined in different segments of S*
3. *$\{X_1, \ldots, X_n\} >_S^{mul} \{Y_1, \ldots, Y_m\}$.*

Note that, if the elements of each set are written in decreasing order w.r.t. $>_S$, then point 3. of the previous definition is equivalent to $\{X_1, \ldots, X_n\} \supset \{Y_1, \ldots, Y_m\}$ or $(X_1, \ldots, X_n) >_S^{lex} (Y_1, \ldots, Y_m)$. In fact, this latter equivalent notion is the one that will be dealt with in the proofs below.

Now consider a relation $x >_S y$ in S, and assume that S' is obtained from S by the splitting transformation. Then, each occurrence of a term $f(x)$ in S becomes in S' a term of the form $f(X + X' + X'' + \ldots)$, and similarly, $f(y)$ becomes $f(Y + Y' + Y'' + \ldots)$; furthermore, the sums

$X + X' + X'' + \ldots$ and $Y + Y' + Y'' + \ldots$ are compared by segments in S'.

According to this, consider the following new assumptions. They are basically needed because as a result of the splitting transformation, terms $f(v_1, \ldots, v_n)$ where the v_i are sums of variables may appear in S, and also because some stronger statements about the ordering can be made after the splitting transformation:

DEFINITION 19. *The* Split FRPO assumptions *on a linear system S*

$$x_1 = t_{1,1} = \ldots = t_{1,k_1} > \ldots > x_n = t_{n,1} = \ldots = t_{n,k_n}$$

are the FRPO assumptions where A2, A6 and A7 are reformulated as follows:

A2′. Each $t_{i,j}$ is either a sum of variables, or 0, or of the form $f(v_1, \ldots, v_n)$ where the v_i are sums of variables.

A6′. If $f(v_1, \ldots, v_m) >_S g(w_1, \ldots, w_k)$ and $g > f$, then $x \geq_S g(w_1, \ldots, w_k)$ for some x in $f(v_1, \ldots, v_m)$.

A7′. If $f(v_1, \ldots, v_m) >_S f(w_1, \ldots, w_m)$ then either $x \geq_S f(w_1, \ldots, w_m)$ for some variable x in $f(v_1, \ldots, v_m)$, or else $(v_1, \ldots, v_m) >^{lex}_{segs(S)} (w_1, \ldots, w_m)$.

LEMMA 20. *Let S be a linear system satisfying the FRPO assumptions, and let $S_1, \ldots, S_n$ be the (disjunction of) linear systems resulting from the possible splitting transformations on S. Then each S_i satisfies the split FRPO assumptions.*

Hence in what follows we can assume that, after the splitting transformation, we deal with linear systems satisfying the split FRPO assumptions.

4.2. Diophantine systems

We are now ready for defining the system of diophantine equations and inequations D_S for a linear system S. Later on we will see that the variables of D_S are interpreted over the positive natural numbers, and that the symbol $+$ in D_S is of course interpreted correspondingly.

DEFINITION 21. *Let S be a linear system satisfying the split FRPO assumptions. The* system of diophantine equations and inequations D_S for S *is the set of all equations and inequations such that for every segment in S of the form*

$$x_0 = s > x_1 = t_{1,1} = \ldots = t_{1,k_1} > \ldots > x_i = t_{i,1} = \ldots = t_{i,k_i} > x_{i+1} = t$$

the following equations and inequations are in D_S:

1. $x_1 > x_2,\ \ x_2 > x_3,\ \ \ldots,\ \ x_i > x_{i+1}$

2. $x_j = t_{j,k},\quad$ *for all* j *in* $1 \ldots i$ *and all* k *in* $1 \ldots k_j$

3. *the equation* $x_{i+1} = 1$.

EXAMPLE 22. *(Example 17 continued) The system of diophantine equations for* $w_1 = f(y + x) > y > w_3 = f(x) > x = 0$ *is*

$$w_1 = 1 \quad y > w_3 \quad w_3 = 1 \quad x = 1$$

We obtain a solution θ *for it by defining* $y\theta = 2$. *Below we will see that from each such a* θ *one can build a solution* σ *for the linear system from right to left. We have* $x\sigma = 0$ *and hence* $w_3\sigma = f(0)$. *Now for each variable* v *with* $v\theta = n$, *we define* $v\sigma = t + \ldots^{n)} + t$, *where* t *is the summand at the lower end of its segment; e.g., we define* $y\sigma$ *to be* $f(0) + f(0)$. *Finally, we have* $w_1\sigma = f(f(0) + f(0) + 0)$. *If one desires to reconstruct the solution for the original constraint of Example 15:* $w_2'\sigma$ *is* 0, *and* $z\sigma$ *is* $f(0) + f(0)$. □

4.3. Deciding the satisfiability of FRPO constraints

The following simple result will be used below when solving ordering constraints on multisets of several elements as multisets over a single element:

LEMMA 23. *Let* C *be a set* $\{e_n, \ldots, e_0\}$ *with an ordering* $\succ$ *where* $e_n \succ \ldots \succ e_0$. *Then for any decreasing sequence of finite multisets over* C

$$M_0 \gg \ldots \gg M_m$$

there exists a weighting function $f : C \to \mathcal{N}$ *from* C *into the natural numbers, with* $f(e_0) = 1$ *such that*

$$F(M_0) > \ldots > F(M_m)$$

where the extension to multisets F *of* f *is defined* $F(\{a_1 \ldots, a_k\}) = f(a_1) + \ldots + f(a_k)$.

Proof: Let k be $|M_0| + \ldots + |M_m|$. Then, for instance, the function $f(e_i) = k^i$ fulfills the requirements. □

LEMMA 24. *Let* $S_1 \ldots S_m$ *be the resulting systems of applying the splitting transformation to a linear system* S. *Then* S *is satisfiable for FRPO if, and only if, some* D_{S_i} *is satisfiable in the positive natural numbers.*

Proof: $\Longleftarrow$: Assume $D_{S'}$ is satisfiable for some S' in $\{S_1 \ldots S_m\}$. Let θ be a solution for $D_{S'}$. We can inductively build a solution σ for S' as follows. For each segment T in S' of the form

$$x_0{=}s \;>\; x_1{=}t_{1,1}{=}\ldots{=}t_{1,k_1} \;>\ldots>\; x_i{=}t_{i,1}{=}\ldots{=}t_{i,k_i} \;>\; x_{i+1}{=}t$$

assume a (partial) solution σ has already been defined for the linear system consisting of all segments to the right of T. Then, for the variables x_j defined in T, we define $x_j\sigma$ to be $t\sigma + \ldots^{n)} + t\sigma$ where $n = x_j\theta$ (note that if T is the rightmost segment, then t is 0). By construction of $D_{S'}$, σ satisfies all equality relations in S', that is, $u\sigma =_{\mathrm{AC}} v\sigma$ for all u and v with $u =_{S'} v$. Furthermore, it also satisfies the relations $x_j\sigma \succ_{frpo} x_{j+1}\sigma$ with j in $\{1 \ldots i\}$ for such segments T.

Hence it only remains to be checked that σ satisfies $s\sigma \succ_{frpo} x_1\sigma$. We know that σ is a solution for all relations between terms strictly to the right of s. One shows that $s\sigma \succ_{frpo} t'\sigma$, where t' is any term to the right of s. This is done inductively from right to left.

The result is trivial if t' is 0, since s is of the form $f(v_1, \ldots, v_n)$ and $f > 0$. If t' is a sum of variables, then, by construction of σ, $t'\sigma$ is of the form $nt''\sigma$, where t'' is the rightmost term in the segment of t'. By the induction hypothesis $s\sigma \succ_{frpo} t''\sigma$, and hence $s\sigma \succ_{frpo} t'\sigma$, since $f > +$. Otherwise, let t' be of the form $g(w_1, \ldots, w_m)$ where the w_i are sums of variables. We distinguish three cases:

1. $f > g$. By assumption $A5$, every variable y occurring in the term $g(w_1, \ldots, w_m)$ appears to the right of it, and by the induction hypothesis, $s\sigma \succ_{frpo} y\sigma$ for such y. Therefore, $s\sigma \succ_{frpo} g(w_1, \ldots, w_m)\sigma$ follows by definition of RPO since $f > g > +$.

2. $g > f$. By assumption $A6'$, $x \geq_S g(w_1, \ldots, w_m)$ for some variable x in s, and hence $x\sigma \succ_{frpo} g(w_1, \ldots, w_m)\sigma$. Since x is a proper subterm of s, we have that $s\sigma \succ_{frpo} x\sigma$, and $s\sigma \succ_{frpo} g(w_1, \ldots, w_m)\sigma$ follows by transitivity.

3. $f = g$. By assumption $A7'$ either $x \geq_S f(w_1, \ldots, w_n)$ for some variable x in $f(v_1, \ldots, v_n)$, or else $(v_1, \ldots, v_n) >^{lex}_{segs(S)} (w_1, \ldots, w_n)$. In first case we conclude in a similar way as for the case $g > f$. For the second case, note that $v_i >_{segs(S)} w_i$ implies $v_i\sigma \succ_{frpo} w_i\sigma$ because v_i and w_i are of the form $y_1 + \ldots$ and $z_1 + \ldots$ respectively, and there exists some k such that $y_j = z_j$ for all j in $1 \ldots k-1$, and $y_k >_S z_k$ if z_k exists, and for all $l > k$ we have that z_l appears in segments to the right of the one where y_k is defined, and hence, $y_k\sigma \succ_{frpo} z_k\sigma + z_{k+1}\sigma + \ldots$ (remember that $+$ has multiset status, and that, by the induction hypothesis, σ satisfies all the relations to the right of s).

Furthermore, by assumption $A5$ and the induction hypothesis, we have $s\sigma \succ_{frpo} y\sigma$ for every variable y in any of the w_i, and since $f > +$, we have that $s\sigma \succ_{frpo} w_i\sigma$. Altogether this implies $s\sigma \succ_{frpo} f(w_1, \ldots, w_n)\sigma$.

Once we have a solution σ for S' of this kind (i.e., where each $x\sigma$ is a sum of $t\sigma$'s, where t is the lower extreme of the segment where x is defined), it can be extended to a solution for S by recursively defining $x\sigma$ to be $X\sigma + x'\sigma$, for each splitting of a variable x into $X + x'$.

$\Longrightarrow$: Assume S is satisfiable. Now we prove that $D_{S'}$ is satisfiable as well for some S' in $\{S_1 \ldots S_m\}$. Let σ be a solution of S. Let S' be the system obtained by applying the splitting transformation according to σ, that is, if x is defined in a segment T of S of the form

$$x_0 = s \ > \ x_1 = t_{1,1} = \ldots = t_{1,k_1} \ > \ldots > \ x_i = t_{i,1} = \ldots = t_{i,k_i} \ > \ x_{i+1} = t$$

then x is split into $X + x'$ if $x\sigma$ contains some summand smaller than $t\sigma$; we proceed similarly for the other guessings, and σ is extended conveniently for the new variables. The extended substitution σ is a solution for S'. Moreover, in a segment of S' like the previous one, for all j in $\{1 \ldots i+1\}$ we have that $x_j\sigma$ contains only top-level summands greater than or equal to $t\sigma$.

Now let $C = \{u_0, \ldots, u_n\}$ be all the different top-level summands of these variables, where $u_n \succ_{frpo} u_{n-1} \succ_{frpo} \ldots \succ_{frpo} u_0$ and u_0 is $t\sigma$. Every $x_j\sigma$ and $t_{j,l}\sigma$ can be seen as a multiset of these summands (the multiset of its top-level summands). By Lemma 23 there exists a function $f : C \to \mathcal{N}$ such that its extension F to multisets satisfies $F(x_1\sigma) \gg \ldots \gg F(x_{i+1}\sigma)$, and $F(x_{i+1}\sigma) = f(u_0) = 1$. Moreover, since $x_j\sigma$ and $t_{j,l}\sigma$ are the same multiset, if $t_{j,l}$ is of the form $x_{j_1} + \ldots + x_{j_l}$, then $F(x_j\sigma) = F(t_{j,l}\sigma) = F(x_{j_1}\sigma) + \ldots + F(x_{j_l}\sigma)$. Therefore, the assignment $x_j\theta = F(x_j\sigma)$ satisfies the equations of $D_{S'}$ corresponding to T. □

THEOREM 25. *The satisfiability problem for FRPO constraints is in NP.*

Proof: Generating one of the linear systems S of the disjunction equivalent to C consists of a polynomial number of guessings of the relations between all the subterms in C, and the size of S is polynomial w.r.t. the size of C. The splitting transformation consists of a polynomial number of guessings. By Lemma 24, S is satisfiable if and only if there exists a sequence of guessings, in the splitting transformation, giving a linear system S', such that $D_{S'}$ is satisfiable. Checking whether $D_{S'}$ is satisfiable is again in NP (Schrijver, 1987). □

5. AC0-RPO Constraints

In this section we consider AC0-RPO constraints over arbitrary signatures of the form $\ldots > f > + > 0$. Observe that all terms of the form $0+\ldots+0$ are equivalent to 0 in this setting and that hence the second smallest term w.r.t. the ordering AC0-RPO is $f(0,\ldots,0)$. Therefore we can add, w.l.o.g., an additional assumption to our linear systems, in particular because otherwise assumption $AC1$ does not hold.

$AC0$ All linear systems S are of the form $S' > x = f(y,\ldots,y) > y = 0$ and no term of the form $t + y$ occurs in S. This is obtained by removing y in such terms.

With this additional assumption, one can check that the whole rest of the steps described in the previous section directly suffice for AC0-RPO constraints. Minor details are that, during the splitting process, the new assumption $AC0$ has to be preserved, and hence no small variables resulting from a splitting can be inserted in the rightmost segment. Moreover, in the diophantine system it is not necessary to create the equations corresponding to the rightmost segment.

Observe that the basic idea of the splitting process is that solutions for the linear system are transformed into new solutions where, at every segment, the variables that appear in it contain only top-level summands of this segment. Therefore, 0 does not appear in segments that are not the rightmost one, and hence everything behaves like in the FRPO case, again solving the diophantine equations over the positive natural numbers. This gives us the following result.

THEOREM 26. *The satisfiability problem for AC0-RPO constraints is in NP.*

6. AG-RPO Constraints

In this section we consider AG-RPO constraints over arbitrary signatures of the form $\ldots > - > + > 0$.

Let us first consider some examples over the signature $f > a > - > + > 0$ where f is unary and a is a constant.

EXAMPLE 27. *The smallest terms over this signature in increasing order w.r.t. $\succ$ are:*

$0,\ a,\ a+a,\ a+a+a,\ \ldots,\ -a,\ -a-a,\ -a-a-a,\ \ldots,\ f(0),\ f(0)+a$

$f(0)+a+a,\ \ldots,\ f(0)-a,\ f(0)-a-a,\ \ldots,\ f(0)+f(0),\ f(0)+f(0)+a,\ \ldots,\ -f(0)$
where $-a$ *is the smallest limit ordinal* ω, $f(0)$ *is* 2ω, $f(0)-a$ *is* 3ω, $f(0)+f(0)$ *is* 4ω, $-f(0)$ *is* ω^2, *and* $f(a)$ *is* $2\omega^2$. □

EXAMPLE 28. *We have* $f(f(a)) \succ f(a-f(0)+f(a-a))$ *since*
$\mathrm{nf}_{R_{AG}}(f(f(a))) = f(f(a)) \succ_{FRPO} f(a) = \mathrm{nf}_{R_{AG}}(f(a-f(0)+f(a-a)))$. □

EXAMPLE 29. *Terms can be smaller than their subterms:* $\sigma \models x > f(x-f(a))$ *if* $x\sigma = f(a)$, *since* $\mathrm{nf}_{R_{AG}}(f(a)) = f(a) \succ_{FRPO} f(0) = \mathrm{nf}_{R_{AG}}(f(f(a)-f(a)))$. □

As we have seen in the previous example, a linear constraint such that x appears to the right of the segment where it is defined may be satisfiable (hence assumption $AC1$ will not be made in this section). Similarly, the following example shows us that terms headed with f may become equal to terms headed with $+$ or $-$ (and hence assumption $AC2$ is also not considered in this section).

EXAMPLE 30. $\sigma \models x-y=f(z)$ *if we have* $x\sigma = f(a)+f(a)$, $y\sigma = f(a)$, $z\sigma = a$. □

Another difficulty to be taken into account is that, after the splitting transformation, contrarily to what happened in the previous sections, a solution for a linear constraint may need more than one different top-level summand for some segments:

EXAMPLE 31. *Suppose that we have a signature of the form* $f > - > + > 0$ *where* f *is unary. Then the smallest terms are ordered like:*
$0,\ \ f(0),\ \ f(0)+f(0),\ \ f(0)+f(0)+f(0),\ \ \ldots,$
$\quad -f(0),\ \ -f(0)-f(0),\ \ -f(0)-f(0)-f(0),\ \ \ldots,\ \ f(f(0)).$
The linear constraint $f(f(0)) > -z > z > y > -y > f(0)$ *is unsatisfiable: since we need to satisfy* $y > -y$, *necessarily* $y\sigma$ *is a sum of negative* $f(0)$*'s. Therefore* $z\sigma$ *is of the form* $-f(0) - \ldots - f(0)$, *with some more negative* $f(0)$*'s. But then* $-z > z$ *is not satisfied by* σ.

However, the linear constraint $f(f(f(0))) > -z > z > y > -y > f(0)$ *has the solution* σ *where* $y\sigma = -f(0) - f(0)$ *and* $z\sigma = f(f(0)) + f(f(0))$. *It has no solution where* $y\sigma$ *and* $z\sigma$ *are built from one single summand.* □

6.1. Only unary symbols

For explanation purposes, in this subsection we first solve the problem under the restriction that all the non-constant function symbols have arity one. Our signature is of the form $\ldots > h > c_1 > \ldots > c_l > - > + > 0$, where h is the smallest non-constant function symbol, i.e., all the c_i are constants.

EXAMPLE 32. *We have the following ordering on summands (from which the ordering on ground terms is easily derived). If $l = 0$ then the smallest summands are, in increasing order:*
$h(0),\ h(h(0)),\ h(h(0)+h(0)), \ldots$ *If $l \neq 0$ then the smallest summands are, in increasing order: $c_l,\ \ldots,\ c_1,\ h(0),\ h(c_l),\ h(c_l + c_l),\ h(c_l + c_l + c_l), \ldots$. These summands will be denoted by $sum_1, sum_2, sum_3, \ldots$*

The successor summand of a summand of the form $h(s)$ is $h(s + sum_1)$ if s is not of the form $s' - sum_1$, and $h(s - sum_1)$ otherwise. The successor summand of a summand $f(s)$ with $f > h$ is always $h(f(s))$. We write $succsum_k(u)$ to denote the k-th successor summand of u.

6.1.1. *Assumptions for the linear systems.*

As before, we generate a disjunction of linear systems, and apart from the global assumptions, we need the following:

DEFINITION 33. *The AG-RPO assumptions for a linear system S consist of the global assumptions plus the following two additional ones:*

AG1. All the constants c_i and the terms sum_1, sum_2 and $h(0)$ appear in S, and in the correct order.
The segment between sum_2 and sum_1 will be called the base segment.

AG2. S is of the form $S' > x = \ldots = sum_1 > y = \ldots = 0$ and no term of the form $t + y$ occurs in S.

Assumption $AG2$ is a modification of assumption $AC0$: in the class of 0, sums of variables defined to the left of it may appear; in a solution for the system, these variables will contain summands that cancel each other out.

In this setting, a sum of variables is a sum of positive and negative variables, and all assumptions have to be interpreted accordingly. For example, assumption $A8$ implies that no term of the form $-x$ is in a segment to the left of the segment where x is defined. If $-x$ appears in a segment to the right of the one where x is defined, we have a case

of unsatisfiability not detected by our assumptions. But the splitting transformation and the check of solutions of the diophantine system detect this case and other additional ones.

6.1.2. *The splitting transformation.*

The splitting transformation is essentially as for FRPO but it is technically more complex. Hence the reader should understand first the FRPO case. Before giving the formal definitions, let us first provide some intuition behind the differences with the FRPO case.

Firstly, note that when it is guessed that some relation is due to the small summands, the small terms cannot be inserted in the class of 0, because adding 0 to a term does not make it larger w.r.t. AG-RPO. Therefore also no splitting of variables in the base segment is done.

Another difference with the FRPO case is that after splitting and removing small variables from a segment T, some variables defined in T could appear to the right of T. To avoid this, we need to *associate* some equations to a segment T during the splitting transformation of T. These equations are of the form $M = 0$, where M is a sum of positive and negative variables defined in T. These equations are not inserted in the linear system, but they are kept because they will produce part of the diophantine system.

Let M denote such a sum of (upper case) variables defined in T, and let m denote a sum not containing any of these variables. Let s be a term in some equivalence class to the right of T, and suppose that s is of the form $M + m$ (no term of the form $f(M + m)$ of $f(M)$ can appear due to assumption $A5$). Then, in any solution σ, the term $M\sigma$ must be equivalent to 0. Therefore, for each such s, the part M is removed from s, and $M = 0$ becomes an associated equation of T. If the part m of s is empty, then s is replaced by the variable of the class of 0.

Finally, for simplicity reasons, we want the rightmost class of each T to be of the form $x = t$. This can be accomplished as follows. Assume that after splitting this class is $x = T_1 + t'_1 = \ldots = T_n + t'_n = t$, where the T_i are the "large" sums, i.e., the sums of the (positive and negative) variables defined in T. Then the class $t'_1 = \ldots = t'_n$ necessarily has to be equal to 0, and hence the T_i's can be removed and added as $x - T_i = 0$ to the associated equations of T. The class $t'_1 = \ldots = t'_n$ will be inserted in the class of 0, although the later transformations of the segments where the variables of the t_i (and their splittings) are defined will turn these t_i into associated equations of those segments.

By processing the segments in this way, from left to right, when we arrive to the segment containing the class of 0, it is of the form $x = sum_1 > x = 0$, since no other variables can appear in this segment.

DEFINITION 34. *Let S be a linear system satisfying the AG-RPO assumptions. The following* splitting transformation *for S treats one segment T at the time, segment by segment from left to right. The rightmost segment (the one ending with 0) needs no treatment.*

After treating each segment T, the AG-RPO assumptions (as well as the definition of linear constraints) are assumed to be imposed eagerly in the segments to the right of T.

One can assume, inductively, that in every segment T' to the left of T, all variables that appear not below a free symbol in T' are defined in T' itself, and that no variable defined in T' appears in a segment to the right of T'. Also inductively, one can assume that in such segments T', every class containing a term headed with a free symbol is of the form $x = f(s)$ where s is a sum of variables. Let T be:

$$x_0 = s \;>\; x_1 = t_{1,1} = \ldots = t_{1,k_1} \;>\; \ldots \;>\; x_i = t_{i,1} = \ldots = t_{i,k_i}$$

$$> \; x_{i+1} = t_{i+1,1} = \ldots = t_{i+1,k_{i+1}} = t$$

and assume the transformation has already been applied to all segments left of T in S. Then the splitting transformation *for T in the AG-RPO case consists of the following steps:*

1. *If T is not the base segment, guess a subset of* split *variables of $\{x_1 \ldots x_i\}$ such that every sum in the class of a split variable contains either a split variable or a variable not defined in S.*
   (Note that this point is different from the FRPO case; in a class $x = y + z$, here it makes sense to split y and z while not splitting x: there can exist solutions σ where the small summands of $y\sigma$ and $z\sigma$ cancel each other out.)

2. *If x is a split variable, then introduce two new variables X and x', and everywhere in S replace x by $X + x'$. If x is a non-split variable of $\{x_0 \ldots x_{i+1}\}$, replace x everywhere in S by a new variable X.*
   (Note that in segments left of T this affects only terms of the form $f(v)$, due to the induction hypothesis. In segments right of T it does not affect terms headed with free symbols, due to assumption A5).

3. *After this, the equivalence classes e in the segment are either of the form $X + x' = v_1 = \ldots = v_k$ or of the form $X = v_1 = \ldots = v_k$. For each v_i there are sums V_i and v_i' such that $v_i =_{AG} V_i + v_i'$ and V_i is the sum of all upper case variables of v_i.*

   If e is such an equivalence class, we denote by E the class $X = V_1 = \ldots = V_k$ and by e' either the class $x' = v_1' = \ldots = v_k'$ or $0 = v_1' = \ldots = v_k'$ depending on whether the corresponding variable x has been split or not. Then we can write T as $e_0 > e_1 > \ldots > e_{i+1}$

and we can guess, for each relation $e_j > e_{j+1}$ whether (i) it is due to the large summands or (ii) (only if x_j is split) to the small ones. According to these guesses, replace T by the new segment T':

$$E_0 \ > \ E_1 \ \# \ \ldots \ \# \ E_{i+1}$$

where the relations $\#$ stand for $>$ or $=$ depending on the guesses made. Insert each e'_j in a segment to the right of T, adding it to an existing equivalence class (that will be the class of 0 iff x_j is not split) or creating a new class, in such a way that, whenever $E_j =_{T'} E_{j+1}$, either $e'_j > e'_{j+1}$ or e'_{j+1} does not exist.

4. *Let s be a term in an equivalence class to the right of T, and suppose that s is of the form $M + m$ or M, where M is a sum of positive and negative variables defined in T, and m does not contain any of these variables. Each such s is replaced by m, or the variable in the class of 0 if the part m of s is empty, and $M = 0$ becomes an associated equation of T.*

5. *Now the rightmost class of T is of the form $x = T_1 = \ldots = T_n = t$, where all the T_i are the "large" sums, i.e., the sums of the positive and negative variables defined in T. Replace this class by $x = t$, and add the equations $x - T_i = 0$ as associated equations of T.*

As in the FRPO case, after the splitting transformation assumptions $A2$, $A6$ and $A7$ can be assumed to have been reformulated into $A2'$, $A6'$ and $A7'$ respectively, and a new assumption is satisfied as well:

DEFINITION 35. *The* Split AG-RPO assumptions *about a linear system S are the AG-RPO assumptions where $A2$, $A6$ and $A7$ have been reformulated into $A2'$, $A6'$ and $A7'$ as in the split FRPO assumptions (Definition 19) plus the following additional assumption:*

AG3. For all terms $f(v)$ in S, where v is a sum of variables, no two variables in v appear in the same segment.

LEMMA 36. *Let S be a linear system satisfying the AG-RPO assumptions, and let $S_1, \ldots, S_n$ be the (disjunction of) linear systems resulting from the possible splitting transformations on S. Then each S_i satisfies the split AG-RPO assumptions.*

Hence in what follows we can assume that, after the splitting transformation, we deal with linear systems satisfying the split AG-RPO assumptions.

EXAMPLE 37. *Let us consider the signature* $h > - > + > 0$. *Suppose during the splitting transformation, just after splitting the variables of the leftmost segment, we obtain:*

$z = h(x_3) > x_3 > x_2 > x_1 > x_0 = x_3 - x_2 - x_1 + y_2 - y_1 - y_1 = h(y_1) >$
$y_3 = x_2 - x_1 - x_0 + y_2 + y_1 > y_2 > y_1 > h(w) > w = 0.$

At this point, if we assume that this splitting of variables has been done according to a solution σ, *then, all the* $x_i\sigma$ *contain top-level summands bigger than or equal to* $h(y_1)\sigma$, *and all the* $y_i\sigma$ *contain top-level summands smaller than* $h(y_1)\sigma$. *Since* $(x_3 - x_2 - x_1 + y_2 - y_1 - y_1)\sigma$ *must coincide with* $h(y_1)\sigma$, *the summands below the* $y_i\sigma$*'s must cancel each other out, i.e.* $(y_2 - y_1 - y_1)\sigma$ *must be* 0. *Therefore, continuing the process according to this solution, we remove* $y_2 - y_1 - y_1$ *from the sum* $x_3 - x_2 - x_1 + y_2 - y_1 - y_1$, *and add it to the class of* 0, *obtaining:*

$z = h(x_3) > x_3 > x_2 > x_1 > x_0 = x_3 - x_2 - x_1 = h(y_1) >$
$y_3 = x_2 - x_1 - x_0 + y_2 + y_1 > y_2 > y_1 > h(w) > w = y_2 - y_1 - y_1 = 0$

Now, in order to leave the treated segment in a normalized form $x_0 = h(y_1)$, *we remove the* $x_3 - x_2 - x_1$ *and we add* $x_0 - x_3 + x_2 + x_1 = 0$ *to the set of associated equations of this segment.*

Finally, since the term $x_2 - x_1 - x_0 + y_2 + y_1$ *is to the right of* $h(y_1)$, *and hence it must contain only summands smaller than* $h(y_1)\sigma$, *we have to force the* x_i*'s to cancel each other out. We remove* $x_2 - x_1 - x_0$ *and we add* $x_2 - x_1 - x_0 = 0$ *to the associated equations of the leftmost segment. Note that this is a different treatment with respect to what was done with* $y_2 - y_1 - y_1$ *before. But remember that the aim is to remove variables of the treated segment from the other segments to the right of it. In fact, this* $y_2 - y_1 - y_1$ *added to the class of* 0 *will be removed from this class when the next segment is treated, since none of the* y_i*'s is defined in the rightmost segment.*

After finishing the treatment of the leftmost segment we obtain:

$z = h(x_3) > x_3 > x_2 > x_1 > x_0 = h(y_1) >$
$y_3 = y_2 + y_1 > y_2 > y_1 > h(w) > w = y_2 - y_1 - y_1 = 0$

and the associated equations $x_0 - x_3 + x_2 + x_1 = 0$ *and* $x_2 - x_1 - x_0 = 0$. □

6.1.3. *Diophantine equations.*

Example 31 shows that now in solutions more than one summand may be needed in a single segment. But only a certain small number of summands play an important role in the comparisons of a segment. These summands will be called the *decisive* ones.

EXAMPLE 38. *If* $a > b > c$, *in the inequation* $a + a + a + b + b + c \succ a + a + a - c - c - c$ *the summand* b *will be called the* decisive summand, *since it is the largest sumand that appears in both terms with a different*

number of occurrences. Note that in each comparison there is exactly one decisive summand. □

DEFINITION 39. *Let s be a term and u a summand. The* number of occurrences *of u in s (notation $\#(u,s)$) is the integer n such that $s =_{AG} nu+s'$, where u is not a top-level summand of s'.*

EXAMPLE 40. *We have $\#(a, f(a+b)-a-a) = -2$.* □

DEFINITION 41. *Let s and t be two ground terms such that $s \succ t$. The* decisive summand *of the inequation $s \succ t$ is the top-level summand u such that, for all summands v with $v \succ_{agrpo} u$, we have $\#(v,s) = \#(v,t)$, and either (i) $\#(u,s) > \#(u,t) \geq 0$ or (ii) $\#(u,s) < \#(u,t)$ and $\#(u,s) < 0$.*

DEFINITION 42. *Let S be a linear system satisfying the split AG-RPO assumptions, and let T be a segment in S of the form*

$$x_0 = s \;>\; x_1 = t_{1,1} = \ldots = t_{1,k_1} \;> \ldots >\; x_i = t_{i,1} = \ldots = t_{i,k_i} \;>\; x_{i+1} = t$$

Let σ be a solution of S.

Then the number ndec for this σ and T is the cardinality of $dec(T\sigma) \cup \{t\sigma\}$, where $dec(T\sigma)$ is the set of decisive summands in the inequations $x_j\sigma \succ x_{j+1}\sigma$ with $j > 0$.

For given T and a solution σ, the number *ndec* is, roughly, the number of different summands that are decisive in some comparison of the segment. Since there is at most one decisive summand per comparison, when σ is unknown one can guess *ndec* to be between 1 and $i+1$. There are some cases of segments where it must be exactly 1, which is when we know that for all solutions σ we have $s\sigma = succsum_1(t\sigma)$, i.e., that $s\sigma$ is the successor summand of $t\sigma$:

- s is some c_j and t is c_{j+1}, or
- t is c_1 and s is $h(0)$, or
- t is sum_1 and s is sum_2, or
- t is headed with some f with $f > h$ and s is $h(x_{i+1})$.

DEFINITION 43. *Let S, T and σ be as in the previous definition. In the following, the elements of $dec(T\sigma) \cup \{t\sigma\}$ are denoted (and ordered) by $u_{ndec} \succ_{agrpo} \ldots \succ_{agrpo} u_1$.*

Intuitively, if the splitting has been done according to σ, then, for such S, T and σ, always $t\sigma$ is u_1. Once the splitting transformation has been applied to S, we can define a system of diophantine equations and inequations D_S for S. For a segment T as the one in the previous definitions, in the definition of D_S below, for every variable x_j with $1 \leq j \leq i+1$, $ndec$ integer variables $x_{j,1}, \ldots, x_{j,ndec}$ are created; intuitively, these variables represent the number of occurrences of each decisive summand in x_j. For the segments where $ndec$ is 1, the variable name x_j is used as well for the corresponding integer variable.

EXAMPLE 44. *Consider* $f > h > - > + > 0$ *and suppose that after the splitting transformation we have:*

$z_4 = h(w_1 + x_2) > w_6 = -w_5 > w_5 > w_4 = -w_3 > w_3 > w_2 = -w_1 >$
$w_1 = f(z_3) >$
$z_3 = h(x_3) > y_4 = -y_3 > y_3 > y_2 = -y_1 > y_1 = h(x_2) >$
$z_2 = h(x_1) > x_3 > x_2 > x_1 = h(z_1) > z_1 = 0$

Now we want to find a solution σ *such that for every variable it contains only summands greater than or equal to the rightmost term of the segment where it is defined. We may guess that the number of decisive summands for the leftmost segment is* 3*. Therefore, we need to guarantee that at least two summands between* $h(w_1 + x_2)\sigma$ *and* $f(z_3)\sigma$ *exist. Note that the only situations where not enough room might be available between two such terms* s *and* t *occur when* s *is a (*k*-th) successor summand of* t*. It is analyzed in Example 32 when this can happen. For example, the successor summand of* $f(z_3)\sigma$ *is* $h(f(z_3))\sigma$ *and the next one is* $h(f(z_3) + h(0))\sigma$*. Since* x_2 *is a variable in the base segment (and hence* $x_2\sigma$ *necessarily is of the form* $sum_1 + \ldots + sum_1$ *or* $-sum_1 - \ldots - sum_1$*, where in this signature* sum_1 *is* $h(0)$*), we need* $x_2\sigma$ *to be greater than or equal to* $h(0) + h(0)$ *in order to make sure that there is enough room between* $h(w_1 + x_2)\sigma$ *and* $f(z_3)\sigma$*. This illustrates the need of adding to the diophantine system either an equation of the form* $x_2 \geq 2$ *or one of the form* $x_2 < 0$*, since* $-h(0)$ *is greater than any sum of positive* $h(0)$*'s.*

Later on, we may guess that the number of decisive summands for the segment $z_3 = h(x_3) > y_4 = -y_3 > y_3 > y_2 = -y_1 > y_1 = h(x_2)$ *is* 2*. We need to guarantee that there exists at least one summand between* $h(x_3)\sigma$ *and* $h(x_2)\sigma$*. Observe that* x_3 *and* x_2 *are defined in the base segment. If we guess* $x_2\sigma$ *to be* $h(0) + \ldots + h(0)$*, then either* $x_3\sigma$ *is also of the form* $h(0) + \ldots + h(0)$ *with at least two more* $h(0)$*'s than* $x_2\sigma$*, or* $x_3\sigma$ *is of the form* $-h(0) - \ldots - h(0)$*. If we guess that* $x_2\sigma$ *is* $-h(0) - \ldots - h(0)$*, then* $x_3\sigma$ *also has to be* $-h(0) - \ldots - h(0)$*, but with at least two more* $-h(0)$*'s than* $x_2\sigma$*.* □

DEFINITION 45. *Let S be a linear system satisfying the split AG-RPO assumptions, and let T be a segment in S of the form*

$$x_0{=}s \;>\; x_1{=}t_{1,1}{=}\ldots{=}t_{1,k_1} \;>\; \ldots \;>\; x_i{=}t_{i,1}{=}\ldots{=}t_{i,k_i} \;>\; x_{i+1}{=}t$$

with associated equations $q_1 = 0, \ldots, q_l = 0$.

The following equations are added to the system D_S in order to express for which inequation which decisive summand is decisive, and whether it decides positively or negatively:

1. *For each j between 1 and i, we guess which index summand k between 1 and $ndec$ is the decisive one for the inequation $x_j > x_{j+1}$. Now, for all $k' > k$ we add the equation $x_{j,k'} = x_{j+1,k'}$. In order to decide the way in which the k-th summand is decisive, we guess adding either (i) $x_{j,k} > x_{j+1,k} > 0$ or (ii) $x_{j,k} < x_{j+1,k}$ and $x_{j,k} < 0$.*

2. *Let $t^k_{j,l}$ be the result of replacing in $t_{j,l}$ every variable $x_{j'}$ by $x_{j',k}$, that is, the integer variable corresponding to the k-th decisive summand (with this replacement, we are transforming a term into an expression with integer variables). Now in order to make sure that the number of occurrences of the k-th summand at each side of the equalities coincides, add $x_{j,k} = t^k_{j,l}$, for all j in $\{1 \ldots i\}$, and all k in $\{1 \ldots ndec\}$, and all l in $\{1 \ldots k_j\}$. We proceed identically with the associated equations.*

3. *We add $x_{i+1,1} = 1$, and for all k in $\{2 \ldots ndec\}$ we add $x_{i+1,k} = 0$.*

We now impose some more diophantine equations ensuring that, in a solution σ, there will be enough space for the decisive summands between $s\sigma$ and $t\sigma$, when $ndec > 1$ (as illustrated in Example 44). Assume $ndec > 1$ and let y and z always denote variables defined in the base segment:

4. *If s is of the form $h(y{+}s')$ and t is of the form $h(z{+}s')$, it has to be guessed whether one adds either the equations (i) $y \geq z{+}ndec$ and $z \geq 0$, or the equations (ii) $y \leq z{-}ndec$ and $z < 0$, or the equations (iii) $y < 0$ and $z \geq 0$ (in these equations y and z are the integer variables corresponding to the term variables with the same name defined in the base segment).*

5. *If s is of the form $h(y + s')$ and t is of the form $h(s')$, there is another choice between the equation (i) $y \geq ndec$, and the equation (ii) $y < 0$.*

6. *If s is of the form $h(x_{i+1}{+}y)$ and t is of the form $f(t')$, either the equation (i) $y \geq ndec - 1$ or (ii) $y < 0$ is added.* □

6.1.4. *Constraint Solving.*
LEMMA 46. *Let S be a linear system satisfying the AG-RPO assumptions. Let $D_1 \ldots D_m$ be the diophantine systems generated from all the linear systems satisfying the split AG-RPO assumptions obtained by applying the splitting transformation to S. Then S is satisfiable if, and only if, some D_i is satisfiable over the integers.*

Proof: $\Longrightarrow$: As for the AC case, assuming that S is satisfiable by some substitution σ, we find some S' resulting from a splitting transformation of S according to σ, and some extension of σ that is a solution of S'; and, moreover, given a segment T of S' of the form

$$x_0{=}s \;>\; x_1{=}t_{1,1}{=}\ldots{=}t_{1,k_1} \;>\ldots>\; x_i{=}t_{i,1}{=}\ldots{=}t_{i,k_i} \;>\; x_{i+1}{=}t$$

for all j in $\{1 \ldots i{+}1\}$ we have that all the $x_j\sigma$ contain only summands greater than or equal to $t\sigma$.

Consider the set $C = \{u_1, \ldots, u_n\}$ including $t\sigma$ and all the different top-level summands of these variables that are decisive in some inequation $x_j\sigma \succ_{agrpo} x_{j+1}\sigma$, for some j in $\{1, \ldots, i\}$, and where $u_n \succ_{agrpo} \ldots \succ_{agrpo} u_1$. Let D' be the diophantine system obtained from S' by doing the guessings according to σ: the chosen $ndec$ is precisely the cardinality n of C for such segments T, and the chosen index for the generated equations from the inequation $x_j > x_{j+1}$ is k if u_k is the decisive summand in $x_j\sigma \succ_{agrpo} x_{j+1}\sigma$, etc. Now, if for such segments T we assign $\#(u_k, x_j\sigma)$ to each variable $x_{j,k}$, then we obtain a solution for D'.

$\Longleftarrow$: Assume that some guessed $D_{S'}$ is satisfiable for some S' resulting from a splitting transformation of S. Let θ be a solution for $D_{S'}$. We can inductively build a solution σ for S' as follows. For each segment T in S of the form

$$x_0{=}s \;>\; x_1{=}t_{1,1}{=}\ldots{=}t_{1,k_1} \;>\ldots>\; x_i{=}t_{i,1}{=}\ldots{=}t_{i,k_i} \;>\; x_{i+1}{=}t$$

assume that a (partial) solution σ has already been defined for the linear system consisting of all segments to the right of T. Then $t\sigma$ is already defined. Let $ndec$ denote the guessed $ndec$ for T. Now, for all j in $\{1, \ldots, i{+}1\}$ we define $x_j\sigma$ to be the sum containing $x_{j,k}\theta$ times the summand $succsum_k(t\sigma)$ for each k in $\{1, \ldots, ndec\}$, and no other summands.

By construction of the diophantine system of equations, σ satisfies all equality relations in S', and all the relations $x_j \succ_{agrpo} x_{j+1}$ with j in $\{1, \ldots, i\}$ for this segment T, and hence it only remains to be checked that σ satisfies $s\sigma \succ_{agrpo} x_1\sigma$. This is done by distinguishing six cases:

1. All cases where $ndec$ is 1.
   Then $s\sigma$ is greater than a sum of positive or negative $t\sigma$'s, and then it is greater than $x_1\sigma$.

2. s and t are of the form $h(s')$ and $h(t')$, respectively, and T is not the base segment.
   By assumption $A7'$, either (a) $x \geq_S h(t')$ for some variable x in s', or (b) $s' >_{segs(S)} t'$. In case (a), by assumption $AG3$, this variable x is the only one in s' satisfying $x \geq_S h(t')$. Therefore, no summand in $x\sigma$ is canceled out in $s'\sigma$, and hence $h(s')\sigma \succeq_{agrpo} h(x\sigma) \succ_{agrpo} x\sigma \succeq_{agrpo} h(t')\sigma$. In case (b), s' and t' are of the form $y_1 + \ldots$ and $z_1 + \ldots$ respectively, and there exists some k such that $y_j = z_j$ for all $j < k$, and $y_k >_S z_k$ if z_k exists, and all the z_j with $j > k$ appear in segments to the right of the one where y_k is defined. In fact, all the y_j's are in different segments, and the same for all the z_j's. Therefore, no summands in $s'\sigma$ cancel each other out, and the same for the summands in $t'\sigma$, and hence $s'\sigma \succ_{agrpo} t'\sigma$, and $y_k\sigma$ contains the decisive summand in this comparison. If y_k is not defined in the base segment, then $h(s')\sigma \succ_{agrpo} succsum_l(h(t')\sigma)$ for all natural numbers l, and we conclude since $x_1\sigma$ is a sum of positive and negative summands of this form. Assume now that y_k is defined in the base segment. If z_k does not exist, then either the equation (i) $y_k \geq ndec$, or the equation (ii) $y_k < 0$ has been added to the diophantine system. In both cases, by the induction hypothesis, $h(s')\sigma \succ_{agrpo} succsum_l(h(t')\sigma)$ for all l in $1 \ldots ndec-1$, and since $x_1\sigma$ is a sum of positive and negative summands of this form, we conclude again. If z_k exists, then either the equations (i) $y_k \geq z_k + ndec$ and $z_k \geq 0$, or the equations (ii) $y_k \leq z_k - ndec$ and $z_k < 0$, or the equations (iii) $y_k < 0$ and $z_k > 0$ have been added to the diophantine system, and by analogous arguments we conclude again.

3. s and t are of the form $f(s')$ and $g(t')$, respectively, with $g > f > h$.
   Then, by assumptions $A6'$ and $AG3$, exactly one variable x in s' is defined in this segment. Therefore the summands in $x\sigma$ do not cancel out with other summands in $s'\sigma$, and hence $f(s')\sigma \succ_{agrpo} x\sigma \succeq_{agrpo} g(t')\sigma$. Moreover, $f(s')\sigma$ is bigger w.r.t. AG-RPO than any sum of positive and negative summands of the form $h(g(t') + sum_1 + \ldots + sum_1)\sigma$, and hence $f(s')\sigma \succ_{agrpo} x_1\sigma$.

4. s and t are of the form $h(s')$ and $f(t')$, respectively.
   Then, by assumptions $A6'$ and $AG3$, exactly one variable x in s' is defined in this segment. If x is not x_{i+1}, then $x\sigma \succeq_{agrpo} f(t')\sigma + f(t')\sigma$. Note that $f(t')\sigma + f(t')\sigma$ is the smallest sum of positive and

negative summands $succsum_l(f(t')\sigma)$ (with $l \geq 0$) that is neither $f(t')\sigma$ nor 0.

Since the summands in $x\sigma$ do not cancel out with any summands in $s'\sigma$, we have that $h(s'\sigma) \succeq_{agrpo} h(f(t')\sigma + f(t')\sigma)$, and since $h(f(t')\sigma + f(t')\sigma)$ is greater than any term of the form $h(f(t') + sum_1 + \ldots + sum_1)\sigma$ we have that $s\sigma \succ_{agrpo} x_1\sigma$, that is a sum of positive and negative summands of such a form. Assume now that x is x_{i+1}. Then s is of the form $h(x_{i+1} + s'')$. If s'' contains some variable not defined in the base segment, since the variables in s'' are not canceled out when applying σ, we have that $s''\sigma$ is bigger than any term of the form $sum_1 + \ldots + sum_1$. For similar reasons, $h(s')\sigma \succ_{agrpo} x_1\sigma$, since $x_1\sigma$ is a sum of positive and negative summands of the form $h(x_{i+1} + sum_1 + \ldots + sum_1)\sigma$ or $x_{i+1}\sigma$. Assume now that s is of the form $h(x_{i+1} + y)$, where y is a variable defined in the base segment. Then either the equation (i) $y \geq ndec - 1$, or the equation (ii) $y < 0$ has been added to the diophantine system. In both cases, by the induction hypothesis, $y\sigma \succeq_{agrpo} sum_1 + \ldots^{ndec-1)} + sum_1$. Since $x_1\sigma$ is a sum of positive and negative summands smaller than $h(x_{i+1} + sum_1 \ldots^{ndec-1)} + sum_1)\sigma$, and the summands in $(x_{i+1} + y)$ do not cancel each other out, we have $s\sigma \succ_{agrpo} x_1\sigma$.

5. s and t are of the form $f(s')$ and $f(t')$, respectively, for $f > h$.
   If s' contains some variable defined in T, we proceed as in previous cases. Otherwise, assume that no variable in s' is defined in T. By assumption $A7'$ and the induction hypothesis we have that $s'\sigma \succ_{agrpo} t'\sigma$. Therefore $f(s')\sigma$ is greater than any summand of the form $h(f(t') + sum_1 + \ldots + sum_1)\sigma$, and we conclude again.

6. s is of the form $f(s')$ and t is headed with h or with some g such that $f > g$.
   If some variable in s' is defined in this segment, we proceed as in previous cases. Then, assume that no variable in s' is defined in the segment T. It suffices to show that $s\sigma \succ_{agrpo} t\sigma$, since automatically we obtain $s\sigma \succ_{agrpo} succsum_k(t\sigma)$ for all natural numbers k. What we do is proving that $s\sigma \succ_{agrpo} t'\sigma$, where t' is either t or any other term to the right of t, and it is done inductively from right to left. Clearly $s\sigma$ is bigger than 0, all the c_j, and all the $x\sigma$ for variables x in the segments delimited by some c_j. If t' is headed by h or g' such that $f > g'$, we obtain $s\sigma \succ_{agrpo} t'\sigma$ inductively. If t' is headed by f, we obtain $s\sigma \succ_{agrpo} t'\sigma$ as in previous case. If t' is headed by g' such that $g' > f$, then, some variable x in s' is defined in a segment to the left of t' in S', but to the right of t, and no other

variable in s' is defined in this segment (no cancellation is possible). Therefore we obtain $s\sigma \succ_{agrpo} t'\sigma$ inductively. If t' is a sum of variables $y_1 + \ldots + y_{k'} - y_{k'+1} - \ldots - y_k$, then, every $y_j\sigma$ is a sum of positive and negative summands of the form $succsum_l(t''\sigma)$, for some t'' that, by the induction hypothesis, satisfies $s\sigma \succ_{agrpo} t''\sigma$. Moreover, since s is headed by f with $f > h$, we have $s\sigma \succ_{agrpo} succsum_l(t''\sigma)$. Therefore, $s\sigma \succ_{agrpo} (y_1 + \ldots + y_k - y_{k'+1} - \ldots - y_k)\sigma$.

This σ can be extended to a solution for S analogously to the FRPO case. □

THEOREM 47. *The satisfiability problem for AG-RPO constraints restricted to signatures with free symbols of arity* 0 *or* 1 *is in NP.*

6.2. Arbitrary arities

The extension to arbitrary signatures is obtained by proceeding analogously to the AC case. What has to be taken into account is that $succsum_1(f(s_1, \ldots, s_k))$ is $h(0, \ldots, 0, f(s_1, \ldots, s_k))$, and $succsum_1(h(s_1, \ldots, s_k))$ is $h(s_1, \ldots, s_k + sum_1)$ if s_k is not of the form $s' - sum_1$, and $h(s_1, \ldots, s_k - sum_1)$ otherwise.

THEOREM 48. *The satisfiability problem for AG-RPO constraints is in NP.*

7. Hardness

Obviously, the satisfiability problems we deal with are NP-hard, because as subcases they include the AC, AC0 and AG-unifiability problems. But one may wonder whether there exists any ordering at all for these E such that at least the satisfiability problem for positive conjunctions of inequations (by which one cannot always encode unification) is in P. Here we answer this question negatively (by reducing 1-in-3-sat with only positive literals), even if monotonicity of the ordering is not required.

THEOREM 49. *Let E be AC, AC0, or AG, and let $\succ$ be any arbitrary well-founded E-compatible ordering on ground terms that is total up to $=_E$. Then the constraint satisfiability problem for $\succ$ and $=_E$ is NP-hard even for constraints that are conjunctions of positive inequations.*

Proof: By reducing 1-in-3-sat with only positive literals. We build a conjunction C_P that is satisfiable if, and only if, the 1-in-3-sat problem P is satisfiable. Let t_3, t_2 and t_1 be the three smallest ground terms w.r.t. $\succ$ such that $t_3 \succ t_2 \succ t_1$. Furthermore, let $min(E)$ and $max(E)$ denote the minimal and maximal term (w.r.t. $\succ$), respectively, of the set $E = \{\ t_1{+}t_1{+}t_1,\ \ t_1{+}t_2{+}t_2,\ \ t_2{+}t_2{+}t_2\ \}$.

Now, for each variable x_i in P, we add an inequation $t_3 > x_i$ to C_P, forcing x_i to be either t_1 or t_2. Furthermore, for each clause in P of the form $x_i \vee x_j \vee x_k$, we force exactly one of its three variables to be t_2, by forcing $x_i{+}x_j{+}x_k$ to be $t_1{+}t_1{+}t_2$:

- if $min(E) \succ t_1{+}t_1{+}t_2$, then add $min(E) > x_i{+}x_j{+}x_k$.
- if $t_1{+}t_1{+}t_2 \succ max(E)$, then add $x_i{+}x_j{+}x_k > max(E)$.
- Otherwise, add $u \ > \ x_i{+}x_j{+}x_k \ > \ v$, where u is the smallest term in E larger than $t_1{+}t_1{+}t_2$, and v is the largest term smaller than $t_1{+}t_1{+}t_2$. □

8. Other kinds of constraints

Our algorithms for dealing with equality and ordering constraints are also extendable to other kind of constraints that are very useful for the applications of (Godoy and Nieuwenhuis, 2000). Let us explain how. In our algorithms, for each variable x the number of different summands at top-level position that appear in the generated solution for x is guessed. Moreover, in some cases it is guessed if such a summand appears positively or negatively. All these guessings can be restricted by additional constraints of the form "the variable x contains only one different summand", or of the form "the variable x contains exactly one summand that is negative", or "all the variables of the sum $x + y - z$ contain only one different summand, the same one for all of them". By doing the guessings adequately to such new constraints, the generated solution will satisfy them, and the diophantine system will be satisfiable if and only if there exists a solution for the set of constraints including the new ones.

Such constraints are useful in an inference system like the one for abelian groups presented in (Godoy and Nieuwenhuis, 2000). For example, in such an inference system equations of the form $e \simeq 0$ are *oriented* into logically equivalent ones $s = t$ where s contains summands and variables, and the only accepted substitutions for such variables are the ones such that all of these variables contain one different summand,

and the same one for all of them. This summand also has to coincide with the rest of summands in s when applying a substitution (that is a solution). Additionally one wants all summands that do not cancel each other out in t to be smaller than the summand appearing in s. The techniques of (Godoy and Nieuwenhuis, 2000) can be combined with the ones presented here for obtaining an efficient inference system for abelian groups.

9. Conclusions and further work

Constraint solving algorithms have been defined for FRPO-based orderings for abelian semigroups, abelian monoids and abelian groups. We believe that the new techniques will lead to reasonably efficient practical algorithms for these orderings. This, as well as building an implementation, is one of the directions for further research.

Finally, we expect that the ideas given here will provide new insights for the development of constraint solving methods over fixed signatures for other E-compatible orderings.

References

Bachmair, L. and D. A. Plaisted: 1985, 'Termination orderings for associative-commutative rewriting systems'. *Journal of Symbolic Computation* **1**, 329–349.

Comon, H.: 1990, 'Solving Symbolic Ordering Constraints'. *International Journal of Foundations of Computer Science* **1**(4), 387–411.

Comon, H., R. Nieuwenhuis, and A. Rubio: 1995, 'Orderings, AC-Theories and Symbolic Constraint Solving'. In: *10th IEEE Symposium on Logic in Computer Science (LICS)*. San Diego, USA, pp. 375–385.

Dershowitz, N.: 1982, 'Orderings for Term-Rewriting Systems'. *Theoretical Computer Science* **17**(3), 279–301.

Dershowitz, N. and J.-P. Jouannaud: 1990, 'Rewrite Systems'. In: J. van Leeuwen (ed.): *Handbook of Theoretical Computer Science*, Vol. B: Formal Models and Semantics. Amsterdam, New York, Oxford, Tokyo: Elsevier Science Publishers B.V., Chapt. 6, pp. 244–320.

Godoy, G. and R. Nieuwenhuis: 2000, 'Paramodulation with Built-in Abelian Groups'. In: *15th IEEE Symposium on Logic in Computer Science (LICS)*. Santa Barbara, USA, pp. 413–424, IEEE Computer Society Press.

Godoy, G. and R. Nieuwenhuis: 2001, 'Ordering Constraints for Deduction with Built-in Abelian Semigroups, Monoids and Groups'. In: *16th IEEE Symposium on Logic in Computer Science (LICS)*. Boston, USA, pp. 38–47, IEEE Computer Society Press.

Jouannaud, J.-P. and M. Okada: 1991, 'Satisfiability of systems of ordinal notations with the subterm property is decidable'. In: *18th International Colloquium Automata, Languages and Programming (ICALP)*. Madrid, Spain, pp. 455–468, Springer-Verlag.

Kirchner, C., H. Kirchner, and M. Rusinowitch: 1990, 'Deduction with Symbolic Constraints'. *Revue Française d'Intelligence Artificielle* **4**(3), 9–52.

Korovin, K. and A. Voronkov: 2000, 'A Decision Procedure for the Existential Theory of Term Algebras with the Knuth-Bendix Ordering'. In: *15th IEEE Symposium on Logic in Computer Science (LICS)*. Santa Barbara, USA, pp. 291–302, IEEE Computer Society Press.

Korovin, K. and A. Voronkov: 2001, 'Knuth-Bendix constraint solving is NP-complete'. In: *Automata, Languages and Programming, International Colloquium, 28th Int. colloquium, (ICALP)*. Crete, Greece, Springer-Verlag.

Marché, C.: 1996, 'Normalized Rewriting: An Alternative to Rewriting Modulo a Set of Equations'. *Journal of Symbolic Computation* **21**(3), 253–288.

McCune, W.: 1997, 'Solution of the Robbins Problem'. *Journal of Automated Reasoning* **19**(3), 263–276.

Narendran, P., M. Rusinowitch, and R. Verma: 1999, 'RPO Constraint Solving Is in NP'. In: G. Gottlob, E. Grandjean, and K. Seyr (eds.): *12th Int. Conference of the European Association of Computer Science Logic (CSL'98)*. Brno, Czech Republic, pp. 385–398, Springer-Verlag.

Nieuwenhuis, R.: 1993, 'Simple LPO constraint solving methods'. *Information Processing Letters* **47**, 65–69.

Nieuwenhuis, R. and A. Rubio: 1995, 'Theorem Proving with Ordering and Equality Constrained Clauses'. *Journal of Symbolic Computation* **19**(4), 321–351.

Nieuwenhuis, R. and A. Rubio: 1997, 'Paramodulation with Built-in AC-Theories and Symbolic Constraints'. *Journal of Symbolic Computation* **23**(1), 1–21.

Nieuwenhuis, R. and A. Rubio: 2000, 'Paramodulation-based theorem proving'. In: J. Robinson and A. Voronkov (eds.): *Handbook of Automated Reasoning*. Elsevier Science Publishers (to appear).

Rubio, A. and R. Nieuwenhuis: 1995, 'A total AC-compatible ordering based on RPO'. *Theoretical Computer Science* **142**(2), 209–227.

Schrijver, A.: 1987, *Theory of Linear and Integer Programming*. New York: John Wiley and Sons.

Vigneron, L.: 1994, 'Associative Commutative Deduction with constraints'. In: A. Bundy (ed.): *12th International Conference on Automated Deduction (CADE)*. Nancy, France, pp. 530–544, Springer-Verlag.