Automated Deduction with Built-in Theories

Completeness Results
and
Constraint Solving Techniques

Tesi doctoral presentada al
Departament de Llenguatges i Sistemes Informàtics
de la Universitat Politècnica de Catalunya

per a optar al grau de
Doctor en Informàtica

per
Guillem Godoy Balil

sota la direcció del doctor
Robert Nieuwenhuis

Barcelona, 1 de setembre de 2001

Aquesta tesi fou llegida el dia 11 de octubre de 2001, davant el tribunal de tesi format per

– Dr. Fernando Orejas	(President)
– Dr. Albert Rubio	(Secretari)
– Dr. Harald Ganzinger	
– Dr. Hubert Comon	
– Dr. Salvador Lucas	

Acknowledgments

Agraeixo a en Roberto tota la ajuda que m'ha prestat al llarg d'aquests anys de tesi, el tracte de companyerisme que ha tingut amb mi, i la seva paciència infinita (o arbitràriament gran). Igualment agraeixo tota la ajuda i el companyerisme que m'han donat la resta de persones relacionades amb el departament, com l'Albert, la Cristina, el Miquel (o Mikuel) el Mateu (espero emboscada), el Jose Miguel i en Josep.

Agraeixo el recolçament de la meva família, l'Anna, el Jesus, l'Oscar i el Jesuset. Però també, per la seva amistat al llarg d'aquests anys, a moltes altres persones, com a la Rosa, l'Anna i la Núria (de també paciència infinita); a l'Alex, l'Oriol, l'Oscar i en Kapdou; a l'Omer i en Ferran; al Manel, el Jaume i el David.

Along the last five years, this work has been partially supported by the ESPRIT Basic Research Action CCL-II, ref. WG # 22457. and the Spanish CICYT project HEMOSS ref. TIC98-0949-C02-01, and by Departament d'Universitats, Recerca i Societat de la Informació de la Generalitat de Catalunya.

Contents

Chapter 1

Introduction and Outline

Automated deduction techniques in first-order logic with equality are applied in many subfields of mathematics and computer science, like (constraint) logic and functional programming, synthesis and verification of hardware and software, security of communication, data bases and knowledge-based systems or computational linguistics.

Hence it is not surprising that during the last decade the field has seen important progress in the development of new theoretical insights, like new inference systems, stronger completeness results, and complexity and decidability issues. Most of this research has focussed on paramodulation calculi with ordering and equality constraints, built-in equational theories and/or redundancy elimination techniques.

In this thesis, several new contributions are described concerning automated deduction with built-in theories. Some fundamental new completeness results are obtained, and new constraint solving techniques for enhancing the efficiency of the inference systems are developed.

In this chapter, first the area is introduced in an informal way, based on [NR01], with emphasis on the intuitive background of the different techniques. After this informal introduction, the structure of this thesis is outlined.

1.1 Paramodulation

Paramodulation originated as a development of resolution [Rob65], one of the main computational methods in first-order logic (see [BG01a]). Robinson showed that resolution together with factoring is *refutation complete*, that is, the empty clause will eventually be inferred by systematically enumerating all consequences of an unsatisfiable set of clauses by *(binary) resolution:*

$$\frac{C \vee A \qquad D \vee \neg B}{(C \vee D)\sigma} \qquad \text{if } \sigma = mgu(A, B)$$

where $mgu(A,B)$ denotes a most general unifier of A and B, and *factoring*:

$$\frac{C \vee A \vee B}{(C \vee A)\sigma} \qquad \text{if } \sigma = mgu(A,B)$$

For dealing with the equality predicate $\simeq$ by resolution, one can specify it by means of the following *congruence axioms* $\mathcal{E}$:

$$\begin{array}{rll}
 & \rightarrow x \simeq x & \textit{(reflexivity)} \\
x \simeq y & \rightarrow y \simeq x & \textit{(symmetry)} \\
x \simeq y \wedge y \simeq z & \rightarrow x \simeq z & \textit{(transitivity)} \\
x_1 \simeq y_1 \wedge \ldots \wedge x_n \simeq y_n & \rightarrow f(x_1,\ldots,x_n) \simeq f(y_1,\ldots,y_n) & \textit{(monotonicity-I)} \\
x_1 \simeq y_1 \wedge \ldots \wedge x_n \simeq y_n & & \\
\wedge P(x_1,\ldots,x_n) & \rightarrow P(y_1,\ldots,y_n) & \textit{(monotonicity-II)}
\end{array}$$

In fact the monotonicity axioms are axiom *schemes*: one *monotonicity-I* axiom is required for each non-constant n-ary function symbol f, and, similarly, one *monotonicity-II* axiom for each predicate symbol P. A set S of clauses is satisfiable in first-order logic with equality if, and only if, $S \cup \mathcal{E}$ is satisfiable in first-order logic without equality[1].

However, resolution and factoring inferences with $\mathcal{E}$ tend to cause the generation of too many (mostly unnecessary) new clauses. Therefore, Robinson and Wos explored another possibility. They tried to avoid the need for specifying equality by treating it as part of the logical language, i.e., directly considering first-order logic with equality. This requires the design of dedicated inference rules, like *paramodulation* [RW69]:

$$\frac{C \vee s \simeq t \qquad D}{(\,C \vee D[t]_p\,)\sigma} \qquad \text{if } \sigma = mgu(s, D|_p)$$

where $D|_p$ is the subterm of D at position p, and $D[t]_p$ denotes the result of replacing in D this subterm by t. Paramodulation, together with resolution and factoring, was proved refutation complete, under the presence of the reflexivity axiom and certain tautologies called the *functional reflexivity* axioms

$$f(x_1,\ldots,x_n) \simeq f(x_1,\ldots,x_n)$$

for every n-ary function symbol f of the alphabet. Later on, Brand [Bra75] proved that the functional reflexivity axioms were unnecessary, as well as paramodulation *into variables*, that is, paramodulations where $D|_p$ is a variable. However, even

[1] Note that there is no logical equivalence. First-order logic (FOL) with equality has more expressive power: for instance, in FOL with equality the clause $x \simeq a \vee x \simeq b$ expresses that the cardinality of models is at most two, which cannot be expressed in FOL without equality.

under these restrictions, paramodulation is difficult to control: unless additional refinements are considered, it quickly produces a large amount of unnecessary clauses, expanding the search space excessively.

The strengths and weaknesses of paramodulation have led to fruitful theoretical research on paramodulation-based theorem proving, and a large number of experiments with paramodulation have been performed at the Argonne group by Wos, Overbeek, Henschen and others (see, e.g., [Wos88, Wos96] for references), and especially by McCune with his provers Otter [McC94] and EQP [McC97a] and his recent automated proof of the Robbins conjecture [McC97c, McC97b].

1.1.1 Knuth-Bendix completion

An important tool in paramodulation is the use of *term orderings* for restricting the number of inferences. Paramodulation is in fact based on Leibniz' law for replacement of equals by equals. Now the basic idea of *ordered* paramodulation is to only perform replacements of *big* terms by *smaller* ones, with respect to the given ordering $\succ$.

This is precisely the idea of (ordered) *rewriting.* Let us consider now unit equations: we address *word problems* of the form $E \models u \simeq v$, where E is a set of equations and $u \simeq v$ is another equation. Assume that $\succ$ is a *reduction ordering* on terms (see Chapter 2 for the precise definitions). A term t is rewritten in one step with an equation $l \simeq r$ (or, equivalently, $r \simeq l$) of E by replacing a subterm $l\sigma$ of t by $r\sigma$, for some substitution σ such that $l\sigma \succ r\sigma$. For example, let E consist of the equations $plus(0,x) \simeq x$ and $plus(s(x),y) \simeq s(plus(x,y))$. Denoting each step by $\rightarrow_E$ (and assuming the steps agree with $\succ$), we have

$$plus(s(s(0)),s(0)) \rightarrow_E s(plus(s(0),s(0))) \rightarrow_E s(s(plus(0,s(0)))) \rightarrow_E s(s(s(0)))$$

This (ordered) rewrite relation terminates: starting from some finite term t, after a finite number of steps a *normal form* (i.e., a term that cannot be rewritten any further) is obtained.

Now let $\rightarrow_E^*$ denote zero or more of these steps (i.e., $\rightarrow_E^*$ is the reflexive-transitive closure of the relation $\rightarrow_E$). A set of equations E is called *confluent* w.r.t. the given $\succ$ if, whenever $s \rightarrow_E^* u$ and $s \rightarrow_E^* v$, there is some t such that $u \rightarrow_E^* t$ and $v \rightarrow_E^* t$. It is not difficult to see that then every term has a unique normal form. Furthermore, rewriting is then a decision procedure for deduction in the theory of E, since $E \models s = t$ if, and only if, s and t have the same normal form[2].

The first instances of ordered paramodulation appeared in *Knuth-Bendix completion* [KB70]. Roughly, a completion procedure attempts to transform a given

[2] More precisely, one rewrites the ground Skolemisations of s and t, and $\succ$ is required to be total on such ground terms.

set of equations into an equivalent confluent one. A crucial step of the transformation process is the computation of *critical pairs* between equations. A critical pair is an equation obtained by *superposition*, the restricted version of paramodulation in which inferences only involve left hand sides of possible rewrite steps, i.e., only the big terms (w.r.t. $\succ$) are considered. During the completion process equations are simplified by rewriting, and tautologies, i.e., equations of the form $s \simeq s$, are removed.

Note that, since the word problem is not decidable in general, a finite confluent E cannot always be obtained. In Knuth and Bendix' original procedure this could be due to *failure*[3] or to non-termination of completion. For completely avoiding failure, *ordered* or *unfailing* completion was introduced [Lan75, HR87, BDP89].

This leads to complete theorem provers for equational theories E, since for every valid equation a rewrite proof will be found after a finite number of steps of the (possibly infinite) completion procedure. Moreover, if the process terminates, it produces a confluent system for ordered rewriting. For improving the efficiency and for reducing the number of cases of non-termination of completion, numerous additional simplification methods and *critical pair criteria* for detecting redundant inferences have been developed [BDH86, Pet90, MN90, Bac91, BD94, CNNR98]. Indeed, nowadays completion has become the method of choice for most state-of-the-art equality reasoning systems.

1.1.2 Ordered paramodulation for general clauses

Extending the notion of critical pair, completion procedures were developed for going beyond unit equations. For instance, for obtaining confluent sets for rewrite relations like *conditional* and *clausal* rewriting, completion procedures were designed for transforming sets of *conditional equations* (definite Horn clauses with equality, i.e., of the form $s_1 \simeq t_1 \wedge \ldots \wedge s_n \simeq t_n \to s \simeq t$) [Kap84, JW86, KR91, Gan91], or *restricted equality* clauses [NO90].

The generalization of this kind of completion procedure to full first-order clauses with equality required the development of more powerful proof techniques for establishing completeness. Using the *transfinite semantic tree* method Hsiang and Rusinowitch [HR91] proved the refutation completeness of *ordered paramodulation*, while Bachmair [Bac89] applied an extension of the so-called *proof ordering* technique for obtaining similar results.

By means of their *model generation* proof method, similar to other *forcing* tech-

[3] Failure could occur because Knuth and Bendix considered rewriting with a terminating set of uni-directional rules, instead of ordered rewriting (applying equations in whatever direction agrees with the given reduction ordering, as explained here). Hence in their view equations had to be *oriented* into terminating rules, which fails if an equation like the commutativity axiom $f(x,y) \simeq f(y,x)$ appears.

niques developed by Zhang [Zha88] and Pais and Peterson [PP91], Bachmair and Ganzinger [BG90, BG94b] proved the completeness of an inference system for full first-order clauses with equality, based on *strict superposition*: paramodulation involving only maximal (w.r.t. the ordering $\succ$) terms of maximal equations of clauses. Such superposition-based inference systems, as well as the model generation method, are explained in detail in Chapter 3 of this thesis.

1.1.3 Selection strategies

A crucial way for reducing the search space in automated deduction are the so-called *selection strategies*. In such strategies the possible inferences between clauses are restricted to the ones involving *selected* literals. This selection can be done in several different ways. The *maximal* (or *ordered*) strategies for a given atom ordering can of course also be seen examples of selection strategies. For example, in a maximal resolution strategy, a (ground) inference between $A \vee C$ and $\neg A \vee D$ is performed only if A is larger in the given atom ordering than all other atoms in C and D. Another well-known selection strategy is the so-called *eager* negative selection strategy, where in each clause a single negative literal is selected whenever there is any. This leads to the so-called *positive* strategies (positive unit strategies in the Horn case) because always the left premise of each (resolution or paramodulation) inference is a positive (unit) clause. These strategies are usually easier to prove complete, but sometimes they are not very efficient, because, roughly speaking, one enumerates all solutions of its conditions before using the positive information of a clause (as discussed in [Der91]). In [BG94b] is is shown that for ordered paramodulation and superposition, it is complete to select in a clause an arbitrary non-empty subset of its negative literals.

1.1.4 Paramodulation with constrained clauses

The advantages of *constrained formulae* are nowadays widely recognized in the context of logic programming. The first ideas for specific applications to paramodulation-based theorem proving were given in [Pet90, KKR90]. The semantics of a clause C with a constraint T, written $C \mid T$, is simply the set of all ground instances $C\sigma$ of C such that σ is a solution of T. For example, if $=$ denotes syntactic equality of terms, the constrained clause $P(x) \mid x = f(y) \ \wedge \ y > a$ denotes[4] all ground atoms $P(f(t))$ such that t is greater than a in the given term ordering $\succ$. Hence if T is unsatisfiable then $C \mid T$ is a tautology.

[4] Note that $>$ and $=$ are used as syntax in the constraint language. Their semantics will be a given term ordering $\succ$ and a given congruence (usually syntactic equality of terms) that depend on the context.

In [KKR90] ordered paramodulation inference rules were expressed for the first time by explicitly formulating the ordering and equality restrictions of the inferences by constraints at the formula level. This gives:

$$\frac{C \vee s \simeq t \mid T \qquad D \mid T'}{C \vee D[t]_p \mid T \wedge T' \ \wedge \ s = D|_p \ \wedge \ OC}$$

where T and T' are the constraints inherited from the premises, the *equality constraint* $s = D|_p$ stores the unification restriction, and OC is an *ordering constraint* of the form $s > t \ \wedge \ldots$ encoding the ordering restrictions imposed by this inference. However, the completeness results of [KKR90] were limited since they required to enumerate the solutions of the constraints and *propagate* (i.e., apply) these solutions to the clause part.

Constraints are closely related to the so-called *basic strategies*, where no inferences need to be computed on subterms generated in unifiers of ancestor inference steps (like its counterpart in E-unification, called *basic narrowing* [Hul80]). It is clear that if such an inference system with inherited constraints is applied without propagation, then it is basic: the inferences only take place on the clause part C of a formula $C \mid T$, and no unifiers are ever applied to C, since the unification restrictions are simply stored in the constraint part T.

Nieuwenhuis and Rubio [NR92a, NR95] showed that, in the context of superposition, indeed propagation of the equality constraints is not needed, thus proving the completeness of *basic superposition*. By using *closure substitutions*, which play the role of equality constraints, the same results were obtained independently by Bachmair and others [BGLS92, BGLS95], giving additional refinements based on *term selection rules* and *redex orderings*. These developments took place independently of much earlier work in Russia by Degtyarev [Deg79], who used *conditional clauses* (which can in fact be seen as clauses with syntactic equality constraints) for describing a form of basic paramodulation without ordering restrictions (see also [DV86]).

In [NR92b] it is shown that by inheriting as well the *ordering constraints* one can restrict the search space even further without losing completeness. In [LS93] equality, disequality and irreducibility constraints are applied for obtaining more powerful redundancy methods in basic equational completion. Finally, in [NR95] the use of constraints in theorem proving procedures is put in a more general framework based on the notion of *constraint inheritance strategies*. The main idea in all these strategies is that the ordering and equality restrictions of the inferences can be kept in constraints and inherited between clauses. If some inference is not compatible with the required restrictions, applied to the current inference rule *and to the previous ones*, then the inference can be blocked. Therefore, for taking advantage of the constraints, algorithms for constraint satisfiability checking are required.

1.1.5 Paramodulation with built-in equational theories

In principle, the aforementioned paramodulation methods apply to any set of clauses with equality, but in some cases special treatments for specific equational subsets of the axioms are preferable. On the one hand, some axioms generate many slightly different permuted versions of clauses, and for efficiency reasons it is many times better to treat all these clauses together as a single one representing the whole class. On the other hand, special treatments can avoid non-termination of completion procedures, like with $f(a,b) \simeq c$ in the presence of associativity and commutativity axioms for f. Also, some equations like the commutativity axiom are more naturally viewed as "structural" axioms (defining a congruence relation on terms) rather than as "simplifiers" (defining a reduction relation). This allows one to extend completion procedures in order to deal with congruence classes of terms instead of single terms, i.e., working with a *built-in* equational theory E, and performing rewriting and completion with special E-matching and E-unification algorithms.

Early results on paramodulation and rewriting *modulo E* were given by Plotkin [Plo72], Slagle [Sla74] and Lankford and Ballantine [LB77] and *extended E-rewriting* was defined by Peterson and Stickel [PS81]. Several E-completion procedures for the equational case were developed e.g. in [LB77, Hue80, PS81, Jou83, JK86, BD89]. Special attention has always been devoted to the case where E includes axioms of associativity and commutativity (AC), which occur very frequently in practical applications, and are well-suited for being built in due to their permutative nature.

The generalization of these E-completion techniques to full first-order clauses with equality has been studied in e.g. [Pau92, Wer92, RV95, BG94a], usually with particular treatments for the AC case. Paramodulation modulo E then becomes roughly the following rule, which has one conclusion for each σ in $U_E(s, D|_p)$, a *minimal complete* set of E-unifiers of $D|_p$ and s:

$$\frac{C \vee s \simeq t \qquad D}{(\, C \vee D[t]_p \,)\sigma} \qquad \text{for all unifiers } \sigma \text{ in } \; U_E(\, s, \; D|_p \,)$$

Note that in general there is no unique most general E-unifier for a given E-unification problem, and that new variables may appear: for example, if f is an AC-symbol, then $f(x,a)$ and $f(y,b)$ have the two AC-unifiers $\sigma_1 = \{x \mapsto b, y \mapsto a\}$ and $\sigma_2 = \{x \mapsto f(b,z), y \mapsto f(a,z)\}$.

Another well-known equational theory that is interesting for being built in is the one of abelian groups (AG). Paramodulation with built-in abelian groups has been investigated by many authors [Che86, Zha93, Mar94, Mar96, GW96, Wal98, Wal99, Stu98]. This is not surprising since abelian groups are of course ubiquitous in many applications of (semi-)automated reasoning. But building in AG is also attractive for at least two more reasons.

On the one hand, due to the fact that diophantine equation solving is easier in the integers than in the natural numbers, AG unification is easier than AC and AC1 unification. If all free symbols are constants, then there is one single most general AG unifier and the decision problem is polynomial, whereas for AC and AC1 the decision problems are NP-complete, and for AC there are exponentially many unifiers. Although with arbitrary free symbols the decision problem is NP-complete in all three cases, AG unification behaves better in practice. Also the number of unifiers is usually much smaller and not doubly exponential as for AC (see [BS93, BS01] for surveys on these results).

Another aspect that makes building-in AG attractive is called *symmetrization* (e.g., by Le Chenadec in [Che86]): modulo abelian groups $(+, -, 0)$, every ground equation can be written as $u + \ldots + u \simeq t$, where u is greater (w.r.t. the given term ordering $\succ$) than the summands in t. As we will see, this allows one to restrict inferences to this maximal summand and to avoid the prolific inferences with extended equations that appear in the AC case.

1.1.6 Basic paramodulation with built-in equational theories

For an equational theory E, the number of E-unifiers of two terms may be large. For instance, the cardinality of a minimal complete set of AC-unifiers is doubly exponential in general [Dom92] (in a sense, this is also an upper bound [KN92]). Hence a single E-paramodulation inference can generate a large number of new clauses.

Therefore, equality constraints become extremely useful in this context. In constrained E-paramodulation, instead of E-unifying the terms, the unification problem is stored in the constraint. Hence in the constrained superposition inference rule the semantics of the symbol '$=$' in the equality constraint $s = D|_p$ becomes E-equality. Dealing with a constrained clause $C \mid s = t$ can be much more efficient than having n clauses $C_1, \ldots, C_n$, one for each E-unifier of s and t, since many inferences are computed at once, and each inference generates one single conclusion. Furthermore, computing E-unifiers is not needed. A clause C with an E-equality constraint T can be proved redundant by means of efficient (sound, but possibly incomplete) methods for detecting unsatisfiable T. If C is the empty clause, a contradiction has been derived if, and only if, the constraint part T is satisfiable, and hence in this case refutation completeness requires a semi-decision procedure for detecting these contradictions. Such a procedure exists for every finite E.

The completeness of such a fully basic strategy for the AC-case (combined with ordering constraints) was first proved in [NR94, NR97], although the first results on (almost basic) constrained deduction methods modulo AC were reported in [Vig94]. The basicness restriction is considered to "have been a key strategy" by McCune [McC97c] in his celebrated AC-paramodulation-based proof of the Robbins problem.

1.1.7 Symbolic constraint solving

Equality constraints are also known as *unification problems*, since they generalize the notion of unification, which usually consists in solving one single equation. Due to the large amount of applications of unification in automated deduction, logic programming and, in general, in symbolic computation, equational constraints have been used in many different applications. Hence for this topic here we refer to [BS01] for a detailed survey.

Concerning ordering constraints, apart from the applications to pruning the search space in automated theorem proving, they are useful in many other contexts like proving termination of term rewrite systems or confluence of ordered term rewrite systems [CNNR98]. Some applications of ordering constraints to ordered strategies in theorem proving 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) [NR92b].

The satisfiability problem for ordering constraints was first shown decidable for fixed signatures when $\succ$ is a total LPO [Com90] or a total RPO [JO91]. For extended signatures, decidability was shown for LPO in [NR95] and for RPO in [Nie93]. Regarding complexity, NP algorithms for LPO (fixed and extended signatures) and RPO (extended ones) were given in [Nie93]. Recently, an NP algorithm has been given as well for RPO under fixed signatures in [NRV99]. For the AC-RPO ordering of [RN95], decidability was shown in [CNR95]. NP-hardness of the satisfiability problem is known, even for one single inequation, for all these cases [CT94]. A new family of algorithms, for full RPO and both semantics, has been introduced recently in [NR99]. These algorithms are based on a new notion of solved form, where properties of orderings like transitivity and monotonicity are taken into account. They are simple and, since guessing is minimised, more efficient. For the Knuth-Bendix Ordering (KBO), frequently used in automated reasoning systems, recently constraint solving algorithms have been given as well [KV00, KV01].

1.2 Overview of this Thesis

In this section the contents of the different chapters of the thesis are outlined.

After this introduction, in Chapter 2 we formally introduce the basic concepts and notations, and in Chapter 3 the model generation technique is introduced.

1.2.1 Paramodulation with non-monotonic orderings

Chapter 4 is the first one introducing new results: it is shown that ordered paramodulation is also complete when orderings with less requirements are used. The main idea is the following.

All the proof techniques for ordered paramodulation that were mentioned in the previous section, the transfinite semantic tree method [HR91], the proof ordering method [BDH86, BD94], and the model generation method [BG94b], require a well-founded, monotonic ordering on ground terms, i.e., a reduction ordering. Moreover, this reduction ordering must be total on the set of ground terms (or extendable to a total one). But in many practical situations these requirements are too strong. For example, for efficiency reasons one may want to use an ordering that cannot be extended to a total one, like $f(a) \succ f(b)$ and $g(b) \succ g(a)$, for which a and b must be uncomparable in any monotonic extension of $\succ$. Another typical situation is deduction modulo built-in equational theories E, where the existence of a total *E-compatible* reduction ordering is a very strong requirement. In fact, for many E such orderings cannot exist. For instance, when E contains an idempotency axiom $f(x,x) = x$, then if $s \succ t$, by monotonicity one should have $f(s,s) \succ f(s,t)$, which by E-compatibility implies $s \succ f(s,t)$ and hence non-well-foudedness.

In Chapter 4 of this thesis we introduce techniques for dropping the monotonicity requirement that open the door to deduction modulo many more classes of equational theories. The only properties required for $\succ$ are well-foundedness and the subterm property. Our results are given there for paramodulation with general first-order clauses with eager selection of negative literals. This solved a well-known open problem, e.g. at the RTA list of open problems [RTA01] since 1995. The only properties required for $\succ$ are well-foundedness and the subterm property. Part of the results given in this chapter have been published at the LICS'99 conference [BGNR99].

1.2.2 Knuth-Bendix completion

In Chapter 5 of this thesis we present a fundamental new result concerning Knuth-Bendix completion: we describe the first practical Knuth-Bendix completion procedure that finds a convergent TRS for a given set of equations E and a (possibly non-totalizable) reduction ordering $\succ$ whenever it exists. This was a well-known open problem (e.g., on the RTA list of open problems since its creation in 1991). Note that being a reduction ordering is the minimal possible requirement on $\succ$, since a TRS terminates if, and only if, it is contained in a reduction ordering. Part of the results given in this chapter have been published as well at the LICS'99 conference [BGNR99].

1.2.3 Completeness of arbitrary selection strategies

For first-order Horn clauses without equality, resolution is complete with an arbitrary selection of one single literal in each clause ([dN96], Theorem 6.7.4). For Horn clauses with built-in equality, i.e., paramodulation-based inference systems, the situation is far more complex. In [Lyn97] some positive and negative results are given for the case where a total *reduction* (well-founded, monotonic) ordering on ground terms is given. Then arbitrary selection strategies are compatible with superposition. Also conditions for eliminating *redundant* clauses are given in [Lyn97], and counter examples indicating the limitations for doing so. For example, in certain circumstances the elimination of tautologies can lead to incompleteness.

In Chapter 6 of this thesis we prove a more general result for Horn clauses with equality: if a paramodulation-based inference system is complete with eager selection of negative equations and, moreover, it is compatible with equality constraint inheritance (like, in particular, it happens for superposition), then it is complete with arbitrary selection strategies.

A first application of this result is the one for paramodulation with non-monotonic orderings of Chapter 4, where the completeness of strategies different from eager negative selection was left open. Here we show that those results are compatible with equality constraint inheritance and hence with the *basic* strategy, thus further restricting the search space. Therefore, our result is applicable, and we obtain the completeness of the same inference system but with arbitrary selection strategies. Part of the contents of this chapter has been published as well at the ICALP'2001 conference [BG01b].

1.2.4 Paramodulation with built-in abelian groups

In Chapter 7 we introduce a new technique for paramodulation with built-in abelian groups (AG). Compared with previous approaches, the technique we introduce in this chapter is simpler, and no inferences with the AG axioms or abstraction rules are needed. This is the first approach where abelian groups are fully built-in in this sense. Furthermore, AG-unification is used instead of the computationally more expensive unification *modulo* associativity and commutativity. Due to the simplicity and restrictiveness of our inference system, its compatibility with redundancy notions and constraints, and the fact that standard term orderings like RPO can be used, we believe that our technique will become the method of choice for practice, as well as a basis for new theoretical developments like logic-based complexity and decidability analysis. For example, we obtain a very simple direct decision procedure for the satisfiability of ground clause sets *modulo* abelian groups. Part of the contents of this chapter has been published as well at the LICS'2000 conference [GN00], whose results have been further developed in the journal version [GN01b].

1.2.5 Ordering constraints for built-in abelian semigroups, monoids and groups

In Chapter 8 we introduce a uniform technique providing the first constraint solving algorithms for a class of ordering constraints used when dealing with built-in abelian semigroups, monoids and groups.

As mentioned in Section 1.1.4 of this introductory chapter, ordered strategies and ordering constraint inheritance can be used without loosing completeness with built-in algebraic theories E, like AC [NR97, Vig94] or AG [GN00]. 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. 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 [McC97c] 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 [RN95] a constraint solving algorithm exists [CNR95]. 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 necessary requirements, 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 $x + 0 \rightarrow x$.

In Chapter 8 of this thesis we give NP algorithms for these RPO-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, and give new insights for the development of constraint solving methods over fixed signatures for other E-compatible orderings. Part of the contents of this chapter has been published as well at the LICS'2001 conference [GN01a].

1.2.6 Directions for further research

This research is far from closed. In Chapter 9 of this thesis we outline a number of interesting open questions, as well as ideas for the implementation of some of the results that have been obtained. Most of these directions for further work could lead to a PhD. thesis on themselves, and are hence outside the scope of this thesis. For example, for some possible extensions of the results of Chapters 4 and 5, counter examples to their completeness exist, but for others, concerning weaker orderings or more powerful redundancy notions, the compleness remains open.

Chapter 2

Basic Concepts and Notation

Here we introduce the main basic tools used: terms, rewriting, term orderings, first-order equality clauses and equality Herbrand interpretations. Most (if not all) of our definitions are the ones of [NR01].

2.1 Terms and (rewrite) relations

Let $\mathcal{F}$ be a *signature*, a (finite) set of function symbols with an arity function *arity*: $\mathcal{F} \to I\!N$ and let $\mathcal{X}$ be a set of variable symbols. Function symbols f with $arity(f) = n$ are called *n-ary* symbols (when $n = 1$, one says *unary* and when $n = 2$, *binary*). If $arity(f) = 0$, then f is a *constant symbol*. The set of *first-order terms* over $\mathcal{F}$ and $\mathcal{X}$, denoted by $\mathcal{T}(\mathcal{F}, \mathcal{X})$, is the smallest set containing $\mathcal{X}$ such that $f(t_1, \ldots, t_n)$ is in $\mathcal{T}(\mathcal{F}, \mathcal{X})$ whenever $f \in \mathcal{F}$, $arity(f) = n$, and $t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. Similarly, $\mathcal{T}(\mathcal{F})$ is the set of variable free or *ground* terms. Note that $\mathcal{T}(\mathcal{F}) = \emptyset$ if there are no constant symbols in $\mathcal{F}$. As usual, along this thesis it is therefore assumed that there is at least one constant symbol in $\mathcal{F}$.

A *position* is a sequence of positive integers. If p is a position and t is a term, then by $t|_p$ we denote the *subterm of t at position p*: we have $t|_\lambda = t$ (where λ denotes the empty sequence) and $f(t_1, \ldots, t_n)|_{i.p} = t_i|_p$ if $1 \leq i \leq n$ (and is undefined if $i > n$). We also write $t[s]_p$ to denote the term obtained by replacing in t the subterm at position p by the term s. For example, if t is $f(a, g(b, h(c)), d)$, then $t|_{2.2.1} = c$, and $t[d]_{2.2} = f(a, g(b, d), d)$. We say that a a variable (or function symbol) x *occurs* (at position p) in a term t if $t|_p$ is (rooted by) x. By $vars(t)$ we denote the set of all variables occurring in t. If t is a term of the form $f(t_1, \ldots, t_n)$, then we define $top(t)$ to be the function symbol f. The syntactic equality of two terms s and t will be denoted by $s \equiv t$.

A *substitution* σ is a mapping from variables to terms. It can be extended to a function from terms to terms in the usual way: using a postfix notation, $t\sigma$

denotes the result of simultaneously replacing in t every $x \in Dom(\sigma)$ by $x\sigma$. Here substitutions are sometimes written as sets of pairs $x \mapsto t$, where x is a variable and t is a term. For example, if σ is $\{x \mapsto f(b,y), y \mapsto a\}$, then $g(x,y)\sigma$ is $g(f(b,y),a)$ (this example illustrates the *simultaneous* replacement: applying σ "from left to right" yields $g(f(b,a),a)$, which is not the intended meaning).

A substitution σ is *ground* if its range is $\mathcal{T}(\mathcal{F})$. Unless stated otherwise, we will assume that ground substitutions σ applied to a term t are also *grounding*, that is, $vars(t) \subseteq Dom(\sigma)$, and hence $t\sigma$ is ground. A term t *matches* a term s if $s\sigma \equiv t$ for some σ. Then t is called an *instance* of s.

A term t is *unifiable* with a term s if $s\sigma \equiv t\sigma$ for some substitution σ. Then σ is called a *unifier* of s and t. Furthermore, a substitution σ is called a *most general unifier* of s and t, denoted $mgu(s,t)$, if $s\sigma \equiv t\sigma$, and for every other unifier θ of s and t, it holds that $s\theta \equiv s\sigma\sigma' \equiv t\theta \equiv t\sigma\sigma'$ for some σ', that is, roughly, if every other unifier θ is a particular instance of σ. We sometimes speak about *the mgu* of s and t because it is unique up to variable renaming (see [BS01] for details and for unification algorithms computing mgu's).

A *multiset* over a set S is a function $M: S \to I\!N$. The union and intersection of multisets are defined as usual by $M_1 \cup M_2(x) = M_1(x) + M_2(x)$, and $M_1 \cap M_2(x) = min(M_1(x), M_2(x))$. We also use a set-like notation: $M = \{a,a,b\}$ denotes $M(a) = 2$, $M(b) = 1$, and $M(x) = 0$ for $x \not\equiv a$ and $x \not\equiv b$. A multiset M is *empty* if $M(x) = 0$ for all $x \in S$.

If $\to$ is a binary relation, then $\leftarrow$ is its inverse, $\leftrightarrow$ is its symmetric closure, $\to^+$ is its transitive closure and $\to^*$ is its reflexive-transitive closure. We write $s \to^! t$ if $s \to^* t$ and there is no t' such that $t \to t'$. Then t is called *irreducible* and a *normal form* of s (w.r.t. $\to$). The relation $\to$ is *well-founded* or *terminating* if there exists no infinite sequence $s_1 \to s_2 \to \ldots$ and it is *confluent* or *Church-Rosser* if the relation $\leftarrow^* \circ \to^*$ is contained in $\to^* \circ \leftarrow^*$. It is *locally confluent* if $\leftarrow \circ \to \ \subseteq\ \to^* \circ \leftarrow^*$. By Newman's lemma, terminating locally-confluent relations are confluent. A relation $\to$ on terms is *monotonic* if $s \to t$ implies $u[s]_p \to u[t]_p$ for all terms s, t and u and positions p. A *congruence* is a reflexive, symmetric, transitive and monotonic relation on terms.

An *equation* is a multiset $\{s,t\}$, denoted $s \simeq t$ or, equivalently, $t \simeq s$. A *rewrite rule* is an ordered pair (s,t), written $s \to t$, and a set of rewrite rules R is a *rewrite system* (sometimes we will also write such rules as $s \Rightarrow t$ to avoid confusion with the arrow $\to$ of clauses in sequent notation). The rewrite relation with R on $\mathcal{T}(\mathcal{F},\mathcal{X})$, denoted $\to_R$, is the smallest monotonic relation such that $l\sigma \to_R r\sigma$ for all $l \to r \in R$ and all σ. If $s \to_R t$ then we say that s *rewrites into* t with R. We say that R is terminating, confluent, etc. if $\to_R$ is. A rewrite system R is called *convergent* if it is confluent and terminating. It is not difficult to see that then every term t has a unique normal form w.r.t. $\to_R$, denoted by $nf_R(t)$, and $s = t$ is a logical consequence of R (where R is seen as a set of equations) if and only if $nf_R(s) = nf_R(t)$. Sometimes the congruence relations (on $\mathcal{T}(\mathcal{F})$) $\leftrightarrow^*_R$ (or $\leftrightarrow^*_E$) are denoted by R^* (E^*) or $=_R$ ($=_E$).

2.2 Term orderings

A (strict partial) ordering $\succ$ is a transitive and irreflexive binary relation. An ordering $\succ$ on terms is *stable* (or *closed*) under substitutions if $s \succ t$ implies $s\sigma \succ t\sigma$ for all s, t and σ; it fulfills the *subterm property* if $u[s]_p \succ s$ for all s, u and $p \neq \lambda$. It is *total* on $\mathcal{T}(\mathcal{F})$ if for all s and t in $\mathcal{T}(\mathcal{F})$, either $s = t$ or $s \succ t$ or $t \succ s$; if $=$ is a congruence different from syntactic equality, we speak about totality *up to* $=$.

A *rewrite ordering* is a monotonic ordering stable under substitutions; a *reduction ordering* is a well-founded rewrite ordering, and a *simplification ordering* is a rewrite ordering with the subterm property.

The following properties are not difficult to check: a reduction ordering total on $\mathcal{T}(\mathcal{F})$ is necessarily a simplification ordering on $\mathcal{T}(\mathcal{F})$; by Kruskal's theorem, simplification orderings are well-founded (for finite, fixed-arity signatures); and a rewrite system R is terminating if and only if all its rules are contained in a reduction ordering $\succ$, i.e., $l \succ r$ for every $l \to r \in R$ (in fact, then $\to^+_R$ is itself a reduction ordering).

Let $\succ$ be an ordering on terms and let $=$ be a congruence relation. Then $\succ$ is called *compatible* with $=$ if $s' = s \succ t = t'$ implies $s' \succ t'$ for all s,s',t and t'. If E is a set of equations, then $\succ$ is called *E-compatible* if it is compatible with $=_E$. Note that if $\succ$ is E-compatible, $s =_E t$ implies $s \not\succ t$ and $t \not\succ s$.

Let $\succ$ be an ordering on terms and let $=$ be a congruence relation such that $\succ$ is compatible with $=$. Then these relations induce relations on tuples and multisets of terms as follows.

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 $1 \ldots n$. It is well-known that, if $\succ$ is well founded, so is $\succ^{lex}$.

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 $\succ\!\!\succ$ (or $\succ_{mul}$) on multisets of terms such that

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

Sometimes the notation $\succ\!\!\succ$ is used without explicitly indicating which is the congruence $=$. In these cases $=$ is assumed to be the syntactic equality relation $\equiv$ on terms. If $\succ$ is well founded on S, so is $\succ\!\!\succ$ on finite multisets over S [DM79].

A way to define suitable orderings for practical purposes (like termination proving or automated deduction) is to construct them directly from a well-founded *precedence*, an ordering $\succ_{\mathcal{F}}$ on $\mathcal{F}$. This is done in the so-called *path orderings*, like the *lexicographic path ordering* (LPO) or the *recursive path ordering (with status)* (RPO) [KL80, Der82].

Let $\succ_{\mathcal{F}}$ be a precedence and let $\mathcal{F}$ be the disjoint union of two sets lex and mul, the symbols with lexicographic and multiset *status*, respectively. By $=_{mul}$ we denote the equality of ground terms up to the permutation of direct arguments of symbols with multiset status: $f(s_1,\ldots,s_m) =_{mul} g(t_1,\ldots,t_n)$ if $f = g$ and hence $m = n$, and $s_{\pi(i)} =_{mul} t_i$ for $1 \leq i \leq n$ and where π is a permutation of $1\ldots n$ which is the identity if $f \in lex$.

In this setting, RPO is defined as follows: $s \succ_{rpo} x$ if x is a variable that is a proper subterm of s or else $s \equiv f(s_1 \ldots s_n) \succ_{rpo} t \equiv g(t_1 \ldots t_m)$ if at least one of the following conditions holds:

- $s_i \succ_{rpo} t$ or $s_i =_{mul} t$, for some $i \in \{1\ldots n\}$
- $f \succ_{\mathcal{F}} g$, and $s \succ_{rpo} t_j$, for all j in $\{1\ldots m\}$
- $f \equiv g$ (and hence n=m) and $f \in mul$ and $\{s_1,\ldots,s_n\} \succ\!\!\succ_{rpo} \{t_1,\ldots,t_n\}$
- $f \equiv g$ (and hence n=m) and $f \in lex$, $\langle s_1,\ldots,s_n\rangle \succ_{rpo}^{lex} \langle t_1,\ldots,t_n\rangle$, and $s \succ_{rpo} t_j$, for all j in $\{1\ldots n\}$

where $\succ_{rpo}^{lex}$ and $\succ\!\!\succ_{rpo}$ are, respectively, the lexicographic and multiset extensions of $\succ_{rpo}$ with respect to $=_{mul}$.

The *lexicographic path ordering* (LPO) is defined as the particular case of an RPO where $\mathcal{F} = lex$, i.e., where all symbols have a lexicographic status.

It is known that RPO is a reduction ordering on $\mathcal{T}(\mathcal{F},\mathcal{X})$, which is moreover total on $\mathcal{T}(\mathcal{F})$ up to $=_{mul}$ (and hence in case of LPO, total up to $\equiv$) if $\succ_{\mathcal{F}}$ is total on $\mathcal{F}$ [KL80, Der82].

LPO's are useful for extending reduction orderings $\succ$ that are total up to a congruence $=$ (like RPO is total up to $=_{mul}$), to reduction orderings total up to $\equiv$. This extension is obtained by a lexicographic combination $\succ_t$ whose first component is $\succ$, and whose second component is a total LPO $\succ_{lpo}$, that is, $s \succ_t t$ if either $s \succ t$ or $s = t$ and $s \succ_{lpo} t$.

It is not difficult to see that RPO is C-compatible (C for commutativity) if commutative symbols have multiset status, but it is not AC-compatible.

2.3 Equality clauses and Herbrand interpretations

A clause is a pair of finite multisets of equations Γ (the *antecedent*) and Δ (the *succedent*), denoted by $\Gamma \rightarrow \Delta$. We sometimes use a comma in clauses to denote the union of multisets or the inclusion of equations in multisets; for example, we write $s \simeq t, \Gamma, \Gamma' \rightarrow \Delta$ instead of $\{s \simeq t\} \cup \Gamma \cup \Gamma' \rightarrow \Delta$. Clauses $e_1, \ldots, e_n \rightarrow e'_1, \ldots, e'_m$ are sometimes (equivalently) written as a disjunction of equations and negated equations $\neg e_1 \vee \ldots \vee \neg e_n \vee e'_1 \vee \ldots \vee e'_m$. Hence, the e_i are called the *negative* equations, and the e'_j the *positive* equations, respectively, of the clause.

A clause $\Gamma \rightarrow \Delta$ is called a *Horn clause* if Δ contains at most one equation. The *empty clause* $\square$ is a clause $\Gamma \rightarrow \Delta$ where both Γ and Δ are empty. A *positive* (resp. *negative*) clause is a clause $\Gamma \rightarrow \Delta$ where Γ (resp. Δ) is empty, and a *unit* clause is a clause with exactly one literal.

We will use all aforementioned notions and notations defined for terms t, like $t|_p$, $t[s]_p$, $vars(t)$, $t\sigma$, etc., as well for equations and clauses in the expected way. For example, a term u *occurs* in a clause $\Gamma \rightarrow \Delta$ if $t \simeq s \in \Gamma \cup \Delta$ and $t|_p \simeq u$ for some position p.

Let R be a set of ground equations (or rewrite rules). Then the congruence $\leftrightarrow^*_R$ defines an equality Herbrand *interpretation* I: the domain of I is $\mathcal{T}(\mathcal{F})$, each n-ary function symbol f of $\mathcal{F}$ is interpreted as the function f_I where $f_I(t_1, \ldots, t_n)$ is the term $f(t_1, \ldots, t_n)$, and where the only predicate $\simeq$ is interpreted by $s \simeq t$ if $s \leftrightarrow^*_R t$. The interpretation I defined by R in this way will be denoted by R^*. We write $s = t \in I$ if $s \leftrightarrow^*_R t$. I *satisfies* (is a model of) a ground clause $\Gamma \rightarrow \Delta$, denoted $I \models \Gamma \rightarrow \Delta$, if $I \not\supseteq \Gamma$ or $I \cap \Delta \neq \emptyset$. The empty clause $\square$ is hence satisfied by no interpretation. I satisfies a non-ground clause C if I satisfies all ground instances of C. I satisfies a set of clauses S, denoted by $I \models S$, if it satisfies every clause in S. A clause C is a logical consequence of (or C *follows from*) a set of clauses S, denoted by $S \models C$, if C is satisfied by every model of S.

2.4 Constraints and constrained clauses

An *(ordering and equality) constraint* is a quantifier-free first-order formula built over the binary predicate symbols $>$ and $=$ relating terms in $\mathcal{T}(\mathcal{F}, \mathcal{X})$. Regarding semantics, the constraints are interpreted in $\mathcal{T}(\mathcal{F})$, and $=$ is interpreted as some congruence $=_c$ on $\mathcal{T}(\mathcal{F})$ (like syntactic equality or AC-equality) and $>$ is interpreted as a given reduction ordering $\succ$ on ground terms that is total up to $=_c$. Hence a *solution* of a constraint T is a ground substitution σ with domain $vars(T)$ and such that $T\sigma$ evaluates to true for the given $=_c$ and $\succ$. If a solution for T exists, then T is called *satisfiable*. If every ground substitution with domain $vars(T)$ is a solution of T then T is a *tautology*.

A *constrained clause* is a pair $C \mid T$ where C is a clause and T is a constraint. A *ground instance* of $C \mid T$ is a ground clause $C\sigma$ where σ is a solution of T. The semantics of $C \mid T$ is the set of all its ground instances. Hence, by definition, an interpretation I satisfies $C \mid T$ if $I \models C\sigma$ for every ground instance $C\sigma$ of $C \mid T$. Therefore, clauses with unsatisfiable constraints are tautologies. A clause $C \mid T$ is the *constrained empty clause*, denoted as well by $\square$, if C is empty and T is satisfiable. Constrained clauses $C \mid T$ where T is a tautology are sometimes denoted by C, omitting the constraint part T.

2.5 Inference Systems

A logical *inference* is a step by which from a multiset of zero or more constrained clauses (the *premises*) a new constrained clause (the *conclusion*) is obtained. An *inference rule* $\mathcal{R}$

$$\frac{C_1 \mid T_1 \ \ldots \ C_n \mid T_n}{D \mid T} \qquad \text{if} \quad \textit{condition}$$

is (a finite representation of) the set of inferences where from the multiset of clauses of the form $\{C_1 \mid T_1 \ \ldots \ C_n \mid T_n\}$ one can infer $D \mid T$ if *condition* holds. One such an inference is called an *inference by* $\mathcal{R}$. An *inference system* $\mathcal{I}$ is a set of inference rules. An *inference by* $\mathcal{I}$ is an inference by one of the rules of $\mathcal{I}$. We will frequently consider inference rules where premises or conclusions have constraints that are tautologies and hence these constraints are omitted.

An inference rule $\mathcal{R}$ is *correct* if, for all inferences by $\mathcal{R}$, the conclusion is a logical consequence of the premises, and an inference system is correct if all its rules are correct. A set of constrained clauses S is *closed* under $\mathcal{I}$ if for every inference by $\mathcal{I}$ with premises in S, the corresponding conclusion is in S. $\mathcal{I}$ is *refutation complete* if $\square \in S$ for every unsatisfiable set of constrained clauses S closed under $\mathcal{I}$. All inference systems in the remainder of this thesis are easily proved correct, and we will focus on completeness.

Chapter 3

The Model Generation Technique

In this chapter, we introduce the *model generation* method, Bachmair and Ganzinger's standard technique for establishing the completeness of ordered paramodulation calculi [BG94b]. We dedicate an independent chapter to this technique because it will be used several times in this thesis. This introduction of the model generation technique has been adapted from [NR01].

3.1 A simple inference system for ground Horn clauses

Here start with a simple example on ground Horn clauses. Note that if $C \mid T$ is a constrained clause where C is ground and T is satisfiable, then it is equivalent to $C \mid \top$ where $\top$ denotes a tautological constraint. Hence while dealing with ground clauses constraints will be omitted.

In the following, let $\succ_{\mathcal{G}}$ be a given total reduction ordering on $\mathcal{T}(\mathcal{F})$, and let $s \succeq t$ denote $s \succ t \vee s \equiv t$. The inference system $\mathcal{G}$ for ground Horn clauses with equality is the following:

superposition right:

$$\frac{\Gamma' \to l \simeq r \qquad \Gamma \to s \simeq t}{\Gamma', \Gamma \to s[r]_p \simeq t} \quad \text{if} \quad \begin{array}{l} s|_p \equiv l,\ l \succ_{\mathcal{G}} r,\ s \succ_{\mathcal{G}} t, \text{ and} \\ l \succ_{\mathcal{G}} u \text{ for all } u \text{ occurring in } \Gamma', \text{ and} \\ s \succ_{\mathcal{G}} v \text{ for all } v \text{ occurring in } \Gamma \end{array}$$

superposition left:

$$\frac{\Gamma' \to l \simeq r \qquad \Gamma, s \simeq t \to \Delta}{\Gamma', \Gamma, s[r]_p \simeq t \to \Delta} \quad \text{if} \quad \begin{array}{l} s|_p \equiv l,\ l \succ_{\mathcal{G}} r,\ s \succ_{\mathcal{G}} t, \text{ and} \\ l \succ_{\mathcal{G}} u \text{ for all } u \text{ occurring in } \Gamma', \text{ and} \\ s \succeq_{\mathcal{G}} v \text{ for all } v \text{ occurring in } \Gamma, \Delta \end{array}$$

equality resolution:

$$\frac{\Gamma, s \simeq s \to \Delta}{\Gamma \to \Delta} \quad \text{if} \quad s \succeq_{\mathcal{G}} v \text{ for all } v \text{ occurring in } \Gamma, \Delta$$

Let us remark that the equality resolution rule is named after the fact that it encodes a resolution inference with the reflexivity axiom of equality $x \simeq x$,

It is sometimes said that in the superposition rules the inferences take place *with* the term l *on* the term s, and that the inference *involves* s and l. Note that in $\mathcal{G}$, superposition right inferences involve only terms s and l that are *strictly maximal* in their respective premises, that is, they are bigger w.r.t. $\succ_{\mathcal{G}}$ than all other occurrences of terms in these premises. Superposition left takes place also with strictly maximal terms, but on (possibly non-strictly) *maximal* terms (that is, they are larger than or equal to all terms in their premise).

3.2 Extending the ordering to literals and clauses

In order to prove the refutation completeness of $\mathcal{G}$ we first define the following total ordering $\succ_c$ on ground clauses. If C is a clause

$$s_1 = s'_1, \ldots, s_n = s'_n \to t_1 = t'_1, \ldots, t_m = t'_m$$

then we define $ms(C)$ as the multiset:

$$\{\{s_1, s_1, s'_1, s'_1\}, \ldots, \{s_n, s_n, s'_n, s'_n\}, \{t_1, t'_1\}, \ldots, \{t_m, t'_m\}\}$$

Finally, let $\succ_c$ be the ordering on clauses defined by comparing these expressions by the two-fold multiset extension of $\succ$, that is, $C \succ_c D$ if $ms(C)(\succ_{mul})_{mul}ms(D)$. The result is a total ordering on ground clauses[1].

[1] Roughly, $\succ_c$ compares the multisets of all equations occurring in the clauses, but where in addition terms occurring negatively have slightly more weight than the ones occurring positively; in fact, in order to make $\succ_c$ total on ground clauses, the information of which equations are positive and which ones are negative has to be present anyway.

3.3 Generating the interpretation

Now we come to the key to the model generation method. Our aim is to prove the completeness of $\mathcal{G}$. We do this by showing that, if S is a set of ground Horn clauses closed under $\mathcal{G}$ and $\Box \notin S$, then S is satisfiable. The satisfiability proof of S is of a constructive nature: first, an equality Herbrand interpretation will be built, and second, it will be shown that this interpretation is a model of S.

We now informally explain the first part. The interpretation we build will be the congruence R^* induced by a set of ground rewrite rules R, where each rule in R has been *generated* by some clause of S (hence the name "model generation"). The generation process of R is defined by induction on $\succ_c$. Each clause C in S generates a rule or not, depending on the set R_C of rules generated by clauses D of S with $C \succ_c D$ (and on the congruence R^*_C induced by R_C). These ideas are formalised as follows:

Definition 2 (Model generation) Let C be a clause in S. Then $Gen(C) = \{l \Rightarrow r\}$, and C is said to *generate* the rule $l \Rightarrow r$, if, and only if, C is of the form $\Gamma \rightarrow l \simeq r$ and the three following conditions hold:

1. $R^*_C \not\models C$,
2. $l \succ_{\mathcal{G}} r$ and $l \succ_{\mathcal{G}} u$ for all u occurring in Γ
3. l is irreducible by R_C

where $R_C = \bigcup_{C \succ_c D} Gen(D)$. In all other cases $Gen(C) = \emptyset$. Finally, R denotes the set of all rules generated by clauses of S, that is, $R = \bigcup_{D \in S} Gen(D)$.

Let us analyse the three conditions. The first one states that a clause only contributes to the model if it does not hold in the partial model built so far and hence we are *forced* to extend this partial model. The second one states that a clause can only generate a rule $l \Rightarrow r$ if l is the strictly maximal term of the clause. The third condition, stating that l is irreducible by the rules generated so far, is, together with the second one, the key for showing that R is convergent, from which the completeness result quite easily follows:

Lemma 3 For every set of ground clauses S, the set of rules R generated for S is convergent (i.e., confluent and terminating). Furthermore, if $R^*_C \models C$ then $R^* \models C$ for all ground C.

Proof: Evidently, R is terminating since $l \succ_{\mathcal{G}} r$ for all its rules $l \Rightarrow r$. To prove confluence, it suffices to show *local* confluence, which in the ground case is well-known (and easily shown) to hold if there are no two different rules $l \Rightarrow r$ and

$l' \Rightarrow r'$ where l' is a subterm of l. This property is fulfilled: clearly when a clause C in S generates $l \Rightarrow r$, no such $l' \Rightarrow r'$ is in R_C; but if $l' \Rightarrow r'$ is generated by a clause D with $D \succ_c C$ then, by definition of $\succ_c$, we must have $l' \succ_{\mathcal{G}} l$ and hence l' cannot be a subterm of l either.

To show $R_C^* \models C$ implies $R^* \models C$, let C be $\Gamma \rightarrow \Delta$, and assume $R_C^* \models C$. If $R_C^* \models \Delta$ then $R^* \models \Delta$ since $R \supseteq R_C$. Otherwise, $R_C^* \not\models \Gamma$. Then $R^* \not\models \Gamma$ follows from the fact a term t occurring negatively in a clause is bigger than the same t occurring positively: all rules in $R \setminus R_C$ are generated by clauses bigger than C, and hence have left hand sides that are too big to reduce any term occurring in Γ. Since R is convergent this implies $R^* \not\models \Gamma$. □

3.4 Completeness for the ground case

Theorem 4 The inference system $\mathcal{G}$ is refutation complete for ground Horn clauses.

Proof: Let S be a set of ground Horn clauses that is closed under $\mathcal{G}$ and such that $\Box \notin S$. We prove that then S is satisfiable by showing that R^* is a model for S. We proceed by induction on $\succ_c$, that is, we derive a contradiction from the existence of a minimal (w.r.t. $\succ_c$) clause C in S such that $R^* \not\models C$. There are a number of cases to be considered, depending on the occurrences in C of its maximal term s, i.e., the term s such that $s \succeq_{\mathcal{G}} u$ for all terms u in C (s is unique since $\succ_{\mathcal{G}}$ is total on $\mathcal{T}(\mathcal{F})$):

1. s occurs only in the succedent and C is $\Gamma \rightarrow s \simeq s$. This is not possible since $R^* \not\models C$.

2. s occurs only in the succedent and C is $\Gamma \rightarrow s \simeq t$ with $s \not\equiv t$. Since $R^* \not\models C$, we have $R^* \supseteq \Gamma$ and $s \simeq t \notin R^*$, i.e., C has not generated the rule $s \Rightarrow t$. This must be because s is reducible by some rule $l \Rightarrow r \in R_C$. Assume $l \Rightarrow r$ has been generated by a clause C' of the form $\Gamma' \rightarrow l \simeq r$. Then there exists an inference by superposition right:

$$\frac{\Gamma' \rightarrow l \simeq r \qquad \Gamma \rightarrow s \simeq t}{\Gamma', \Gamma \rightarrow s[r]_p \simeq t}$$

   whose conclusion D has only terms u with $s \succ_{\mathcal{G}} u$, and hence $C \succ_c D$. Moreover, D is in S and $R^* \not\models D$, since $R^* \supseteq \Gamma \cup \Gamma'$ and $s[r]_p \simeq t \notin R^*$ (since otherwise $s[l]_p \simeq t \in R^*$). This contradicts the minimality of C.

3. s occurs in the antecedent and C is $\Gamma, s \simeq s \rightarrow \Delta$. Then there exists an inference by equality resolution:

$$\frac{\Gamma, s \simeq s \rightarrow \Delta}{\Gamma \rightarrow \Delta}$$

for whose conclusion D it holds that $C \succ_c D$. Moreover, D is in S and $R^* \not\models D$, which is a contradiction as in the previous case.

4. s occurs in the antecedent and C is $\Gamma, s \simeq t \rightarrow \Delta$ with $s \succ_g t$. Since $R^* \not\models C$, we have $s \simeq t \in R^*$ and since R is convergent, s and t must have the same normal forms w.r.t. R, so s must be reducible by some rule $l \Rightarrow r \in R$. Assume $l \Rightarrow r$ has been generated by a clause C' of the form $\Gamma' \rightarrow l \simeq r$. Then there exists an inference by superposition left:

$$\frac{\Gamma' \rightarrow l \simeq r \qquad \Gamma, s[l]_p \simeq t \rightarrow \Delta}{\Gamma', \Gamma, s[r]_p \simeq t \rightarrow \Delta}$$

for whose conclusion D it holds that $C \succ_c D$. Moreover, D is in S and $R^* \not\models D$, which again contradicts the minimality of C. □

The following example shows how the rewrite system R changes during a closure of a set of ground clauses and that, although for the intermediate sets the obtained R^* is not a model, the R^* obtained for the closed set is a model.

Example 5 Consider the lexicographic path ordering generated by the precedence $f \succ_{\mathcal{F}} a \succ_{\mathcal{F}} b \succ_{\mathcal{F}} c \succ_{\mathcal{F}} d$. The following table shows in the left column the ground Horn clauses (sorted with respect to the ordering) at each closure step, in which the first one is the initial set, and in the right column the set R corresponding to each intermediate set. The maximal term of every clause is underlined and the subterms of the clauses involved in the inference are framed.

S	R
$\to \boxed{\underline{\mathrm{c}}} \simeq d$ $\underline{f(d)} \simeq d \to a \simeq b$ $\to \underline{f(\boxed{\mathrm{c}})} \simeq d$	$c \Rightarrow d$
$\to \underline{c} \simeq d$ $\to \boxed{\underline{\mathrm{f(d)}}} \simeq d$ $\boxed{\underline{\mathrm{f(d)}}} \simeq d \to a \simeq b$ $\to \underline{f(c)} \simeq d$	$c \Rightarrow d$ $f(d) \Rightarrow d$
$\to \underline{c} \simeq d$ $d \simeq d \to \underline{a} \simeq b$ $\to \underline{f(d)} \simeq d$ $\underline{f(d)} \simeq d \to a \simeq b$ $\to \underline{f(c)} \simeq d$	$c \Rightarrow d$ $a \Rightarrow b$ $f(d) \Rightarrow d$

Let us conclude this section with a remark on additional ordering restrictions. In superposition left as well as in equality resolution, it is possible to strengthen the conditions in such a way that only one negative literal becomes eligible for inferences. For example, in superposition left on an equation $s \simeq t$, one can require that $t \succ t'$ for all equations $s \simeq t'$ in Γ, that is, we use the maximal equation rather than just the maximal term; if two equations have the same maximal terms, we compare the other terms. Similarly, in equality resolution we can require $s \succ t'$ for all equations $s \simeq t'$ in Γ. In the inference system for general clauses (see Section 3.8) we have included these restrictions, since such comparisons between equations are needed there anyway. We did not consider them for $\mathcal{G}$ for simplicity reasons, and also because by means of *selection of negative equations* we will be able to obtain stronger results in a simpler way (see Section 3.9).

3.5 Non-equality predicates

In this framework, equality can be considered to be the only predicate, since for every other predicate symbol p, (positive or negative) atoms $p(t_1 \ldots t_n)$ can be expressed as (positive or negative) equations $p(t_1 \ldots t_n) \simeq$ *true*, where *true* is a new special symbol, and where p is considered as a function symbol rather than as a predicate symbol. Note however that, in order to avoid meaningless expressions in which predicate symbols occur at proper subterms one should adopt a two-sorted type discipline on terms in the encoding.

It is easy to see that this transformation preserves satisfiability. Very roughly: one can "translate" the interpretations such that a ground atom is true in a Herbrand interpretation I if and only if in the equality Herbrand interpretation I' over the modified signature the term $p(t_1 \ldots t_n)$ is congruent to *true*. Be we remark that two ground atoms that are false in I need not be in the same congruence class of I'.

After this satisfiability preserving transformation, ordered resolution (ground) inferences of the form:

$$\frac{\Gamma' \rightarrow A \qquad \Gamma, A \rightarrow \Delta}{\Gamma', \Gamma \rightarrow \Delta} \quad \text{if} \quad A \succ_{\mathcal{G}} \Gamma' \text{ and } A \succeq_{\mathcal{G}} \Gamma, \Delta.$$

become a special case of superposition left:

$$\frac{\Gamma' \rightarrow A \simeq \mathit{true} \qquad \Gamma, A \simeq \mathit{true} \rightarrow \Delta}{\Gamma', \Gamma, \mathit{true} \simeq \mathit{true} \rightarrow \Delta}$$

combined with equality resolution (or simplification, as we will see) for eventually eliminating the trivial equation *true* $\simeq$ *true*.

For efficiency reasons it is convenient to make *true* small in the ordering. Sometimes it is also useful to take into account that p is a predicate symbol when handling the ordering restrictions. For example, in orderings like RPO, if the predicate symbols p are bigger in the precedence than function symbols then $p \succ_{\mathcal{F}} q$ implies $p(t_1, \ldots, t_n)\sigma \succ_{lpo} q(s_1, \ldots, s_m)\sigma$ for all ground σ.

3.6 Clauses with variables

Up to now, in this chapter we have only dealt with ground clauses. If we consider that a non-ground clause represents the set of all its ground instances[2], a refutation complete method for the non-ground case would be to systematically enumerate all ground instances of the clauses, and to perform inferences by $\mathcal{G}$ between those instances. But fortunately it is possible to perform inferences between non-ground clauses, covering in one step a possibly large number of ground inferences. We now adapt $\mathcal{G}$ according to this view.

For example, at the ground level, in the superposition right inference

$$\frac{\Gamma' \rightarrow l \simeq r \qquad \Gamma \rightarrow s \simeq t}{\Gamma', \Gamma \rightarrow s[r]_p \simeq t} \quad \text{if} \quad \begin{array}{l} s|_p \equiv l,\ l \succ_{\mathcal{G}} r,\ s \succ_{\mathcal{G}} t, \text{ and} \\ l \succ_{\mathcal{G}} u \text{ for all } u \text{ occurring in } \Gamma', \text{ and} \\ s \succ_{\mathcal{G}} v \text{ for all } v \text{ occurring in } \Gamma \end{array}$$

[2] By Herbrand's theorem, considering only the ground instances preserves satisfiability; in fact, this is a consequence of (the proof of) Theorem 11.

we required $s|_p$ and l to be the same term. At the non-ground level, this becomes a constraint $s|_p = l$ on the possible instances of the conclusion, that is, the conclusion is a constrained clause $D \mid T$. Hence if the conclusion is $D \;\mid s|_p = l \wedge \ldots$, the instances $D\sigma$ for which $s|_p\sigma \not\equiv l\sigma$ are not created. The same is done for the ordering restrictions. For instance, instead of requiring $l \succ_g r$ as a condition of the inference, it becomes part of the constraint of the conclusion, excluding those instances $D\sigma$ of the conclusion that correspond to ground inferences between instances of the premises for which $l\sigma \succ_g r\sigma$ does not hold:

$$\frac{\Gamma' \rightarrow l \simeq r \qquad \Gamma \rightarrow s \simeq t}{\Gamma', \Gamma \rightarrow s[r]_p \simeq t \;\mid s|_p{=}l \;\wedge\; l{>}r \;\wedge\; s{>}t \;\wedge \ldots}$$

Note that here we have written the inference rule without constraints in its premises, since at this point we are only interested in the constraints that are generated in this concrete inference.

This inference rule can be further restricted with the additional condition stating that the inference is not necessary if $s|_p$ is a variable. This shows that, by working on the non-ground level, certain inferences between ground instances of the premises turn out to be redundant: at the non-ground level we do not perform, for an instance with σ, the inferences *inside* σ (also called inferences *below variables*), that is, on positions $s\sigma|_p$ where $s|_{p'}$ is a variable for some prefix p' of p.

Note that, as usual, it may be necessary to rename variables in the premises in order to avoid name clashes: the premises C and D are assumed to fulfill $vars(C) \cap vars(D) = \emptyset$.

Now we define the inference system $\mathcal{H}$ for non-ground Horn clauses, writing $s{>}\Gamma$ as a shorthand for the constraint $s{>}u_1 \;\wedge\; s{>}v_1 \;\wedge \ldots \wedge\; s{>}u_n \;\wedge\; s{>}v_n$ if Γ is a multiset of equations $\{u_1 \simeq v_1, \ldots, u_n \simeq v_n\}$ (and similarly, we write $s \geq \Gamma$ for $s{\geq}u_1 \;\wedge\; s{\geq}v_1 \;\wedge \ldots \wedge\; s{\geq}u_n \;\wedge\; s{\geq}v_n$):

superposition right:

$$\frac{\Gamma' \rightarrow l \simeq r \qquad \Gamma \rightarrow s \simeq t}{\Gamma', \Gamma \rightarrow s[r]_p \simeq t \;\mid s|_p{=}l \;\wedge\; l{>}r \;\wedge\; l{>}\Gamma' \;\wedge\; s{>}t \;\wedge\; s{>}\Gamma}$$

superposition left:

$$\frac{\Gamma' \rightarrow l \simeq r \qquad \Gamma, s \simeq t \rightarrow \Delta}{\Gamma', \Gamma, s[r]_p \simeq t \rightarrow \Delta \;\mid s|_p{=}l \;\wedge\; l{>}r \;\wedge\; l{>}\Gamma' \;\wedge\; s{>}t \;\wedge\; s{\geq}\Gamma, \Delta}$$

equality resolution:

$$\frac{\Gamma, s \simeq t \rightarrow \Delta}{\Gamma \rightarrow \Delta \;\mid s{=}t \;\wedge\; s \geq \Gamma, \Delta}$$

where in both superposition rules $s|_p$ is required not to be a variable.

Example 6 Consider the lexicographic path ordering generated by the precedence $h \succ_{\mathcal{F}} a \succ_{\mathcal{F}} f \succ_{\mathcal{F}} g \succ_{\mathcal{F}} b$. In the following inference

$$\frac{g(x) \simeq x \rightarrow f(a,x) \simeq f(x,x) \qquad\qquad \rightarrow h(f(a,g(y))) \simeq h(y)}{\begin{array}{ll} g(x) \simeq x \rightarrow h(f(x,x)) \simeq h(y) & \mid\ f(a,x)=f(a,g(y)) \ \wedge\ h(f(a,g(y)))>h(y) \wedge \\ & \ \ f(a,x)>f(x,x) \ \wedge\ f(a,x)>g(x) \ \wedge\ f(a,x)>x \end{array}}$$

the constraint of the conclusion is satisfiable: using the properties of the ordering and solving the unification problem, the constraint can be simplified into

$$x=g(y) \ \wedge \ a>x$$

which has, for instance, the solution $\{y \mapsto b, x \mapsto g(b)\}$.

On the other hand, the following inference is not needed

$$\frac{\rightarrow f(x,x) \simeq f(a,x) \qquad\qquad \rightarrow f(g(y),z) \simeq h(z)}{\begin{array}{ll} \rightarrow f(a,x) \simeq h(z) & \mid\ f(x,x)=f(g(y),z) \ \wedge\ f(g(y),z)>h(z) \wedge \\ & \ \ f(x,x)>f(a,x) \end{array}}$$

since the constraint of the conclusion has no solution; it can be simplified to

$$x=g(y) \ \wedge \ x=z \ \wedge \ y \geq h(z) \ \wedge \ x>a$$

which implies $y \geq h(g(y))$. Note that the equality constraint and the ordering constraint considered separately are both satisfiable but their conjunction is not. □

Let us also remark that, at the non-ground level, several terms in a premise C may be involved in paramodulation inferences; for a term t it may be the case that for some ground instances $C\sigma$ the term $t\sigma$ is the maximal one, and for other instances it is not.

3.7 Completeness without constraint inheritance

There are several possible treatments for the constrained clauses generated by the inference system $\mathcal{H}$. The classical view is to deal only with unconstrained clauses. Conclusions of the form $C \mid s = t \wedge OC$, for some ordering constraint OC, are then immediately converted into $C\sigma$ where $\sigma = mgu(s,t)$. This strategy will be called here *$\mathcal{H}$ without constraint inheritance*, in contrast with other possibilities which will be introduced later on.

Of course, the clause $C\sigma$ has to be generated only if the constraint $s = t \wedge OC$ is satisfiable in $\mathcal{T}(\mathcal{F})$, where $=$ is interpreted as the syntactic equality relation $\equiv$, and $>$ as the given reduction ordering $\succ$. If $\succ$ is the lexicographic path ordering (LPO) the satisfiability of such constraints is decidable [Com90, Nie93]. But traditionally in the literature weaker approximations by non-global tests are used; for example, inference systems are sometimes expressed with local conditions of the form $r \not\succeq l$ when in our framework we have $l > r$ as a part of the global constraint OC. Note that such weaker approximations do not lead to unsoundness, but only to the generation of unnecessary (for completeness) clauses.

In the following, we call a set of (unconstrained) Horn clauses S *closed under $\mathcal{H}$ without constraint inheritance* if $D\sigma \in S$ for all inferences by $\mathcal{H}$ with premises in S and conclusion $D \mid s = t \wedge OC$ such that $s = t \wedge OC$ is satisfiable and $\sigma = mgu(s,t)$.

Theorem 7 The inference system $\mathcal{H}$ is refutation complete without constraint inheritance for Horn clauses.

Proof: Let S be a set of Horn clauses closed under $\mathcal{H}$ without constraint inheritance such that $\square \notin S$. The proof is very similar to the one for $\mathcal{G}$: we exhibit a model R^* for S. We proceed again by induction on $\succ_c$, but now the role of the ground clauses in the proof for $\mathcal{G}$ is played by all *ground instances* of clauses in S, and the generation of rules in R from these ground instances is the same as for $\mathcal{G}$. Now we derive a contradiction from the existence of a minimal (w.r.t. $\succ_c$) *ground instance $C\sigma$ of a* clause C in S such that $R^* \not\models C\sigma$. The cases considered are the same ones as well, again depending on the occurrences in $C\sigma$ of its maximal term $s\sigma$.

The only difference lies in the *lifting* argument, which is the same in all cases and is hence analyzed here for only one of them: C is $\Gamma, s \simeq t \rightarrow \Delta$ and $s\sigma \succ_{\mathcal{G}} t\sigma$. Since $R^* \not\models C\sigma$, we have $s\sigma \simeq t\sigma \in R^*$ and since R is convergent, $s\sigma$ must be reducible by some rule $l\sigma \Rightarrow r\sigma \in R$, generated by a clause C' of the form $\Gamma' \rightarrow l \simeq r$. (Note that, since we assume that there are no name clashes between the variables of C and C', we can consider that the instances of C and of C' under consideration are both by the same ground σ.) Now we have $s\sigma|_p = l\sigma$, and there are two possibilities:

An inference. $s|_p$ is a non-variable position of s.
Then there exists an inference by superposition left:

$$\frac{\Gamma' \rightarrow l \simeq r \qquad \Gamma, s \simeq t \rightarrow \Delta}{\Gamma', \Gamma, s[r]_p \simeq t \rightarrow \Delta \;\; \mid s|_p{=}l \,\wedge\, l{>}r \,\wedge\, l{>}\Gamma' \,\wedge\, s{>}t \,\wedge\, s{\geq}\Gamma,\Delta}$$

whose conclusion $D \mid T$ has an instance $D\sigma$ (i.e., σ is a solution of T) such that $C\sigma \succ_c D\sigma$, where $R^* \not\models D\sigma$, contradicting the minimality of $C\sigma$.

Lifting. $s|_{p'}$ is a variable x for some prefix p' of p.
Then $p = p' \cdot p''$ for some p'', and $x\sigma|_{p''}$ is $l\sigma$. Now let σ' be the ground substitution with the same domain as σ but where $x\sigma' = x\sigma[r\sigma]_{p''}$ and $y\sigma' = y\sigma$ for all other variables y. Then $R^* \not\models C\sigma'$ and $C\sigma \succ_c C\sigma'$, contradicting the minimality of $C\sigma$.
□

3.8 General clauses

In this section general clauses are considered, i.e., clauses that may have several equations in their succedents. For this purpose, the inference system $\mathcal{H}$ is adapted. In order to restrict the amount of inferences to be performed, it is desirable to preserve the property of $\mathcal{H}$ that for each ground clause (or instance) C, only one literal of C is involved in superposition inferences with C. Since now the maximal term of C may occur in more than one equation in the succedent, it is decided that among these equations the one whose other side is maximal will be used. This leads to the notion of maximal and strictly maximal equations in C. In order to express maximality and strict maximality of equations as constraints, we use the following notation. The constraint $gr(s \simeq t, \Delta)$ expresses that the equation $s \simeq t$, i.e., the multiset $\{s,t\}$, is strictly greater, w.r.t. the multiset extension of $\succ$, than all equations $u \simeq v$ in Δ. For each $u \simeq v$ this condition $s \simeq t \succ\!\!\succ u \simeq v$ can be expressed for instance by the constraint:

$$\begin{array}{l} s>u \wedge (s>v \vee t\geq v) \;\vee\; s>v \wedge (s>u \vee t\geq u) \;\vee \\ t>u \wedge (s\geq v \vee t>v) \;\vee\; t>v \wedge (s\geq u \vee t>u) \end{array}$$

Similarly, the constraint $greq(s \simeq t, \Delta)$ expresses that $s \simeq t \succeq\!\!\succeq u \simeq v$ for all $u \simeq v$ in Δ. The full inference system $\mathcal{I}$ for general clauses is

superposition right:

$$\frac{\Gamma' \to l \simeq r, \Delta' \qquad\qquad \Gamma \to s \simeq t, \Delta}{\Gamma', \Gamma \to s[r]_p \simeq t, \Delta', \Delta \quad \Big| \quad \begin{array}[t]{l} s|_p = l \;\wedge \\ l>r \;\wedge\; l>\Gamma' \;\wedge\; gr(l \simeq r, \Delta') \;\wedge \\ s>t \;\wedge\; s>\Gamma \;\wedge\; gr(s \simeq t, \Delta) \end{array}}$$

superposition left:

$$\frac{\Gamma' \to l \simeq r, \Delta' \qquad\qquad \Gamma, s \simeq t \to \Delta}{\Gamma', \Gamma, s[r]_p \simeq t \to \Delta', \Delta \quad \Big| \quad \begin{array}[t]{l} s|_p = l \;\wedge \\ l>r \;\wedge\; l>\Gamma' \;\wedge\; gr(l \simeq r, \Delta') \;\wedge \\ s>t \;\wedge\; greq(s \simeq t, \Gamma \cup \Delta) \end{array}}$$

equality resolution:

$$\frac{\Gamma, s \simeq t \to \Delta}{\Gamma \to \Delta \;\mid\; s=t \;\wedge\; greq(s \simeq t, \Gamma \cup \Delta)}$$

equality factoring:

$$\frac{\Gamma \to s \simeq t, s' \simeq t', \Delta}{\Gamma, t \simeq t' \to s \simeq t', \Delta \;\mid s=s' \wedge s>t \wedge s>\Gamma \;\wedge\; greq(s \simeq t, \Delta \cup \{s' \simeq t'\})}$$

where as in the Horn case in both superposition rules $s|_p$ is not a variable.

Here the superposition rules and the equality resolution rule play the same role as their counterparts in the inference system $\mathcal{H}$. The equality factoring rule is new. Intuitively, it expresses that, if s and s' are syntactically equal, and t and t' are semantically equal, then the two equations in the succedent express the same information, and one of them can be omitted.

Example 8 Consider the lexicographic path ordering generated by the precedence $f \succ_{\mathcal{F}} g \succ_{\mathcal{F}} h$ and the following inference by superposition right

$$\frac{\rightarrow g(z) \simeq h(z) \qquad\qquad \rightarrow f(g(x),y) \simeq g(x), f(g(x),y) \simeq y}{\begin{array}{ll} \rightarrow f(h(z),y) \simeq g(x), f(g(x),y) \simeq y & \mid\ g(x) = g(z)\ \wedge\ g(z) > h(z)\ \wedge \\ & \ \ f(g(x),y) > g(x)\ \wedge \\ & \ \ gr(f(g(x),y) \simeq g(x), \{f(g(x),y) \simeq y\}) \end{array}}$$

where $gr(f(g(x),y) \simeq g(x), \{f(g(x),y) \simeq y\})$ can be simplified into $g(x) > y$. Now, simplifying the rest of the constraint, the conclusion of the inference can be written as

$$\rightarrow f(h(z),y) \simeq g(x), f(g(x),y) \simeq y \ \mid\ x = z\ \wedge\ g(x) > y$$

Below an overview of the new aspects for the completeness proof of $\mathcal{I}$ with respect to $\mathcal{H}$ is given. For simplicity, only the ground case is considered; lifting to clauses with variables is analogous to what was done for $\mathcal{H}$. First, a new condition is added in the generation of the rewrite system R for a set of clauses S, and the second condition is adapted in order to select the strictly maximal positive equation that produces the rule:

Definition 9 Let S be a set of ground clauses and let C be a clause in S. Then $Gen(C) = \{l \Rightarrow r\}$, and C is said to *generate* the rule $l \Rightarrow r$, if, and only if, C is of the form $\Gamma \rightarrow l \simeq r, \Delta$ and

1. $R_C^* \not\models C$
2. $l \succ_{\mathcal{G}} r$, $l \succ_{\mathcal{G}} \Gamma$, and $l \simeq r \succ\!\!\succ u \simeq v$ for all $u \simeq v$ in Δ
3. l is irreducible by R_C
4. $R_C^* \not\models r \simeq t'$ for every $l \simeq t' \in \Delta$

where $R_C = \bigcup_{C \succ_c D} Gen(D)$. In all other cases $Gen(C) = \emptyset$. Finally, R denotes the set of all rules generated by clauses of S, that is, $R = \bigcup_{D \in S} Gen(D)$.

The proof of Lemma 3 can be easily adapted to show that here again R is convergent and that if $R_C^* \models C$ then $R^* \models C$. In a very similar way, it can be shown that the new conditions force clauses generating rules to have only one positive literal satisfied by the interpretation:

Lemma 10 If a clause C of the form $\Gamma \rightarrow l \simeq r, \Delta$ generates the rule $l \Rightarrow r$ then $R^* \models \Gamma$ and $R^* \not\models \Delta$.

Theorem 11 The inference system $\mathcal{I}$ is refutation complete for general clauses.

Proof: Since lifting is done as for $\mathcal{H}$, here we only extend the proof for the ground case $\mathcal{G}$. There is one additional case due to the new conditions for generating rules in R. The other cases of the proof for $\mathcal{G}$ are straightforwardly adapted by using lemma 10 to show that the conclusion of the required inference is not satisfied by the model.

The new case is: C is of the form $\Gamma \rightarrow s \simeq t, \Delta$, with $s \succ t, \Gamma$ and $s \simeq t$ is maximal in Δ, and it has not generated a rule because there is an equation $s \simeq t'$ in Δ such that $R^*_C \models t \simeq t'$ (note that this case includes also the case in which $s \simeq t$ is maximal in Δ, but not strictly maximal).

Then, with $\Delta = s \simeq t', \Delta'$, there exists an inference by equality factoring

$$\frac{\Gamma \rightarrow s \simeq t, s \simeq t', \Delta'}{\Gamma, t \simeq t' \rightarrow s \simeq t', \Delta'}$$

whose conclusion D is such that $C \succ_c D$ and $R^* \not\models D$, contradicting the minimality of C. □

3.9 Selection of negative equations

The inference system $\mathcal{I}$ includes strong ordering restrictions: roughly, a superposition inference is needed only if the terms involved are maximal sides of maximal equations in their respective premises, and even strictly maximal in case they occur in positive equations. But more constraints can be imposed. If a clause C has a non-empty antecedent, it is possible to arbitrarily *select* exactly one of its negative equations. Then completeness is preserved even if C is not used as left premise of any superposition inference and the only inferences involving C are equality resolution or superposition left on its selected equation.

The inference system $\mathcal{S}$ (for selection) for general clauses is defined to consist of the four rules of inference system I where for all premises of the inference rules no negative equation has been selected, plus the following two additional rules, where the selected equations have been underlined:

superposition left on a selected equation:

$$\frac{\Gamma' \rightarrow l \simeq r, \Delta' \qquad\qquad \Gamma, \underline{s \simeq t} \rightarrow \Delta}{\Gamma', \Gamma, s[r]_p \simeq t \rightarrow \Delta', \Delta \quad \Big| \quad \begin{array}{l} s|_p = l \;\wedge \\ l > r \;\wedge\; l > \Gamma' \;\wedge\; gr(l \simeq r, \Delta') \;\wedge\; s > t \end{array}}$$

equality resolution on a selected equation:

$$\frac{\Gamma, \underline{s \simeq t} \rightarrow \Delta}{\Gamma \rightarrow \Delta \;|\; s = t}$$

where, as usual in superposition rules, $s|_p$ is not a variable.

Note that an adequate selection strategy gives us a strictly more restrictive inference system: among the set of maximal negative equations, just select one of them, and select no equation if this set is empty. It is clear that in the inference system $\mathcal{I}$ *all* maximal equations of the antecedent are eligible for superposition left or equality resolution, whereas in $\mathcal{S}$ only the selected one is eligible.

The intuition behind selection is, roughly, that a clause with negative equations does not need to contribute to the deduction process until its whole antecedent has been proved from other clauses, and in particular one can require the selected equation to be proved first.

In practice one can select for example always a maximal equation (under some arbitrary ordering) of the antecedent. Selecting always a negative equation, whenever there is one, leads in the Horn case to the so-called *positive unit literal strategies*, that is, the left premise of superposition inferences is always a positive unit clause [Der91, NN91]. For general clauses eager selection leads to *positive* strategies, where the left premise is always a positive clause, i.e., it has only positive literals. Adapting the proof of completeness of Theorem 11 to this framework with selection is an easy exercise: it suffices to consider that clauses with selected equations generate no rules.

3.10 Completeness with constraint inheritance

Here we consider sets of *constrained* clauses, rather than unconstrained ones, as in the previous sections. For simplicity, we deal with the Horn case only.

Definition 12 A set of constrained Horn clauses S is *closed under $\mathcal{H}$ with constraint inheritance* if $D \mid T_1 \wedge \ldots \wedge T_n \wedge s = t \wedge OC$ is in S whenever $C_1 \mid T_1, \ldots, C_n \mid T_n$ are clauses in S and there is an inference by $\mathcal{H}$ with premises $C_1, \ldots, C_n$ and conclusion $D \mid s = t \wedge OC$ such that the constraint $T_1 \wedge \ldots \wedge T_n \wedge s = t \wedge OC$ is satisfiable.

This strategy is incomplete in general: the closure under $\mathcal{H}$ with constraint inheritance of an unsatisfiable set of constrained Horn clauses needs not contain the empty clause.

Example 13 Let $\succ$ be the lexicographic path ordering where $a \succ_{\mathcal{F}} b$. Consider the following unsatisfiable clause set S:

$$\begin{array}{lrl} 1. & & \to a \simeq b \\ 2. & & \to P(x) \mid x = a \\ 3. & P(b) & \to \end{array}$$

S is clearly closed under $\mathcal{H}$ with equality constraint inheritance, since no inferences by $\mathcal{H}$ that are compatible with the constraint of the second clause can be made. We have $a \succ_{\mathcal{F}} b$ and hence the first clause could only be used by superposing a on some non-variable subterm, while superposition left (i.e., resolution) between 2 and 3 leads to a clause with an unsatisfiable constraint $x = a \wedge b = x$. However, S does not contain the empty clause. This incompleteness is due to the fact that the usual lifting arguments, like the ones in Theorem 7, do not work here, since they are based on the existence of *all* ground instances of the clauses. Note that this is not the case here: the only instance of the second clause is $P(a)$, whereas the lifting argument in Theorem 7 requires the existence of the instance $P(b)$. □

Fortunately, the strategy is complete for what we will call *well-constrained* sets of clauses, which turn out to be adequate for many practical situations. A key idea in this context is the following (quite intuitive) notion of *irreducible ground substitution.* Let R be a ground rewrite system contained in the given ordering $\succ$ (that is, $l \succ r$ for all rules $l \Rightarrow r$ of R). A ground substitution σ is *reducible* by R if $x\sigma$ is reducible by R for some $x \in Dom(\sigma)$; if there is no such x then σ is *irreducible.* Furthermore, if S is a set of constrained clauses, then $irred_R(S)$ is its set of *irreducible instances*, that is, the set of ground instances $C\sigma$ of clauses $C \mid T$ in S such that σ is a solution of T and $x\sigma$ is irreducible by R for all $x \in vars(C)$.

Definition 14 A set of constrained clauses S is *well-constrained* if either there are no clauses with equality predicates in S or else for all R contained in $\succ$ we have $irred_R(S) \cup R \models S$.

Example 15 (Example 13 continued) The clause set S of the previous example is not well-constrained: if R is $\{a \Rightarrow b\}$ then the instance $P(a)$ of the second clause is not a logical consequence of $irred_R(S) \cup R$ (in fact, the second clause has no irreducible instances for this R). □

Let us give some more intuition behind the definition of well-constrained sets. For clauses without equality predicates, the situation is clear: all such sets are well-constrained (this is why logic programming without equality is compatible with arbitrary constraint systems).

Now let us consider clause sets S including equality predicates. First, note that if S is a well-constrained set, so is its closure w.r.t. any sound inference system, since the property of well-constrainedness is preserved under the addition of logical consequences. Second, it is not difficult to see that if all clauses in S have only tautologic constraints then S is well-constrained: every instance $C\sigma$ is either in $irred_R(S)$, or else σ is reducible by R. Then σ can be reduced into a "normal form" σ', where $C\sigma'$ is in $irred_R(S)$, and we have $irred_R(S) \cup R \models C\sigma$.

But there are other non-trivial examples of well-constrained sets.

Example 16 Let $\succ$ be the lexicographic path ordering where $g \succ_{\mathcal{F}} a \succ_{\mathcal{F}} f \succ_{\mathcal{F}} b$. Then, constrained clauses like $g(x,x) \simeq b \mid a > x$ may appear in well-constrained sets, since the variable x is not "lower bounded": as for unconstrained clauses, for all σ the term $x\sigma$ can be reduced into a "normal form" $x\sigma'$, where $g(x\sigma', x\sigma') \simeq b$ is in $irred_R(S)$, and hence we have $irred_R(S) \cup R \models g(x\sigma, x\sigma)$. Here $g(x,x) \simeq b \mid a > x$ denotes the infinite set of clauses of the form $g(f^n(b), f^n(b)) \simeq b$ for $n \geq 0$, that is, $g(b,b) \simeq b$, $g(f(b),f(b)) \simeq b$, $g(f(f(b)),f(f(b))) \simeq b\ldots$ Note that such (in this case even non-regular) tree languages cannot be captured by standard first-order clauses. □

Furthermore, it will become clear from the completeness proof below that the notion of well-constrained clause could be modified in order to capture more cases by not considering *all* R contained in $\succ$, but only those R whose rules could be generated in the model generation technique applied to the given clause set. Then, one can know in advance that certain (e.g., constructor) terms will be irreducible w.r.t. such R. Here we have not done this in order to keep the definition of well-constrainedness simple.

The refutation completeness of $\mathcal{H}$ for well-constrained clause sets S can now be established by applying a simple variant of the model generation technique. Before we give the formal proof, let us explain the main ideas. Let S be a set of well-constrained clauses that is closed under $\mathcal{H}$ with equality constraint inheritance, and assume $\Box \notin S$. As always, we show that then S is satisfiable by generating a rewrite system R for S (in a similar way as before) and then proving that $R^* \models S$.

For this purpose, we first show that $R^* \models irred_R(S)$ like in Theorem 7, but where the lifting case never needs to be applied (since we only consider the set of irreducible instances of S). Once we have $R^* \models irred_R(S)$, then also $R^* \models S$, since of course $R^* \models R$ and by well-constrainedness of S (where well-constrainedness is required only with respect to the particular R that has been generated) we have

$irred_R(S) \cup R \models S$ (note that if there are no equality literals in S then $irred_R(S)$ coincides with S).

Theorem 17 The inference system $\mathcal{H}$ is refutation complete with constraint inheritance for well-constrained sets S of Horn clauses.

Proof: Let S be closed under $\mathcal{H}$ with equality constraint inheritance. Again we build a model R^* for S whenever $\Box \notin S$. As said, we prove that $R^* \models irred_R(S)$, which implies $R^* \models S$ by well-constrainedness.

We build R as for Theorem 7, but now only the irreducible (w.r.t. R_C) instances of S contribute to its construction: a ground instance C of the form $\Gamma \rightarrow l \simeq r$ in $irred_{R_C}(S)$ generates the rule $l \Rightarrow r$ of R if the usual conditions (i), (ii) and (iii) apply.

Now again we derive a contradiction from the existence of a minimal (w.r.t. $\succ_c$) ground instance $C\sigma \in irred_R(S)$ for some $C \mid T \ \in S$, where σ is a solution of T, such that $R^* \not\models C\sigma$. Again we consider several cases, depending on the occurrences in $C\sigma$ of its maximal term $s\sigma$. Let us analyse only the case where C is $\Gamma, s \simeq t \rightarrow \Delta$ and $s\sigma \succ_g t\sigma$. Since $R^* \not\models C\sigma$, we have $R^* \models s\sigma \simeq t\sigma$, and hence the term $s\sigma$ is reducible by some rule $l\sigma \Rightarrow r\sigma \in R$, generated by an instance $C'\sigma$ of some $C' \mid T'$, where C' is of the form $\Gamma' \rightarrow l \simeq r$.

Now we have $s\sigma|_p = l\sigma$, and, since σ is irreducible by R, the only possibility is now that $s|_p$ is a non-variable position of s. Then there exists an inference by superposition left:

$$\frac{\Gamma' \rightarrow l \simeq r \qquad \Gamma, s \simeq t \rightarrow \Delta}{\Gamma', \Gamma, s[r]_p \simeq t \rightarrow \Delta \ \mid s|_p{=}l \ \wedge \ l{>}r \ \wedge \ l{>}\Gamma' \ \wedge \ s{>}t \ \wedge \ s{\geq}\Gamma, \Delta}$$

whose conclusion has an instance $D\sigma$ where σ is a solution of $T \ \wedge \ T' \ \wedge \ s|_p = l \ \wedge \ l > r \ \wedge \ l > \Gamma' \ \wedge \ s > t \ \wedge \ s \geq \Gamma, \Delta$ such that $C\sigma \succ_c D\sigma$ and where $R^* \not\models D\sigma$. Furthermore, $D\sigma \in irred_R(S)$: indeed $x\sigma$ is irreducible by R for all variables $x \in vars(D)$. This is clearly the case if $x \in vars(C)$. For $x \in C'$, there are two cases: if $x \equiv l$ then $x \notin vars(D)$ since $l\sigma \succ r\sigma, \Gamma'\sigma$; if $x \not\equiv l$ then $x\sigma$ is irreducible w.r.t. $R_{C'}$ by construction of R, and hence also w.r.t. R, since for all rules $l' \Rightarrow r' \in R \setminus R_{C'}$ we have $l' \succeq_g l\sigma \succ_g x\sigma$ and hence such rules cannot reduce $x\sigma$. Altogether, this contradicts the minimality of $C\sigma$. □

Chapter 4

Paramodulation with Non-Monotonic Orderings

Up to now, all existing completeness results for ordered paramodulation require the term ordering $\succ$ to be well-founded, monotonic and total(izable) on ground terms. For several applications, these requirements are too strong, and hence weakening them has been a well-known research challenge.

In this chapter we introduce a new completeness proof technique for ordered paramodulation where the only properties required on $\succ$ are well-foundedness and the subterm property. The technique is a relatively simple and elegant application of some fundamental results on the termination and confluence of ground term rewrite systems (TRS).

4.1 Introduction

All main techniques for proving the completeness of paramodulation-based inference systems, like the transfinite semantic tree method [HR91], the proof ordering method of [BDH86, BD94] and the model generation method [BG94b] explained in Chapter 3, rely at some point on the requirement that the term ordering $\succ$ is well-founded, monotonic and total (or extendable to a total ordering) on ground terms.

But, as said, in many practical situations these requirements are too strong. A typical situation is deduction modulo built-in equational theories E, where the existence of a total *E-compatible* reduction ordering is a very strong requirement. For example, the existence of such an ordering for the case where E consists of associativity and commutativity (AC) properties for some symbols remained open for a long time, and, once it was found, it triggered quite a number of results, like the decidability of the ground AC-word and -unification problems. Unfortunately, for many E such orderings cannot exist. For instance, when E contains an idempotency axiom $f(x,x) = x$, then if $s \succ t$, by monotonicity one should have $f(s,s) \succ f(s,t)$, which by E-compatibility implies $s \succ f(s,t)$ and hence non-well-foundedness.

In this chapter we introduce techniques for dropping the monotonicity requirement that, among other applications, open the door to deduction modulo many more classes of equational theories. The only properties required for $\succ$ are well-foundedness and the subterm property. This solves a well-known open problem (e.g. at the RTA list of open problems [RTA01] since 1995).

Our technique (given in Sections 4.2 to 4.7) is a variant of the model generation technique, with the main difference that the termination of the ground rewrite system R that defines the model is not a consequence of the ordering. Instead, termination of R follows from other properties. In one of the settings treated here, it follows from the irreducibility, w.r.t. R itself, of the right hand sides of all rules of R; in another setting, if R is contained in a well-founded ordering $\succ$ with the subterm property, its termination follows from the irreducibility of the right hand sides at non-topmost positions only. Since each terminating TRS R induces a reduction (i.e., well-founded, monotonic) ordering $\rightarrow^+_R$, we can then use induction on (an extension of) this reduction ordering for proving the main completeness results. These results are given here for paramodulation with general first-order clauses with eager *selection* of negative literals (see Chapter 3). In Chapter 6 we analyze strategies with selection of positive literals. In Section 4.8 we shortly mention the applicability of techniques for *redundancy elimination* and *constraint inheritance* in the context of these techniques (in Chapter 6 constraint inheritance in this setting is considered in more detail). Finally, in Section 4.9 we give some counterexamples indicating the limitations for some of the extensions.

4.2 Some properties of ground TRS and orderings

Let us recall from Chapter 2 that a (strict partial) *ordering* on $T(\mathcal{F},\mathcal{X})$ is an irreflexive transitive relation $\succ$. It is a *reduction* ordering if it is well-founded and monotonic, and moreover, it is *stable under substitutions:* $s \succ t$ implies $s\sigma \succ t\sigma$ for all substitutions σ. It fulfils the *subterm property* if $\succ \supseteq \rhd$, where $\rhd$ denotes the strict subterm ordering.

Definition 18 A *west ordering* is a well-founded ordering on $T(\mathcal{F})$ that fulfils the subterm property and that is total on $T(\mathcal{F})$ (it is called *west* after well-founded, subterm and total).

Not all well-founded orderings on terms can be extended to west orderings, even if they do not contradict the subterm property. For example, if $a \succ_1 f(b)$ and

$b \succ_1 f(a)$, then, if $\succ$ is $(\succ_1 \cup \rhd)^+$, we get $a \succ f(b) \succ b \succ f(a) \succ a$. But every well-founded ordering can be totalized [Wec91], and hence every well-founded ordering satisfying the subterm property can be extended to a west ordering. We also have the following:

Lemma 19 Every reduction ordering $\succ_r$ can be extended to a west ordering.

Proof: Let $\succ_{rs}$ be $(\succ_r \cup \rhd)^+$. Then $\succ_{rs}$ is well-founded. We will derive a contradiction from the existence of an infinite sequence $s_1 \succ_{rs} s_2 \succ_{rs} \ldots$ with s_1 minimal w.r.t. $\succ_r$. By monotonicity of $\succ_r$ we have that $s[t] \rhd t \succ_r u$ implies $s[t] \succ_r s[u] \rhd u$, i.e. the relations commute in this direction. Now since $\rhd$ is well-founded there should be some s_k in the sequence which is the first one such that $s_k \succ_r s_{k+1}$, and hence by applying the commutation property we can re-arrange the sequence obtaining an infinite sequence $s_1 \succ_r s'_2 \succ_{rs} \ldots s'_k \succ_{rs} s_{k+1} \succ_{rs} \ldots$ for some $s'_2, \ldots, s'_k$, which contradicts the minimality of s_1. □

Lemma 20 Let R be a ground TRS such that for all rules $l \rightarrow r$ in R the term r is irreducible by R. Then R is terminating.

Proof: Assume R is non-terminating. Then there exists an infinite rewrite sequence $t_1 \rightarrow_R t_2 \rightarrow_R \ldots$ It is easy to extract an infinite subsequence $s_1 \rightarrow_R s_2 \rightarrow_R \ldots$ of it where there is at least one rewrite step $s_i \rightarrow_R s_{i+1}$ at the topmost position, i.e., where $s_i \equiv l$ and $s_{i+1} \equiv r$ for some rule $l \rightarrow r$ in R. But then s_{i+1} is irreducible by R, contradicting the infiniteness assumption. □

Lemma 21 Let $\succ$ be a west ordering, and let R be a ground TRS such that, for all $l \rightarrow r$ in R, $l \succ r$ and r is irreducible by R at non-topmost positions. Then R is terminating.

Proof: If R is non-terminating there is an infinite sequence $s_1 \rightarrow_R s_2 \rightarrow_R \ldots$ with at least one rewrite step $s_i \rightarrow_R s_{i+1}$ at the topmost position, i.e., where $s_i \equiv l$ and $s_{i+1} \equiv r$ for some rule $l \rightarrow r$ in R. But then $s_{i+1} \rightarrow_R s_{i+2} \rightarrow_R \ldots$ is an infinite sequence with steps only at topmost positions, where $s_{i+1} \succ s_{i+2} \succ \ldots$ contradicting the well-foundedness of $\succ$. □

Finally we will also use the following well-known results on orderings:

Lemma 22 ([DJ90]) If R is a terminating TRS then $\rightarrow_R^+$ is a reduction ordering.

4.3 West orderings in practice

In practical applications (theorem provers, implementations of Knuth-Bendix completion) the west ordering $\succ$ can be defined and dealt with in different ways.

One possibility is that an approximation of $\succ$ is available by a non-total ordering $\succ_v$ on terms with variables such that $s \succ_v t$ implies $s\sigma \succ t\sigma$ for all ground σ (e.g., when $\succ_v$ is a reduction ordering of which $\succ$ is an extension, as in Lemma 19). Then, inferences with ordering restrictions like $l\sigma\theta \succ r\sigma\theta$ for some ground substitution θ can be proved redundant by showing that $r\sigma \succeq_v l\sigma$. Indeed, the actual west ordering need not always be really built for application purposes. It suffices for completeness that it exists, and for practice that a reasonably good approximation is available.

We now mention two general-purpose techniques for defining $\succ$ such that in practice it can be used or approximated efficiently.

4.3.1 Semantic path orderings

The *recursive path ordering* with status (RPO, [Der82]) (which includes the *lexicographic path ordering*, LPO), is a well-known, easy to implement, general-purpose ordering for deduction purposes. It is a reduction ordering that is total on ground terms. It compares the head symbols of the terms (with a *precedence* ordering on the function symbols), and then recursively it applies a (sometimes lexicographic or multiset) comparison on the arguments. Since RPO is monotonic and includes the subterm relation, it cannot prove termination of rules like $f(f(x)) \rightarrow f(g(f(x)))$, because the term $g(f(x))$ is larger than its subterm $f(x)$, and hence, by monotonicity, $f(g(f(x)))$ will always be larger than $f(f(x))$.

The *semantic path ordering* (SPO, [KL80]) is a well-known powerful generalization of the RPO, where the precedence on function symbols is replaced by any (well-founded) underlying (quasi-)ordering involving the whole term rather than only its head symbol. This makes the ordering much more powerful. In fact, for every terminating TRS R there is some SPO that includes $\rightarrow_R^+$. SPO includes the subterm relation, but it is not monotonic in general. In fact, it can handle rules like $f(f(x)) \rightarrow f(g(f(x)))$. The price to be paid is that for proving termination of TRS by an SPO, one needs to prove in an ad-hoc way its monotonicity for contexts of rule instances, that is, for all terms s and t such that s rewrites to t by R in one step.

Since the results of this chapter imply that for deduction by ordered paramodulation monotonicity is not needed, and any SPO can be extended to a west ordering, SPO is an interesting candidate for these purposes. Furthermore, in Chapter 5 of this thesis we will show that if R is a convergent TRS for some set of equations E, and $\rightarrow_R^+$ is included in a west ordering (like an SPO), then unfailing Knuth-Bendix completion on E with this west ordering will compute this R.

4.3.2 Non-monotonic E-compatible orderings

As mentioned in the beginning of this chapter, an important bottleneck for defining deduction techniques modulo built-in equational theories E is to find the required total (up to E-equal ground terms), *E-compatible* reduction ordering. But if the monotonicity requirement can be dropped, this becomes a much simpler task.

Let us consider as an example such an ordering for the case where E consists of associativity and commutativity (AC) properties. The AC-compatibility of such an ordering $\succ$ means that $s =_{AC} s' \succ t =_{AC} t'$ implies $s' \succ t'$. It is easy to check whether two ground terms are AC-equal by using their *flattened* forms: a term s can be *flattened* by removing all AC-operators f that are immediately below another f. For example, if f and g are AC-operators, then the term $h(f(f(a,a),f(b,g(c,g(d,e)))))$ is flattened into $h(f(a,a,b,g(c,d,e)))$. Two terms are AC-equal if, and only if, their flattened forms are equal up to permutation of arguments of AC-operators.

Therefore, when trying to define total AC-compatible orderings, the first idea that comes to mind is to apply general-purpose orderings on the flattened forms of the terms to be compared, like an RPO where AC-symbols have multiset status. But the resulting ordering may not be monotonic: if f is an AC-operator that is larger than g, then $f(a,a)$ will be larger than $g(a,a)$, but $f(g(a,a),a)$ will be larger than $f(a,f(a,a))$, because the latter term becomes $f(a,a,a)$ after flattening. However, it is not difficult to see that one obtains an AC-compatible west ordering from this simple approach.

4.4 Paramodulation with equations

In this section we introduce part of our ideas for the purely equational case. Dealing with this simple case first is useful not only for explanation purposes, but also because its results will be used in Chapter 5 on Knuth-Bendix completion. In the following, let $\succ$ be a given west ordering.

Definition 23 The inference rule of (equational) *ordered paramodulation* with respect to $\succ$ is:

$$\frac{l \simeq r \qquad s \simeq t}{(s[r]_p \simeq t)\sigma}$$

where $\sigma = mgu(s|_p, l)$, the most general unifier of $s|_p$ and l, where $s|_p$ is not a variable, and l is maximal in its premise, that is, for some ground substitution θ, it holds that $l\sigma\theta \succ r\sigma\theta$.

As said, the usefulness of the ordering restrictions in practical applications depends on the way the west ordering $\succ$ is given. For example, if only an approximation is given by a non-total ordering $\succ_v$ on terms with variables such that $s \succ_v t$ implies $s\sigma \succ t\sigma$ for all ground σ, then the inference is not needed if, for instance, $r\sigma \succeq_v l\sigma$.

We now define, by induction on $\succ_{mul}$, a ground TRS R_E *generated* by a set of equations E:

Definition 24 Let E be a set of equations. An instance e of the form $l \simeq r$ of an equation in E *generates* the rule $l \rightarrow r$ if

1. $l \succ r$, and
2. l and r are irreducible by R_e

where R_e is the set of rules generated by all instances d of equations in E such that $e \succ_{mul} d$. We denote by R_E the set of rules generated by all ground instances of E.

The previous construction is similar to the one of [BG94b] we explained in detail in Chapter 3, but here the rules are oriented only by a (possibly non-monotonic) west ordering. Hence the termination of R_E has to be ensured otherwise, and therefore we require not only the left hand sides to be irreducible, but also the right hand sides.

Property 25 Let E be a set of equations. Then for all rules $l \rightarrow r$ in R_E we have

1. r is irreducible by R_E
2. l is irreducible by $R_E \setminus \{l \rightarrow r\}$

Proof: For the first property, by construction, if an instance e generates $l \rightarrow r$, the term r is irreducible by R_e. Since $l \succ r$, and $\succ$ is irreflexive and fulfils the subterm property, clearly $l \rightarrow r$ itself does not reduce r either. Finally, for every rule $l' \rightarrow r'$ generated by an instance d with $d \succ_{mul} e$, we must have $l' \succeq l$ and hence $l' \succ r$ which implies that l' cannot be a subterm of r either.

For the second property, by construction, if an instance e generates $l \rightarrow r$, then the term l is irreducible by R_e. Now, as before, for every rule $l' \rightarrow r'$ generated by an instance d with $d \succ_{mul} e$ we must have $l' \succeq l$. Since l' must be irreducible by R_d which contains $l \rightarrow r$ we have $l' \not\equiv l$, and hence $l' \succ l$, which implies that l' cannot be a subterm of l. □

Lemma 26 For every set of equations E, the ground TRS R_E is convergent.

Proof: Termination follows by Property 25.1 and Lemma 20. For confluence, since R_E is terminating, by Newman's lemma we only need to show that R_E is locally confluent, which holds by Property 25.2. □

Theorem 27 Let E be a set of equations closed under ordered paramodulation with respect to a west ordering $\succ$. Then $R_E^* \models E$.

Proof: Since R_E is terminating, by Lemma 22 it follows that $\to^+_{R_E}$ is a reduction ordering. We now proceed by induction on the well-founded ordering on ground equations $(\to^+_{R_E})_{mul}$, denoted in the remainder of this proof by $\succ_R$. A contradiction is derived from the existence of a minimal w.r.t. $\succ_R$ ground instance e of the form $s\sigma \simeq t\sigma$ of an equation $s \simeq t$ in E such that $R_E^* \not\models e$. Since $R_E^* \not\models e$, the equation e is not a tautology of the form $u \simeq u$, and hence one its sides is strictly larger w.r.t. the (total) ordering $\succ$ than the other one. Moreover, again since $R_E^* \not\models e$, the equation e has not generated any rule of R_E. This must be because either $s\sigma$ or $t\sigma$ is reducible by R_e. Then there exists some equation $l \simeq r$ that has generated a rule $l\sigma \to r\sigma$ reducing $s\sigma$ or $t\sigma$. We consider the case where $s\sigma$ is reducible; the other one is analogous. Now we have $s\sigma|_p \equiv l\sigma$, and there are two possibilities:
(i) An inference: $s|_p$ is a non-variable position of s.
Then there exists an inference by ordered paramodulation:

$$\frac{l \simeq r \qquad s \simeq t}{(s[r]_p \simeq t)\theta}$$

whose conclusion has an instance d of the form $(s[r]_p \simeq t)\sigma$, such that $e \succ_R d$ and $R_E^* \not\models d$, contradicting the minimality of e. Note that this inference satisfies the ordering constraints of ordered paramodulation, since $l\sigma \succ r\sigma$.
(ii) Lifting: $s|_{p'}$ is a variable x for some prefix p' of p.
Then $p = p' \cdot p''$ and $x\sigma|_{p''}$ is $l\sigma$. Now let σ' be the ground substitution with the same domain as σ but where $x\sigma' \equiv x\sigma[r\sigma]_{p''}$ and $y\sigma' \equiv y\sigma$ for all other variables y. Then $R_E^* \not\models s\sigma' \equiv t\sigma'$ and $e\sigma \succ_R e\sigma'$, contradicting the minimality of e. □

4.5 A slightly stronger paramodulation rule

In the proof of termination of R_E (lemma 26) we can as well use lemma 21 instead of lemma 20. More precisely, we only need the right hand sides of the rules in R_E to be irreducible at non-topmost positions (instead of being completely irreducible). Due to this observation, we can restrict our paramodulation rule avoiding its application at topmost positions of small sides of equations. Then we obtain the following *strict ordered paramodulation* rule:

Definition 28 The inference rule of (equational) *strict ordered paramodulation* with respect to $\succ$ is:

$$\frac{l \simeq r \qquad s \simeq t}{(s[r]_p \simeq t)\sigma}$$

where $\sigma = mgu(s|_p, l)$, the most general unifier of $s|_p$ and l, where $s|_p$ is not a variable, and l is maximal in its premise, and if $p = \lambda$ then s is also maximal in its premise, that is, for some ground substitution θ, it holds that $l\sigma\theta \succ r\sigma\theta$, and, if $p = \lambda$ then we also have $s\sigma\theta \succ t\sigma\theta$.

Now to prove theorem 27, we slightly modify Definition 24, the generation of R_E, obtaining the following variant of it:

Definition 29 Let E be a set of equations. An instance e of the form $l \simeq r$ of an equation in E *generates* the rule $l \rightarrow r$ if

1. $l \succ r$,
2. l is irreducible by R_e, and
3. r is irreducible by R_e at non-topmost positions.

where R_e is the set of rules generated by all instances d of equations in E such that $e \succ_{mul} d$. In the remainder of this section, we denote by R_E the set of rules generated by all ground instances of E.

We can now adapt Property 25, and prove it analogously. From this property, lemma 26 holds as before but using lemma 21 to conclude termination.

Property 30 Let E be a set of equations. Then for all rules $l \rightarrow r$ in R_E we have

1. r is irreducible by R_E at non-topmost positions.
2. l is irreducible by $R_E \setminus \{l \rightarrow r\}$.

Now we adapt the proof of Theorem 27 for the strict ordered paramodulation rule.

Theorem 31 Let E be a set of equations closed under strict ordered paramodulation with respect to a west ordering $\succ$. Then $R_E^* \models E$.

Proof: As in Theorem 27, we proceed by induction on the well-founded ordering on ground equations $\succ_R$, deriving a contradiction from the existence of a minimal w.r.t. $\succ_R$ ground instance e of the form $s\sigma \simeq t\sigma$ of an equation $s \simeq t$ in E such that $R_E^* \not\models e$. Since $s\sigma$ cannot be equal to $t\sigma$, we assume w.l.o.g. that $s\sigma \succ t\sigma$.

The equation e has not generated any rule because $s\sigma$ is reducible by R_e or $t\sigma$ is reducible by R_e at a non-topmost position. Then there exists some equation $l \simeq r$ that has generated a rule $l\sigma \rightarrow r\sigma$, which reduces $s\sigma$ or $t\sigma$. The case where $s\sigma$ is reducible is analogous as in Theorem 27. For the other case we have $t\sigma|_p \equiv l\sigma$, for some position $p \neq \lambda$. If p is below a variable position in t then we apply the same lifting argument as in Theorem 27, and otherwise, since $p \neq \lambda$, we can conclude as well by the existence of an ordered paramodulation inference with a smaller conclusion. □

This strict ordered paramodulation rule of Definition 28 will be adapted for the Horn case and the case of general clauses in Sections 4.6 and 4.7 respectively, and the non-strict one of Definition 23 will be used to obtain the results on unfailing Knuth-Bendix completion in Chapter 5.

4.6 The Horn case

In this section we generalise the results of the previous section to Horn clauses. In the following inference system it is assumed that in each clause with a non-empty antecedent one of these negative equations, the one that is written underlined, has been *selected* (see Chapter 3 and [BG98]). In the Horn case this leads to positive unit strategies (and in the non-Horn case to positive strategies): left premises of paramodulations are unit clauses, and the only inferences involving non-unit clauses are equality resolution or paramodulation left on its selected equation.

Definition 32 The inference system $\mathcal{H}$ for Horn clauses with respect to the west ordering $\succ$ is defined as follows:

paramodulation right:

$$\frac{\rightarrow l \simeq r \qquad \rightarrow s \simeq t}{\rightarrow (s[r]_p \simeq t)\sigma} \qquad \text{where } \sigma = mgu(l, s|_p)$$

paramodulation left:

$$\frac{\rightarrow l \simeq r \qquad \Gamma, \underline{s \simeq t} \rightarrow \Delta}{(\Gamma, s[r]_p \simeq t \rightarrow \Delta)\sigma} \qquad \text{where } \sigma = mgu(l, s|_p)$$

equality resolution:

$$\frac{\Gamma, \underline{s \simeq t} \rightarrow \Delta}{(\Gamma \rightarrow \Delta)\sigma} \qquad \text{where } \sigma = mgu(s, t)$$

where moreover in both paramodulation rules $s|_p$ is not a variable, l is maximal in its premise, and if $p = \lambda$ then s is also maximal in its premise, that is, for some ground substitution θ, it holds that $l\sigma\theta \succ r\sigma\theta$, and, if $p = \lambda$ then also $s\sigma\theta \succ t\sigma\theta$.

Definition 33 Let S be a set of Horn clauses. We denote by E_S the subset of all positive unit clauses in S, that is, $E_S = \{\ s \simeq t \ \mid \rightarrow s \simeq t \in S\ \}$, and we denote by R_S the set of rules generated by E_S as in Definition 29 (i.e. $R_S = R_{E_S}$).

The following constructions follow the same lines as the model generation method explained in Chapter 3. We now use multiset extensions for lifting orderings $\succ$ on terms to orderings on equations and clauses. Let C be a ground clause, and let $emul(s \simeq t)$ be $\{s,t\}$ if $s \simeq t$ is a positive equation in C, and $\{s,s,t,t\}$ if it is negative. Then, if $\succ$ is an ordering, we define the ordering $\succ_e$ on (occurrences of) ground equations in a clause by $e \succ_e e'$ if $emul(e) \succ_{mul} emul(e')$. Similarly, $\succ_c$ on ground clauses is defined $C \succ_c D$ if $mse(C)$ $(\succ_{mul})_{mul}$ $mse(D)$, where $mse(C)$ is the multiset of all $emul(e)$ for ocurrences e of equations in C.

Theorem 34 (refutation completeness of $\mathcal{H}$ for Horn clauses)
Let S be a set of Horn clauses closed under $\mathcal{H}$. Then $\Box \in S$ if, and only if, S is unsatisfiable.

Proof: The left to right implication is trivial. For the other one, let S be a set of clauses closed under $\mathcal{H}$ such that $\Box \notin S$. As in Theorem 27, we proceed by induction on $\succ_c^{R_S}$, which will be denoted in the remainder of this proof by $\succ_R$. This ordering is monotonic where needed (see the cases below) and well-founded. Again it is proved that R_S^* is a model of S by deriving a contradiction from the existence of a minimal w.r.t. $\succ_R$ ground instance C of a clause in S such that $R_S^* \not\models C$. Since E_S is closed under strict ordered paramodulation, by Theorem 31, we have $R_S^* \models E_S$, and therefore C cannot be a positive unit clause. Two cases have to be considered:

1. C is an instance $\Gamma\sigma, \underline{s\sigma \simeq t\sigma} \rightarrow \Delta\sigma$ of a clause $\Gamma, \underline{s \simeq t} \rightarrow \Delta$, where $s\sigma \equiv t\sigma$. Then there is an inference by equality resolution

$$\frac{\Gamma, \underline{s \simeq t} \rightarrow \Delta}{(\Gamma \rightarrow \Delta)\theta}$$

whose conclusion has an instance D of the form $\Gamma\sigma \rightarrow \Delta\sigma$ such that $C \succ_R D$ and moreover, D is in S and $R_S^* \not\models D$, which is a contradiction.

2. C is an instance $\Gamma\sigma, \underline{s\sigma \simeq t\sigma} \rightarrow \Delta\sigma$ of a clause $\Gamma, \underline{s \simeq t} \rightarrow \Delta$, where, w.l.o.g., $s\sigma \succ t\sigma$.

   Then, since $R_S^* \not\models C$, we have $R_S^* \models s\sigma \simeq t\sigma$, and, since R_S is convergent, there must be a rewrite proof of $s\sigma \simeq t\sigma$ by R_S, that is, $s\sigma$ and $t\sigma$ must rewrite into the same normal form by R_S. This implies that either $t\sigma$ is reducible at a non-topmost position or else $s\sigma$ is reducible. (Note that it cannot be the case that the only possible reduction step on $s\sigma \simeq t\sigma$ is at the topmost position of $t\sigma$. By such a step, a new term t' is obtained with $t\sigma \succ t'$ and t' again irreducible at non-topmost positions; since $s\sigma \succ t\sigma$, such a sequence of topmost steps on $t\sigma$ can never produce $s\sigma$).

Then either the lifting argument applies like in Theorem 27, or else there is an inference by paramodulation left:

$$\frac{\rightarrow l \simeq r \qquad \Gamma, \underline{s \simeq t} \rightarrow \Delta}{(\Gamma, s[r]_p \simeq t \rightarrow \Delta)\theta}$$

whose conclusion has an instance D of the form $(\Gamma, s[r]_p \simeq t \rightarrow \Delta)\sigma$ such that $C \succ_R D$ and $R_S^* \not\models D$, which is a contradiction. □

4.7 General clauses

Now we consider general clauses. As for the Horn case, we consider that in each clause with a non-empty antecedent one of its negative equations, the one that is written underlined, has been selected.

Definition 35 The inference system $\mathcal{I}$ with respect to the given west ordering $\succ$ is defined as follows:

paramodulation right:

$$\frac{\rightarrow l \simeq r, \Delta \qquad \rightarrow s \simeq t, \Delta'}{(\rightarrow s[r]_p \simeq t, \Delta, \Delta')\sigma} \qquad \text{where } \sigma = mgu(l, s|_p)$$

paramodulation left:

$$\frac{\rightarrow l \simeq r, \Delta \qquad \Gamma, \underline{s \simeq t} \rightarrow \Delta'}{(\Gamma, s[r]_p \simeq t \rightarrow \Delta, \Delta')\sigma} \qquad \text{where } \sigma = mgu(l, s|_p)$$

equality resolution:

$$\frac{\Gamma, \underline{s \simeq t} \rightarrow \Delta}{(\Gamma \rightarrow \Delta)\sigma} \qquad \text{where } \sigma = mgu(s, t)$$

equality factoring:

$$\frac{\rightarrow s \simeq t, s' \simeq t', \Delta}{(t \simeq t' \rightarrow s \simeq t, \Delta)\sigma} \qquad \text{where } \sigma = mgu(s, s')$$

where in both paramodulation rules $s|_p$ is not a variable. The ordering restrictions are as follows:

In paramodulation right, l is strictly maximal in $l \simeq r$, and $l \simeq r$ is the strictly maximal equation in its premise, and if $p = \lambda$, then s is strictly maximal in $s \simeq t$ and $s \simeq t$ is the strictly maximal equation in its premise. Formally: for some ground substitution θ it holds that $l\sigma\theta \succ r\sigma\theta$ and $(l \simeq r)\sigma\theta \succ_e e\sigma\theta$ for all equations e in Δ, and if $p = \lambda$, then $s\sigma\theta \succ t\sigma\theta$ and $(s \simeq t)\sigma\theta \succ_e e\sigma\theta$ for all equations e in Δ'.

In paramodulation left, l is strictly maximal in $l \simeq r$, and $l \simeq r$ is strictly maximal in its premise, and if $p = \lambda$, then s is strictly maximal in $s \simeq t$.

Finally, in equality factoring, s is strictly maximal in $s \simeq t$ and $s \simeq t$ is maximal.

Definition 36 Let S be a set of clauses. An instance C of the form $\rightarrow l \simeq r, \Delta$ of a clause in S *generates* the rule $l \rightarrow r$ if

1. $R_C^* \not\models C$,
2. $l \succ r$, and $(l \simeq r) \succ_e e$ for all equations e in Δ,
3. l is irreducible by R_C,
4. r and Δ are irreducible at non-topmost positions by R_C
5. $R_C^* \models r \simeq t$ for no equation $l \simeq t$ in Δ.

where R_C is the set of rules generated by all instances D of clauses in S such that $C \succ_c D$. In the remainder of this section, we denote by R_S the set of rules generated by all ground instances of S.

Again we have the following property which implies convergence of R_S.

Property 37 Let S be a set of clauses. Then for all rules $l \rightarrow r$ in R_S we have

1. r is irreducible by R_S at non-topmost positions.
2. l is irreducible by $R_S \setminus \{l \rightarrow r\}$.

Lemma 38 Let S be a set of clauses. Then the ground TRS R_S is convergent.

Lemma 39 Let S be a set of clauses. If $\rightarrow l \simeq r, \Delta$ is an instance C of a clause in S that generates the rule $l \rightarrow r$ in R_S, then $R_S^* \not\models \Delta$.

Proof: We first prove that if $R_S^* \models s \simeq t$ for ground s and t that are irreducible at non-topmost positions and such that $s \succ t$, then no rules with left hand sides greater than s (w.r.t. $\succ$) are used in the rewrite proof. We proceed by induction on the size of $s \simeq t$ w.r.t. $\succ_{mul}$. The first step can only apply at topmost position of s or t. If it is on t then we obtain $s \simeq r$, where $t \rightarrow r$ is the applied rule in R_S, and hence $s \succ t \succ r$, by definition of R_S, and r is irreducible at non-topmost positions by Property 37. 1, and then we can conclude by induction hypotheses. Otherwise, we obtain $r \simeq t$, where $s \rightarrow r$ is the applied rule in R_S, and hence as before $s \succ r$ and r is irreducible at non-topmost positions. If $r \equiv t$ we are done, and otherwise if $r \succ t$ or $t \succ r$ we can conclude by induction.

Now, assume $s \simeq t$ is a ground equation in Δ with a rewrite proof using R_S and such that $s \succ t$. The rules generated by clauses D with $D \succ_c C$ cannot be used, since they have left hand sides greater than l and hence greater than s w.r.t. $\succ$, and s and t are irreducible at non-topmost positions. But since $R_C^* \not\models s \simeq t$ the rule $l \rightarrow r$ is used. Hence $s \equiv l$ and $s \simeq t$ rewrites into $r \simeq t$, to which $l \rightarrow r$ cannot be applied any more, which contradicts the last condition of Definition 36. □

We now introduce a well-founded ordering $\succ_R$ that will be used in the proof of Theorem 43. In contrast to the Horn and equational cases, it does not coincide with $\rightarrow_{R_S}^+$.

Definition 40 Let S be a set of clauses. By $\succ_R$ we denote the smallest transitive relation such that $s \succ_R t$ whenever (i) $s \rightarrow_{R_S}^+ t$ or (ii) $s \rhd t$ or (iii) s and t are irreducible at non-topmost positions w.r.t. R_S and $s \succ t$.

The ordering $\succ_R$ fulfils the following properties wrt. R_S.

Property 41 Let S be a set of clauses. Then for all ground terms s and t s.t. s is irreducible at non-topmost positions and $s \succ_R t$ we have

1. $s \succ t$

2. t is irreducible at non-topmost positions

Proof: If $s \succ_R t$ by case (ii) or (iii) it trivially holds. Otherwise, $s \rightarrow_{R_S} s_1 \rightarrow_{R_S} \ldots \rightarrow_{R_S} s_n \rightarrow_{R_S} t$. Since s is irreducible at non-topmost positions and, by property 37.1, all right hand sides of rules in R_S are irreducible at non-topmost positions, all steps in the sequence are at topmost positions, which implies on one hand that t is irreducible at non-topmost positions, and on the other, by definition of R_S, that $s \succ s_1 \succ \ldots \succ s_n \succ t$. □

Lemma 42 Let S be a set of clauses. Then $\rightarrow_{R_S}^+$ is a reduction ordering and $\succ_R$ is well-founded.

Proof: Since R_S is a terminating TRS, $\rightarrow_{R_S}^+$ is a reduction ordering by Lemma 22. Now assume $\succ_R$ is not well-founded. Since $(\rightarrow_{R_S}^+ \cup \rhd)^+$ is well-founded by Lemma 19, there is also some such infinite sequence $t_1 \succ_R t_2 \succ_R t_3 \succ_R \ldots$ starting with case (iii): where t_1 and t_2 are irreducible w.r.t. R_S at non-topmost positions and $t_1 \succ t_2$. Then, by Property 41, t_3, t_4 etc, are all irreducible at non-topmost positions, and $t_1 \succ t_2 \succ t_3 \ldots$ w.r.t. the well-founded west ordering $\succ$, which is a contradiction. □

Theorem 43 (refutation completeness of $\mathcal{I}$ for general clauses)
Let S be a set of clauses closed under $\mathcal{I}$ with respect to a west ordering $\succ$. Then $\square \in S$ if, and only if, S is unsatisfiable.

Proof: We prove that if $\Box \notin S$ then R_S^* is a model of S by induction on $(\succ_R)_c$, which is a well-founded ordering on clauses. In the following, we (ambiguously) write $\succ_R$ for terms, equations and clauses instead of $\succ_R$, $(\succ_R)_e$ and $(\succ_R)_c$ respectively. We derive a contradiction from the existence of a minimal w.r.t. $\succ_R$ ground instance C of a clause in S such that $R_S^* \not\models C$.

1. We first consider the case where C is an instance with σ of a positive clause $\to s \simeq t, \Delta$, where $s\sigma \simeq t\sigma$ is strictly maximal with respect to $\succ_e$ in C and w.l.o.g. $s\sigma \succ t\sigma$. Since $R_S^* \not\models C$, we know C has not generated any rule due to one of the following reasons:

1a. $t\sigma$ is reducible by R_C at a non-topmost position. Then there exists some clause $\to l \simeq r, \Delta'$ in S whose instance C' with $C \succ_c C'$ generates the rule $l\sigma \to r\sigma$ reducing $t\sigma$.

If, for some prefix p' of p, the term $t|_{p'}$ is a variable x, then the same lifting argument as in proof of Theorem 27 applies.

Otherwise, $t|_p$ is a non-variable subterm of t, and there exists an inference by paramodulation right:

$$\frac{\to l \simeq r, \Delta' \qquad \to s \simeq t, \Delta}{(\to s \simeq t[r]_p, \Delta, \Delta')\theta}$$

whose conclusion has an instance D of the form $(\to s \simeq t[r]_p, \Delta, \Delta')\sigma$ such that by Lemma 39 $R_S^* \not\models D$. This contradicts the minimality of C since $C \succ_R D$ for the following reasons: $mse(C) = mse(\Delta) \cup \{\ \{s\sigma, t\sigma\}\ \}$ and $mse(D) = mse(\Delta) \cup \{\ \{s\sigma, t\sigma[r\sigma]_p\}\ \} \cup mse(\Delta')$. Then we need to prove that $\{s\sigma, t\sigma\} \succ_{R_{mul}} \{s\sigma, t\sigma[r\sigma]_p\}$ and $(s\sigma \simeq t\sigma) \succ_R e\sigma$ for all equations e in Δ'. The first one holds since, by monotonicity of $\to_{R_S}^+$, we have $t\sigma[l\sigma]_p \succ_R t\sigma[r\sigma]_p$. For the second one, we have $l\sigma \succeq u\sigma$ for all terms u in Δ', and since they are also irreducible at non-topmost positions, we have $l\sigma \succeq_R u\sigma$. Furthermore $t\sigma \succ_R l\sigma$, since $t\sigma \rhd l\sigma$, which implies $t\sigma \succ_R u\sigma$ for all terms u in Δ', and hence we can conclude.

1b. $s\sigma$ is reducible by R_C and case 1a. does not apply. Then we argue as in the previous case by lifting or an inference in s whose conclusion has an instance D of the form $(\to s[r]_p \simeq t, \Delta, \Delta')\sigma$, contradicting the minimality of C. Here $C \succ_R D$ if $p \neq \lambda$ as in the previous case. If $p = \lambda$ then $(s\sigma \simeq t\sigma) \succ_R e\sigma$ for all equations e in Δ': since $C \succ_c C'$, and $s\sigma \simeq t\sigma$ is the strictly maximal equation of C, we have $(s\sigma \simeq t\sigma) \succeq_e (l\sigma \simeq r\sigma) \succ_e e\sigma$, and since all these equations are irreducible at non-topmost positions, $(s\sigma \simeq t\sigma) \succ_R e\sigma$.

1c. An equation $u\sigma \simeq v\sigma$ in Δ is reducible by R_C at a non-topmost position. The proof is like case 1a.

1d. None of the previous cases applies and Δ is of the form $s' \simeq t', \Delta'$ where $s\sigma \equiv s'\sigma$ and $R_S^* \models t\sigma \simeq t'\sigma$, that is, the last condition of Definition 36 fails. Then there exists some inference by equality factoring

$$\frac{\to s \simeq t, s' \simeq t', \Delta'}{(t \simeq t' \to s \simeq t, \Delta')\theta}$$

whose conclusion has an instance D such that $R^*_S \not\models D$. This contradicts the minimality of C: since $s\sigma$, $t\sigma$, and $t'\sigma$ are irreducible at non-topmost positions, and $s'\sigma \succ t\sigma$ and $s'\sigma \succ t'\sigma$, we have $C \succ_R D$.

2. If C is an instance with σ of $\to s \simeq t, \Delta$, where $s\sigma \simeq t\sigma$ is maximal but not strictly maximal with respect to $\succ_e$ in C, then condition 2 of Definition 36 fails. If $s\sigma \simeq t\sigma$ is reducible at non-topmost positions, then the same reasoning as in case 1a. applies. Otherwise, the proof of case 1d. applies.

3. If C is an instance with σ of a clause $\Gamma, \underline{s \simeq t} \to \Delta$, where $s\sigma \equiv t\sigma$, we conclude, as in the proof of Theorem 34, by equality resolution.

4. If C is an instance with σ of a clause $\Gamma, \underline{s \simeq t} \to \Delta$, where $s\sigma \not\equiv t\sigma$. Then $R^*_S \models s\sigma \simeq t\sigma$. Then, since R_S is convergent, there is a rewrite proof for $s\sigma \simeq t\sigma$. W.l.o.g. assume $s\sigma \succ t\sigma$. This implies that either $s\sigma$ is reducible or $t\sigma$ is reducible at a non-topmost position (otherwise the only possible reduction step is at the top of $t\sigma$ and a new term t' is obtained with $t\sigma \succ t'$ and t' again irreducible except at the top; since $s\sigma \succ t\sigma$, a sequence of topmost steps on $t\sigma$ can never produce $s\sigma$).

Then, as before, either the lifting argument applies or there is an inference by paramodulation left (we only develope the case where the step takes place in s):

$$\frac{\to l \simeq r, \Delta' \qquad \Gamma, \underline{s \simeq t} \to \Delta}{(\Gamma, s[r]_p \simeq t \to \Delta', \Delta)\theta}$$

whose conclusion has an instance D of the form $(\Gamma, s[r]_p \simeq t \to \Delta, \Delta')\sigma$ such that $R^*_S \not\models D$. This is a contradiction as before, since also here $C \succ_R D$ for the following reasons: we have $s\sigma \trianglerighteq l\sigma$ and hence $s\sigma \succeq_R l\sigma$. We also have $l\sigma \succ_R r\sigma$, and $l\sigma \simeq r\sigma \succ_R u\sigma \simeq v\sigma$ for all equations $u \simeq v$ in Δ' and hence
$\{s\sigma, s\sigma, t\sigma, t\sigma\} \succ_{Rmul} \{l\sigma, r\sigma\} \succ_{Rmul} \{u\sigma, v\sigma\}$. □

4.8 Redundancy and constraints

In this section we shortly mention the applicability of techniques for *redundancy elimination* and *constraint inheritance* in the context of this chapter.

There are standard ways for uniformly covering simplification and deletion techniques that are compatible with refutation completeness by notions of redundancy for inferences and clauses, where *saturation* amounts to the closure under $\mathcal{I}$ *up to redundant inferences*.

In the setting of this chapter, there is an important difference, however, because two different orderings are considered. The inferences are computed w.r.t. the west ordering $\succ$, and the completeness proof uses the reduction ordering $\succ_R$. Unfortunately, redundancy should hence be defined w.r.t. $\succ_R$, which is unknown during the saturation process. But in many cases it is clear that $\succ_R$ can be sufficiently approximated. For example, practical redundancy notions like tautology deletion or subsumption are (trivially) correct w.r.t. any $\succ_R$.

Ordered paramodulation is a very adequate inference rule for dealing with constrained clauses and the *basic* strategy. These ideas are directly applicable here: subterms created by unifiers of inferences on ancestors can be *blocked* for inferences, and in our setting this is true also for proper subterms of the term $r\sigma$ (which are irreducible by R_S in our proof) in the conclusions of paramodulation inferences.

4.9 Conclusions

One may wonder whether more restrictive inference systems could be complete as well. But a number of negative results have been obtained, which are best described by the following counterexamples.

Example 44 Inferences on non-maximal positive atoms are needed: if the inconsistent set S consists of the clauses

$$\begin{array}{rl} & \rightarrow P(b,b), P(a,b) \\ \underline{P(b,b)}, P(a,b) & \rightarrow \\ & \rightarrow a \simeq b \end{array}$$

with a west ordering $\succ$ where $a \succ b$ and $P(b,b) \succ P(a,b)$, then the only other possible inference produces the tautology $P(a,b) \rightarrow P(a,b)$.

Example 45 Also inferences on small sides of positive equations are needed: if the inconsistent set S consists of the clauses

$$\begin{array}{rl} & \rightarrow a \simeq b \\ & \rightarrow g(b) \simeq g(g(a)) \\ x \simeq g(x) & \rightarrow \end{array}$$

with a west ordering $\succ$ where $a \succ b$ and $g(b) \succ g(g(a))$, then the only other possible inferences produce clauses of the form $b \simeq g^n(b) \rightarrow$ and $b \simeq g^n(a) \rightarrow$ for $n > 1$.

Example 46 Also inferences on small sides of negative equations are needed: if the inconsistent set of clauses S is

$$\begin{array}{rl} & \rightarrow a \simeq b \\ f(x,x) \simeq f(a,b) & \rightarrow \end{array}$$

with a west ordering $\succ$ where $f(a,a) \succ f(a,b)$ and $f(b,b) \succ f(a,b)$, then S is closed.

Example 47 [Lyn97] For arbitrary selection strategies, these techniques are incompatible with tautology deletion. Consider the set

$$\begin{array}{lll} 1. & & \rightarrow P(c,b,b) \\ 2. & P(c,c,b), P(c,b,c) & \rightarrow b \simeq c \\ 3. & P(x,y,y) & \rightarrow P(x,y,x) \\ 4. & P(x,y,y) & \rightarrow P(x,x,y) \\ 5. & P(c,c,c) & \rightarrow \end{array}$$

This clause set is inconsistent: from 1. and 3. we get $P(c,b,c)$, and from 1. and 4. we get $P(c,c,b)$; these two atoms together with 2. produce $b \simeq c$, which gives the empty clause with 1. and 5. But the empty clause cannot be obtained by ordered paramodulation on non-tautology clauses with an ordering where $b \succ c$, and where always the positive literals are selected (except in clause 5., which has none). In fact, the only new clauses obtained are tautologies.

Chapter 5

Knuth-Bendix completion

In this chapter a well-known open problem concerning the Knuth-Bendix completion procedure is solved. It was posed by N. Dershowitz and J-P. Jouannaud on the RTA list of open problems [RTA01] since its creation in 1991.

5.1 Introduction

As explained in Subsection 1.1.1 of Chapter 1, the aim of the Knuth-Bendix completion procedure is to build a convergent rewrite system R for a given a set of equations E and an ordering $\succ$ on terms. In fact, all current state-of-the-art theorem provers in pure equational logic, like Waldmeister [HBVL97], are based on variations of the Knuth-Bendix completion procedure. When given an ordering $\succ$ that can be extended to a total reduction ordering on ground terms, this procedure is complete in the sense that it always finds a (possibly infinite) convergent TRS R, logically equivalent to E, contained in $\succ$. This is very useful for automatically proving that an equation $s \simeq t$ follows from a set of equations E, because after a finite number of steps of such a procedure always a TRS R is reached by which a rewrite proof for $s \simeq t$ exists.

But a rewrite system already terminates if $\succ$ is only a reduction ordering (in fact, a rewrite system terminates if, and only if, it is contained in a reduction ordering). Hence a very natural (frequently asked) question is: what happens if we apply completion with a given desirable orientation by a reduction ordering that cannot be extended to a total one, like $f(a) \rightarrow f(b)$ and $g(b) \rightarrow g(a)$, for which a and b must be uncomparable in any monotonic extension? Here this question is answered affirmatively: by a careful further analysis of our technique, we obtain the first practical Knuth-Bendix completion procedure that finds a convergent TRS for a given set of equations E and a (possibly non-totalizable) reduction ordering $\succ$ whenever it exists. Note that for arbitrary reduction orderings it does not always exist: in each E-congruence class there should be a single minimal element. For example, if $E = \{a \simeq b, a \simeq c\}$ then one of a, b or c should be smaller than the other two.

In this chapter, E denotes a set of equations and $\succ_r$ a reduction ordering on $T(\mathcal{F}, \mathcal{X})$. Then a *convergent TRS for E and $\succ_r$* is a convergent TRS, logically equivalent to E, and such that $l \succ_r r$ for all its rules $l \rightarrow r$. The problem we deal with is finding a convergent TRS for the given E and $\succ_r$ whenever it exists, and find it in finite time if it is finite.

5.2 A theoretical procedure

It is not difficult to devise a procedure of theoretical nature, i.e., without much practical value, for finding a convergent TRS for E and $\succ_r$. The idea is to systematically enumerate all equational consequences of E, say $s_1 \simeq t_1, s_2 \simeq t_2, \ldots$ If a finite convergent TRS R exists, there exists some (probably huge) i, such that R is contained in (the orientations of) a subset of $\{s_1 \simeq t_1, \ldots, s_i \simeq t_i\}$. One can find this R by periodically checking during the enumeration process whether (i) the subset R of orientable (with $\succ_r$) rules of $\{s_1 \simeq t_1, \ldots s_i \simeq t_i\}$ is confluent and (ii) whether R entails E. The confluence of R can be decided by checking joinability of all its critical pairs. After this, entailment of E can be decided by rewriting.

The following lemma states that, after enumerating i equational consequences of E, it is not necessary to check for confluence of the orientations for each subset of $\{s_1 \simeq t_1, \ldots, s_i \simeq t_i\}$, but that it suffices to consider the orientations of the whole set $\{s_1 \simeq t_1, \ldots, s_i \simeq t_i\}$, because unnecessary rules do not destroy convergence.

Lemma 48 Let E be a set of equations, let $\succ_r$ be a reduction ordering on $T(\mathcal{F}, \mathcal{X})$, and let R be a convergent TRS for E and $\succ_r$. Let R' be any set of rules $l \rightarrow r$ such that $l \succ_r r$ and $E \models l \simeq r$. Then $R \cup R'$ is a convergent TRS for E and $\succ_r$.

Proof: Clearly all critical pairs between rules in $R \cup R'$ are logical consequences of E. Since R is a convergent TRS for E, all these critical pairs have rewrite proofs by R, i.e., they are joinable. Hence $R \cup R'$ is a convergent TRS for E and $\succ_r$. □

The TRS R found in this way may not be minimal, that is, it may have some proper subset R' that is also a convergent TRS for E and $\succ_r$, but such a minimal TRS always exists, and it can be effectively computed from a finite R:

Lemma 49 ([DMT88]) For every convergent TRS its unique *canonical*[1] version can be obtained by *interreducing* it by (i) normalizing all right hand sides and then (ii) removing all rules whose left hand sides are reducible by other rules.

[1] Unfortunately, the word canonical is sometimes also used as equivalent of convergent.

5.3 Practical procedures

Regarding practically useful procedures, Devie showed that for left- and right *linear* E (i.e., no variable occurs more than once in a side of an equation) standard Knuth-Bendix completion finds R [Dev90]. For the general problem, the previously existing procedures still relied on the enumeration of all equational consequences (see e.g., [Dev92]).

In the following, R will denote the canonical TRS for E and $\succ_r$ (we assume R exists), and we denote by R_g the canonical TRS for $gnd(R)$, the set of all ground instances of rules of R.

The following lemma strengthens the uniqueness result of [DMT88] to TRS included in a west ordering, but only for the ground case:

Lemma 50 Let $\succ$ be a west ordering, and let R_1 and R_2 be interreduced TRS over $T(\mathcal{F})$, both included in $\succ$, and such that $R_1^* = R_2^*$. Then $R_1 = R_2$.

Proof: Consider the rule $l \rightarrow r$ in $(R_1 \cup R_2) \setminus (R_1 \cap R_2)$ with minimal l w.r.t. $\succ$. Assume w.l.o.g. $l \rightarrow r \in R_1$. Since $R_1^* \models l \simeq r$ and hence $R_2^* \models l \simeq r$ and R_2 is convergent, there must be a rewrite proof by R_2 for $l \simeq r$. But, since R_1 is interreduced, r and all strict subterms of l are irreducible w.r.t. R_1, and hence also w.r.t. R_2, since any rule reducing them has a lhs smaller than l w.r.t. $\succ$. Hence there is a rule $l \rightarrow r'$ in R_2. But then r' and r are both irreducible by R_2 and $R_2 \models r \simeq r'$. This implies that r and r' are the same term, contradicting the assumption that $l \rightarrow r \notin R_2$. □

In the following we rely on some definition given in Chapter 4. We consider the ordered equational paramodulation rule given in Definition 23 w.r.t. a west ordering $\succ$ extending $\succ_r$, and the ground TRS R_E as in Definition 24. Then, by Property 25, for all rules $l \rightarrow r$ in R_E we have that $l \succ r$, that r is irreducible by R_E and that l is irreducible by $R_E \setminus \{l \rightarrow r\}$.

Lemma 51 Let E' be the closure of E under ordered equational paramodulation w.r.t. a west ordering $\succ$ extending $\succ_r$. Then $R_g = R_{E'}$.

Proof: By Theorem 27 we have $R_{E'}^* \models E'$, and hence $R_{E'}^* \models E$, and hence $E^* = R_{E'}^*$ since all rules in $R_{E'}$ are logical consequences of E. Furthermore, we clearly have $E^* = R^* = R_g^*$, and hence $R_g^* = R_{E'}^*$. We also have $R_g \subseteq \succ_r \subseteq \succ$, and $R_{E'} \subseteq \succ$. Therefore, since both are interreduced we conclude by Lemma 50 that $R_{E'} = R_g$. □

Now we come to the main theorem of this chapter. It says that R is a subset of the closure of E under the inference rule of ordered paramodulation (which we recall here from Definition 23):

$$\frac{l \simeq r \qquad s \simeq t}{(s[r]_p \simeq t)\sigma}$$

where $\sigma = mgu(s|_p, l)$ and $s|_p$ is not a variable, and with ordering restrictions saying that the inference is not needed if $r\sigma \succeq_r l\sigma$ (or, more generally, that the inference is needed only if $l\sigma\theta \succ r\sigma\theta$ for some ground substitution θ, where $\succ$ is the west ordering extending $\succ_r$).

Theorem 52 Let E' be the closure of E under equational ordered paramodulation w.r.t. a west ordering $\succ$ extending $\succ_r$. Then $E' \supseteq R$.

Proof: W.l.o.g. assume there are sufficiently new constants in $\mathcal{F}$ that do not occur in E or R. Let $l \to r$ be an arbitrary rule in R. We prove that $l \simeq r \in E'$. Let σ be the ground substitution replacing each variable with a distinct new constant. Then $l\sigma \to r\sigma$ is in $gnd(R)$ and also in R_g: since R is interreduced, r is irreducible w.r.t. R (and hence $r\sigma$ by $gnd(R)$), and l is irreducible w.r.t. $R \setminus \{l \to r\}$ (and hence $l\sigma$ by $gnd(R) \setminus \{l\sigma \to r\sigma\}$). If $l\sigma \to r\sigma$ is in R_g and by the previous lemma $R_g = R_{E'}$, then $l\sigma \to r\sigma$ is in $R_{E'}$. Since the new constants do not occur in E, if $l\sigma \to r\sigma \in R_{E'}$, then some $l' \simeq r'$ is in E', and $l\sigma \equiv l'\theta\sigma$ and $r\sigma \equiv r'\theta\sigma$ for some θ. We conclude by showing that θ is the identity substitution: there is no equation $l' \simeq r'$ that strictly subsumes a rule in R; otherwise, the rewrite proof by R of $l' \simeq r'$ would apply some rule different from $l \to r$, which would then also reduce $l \to r$, contradicting $l \to r \in R$. □

In Chapter 9 some interesting research directions are given concerning Knuth-Bendix completion with (non-totalizable) reduction orderings.

Chapter 6

Completeness of Arbitrary Selection Strategies

A crucial way for reducing the search space in automated deduction are the so-called *selection strategies*: in each clause, the subset of *selected* literals are the only ones involved in inferences. For first-order Horn clauses without equality, resolution is complete with an arbitrary selection of one single literal in each clause [dN96]. For Horn clauses with built-in equality, i.e., paramodulation-based inference systems, the situation is far more complex.

In this chapter we show that if a paramodulation-based inference system is complete with eager selection of negative equations and, moreover, it is compatible with equality constraint inheritance, then it is complete with arbitrary selection strategies. A first important application of this result is the one for paramodulation wrt. non-monotonic orderings, which was left open in Chapter 4.

6.1 Introduction

As explained in Chapters 1 and 3, a crucial way for reducing the search space in automated deduction are the so-called *selection strategies.* In such strategies the possible inferences between clauses are restricted to the ones involving *selected* literals. This selection can be done in several different ways. Well-known examples of selection strategies are the *maximal* (or *ordered*) strategies for a given atom ordering. For example, in a maximal resolution strategy, a (ground) inference between $A \vee C$ and $\neg A \vee D$ is performed only if A is larger in the given atom ordering than all other atoms in C and D. Another well-known selection strategy is the so-called *eager* negative selection strategy, where in each clause a single negative literal is selected whenever there is any. This leads to the so-called *positive* strategies (positive unit strategies in the Horn case) because always the left premise of each (resolution or paramodulation) inference is a positive (unit) clause. These strategies are usually easier to prove complete, but sometimes they are not very efficient, because, roughly speaking, one enumerates all solutions of its conditions before using the positive information of a clause (as discussed in [Der91]).

For first-order Horn clauses without equality, resolution is complete with an arbitrary selection of one single literal in each clause ([dN96], Theorem 6.7.4). For Horn clauses with built-in equality, i.e., paramodulation-based inference systems, the situation is far more complex. In [Lyn97] some positive and negative results are given for the case where a total *reduction* (well-founded, monotonic) ordering on ground terms is given. Then arbitrary selection strategies are compatible with superposition (that is, paramodulation involving only maximal sides of equations). Also conditions for eliminating *redundant* clauses are given in [Lyn97], and counter examples indicating the limitations for doing so. For example, in certain circumstances the elimination of tautologies can lead to incompleteness.

In this chapter we obtain a more general result for Horn clauses with equality, namely that, if a paramodulation-based inference system is complete with eager selection of negative equations and, moreover, it is compatible with equality constraint inheritance (like, in particular, it happens for superposition), then it is complete with arbitrary selection strategies.

Our completeness result is based on transformations of proof trees. Its generality allows us to obtain directly the completeness of arbitrary selection strategies for other inference systems, apart from the one of superposition with total reduction orderings. A first important application of our result is the one for paramodulation with non-monotonic orderings of Chapter 4, where the completeness of strategies different from eager negative selection was left open. There, techniques for dropping the monotonicity requirement were introduced, with the only properties required for the ordering being well-foundedness and the subterm property. However, the inference system of Chapter 4 still required the eager selection of negative equations. In Section 6.4 we show that those results are compatible with equality constraint inheritance and hence with the *basic* strategy, thus further restricting the search space. Therefore, our transformation method is applicable, and we obtain the completeness of the same inference system but with arbitrary selection strategies.

The structure of the chapter is the following. In Section 6.2 we present our transformation method for proofs, and in Section 6.3 we give our main result on completeness of arbitrary selection strategies. In Section 6.4 we apply this new technique to the case of the inference system of Chapter 4. Finally, in Section 6.5 we give some conclusions.

6.2 The transformation method

In the following we deal with inference systems that are based on some selection strategy. A selection strategy is a function from ground clauses to non-empty sets of literals, such that the selected literals for a clause appear in the clause. An inference between two ground clauses is allowed only if the literals involved in the inference are selected. As usual, a non-ground inference represents all its ground instances fulfilling the required conditions. In our case, a non-ground inference is allowed only if, for some ground instance of the inference, the involved literals are selected.

Our hypothesis in this section is that we have at hand a paramodulation-based inference system $\mathcal{N}$ for first-order Horn clauses, which is compatible with equality constraint inheritance and complete with a concrete strategy with eager selection of negative equations. Let $\mathcal{N}$ consist of the following inference rules:

paramodulation right:

$$\frac{\rightarrow l \simeq r \mid T_1 \qquad \rightarrow s \simeq t \mid T_2}{\rightarrow s[r]_p \simeq t \mid s|_p = l \wedge T_1 \wedge T_2} \qquad \text{if } s|_p \notin \mathcal{X}$$

paramodulation left:

$$\frac{\rightarrow l \simeq r \mid T_1 \qquad \Gamma, \underline{s \simeq t} \rightarrow \Delta \mid T_2}{\Gamma, s[r]_p \simeq t \rightarrow \Delta \mid s|_p = l \wedge T_1 \wedge T_2} \qquad \text{if } s|_p \notin \mathcal{X}$$

equality resolution:

$$\frac{\Gamma, \underline{s \simeq t} \rightarrow \Delta \mid T}{\Gamma \rightarrow \Delta \mid s = t \wedge T}$$

where the equations written underlined must belong to the set of selected literals. Here $=$ is interpreted as the syntactic equality relation $\equiv$ when dealing with instances. That is, we forbid those instances of the conclusion that correspond to ground inferences between instances of the premises for which the constraints do not hold.

Our aim is to prove completeness of the following inference system $\mathcal{A}$, which is a modification of $\mathcal{N}$ allowing an arbitrary selection strategy where a single arbitrary literal is selected in each ground clause:

paramodulation right:

$$\frac{\Gamma_1 \rightarrow \underline{l \simeq r} \mid T_1 \qquad \Gamma_2 \rightarrow \underline{s \simeq t} \mid T_2}{\Gamma_1, \Gamma_2 \rightarrow s[r]_p \simeq t \mid s|_p = l \wedge T_1 \wedge T_2} \qquad \text{if } s|_p \notin \mathcal{X}$$

paramodulation left:

$$\frac{\Gamma_1 \to \underline{l \simeq r} \mid T_1 \qquad \Gamma_2, \underline{s \simeq t} \to \Delta \ \mid T_2}{\Gamma_1, \Gamma_2, s[r]_p \simeq t \to \Delta \mid s|_p = l \wedge T_1 \wedge T_2} \qquad \text{if } s|_p \notin \mathcal{X}$$

equality resolution:

$$\frac{\Gamma, \underline{s \simeq t} \to \Delta \mid T}{\Gamma \to \Delta \mid s = t \wedge T}$$

In order to prove the completeness of $\mathcal{A}$ we will proceed as follows. Assume S is a set of constrained clauses that is closed under $\mathcal{A}$. Furthermore, let P be a proof by $\mathcal{N}$ deriving the empty clause from S. Then we will show that if P is non-trivial, i.e., it has more than zero steps, then there exists another proof by $\mathcal{N}$ from S of the empty clause with a smaller number of steps. By induction on this proof transformation process, it follows that the empty clause belongs to S.

Let S be a set of constrained clauses and let $C \mid T$ be a constrained clause that is in the closure of S wrt. $\mathcal{N}$. Then, as usual, the proof by $\mathcal{N}$ of $C \mid T$ from S can be expressed as a tree rooted by $C \mid T$, and whose leaves are in S. Now assume T is satisfiable, and let σ be a ground solution of T. Furthermore, σ can be taken such that its domain contains all variables ocurring in the proof. Therefore we can deal with ground proofs where the constraints are replaced by their solution σ (where a ground substitution σ itself is seen as an equality constraint): by a (ground) *$\mathcal{N}$-proof* P of $C \mid \sigma$ from S we mean a proof tree by $\mathcal{N}$, whose nodes are clauses of the form $D \mid \sigma$, and whose leaves are clauses $D' \mid \sigma$ where $D' \mid T'$ is in S and $\sigma \models T'$. By *steps*(P) we refer to its number of proof steps (or, equivalently, to its number of non-leaf nodes). The following is an example of an $\mathcal{N}$-proof.

Example 53

$$\dfrac{\to c \simeq a \mid x = a \qquad \dfrac{x \simeq a \to b \simeq c \mid x = a}{\to b \simeq c \mid x = a}}{\to b \simeq a \mid x = a}$$

When dealing with $\mathcal{N}$-proofs, we will frequently speak about its rightmost leaf ($x \simeq a \to b \simeq c \mid x = a$ in the example), its rightmost inner node ($\to b \simeq c \mid x = a$), its rightmost step (the inference obtaining $\to b \simeq c \mid x = a$ from $x \simeq a \to b \simeq c \mid x = a$), and its rightmost path (the nodes $x \simeq a \to b \simeq c \mid x = a$, $\to b \simeq c \mid x = a$, $\to b \simeq a \mid x = a$).

An $\mathcal{N}$-proof is called *antecedent elimination of* Γ if its rightmost leaf is of the form $\Gamma \to \Delta \mid \sigma$, its root is $\to \Delta \mid \sigma$, and no node on its rightmost path is obtained by a paramodulation-right step. In the given proofs, the substitution part $\mid \sigma$ of the clauses is omitted in order to improve readability.

6.3 Completeness proof

Lemma 54 *(fusion lemma)* Let P_1 and P_2 be two antecedent elimination $\mathcal{N}$-proofs of Γ_1 and Γ_2 respectively.

Then for an arbitrary Δ, there exists an antecedent elimination $\mathcal{N}$-proof P such that its rightmost leaf is $\Gamma_1, \Gamma_2 \rightarrow \Delta$. Moreover, $steps(P) = steps(P_1) + steps(P_2)$, and every non-rightmost leaf of P is a non-rightmost leaf of P_1 or of P_2.

In the following, the $\mathcal{N}$-proof P built as in the previous lemma will be called the *fusion* of P_1 and P_2.

Lemma 55 (separation lemma) Let P be an antecedent elimination $\mathcal{N}$-proof of Γ_1, Γ_2.

Then, there exist two antecedent elimination $\mathcal{N}$-proofs P_1 and P_2 of Γ_1 and Γ_2 respectively. Moreover $steps(P) = steps(P_1) + steps(P_2)$, and all the non-rightmost leaves of P_1 or of P_2 are non-rightmost leaves of P.

Proof: We give a proof of the fusion lemma that uses the separation lemma, and vice versa, resulting in a combined proof by induction.

1. (Proof of the fusion lemma, assuming the separation lemma is true for $\mathcal{N}$-proofs with a number of steps less than or equal to $steps(P_1) + steps(P_2)$)

If Γ_1 and Γ_2 are empty, then the result is trivial. Otherwise, we analyse which equations are selected in the clause $\Gamma_1, \Gamma_2 \rightarrow \Delta$. Wlog. suppose $\mathcal{N}$ selects an equation e in Γ_1. We take $\Gamma_1 = \Gamma_1', e$. We now apply the separation lemma to P_1 and then we have two $\mathcal{N}$-proofs P_{11} and P_{12} that are antecedent eliminations of Γ_1' and e respectively. Moreover, $steps(P_1) = steps(P_{11}) + steps(P_{12})$ and every non-rightmost leaf of P_{11} or of P_{12} is a non-rightmost leaf of P_1. Now, we consider different cases depending on which kind of $\mathcal{N}$-inference is the rightmost step of P_{12}.

1a. If it is an equality resolution then it is of the form:

$$\frac{\underline{e} \rightarrow \Delta_e}{\rightarrow \Delta_e}$$

It does not matter for us which is Δ_e. Observe that in this case $steps(P_{12}) = 1$, and hence $steps(P_{11}) + steps(P_2) + 1 = steps(P_1) + steps(P_2)$. Then we can apply induction hypothesis to P_{11} and P_2 to obtain an antecedent elimination $\mathcal{N}$-proof P' of $\rightarrow \Delta$ s.t. its rightmost leaf is $\Gamma_1', \Gamma_2 \rightarrow \Delta$. Moreover, $steps(P') = steps(P_{11}) + steps(P_2)$, and every non-rightmost leaf of P' is a non-rightmost leaf of P_{11} or of P_2, and then of P_1 or of P_2. Now we create P by inserting in the rightmost leaf of P' the $\mathcal{N}$-inference:

$$\frac{\Gamma_1', \underline{e}, \Gamma_2 \rightarrow \Delta}{\Gamma_1', \Gamma_2 \rightarrow \Delta}$$

And then P satisfies the desired conditions.

1b. If the rightmost step of P_{12} is a paramodulation left inference, then it is of the form:

$$\frac{\rightarrow \Delta_3 \qquad \underline{e} \rightarrow \Delta_e}{e' \rightarrow \Delta_e}$$

Let P_{123} be the subproof of P_{12} rooted by $\rightarrow \Delta_3$. And let P_{12}' be like P_{12} but where P_{123} and the rightmost leaf are removed. The rightmost leaf of P_{12}' is $e' \rightarrow \Delta_e$, and $steps(P_{12}) = steps(P_{123}) + steps(P_{12}') + 1$. Every non-rightmost leaf of P_{12}' is a non-rightmost leaf of P_{12}, and then of P_1. We have $steps(P_{11}) + steps(P_{12}') + steps(P_2) < steps(P_1) + steps(P_2)$. Then we can apply twice the induction hypothesis to P_{11}, P_{12}' and P_2 to obtain an antecedent elimination $\mathcal{N}$-proof P' of $\rightarrow \Delta$ s.t. its rightmost leaf is $\Gamma_1', e', \Gamma_2 \rightarrow \Delta$. Moreover, $steps(P') = steps(P_{11}) + steps(P_{12}') + steps(P_2)$, and every non-rightmost leaf of P' is a non-rightmost leaf of P_{11} or of P_{12}' or of P_2, and then of P_1 or of P_2. Now we create P by inserting in the rightmost leaf of P' the $\mathcal{N}$-inference

$$\frac{\rightarrow \Delta_3 \qquad \Gamma_1', \underline{e}, \Gamma_2 \rightarrow \Delta}{\Gamma_1', e', \Gamma_2 \rightarrow \Delta}$$

and inserting P_{123} above $\rightarrow \Delta_3$. Then, P satisfies then the desired conditions.

2. (Proof of the separation lemma, assuming the fusion lemma is true for pairs of $\mathcal{N}$-proofs s.t. the sum of their steps is less than $steps(P)$)

If Γ_1 or Γ_2 is empty then the result is trivial. Otherwise suppose wlog. that the selected equation e in $\Gamma_1, \Gamma_2 \rightarrow \Delta$ is in Γ_1. We take $\Gamma_1 = \Gamma_1', e$. Then there are two cases to be considered, depending on what kind of inference is the rightmost step of P.

2a. If it is an equality resolution, then it is of the form:

$$\frac{\Gamma_1', \underline{e}, \Gamma_2 \rightarrow \Delta}{\Gamma_1', \Gamma_2 \rightarrow \Delta}$$

Let P' be like P but where this step is removed. We have $steps(P') + 1 = steps(P)$. Therefore we can apply induction hypothesis to P' and, then, there exist P_1' and P_2 s.t. they are antecedent elimination of Γ_1' and Γ_2 respectively. Moreover, their non-rightmost leaves are non-rightmost leaves of P', and then of P, and $steps(P_1') + steps(P_2) = steps(P')$. Consider now the following $\mathcal{N}$-inference:

$$\frac{\underline{e} \to \Delta}{\to \Delta}$$

It can be seen as an $\mathcal{N}$-proof with only one step. It is an antecedent elimination proof of e, that we name P_e. Now observe that, since Γ_2 is not empty, we have $steps(P_2) \geq 1$, and then $steps(P_1') + steps(P_e) < steps(P)$. Therefore we can apply, inductively, the fusion lemma to P_1' and P_e, obtaining an antecedent elimination $\mathcal{N}$-proof P_1 of Γ_1', e, that is Γ_1. Every non-rightmost leaf of P_1 is a non-rightmost leaf of P_1', and then, of P; and $steps(P_1) = steps(P_1') + steps(P_e) = steps(P_1') + 1$. Therefore $steps(P) = steps(P_1) + steps(P_2)$.

2b. If the rightmost step of P is a paramodulation left inference, then, it is of the form:

$$\frac{\to \Delta_3 \qquad \Gamma_1', \underline{e}, \Gamma_2 \to \Delta}{\Gamma_1', e', \Gamma_2 \to \Delta}$$

Let P_3 be the subproof of P rooted by $\to \Delta_3$. Let P' be like P but where P_3 and the rightmost leaf are removed. The root of P' is $\Gamma_1', e', \Gamma_2 \to \Delta$, and $steps(P) = steps(P') + steps(P_3) + 1$. We can apply induction hypothesis on P', two times, and obtain three antecedent elimination $\mathcal{N}$-proofs P_1', $P_{e'}$ and P_2 of Γ_1', e' and Γ_2 respectively, s.t. all their non-rightmost leaves are non-rightmost leaves of P', and then of P. Moreover, $steps(P') = steps(P_1') + steps(P_{e'}) + steps(P_2)$. Now we create P_e by inserting in the rightmost leaf of $P_{e'}$ the inference

$$\frac{\to \Delta_3 \qquad \underline{e} \to \Delta}{e' \to \Delta}$$

and inserting P_3 above $\to \Delta_3$. Observe that every non-rightmost leaf of P_e is a non-rightmost leaf of P, since it is a non-rightmost leaf of $P_{e'}$ or a leaf of P_3. We have $steps(P) = steps(P_1') + steps(P_e) + steps(P_2)$. Now, since Γ_2 is not empty, we have $steps(P_2) \geq 1$, and then $steps(P_1') + steps(P_e) < steps(P)$. Therefore, we can apply the fusion lemma to P_1' and P_e, obtaining P_1, an antecedent elimination $\mathcal{N}$-proof of Γ_1', e, that is Γ_1. Moreover, their non-rightmost leaves are non-rightmost leaves of P_1' or of P_e, and then of P. Furthermore $steps(P_1) = steps(P_1') + steps(P_e)$. Therefore $steps(P) = steps(P_1) + steps(P_2)$. □

Lemma 56 Let P be a $\mathcal{N}$-proof that is antecedent elimination of Γ. Then, for all Δ, there exists an antecedent elimination $\mathcal{N}$-proof P' such that its rightmost leaf is $\Gamma \to \Delta$. Moreover, $steps(P) = steps(P')$.

Proof: By applying the fusion lemma to P and the empty proof we obtain the desired P'. □

Lemma 57 (general fusion lemma) Let P_1 be an antecedent elimination $\mathcal{N}$-proof of Γ_1. Let P_2 be an $\mathcal{N}$-proof of $\rightarrow \Delta$ with rightmost leaf $\Gamma_2 \rightarrow \Delta_2$ (therefore a subtree in its rightmost path is antecedent elimination, i.e. the antecedent Γ_2 is eliminated in P_2, but after this elimination, some paramodulation right inferences can be made on the rightmost path).

Then there exists an $\mathcal{N}$-proof P of $\rightarrow \Delta$ such that its rightmost leaf is $\Gamma_1, \Gamma_2 \rightarrow \Delta_2$. Moreover $steps(P) = steps(P_1) + steps(P_2)$, and every non-rightmost leaf of P is a non-rightmost leaf of P_1 or of P_2.

Proof: Let P_{21} be the antecedent elimination $\mathcal{N}$-proof on the rightmost path of P_2. Let P_2' be like P_2 but where P_{21} is removed. Note that the rightmost leaf of P_2' is $\rightarrow \Delta_2$, and P_2 can be obtained by inserting P_{21} on the rightmost path of P_2'. Moreover $steps(P_2) = steps(P_2') + steps(P_{21})$. Now, since P_1 and P_{21} are antecedent eliminations, by the fusion lemma we have there exists P' s.t. it is an antededent elimination of $\Gamma_1 \cup \Gamma_2$, $steps(P') = steps(P_1) + steps(P_{21})$ and all the non-rightmost leaves of P' are non-rightmost leaves of P_1 or of P_{21} (and then of P_2). Moreover, the root of P' can be chosen to be $\rightarrow \Delta_2$, maintaining these conditions. Then the rightmost leaf of P' is $\Gamma_1, \Gamma_2 \rightarrow \Delta_2$. Now let P be the $\mathcal{N}$-proof formed by inserting P' on the rightmost leaf of P_2'. Then P satisfies all the conditions we are looking for. □

Lemma 58 (general separation lemma) Let P be an $\mathcal{N}$-proof of $\rightarrow \Delta$ such that its rightmost leaf is $\Gamma_1, \Gamma_2 \rightarrow \Delta'$.

Then, there exist two $\mathcal{N}$-proofs P_1 and P_2 such that P_1 is an antecedent elimination proof of Γ_1 and P_2 is an $\mathcal{N}$-proof of $\rightarrow \Delta$ with rightmost leaf $\Gamma_2 \rightarrow \Delta'$. Moreover $steps(P) = steps(P_1) + steps(P_2)$, and all the non-rightmost leaves of P_1 or of P_2 are non-rightmost leaves of P.

Proof: Let P_{21} be the subtree in the rightmost path of P that is an antecedent elimination of Γ_1, Γ_2. We have that P_{21} is a proof of $\rightarrow \Delta'$. Let P_2' be like P but removing P_{21} from it. We have that the rightmost leaf of P_2' is $\rightarrow \Delta'$ and $steps(P) = steps(P_2') + steps(P_{21})$. Now we apply the separation lemma to P_{21} and we obtain a proof P_1 that is an antecedent elimination of Γ_1, and a proof P_{21}' that is an antecedent elimination of Γ_2 and, by lemma 56, we can assume that its rightmost leaf is $\Gamma_2 \rightarrow \Delta'$. Now we define P_2 to be like P_2' but inserting P_{21}' in its rightmost leaf. Then P_1 and P_2 satisfy the required conditions. □

Lemma 59 Let S be a set of clauses closed under $\mathcal{A}$, and let P be an $\mathcal{N}$-proof of $\rightarrow \Delta$ from S.

Then there exists an antecedent elimination $\mathcal{N}$-proof P' of $\rightarrow \Delta$ from S whose rightmost leaf is of the form $\Gamma \rightarrow \Delta$, where $steps(P') \leq steps(P)$ and, if Γ is non-empty, then Δ is non-empty and its equation is the selected one in $\Gamma \rightarrow \Delta$ by $\mathcal{A}$.

Proof: We will proceed by induction on $steps(P)$. Let the rightmost leaf of P be of the form $\Gamma_1 \rightarrow \Delta_1$. There are several cases to be considered:

0. If Γ_1 and Δ_1 are both empty, then P has no steps and P' can be P itself.

1. One of the selected equations of $\Gamma_1 \rightarrow \Delta_1$ by $\mathcal{A}$ is Δ_1. We consider two possibilities depending on whether some paramodulation right inference is made or not on the rightmost path of P.

1a. In the case that no paramodulation right inference is made on the rightmost path of P, we have that Δ and Δ_1 coincide and P is antecedent elimination. Therefore the P' we are looking for is directly P.

1b. Suppose now there are some paramodulation right inferences on the rightmost path of P. Then the highest one is of the form:

$$\frac{\rightarrow \Delta_2 \qquad \rightarrow \Delta_1}{\rightarrow \Delta_3}$$

Let P_1 be the subproof rooted by $\rightarrow \Delta_1$. It is an antecedent elimination $\mathcal{N}$-proof. Let P_2 be the subproof rooted by $\rightarrow \Delta_2$. Let P_3 be like P, but where P_1 and P_2 are removed (the rightmost leaf of P is $\rightarrow \Delta_3$). Then we have $steps(P) = steps(P_1) + steps(P_2) + steps(P_3) + 1$. By applying induction hypothesis to P_2, we obtain an antecedent elimination $\mathcal{N}$-proof P_2' of $\rightarrow \Delta_2$ from S s.t. its rightmost leaf is of the form $\Gamma_2 \rightarrow \Delta_2$, where Δ_2 is selected by $\mathcal{A}$, and $steps(P_2') \leq steps(P_2)$. If we apply the fusion lemma to P_2' and P_1, we obtain an antecedent elimination $\mathcal{N}$-proof P_4 of $\rightarrow \Delta_3$, where its rightmost leaf is $\Gamma_1, \Gamma_2 \rightarrow \Delta_3$, and $steps(P_4) = steps(P_2') + steps(P_1) \leq steps(P_2) + steps(P_1)$. Now let P_5 be the $\mathcal{N}$-proof formed by P_3, and where above its rightmost leaf we insert P_4. We have $\rightarrow \Delta$ in the root of P_5, and $steps(P_5) = steps(P_3) + steps(P_4) \leq steps(P_3) + steps(P_2) + steps(P_1) < steps(P)$ and the rightmost leaf of P_5 is $\Gamma_2, \Gamma_1 \rightarrow \Delta_3$. Moreover, every non-rightmost leaf of P_5 is from S: all the non-rightmost leaves of P_3 are from S, and, since P_4 is the fusion of P_2' and P_1, then, its non-rightmost leaves are from S too. But also, the rightmost leaf of P_5 is from S, since the next inference is a $\mathcal{A}$-inference from S:

$$\frac{\Gamma_2 \rightarrow \underline{\underline{\Delta_2}} \qquad \Gamma_1 \rightarrow \underline{\Delta_1}}{\Gamma_2, \Gamma_1 \rightarrow \Delta_3}$$

Then the $\mathcal{N}$-proof P' we are looking for is the one obtained by applying the induction hypothesis to P_5.

2. Assume now that Γ_1 is of the form Γ_{11}, e and e is a selected equation of $\Gamma_1 \rightarrow \Delta_1$ by $\mathcal{A}$. We apply the general separation lemma to P and we obtain two $\mathcal{N}$-proofs P_1 and P_2 s.t. P_1 is an antecedent elimination proof of e and P_2 is an $\mathcal{N}$-proof of $\rightarrow \Delta$

with rightmost leaf $\Gamma_{11} \rightarrow \Delta_1$. Moreover $steps(P) = steps(P_1) + steps(P_2)$, and all the non-rightmost leaves of P_1 or of P_2 are non-rightmost leaves of P.

We distinguish two cases depending on wether the rightmost step of P_1 is an equality resolution or a paramodulation left inference on e:

2a. If it is an equality resolution step, it is of the form:

$$\frac{e \rightarrow}{\rightarrow}$$

Then P_1 consists in only this step, and we have the $\mathcal{A}$-inference

$$\frac{\underline{e}, \Gamma_{11} \rightarrow \Delta_1}{\Gamma_{11} \rightarrow \Delta_1}$$

and hence $\Gamma_{11} \rightarrow \Delta_1$ is in S. Therefore the $\mathcal{N}$-proof P' we are looking for is the one obtained by applying the induction hypothesis to P_2.

2b. If the rightmost step of P_1 is a paramodulation left inference, it is of the form:

$$\frac{\rightarrow \Delta_3 \qquad e \rightarrow}{e' \rightarrow}$$

Let P_3 be the subproof of P_1 rooted by $\rightarrow \Delta_3$. Let P_1' be like P_1 but where the subproof P_3 and the rightmost leaf are removed (the rightmost leaf of P_1' is $e' \rightarrow$). Note that all the non-rightmost leaves of P_1' are clauses from S. We have $steps(P_1) = steps(P_3) + steps(P_1') + 1$. We apply induction hypothesis to P_3, and we obtain an antecedent elimination $\mathcal{N}$-proof P_3' of $\rightarrow \Delta_3$ from S s.t. its rightmost leaf is of the form $\Gamma_3 \rightarrow \Delta_3$, where Δ_3 is selected by $\mathcal{A}$. Now, we apply the fusion lemma to P_3' and P_1', and we get an $\mathcal{N}$-proof P_4 s.t. is antecedent elimination of Γ_3, e', and $steps(P_4) = steps(P_3') + steps(P_1')$. Applying now the general fusion lemma to P_4 and P_2 we obtain an $\mathcal{N}$-proof P_5 s.t. its rightmost leaf is $\Gamma_{11}, e', \Gamma_3 \rightarrow \Delta_1$, $steps(P_5) = steps(P_3') + steps(P_1') + steps(P_2) < steps(P)$ and all the non-rightmost leaves of P_5 are non-rightmost leaves of P_3' or of P_1' or of P_2 and, hence, from S. But also the rightmost leaf of P_5 is a clause from S, since the next inference is an $\mathcal{A}$-inference from S:

$$\frac{\Gamma_3 \rightarrow \underline{\Delta_3} \qquad \Gamma_{11}, \underline{e} \rightarrow \Delta_1}{\Gamma_3, \Gamma_{11}, e' \rightarrow \Delta_1}$$

Therefore the $\mathcal{N}$-proof P' we are looking for is the one obtained by applying the induction hypothesis to P_5. □

Theorem 60 (Completeness theorem) Let S_0 be an unconstrained set of clauses, and let S be the closure of S_0 under $\mathcal{A}$. If S_0 is unsatisfiable, then $\square \in S$.

Proof: By completeness of $\mathcal{N}$ there is an $\mathcal{N}$-proof of $\Box$ from S. Then, applying lemma 59 to the case where Δ is empty gives us a trivial $\mathcal{N}$-proof of $\Box$, since Δ can not be selected by $\mathcal{A}$ and, hence, Γ must be empty. Consequently $\Box \in S$. □

6.4 Application to paramodulation with non-monotonic orderings

Here we prove the refutational completeness of the inference system presented in [BGNR99] with added equality constraint inheritance, for the case of Horn clauses.

Definition 61 A *west-ordering* is a well-founded ordering on $T(\mathcal{F})$ that fulfils the subterm property and that is total on $T(\mathcal{F})$ (it is called *west* after well-founded, subterm and total).

For a given a west ordering $\succ$, the inference system $\mathcal{J}$ for Horn clauses with equality is (selected equations are written underlined):

paramodulation right:

$$\frac{\rightarrow l \simeq r \mid T_1 \qquad \rightarrow s \simeq t \mid T_2}{\rightarrow s[r]_p \simeq t \mid s|_p = l \wedge T_1 \wedge T_2}$$

if $s|_p \notin \mathcal{X}$ and $l\sigma \succ r\sigma$ for some ground substitution σ which is a solution of $s|_p = l \wedge T_1 \wedge T_2$.

paramodulation left:

$$\frac{\rightarrow l \simeq r \mid T_1 \qquad \Gamma, \underline{s \simeq t} \rightarrow \Delta \ \mid T_2}{\Gamma, s[r]_p \simeq t \rightarrow \Delta \mid s|_p = l \wedge T_1 \wedge T_2}$$

if $s|_p \notin \mathcal{X}$ and $l\sigma \succ r\sigma$ for some ground substitution σ which is a solution of $s|_p = l \wedge T_1 \wedge T_2$.

equality resolution:

$$\frac{\Gamma, \underline{s \simeq t} \rightarrow \Delta \mid T}{\Gamma \rightarrow \Delta \mid s = t \wedge T}$$

The following theorem can be proved according to the standard model generation techniques explained in Chapter 3.

Theorem 62 The inference system $\mathcal{J}$ with equality constraint inheritance is refutationally complete for first-order Horn clauses.

We can use our result of theorem 60 for proving the completeness of a modification $\mathcal{JA}$ of the inference system $\mathcal{J}$. In $\mathcal{JA}$ any strategy selecting a single (positive or negative) equation in each clause is allowed. Note that this new result is not an immediate consequence of theorem 60. The reason is that for right and left paramodulation there is an ordering restriction $l\sigma \succ r\sigma$, and, for explanatory reasons, we did not consider this kind of restrictions in the definition of $\mathcal{N}$. But in the proof transformation from $\mathcal{N}$ to $\mathcal{A}$ one uses a given substitution σ that satisfies all the constraints. This also holds in the case of a proof tree by $\mathcal{J}$ and, moreover, this σ satisfies all the required restrictions $l\sigma \succ r\sigma$ appearing in it. Hence the transformation process works exactly in the same way.

Theorem 63 The inference system $\mathcal{JA}$ with equality constraint inheritance is refutationally complete for first-order Horn clauses.

6.5 Conclusions

In this chapter we have shown that if a paramodulation-based inference system is complete with a concrete strategy with eager selection of negative equations and, moreover, it is compatible with equality constraint inheritance, then it is complete with arbitrary selection strategies.

Therefore we have generalized the result in [Lyn97] about refutation completeness of arbitrary selection strategies for superposition. Moreover, the generality of our proof transformation method allows us to obtain directly the completeness of arbitrary selection strategies for other inference systems. We have shown that the results of Chapter 4 for paramodulation with non-monotonic orderings are compatible with equality constraint inheritance, thus further restricting the search space and allowing arbitrary selection strategies.

We have also generalized, in a sense, the result in [dN96] for Horn clauses *without* equality, about completeness of resolution with an arbitrary selection of one single literal in each clause.

In [BG94b] standard methods for proving compatibility with redundancy elimination techniques are given, by which, roughly, a clause is redundant if it follows from smaller clauses. These notions are not applicable to our proof transformation technique. But this is not surprising, since by these standard techniques all tautologies are redundant, which is not the case here. Some kind of tautologies have to be kept in order to preserve completeness in the case of arbitrary selection strategies, as shown by the counter example from [Lyn97] that was given in Example 47 of Chapter 4.

Chapter 7

Paramodulation with Built-in Abelian Groups

In this chapter, a new technique is presented for superposition with first-order clauses with built-in abelian groups (AG). Compared with previous approaches, it is simpler, and AG-unification is used instead of the computationally more expensive unification modulo associativity and commutativity. Furthermore, no inferences with the AG axioms or abstraction rules are needed; in this sense this is the first approach where AG is completely built in.

7.1 Introduction

As explained in previous chapters, it is crucial for the performance of a deduction system that it incorporates specialized techniques to work efficiently with certain theories, since a naïve handling of their axioms leads to an explosion of the search space. Perhaps the most important example of this is *paramodulation*, an inference rule specialized to equality in the context of resolution-based systems. Essentially, paramodulation builds the congruence axioms in inside the inference system.

Another well-investigated line of research concerns building-in equational theories inside paramodulation and resolution-based systems. Some axioms generate many slightly different permuted versions of clauses, and for efficiency reasons it is many times better to treat all these clauses together as a single one representing the whole class, i.e., to work with a *built-in* equational theory E, and performing deduction with specialized E-matching and E-unification algorithms.

Early results on paramodulation *modulo* E were given by Plotkin [Plo72], Slagle [Sla74] and Lankford and Ballantine [LB77] and *extended* E*-rewriting* was defined by Peterson and Stickel [PS81]. Special attention has always been devoted to the case where E includes axioms of associativity and commutativity (AC), which occur very frequently in practical applications, and are well-suited for being built in due to their permutative nature. Note that in general there is no unique most general E-unifier for a given E-unification problem, and that new variables may appear: for example, if f is an AC-symbol, then $f(x,a)$ and $f(y,b)$ have the two AC-unifiers $\sigma_1 = \{x \mapsto b, y \mapsto a\}$ and $\sigma_2 = \{x \mapsto f(b,z), y \mapsto f(a,z)\}$.

Resolution modulo E is relatively simple: there exist general completeness results for resolution with constraints, which essentially say that completeness is preserved when unification is replaced by E-unification. The reason is that resolution inferences, which take place at the atom level, do not interfere with the built-in equational theories, which affect only the term level, and hence *lifting* can still be done (see [NR01]). Unfortunately, for paramodulation this is far from true, and for each built-in theory special inference rules have to be designed and their completeness proved.

Paramodulation with built-in abelian groups (AG) has been investigated by many authors [Che86, Zha93, Mar94, Mar96, GW96, Wal98, Wal99, Stu98]. This is not surprising since abelian groups are of course ubiquitous in many applications of (semi-)automated reasoning. But building-in AG is also attractive for at least two more reasons.

On the one hand, due to the fact that diophantine equation solving is easier in the integers than in the natural numbers, AG unification is easier than AC and AC1 (i.e., abelian monoid) unification. If all free symbols are constants, then there is one single most general AG unifier and the decision problem is polynomial, whereas for AC and AC1 the decision problems are NP-complete, and for AC there are exponentially many unifiers. Although with arbitrary free symbols the decision problem is NP-complete in all three cases, AG unification behaves better in practice. Also the number of unifiers is usually much smaller and not doubly exponential as for AC (see [BS93, BS01] for surveys on these results).

Another aspect that makes building-in AG attractive is called *symmetrization* (e.g., by Le Chenadec in [Che86]): modulo abelian groups $(+,-,0)$, every ground equation can be written as $u + \ldots + u \simeq t$, where the summand u is greater (w.r.t. the given term ordering $\succ$) than the summands in t. As we will see, this allows one to restrict inferences to this maximal summand and to avoid the prolific inferences with extended equations that appear in the AC case.

Symmetrisation is also exploited in Marché's framework for Knuth-Bendix completion of unit equations with built-in theories (ranging from AC to commutative rings) [Mar94, Mar96]. His completion procedure decides the ground word problem modulo AG by building a finite convergent rewrite system. However, his procedure is not refutation complete for equations with variables: in many cases it fails since it cannot handle symmetrisation at the non-ground level.

Full first-order clauses are considered by Ganzinger and Waldmann in [GW96, Wal97], where symmetrisation is also central. This work focusses not on AG, but on the more general theory of cancellative abelian monoids. It applies AC1 unification and *abstraction* rules, which, roughly, turn clauses like $C \vee f(s) \simeq t$ into $C \vee x \not\simeq s \vee f(x) \simeq t$, where x is a new variable; this of course increases the number of possible inferences on f. By specialising to torsion-free divisible abelian groups, AC-unification and inferences into variables can be avoided, but abstraction remains necessary [Wal98, Wal99].

In Stuber's work on paramodulation for abelian groups represented as integer modules [Stu98], symmetrization is again crucial, but AG unification is not applied. Instead, AC unification is used, and hence paramodulation inferences with the AG axioms on the remaining clauses are needed. For example, refuting a clause like $f(-b + x + a) \not\simeq f(0)$ requires inferences with the AG axioms, instead of directly finding the contradictory instance $b - a$ for x by AG-unification. Technically, even for the ground case, his inference rules and proofs are rather involved. In Stuber's PhD. Thesis [Stu99], proofs for the ground case are given in a uniform framework for AG and several other commutative theories.

Here we apply a variant of Bachmair and Ganzinger's model generation technique (see Chapter 3 and [BG94b]), where the model is defined by rewriting, modulo associativity and commutativity of $+$, with the well-known convergent rewrite system R_{AG} for AG, plus a set of ground rewrite rules R that consists of symmetrised rules $u + \ldots^{n)} + u \to t$ and their *inverse* version $-u \to u + \ldots^{n-1)} + u - t$. Hence $\succ$ has to be an AC-compatible reduction ordering orienting these rules, which can be fulfilled by simple general-purpose orderings like RPO (this was already mentioned by Marché). This gives relatively simple completeness proofs for full first-order ground clauses. From our results it is easy to obtain a decision procedure for the satisfiability of arbitrary sets of ground clauses modulo AG.

For completely building-in AG at the non-ground level, and hence avoiding all inferences with the AG axioms by applying AG-unification, the main problem is: how to lift, to inferences on non-ground clauses C, the rewrite steps with $R \cup R_{AG}$ on ground instances $C\sigma$? The steps with R indeed become inferences, but for the steps with R_{AG} this is precisely what we want to avoid. The key ideas to our solution are roughly as follows. We keep non-ground clauses C fully simplified w.r.t. R_{AG} (which is a cheap and useful simplification anyway). Furthermore, in the completeness proofs we consider instances with reduced[1] substitutions σ (extending some ideas from the *basic* superposition approach [NR95, BGLS95]). Some steps with R_{AG} may then still be needed in $C\sigma$ at the frontier between C and σ. But a careful analysis of these steps reveals that they can be covered by considering inferences with AG-unification on adequate subterms.

[1] In the preliminary version of this work, [GN00], we used a different notion of irreducibility. In this chapter the definitions are more intuitive and we obtain shorter and simpler proofs.

Our AG-superposition inference rules have strong ordering restrictions implying that inferences only need to involve the maximal summands of the clause. This generalises standard superposition: summands play the role of terms.

Due to the simplicity and restrictiveness of our inference system, its compatibility with redundancy notions and constraints, and the fact that standard term orderings like RPO can be used, we believe that our techniques will become the method of choice for practice. On the theoretical side, we expect that our techniques and results will also lead to logic-based decidability and complexity results, along the lines of, e.g., [BG96, Nie96, Nie98, GMV99, GdN99, Wal99].

This chapter is structured as follows. After the basic notions and notation given in Section 7.2, in Section 7.3 we introduce our techniques for the simple case of ground Horn clauses, and show that this can be used for deciding the satisfiability of set of general ground clauses modulo AG. Sections 7.4, 7.5 and 7.6 are the core of this chapter. There, the ideas of the ground case are extended to Horn clauses with variables. This is again extended to general clauses with variables in Section 7.7. Finally, in Section 7.8 we give conclusions and mention some optimizations.

7.2 Basic notions

We use the standard notation and terminology of Chapter 2. Furthermore, we use the following terminology for positions p and q in a term t: we say that p is (strictly) *below* q if q is a (proper) prefix of p, and then q is (strictly) *above* p. Similarly, p is *beside* q (or *disjoint* with q) if no one is a prefix of the other. We also say that p is *below* a function symbol f in t if $t|_q$ is headed by f for some q above p, and then p is *immediately below* f if p is $q.i$ for some natural number i.

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

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

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 $=_{AC}$ and $=_{AG}$ we denote the corresponding congruences on terms. In this chapter, rewriting with a set of rules R is always considered *modulo* AC, that is, when writing $\rightarrow_R$, we mean the relation $=_{AC} \rightarrow_R =_{AC}$. We denote by $nf_R(t)$ the normal form of a term t by rewriting with R, and instead of writing $nf_{R_{AG}}(t)$ we sometimes write $AG\text{-}nf(t)$. By *free* function symbols we mean symbols different from $+$, $-$ and 0.

We sometimes write terms with $+$ in infix notation, without parenthesis. For example, $+(a, +(+(b, c), d))$ is written $a+b+c+d$. But we remark that this is only done at the notation level (and terms are not considered to be in flattened form as in other approaches). A *summand* is a term u headed by a free symbol. We write nu as a shorthand for $u+\ldots^{n)}+u$, and $-nu$ as a shorthand for $(-u)+\ldots^{n)}+(-u)$, and $a-b$ as a shorthand for $a+(-b)$.

In this chapter, we assume that $\succ$ is a well-founded strict ordering on terms satisfying:

1. $\succ$ is *AC-compatible*, that is, $s' =_{AC} s \succ t =_{AC} t'$ implies $s' \succ t'$

2. $\succ$ is total up to $=_{AC}$ on the set of ground terms, that is, for all ground terms s and t, we have $s \succ t$ or $t \succ s$ or $s =_{AC} t$.

3. $\succ$ orients all rules of R_{AG}, that is, $l \succ r$ for every rule $l \to r$ of R_{AG}

4. $\succ$ is monotonic on ground terms, that is, for all ground terms s, t and u, we have $u[s]_p \succ u[t]_p$ whenever $s \succ t$

One way to build such an ordering $\succ$ is to simply use the recursive path ordering (RPO) [Der82], applied to the terms to be compared in *flattened* form w.r.t. $+$. This flattening consists of removing all operators $+$ that are immediately below another $+$. For example, $+(\,a,\ +(\,f(\,+(a, +(b, c))\,),\ c)\,)$ becomes $+(a, f(+(a, b, c)), c)$, which can also be written $a+f(a+b+c)+c$. Note that in the flattened form of a term t, denoted by $\flat(t)$, different occurrences of $+$ can have different arities (but all greater than 1).

Lemma 64 Let $\succ$ be defined by: $s \succ t$ if $\flat(s) \succ_{rpo} \flat(t)$, where $\succ_{rpo}$ is an RPO with a total precedence $\succ_{\mathcal{F}}$ such that $f \succ_{\mathcal{F}} - \succ_{\mathcal{F}} + \succ_{\mathcal{F}} 0$ for all free symbols f and where all symbols have a lexicographic status, except $+$, whose status is multiset. Then $\succ$ fulfils the aforementioned requirements.

Definition 65 A ground equation $nu \simeq n_1v_1 + \ldots + n_kv_k$ in normal form w.r.t. R_{AG} is said to be in *reductive form* if $n > 0$, the n_i are non-zero integers, and u and the v_i are summands with $u \succ v_i$. The (logically equivalent w.r.t. AG-models) *inverse reductive form* of this equation is $-u \simeq (n-1)u - n_1v_1 - \ldots - n_kv_k$.

For every equation $s \simeq t$, its reductive form can be obtained by normalising $s + (-t) \simeq 0$ w.r.t. R_{AG} into $n_1u_1 + \ldots + n_ku_k \simeq 0$ where, say, u_1 is the maximal summand, and then, if n_1 is positive, the reductive form is $n_1u_1 \simeq -n_2u_2 - \ldots - n_ku_k$; otherwise, it is $-n_1u_1 \simeq n_2u_2 + \ldots + n_ku_k$. Note that the unary minus operator $-$ is overloaded in our notation since it is applied as well to coefficients (but remember that coefficients are not part of our logical language but just a shorthand in our notation).

Example 66 If $a \succ b \succ c$ then the equation $(-a) + c + 0 + (-(-c)) + (-b) \simeq (-c) + a + b + 0$ is equivalent to $(-a) + (-a) + c + c + c + (-b) + (-b) \simeq 0$, written shortly $-2a + 3c - 2b \simeq 0$, and becomes in reductive form $2a \simeq 3c - 2b$, and in inverse reductive form $-a \simeq a - 3c + 2b$. □

Example 67 Equations in reductive form can be adequately used as terminating rewrite rules. Assume we have $a \succ b \succ c$ and the equation (in reductive form) $3a \simeq -b + c$. It can be applied either as it is, or in its inverse form $-a \simeq 2a + b - c$.

For example, $4a$ is AG-equivalent by this equation to $-2a - 2b + 2c$. Let us prove it by rewriting both terms into their respective normal forms. On the one hand, by simply applying the equation to three of its four a's, $4a$ rewrites into the normal form $a - b + c$. On the other hand, by applying the inverse form, $-2a - 2b + 2c$ rewrites into $-a - 2b + 2c + 2a + b - c$ which simplifies with R_{AG} into $a - b + c$.

Note that normal forms w.r.t. both ways of rewriting with such equations $nu \simeq v$ will always have a positive number of u's between 0 and $n - 1$, and that the inverse kind of steps is not needed if $n = 1$. The two ground inference rules of AG-superposition that are given below in fact correspond to these two ways of rewriting. □

7.3 Ground Horn Case

Here we first introduce part of our techniques on the simple subcase of ground Horn clauses. We assume all equations in clauses to be eagerly maintained in reductive form, and moreover we assume negative equations $0 \not\simeq 0$ to be removed eagerly from all clauses.

Definition 68 The inference rules for ground AG-superposition are as follows:

direct AG-superposition:

$$\frac{C \vee nu \simeq r \qquad D[nu]_p}{C \vee D[r]_p}$$

inverse AG-superposition:

$$\frac{C \vee nu \simeq r \qquad D[-u]_p}{C \vee D[(n-1)u - r]_p} \quad \text{if } n > 1$$

where $D|_p$ denotes a subterm of D modulo AC, that is, each $D'|_q$ is such a subterm if $D =_{AC} D'$.

The ordering restrictions of AG-superposition are such that inferences are needed only if they take place *with* the strictly maximal summand and *on* a maximal summand (that is strictly maximal if it occurs in a positive equation), that is, denoting by $s \succ C$ the fact that $s \succ t$ for every summand t occuring in C, these inferences are needed only if:

1. $u \succ C$ (and remind that, by expression in reductive form, also $u \succ r$)

2. $s \succ D'$ whenever D is $D' \vee ms \simeq t$ (in reductive form) with $D|_p$ in ms

3. $s \succeq D'$ whenever D is $D' \vee ms \not\simeq t$ (in reductive form) with $D|_p$ in ms

Note that hence inverse AG-superposition is needed only on proper subterms of summmands s since in an (in)equation in reductive form the term $-u$ cannot occur elsewhere.

7.3.1 Completeness for the ground Horn case

The following definition follow the same lines as the model generation method explained in Chapter 3. We now use multiset extensions for lifting the ordering $\succ$ on terms to orderings on ground equations (in reductive form) and clauses in the usual way.

Definition 69 Let C be a ground clause, and let $emul(s \simeq t)$ be $\{s, t\}$ if $s \simeq t$ is a positive equation in C, and $\{s, s, t, t\}$ if it is negative. Then we define the ordering $\succ_e$ on (occurrences of) ground equations in a clause by $e \succ_e e'$ if $emul(e) \succ_{mul} emul(e')$. Similarly, $\succ_c$ on ground clauses is defined $C \succ_c D$ if $mse(C) \ \ (\succ_{mul})_{mul} \ \ mse(D)$, where $mse(C)$ is the multiset of all $emul(e)$ for ocurrences e of equations in C.

Lemma 70 Let C and D be ground clauses. If D is the reductive form of C then $C \succeq_c D$.

Proof: Let u be the maximal summand of an equation $s \simeq t$ occurring (positively or negatively) in C. If u does not occur in the reductive form of $s \simeq t$, i.e., it has been cancelled out, then the reductive form is smaller. Otherwise the reductive form of $s \simeq t$ is of the form $nu \simeq r$ where $u \succ r$. If $-u$ occurs in $s \simeq t$ then again $nu \simeq r$ is smaller. Otherwise $s \simeq t$ is of the form $nu + s' \simeq t$ and $nu \simeq r$ is smaller (if s' is non-empty) or equal (if s' is empty). □

We now show how to construct a model for sets S of ground Horn clauses closed under ground AG-superposition and where $\square \notin S$ (note that this implies the refutation completeness of ground AG-superposition). As usual (see [BG94b]), in order to construct the model we will generate a set of rewrite rules R_S by induction on $\succ_c$. But here the model will contain as well the rules of R_{AG}, and, as said, all rules will be applied modulo AC.

Definition 71 Let S be a set of ground Horn clauses in reductive form, and let C be a clause in S of the form $C \vee nu \simeq r$. Then C *generates* the rule $nu \rightarrow r$ if the following three conditions are satisfied:

1. $(R_C \cup AG)^* \not\models C$

2. $u \succ r$ and $u \succ C$

3. nu is irreducible by R_C

where R_C is the set of rules generated by clauses of S smaller than C w.r.t. $\succ_c$. Furthermore, if C generates $nu \rightarrow r$ with $n > 1$, in addition C generates its inverse form $-u \rightarrow (n-1)u - r$. The set of all rules generated by clauses in S is denoted by R_S.

We now state an essential result: $R_S \cup R_{AG}$ is convergent modulo AC.

Lemma 72 Let S be a set of ground Horn clauses in reductive form. $R_S \cup R_{AG}$ is terminating and confluent modulo AC on ground terms.

Proof: All rules in $R_S \cup R_{AG}$ are oriented w.r.t. $\succ$, and hence $R_S \cup R_{AG}$ is terminating for rewriting modulo AC, since $\succ$ is AC-compatible, well-founded, and monotonic on ground terms. Confluence is a consequence of the following facts. By construction of R_S, for all ground rules $l \rightarrow r$ in R_S, the term l is irreducible by the ground rules in $R_S \setminus \{l \rightarrow r\}$. Furthermore, R_{AG} is well-known to be confluent. Finally, the (extended) critical pairs between R_{AG} and R_S are easily shown to be joinable (this is a bit long and tedious, so we omit this part here). □

Theorem 73 AG-superposition is refutation complete for ground Horn clauses.

Proof: Let S be a set of ground Horn clauses (whose equations are in reductive form) such that S is closed under AG-superposition and $\Box \notin S$. We prove that then S is satisfiable by exhibiting an AG-model I for S, where I is the equality Herbrand interpretation defined as the congruence on ground terms generated by $R_S \cup AG$. Note that, since $R_S \cup R_{AG}$ is terminating and confluent, $I \models s \simeq t$ if, and only if, $s \rightarrow^*_{R_S \cup R_{AG}} \leftarrow^*_{R_S \cup R_{AG}} t$. We proceed by induction on $\succ_c$, that is, we derive a contradiction from the existence of a minimal (w.r.t. $\succ_c$) clause D (in reductive form) of S such that $I \not\models D$.

Let s be the maximal summand in D. Then D is either of the form $D' \vee ms \simeq t$ with $s \succ D'$ (a), or else it is $D' \vee ms \not\simeq t$ with $s \succeq D'$ (b). We first show that in both cases ms is reducible by R_S.

(a) Since $I \not\models D$, it has generated no rule of R_S. According to Definition 71, this can only be because ms is reducible by R_D. (b) Since $I \not\models D$, we have $I \models ms \simeq t$. Therefore ms and t is joinable by $R_S \cup R_{AG}$, and since $ms \succ t$, the maximal side ms, which is in normal form w.r.t. R_{AG}, has to be reducible by R_S. The rule reducing

ms has been generated by a clause of the form $C \vee nu \simeq v$, and there exists an inference by (direct or inverse) AG-superposition

$$\frac{C \vee l \simeq r \qquad D[l]_p}{C \vee D[r]_p}$$

where $I \not\models C \vee D[r]_p$ and D is larger w.r.t. $\succ_c$ than $C \vee D[r]_p$, and therefore, by Lemma 70, also larger than the reductive form of $C \vee D[r]_p$, contradicting the minimality of D. □

7.3.2 Selecting negative literals

It is easy to see that our inference rules remain complete with *selection* of negative literals (see, e.g., [BG94b]), where it is assumed that in each clause with a non-empty antecedent one of its negative equations has been *selected*. In the Horn case this leads to positive unit strategies (and in the non-Horn case to positive strategies): all left premises of AG-superpositions are positive unit clauses, and the only inferences involving non-unit clauses are AG-superpositions on the selected negative equation. The following result is a simple modification of the previous one; it is immediate if we define R_S such that only unit clauses generate rules:

Theorem 74 AG-superposition with selection is refutation complete for ground Horn clauses.

7.3.3 Deciding the satisfiability of sets of ground clauses

From our results it is not difficult to obtain a decision procedure for the satisfiability of arbitrary sets of ground clauses modulo AG.

For the Horn inference system with selection, each inference of $l \simeq r$ on a clause D produces a smaller clause D'. Furthermore, D is a logical consequence (modulo AG) of the smaller clauses $l \simeq r$ and D', i.e., D has become *redundant* in the sense [BG94b]. In our procedure such redundant clauses can be removed without loss of completeness (redundant clauses never generate any rules, and in the proof of the completeness theorem, they are never the smallest counter example; see, e.g., [BG94b] for details). Hence, if after each inference the maximal premise D is removed, the procedure remains complete, and at each inference the clause set decreases w.r.t. the multiset extension of the ordering and hence the process terminates, thus deciding satisfiability.

A decision procedure for the satisfiability of sets of arbitrary ground clauses modulo AG can be obtained by first transforming into Horn clauses (where $S \cup C \vee A_1 \vee \ldots \vee A_n$ is split into the disjunction of sets S_i of the form $S \cup C \vee A_i$; then S is satisfiable if some of the S_i is).

Theorem 75 AG-superposition with selection decides the satisfiability of sets of ground clauses modulo AG.

7.4 Inference rules for clauses with variables

In this section, we adapt the inference system in order to deal with equality constrained clauses with variables, where constraints are conjunctions of equalities $s = t$. As usual, the semantics of a constrained clause $C \mid T$ is the set of its ground instances, that is, the ground instances $C\sigma$ such that $T\sigma$ evaluates to true if $=$ is interpreted as $=_{AG}$. Then σ is called a solution for T. The empty clause with a constraint T is hence a contradiction, denoted simply by $\Box$, if, and only if, $T\theta$ is true for some ground θ.

Very roughly, the following is needed for lifting our completeness results from the ground case to equality constrained clauses with variables. If for clauses $C_1 \mid T_1$ and $C_2 \mid T_2$ there is an inference between ground instances

$$\frac{C_1\sigma \qquad C_2\sigma}{D}$$

then there exists an inference by the non-ground version of the inference rules

$$\frac{C_1 \mid T_1 \qquad C_2 \mid T_2}{D' \mid T}$$

such that D is a ground instance of $D' \mid T$.

As we will see in Section 7.5, for completeness it suffices to be able to do this only for instances with σ of C_1 and C_2 that are, in some technically rather involved sense, *irreducible* with respect to R_S, where R_S is the set of rules generated in a way similar to the previous section (but now by ground instances of clauses).

Definition 76 An equation $s = t$ is in *one-sided form* if it is of the form $e \simeq 0$ where e is in normal form w.r.t. R_{AG}.

Note that each equation has two (AG-equivalent) one-sided forms: for example, $x + y - z \simeq 0$ is equivalent to $-x - y + z \simeq 0$. In the following, we assume that all equations in clauses are kept in one-sided form. Unless explicitly stated otherwise, it does not matter which one of the two. Furthermore, for all substitutions σ, we assume w.l.o.g. that $x\sigma$ is in normal form w.r.t. R_{AG} for all x.

In order to define the non-ground inference rules, we now analyse for each inference rule how their premises have to be expressed. For simplicity, we omit the constraints, since they do not matter at this point; let us only remark that the amount of possible inferences can be further restricted in many different ways by checking their compatibility with the constraints.

7.4.1 Left premises of direct AG-superposition

Intuitively, our aim is the following. Let C be a clause with a positive equation $e \simeq 0$, and assume a ground instance $C\sigma$ of it generates a rule $nu \rightarrow r$ with $n > 0$, and $C\sigma$ is the left premise of an AG-superposition. Then for the non-ground case we have to be able to express $e \simeq 0$ as $s \simeq t$ such that the terms $s\sigma$ and $t\sigma$ have, respectively, nu and r as normal forms w.r.t. R_{AG}, and then perform the inference with AG-unification between s and the corresponding subterm of the right premise. *Orienting* $e \simeq 0$ as $s \simeq t$ in this way may require to *split* the variables of e into two parts:

Example 77 Consider the clauses $a + 2x \simeq b$ and $f(4a) \not\simeq f(a + b - 2c)$, where $a \succ b \succ c$. Assume that, for the instance where $x \mapsto a + c$, the equation $a + 2x \simeq b$ generates the rule $3a \rightarrow b - 2c$. Then there exists a ground inference

$$\frac{3a \rightarrow b - 2c \qquad f(4a) \not\simeq f(a + b - 2c)}{f(a + b - 2c) \not\simeq f(a + b - 2c)}$$

applied to three of the a's in $f(4a)$, where the conclusion in reductive form becomes $0 \not\simeq 0$ and hence the empty clause.

To cover this inference at the non-ground level, x has to be split into y (which, roughly, will contain the maximal summands in $x\sigma$) and z (for the remaining summands). Hence $a + 2x \simeq b$ can be *oriented* as $a + 2y \simeq b - 2z$. Then there is a non-ground inference

$$\frac{a + 2y \simeq b - 2z \qquad f(4a) \not\simeq f(a + b - 2c)}{f(a + b - 2z) \not\simeq f(a + b - 2c)}$$

unifying $a + 2y$ with three of the a's in $f(4a)$. AG-unifying both sides of the conclusion (which will be another inference rule; see below) detects the instance where z is c; the corresponding instance has a reductive form $0 \not\simeq 0$ and hence the contradiction is found. □

Definition 78 Let e be a term of the form $n_1 s_1 + \ldots + n_p s_p + m_1 x_1 + \ldots + m_q x_q$ where the s_i are non-variable summands, the x_i are variables, and the n_i and m_i are non-zero integers. By splitting each x_i into two new variables y_i and z_i, and splitting the summands into two disjoint sets, the equation $e \simeq 0$ can be written as an equivalent equation $s \simeq t$ of the form

$$n_1 s_1 + \ldots + n_k s_k + m_1 y_1 + \ldots + m_q y_q \simeq -n_{k+1} s_{k+1} - \ldots - n_p s_p - m_1 z_1 - \ldots - m_q z_q$$

In the following, we call each such an equation $s \simeq t$ an *orientation* for $e \simeq 0$ and we call the corresponding constraint τ of the form

$$x_1 = y_1 + z_1 \wedge \ldots \wedge x_q = y_q + z_q$$

the *splitting constraint* for this orientation.

It is not difficult to see that this notion of orientation fulfils what we wanted: if $e\sigma \simeq 0$ generates a rule $nu \rightarrow r$ then indeed for some orientation $s \simeq t$ of $e \simeq 0$ and some extension of σ in order to include the y_i and z_i, the terms $s\sigma$ and $t\sigma$ have, respectively, nu and r as normal forms w.r.t. R_{AG}. This we will see in detail in the completeness proofs.

Of course, the fewer orientations have to be considered for a given equation $e \simeq 0$, the fewer inferences will be performed, which is better for efficiency in practice. Indeed, a little more careful analysis reveals that a large number of optimizations are possible. In Section 7.8 we will mention some of them. It is also important for efficiency to exploit the unifiability and ordering restrictions as the strongest possible filters to avoid redundant inferences with such orientations $s \simeq t$. For example, apart from the unification restrictions of the inference itself, where s is unified with a subterm of the right premise, in the above orientation we can add $s_1 = \ldots = s_k$ to the constraint; in particular, this means, e.g., that if e is $f(\ldots) + g(\ldots) + \ldots$, then no orientation $s \simeq t$ is needed where both summands headed with f and g are in the left hand side s. In Section 7.8 the problem of checking the ordering restrictions is addressed.

Note that this notion of orientation does not depend on which one of $e \simeq 0$ or $-e \simeq 0$ we consider as the one-sided form, and that the non-deterministic aspect of orientation is the guess of a subset $s_1 \ldots s_k$ of the (non-variable) summands (where the guess is constrained by the requirement that all of them are AG-unifiable and by the requirements on $\succ$).

7.4.2 Left premises of inverse AG-superposition

Example 79 Consider $a \succ b \succ c$ and the clauses $f(-a+b+c) \not\simeq f(a-c)$ and $2x \simeq b$. With the instance $x \mapsto a-c$, the second equation becomes $2a \simeq b+2c$. At the ground level, there exists an inference with inverse AG-superposition which produces $f(a-c) \not\simeq f(a-c)$. At the non-ground level, x is split into $y+z$, and the inference is performed with $-y \simeq y+2z-b$, and we obtain $f(a+c+2z) \not\simeq f(a-c)$. From this, by AG-unification the instance $z \mapsto -c$ is found and the empty clause is obtained. □

Definition 80 Let e (or $-e$) be a term of the form $x_1 + \ldots + x_n + v$, where v contains only negative variables and (positive or negative) summands, and let e' be e but where every occurrence of x_i at top-level position has been replaced by $y_i - z_i$, where y_i and z_i are new variables. The splitting constraint τ is $x_1 = y_1 - z_1 \wedge \ldots \wedge x_n = y_n - z_n$. Hence e' is of the form $y_1 - z_1 + \ldots + y_n - z_n + v$.

Then, if e' is of the form $s + e''$ where s is a positive summand, then $-s \simeq e''$ is an *inverse orientation* for $e \simeq 0$ with splitting constraint τ.

Furthermore, if e' is of the form $w + e''$ where w is a variable (i.e., w is some y_i), then $-w_1 \simeq w_2 + e''$ is an *inverse orientation* for $e \simeq 0$ with splitting constraint $\tau \wedge w = w_1 + w_2$.

Finally, if e' is of the form $-w + e''$ where w is a variable, but none of the z_i, then $w_1 \simeq -w_2 + e''$ is an *inverse orientation* for $e \simeq 0$ with splitting constraint $\tau \wedge w = w_1 + w_2$.

The splitting of the variable w in the second case of inverse orientation is the one illustrated by the previous example. Example 86 shows the necessity of the splittings of the constraint τ.

7.4.3 Right premises for direct AG-superposition

Example 81 Consider $a \succ b \succ c$, the left premise $3a \simeq b$, and the right premise $f(2x, x) \not\simeq f(a + b + 2c, 2a + c)$. With the instance $\{x \mapsto 2a + c\}$, the right premise is $f(4a + 2c, 2a + c) \not\simeq f(a + b + 2c, 2a + c)$ which gives in one ground inference $f(a + b + 2c, 2a + c) \not\simeq f(a + b + 2c, 2a + c)$, which in reductive form is $0 \not\simeq 0$.

Now the question is: how can we, at the nonground level, perform the inference into the term $2x$? (which is the term t in the definition below). By splitting x into only the two variables y and z, one gets $f(y+y+z+z, y+z) \not\simeq f(a+b+2c, 2a+c)$, for which the ground inference cannot be lifted: it is impossible to split $y + y + z + z$ into $t_1 + t_2$ such that $t_1\sigma$ is $3a$, and $t_2\sigma$ is $a + 2c$ for some σ.

As we will see, by splitting x into three variables y, y', and z, lifting is always possible. In our example, then one gets $f(2y+2y'+2z, y+y'+z) \not\simeq f(a+b+2c, 2a+c)$, where $2y + 2y' + 2z$ is split into $2y + y'$ and $y' + 2z$ (these are the terms t_1 and t_2 in the definition below). Then an AG-unifier of $3a$ and $2y + y'$ instantiates y and y' with a, and the conclusion of the non-ground inference is $f(a + b + 2z, 2a + z) \not\simeq f(a+b+2c, 2a+c)$, which by one more AG-unification, where z is instantiated with c, becomes $0 \not\simeq 0$. □

Definition 82 Let t be a non-variable subterm of e in a literal $e \simeq 0$ or $e \not\simeq 0$ where t is not immediately below an AG-symbol and the head symbol of t is free or $+$. W.l.o.g., let t be of the form

$$n_1 s_1 + \ldots + n_p s_p + m_1 x_1 + \ldots + m_q x_q + t'$$

where all s_i are summands, all x_i variables, all n_i and m_i are positive coefficients, and t' contains only negative summands and variables.

Then $t_1 + t_2$ is a *splitting* for t if t_1 is a term whose head symbol is free or $+$ of the form

$$k_1 s_1 + \ldots + k_p s_p + m_1 y_1 + \ldots + m_q y_q + l_1 y'_1 + \ldots + l_q y'_q$$

where $0 \leq k_i \leq n_i$ and $0 \leq l_i < m_i$, and t_2 is

$$(n_1 - k_1)s_1 + \ldots + (n_p - k_p)s_p + m_1 z_1 + \ldots + m_q z_q + l'_1 y'_1 + \ldots + l'_q y'_q + t'$$

where l'_i is 0 if l_i is 0 (i.e., then x_i is split only into two parts y_i and z_i), and l'_i is $m_i - l_i$ otherwise. Again we denote by τ the corresponding splitting constraint.

As before, other restrictions apply; for example it is also not necessary to consider t_1 of the form $y_i + y'_i$ (i.e., if m_i is 1).

7.4.4 Right premises for inverse AG-superposition

Definition 83 Let t be a non-variable subterm of e in a literal $e \simeq 0$ or $e \not\simeq 0$ where t is not immediately below an AG-symbol.

If t is of the form $-s + t'$, where s is a summand, then $t_1 + t_2$ is an *inverse splitting* for t with empty splitting constraint τ if t_1 is $-s$ and t_2 is t'.

If t is of the form $-x + t'$, where x is a variable, then $t_1 + t_2$ is an *inverse splitting* for t if t_1 is $-y$ and t_2 is $-z + t'$, and the splitting constraint τ is $x = y + z$.

7.4.5 AG-superposition rules

Based on the notions of orientations and splittings defined in the previous subsections, we are now ready to define the inference system for Horn clauses with variables.

Definition 84 In the left premise $C \vee l \simeq r$ of the direct AG-superposition rule below, it is assumed that the actual clause is $C \vee e \simeq 0$ and that $l \simeq r$ is an orientation of $e \simeq 0$. Similarly, in the right premise, $D[t_1 + t_2]_p$ denotes that $D|_p$ is a non-variable term t that is not immediately below an AG symbol, with a splitting $t_1 + t_2$. In the same way, for the inverse AG-superposition rule, they denote inverse orientations and splittings. In all cases, τ is the conjunction of the splitting constraints of the two premises. The inference system $\mathcal{H}$ consists of the following three rules for constrained clauses:

direct AG-superposition:

$$\frac{C \vee l \simeq r \mid T \qquad D[t_1 + t_2]_p \mid T'}{C \vee D[r + t_2]_p \mid T \wedge T' \wedge l = t_1 \wedge \tau}$$

inverse AG-superposition:

$$\frac{C \vee l \simeq r \mid T \qquad D[t_1 + t_2]_p \mid T'}{C \vee D[r + t_2]_p \mid T \wedge T' \wedge l = t_1 \wedge \tau}$$

AG-zero-instance:

$$\frac{C \vee e \not\simeq 0 \mid T}{C \mid T \wedge e = 0}$$

The ordering restrictions of the superposition rules are the ones corresponding to the ground rules. More precisely, a direct (or inverse) superposition with premises $C_1 \mid T_1$ and $C_2 \mid T_2$ and conclusion $D \mid T$ is needed if, for some solution θ of T, there is a ground direct (resp. inverse) inference between the reductive forms of $C_1\theta$ and $C_2\theta$, and with conclusion $D\theta$. The AG-zero-instance rules can be restricted to maximal equations of the clause.

In the following sections, we will prove the refutation completeness of this inference system. But let us first illustrate some of the limitations and technical difficulties when dealing with constrained clauses, by means of an example taken from [NR01]. Note that in such examples where only free symbols occur, AG-superposition boils down to normal superposition.

Example 85 Consider the unsatisfiable clause set, with the ordering as in Lemma 64 based on $f \succ_{\mathcal{F}} a \succ_{\mathcal{F}} b \succ_{\mathcal{F}} c$:

1. $a \simeq b$
2. $f(x) \simeq c \mid x = a$
3. $f(b) \not\simeq c$

No inferences that are compatible with the constraint of the second clause can be made (a superposition inference between 2 and 3 leads to a clause with an unsatisfiable constraint $x = a \wedge b = x$). This incompleteness is due to the fact that the usual lifting arguments for superposition (see [NR01]) do not work here, since they are based on the existence of *all* ground instances of the clauses; in this case, it requires an instance $f(b) \simeq c$ of clause 2, which does not exist. This example also shows that one cannot deal with arbitrary initial constraints. For constrained clauses, the alternative technique for lifting is based on the notion of irreducible instances [NR01]. In this chapter we extend this idea of irreducible substitution. It becomes technically more complex due to the built-in properties of AG (example 91 gives an idea of it). □

Example 86 In this example it is shown how the inference system performs and also the need of the splitting of variables in the right premise of inverse AG-superposition is illustrated. Consider the clause $f(x) \not\simeq f(-a) \vee x + 3a \simeq 0$. With the instance $\{x \mapsto -a\}$, the negative equation in reductive form is $0 \simeq 0$. The positive equation is $2a \simeq 0$, which may generate the two rules $2a \to 0$ and $-a \to a$. If one wants to refute $f(-3a) \not\simeq f(a)$, then the inverse rule has to be used. Indeed, with $-a \to a$, the term $f(-3a)$ rewrites into $f(-2a + a)$, which is $f(-a)$, which rewrites into $f(a)$.

Now we want to perform, at the non-ground level, the ground refutation corresponding to these two rewrite steps. Assume that, at the non-ground level, we consider the orientation $-a \simeq 2a + x$, i.e., without the additional splitting of x as explained in definition 80. Then, by the corresponding inverse AG-superposition inference we obtain $f(x) \not\simeq f(-a) \vee f(x) \not\simeq f(a)$. If one adds constraints forcing a to be the maximal summand in the clause $f(x) \not\simeq f(-a) \vee x + 3a \simeq 0$ and such constraints are inherited, then no substitution different from $\{x \mapsto -a\}$ is possible (such constraints can be handled with the methods presented in Chapter 8. Now, one would want to do a new inference on x, but in $\mathcal{H}$ no inferences below variables are computed. So this shows the need of a splitting of x into $y - z$ in an inverse AG-superposition inference.

Indeed, if we do this additional splitting, the orientation becomes $-a \simeq 2a+y-z$. Then the instance under consideration is extended such that $\{y \mapsto 0, z \mapsto a\}$, and the obtained clause is $f(x) \not\simeq f(-a) \vee f(y - z) \not\simeq f(a)$, with the splitting constraint $x = y - z$. Now, it is possible to do the second inverse AG-superposition inference (the one corresponding to the second rewrite step with $-a \to a$). Applying $-a$ on $-z$, one obtains $f(x) \not\simeq f(-a) \vee f(x') \not\simeq f(-a) \vee f(y+2a+y'-z') \not\simeq f(a)$ (here, the x of the left premise is renamed into x') with the splitting constraint $x' = y' - z'$, and extending the substitution $\{y' \mapsto 0, z' \mapsto a\}$. With this substitution, all these equations are of the form $0 \simeq 0$, and three AG-zero-instance inferences give us the desired refutation. □

7.5 Completeness for a simple subcase

For explanation purposes, in this section we consider the simpler subcase where all free symbols are constants. Hence this is assumed in all results of this section. It is interesting to observe that in this subcase the inference rule of inverse AG-superposition is not needed.

As said before, we will deal with instances with ground substitutions σ of clauses C that are in some sense irreducible with respect to R_S, where R_S is the set of rules generated in a way similar to how it was done for the ground case in the previous section.

Example 87 Let s be a term and σ a substitution, both in normal form w.r.t. R_{AG}. Then still $s\sigma$ needs not be in normal form w.r.t. R_{AG}.

For example, if s is $-x + y + a$, $x\sigma$ is $a + b$, and $y\sigma$ is b, then $-x\sigma$ is AG-equal to $-a + (-b)$ and $s\sigma$ in AG-normal form is 0. □

Example 88 The problems illustrated in Example 85 still occur in this simple case where all free symbols are constants. Again with the ordering $a \succ b \succ c$, consider

1. $a \simeq b$
2. $b + x \simeq c \mid x = a$
3. $2b \not\simeq c$

No inferences are possible on this unsatisfiable set. □

Definition 89 Let C be a clause, let t be a term, let σ be a substitution in AG-normal form, and let R be a ground TRS.

The pair (t, σ) is *irreducible* w.r.t. R if for all variables x occurring in t, the term $x\sigma$ is irreducible w.r.t. $R^{\preceq -u}$ where u is the maximal (w.r.t. $\succ$) summand of $AG\text{-}nf(t\sigma)$.

The pair (C, σ) is irreducible w.r.t. R if (e, σ) is irreducible w.r.t. R for all equations $e \simeq 0$ of C.

Note that the notion of irreducibility for (C, σ) does not depend on which one-sided form $e \simeq 0$ is considered.

We now adapt the notion of rule generation to the non-ground case. Instead of by ground clauses in reductive form, now the rules are generated by the reductive forms of instances $C\sigma$ of clauses $C \mid T$ of S, where (C, σ) is irreducible:

Definition 90 Let S be a set of constrained Horn clauses, let $C \mid T$ be a clause in S with a ground instance $C\sigma$, and let G be the (ground) reductive form of $C\sigma$, where G is of the form $G' \vee nu \simeq r$. Then G *generates* the rule $nu \to r$ if the following four conditions are satisfied:

1. $(R_G \cup AG)^* \not\models G$
2. $u \succ r$ and $u \succ G'$
3. nu is irreducible by R_G
4. (C, σ) is irreducible w.r.t. R_G.

where R_G is the set of rules generated by reductive forms of instances of clauses of S that are smaller than G w.r.t. $\succ_c$. Furthermore, for each generated rule $nu \to r$ with $n > 1$, in addition the rule $-u \to (n-1)u - r$ is generated. The set of all rules generated by clauses in S is denoted by R_S.

In the remainder of this section R_S always denotes the ground TRS generated for a given S as in the previous definition.

Example 91 This example illustrates how the application of generated rules correspond to inferences at the non-ground level. It also shows why the irreducibility notion is more complicated than the standard one of superposition with constraints of [BGLS95] and [NR95], where, roughly speaking, one simply imposes that for every variable x the term $x\sigma$ has to be irreducible w.r.t. the rewrite system R.

Consider the equation $e \simeq 0$ of the form $2x-2a-2b+c \simeq 0$ where $a \succ b \succ c$, and the substitution σ such that $x\sigma$ is $a+b$. We have that $e\sigma \simeq 0$ is $2a+2b-2a-2b+c \simeq 0$, and its reductive form is $c \simeq 0$. The corresponding orientation at the non-ground level is $c \simeq 2a + 2b - 2x$. Due to this instance the rule $c \rightarrow 0$ may be generated. Later on, the rule $b \rightarrow 0$ may be generated too, due to other equations. The variable x with the substitution σ is reducible by such a rule $b \rightarrow 0$. So with the standard notion of irreducibility, rules generated later on could reduce the substitution of clauses generating smaller ones. Therefore this classical notion is not adequate in our context. Roughly speaking, we need to allow such big summands that are cancelled out to be reducible.

Indeed, with the notion used here, the one of Definition 89, we will see in Lemma 92 that $x\sigma$ will be irreducible w.r.t. all generated rules with maximal summand smaller than or equal to $-c$. And indeed this irreducibility is preserved in the conclusions of inferences. Assume we want to refute $2c + y \not\simeq 0$, where $y\sigma$ is 0 with the rule $c \rightarrow 0$. Observe that $(2c + y, \sigma)$ is irreducible w.r.t. the generated R. At the non-ground level, the reduction with $c \rightarrow 0$ corresponds to an inference with the orientation $c \simeq 2a + 2b - 2x$, and the resulting clause is $c + 2a + 2b - 2x + y \not\simeq 0$. Observe that $(c + 2a + 2b - 2x + y, \sigma)$ is irreducible w.r.t. R, since the maximal summand of $AG\text{-}nf(+2a+2b-2x\sigma)$ is c. Here, some constraints can be added, like for example $c > 2a + 2b - 2x$. Such constraints can be handled with the methods presented in Chapter 8. In this case, the only possible solution σ is $x\sigma = a + b$. □

The following lemma shows that our notion of orientation for left premises of direct AG-superposition fulfills the requirements.

Lemma 92 Let $C \mid T$ be a clause whose instance $C\sigma$ with reducive form C_r generates the rule $nu \rightarrow r$.

Then there exists an orientation $l_1 \simeq r_1$ of the positive equation $e \simeq 0$ of C, and some extension of σ in AG-normal form satisfying the splitting constraint of the orientation, and $AG\text{-}nf(l_1\sigma) = nu$ and $AG\text{-}nf(r_1\sigma) = r$. Furthermore, all variables x in r_1 satisfy that $x\sigma$ is irreducible w.r.t. $R_{S}^{\preceq -u}$.

Proof: W.l.o.g., let e be of the form $k_1x_1 + \ldots + k_px_p \ + \ ku \ + \ v$ where the k_i and k are (possibly zero) integers, the x_i are variables, and v is the (possibly 0) sum of constants different from u. Now consider the orientation of $e \simeq 0$ into $l_1 \simeq r_1$ where

$$l_1 = k_1y_1 + \ldots + k_py_p \ + \ ku$$

$$r_1 = -k_1 z_1 - \ldots - k_p z_p \; - \; v$$

i.e., where each x_i has been split into $y_i + z_i$. Furthermore, consider the extension of σ where $y_i\sigma$ consists of all (positive or negative) u in $x_i\sigma$, and z_i is the sum of the remaining constants, that is, if $x_i\sigma =_{ACO} m_i u + v_i$ where u does not occur in v_i, then $y_i\sigma = m_i u$ and $z_i\sigma = v_i$. Note that in v_i constants larger or smaller than u may appear, but not u itself.

Then $AG\text{-}nf(l_1\sigma) = nu$ and $AG\text{-}nf(r_1\sigma) = r$. It remains to be shown that every variable z_i in r_1 satisfy that $x\sigma$ is irreducible w.r.t. $R_S^{\preceq -u}$. We know that (e, σ) is irreducible w.r.t R_{C_r}, i.e., $x_i\sigma$ is irreducible w.r.t. $R_{C_r}^{\preceq -u}$. Then, since $z_i\sigma$ is a sum of constants that already appear in $x_i\sigma$, we have that $z_i\sigma$ is irreducible w.r.t. $R_{C_r}^{\preceq -u}$.
□

Note that in this case where all free symbols are constants, for a given clause with positive equation $e \simeq 0$ there are at most two orientations $l_1 \simeq r_1$: one where the maximal constant symbol of e (if there is any) is in l_1, and another one where there is no constant symbol at all in l_1 (if there is any variable in e).

Lemma 93 Let e be a term such that (e, σ) is irreducible w.r.t. R_S, and let $e\sigma \simeq 0$ in reductive form be $mu \simeq v$. Furthermore, let $nu \rightarrow r$ be a rule in R_S with $1 \leq n \leq m$.

Then there exists a splitting $e_1 + e_2$ of e and an extension of σ in AG-normal form satisfying the corresponding splitting constraint, and $(e_1 + e_2)\sigma =_{AG} e\sigma$ and $e_1\sigma =_{AG} nu$. Moreover, all variables x in e_2 satisfy that $x\sigma$ is irreducible w.r.t. $R_S^{\preceq -u}$.

Proof: For every variable x_i in e, w.l.o.g. we have $x_i\sigma =_{ACO} m_i u + v_i$ where u does not occur in v_i, and where $m_i \geq 0$ because (e, σ) is irreducible w.r.t. R_S (which contains $nu \rightarrow r$ and hence if $n > 1$ also $-u \rightarrow (n-1)u - r$).

Therefore, since $e\sigma \simeq 0$ in reductive form is $mu \simeq v$, and $m \geq n$, we can assume that $e \simeq 0$ (in one of its one-sided forms) is of the form

$$k_1 x_1 + \ldots + k_p x_p \; + \; ku \; + \; e' \simeq 0$$

where $k \geq 0$, e' is the (possibly 0) sum of the remaining constants and variables, and $\{x_1, \ldots, x_p\}$ is a minimal set of variables with positive coefficients k_i such that $k_1 m_1 + \ldots + k_p m_p + k \geq n$ or, if k is negative, $k_1 m_1 + \ldots + k_p m_p \geq n$.

Now we distinguish three possible situations:

1. $k \geq n$, and hence p is 0. Then some splitting of the form

$$\begin{aligned} e_1 &= nu \\ e_2 &= (k-n)u \; + \; e' \end{aligned}$$

   fulfils the requirements. Note that (e_2, σ) is irreducible w.r.t. R_S since e_2 has the same variables as e and the maximal summand of $AG\text{-}nf(e_2\sigma)$ is smaller than or equal to u, the maximal summand of $AG\text{-}nf(e\sigma)$.

2. $n > k > 0$. Then $k_1m_1+\ldots+k_pm_p+k \geq n > k_2m_2+\ldots+k_pm_p+k$ (the latter relation by minimality of the set $\{x_1,\ldots,x_p\}$). Now let l be $n-(k_2m_2+\ldots+k_pm_p+k)$, i.e., intuitively, l is the number of u's we need from the k_1m_1 u's in $x_1\sigma$. We assume that l *mod* k_1 is not 0 (the case of l *mod* $k_1 = 0$ is analogous and the diferences are commented below). Now let m' be l *div* k_1, let k' be l *mod* k_1, and consider the splitting

$$\begin{array}{rl} e_1 = & ku \;+\; k_1y \;+\; k'y' \;+\; k_2y_2+\ldots+k_py_p \\ e_2 = & (k_1-k')y'+k_1z \;+\; k_2z_2+\ldots+k_pz_p \;+\; e' \end{array}$$

where every x_i is split into y_i+z_i, except for x_1 that is split into $y+y'+z$ (if l *mod* k_1 is 0 then the variable y' is not needed in the splitting and x is split into $y+z$) and let $y\sigma$ be $m'u$, let $y'\sigma$ be u, let $z\sigma$ be $(m_1-m'-1)u+v_1$, and for i in $2\ldots p$, let $y_i\sigma$ be m_iu, and let $z_i\sigma$ be v_i. This fulfils the requirements, and, for similar reasons as in Lemma 92 we have that every variable x in e_2 satisfies that $x\sigma$ is irreducible w.r.t. $R_{\overline{S}}^{\prec -u}$.

3. $k \leq 0$. Then $k_1m_1+\ldots+k_pm_p \geq n > k_2m_2+\ldots+k_pm_p$. As in the previous case, assume that l *mod* k_1 is not 0, and let l be $n-k_2m_2+\ldots+k_pm_p$, let m' be l *div* k_1, let k' be l *mod* k_1, and consider the splitting

$$\begin{array}{rl} e_1 = & k_1y \;+\; k'y' \;+\; k_2y_2+\ldots+k_py_p \\ e_2 = & (k_1-k')y'+k_1z \;+\; k_2z_2+\ldots+k_pz_p \;+\; ku \;+\; e' \end{array}$$

and let $y\sigma$ be $m'u$, let $y'\sigma$ be u, let $z\sigma$ be $(m_1-m'-1)u+v_1$, and for i in $2\ldots p$, let $y_i\sigma$ be m_iu, and let $z_i\sigma$ be v_i. This fulfils the requirements, and, for similar reasons as in Lemma 92, every variable x in e_2 satisfies that $x\sigma$ is irreducible w.r.t. $R_{\overline{S}}^{\prec -u}$. □

The proof of the previous lemma reveals that the definition of splitting of right premises (Definition 82) could be made more restrictive. Indeed this is possible, thus reducing the number of inferences that need to be considered. In fact, the following more restrictive definition is also adequate for the general case handled in the next section, where we consider arbitrary free symbols. We decided to give Definition 82 as it is because it is simpler, but here we give the more restrictive alternative (it can be skipped by all readers except the ones interested in implementing these techniques in the most optimized way).

Let t be a non-variable subterm of e in a literal $e \simeq 0$ or $e \not\simeq 0$ where t is not immediately below an AG-symbol and the head symbol of t is free or $+$. W.l.o.g., let t be of the form

$$n_1s_1+\ldots+n_ps_p+m_1x_1+\ldots+m_qx_q+t'$$

where all s_i are summands, all x_i variables, all n_i and m_i are positive coefficients, and t' contains only negative summands and negative variables.

We choose a subset of the s_i, say $\{s_1 \dots s_{p'}\}$ with $p' \leq p$, and a subset of the x_i, say $\{x_1, \dots, x_{q'}\}$ with $q' \leq q$. The case where the subset of summands is empty, the subset of variables contains only x_1 and m_1 is 1 is not accepted (no inferences in variables are permitted). If (i) the subset of variables is empty, we choose a summand in $\{s_1 \dots s_{p'}\}$, say s_1, and a number $n'_1 \leq n_1$. Otherwise, if (ii) the subset of variables is non-empty we choose one of those variables, say x_1 and a number $m'_1 < m_1$.

In case (i), $t_1 + t_2$ is a *splitting* for t if t_1 and t_2 are of the form:

$$n'_1 s_1 + \dots + n_{p'} s_{p'}$$

$$(n_1 - n'_1) s_1 + n_{p'+1} s_{p'+1} \dots + n_p s_p + m_1 x_1 + \dots + m_q x_q + t'$$

respectively. In case (ii), split every variable x_i of $\{x_2, \dots, x_{q'}\}$ into $y_i + z_i$. If (ii.1) m'_1 is 0, then split x_1 into $y_1 + z_1$, and otherwise, if (ii.2) m'_1 is not 0, then split x_1 into $y_1 + y'_1 + z_1$. In case (ii.1), $t_1 + t_2$ is a *splitting* for t if t_1 and t_2 are of the form:

$$n_1 s_1 + \dots + n_{p'} s_{p'} m_1 y_1 + \dots + m_{q'} y_{q'} + t'$$

$$n_{p'+1} s_{p'+1} \dots + n_p s_p + m_1 z_1 + \dots + m_{q'} z_{q'} + m_{q'+1} x_{q'+1} + \dots + m_q x_q + t'$$

respectively. In case (ii.2), $t_1 + t_2$ is a *splitting* for t if t_1 and t_2 are of the form:

$$n_1 s_1 + \dots + n_{p'} s_{p'} m_1 y_1 + m'_1 y'_1 + m_2 y_2 + \dots + m_{q'} y_{q'} + t'$$

$$n_{p'+1} s_{p'+1} \dots + n_p s_p + (m_1 - m'_1) y'_1 + m_1 z_1 + \dots + m_{q'} z_{q'} + m_{q'+1} x_{q'+1} + \dots + m_q x_q + t'$$

respectively.

Theorem 94 $\mathcal{H}$ is refutation complete for constrained Horn clauses where all free symbols are constants and the initial set of clauses has only empty constraints.

Proof: In fact, we will show that in this case where all free symbols are constants, no inferences by inverse superposition are needed. Let S be the closure under $\mathcal{H}$ of a set of Horn clauses S_0 without constraints, and assume $\square \notin S$. Again we prove that then the equality Herbrand interpretation I defined as the congruence on ground terms generated by $R_S \cup AG$ is an AG-model for S. But now this is done in two steps. Let $Ir_{R_S}(S)$ denote the set of ground instances $C\sigma$ of $C \mid T$ in S such that (C, σ) is irreducible w.r.t. R_S.

1. First, it is proved that $I \models Ir_{R_S}(S)$, in a very similar way as for the ground case, by deriving a contradiction from the existence of such a $C\sigma$ whose reductive form is minimal w.r.t. $\succ_c$. This is done in detail below.

2. Second, from $I \models Ir_{R_S}(S)$ it follows that $I \models S$ for the following reasons. For each ground instance $C\sigma$ of a clause $C \mid T$ in S_0, consider another instance $C\sigma'$ of C, where $x\sigma'$ is the normal form w.r.t. R_S of $x\sigma$ for every variable x of C. Since T is empty (as S_0 has no constraints), $C\sigma'$ is also an instance of S_0. It is also in $Ir_{R_S}(S_0)$, since (C, σ') is obviously irreducible. Since $S_0 \subseteq S$ and $I \models Ir_{R_S}(S)$ we have $I \models Ir_{R_S}(S_0)$ and hence $I \models C\sigma'$, which implies $I \models C\sigma$, and hence we also have $I \models S_0$. But since $S_0 \models S$, this gives us $I \models S$.

We now prove the first part. Let C_r be the minimal, w.r.t $\succ_c$, reductive form of some $C\sigma$ in $Ir_{R_S}(S)$ that is an instance of a clause $C \mid T_C$ such that $I \not\models C_r$.

If C_r is a disjunction of literals of the form $0 \not\simeq 0$, then an inference by AG-zero-instance applies to any one of these literals, eliminating it, and its conclusion has a smaller false counter example.

Otherwise, as in the ground case (the proof of Theorem 73), let s be the maximal summand in C_r. Then C_r is either of the form $C'_r \vee ms \simeq t$ with $s \succ C'_r$, or else it is $C'_r \vee ms \not\simeq t$ with $s \succeq C'_r$. Then C is of the form $C' \vee e \simeq 0$ or $C' \vee e \not\simeq 0$, where the reductive forms of C' and $e\sigma \simeq 0$ are C'_r and $ms \simeq t$ respectively.

As in Theorem 73, in both cases ms is reducible by R_S. Since all free symbols are constants, the rule reducing ms must be of the form $ns \rightarrow r$, with $m \geq n \geq 1$. This rule has been generated by the reductive form D_r of an instance $D\sigma$ of a clause $D \mid T_D$. Let D be of the form $D' \vee e' \simeq 0$.

Then by Lemma 92 there exists an orientation $l_1 \simeq r_1$ of $e' \simeq 0$ and an extension of σ preserving AG-equality such that $AG\text{-}nf(l_1\sigma)$ is ns and $AG\text{-}nf(r_1\sigma)$ is r, and such that every variable x in r_1 satisfies that $x\sigma$ is irreducible w.r.t. $R_S^{\preceq-u}$.

Furthermore, by Lemma 93, there exists a splitting $e_1 + e_2$ of e and a new extension of σ (here we assume as usual that both clauses C and D contain diferent variables and that the splittings in them are done also with diferent variables) that is AG-preserving such that $(e_1 + e_2)\sigma =_{AG} e\sigma$, and $e_1\sigma =_{AG} ns$, and where every variable x in e_2 satisfies that $x\sigma$ is irreducible w.r.t. $R_S^{\preceq-u}$.

Now, since every variable x of $r_1 + e_1$ satisfies that $x\sigma$ is irreducible w.r.t. $R_S^{\preceq-u}$, and since the maximal summand of $AG\text{-}nf((r_1 + e_1)\sigma)$ is smaller than or equal to u, it holds that $(r_1 + e_1, \sigma)$ is irreducible w.r.t. R_S.

Now, the following inference exists:

$$\frac{D' \vee l_1 \simeq r_1 \mid T_D \qquad C' \vee e_1 + e_2 \simeq 0 \mid T_C}{C' \vee r_1 + e_2 \simeq 0 \mid T_D \wedge T_C \wedge l = e_1 \wedge \tau}$$

Its conclusion belongs to S, since S is closed under $\mathcal{H}$, and it has an instance with σ that contradicts the minimality of C_r. □

7.6 Completeness for Arbitrary Horn clauses

In this section we drop the restriction that all free symbols are constants. All definitions and proofs that are needed for this purpose follow the same intuition as what was done in the constants-only case, but several aspects become technically a bit more involved.

Example 95 This example shows that in the presence of arbitrary free symbols a more refined notion of irreducibility than the one of Definition 89 is needed. We continue with Example 91, and consider new problems due to the non-constant symbols. Suppose we have a unary symbol f bigger than a, b and c, and an equation $f(c) \simeq 0$. It is reducible with the rule $c \rightarrow 0$, that, at the non-ground level is $c \rightarrow 2a + 2b - 2x$, with the substitution $\{x \mapsto a + b\}$. By the corresponding direct AG-superposition inference we obtain $f(2a + 2b - 2x) \simeq 0$. At the ground level it is of the form $f(0) \simeq 0$. Observe that $f(0) \succ x\sigma$, and hence, $x\sigma$ would be reducible by a rule with left hand side b, that is smaller than the maximal summand of the equation. For this reason, we need a more complex notion of irreducibility, where the irreducibility of a variable x in an AG-context is only necessary for summands in $x\sigma$ that are smaller than or equal to the maximal reducible summand of such an AG-context, and not to the maximal summand in the equation. □

The following definitions are parameterized by the given rewrite system R, and we always denote (possibly with subindices) terms by s, t, u, v, positions by p, q and variables by x, y, z.

We first define irreducibility for pairs (s, σ) where s is a term and σ a substitution, both in normal form w.r.t. R_{AG}. Then still $s\sigma$ needs not be in normal form w.r.t. R_{AG}, because the following two kinds of steps may be applicable: (i) if x is a variable occurring imediately below a $-$ in s and $x\sigma$ is headed by $+$, then this $-$ is "moved inwards"; (ii) after this, some "complementary" pairs u and $-u'$ below the same $+$ are cancelled if u and u' are summands with $u =_{AG} u'$.

Definition 96 Let s be a ground term, and let R be a ground TRS. We define $maxred_R(s)$ to be the maximal summand u such that either:

- *AG-nf*(s) is of the form $nu + v$ and $nu \rightarrow r \in R$;
  In this case we say that u is determined by a *top-level positive reduction.*

- *AG-nf*(s) is of the form $-u + v$, and $-u \rightarrow r \in R$;
  Then u is determined by a *top-level negative reduction.*

- *AG-nf*(s) is of the form $u + v$ or $-u + v$ and u is reducible at non-top-level by R;
  Then u is determined by a *non-top-level reduction.*

Definition 97 Let s be a term and let σ be a substitution, both in normal form w.r.t. R_{AG}, and let R be a ground TRS.

The pair (s, σ) is called *recursively irreducible* w.r.t. R if the following conditions hold. Let u be $maxred_R(s\sigma)$.

1. for all x such that s is of the form $x + s'$, and all summand v with $u \succeq v$ and such that $x\sigma$ is of the form $mv + v'$, if u is determined by a top-level negative reduction, then either $u \succ v$ and mv is irreducible w.r.t. R, or v is u and m is positive; otherwise (top-level positive or non-top level reduction) mv is irreducible w.r.t. R.

2. for all x such that s is of the form $-x + s'$, and all summand v with $u \succeq v$ and such that $x\sigma$ is of the form $mv + v'$, if u is determined by a top-level negative reduction, then either $u \succ v$ and mv is irreducible w.r.t. R, or v is u; otherwise (top-level positive or non-top level reduction) mv is irreducible w.r.t. R.

3. for all t of the form $f(t_1, \ldots, t_m)$ such that $AG\text{-}nf(s\sigma)$ is of the form $t + v$ or $-t + v$ and $u \succeq AG\text{-}nf(t\sigma)$, each (t_i, σ) is recursively irreducible w.r.t. R.

Definition 98 Let s be a term, let u be a summand, and let σ be a substitution, both in normal form w.r.t. R_{AG}, let C be a clause, and let R be a ground TRS.

The pair (s, σ) is called *$(u \succeq)$-irreducible* (resp. *$(u \succ)$-irreducible*) w.r.t. R if the following conditions hold.

1. for all x such that s is of the form $x + s'$ or $-x + s$, and all summand v with $u \succeq v$ (resp. $u \succ v$) and such that $x\sigma$ is of the form $mv + v'$, the term mv is irreducible w.r.t. R.

2. for all t of the form $f(t_1, \ldots, t_m)$ such that $AG\text{-}nf(s\sigma)$ is of the form $t + v$ or $-t + v$ and $u \succeq AG\text{-}nf(t\sigma)$, each (t_i, σ) is recursively irreducible.

If u is the maximal summand of $AG\text{-}nf(s\sigma)$ w.r.t. $\succ$, then, we simply say that the pair (s, σ) is *irreducible* w.r.t. R.

The pair (C, σ) is irreducible w.r.t. R if (e, σ) is irreducible for all its equations $e \simeq 0$ (note that this notion does not depend on which one of the two possibilities of writing the equation like $e \simeq 0$ is chosen).

7.6.1 Model generation

As in the case where all free symbols are constants, which was explained in Section 7.5, now the AG-model induced by R is built. Again the rules are generated, exactly as in Definition 90 of Section 7.5, by the reductive forms of instances $C\sigma$ of clauses $C \mid T$ of S, where (C, σ) is irreducible. But now the notion of irreducibility is according to Definition 98. The main theorem of this section says that $\mathcal{H}$ is refutation complete for constrained Horn clauses if the initial set of clauses has only empty constraints. Its proof follows the same arguments as its analogous of the constants-only case, Theorem 94. Lemma 106 finds, for a given term that is reducible by R, a context inside it where the maximal summand is reducible at the top. This gives us an inference at the ground level. Lemmas 107, 108, 109 and 110 justify that there exist orientations and splittings at the non-ground level corresponding to the inference at the ground level. This new inference at the non-ground level has to satisfy some conditions of irreducibility that are justified by Lemmas 113, 115 and 116.

Lemma 99 Let u be the maximal summand in $AG\text{-}nf(s\sigma)$, let R_1 be a rewrite system and let R_2 be a rewrite system with left hand sides of the form nw or $-w$, where $n > 0$ and w is a summand such that $w \succ u$. Let (s, σ) be recursively irreducible w.r.t. R_1.

Then, (s, σ) is recursively irreducible w.r.t. $R_1 \cup R_2$.

Proof: We prove it by induction on the size of s. Let v be $maxred_{R_1}(s\sigma)$. Observe that $u \succeq v$. Since u is the maximal summand of $AG\text{-}nf(s\sigma)$, and for all the w, we have that $w \succ u$, then $maxred_{R_1 \cup R_2}(s\sigma)$ is v. Moreover, the sets $R_1^{\prec mv}$ and $(R_1 \cup R_2)^{\prec mv}$ coincide for any m. Therefore, the conditions of recursive-irreducibility for variables x such that s is of the form $x + s'$ or $-x + s'$ are satisfied. Let s be of the form $t + s'$ or $-t + s'$, for a summand t of the form $f(t_1, \ldots, t_n)$, and such that $v \succeq AG\text{-}nf(t\sigma)$. Then, we have that $v \succ AG\text{-}nf(t_i\sigma)$. Therefore, for all the w, we have that w is greater than the maximal summand in $AG\text{-}nf(t_i\sigma)$. By induction hypothesis, (t_i, σ) is recursively irreducible w.r.t. $R_1 \cup R_2$. □

Lemma 100 Let u be the maximal summand in $AG\text{-}nf(s\sigma)$, let R_1 be a rewrite system and let R_2 be a rewrite system with left hand sides of the form nw or $-w$, where $n > 0$ and w is a summand. Let v be a ground summand in AG-normal form such that $v \succeq u$ and (s, σ) is $(v \succeq)$-irreducible w.r.t. R_1.

If all such w satisfy that $w \succ v$, then, (s, σ) is $(v \succeq)$-irreducible w.r.t. $R_1 \cup R_2$.

If all such w satisfy that $w \succeq v$, then, (s, σ) is $(v \succ)$-irreducible w.r.t. $R_1 \cup R_2$.

Proof: We only prove the first statement (the second one is analogous). Observe that the sets $R_1^{\preceq mv}$ and $(R_1 \cup R_2)^{\preceq mv}$ coincide for any m. Therefore, the conditions of $(v \succeq)$-irreducibility for variables x such that s is of the form $x + s'$ or $-x + s'$ are satisfied. Let s be of the form $t + s'$ or $-t + s'$, for a summand t of the form $f(t_1, \ldots, t_n)$, and such that $v \succeq AG\text{-}nf(t\sigma)$. Then, we have that $v \succ AG\text{-}nf(t_i\sigma)$.

Therefore, for all the w, we have that w is greater than the maximal summand in $AG\text{-}nf(t_i\sigma)$. By Lemma 99, (t_i, σ) is recursively irreducible w.r.t. $R_1 \cup R_2$. □

Lemma 101 If, as in the definition of R, the reductive form $Cred$ of $C\sigma$ generates the rules $nu \rightarrow r$ and $-u \rightarrow (n-1)u - r$, then (C, σ) is irreducible not only w.r.t. R_{Cred}, but w.r.t. $R \setminus \{nu \rightarrow r,\ -u \rightarrow (n-1)u - r\}$. Moreover, if $e \simeq 0$ is a negative equation of C, then (e, σ) is irreducible w.r.t. R.

Proof: Let $e \simeq 0$ be an equation of C. Let R^{Cred} be the set of rules generated by reductive forms bigger than $Cred$ w.r.t. $\succ_c$. Then, R^{Cred} is of the form $\bigcup_{i \in I}\{n_i u_i \rightarrow r_i,\ -u_i \rightarrow (n_i - 1)u_i - r_i\}$. All these u_i's are larger than the maximal summand of $AG\text{-}nf(e\sigma)$. Moreover, if $e \simeq 0$ is a negative equation, u is larger than the maximal summand $AG\text{-}nf(e\sigma)$. By applying Lemma 100 with R^{Cred} and $R^{Cred} \cup \{nu \rightarrow r, -u \rightarrow (n-1)u - r\}$ for negative equations, the lemma follows. □

Lemma 102 Let s be a term of the form $s_1 + s_2$. Let (s, σ) be $(u \succeq)$-irreducible (resp. $(u \succ)$-irreducible) w.r.t. R', for a given summand u. Then, (s_1, σ) and (s_2, σ) are $(u \succeq)$-irreducible (resp. $(u \succ)$-irreducible) w.r.t. R'.

Lemma 103 Let s be a term of the form $s_1 + s_2$. Let (s, σ) be recursively irreducible w.r.t. R'. Let $maxred_R(s\sigma) \succ maxred_R(s_1\sigma)$.

Then, (s_1, σ) is recursively irreducible w.r.t. R'.

Lemma 104 Let (s, σ) be $(u \succeq)$-irreducible (resp. $(u \succ)$-irreducible or recursively irreducible) w.r.t. R. Let s be of the form (i) $nx + s'$ or (ii) $-nx + s$. Let x_1 and x_2 be variables not in s such that $x_1\sigma$ and $x_2\sigma$ are in AG-normal form, and $(x_1 + x_2)\sigma =_{ACO} x\sigma$.

Then, we have that (i) $(nx_1 + nx_2 + s', \sigma)$ or (ii) $(-nx_1 - nx_2 + s', \sigma)$ is $(u \succeq)$-irreducible (resp. $(u \succ)$-irreducible or recursively irreducible) w.r.t. R.

Lemma 105 Let (s, σ) be $(u \succ)$-irreducible w.r.t. R. Let s be of the form $nx + s'$. Let $x\sigma$ be (i) $u_1 + \ldots + u_m - v_1 - \ldots - v_k$, or (ii) $u_1 + \ldots + u_m$ or (iii) $-v_1 - \ldots - v_k$, where the u_i and v_i are summands. Let x_1 and x_2 be variables not in s. Let $x_1\sigma$ be $u_1 + \ldots + u_m$ in cases (i) and (ii), and 0 otherwise. Let $x_2\sigma$ be $v_1 + \ldots + v_k$ in cases (i) and (iii), and 0 otherwise.

Then, we have that $(nx_1 - nx_2 + s', \sigma)$ is $(u \succ)$-irreducible w.r.t. R.

Proof: Since s' is a subsum of s and $x_1\sigma$ is a subsum of $x\sigma$, the only doubt for reducibility is what happens with $x_2\sigma$. If $x_2\sigma$ is of the form $mv_i + v'$ for some v_i such that $u \succ v_i$, then $x\sigma$ is of the form $-mv_i + v''$. Since (s, σ) is $(u \succ)$-irreducible, v_i is irreducible w.r.t. R, and no rule with left-hand-side $-v_i$ nor v_i appears in R, and hence, a term of the form $n'v_i$ is not a left-hand-side of a rule of R. Hence, such variables x_2 satisfy the conditions for irreducibility, and $(nx_1 - nx_2 + s', \sigma)$ is $(u \succ)$-irreducible w.r.t. R. □

Lemma 106 Let t be a term in AG-normal form and reducible by R. Then, there exists an AG-context t' of t, and a summand u such that u is $maxred_R(t')$ by top-level reduction.

Proof: This can be proved by induction on the size of t. The term t by itself is an AG-context of t. Let v be $maxred_R(t)$. If it is by top-level reduction, then, u is v, and we are done. Otherwise, it is by non-top-level reduction, and then, v is of the form $f(v_1, \ldots, v_n)$, and one of the v_i is reducible. Then, by induction hypothesis, this v_i contains the t' and u satisfying the required condition. □

Lemma 107 Let the reductive form *Cred* of $C\sigma$ generate the rule $nu \to r'$.

Then there exists an orientation $l \simeq r$ of the positive equation $e \simeq 0$ of C, and an extension of σ satisfying the splitting constraint of the orientation, such that $AG\text{-}nf(l\sigma)$ is nu, $AG\text{-}nf(r\sigma)$ is r', and (r, σ) is $(u \succeq)$-irreducible w.r.t. R.

Proof: By Lemma 101, (e, σ) is irreducible w.r.t. $R \setminus \{nu \to r, -u \to (n-1)u - r\}$. In fact, it is $(u \succeq)$-irreducible w.r.t. $R \setminus \{nu \to r, -u \to (n-1)u - r\}$, since u is the maximal summand of $AG\text{-}nf(e\sigma)$. Observe that AC-changes in the substitution do not affect irreducibility. Hence we can suppose that $x_i\sigma$ is of the form (i) l_iu, or (ii) $l_iu + v_i$, or (iii) v_i, for all variables x_i in e, where v_i has no ocurrences of u at top-level position. Let e' be the result of replacing each ocurrence of x_i at top-level position by $y_i + z_i$, where y_i and z_i are new variables. Let σ be extended such that $y_i\sigma$ is l_iu (in cases i and ii) or 0 (in case iii), and $z_i\sigma$ is 0 (case i) and v_i otherwise. By Lemma 104, (e', σ) is $(u \succeq)$-irreducible w.r.t. $R \setminus \{nu \to r, -u \to (n-1)u - r\}$.

Now, we may write e' as $l + l'$, for terms l and l' such that l contains all the y_i, and all the summands t at top-level position in e' such that $AG\text{-}nf(t\sigma)$ is u; and l' contains all the z_i, and the rest of the summands. By Lemma 102, (l', σ) is $(u \succeq)$-irreducible w.r.t. $R \setminus \{nu \to r, -u \to (n-1)u - r\}$.

We have that $AG\text{-}nf(l\sigma)$ is nu, and $AG\text{-}nf(l'\sigma)$ is $AG\text{-}nf(-r')$. Moreover, if l' is of the form $x + l''$ or $-x + l''$, and $x\sigma$ is of the form $mv + v'$ for some summand v with $u \succeq v$, then, necessarily $u \succ v$, due to the way we have extended σ to the variables in l'. Therefore, mv is irreducible w.r.t. R, because it is irreducible w.r.t. $\{nu \to r, -u \to (n-1)u - r\}$, and w.r.t. $R \setminus \{nu \to r, -u \to (n-1)u - r\}$, since (l', σ) is $(u \succeq)$-irreducible w.r.t. $R \setminus \{nu \to r, -u \to (n-1)u - r\}$.

Furthermore, if l' is of the form $t + l''$ or $-t + l''$, for some summand t of the form $f(t_1, \ldots, t_n)$ such that $u \succeq AG\text{-}nf(t\sigma)$, we have that $u \succ AG\text{-}nf(t_i\sigma)$, and, by Lemma 99, (t_i, σ) is recursively irreducible w.r.t. R.

Therefore, (l', σ) is $(u \succeq)$-irreducible w.r.t. R. And, if we take r as $AG\text{-}nf(-l')$, (r, σ) is $(u \succeq)$-irreducible w.r.t. R. □

Lemma 108 Let the reductive form *Cred* of $C\sigma$ generate the rule $-u \to (n-1)u - r'$.

Then there exists an orientation $l \simeq r$ of the positive equation $e \simeq 0$ of C, and an extension of σ satisfying the splitting constraint of the orientation, such that $AG\text{-}nf(l\sigma)$ is $-u$, $AG\text{-}nf(r\sigma)$ is $(n-1)u - r'$, and (r,σ) is $(u \succ)$-irreducible w.r.t. R. Moreover, for all x such that r is of the form $x+s$, we have that $x\sigma$ is not of the form $-u+s'$.

Proof: By Lemma 101, (e,σ) is irreducible (in fact $(u \succeq)$-irreducible) w.r.t. $R \setminus \{nu \rightarrow r, -u \rightarrow (n-1)u - r\}$. This implies that if e is of the form $x + e_2$ or $-x + e_2$ and $x\sigma$ is of the form $mv + v'$ for some summand v with $u \succ v$, then mv is irreducible w.r.t. $R \setminus \{nu \rightarrow r, -u \rightarrow (n-1)u - r\}$, and, in fact, w.r.t. R. Additionally, if e is of the form $t + e_2$ or $-t + e_2$ for some summand t of the form $f(t_1,\ldots,t_n)$ such that $u \succeq AG\text{-}nf(t\sigma)$, we have that (t_i,σ) is recursively irreducible w.r.t. $R \setminus \{nu \rightarrow r, -u \rightarrow (n-1)u - r\}$. But observe that, since $u \succ AG\text{-}nf(t_i\sigma)$, by Lemma 99, (t_i,σ) is recursively irreducible w.r.t. R. Alltogether this implies that (e,σ) is $(u \succ)$-irreducible w.r.t. R.

Let us consider now a certain variable x that appears in e at top-level positive variable position. AC-changes in the substitution do not affect irreducibility. Hence we can assume that $x\sigma$ is of the form (i) v, or (ii) $v+w$, or (iii) w, where v (resp. w) contains only positive (resp. negative) summands at their top-level positions. Let e' be the result of replacing each ocurrence of x at top-level positive variable position by $y - z$, where y and z are new variables. Let σ be extended such that $y\sigma$ is v (in cases i and ii) or 0 (in case iii), and $z\sigma$ is 0 (case i) and $AG\text{-}nf(-w)$ otherwise. By Lemma 105, (e',σ) is $(u \succ)$-irreducible w.r.t. R. We can repeat this process with all the variables in e at top-level position. Let the resulting term be e'. Again, (e',σ) is $(u \succ)$-irreducible w.r.t. R.

Since $AG\text{-}nf(e'\sigma)$ is $nu - r$, either (i) e' is of the form $x + e''$ for some variable x and $x\sigma$ is of the form $u + e'''$ or u, or (ii) e' is of the form $-x + e''$, and $x\sigma$ is of the form $-u + e'''$ or $-u$, or (iii) e' is of the form $t + e''$ for some summand t such that $t\sigma =_{AG} u$.

In case (i), we replace this occurrence of x by $x_1 + x_2$, where x_1 and x_2 are new variables, and we extend σ such that $x_1\sigma$ is u, and $x_2\sigma$ is e''' or 0, depending on the case. By Lemma 104, we have that $(x_1 + x_2 + e'', \sigma)$ is $(u \succ)$-irreducible w.r.t. R. By Lemma 102 $(x_2 + e'', \sigma)$ is $(u \succ)$-irreducible w.r.t. R. Therefore $-x_1 \simeq x_2 + e''$ is an orientation that satisfies the required conditions.

Case (ii) is identical to case (i), but now, the obtained orientation is $x_1 \simeq -x_2 + e''$.

In case (iii), by Lemma 102, we have that (e'',σ) is $(u \succ)$-irreducible w.r.t. R. Therefore $-t \simeq e''$ is an orientation that satisfies the required conditions. □

Lemma 109 Let $nu \rightarrow r'$ be a rule of R. Let (s,σ) be (i) irreducible or (ii) recursively irreducible w.r.t. R. Let $AG\text{-}nf(s\sigma)$ be of the form $nu + s'$. Let u be in case (i) the maximal summand of $AG\text{-}nf(s\sigma)$, or in case (ii) $maxred_R(s\sigma)$.

Then, there exists a splitting $s_1 + s_2$ of s, and an extension of σ satisfying the corresponding splitting constraint, such that $(s_1 + s_2)\sigma =_{AG} s\sigma$, and $s_1\sigma$ is nu, and (s_2, σ) is $(u \succeq)$-irreducible w.r.t. R.

Moreover, in case (i), the maximal summand of $AG\text{-}nf(s_2\sigma)$, and, in case (ii), the summand $maxred_R(s_2\sigma)$, is smaller than or equal to u w.r.t. $\succ$.

Proof: From our hypothesis, it follows that (s, σ) is $(u \succeq)$-irreducible w.r.t. R (observe that for the case (ii) u is not determined by top-level negative reduction). Moreover, if s is of the form $-x+t$, then x is not of the form $-u+t'$. Since $AG\text{-}nf(s\sigma)$ is $nu+s'$, these n u's can not be provided by negative variables at top-level position. Thus s has to be of the form $m_1x_1 + \ldots + m_qx_q + n_1t_1 + \ldots + n_pt_p + s''$, where the m_i and n_i are positive, $AG\text{-}nf(t_i\sigma)$ is u for i in $1 \ldots p$, and $x_i\sigma$ is of the form $k_iu + v_i$ for i in $1 \ldots q$ with positive k_i, and $m_1 * k_1 + \ldots + m_q * k_q + n_1 + \ldots n_p \geq n$. Moreover, such x_i and t_i can be chosen to satisfy the following conditions: the x_i and t_i do not appear in s'', and p is maximal (i.e. if q is not 0, that is there is at least one chosen variable x_1, then no summand t such that s'' is of the form $t + s'''$ satisfies $t\sigma =_{AG} u$), and q is minimal (i.e. by eliminating one variable, say x_1, we have that $m_2 * k_2 + \ldots + m_q * k_q + n_1 + \ldots + n_p < n$). The case where p is 0 and q is 1 and m_1 is 1 is not possible, since $x_1\sigma$ cannot contain more than $n-1$ u's, because it would be reducible w.r.t. R, contradicting the $(u \succeq)$-irreducibility of (s, σ).

For facility of explanations, we assume that $m_1 * k_1 + \ldots + m_q * k_q + n_1 + \ldots n_p$ is exactly n. Other situations are treated analogously, by doing the corresponding additional splittings as explained in Lemma 93.

Now, we split every x_i into $y_i + z_i$, where y_i and z_i are new variables, and σ is extended such that $y_i\sigma$ is k_iu, and $z_i\sigma$ is v_i. Thanks to Lemma 104, the obtained term is $(u \succeq)$-irreducible w.r.t. R. It may be written $s_1 + s_2$, where s_1 contains all the t_i and all the y_i, and s_2 contains the rest of summands and variables. The AG-normal form of $s_1\sigma$ is nu, and of $s_2\sigma$ is s', and $s_1 + s_2$ is a splitting for s. By Lemma 102, (s_2, σ) is $(u \succeq)$-irreducible w.r.t. R.

Moreover, in case (i), the maximal summand of $AG\text{-}nf(s_2\sigma)$, and, in case (ii), the summand $maxred_R(s_2\sigma)$, is smaller than or equal to u w.r.t. $\succ$. Observe that, ommiting the summand u, the AG-normal forms of $s\sigma$ and $s_2\sigma$ coincide. □

Lemma 110 Let $n > 1$ and $-u \rightarrow (n-1)u - r'$ be a rule of R. Let (s, σ) be recursively irreducible w.r.t. R. Let $AG\text{-}nf(s\sigma)$ be of the form $-u + s'$. Let u be $maxred_R(s\sigma)$.

Then, there exists a splitting $s_1 + s_2$ of s, and an extension of σ satisfying the corresponding splitting constraint, such that $(s_1 + s_2)\sigma =_{AG} s\sigma$, and $s_1\sigma$ is $-u$, and (s_2, σ) is $(u \succ)$-irreducible and recursively irreducible w.r.t. R. Moreover, we have that $u \succeq maxred_R(s_2\sigma)$.

Proof: From our hypothesis, it follows that (s, σ) is $(u \succ)$-irreducible w.r.t. R. Moreover, if s is of the form $x + s''$, then $x\sigma$ is not of the form $-u + s'''$. Since the AG-normal form of $s\sigma$ is $-u + s'$, then, either (i) s is of the form $-v + t$ for some summand v such that $AG\text{-}nf(v\sigma)$ is u, or (ii) s is of the form $-x + t$ for some variable x such that $x\sigma$ is of the form $u + s'''$ or u.

In case (i), we may take $-v$ as s_1, and t as s_2. Then $s_1 + s_2$ is a splitting for s, and $u \succeq maxred_R(s_2\sigma)$.

In case (ii), we may split x into $x_1 + x_2$, for new variables x_1 and x_2, and extend σ such that $x_1\sigma$ is u and $x_2\sigma$ is s''' or 0, depending on the case. By Lemma 104, $-x_1 - x_2 + t$ is $(u \succ)$-irreducible and recursively irreducible w.r.t. R. We may take $-x_1$ as s_1, and $-x_2 + t$ as s_2. Then $s_1 + s_2$ is a splitting for s, and $u \succeq maxred_R(s_2\sigma)$.

In both cases, by Lemmas 102 and 103, we have that (s_2, σ) is $(u \succ)$-irreducible and recursively irreducible w.r.t. R. □

Lemma 111 Let t be a summand. Let $(t + s, \sigma)$ be recursively irreducible w.r.t. R. Let $AG\text{-}nf(t\sigma)$ be smaller than or equal to $maxred_R((t + s)\sigma)$ w.r.t. $\succ$. Let t' be a summand such that (t', σ) is recursively irreducible w.r.t. R, and $AG\text{-}nf(t\sigma) \succ AG\text{-}nf(t'\sigma)$.

Then, $(AG\text{-}nf(t' + s), \sigma)$ recursively irreducible w.r.t. R.

Proof: Let u be $maxred_R((t + s)\sigma)$. After replacing t by t', this maximal reducible summand does not increase. Moreover, if u is $maxred(t' + s\sigma)$, then it is due to the same reason as before (top-level positive reduction, or top-level negative reduction or non-top-level reduction). Except for t', the variables and summands that appear in $AG\text{-}nf(t' + s)$ at top-level position are the same ones that appear in $t + s$ at top-level position, and with the same sign. Therefore, the conditions for irreducibility are satisfied for the variables at top-level position. But also for t', since it is recursively irreducible w.r.t. R. □

Lemma 112 Let (s, σ) be recursively irreducible w.r.t. R. Let t be an AG-normal form of $s\sigma$. Let p be an AG-context position in t such that $t|_p$ is reducible w.r.t. R.

Then there exists an AG-context position q in s such that $s\sigma|_q =_{AG} t|_p$, and $(s|_q, \sigma)$ is recursively irreducible, and for all term r, $s[r]_q\sigma =_{AG} t[r\sigma]_p$.

Moreover, let (r, σ) be recursively irreducible w.r.t. R, and let $t|_p \succ AG\text{-}nf(r\sigma)$.

Then $(AG\text{-}nf(s[r]_q), \sigma)$ is recursively irreducible w.r.t. R.

Proof: This is proved by induction on the size of s. In the case where p is λ, q is λ, and all the results are obvious. Therefore, suppose that p is not λ. Then, p is of the form $p'.p''$, where $t|_{p'}$ is a summand of the form $f(t_1, \ldots, t_n)$ at the AG-context λ. Let u be $maxred(t)$. Since $t|_{p'}$ is reducible, we have that $u \succeq t|_{p'}$. An AG-context of $t|_{p'}$ is reducible, and therefore there is an i such that t_i is reducible, and p is of the form $p'.i.p'''$.

Since t is an AG-normal form of $s\sigma$, we have that, either (i) s is of the form $x+s'$ or $-x+s'$, and $x\sigma$ is of the form $t|_{p'}+s''$ or $-t|_{p'}+s''$; or (ii) s is of the form $v+s'$ for some summand v such that $v\sigma =_{AG} t|_{p'}$ and t is of the form $t|_{p'}+t'$; or (iii) s is of the form $-v+s'$ for some summand v such that $v\sigma =_{AG} t|_{p'}$ and t is of the form $-t|_{p'}+t'$.

In case (i), $x\sigma$ is of the form $mt|_{p'}+s''$, and $t|_{p'}$ is reducible at non-top position by R, and $maxred_R(t) \succeq t|_{p'}$, but $t|_{p'}$ cannot be $maxred_R(t)$ by top-level reduction (observe that for the rules $nu \to r$ of R such u's are irreducible at non-top by R). Altogether this contradicts the hypothesis of recursive-irreducibility, and therefore, only cases (ii) and (iii) are possible. In fact, we consider only case (ii), since case (iii) is analogous. The summand v has to be of the form $f(v_1,\ldots,v_n)$, and $v_i\sigma =_{AG} t_i$. By induction hypothesis, there exists an AG-context position q' in v_i such that $v_i\sigma|_{q'} =_{AG} t_i|_{p'''}$, and for all term r, $v_i[r]_{q'}\sigma =_{AG} t[r\sigma]_{p'''}$. Moreover, if (r,σ) is recursively irreducible w.r.t. R, and $t|_p \succ AG\text{-}nf(r\sigma)$, we have that $(AG\text{-}nf(v_i[r]_{q'}),\sigma)$ is recursively irreducible w.r.t. R. Moreover, $(f(v_1,\ldots,AG\text{-}nf(v_i[r]_{q'}),v_n),\sigma)$ is recursively irreducible w.r.t. R. Finally, by Lemma 111, $(f(v_1,\ldots,AG\text{-}nf(v_i[r]_{q'}),v_n)+s',\sigma)$ is recursively irreducible w.r.t. R. □

Lemma 113 Let (s,σ) be irreducible w.r.t. R. Let t be an AG-normal form of $s\sigma$. Let p be an AG-context position in t different from λ such that $t|_p$ is reducible w.r.t. R.

Then there exists an AG-context position q in s different from λ, such that $s|_q\sigma =_{AG} t|_p$, and $(s|_q,\sigma)$ is recursively irreducible, and for all term r, $s[r]_q\sigma =_{AG} t[r\sigma]_p$.

Moreover, let (r,σ) be recursively irreducible w.r.t. R, and let $t|_p \succ AG\text{-}nf(r\sigma)$. Then $(AG\text{-}nf(s[r]_q),\sigma)$ is irreducible w.r.t. R.

Proof: The proof is analogous to the previous one, except for the fact that, instead of doing induction, it refers to the previous lemma, and that we need a modification of Lemma 111 for dealing with irreducible pairs instead of recursively irreducible pairs w.r.t. R. □

Lemma 114 Let (s,σ) be $(u\succeq)$-irreducible w.r.t. R.

If $maxred_R(s\sigma)$ is smaller than or equal to u, then (s,σ) is recursively irreducible w.r.t. R.

If the maximal summand of $AG\text{-}nf(s\sigma)$ is smaller than or equal to u, then (s,σ) is irreducible w.r.t. R.

Proof: Direct by applying the definition. □

Lemma 115 Let (r, σ) be $(u \succeq)$-irreducible w.r.t. R. Let (t, σ) be $(u \succeq)$-irreducible w.r.t. R.

Then, $(AG\text{-}nf(r+t), \sigma)$ is $(u \succeq)$-irreducible w.r.t. R.

Additionally, suppose that $maxred_R((r+t)\sigma)$ is smaller than or equal to u w.r.t. $\succ$. Then $(AG\text{-}nf(r+t), \sigma)$ is recursively irreducible w.r.t. R.

Moreover, if the maximal summand of $AG\text{-}nf((r+t)\sigma)$ is smaller than or equal to u, then $(AG\text{-}nf(r+t), \sigma)$ is irreducible w.r.t. R.

Proof: Observe that r and t are in AG-normal form. Therefore, the AG-normal form of $r+t$ is obtained by eliminating some summands at the AG-context λ, by the inverse rule. If $AG\text{-}nf(r+t)$ is of the form $x+s$ or $-x+s$, then, either r or t is of the form $x+s'$ or $-x+s'$, and, therefore, $x\sigma$ satisfies the corresponding requirements. If $AG\text{-}nf(r+t)$ is of the form $v+s$ or $-v+s$ for a given summand $v = f(v_1, \ldots, v_n)$ such that $u \succeq AG\text{-}nf(v\sigma)$, then, either r or t is of the form $v+s'$ or $-v+s'$, and hence such a v satisfies the corresponding requirements. Therefore $(AG\text{-}nf(r+t), \sigma)$ is $(u \succeq)$-irreducible w.r.t. R.

The rest of the proof is a direct consequence of Lemma 114. □

Lemma 116 Let $n > 1$, and $-u \to (n-1)u - r'$ be a rule of R. Let $AG\text{-}nf(r\sigma)$ be $(n-1)u - r'$.

Let (r, σ) be $(u \succ)$-irreducible w.r.t. R, and if r is of the form $x+s$, then, $x\sigma$ is not of the form $-u+s'$.

Let (t, σ) be $(u \succ)$-irreducible, and recursively irreducible w.r.t. R, and $AG\text{-}nf(t\sigma)$ is not of the form $u+s'$.

Let $maxred_R(t\sigma)$ be smaller than or equal to u w.r.t. $\succ$.

Then, $(AG\text{-}nf(r+t), \sigma)$ is recursively irreducible w.r.t. R.

Proof: Since r and t are in AG-normal form, the AG-normal form of $r+t$ is obtained by eliminating some summands at the AG-context λ, by the inverse rule.

Observe that, since $AG\text{-}nf(r\sigma)$ is $(n-1)u - r'$, and $AG\text{-}nf(t\sigma)$ is not of the form $u+s'$, it holds that $AG\text{-}nf((r+t)\sigma)$ is of the form $mu + s''$ or s'', where s'' does not contain u's at the AG-context λ, and m is negative, or positive but smaller than n.

If m is positive, or $AG\text{-}nf((r+t)\sigma$ is of the form s'', then, $maxred_R((r+t)\sigma)$ is a certain v smaller than u w.r.t. $\succ$, and $(AG\text{-}nf(r+t), \sigma)$ is $(v \succeq)$-irreducible w.r.t. R, since both (r, σ) and (t, σ) are $(u \succ)$-irreducible. By Lemma 114, $(AG\text{-}nf(r+t), \sigma)$ is recursively irreducible w.r.t. R.

From now on, we assume that $AG\text{-}nf((r+t)\sigma)$ is of the form $mu + s''$, for a given negative m. In this case, $AG\text{-}nf(t\sigma)$ contains more than $n-1$ negative u's, and hence $maxred_R(t\sigma)$ and $maxred_R(r + t\sigma)$ has to be u by top-level negative reduction.

If $AG\text{-}nf(r+t)$ is of the form $x+s$ and $x\sigma$ is of the form $kv + v'$ for a given summand v with $u \succeq v$, then, either r or t is of the form $x + s_1$. In both cases, if

$u \succ v$, then, since both (r, σ) and (t, σ) are $(u \succ)$-irreducible, we have that kv is irreducible w.r.t. R. Therefore assume that $x\sigma$ is of the form $ku+v'$ (i.e. v is u), and then, for satisfying the recursive-irreducibility conditions it is enough to show that k is positive. If r is of the form $x + s_1$, by our hypothesis k is positive. If t is of the form $x + s_1$ then, k is positive due to the fact that (t, σ) is recursively irreducible, and u is $maxred_R(t\sigma)$ determined by top-level negative reduction.

If $AG\text{-}nf(r + t)$ is of the form $-x + s$, and $x\sigma$ is of the form $kv + v'$ for a given summand v with $u \succeq v$, then, either r or t is of the form $x + s_1$. In both cases, if $u \succ v$, then, since both (r, σ) and (t, σ) are $(u \succ)$-irreducible, we have that kv is irreducible w.r.t. R. Otherwise, if v is u, the recursive-irreducibility conditions for such $-x + s$ and kv are satisfied trivially.

If $AG\text{-}nf(r+t)$ is of the form $v+s$ or $-v+s$ for a given summand $v = f(v_1, \dots, v_n)$ such that $u \succeq AG\text{-}nf(v\sigma)$, then, either r or t is of the form $v + s'$ or $-v + s'$. Since both (r, σ) and (t, σ) are $(u \succ)$-irreducible, it holds that all the (v_i, σ) are recursively irreducible w.r.t. R. □

Theorem 117 $\mathcal{H}$ is refutation complete for constrained Horn clauses if the initial set of clauses has only empty constraints.

Proof: This proof is analogous to the one for Theorem 94. The diferences are in how it is proved that $I \models Ir_{R_S}(S)$.

Let $Cred$ be the minimal, w.r.t $\succ_c$, reductive form of some $C\sigma$ in $Ir_{R_S}(S)$ that is an instance of a clause $C \mid T_C$ such that $I \not\models Cred$.

If $Cred$ is a disjunction of literals of the form $0 \not\simeq 0$, then an inference by AG-zero-instance applies to any one of these literals, eliminating it, and its conclusion has a smaller false counter example.

Otherwise, as in the ground case (the proof of Theorem 73), let s be the maximal summand in $Cred$. Then $Cred$ is either of the form $Cred' \vee ms \simeq t$ with $s \succ Cred'$ (a), or else it is $Cred' \vee ms \not\simeq t$ with $s \succeq Cred'$ (b). As in Theorem 73, in both cases ms is reducible by R. Then, by Lemma 106 there exists an AG-context s' that is a subterm of ms, and a summand u such that u is $maxred_R(s')$ by top-level reduction. Therefore, a rule in R of the form $nu \to r'$ or $-u \to (n-1)u - r'$ reduces s', and it has to be $nu \to r'$ if s' is ms; and moreover, no rule with left-hand-side bigger reduces s'.

Therefore, C is of the form $C' \vee e \simeq 0$ or $C' \vee e \not\simeq 0$, where $ms-t$ is an AG-normal form of $e\sigma$.

The rule reducing $ms - t$ (at the AG-context $ms - t$ or in an AG-context inside s), has been generated by the reductive form $Dred$ of an instance $D\sigma$ of a clause $D \mid T_D$. Let D be of the form $D' \vee d \simeq 0$. Now, we distinguish two cases:

- (a) If the rule reducing ms is $nu \to r'$, then, by Lemma 107, there exists an orientation $l \simeq r$ of $d \simeq 0$ such that $AG\text{-}nf(l\sigma)$ is nu and $AG\text{-}nf(r\sigma)$

is r'. Moreover, (r,σ) is $(u\ \succeq)$-irreducible w.r.t. R. Now, we analyse two possibilities:

- (a.1) If s' is ms, then s is u, and $AG\text{-}nf(e\sigma)$ is $mu - t$, for $m \geq n$. Moreover, u is the maximal summand of $ms - t$ and (e,σ) is irreducible w.r.t. R. By Lemma 109, there exists a splitting $e_1 + e_2$ of e such that $(e_1 + e_2)\sigma =_{AG} e\sigma$, and $e_1\sigma$ is nu, and (e_2,σ) is $(u\ \succeq)$-irreducible w.r.t. R, and the maximal summand of $AG\text{-}nf(e_2\sigma)$ is smaller than or equal to u. By Lemma 115, $(AG\text{-}nf(r + e_2),\sigma)$ is irreducible w.r.t. R. Now, the following inference exists:

  $$\frac{D' \vee l \simeq r \mid T_D \qquad C' \vee e_1 + e_2 \simeq 0 \mid T_C}{C' \vee r + e_2 \simeq 0 \mid T_D \wedge T_C \wedge l = e_1 \wedge \tau}$$

  Its conclusion belongs to S, since S is closed under $\mathcal{H}$, and it has an instance with σ contradicting the minimality of *Cred.*

- (a.2) If s' is inside s, i.e. $(ms - t)|_p$ is s' for some position p below some s, then, by Lemma 113, there exists an AG-context position q in e such that $e|_q\sigma =_{AG} s'$, and $(e|_q,\sigma)$ is recursively irreducible w.r.t. R, and for all term r'', $e[r'']_q\sigma =_{AG} (ms-t)[r''\sigma]_p$. Moreover, if (r'',σ) is recursively irreducible and $s' \succ AG\text{-}nf(r''\sigma)$, then, $(e[r'']_q,\sigma)$ is irreducible w.r.t. R. Now, we will obtain the concrete r'' that is interesting for us. Denote $e|_q$ by e'. Observe that e' is recursively irreducible w.r.t. R, and s' is of the form $nu + s''$, and $maxred_R(s')$ is u. By Lemma 109, there exists a splitting $e'_1 + e'_2$ of e' such that $(e'_1 + e'_2)\sigma =_{AG} e'\sigma$, and $e'_1\sigma$ is nu, and (e'_2,σ) is $(u\ \succeq)$-irreducible w.r.t. R, and $maxred_R(e'_2\sigma)$ is smaller than or equal to u. By Lemma 115, $(AG\text{-}nf(r + e'_2),\sigma)$ is recursively irreducible w.r.t. R. This $AG\text{-}nf(r + e'_w)$ is the r'' we wanted. Now, the following inference exists:

  $$\frac{D' \vee l \simeq r \mid T_D \qquad C' \vee e[e'_1 + e'_2]_q \simeq 0 \mid T_C}{C' \vee e[r + e'_2]_q \simeq 0 \mid T_D \wedge T_C \wedge l = e'_1 \wedge \tau}$$

  Its conclusion belongs to S, since S is closed under $\mathcal{H}$, and it has an instance with σ contradicting the minimality of *Cred.*

- (b) If the rule reducing ms is $-u \rightarrow (n-1)u - r'$, then, the contradiction of the minimality of *Cred* follows, now, from Lemmas 108, 113, 110, and 116; in a similar way to case (a.2). □

7.7 General Clauses

The inference system is extended to non-Horn clauses in the standard way, with (equality) factoring, which in the ground case is:

$$\textit{AG-factoring:} \qquad \frac{C \vee nu \simeq r \vee nu \simeq r'}{C \vee r \not\simeq r' \vee nu \simeq r'}$$

with the ordering restrictions that u is the maximal summand in the clause, which does not appear in a negative equation, and where $nu \simeq r$ is maximal w.r.t. $\succ_e$.

For the non-ground case, the two equations involved have to be oriented as the left premises of AG-superposition (note that if both orientations require to split a certain variable x, then it needs to be split only once). Let us denote by $\mathcal{I}$ the rules of $\mathcal{H}$ (with the same ordering restrictions as the factoring rule) plus this additional rule. By a relatively standard adaptation of the rule generation with respect to the Horn case (i.e., as for standard superposition, see [BG94b]), we obtain the following:

Theorem 118 The inference system $\mathcal{I}$ is refutation complete for general clauses.

7.8 Extensions

We now very briefly comment on a few aspects that have not been treated yet in this chapter.

Our completeness proofs are compatible with the notions for redundancy and saturation as in the *basic* framework [NR95, BGLS95]. Note that, by dealing with constrained clauses, no AG-unifiers are computed. Instead, the unification problems are stored in the constraints and a constrained clause $C \mid T$ is redundant if T is unsatisfiable. Apart from the well-known basicness restriction, an additional advantage is that only one conclusion is generated, instead of one conclusion for each AG-unifier[Vig94, NR97].

Checking the ordering restrictions in our framework is different from the usual situation. Instead of checking whether, say, for given terms s and t, there exists some ground σ such that $s\sigma \succ_{rpo} t\sigma$, we need to check whether this holds after normalising both sides by R_{AG}, that is, whether $AG\text{-}nf(s\sigma) \succ_{rpo} AG\text{-}nf(t\sigma)$.. Deciding the satisfiability of such constraints is NP-complete, as we will see in Chapter 8. One can also add information to the constraint language defined in that chapter for stating that if $n_1 s_1 + \ldots + m_1 y_1 + \ldots$ is the left hand side of an orientation (Definition 78) then all s_i are equal and all summands in the y_i are equal to these s_i.

It is also possible to find sufficient conditions for ruling out redundant inferences without fully deciding satisfiability. In practice, for efficiency reasons, such approximations are used as well for standard superposition. Neither soundness nor completeness require to actually decide ordering constraints.

Example 119 Suppose s is $f(f(0)-x)$ and t is x. It is easy to see that $s\sigma \succ_{rpo} t\sigma$ for all σ. But if σ is $\{x \to f(0)\}$, both terms normalise w.r.t. R_{AG} into $f(0)$. □

The fact that ordering restrictions are checked after normalisation w.r.t. R_{AG} complicates optimisations related to the analysis of the so-called *shielded* variables of a clause C, that is, variables that occur below a free symbol in C.

Example 120 In the context of [GW96, Stu98], shieldedness of variables like x in the clause $f(x-f(a)) \not\simeq 0 \vee 2x \simeq b$ allow one to conclude that $2x$ cannot contain the maximal summand of $C\sigma$ for any σ and hence $2x$ need not be used as left premise in any inference. In our case, the instance where $x\sigma$ is $f(a)$ may generate the rule $2f(a) \to b$, and hence we can rule out the inferences only for other instances. Similar optimisations apply to right premises. □

Also other shieldedness-related optimisations can be used. For example, let $e \simeq 0$ be an equation of a clause C where e is of the form $s + n_1x_1 + \ldots + n_kx_k \simeq 0$ and the distinct variables x_i do not occur elsewhere in s or in C. If $n_i = 1$ (or $n_i = -1$) for some i, then such an equation $e \simeq 0$ collapses the theory: $s + x \simeq 0$ implies $s + (-s+t) \simeq 0$ and hence $t \simeq 0$ for every t. Hence one can assume that any such a clause $C \vee s+x \simeq 0$ is eagerly replaced by C. This can be combined with the fact that $e \simeq 0$ is logically equivalent modulo AG to $s + nz \simeq 0$, where $n = gcd(n_1, \ldots, n_k)$ and z is a new variable.

Chapter 8

Ordering Constraints for Built-in Abelian Groups

We have mentioned in previous chapters that 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.

In this chapter 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.

8.1 Introduction

As we have mentioned already several times in this thesis, 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). We have seen that 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 [KKR90]. This allows one to reduce the search space by inheriting the ordering restrictions while keeping completeness [NR95, NR01].

Let us recall here that 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 [Der82], for fixed signatures [Com90, JO91] and extended ones [NR95, Nie93]. NP algorithms (fixed and extended signatures) were given in [Nie93, NRV99]. For the Knuth-Bendix ordering (KBO) these results have only recently been obtained in [KV00, KV01].

Ordered strategies and ordering constraint inheritance can be used without loosing completeness with built-in algebraic theories E, like AC [NR97, Vig94] or AG [GN00]. 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 [McC97c] 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 [RN95] a constraint solving algorithm exists [CNR95]. 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 [GN00], 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 chapter is structured as follows. After the basic definitions of Section 8.2, in Section 8.3 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 8.4 is on pure FRPO constraints. Then, after explaining the relatively simple extension to AC0-RPO in Section 8.5, in Section 8.6 we deal with the technically more complex part of the chapter, namely the techniques for AG-RPO.

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 8.7, 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.

8.2 Basic Definitions

In this chapter 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 $-u + t'$.

As seen in chapter 7, the rewrite system R_{AG} consists of the following five rules:

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

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 $\{x+0 \rightarrow x\}$ 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 chapter, rewriting with a set of rules R is always considered *modulo* AC. For instance, when writing $\rightarrow_{R_{AG}}$, we mean the relation $=_{AC} \rightarrow_{R_{AG}} =_{AC}$. 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 $=_{AC}$) normal form w.r.t. R_{AG}; see, e.g., [Mar96].

Furthermore, as usual, terms will always be (eagerly) considered in *flattened form* w.r.t. AC. As seen in chapter 7, 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 *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.*

Definition 121 Let $>$ be a precedence and assume $mul = \{+\}$. The *RPO on flattened terms*, denoted by FRPO, is defined as follows:
$s \succ_{frpo} t$ if *flat*$(s) \succ_{rpo}$ *flat*(t).

Example 122 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 chapter, but we give it here for showing the applicability of FRPO in practice:

Property 123 [BP85] 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 $=_{AC}$.

Definition 124 The AC0-RPO and AG-RPO orderings are defined as follows. Let s and t be two ground terms. We define:

$$\begin{array}{lll} s \succ_{acOrpo} t & \text{if} & nf_{R_0}(s) \succ_{frpo} nf_{R_0}(t) \\ & \text{and} & \\ s \succ_{agrpo} t & \text{if} & nf_{R_{AG}}(s) \succ_{frpo} nf_{R_{AG}}(t) \end{array}$$

The following results are stated here again only for showing that AC0-RPO and AG-RPO are also useful for practical applications like [GN00]. They are not difficult to prove (see also [GN00]).

Property 125 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 126 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.

8.3 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 127 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 $\{x_1, \ldots, x_n\} = \text{vars}(S)$, 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 128 [Com90, Nie93]

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 chapter, 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 129 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 $(y_1, \ldots, y_m) >^{lex}_S (z_1, \ldots, z_m)$.

$A8$. 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 130 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 128, 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 $A4$-$A8$ 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 131 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.*

8.4 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 chapter.

Definition 132 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:

$AC1$. If some $t_{i,j}$ is $y_1 + \ldots + y_m$ then $t_{i,j} >_S y_i$ for all i in $1 \ldots m$.

$AC2$. 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 133 Any linear system S that fulfills the global assumptions and that is satisfiable w.r.t. FRPO also fulfills the FRPO assumptions.

Example 134 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. □

8.4.1 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 143 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 135 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 136 (Example 134 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 134. 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 137 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 138 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 139 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.

8.4.2 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 140 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. $x_j = t_{j,k}$, for all j in $1\ldots i$ and all k in $1\ldots k_j$
3. the equation $x_{i+1} = 1$.

Example 141 (Example 136 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 134: $w_2'\sigma$ is 0, and $z\sigma$ is $f(0) + f(0)$. □

8.4.3 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 142 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

$$M_0 \succ\!\!\succ \ldots \succ\!\!\succ M_m$$

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 143 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 142 there exists a function $f : C \to \mathcal{N}$ such that its extension F to multisets satisfies $F(x_1\sigma) \succ\!\succ \ldots \succ\!\succ F(x_{i+1}\sigma)$, 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 144 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 143, 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 [Sch87]. □

8.5 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 145 The satisfiability problem for AC0-RPO constraints is in NP.

8.6 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 146 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 147 We have $f(f(a)) \succ f(a - f(0) + f(a - a))$ since

$nf_{R_{AG}}(f(f(a))) = f(f(a)) \succ_{FRPO} f(a) = nf_{R_{AG}}(f(a - f(0) + f(a - a)))$. □

Example 148 Terms can be smaller than their subterms: $\sigma \models x > f(x - f(a))$ if $x\sigma = f(a)$, since $nf_{R_{AG}}(f(a)) = f(a) \succ_{FRPO} f(0) = nf_{R_{AG}}(f(f(a) - f(a)))$. □

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 149 $\sigma \models x - y = f(z)$ 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 150 Suppose that we have a signature of the form $f > - > + > 0$ where f is unary. Then the smallest terms are ordered like:

$0,\ f(0),\ f(0){+}f(0),\ f(0){+}f(0){+}f(0),\ \ldots,$

${-f(0)},\ {-f(0)}{-}f(0),\ {-f(0)}{-}f(0){-}f(0),\ \ldots,\ f(f(0))$.

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. □

8.6.1 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 151 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.

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 152 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.

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 153 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 154 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 138) plus the following additional assumption:

*AG*3. 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 155 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 156 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 $x_2-x_1-x_0 = 0$. □

Diophantine equations.

Example 150 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 157 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 158 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 159 We have $\#(a, f(a+b)-a-a) = -2$. □

Definition 160 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 161 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 162 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 $u_{ndec} \succ_{agrpo} \cdots \succ_{agrpo} u_1$.

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 163 Consider $f > h > - > + > 0$ and suppose that after the splitting transformation we have:

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

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 151 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 164 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 $\{1 \dots i\}$, and all k in $\{1 \dots ndec\}$, and all l in $\{1 \dots k_j\}$. We proceed identically with the associated equations.

3. We add $x_{i+1,1} = 1$, and for all k in $\{2 \dots ndec\}$ 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 163). 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. □

Constraint Solving.

Lemma 165 Let S be a linear system satisfying the AG-RPO assumptions. Let $D_1 \dots D_m$ 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

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

for all j in $\{1 \dots i+1\}$ 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 166 The satisfiability problem for AG-RPO constraints restricted to signatures with free symbols of arity 0 or 1 is in NP.

8.6.2 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 167 The satisfiability problem for AG-RPO constraints is in NP.

8.7 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 168 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 $t_1{+}t_1{+}t_2$. □

8.8 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 Chapter 7. 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 Chapter 7. For example, as we saw in that chapter, 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 Chapter 7 can be combined with the ones presented here for obtaining an efficient inference system for abelian groups.