The Blossom of Finite Semantic Trees

Jean Goubault-Larrecq[1⋆] and Jean-Pierre Jouannaud[2 ⋆⋆ ⋆⋆⋆]

[1] LSV/CNRS UMR 8643 & INRIA Futurs projet SECSI & ENS Cachan
61 avenue du président-Wilson, F-94235 Cachan Cedex
Email: goubault@lsv.ens-cachan.fr
Web: http://www.lsv.ens-cachan.fr/~goubault/
[2] LIX/CNRS FRE 2653 & École Polytechnique, F-91400 Palaiseau
Email: jouannaud@lix.polytechnique.fr
Web: http://www.lix.polytechnique.fr/Labo/Jean-Pierre.Jouannaud/

This paper is dedicated to the memory of Harald Ganzinger.

1 Introduction

Automated deduction in first-order logic finds almost all its roots in Herbrand's work, starting with Herbrand's interpretations, a clausal calculus, and rules for unification. J.A. Robinson's key contribution was the formulation of resolution and its completeness proof, in which semantic trees were semi-apparent. Robinson and Wos introduced the specific treatment of equality commonly called paramodulation. The systematic introduction of orderings to cut the search space is due to Lankford. Kowalski studied in more details the case of Horn clauses, while Peterson gave the first proof that paramodulation inside variables was superfluous, assuming a term ordering order-isomorphic to the natural numbers. Knuth studied the case of equality unit clauses, under the name of completion. All these works were done by using standard proof techniques, including semantic trees [Kow69].

Further progress required more powerful proof techniques.

The first was proposed by Huet with Noetherian orderings on terms, allowing the use of the powerful noetherian induction principle to establish a strong theory of abstract and concrete rewriting, another name for the case of equality unit clauses [Hue80]. The

⋆ Partially supported by the RNTL project Prouvé and the ACI Rossignol.

⋆⋆ Project LogiCal, Pôle Commun de Recherche en Informatique du Plateau de Saclay, CNRS, École Polytechnique, INRIA, Université Paris-Sud.

⋆⋆⋆ Harald Ganzinger and I had the opportunity to closely collaborate within the European projects *COMPASS* and *Combimation of Computational Logics*. As a member of the Fachbeirat of the Max Planck Institute für Informatik in Saarbrücken for 6 years from the beginning, I had also the privilege to closely collaborate with Harald in his dedication to make the institute stronger.

method was then extended by Jouannaud and Kirchner who introduced induction on proofs abstracted by multisets of terms [JK86]. Bachmair, Dershowitz and Hsiang made the last step with the proof reduction method [BD94]. This tool allowed this subfield to make very fast progress until a new bottleneck was encoutered with constrained equality unit clauses.

The second proof method was proposed by Hsiang and Rusinowitch [HR86], who invented transfinite semantic trees, a generalization of semantic trees generated from a transfinite ordering on the Herbrand base. They were able to generalize Peterson's result to arbitrary well-founded orderings. Considering again the case of equality unit clauses, they showed the completeness of ordered completion, an old conjecture of Lankford, which was found to have many theoretical applications by providing with a true semi-decision procedure for equality based on computing normal forms. Being conceptually complex constructions, transfinite semantic trees did not make their way through in the scientific community.

The third was proposed by Bachmair and Ganzinger, which allowed to make tremendous progress in all directions ever since, to a point that people did not find the need to look for new methods. Bachmair and Ganzinger's model generation technique [BG01a] is based on *forcing* a specific interpretation which can be seen as characterizing the satisfiability property of a given set of clauses. Many groups throughout the world studied and used this method, which was found a bit mysterious at first. Our goal here is to shed a new light on this important approach, by adopting a presentation based on semantic trees which we think is easy to grasp.

As transfinite semantic trees, Bachmair and Ganzinger's model generation technique is based on a well-founded ordering on terms which can be transfinite. It aims at showing the refutation-completeness property of a set of inference rules $\mathcal{I}$ used for generating the empty clause from a given unsatisfiable set $\mathcal{S}$ of clauses. The ordering is used to restrict the possible inferences to those involving maximal atoms.

Our first problem was to construct finite semantic trees with transfinite orderings. The answer is provided by Herbrand's compactness theorem[3]: only finitely many ground instances of $\mathcal{S}$ suffice.

[3] The solution was hinted at by Michael Rusinowitch in a discussion with the first author.

These ground instances generate finitely many atoms which define interpretations which are finitely refuted, hence a finite semantic tree. A consequence of this construction is that the ordering need not be total, nor well-founded: it needs only be strict. It can then be completed into a total strict ordering on the finite set of atoms. The well-foundedness assumption becomes however necessary in presence of an equality predicate.

Our second problem was to guess which node in the semantic tree of an unsatisfiable set of ground clauses would allow us to make an inference. The answer is easy: the model generation technique builds an interpretation which defines indeed a path in the semantic tree ending in an inference node.

The third problem was to show that this inference decreases the semantic tree in some well-founded ordering, allowing to conclude by induction that the tree could be reduced in finite time to its root, hence showing that the empty clause had been generated. Building well-founded orderings on the semantic tree is much easier than on the set of clauses itself, allowing us again to slightly improve over the existing literature in some cases.

We do not think that our contribution lies in any improvement over the current literature. Our first main contribution, as we feel, is to show that all these concepts elaborated by Ganzinger and his collaborators are *intrinsic* to the entire field of automated deduction, rather than *specific* to his model generation proof method as one might have thought. And the second is the use of a single proof method to obtain them all, suggesting that some of these restrictions may be combined. We will treat here a few basic results only: ordered resolution, ordered resolution with selection, ordered linear resolution, and ordered resolution and paramodulation. We consider the systematic use of our technique as an exercice which will allow the reader to better grasp the subtleties of Ganzinger's work.

2 Ordered Resolution with Selection

The semantic tree technique makes it relatively clear that not only resolution is complete, but also *ordered* resolution, where only literals that are maximal in their respective clauses are resolved upon [CL73]. This is a very effective restriction of resolution. We recall the completeness argument for ordered resolution in Section 2.1. We also improve it, by showing that ordered resolution is complete for any stable ordering (even, say, not well-founded).

Another very effective restriction is ordered resolution with *selection*, where a selection function is used to denote selected exceptions to the ordering restriction. This refinement of resolution generalizes both ordered resolution and hyperresolution. It has been known for a long time to resist semantic tree arguments, and Bachmair and Ganzinger's forcing technique [BG01a] provided an elegant completeness argument. We show how the two techniques blend naturally together in Section 2.2. In Section 2.3, we deal briefly with redundancy elimination strategies, an important part of Bachmair and Ganzinger's work in automated deduction. We sketch how our technique generalizes to the completeness of linear resolution in Section 2.4, a refinement of resolution whose completeness was traditionally thought to require different arguments.

2.1 Ordered Resolution

Let $\gtrsim$ be any stable quasi-ordering on atoms which restricts to an ordering on ground atoms. By *stable*, we mean that for any two atoms A, B, if $A \gtrsim B$, then $A\sigma \gtrsim B\sigma$ for every substitution σ. Let $\lesssim$ be the converse of $\gtrsim$, $\succ$ the strict part of $\gtrsim$, and $\prec$ the converse of $\succ$. The rule of *ordered resolution* is as follows, where clauses are sets of literals separated by $\vee$, and the two premises are assumed renamed, without loss of generality, so as to have no variable in comon.

$$\frac{+A_1 \vee \ldots + A_m \vee C \quad -A'_1 \vee \ldots \vee -A'_{m'} \vee C'}{C\sigma \vee C'\sigma} \quad \begin{array}{c} m \geq 1, n \geq 1, \\ \sigma = mgu(A_1 = A_2 = \ldots = A_m = \\ A'_1 = \ldots = A'_{m'}), \\ A_i\sigma \not\precsim B, A'_{i'}\sigma \not\precsim B \ \forall B \in C\sigma \vee C'\sigma, \\ 1 \leq i \leq m, 1 \leq i' \leq m' \end{array}$$

We write $mgu(E)$ the most general unifier of any given set of term equations. As usual, we let σ be *more general than* θ if and only if $\theta = \sigma\sigma'$ for some substitution σ', and we write $\sigma \sqsubseteq \theta$.

Ordered resolution is sound and complete, in the sense that, starting from a set S of clauses, we may deduce the empty clause $\Box$ by finitely many applications of the above rule if and only if S is unsatisfiable. We may in fact restrict m' to be 1 (no negative factoring), or m to be 1 (no positive factoring) without breaking completeness, but not both. Alternative presentations split this rule in one binary ordered resolution rule, and additional positive/negative factoring rules. We shall do this in later sections. For now, the current presentation will be more practical.

Soundness is trivial. Completeness is, of course, harder, so let's start by showing how semantic trees can be used to show that ordered resolution is complete when $\succsim$ is *enumerable*, i.e., when it satisfies the following property:

$(*)$ there is an enumeration $A_1^0, A_2^0, \ldots, A_i^0, \ldots$ of ground atoms such that $i > j$ whenever $A_i^0 \succ A_j^0$.

This much had been known since [Joy76]. Plain, unordered resolution, will in particular be complete, since this is the case where $\succsim$ is just the equality relation on atoms, which is clearly enumerable. We shall show that property $(*)$ is not required later.

Let $A_1^0, A_2^0, \ldots, A_i^0, \ldots$ be any given enumeration of ground atoms satisfying $(*)$. A *partial interpretation* I on this enumeration is a finite list $\pm_1 A_1^0, \pm_2 A_2^0, \ldots, \pm_k A_k^0$. If A_i^0 occurs under the $+$ sign, then A_i^0 is *true* in I; A_i^0 is *false* if it occurs under the $-$ sign, and *undefined* otherwise.

The *Herbrand tree* is the binary tree whose vertices are partial interpretations. The partial interpretation $I = \pm_1 A_1^0, \pm_2 A_2^0, \ldots, \pm_k A_k^0$ has two sons $\pm_1 A_1^0, \pm_2 A_2^0, \ldots, \pm_k A_k^0, -A_{k+1}^0$ and $\pm_1 A_1^0, \pm_2 A_2^0, \ldots,$ $\pm_k A_k^0, +A_{k+1}^0$—provided A_{k+1}^0 exists, otherwise I is a leaf. The root of the tree is the empty partial interpretation ϵ.

The maximal paths of the Herbrand tree are naturally in bijection with Herbrand interpretations., i.e., sets of ground atoms. If I_H is a Herbrand interpretation, we follow the maximal path going through ϵ, then $\pm_1 A_1^0$, then $\pm_1 A_1^0, \pm_2 A_2^0$, ..., where $\pm_i$ is $+$ if $A_i^0 \in I_H$, $-$ otherwise. Conversely, any path goes through nodes that mention each atom A with a unique sign; collect those that occur with the $+$ sign, thus defining a Herbrand interpretation.

Figure 1 shows a (finite) semantic tree on the three atoms r, q, p in this order. I.e., $A_1^0 = r$, $A_2^0 = q$, $A_3^0 = p$. Vertex **1** is the empty partial interpretation ϵ, vertex **2** is $-r$, **3** is $+r$, **4** is $-r, -q$, etc.

Let us say that a ground clause C is *false* at vertex $I = \pm_1 A_1^0,$ $\pm_2 A_2^0, \ldots, \pm_k A_k^0$ if and only if, for every literal $\pm A$ of C, the opposite literal $\mp A$ is listed in I. In Figure 1, the clause $+r \vee -q$ is false at $-r, +q$ (vertex **5**), and also, say, at $-r, +q, -p$ (vertex **10**).

Let S be an unsatisfiable set of clauses: for every Herbrand interpretation I_H, there is a ground instance $C\theta$ of a clause $C \in S$

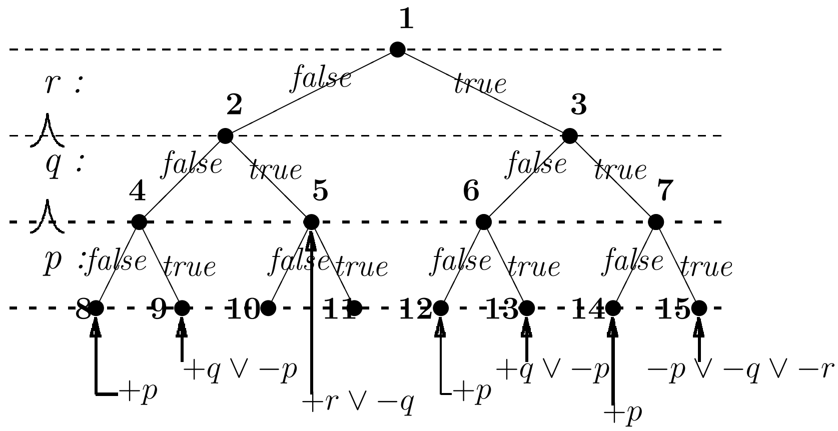


Fig. 1. A semantic tree

such that I_H makes $C\theta$ false. Since the value of $C\theta$ depends on the truth value of only finitely many atoms, there is a partial interpretation, i.e., a vertex along I_H where $C\theta$ is false—e.g., vertex **10** makes $+r \vee -q$ false, assuming $+r \vee -q$ is a ground instance of some clause in S. A *failure node* is any highest vertex in the Herbrand tree that makes some ground instance $C\theta$ of some clause $C \in S$ false.

By König's Lemma, if S is unsatisfiable, then the *closed tree* T_S obtained from the Herbrand tree by cutting it at failure nodes is finite. The compactness theorem for first-order logic obtains easily: only finitely elements of S account for the finitely many leaves of T_S.

Given a finite closed tree T_S, either the root ϵ is a failure node, so that S must contain the empty clause $\Box$; or there must be a lowest non-failure vertex I, called an *inference node*. For example, $-r, -q$ (vertex **4**) in Figure 1 is an inference node. Its two sons, which must be of the form $I, -A$ and $I, +A$ respectively, must be failure nodes for some ground instances of first-order clauses C_+ and C_- respectively, in S, say $C_+\theta_+$ and $C_-\theta_-$. By the definition of failure nodes, $C_+\theta_+$ must be a disjunction of $+A$ with some literals above A (i.e., appearing before A in the enumeration A^0_1, A^0_2, ...), and $C_-\theta_-$ must be a disjunction of $-A$ with some literals above A again. Write C_+ as $+A_1 \vee \ldots + A_m \vee C$, where $+A_1, \ldots, +A_m$ are the literals L in C_+ such that $L\theta_+ = +A$, and write C_- as $-A'_1 \vee \ldots \vee -A'_{m'} \vee C'$, where $-A'_1, \ldots, -A'_{m'}$ are the literals L' in C_- such that $L'\theta_- = -A$. Renaming apart the free variables of C_+

and C_-, in particular, $A_1, \ldots, A_m, A'_1, \ldots, A'_n$ are unifiable. Call σ their most general unifier; since $\succsim$ is stable, and using assumption $(*)$ above, $A_i\sigma \not\precsim B$ and $A'_{i'}\sigma \not\precsim B$ for every atom B in $C\sigma \vee C'\sigma$, $1 \leq i \leq m$, $1 \leq i \leq m'$. So the ordered resolution rule applies, and we may generate the resolvent $C\sigma \vee C'\sigma$. E.g., in Figure 1, the inference node $-r, -q$ (vertex **4**) allows one to resolve between the two clauses whose respective ground instances decorate the failure nodes below it, namely $+p$ and $+q \vee -p$, yielding a clause with $+q$ as ground instance.

Let S' be S union $C\sigma \vee C'\sigma$. Since $C\sigma \vee C'\sigma$ is now false at the inference node I, $T_{S'}$ is a closed tree with strictly less vertices than T_S. This process must therefore terminate; then ϵ will be a failure node, at which point $\Box$ has been inferred: completeness obtains.

There are several degrees of freedom that we can exploit in this argument. First, the usual argument goes by considering the ground instances of clauses in S (which form an unsatisfiable set), showing that propositional ordered resolution is complete for the latter, then lifting propositional resolution refutations to the first-order level by so-called *lifting*. The argument above shows that we can reason directly at the level of first-order clauses, considering ground instances on the fly. While this makes no difference in ordered resolution, this is definitely needed when selection functions are introduced (Section 2.2), because nothing like stability will be required of selection functions.

Second, assumption $(*)$ can be completely dispensed with, as we promised, using compactness: if S is unsatisfiable, then some finite set of ground instances of S is already unsatisfiable. Clearly, this finite set uses only finitely many ground atoms $A_1^0, \ldots, A_n^0$, and we can replay the argument above by using only these atoms. Now it is easy to enumerate them in such a way that $A_i^0 \succ A_j^0$ implies $i > j$, whether $(*)$ holds or not: just find a topological sort of the A_i^0 with respect to the ordering $\succ$. (This is where we are using that $\succsim$ restricts to an ordering on ground atoms.)

Third, the way we pick interesting nodes (here, inference nodes) in the tree clearly dictates what constraints we may add to the resolution rule while retaining completeness. Picking inference nodes is a good match for ordered resolution. Other forms of resolution will require us to find other vertices in T_S. In the context of semantic trees, the import of the Bachmair-Ganzinger forcing method can be seen as a clever way of finding alternative vertices in T_S. This is simple and elegant: any vertex I is just a partial interpretation, and we shall find it by constructing I as a partial interpretation, alternatively as specifying which ground atoms should be true and which should be false while going down the closed tree T_S.

Fourth, and finally, we are free to apply alternative termination arguments. Taking the notations above, we have argued that we could produce a finite ordered resolution refutation by showing that we could rewrite T_S into another closed tree $T_{S'}$ by generating the right ordered resolvent. This terminates because the size $|T_S|$ of T_S is greater than that of $T_{S'}$. However, any well-founded measure of finite closed trees T_S would work equally well. This is precisely what we shall exploit next.

2.2 Ordered Resolution with Selection

Let *sel* be any fixed *selection function*, by which we mean any function that maps each clause C to a possibly empty subset of the negative literals in C—the *selected* literals in C. The idea is that, if $sel\ (C)$ is non-empty, then we require to resolve on all selected literals; if $sel\ (C) = \emptyset$, then we revert to resolving upon $\succ$-maximal literals. On the other hand, we additionally require that the other premise $+A_1 \vee \ldots + A_m \vee C$ contains no selected literal at all.

Again assume a given stable quasi-ordering $\succsim$ whose restriction to ground atoms is an ordering, and assume additionally that $\succ$ is also stable: $A \succ B$ implies $A\sigma \succ B\sigma$ for every atoms A, B, and substitution σ. In case all these conditions are satisfied, we say that $\succsim$ is *strongly stable*. E.g., any reflexive closure $\succeq$ of a strict stable ordering $\succ$—the traditional setting for ordered resolution—is a strongly stable quasi-ordering.

The rule of *ordered resolution with selection* is

$$\frac{\overbrace{C_i \vee +A_{i1} \vee \ldots \vee +A_{in_i}}^{1 \leq i \leq \ell} \quad C' \vee -A'_1 \vee \ldots \vee -A'_\ell}{C_1\sigma \vee \ldots \vee C_\ell\sigma \vee C'\sigma}$$

with the following side-conditions:

(i) $n_i \geq 1$ for every i, $1 \leq i \leq \ell$;
(ii) $\sigma = mgu\{A_{ij} = A'_i | 1 \leq i \leq \ell, 1 \leq j \leq n_i\}$;
(iii) $sel(C_i \vee +A_{i1} \vee \ldots \vee +A_{in_i}) = \emptyset$ and $A_{i1}\sigma \not\precsim B$ for every atom B in $C_i\sigma$, for every i, $1 \leq i \leq \ell$;

(iv) $sel(C' \vee -A'_1 \vee \ldots \vee -A'_\ell) = \{-A'_1, \ldots, -A'_\ell\}$ and $\ell \geq 1$, or no literal is selected, $\ell = 1$ and $-A'_1\sigma \not\succsim B$ for every atom B in $C'\sigma$.

Note that *sel* is *arbitrary*. In particular, imagine that we select $\{-p(X)\}$ in $+q(X) \vee -p(X) \vee -r(X)$. While it would be natural to also select $\{-p(a)\}$ in its instance $+q(a) \vee -p(a) \vee -r(a)$, selection functions are not required in any way to do so, and we may perfectly well choose to select $\{-r(a)\}$, or $\{-p(a), -r(a)\}$, or nothing instead. This fact alone ruins any hope of proving completeness by lifting a completeness argument from the propositional to the first-order case.

Note also that, while we still require positive factoring (in general $n_i \neq 1$) in the *side clauses* $C_i \vee +A_{i1} \vee \ldots \vee +A_{in_i}$, we dispense with negative factoring in the *main clause* $C' \vee -A'_1 \vee \ldots \vee -A'_\ell$.

Theorem 1. *Ordered resolution with selection is complete: for any strongly stable quasi-ordering* $\succsim$*, for any selection function sel, for any set of clauses* S*,* S *is unsatisfiable if and only if we can derive* $\Box$ *from* S *by ordered resolution with selection.*

Proof. We spend the rest of this section proving this. We shall do this in detail here, and go faster in similar proofs in later sections.

The "if" direction is obvious. Conversely, fix a finite enumeration $A^0_1, \ldots, A^0_n$ of all ground atoms in the finite unsatisfiable set of ground instances of clauses in S secured by the compactness theorem. Sort them so that $A^0_i \succ A^0_j$ implies $i > j$. A closed tree T_S is *adequate* if and only if its vertices are of the form $\pm_1 A^0_1, \ldots, \pm A^0_k$ with $k \leq n$. By construction, there is an adequate closed tree T_S. Also, for each failure node I of T_S, there is a clause C_I in S and a substitution θ_I such that $C_I\theta_I$ is ground and false at I.

Given any set S' of clauses, call a *decorated tree* any tuple $(T, C_\bullet, \theta_\bullet)$, where T is an adequate closed tree, $C_\bullet$ maps each leaf I of T to a clause C_I of S', and $\theta_\bullet$ maps each leaf I to a substitution θ_I such that $C_I\theta_I$ is ground and false at I. The discussion above shows that S has a decorated tree.

Given a decorated tree $(T, C_\bullet, \theta_\bullet)$ for S', either the root ϵ is a leaf, then C_ϵ is necessarily the empty clause $\Box$, and we are done. Or we find a path through T as follows. Define the ground atom H_I and the sign $\pm_I$, for each leaf I, so that $\pm_I H_I$ is the literal $\pm A^0_i$ in $C_I\theta_I$ with the highest index i; i.e., the lowest (largest) literal on the path leading to I.

Definition 1 (Generative). *Let us say that C_I, and by extension $C_I\theta_I$, is* generative *if and only if $\pm_I$ is the $+$ sign, and no literal is selected: sel $(C_I) = \emptyset$.*

This is our version of Bachmair and Ganzinger's notion of *productive* clauses. Any clause C_I can be written uniquely as $\pm_I H_I \vee +\mathcal{P}_I \vee -\mathcal{N}_I$, where $\mathcal{P}_I$ is the set of atoms occurring under the $+$ sign in C_I (except H_I), and $\mathcal{N}_I$ is the set of atoms occurring under the $-$ sign in C_I. (We write $+\mathcal{P}$ for the disjunction of all $+B$, $B \in \mathcal{P}$, and $-\mathcal{N}$ for the disjunction of all $-B$, $B \in \mathcal{N}$.) Generative clauses are those where $\pm_I$ is the $+$ sign, and no literal is selected in $-\mathcal{N}_I$.

Now build a specific interpretation by Bachmair-Ganzinger forcing. Intuitively, each productive clause can be written as a Horn-like clause $H_I \Leftarrow -\mathcal{P}_I \wedge +\mathcal{N}_I$, stating that H_I should be set to true whenever all atoms in $\mathcal{P}_I$ are false and all atoms in $\mathcal{N}_I$ are true. We say that $-\mathcal{P}_I \wedge +\mathcal{N}_I$ is *true*, and that H_I is *forced* whenever this happens; otherwise, H_I will be set to the default value "false". We shall do so while traveling downwards inside T. E.g., look at Figure 1. The clause $+p$ is necessarily generative. The clause $+q \vee -p$ cannot be generative, because the only positive atom is not maximal, and similarly for $+r \vee -q$. Then, starting from vertex **1**, we let r be set to the default value false—no generative clause forces it to true. So we must go down left, and arrive at vertex **2**. Then we let q be false, go to **4**, and finally force p to true, arriving at **9**. Formally:

Definition 2. *Let $(T, C_\bullet, \theta_\bullet)$ be a decorated tree. Define a failure node I in T as follows. Let $I_0 = \epsilon$ be the root node of T. Then define I_k, $k \geq 1$, by induction on k as follows. Let I_k be given. If I_k is a failure node, then stop, and let $I = I_k$. Otherwise, if there is a generative clause $C_{I'} = +H_{I'} \vee +\mathcal{P}_{I'} \vee -\mathcal{N}_{I'}$ such that $-\mathcal{P}_{I'}\theta_{I'} \wedge +\mathcal{N}_{I'}\theta_{I'}$ is true in I_k and $H_{I'}\theta_{I'} = A^0_{k+1}$, then force A^0_{k+1} to true: define I_{k+1} as $I_k, +A^0_{k+1}$. Otherwise, let I_{k+1} be I_k.*

Clearly,

Lemma 1. *The partial interpretation I of Definition 2 satisfies the following two properties:*

(I.1) *For every generative clause $C_{I'}$ such that $-\mathcal{P}_{I'}\theta_{I'} \wedge +\mathcal{N}_{I'}\theta_{I'}$ is true in I, $H_{I'}\theta_{I'}$ is true in I.*

(I.2) *If H is a true atom in I, then there is a generative clause $C_{I'}$ such that $H_{I'}\theta_{I'} = H$. Moreover, $-\mathcal{P}_{I'}\theta_{I'} \wedge +\mathcal{N}_{I'}\theta_{I'}$ is true in I.*

These properties crucially depend on the fact that once an atom has been forced to true, resp. false, in I_k, it will remain so in all subsequent $I_{k'}$, $k' \geq k$. (Whence the name of forcing.)

The failure node I will be the place where resolution takes place, much as inference nodes were the places where resolution took place in Section 2.1. Let us see how I provides us with the main clause $C' \vee -A'_1 \vee \ldots \vee -A'_\ell$, so that condition *(iv)* is satisfied:

Lemma 2. *If there is at least one selected literal in C_I, C_I can be written as $C' \vee -A'_1 \vee \ldots \vee -A'_\ell$, where $-A'_1, \ldots, -A'_\ell$ are exactly the selected literals of C_I, and $\ell \geq 1$. Otherwise, let σ be any substitution that is more general than θ_I. Then, C_I is necessarily of the form $C' \vee -A'_1$, where $-A'_1\sigma$ is maximal in $C_I\sigma$, i.e., where $-A'_1\sigma \not\prec B$ for every atom B in $C'\sigma$.*

Proof. If $sel(C_I)$ is non-empty, this is clear. So assume $sel(C_I) = \emptyset$. Consider $\pm_I H_I$. If $\pm_I$ were $+$, C_I would be generative. But since $C_I\theta_I$ is false at I, $-\mathcal{P}_I\theta_I \wedge +\mathcal{N}_I\theta_I$ is true in I. By $(I.1)$ $H_I\theta_I$ would be true in I, too. This would make $C_I\theta_I$ true at I, contradiction. So $\pm_I$ is $-$. Let $-A'_1$ be H_I. Clearly, $A'_1\theta_I$ is below or equal to $A\theta_I$ for any A in C'. So $A'_1\sigma \not\prec A\sigma$, since $\succ$ is stable and $\sigma \sqsubseteq \theta_I$. □

We now show that the other conditions *(i)*, *(ii)*, *(iii)* on the rule of ordered resolution with selection also apply:

Lemma 3. *Let $A'_1, \ldots, A'_\ell$ be defined as in Lemma 2. For each i, $1 \leq i \leq \ell$, there is a generative clause $C_{I'_i}$ such that $H_{I'_i}\theta_{I'_i} = A'_i\theta_I$.*

Write $C_{I'_i}$ as $C_i \vee +A_{i1} \vee \ldots \vee +A_{in_i}$, where $+A_{i1}, \ldots, +A_{in_i}$, are all the literals L in $C_{I'_i}$ such that $L\theta_{I'_i} = +A'_i\theta_I$. Then:

(i) $n_i \geq 1$;
(ii) the mgu $\sigma = mgu\{A_{ij} = A'_i | 1 \leq i \leq \ell, 1 \leq j \leq n_i\}$ *exists, and* $\sigma \sqsubseteq \theta$, *where* $\theta = \theta_I \cup \theta_{I'_1} \cup \theta_{I'_2} \cup \ldots \cup \theta_{I'_\ell}$;
(iii) $sel(C_i \vee +A_{i1} \vee \ldots \vee +A_{in_i}) = \emptyset$ *and* $A_{i1}\sigma \not\precsim B$ *for every atom* B *in* $C_i\sigma$, *for every* i, $1 \leq i \leq \ell$;

Proof. Since $C_I\theta_I$ is false at I, all the atoms $A'_i\theta_I$ are true in I. By $(I.2)$, first part, there is a generative clause $C_{I'_i}$ such that $H_{I'_i}\theta_{I'_i} = A'_i\theta_I$. Necessarily, $C_{I'_i}\theta_{I'_i}$ contains the literal $+A'_i\theta_I$.

Let therefore $+A_{i1}, \ldots, +A_{in_i}$, $n_i \geq 1$, all the literals L in $C_{I'_i}$ such that $L\theta_{I'_i} = +A'_i\theta_I$, and let C_i be the disjunction of the remaining literals of $C_{I'_i}$. (Note that there may be *several* such literals L, whence

n_i may be different from 1, requiring positive factoring.) We have just found the side premise $C_{I'_i} = C_i \vee +A_{i1} \vee \ldots \vee +A_{in_i}$. Since $n_i \geq 1$, *(i)* obtains.

Then, $A_{ij}\theta_{I'_i} = A'_i\theta_I$. Since without loss of generality, A_{ij} and $A_{i'j'}$ have no free variable in common whenever $i \neq i'$, and since A_{ij} and $A'_{i'}$ have no free variable in common (for every i, i', j), the substitution $\theta_I \cup \theta_{I'_1} \cup \theta_{I'_2} \cup \ldots \cup \theta_{I'_\ell}$ makes sense, and unifies all A_{ij}s and A'_is: *(ii)* follows.

Since $C_{I'_i}$ is generative, no literal is selected in it. Assume that $A_{ij}\sigma \precsim B\sigma$ for some $B \in C_i$; by stability, using $\sigma \sqsubseteq \theta_{I'_i}$, $A_{ij}\theta_{I'_i} \precsim B\theta_{I'_i}$, that is, $H_{I'_i}\theta_{I'_i} \precsim B\theta_{I'_i}$. This is impossible, since $H_{I'_i}\theta_{I'_i}$ is the largest literal in $C_{I'_i}\theta_{I'_i}$, since by construction $B\theta_{I'_i} \neq H_{I'_i}\theta_{I'_i}$, and since $\precsim$ restricts to an ordering on ground atoms. So *(iii)* obtains. $\square$

Therefore $C_1\sigma \vee \ldots \vee C_\ell\sigma \vee C'\sigma$ is indeed inferrable by the rule of ordered resolution with selection.

We now turn to termination. Let $(T', C'_\bullet, \theta'_\bullet)$ be the new decorated tree obtained from $(T, C_\bullet, \theta_\bullet)$ as follows. Let S' be the set of clauses of which $(T, C_\bullet, \theta_\bullet)$ is a decorated tree, and let S'' be S' union the resolvent $C_1\sigma \vee \ldots \vee C_\ell\sigma \vee C'\sigma$. We must show that $(T', C'_\bullet, \theta'_\bullet)$ is less than $(T, C_\bullet, \theta_\bullet)$ in some well-founded ordering.

This ordering must be more sophisticated than the natural ordering on sizes $|T|$ of trees T that we used in Section 2.1: if we were to use this ordering, we should show that the resolvent is false at some vertex strictly above I. This won't be the case here, mainly because we do not implement negative factoring. However, at least the resolvent $(C_1\sigma \vee \ldots \vee C_\ell\sigma \vee C'\sigma)\theta = C_1\theta_{I'_1} \vee \ldots \vee C_\ell\theta_{I'_\ell} \vee C'\theta_I$ is false at I (if not higher in the tree). Indeed, $C'\theta_I$ is false at I, since $C'\theta_I$ is a sub-clause of $C_I\theta_I$, which is false at I. And each $C_i\theta_{I'_i}$, $1 \leq i \leq \ell$, is false at I, by the following argument. The generative clause $C_{I'_i}$ equals $+H_{I'_i} \vee +\mathcal{P}_{I'_i} \vee -\mathcal{N}_{I'_i}$. By construction of $C_{I'_i}$ and by $(I.2)$, second part, $-\mathcal{P}_{I'_i}\theta_{I'_i} \wedge +\mathcal{N}_{I'_i}\theta_{I'_i}$ is true at I. By construction, $C_i\theta_{I'_i}$ is exactly the sub-clause $+\mathcal{P}_{I'_i}\theta_{I'_i} \vee -\mathcal{N}_{I'_i}\theta_{I'_i}$, which is false at I. So $(C_1\sigma \vee \ldots \vee C_\ell\sigma \vee C'\sigma)\theta$ is indeed false at I. Since it only contains atoms not lower than the atoms in the premises, it is false at I.

It is therefore meaningful to consider:

Definition 3. *Let I' be the highest vertex in T, above I, where the resolvent $(C_1\sigma \vee \ldots \vee C_\ell\sigma \vee C'\sigma)\theta$ is false. Define the new decorated tree $(T', C'_\bullet, \theta'_\bullet)$ as follows:*

(a) If I' is strictly higher than I in T, then let T' be the closed tree whose leaves are I' plus all the leaves of T that are not below I'. ("Chop at I'.") Let $C'_{I'}$ be the resolvent $C_1\sigma \vee \ldots \vee C_\ell\sigma \vee C'\sigma$, and $\theta'_{I'}$ be θ. Let $C'_{I''}$ be $C_{I''}$ and $\theta'_{I''}$ be $\theta_{I''}$ for every $I'' \neq I'$.

(b) If $I' = I$, let T' be just T, C'_I be the resolvent $C_1\sigma\vee\ldots\vee C_\ell\sigma\vee C'\sigma$ (which thereforce replaces $C_I = C' \vee -A'_1 \vee \ldots \vee -A'_\ell$ at the leaf I), θ'_I be θ; let $C'_{I''}$ be $C_{I''}$ and $\theta'_{I''}$ be $\theta_{I''}$ for every $I'' \neq I$.

The latter case can only happen when the lowest atom of $(C_1\sigma\vee\ldots\vee C_\ell\sigma\vee C'\sigma)\theta$ is the same as that of $C_I\theta_I$, i.e., $H_I\theta_I$. Consider the other literals of $(C_1\sigma \vee \ldots \vee C_\ell\sigma \vee C'\sigma)\theta = C_1\theta_{I'_1} \vee \ldots \vee C_\ell\theta_{I'_\ell} \vee C'\theta_I$. The literals in $C_i\theta_{I'_i}$, $1 \leq i \leq \ell$, are, by definition of C_i, strictly higher than $H_{I'_i}\theta_{I'_i} = A'_i\theta_I$, which is an atom of $C_I\theta_I$, and is therefore higher than or equal to H_I. The literals of $C_i\theta_{I'_i}$ are then always strictly higher than H_I. The only reason why H_I can occur in $(C_1\sigma \vee \ldots \vee C_\ell\sigma \vee C'\sigma)\theta = C_1\theta_{I'_1} \vee \ldots \vee C_\ell\theta_{I'_\ell} \vee C'\theta_I$ is therefore that it occurs in $C'\theta_I$. What matters here is that by replacing $C_I\theta_I$ by $C_1\theta_{I'_1} \vee \ldots \vee C_\ell\theta_{I'_\ell} \vee C'\theta_I$ as the clause at leaf I, we have replaced large literals $H_I\theta_I$ by clauses $C_i\theta_{I'_i}$ which contain an arbitrary number of strictly smaller literals.

This suggests defining a measure based on multiset extensions. Formally:

Definition 4. *For every failure node I' in a decorated tree $(T, C_\bullet, \theta_\bullet)$, let $\mu_1(C_{I'}, \theta_{I'})$ be the multiset of all $A\theta_{I'}$, where $\pm A$ ranges over the literals of $C_{I'}$. This is ordered by the multiset extension $\succ_{mul}$ of $\succ$.*

In case *(b)*, where $I' = I$, we therefore obtain $\mu_1(C_I, \theta_I) \succ_{mul} \mu_1(C'_I, \theta'_I)$.

Definition 5. *Define $\mu^-(T, C_\bullet, \theta_\bullet)$ as the multiset of all measures $\mu_1(C_{I'}, \theta_{I'})$, when I' ranges over the failure nodes of T.*

In case *(b)*, $\mu_1(C_{I'}, \theta_{I'})$ decreases strictly, while $\mu_1(C_{I''}, \theta_{I''})$ remains unchanged for the other leaves I''. So $\mu^-(T, C_\bullet, \theta_\bullet)\ (\succ_{mul})_{mul}\ \mu^-(T', C'_\bullet, \theta'_\bullet)$ in case *(b)*. Let $|T|$ denote the size of T, and note that $|T| = |T'|$ in this case. In case *(a)*, clearly $|T| > |T'|$, so in any case $\mu(T, C_\bullet, \theta_\bullet)\ (>, (\succ_{mul})_{mul})_{lex}\ \mu(T', C'_\bullet, \theta'_\bullet)$, where:

Definition 6. *The measure $\mu(T, C_\bullet, \theta_\bullet)$ is defined as the pair $(|T|, \mu^-(T, C_\bullet, \theta_\bullet))$.*

Since $>$ is well-founded, and since $\succ$, which is an ordering on a *finite* set of atoms $A^0_1, \ldots, A^0_n$, is also well-founded, we conclude:

Lemma 4. *The reduction relation that replaces* $(T, C_\bullet, \theta_\bullet)$ *by* $(T', C'_\bullet, \theta'_\bullet)$, *as defined in Definition 3, terminates.*

We now terminate the proof of Theorem 1. Assume S unsatisfiable. Starting from a decorated tree for S, we build a derivation by ordered resolution with selection of $S = S_0, S_1, \ldots, S_k, \ldots$, each mapped to a decorated tree $(T_0, C_{0\bullet}, \theta_{0\bullet})$, $(T_1, C_{1\bullet}, \theta_{1\bullet})$, …, $(T_k, C_{k\bullet}, \theta_{k\bullet})$, …., where each decorated tree is obtained from the previous one by the reduction defined in Definition 3. By Lemma 4, this terminates, say at step k. Then the root of T_k must be a failure node, so S_k contains the empty clause $\Box$. $\Box$

This proof clearly takes its roots in both the semantic tree technique and Bachmair and Ganzinger forcing. Note that we only require $\gtrsim$ to be strongly stable. We don't need it to be a reduction ordering, or to be total on ground atoms, or even to be well-founded.

2.3 Redundancy Elimination and Games

An important component of every automated deduction system is a set of *redundancy elimination* rules. Classic redundant clauses include tautologies and subsumed clauses [BG01a]. Other useful redundancy elimination rules include simplification rules. A crucial import of Bachmair and Ganzinger's approach to resolution was to define standard redundancy criteria, a unified approach justifying which redundant clauses can be eliminated, and which simplification rules can be applied while preserving completeness.

We may see the subtle interaction between resolution and redundancy rules as a two-player game [dN95] between a *player* P and an *opponent* O. At each turn, either the empty clause $\Box$ has been derived, and P wins, or P chooses a resolvent to produce, then O applies any finite number of redundancy rules. Completeness is then equivalent to the existence of a winning strategy for P, starting from any unsatisfiable set S of clauses.

For simplicity, and without loss of generality, we shall assume that O can only add clauses, or remove clauses. Replacing and simplifying clauses will be implemented by adding the replacement clauses and removing the replaced clauses.

The proof of Theorem 1 shows what resolvent P should play at each turn; this resolvent is the one we constructed, which makes

$\mu(T, C_\bullet, \theta_\bullet)$ decrease strictly. Completeness in the presence of redundancy elimination rules obtains as soon as, whatever O does, it can only make the chosen measure $\mu(T, C_\bullet, \theta_\bullet)$ decrease or stay the same. This is obvious when O adds a clause: $(T, C_\bullet, \theta_\bullet)$ stays the same. This is trickier when O removes a clause. We need to make sure that: (†) whatever clause C is removed by O from the current clause set S', for any leaf I' of T such that $C = C_{I'}$ (note that there might be 0, 1, or several such leaves), there is another clause $C'_{I'}$ in S such that some ground instance $C'_{I'}\theta'_{I'}$ of $C'_{I'}$ is false at I', and $\mu_1(C_{I'}, \theta_{I'}) \succeq_{mul} \mu_1(C'_{I'}, \theta'_{I'})$, where $\succeq_{mul}$ is the reflexive closure of $\succ_{mul}$. If so, we shall change $(T, C_\bullet, \theta_\bullet)$ into $(T', C'_\bullet, \theta'_\bullet)$, where $T' = T$, $C'_{I'}$ and $\theta'_{I'}$ are as given above for all leaves I' such that $C = C_{I'}$ (note that $C'_{I'}\theta'_{I'}$ cannot be false strictly above I', since I' is a failure node, whence $T' = T$), and $C'_{I'} = C_{I'}$, $\theta'_{I'} = \theta_{I'}$ for all other leaves I'. It is clear that $\mu(T, C_\bullet, \theta_\bullet)$ will be larger than $\mu(T', C'_\bullet, \theta'_{I'})$ in the reflexive closure of $(>, (\succ_{mul})_{mul})_{lex}$, whence completeness is preserved.

Let us find a more readable criterion than condition (†) above. Recall that $C_1, \ldots, C_k \models C$ if and only if every Herbrand interpretation that makes all ground instances of $C_1, \ldots, C_k$ true also makes every ground instance of C true. Equivalently, every Herbrand interpretation that makes some ground instance of C false must make some ground instance of some C_i, $1 \leq i \leq k$, false. By analogy, let us say that $C_1 \ldots, C_k \models^* C$ if and only if every *partial* interpretation that makes some ground instance of C false must make some ground instance of some C_i false, too, $1 \leq i \leq k$.

Imitating Bachmair and Ganzinger's standard redundancy criterion, we may enforce the above condition (†) by requiring the stronger property that $C_1, \ldots, C_k \models^* C$, for some clauses $C_1, \ldots, C_k$ in the current clause set S such that $C \succ_{mul} C_1, \ldots, C \succ_{mul} C_k$. Here $\succ_{mul}$ makes sense provided we see clauses as multisets of literals, ignoring signs. (The standard redundancy criterion uses $\models$ instead of $\models^*$, and the ordering used is slightly different in Bachmair and Ganzinger's work. The latter is inconsequential.) Let us show that indeed (†) must hold. For each leaf I' where $C = C_{I'}$, since $C_1, \ldots, C_k \models^* C$, there is a clause C_i, $1 \leq i \leq k$, having a ground instance that is false at I'. Let $C'_{I'}$ be C_i, and $C'_{I'}\theta'_{I'}$ be the corresponding ground instance. We must show that $\mu_1(C_{I'}, \theta_{I'}) \succeq_{mul} \mu_1(C'_{I'}, \theta'_{I'})$. Since $C_{I'} = C \succ_{mul} C_i = C'_{I'}$, we may obtain $C'_{I'}$ from $C_{I'}$ by repetitively replacing atoms by finitely many smaller ones in the $\succ$ strict ordering. Since $\succ$ is stable, we may reproduce this at the ground level, and obtain $C'_{I'}\theta'_{I'}$ from $C_{I'}\theta_{I'}$ by repetitively replacing ground atoms by smaller ones in the $\succ$ strict ordering. So $\mu_1(C_{I'}, \theta_{I'}) \succeq_{mul} \mu_1(C'_{I'}, \theta'_{I'})$, and (†) obtains.

In case C is a tautology $C_0 \vee +A \vee -A$, k is zero, and the criterion is vacuously satisfied: we can always eliminate tautologies without breaking completeness in ordered resolution with selection. In case $C = C_{I'}$ is subsumed by some clause $C_1 = C'_{I''}$ ($k = 1$), it is not necessarily the case that $C \succ_{mul} C_1$, or even $\mu_1(C_{I'}, \theta_{I'}) \succeq_{mul} \mu_1(C'_{I''}, \theta'_{I''})$. E.g., take $C = +P(x)$, $C_1 = +P(x) \vee +P(y)$, which subsume each other, while $C \not\succeq_{mul} C_1$. This suggests that eliminating subsumed clauses is fraught with danger. And indeed, it is well-known that eliminating backward-subsumed clauses may break completeness. We shall let the reader check that we indeed obtain $\mu_1(C_{I'}, \theta_{I'}) \succeq_{mul} \mu_1(C'_{I''}, \theta'_{I''})$ as soon as $C'_{I''}$ subsumes C *linearly*, i.e., C is of the form $C'_{I''}\sigma \vee C''$, where σ does not unify any distinct literals in $C'_{I''}$ (i.e., $C'_{I''}\sigma$ is not a factor of $C'_{I''}$). This justifies that eliminating linearly subsumed clauses (whether backward or forward) does not break completeness. Eliminating linearly subsumed clauses is implemented in SPASS [WBH+02].

Our argument shows that completeness is in fact preserved if we remove $C = C'_{I''}\sigma \vee C''$, when both C and $C'_{I''}$ are in S, whatever σ is (i.e., even when C is subsumed non-linearly by $C'_{I''}$), provided C'' contains an atom A such that $A \succ B$ for every B in $C'_{I''}\sigma$: indeed in this case C can only be false at a vertex strictly below I'', hence C cannot be of the form $C_{I'}$ for any failure node I'.

Many other redundancy elimination rules are listed in [BG01a], on which the arguments above apply. We would like to end this section by examining the subtle case of the splitting-with-naming rule of [RV01a] (which was called *splittingless splitting* in [GLRV04], by analogy with inductionless induction). This will in particular show where using $\models^*$ instead of $\models$ matters. Assume we are given an initial set of clauses on a set $\mathcal{P}$ of predicates. Call these $\mathcal{P}$-clauses. For each equivalence class of $\mathcal{P}$-clauses C modulo renaming, let $\ulcorner C \urcorner$ be a fresh nullary predicate symbol not in $\mathcal{P}$. Call these fresh symbols the *splitting symbols*. The splittingless splitting rule allows one to replace a clause of the form $C \vee C'$, where C and C' are non-empty clauses that have no variable in common, where C' is a $\mathcal{P}$-clause, and where C contains at least one atom $P(t_1, \ldots, t_n)$ with $P \in \mathcal{P}$, by the two clauses $C \vee -q$ and $+q \vee C'$, where $q = \ulcorner C' \urcorner$. This rule is not only effective in practice [RV01a], it is also an important tool in proving certain subclasses of first-order logic decidable, and to obtain optimal complexity bounds (see e.g., [GL05]). Take $\succ$ so that $P(t_1, \ldots, t_n) \succ q$ for every $P \in \mathcal{P}$ and for any splitting symbol q. Then it is easy to see that the standard redundancy criterion is satisfied, and we can indeed *replace* $C \vee C'$ by the smaller clauses $C \vee -q$ and $+q \vee C'$. So completeness is preserved, as shown by Bachmair and Ganzinger, as soon as $\succ$ is a well-founded reduction ordering that is total on ground terms.

Our approach, as it is, does *not* apply here. We are paying the dues for all the benefits that our use of compactness brought us. Indeed, remember our proof started by taking a finite subset of ground atoms $A_1^0, \ldots, A_n^0$ that are required for finding a contradiction. While P is only required to play clauses with ground instances among the latter, O is not limited in any such way. Here, O may indeed produce $C \vee -q$ and $+q \vee C'$, where q is *not* among $A_1^0, \ldots, A_n^0$. Then we cannot remove $C \vee C'$. Assume that $C \vee C'$ is $C_{I'_i}$, for some leaves I'_i, $1 \leq i \leq k$. There is no reason why $C \vee -q$ or $+q \vee C'$ should be false at any I'_i: indeed q is *undecided*. In other words, while $(C \vee -q), (+q \vee C') \models C \vee C'$, we do *not* get $(C \vee -q), (+q \vee C') \models^* C \vee C'$. Bachmair and Ganzinger's standard redundancy criterion applies, but our variant does not.

This can be repaired easily if O can only generate finitely many splitting symbols. In this case, just assume they are all among $A_1^0, \ldots, A_n^0$, and completeness again obtains. E.g., in [GL05], the only splitting symbols we ever need are of the form $\ulcorner B(X) \urcorner$, where $B(X)$ is any disjunction of literals $-P(X)$, where P is taken from a finite set. So there are finitely many splitting symbols, and we can without loss of generality assume they are all among $A_1^0, \ldots, A_n^0$.

Despite these difficulties, completeness still holds in the general case. However, this is more complex: first, we need to assume a form of our old condition $(*)$, namely that the ordering $\succ$ on splitting symbols can be extended to a total ordering on the splitting symbols $q_1, q_2, \ldots, q_i, \ldots$ (a similar condition is used in [SV05, Theorem 4]); second, we need to consider transfinite semantic trees [HR91] based on the transfinite (indexed by the ordinal $\omega + n$) enumeration $q_1, q_2, \ldots, q_i, \ldots, A_1^0, \ldots, A_n^0$, where $A_1^0, \ldots, A_n^0$ are the ground atoms

$P(t_1, \ldots, t_n)$, $P \in \mathcal{P}$, given by the compactness theorem… but this is Bachmair and Ganzinger's usual forcing argument in disguise.

2.4 Where Trees Matter: Completeness of Linear Resolution

Until now, we have only used semantic trees as a convenient way of organizing paths, i.e., Herbrand interpretations. Similarly, Bachmair and Ganzinger's forcing argument builds an interpretation. One might therefore ask whether the use of *trees* brings any additional benefit than just reasoning on paths.

We claim that *linear resolution* can be shown complete using a semantic tree technique. This appears to be new by itself: the standard proof of completeness of linear resolution is by Anderson and Bledsoe's excess literal argument, applied to so-called minimally unsatisfiable sets of clauses. Furthermore, our semantic tree technique will really use trees, not just the paths inside the trees.

We won't make the full proof of completeness explicit. In particular, we will do as though all clauses were propositional. This is in the name of clarifying the argument. Also, this will let the readers enjoy filling in all details by themselves.

The rule of *linear resolution* can be explained as follows. Start from a clause set S_0, and pick a clause C_0 in S_0, non-deterministically. Find a resolvent of C_0 (the *center* clause) with some clause in S_0 (the *side* clause). Name this resolvent C_1; this is the *top* clause. The current clause set is now $S_1 = S_0 \cup \{C_1\}$. Then find a resolvent of the top clause C_1 (now the new center clause) with some side clause in S_1, call it C_2 (the new top clause). Proceed, getting a sequence of successive resolvents C_i, $i \geq 0$, until (hopefully) the empty clause $\Box$ is obtained. Observe that this is a non-deterministic procedure. The point in linear resolution is that the only allowed center clause at the next step is the previous top clause.

That linear resolution is complete means that, if S_0 is unsatisfiable, then there is a sequence of choices, first of C_0, then of each side clause, so that the empty clause $\Box$ eventually occurs as the top clause. Our technique will establish a more general result: linear *ordered* resolution, where each resolvent is constrained to be ordered (see Section 2.1), is complete again. This holds even if we only allow factoring in center clauses but disallow it in side clauses.

This refinement of linear resolution can be formalized as follows. The only deduction rule is:

$$\frac{\mp A_1' \vee C' \quad \pm A_1 \vee \ldots \pm A_m \vee C}{C\sigma \vee C'\sigma} \quad \begin{array}{l} m \geq 1, \\ \sigma = mgu(A_1 = A_2 = \ldots = A_m = A_1'), \\ A_i\sigma \not\succeq B, A_1'\sigma \not\succeq B \; \forall B \in C\sigma \vee C'\sigma, \\ \qquad 1 \leq i \leq m \end{array}$$

where $\pm$ is the same sign throughout, and $\mp$ is its opposite. The left premise is meant to be the side clause, and the right premise is the center clause.

The process of linear resolution is then defined through a transition relation. A *state* of the linear resolution procedure is a pair (S, C), where C is a clause in S. The *transition relation* of linear resolution) is given by

$$(S, C) \rightsquigarrow (S \cup \{C'\}, C')$$

where

$$\frac{C'' \quad C}{C'}$$

by the ordered linear resolution rule above, for some $C'' \in S$. Remember that C is the center clause, C'' is the side clause, and C' is the top clause.

Completeness means that, if S is unsatisfiable, then $(S, C) \rightsquigarrow^* (S', \Box)$ for some $C \in S$ and some clause set S'.

We prove this by modifying the notion of semantic tree slightly. E.g., consider the example of Figure 1, this time with the ordering $q \prec r \prec p$, see Figure 2.

Now look at vertex **2**. The choice on r here is irrelevant: there is no clause labeling any failure node below **2** that depends on the truth value of r. It is therefore tempting to reduce the semantic tree to the one shown in Figure 3, where vertex **2** has been replaced by the subtree rooted at vertex **5**. This reduction process is similar to that used in BDDs [Ake78].

Consider again a finite enumeration $A_1^0, \ldots, A_n^0$ of all ground atoms in the finite unsatisfiable set of ground instances of S. We now allow paths in semantic trees to skip over some atoms, as in Figure 3, where r is skipped in the paths on the left: r is a *don't care*. But atoms will still be enumerated in the same ordering on each path, i.e., we disallow flipping q and r on some paths but not all.

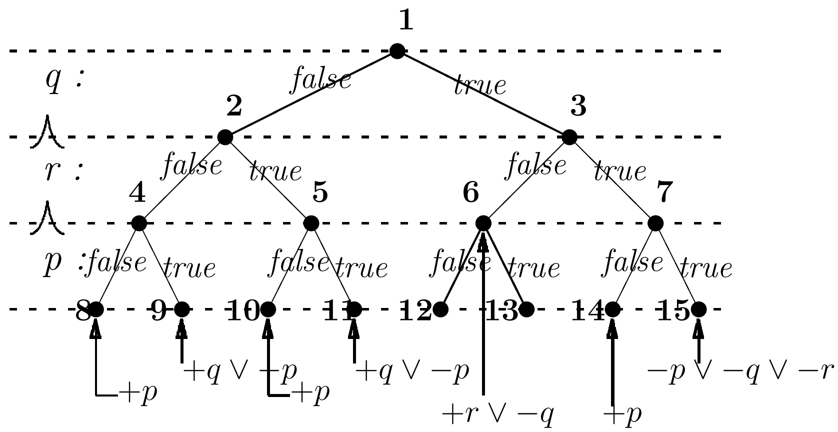


Fig. 2. Another semantic tree, based on a different ordering

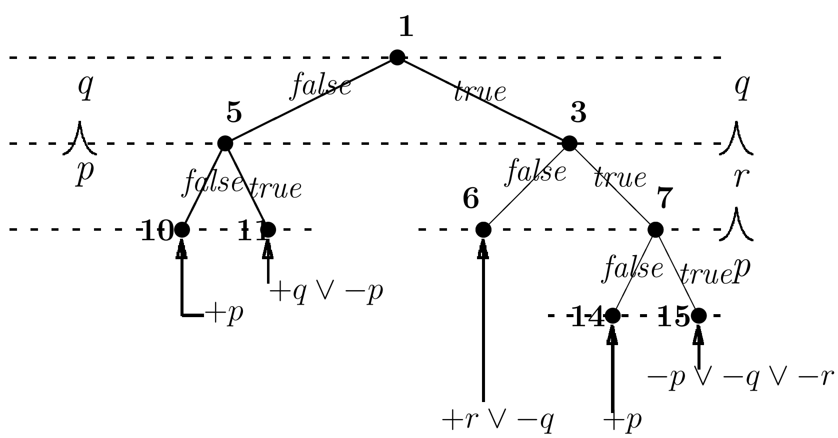
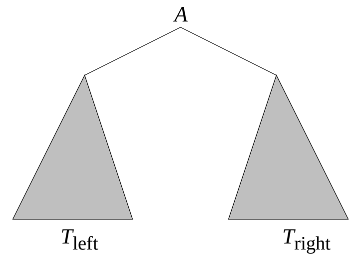


Fig. 3. A reduced semantic tree

Formally, define a *reduction* rule as follows. A *redex* in a decorated tree T for S is any subtree of T of the form shown on the right, where there is no ground clause $C' \vee -A$ labeling any failure node in T_{right} (or, symmetrically, where no ground clause $C \vee +A$ labels any failure node in T_{left}).



(1)

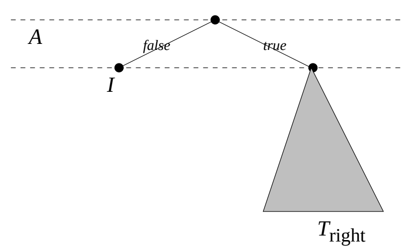
Given such a redex, the ground clauses that label the failure nodes in T_{right} do not contain $-A$ by assumption, and cannot contain $+A$, because they have to fail when A is taken to be true. It follows that replacing the whole subtree by T_{right} in T yields a decorated tree for S again. (The symmetric case where there is no ground clause $C \vee +A$ labeling any failure node in T_{left} leads to replacing the whole subtree by T_{left} instead.)

It is easy to see that reducing bottommost redexes first terminates. Moreover, the number of distinct clauses labeling failure nodes in the normal form is at most the same number in the input tree.

A *reduced* semantic tree is one that contains no redex. Any path in a reduced semantic tree is a partial Herbrand interpretation $\pm_1 A^0_{i_1}$, $\pm_2 A^0_{i_2}, \ldots, \pm_k A^0_{i_k}$, $1 \leq i_1 < i_2 < \ldots < i_k \leq n$.

Now take a reduced semantic tree T for the unsatisfiable set S not containing $\Box$. The set of clauses S_T decorating the failure nodes of T will play the role of the minimally unsatisfiable sets in the standard completeness argument. (Technically, S_T is minimally unsatisfiable among those subsets of S consisting only of atoms among $A^0_1, \ldots, A^0_n$, but need not be minimally unsatisfiable in the absolute sense.)

For every clause C in S_T, there is a failure node I in T such that $C = C_I$. Look at the tree T_{right} that is the brother of I in T. (Symmetrically, I may be on the right). We have named A the lowest (largest) atom in C_I.



(2)

In particular, C_I is of the form $C_1 \vee +A$. Since T is reduced, there is a failure node I' inside T_{right} such that $C_{I'}$ is of the form $C_2 \vee -A$.

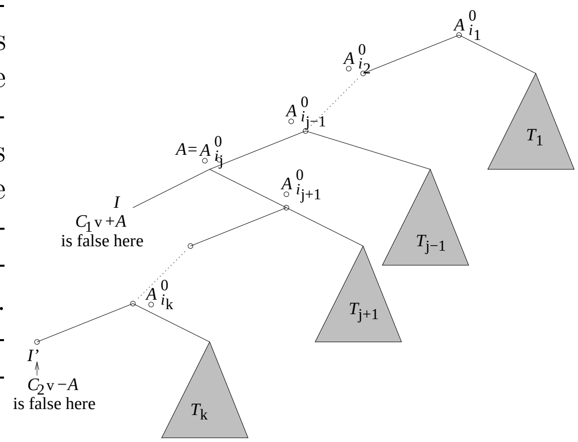
Now resolve $C_I = C_1 \vee +A$ with $C_{I'} = C_2 \vee -A$, getting the resolvent $C' = C_1 \vee C_2$. C' must be false at I', because C_1 is already false at the vertex right above I, and C_2 is false at I', which is below that same vertex. Note that I' may lie inside T_{right}, or above the top vertex of the subtree shown in Figure 2.

Look at the vertex I'' where C' *fails*, i.e., the highest vertex above I' where C' is false. Cut the tree T at I'', making I'' a new failure node, and decorate I'' with C' in the new semantic tree T'.

It may be the case that T' is not reduced, so reduce it. Let T'' be some normal form of T' for the notion of reduction defined above. (Beware that reduction need not be confluent.)

The important observation is that C'—our top clause—will still decorate some failure node in T'', provided we reduce T' to T'' in the right way. This can be observed by looking more finely at the shape of the subtrees involved.

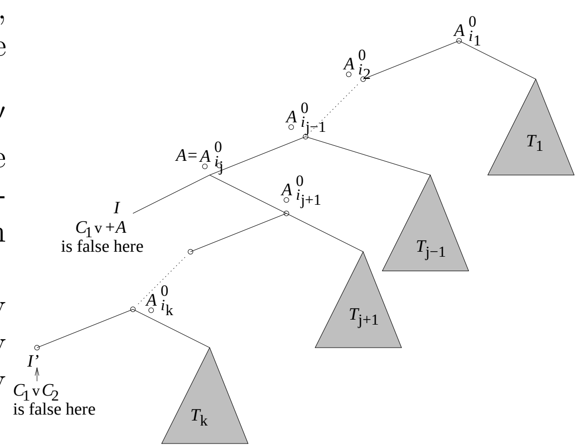
Before we resolve, our reduced tree T looks as shown on the right. We have assumed in this illustration that the branches leading to I and I' were leftmost, which would imply that C_1 and C_2 contain only positive atoms. The general situation reduces to this one by appropriate sign changes, i.e., by consistently exchanging $+A_i^0$ with $-A_i^0$ in all clauses and in the tree whenever needed.



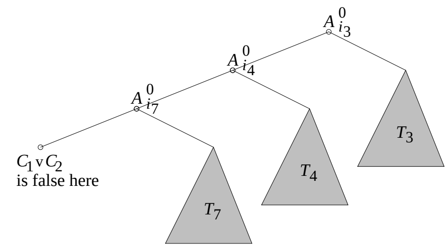
Note that the subtrees T_1, ..., T_{j-1}, T_{j+1}, ..., T_k are reduced.

After resolution, the tree T' will be as shown on the right, or with $C_1 \vee C_2$ labeling some node higher in the tree.

Since A does not occur any longer in $C_1 \vee C_2$, this may create redexes. But we may easily reduce T' as follows.



Pick the subsequence of the sequence of atoms $A^0_{i_1}, \ldots, A^0_{i_k}$ that actually occur in $C_1 \vee C_2$, say $A^0_{i_3}$, $A^0_{i_4}$, $A^0_{i_7}$, build the branch going from the root to the leaf decorated with $C_1 \vee C_2$ and containing just those atoms, and attach the corresponding side trees as follows:



This is our tree T''. It is clearly reduced, and the top clause $C_1 \vee C_2$ clearly labels one of its failure nodes. We can therefore reuse it as the center clause in the next resolution step.

It only remains to show that this process (mixing going from T to T' by one resolution step, then reducing T' to T'' in the appropriate way) terminates: the measure μ of Definition 6 is still defined on our new notion of semantic trees, and decreases not only when going from T to T', but also during reduction.

As we said earlier, we leave the formal proof to the reader. Since we use the same measure μ as in Section 2.2, we can dispense with factoring in the side clause, as announced. Indeed, going from T to T' amounts to replacing the side clause by a resolvent where the literal resolved upon is replaced by finitely many strictly smaller literals, coming from the center clause, as in Lemma 4.

A nice consequence of this new completeness proof is, as for any other proof obtained by semantic trees, that completeness is easily seen to be retained in the presence of redundancy elimination techniques.

E.g., we can remove tautologies, because tautologies cannot decorate any failure node. But this should be understood in a slightly different manner as for ordinary resolution, because linear resolution is a non-deterministic process. The completeness argument above shows that *there is* a way of doing linear resolution that leads to the empty clause without deriving any tautology as top clause. So, whenever we use linear resolution and derive a tautology as top clause, we can immediately stop deriving new clauses and backtrack.

Similarly, we can eliminate linearly subsumed clauses. Backward subsumption is not an issue here. Forward subsumption is as subtle as tautology elimination: if the top clause is subsumed, then we can stop and backtrack. Alternately, the completeness argument shows that we can *replace* C' by C'_1, and continue with C'_1 as the new top clause, thus restarting a proof.

We would like to stress that the tree structure is important here: the above proof crucially rests on reduction, which cannot be defined by just considering the paths of the tree T: brother subtrees have to be examined to recognize redexes.

3 Ordered Resolution, Paramodulation and Factoring

We now move to clauses involving the equality predicate.

3.1 Inference rules

Inference rules. First, we give inference rules applying to clauses defined as multisets of atoms: the same atom may appear several times in a clause. A ground instance of a clause is a true instance, there is no need to apply contractions. We use also an ordering on atoms extending an ordering $\succ$ on terms that will be defined later.

Resolution

$$\frac{+A \vee C \quad -A' \vee D}{C\sigma \vee D\sigma} \qquad \sigma = mgu(A = A'), A\sigma \not\prec B\ \forall B \in C\sigma \vee D\sigma$$

Monotonic Paramodulation

$$\frac{C \vee l = r \qquad D \vee \pm A[u]}{C\sigma \vee D\sigma \vee \pm A\sigma[r\sigma]} \qquad \begin{cases} \sigma = mgu(l = u), l\sigma = r\sigma \not\prec B\ \forall B \in C\sigma \\ r\sigma \not\prec l\sigma,\ A\sigma \not\prec B\ \forall B \in D\sigma \end{cases}$$

Factoring

$$\frac{+A \vee +A' \vee C}{+A\sigma \vee C\sigma} \qquad \sigma = mgu(A = A'), A\sigma \not\prec B\ \forall B \in C\sigma$$

Reflexivity

$$\frac{-u = v \vee C}{C\sigma} \qquad \sigma = mg(u = v),\ u\sigma = v\sigma \not\prec B\ \forall B \in C\sigma$$

Fig. 4. $\mathcal{ORMP}$: Ordered versions of Resolution, Monotonic Paramodulation, Factoring and Reflexivity

Reflexivity is also called *equality resolution* in the literature, because it appears to be a resolution between the clause $-u = v \vee C$ and the reflexivity axiom $x = x$.

This inference system is known to be complete when the ordering is a stable ordering, which is monotonic, total and well-founded on ground terms, in which case it must have the subterm property as well. Relaxing any one of these properties raises the question of what the new inference rule should be. Some authors [BG01b,BGNR99] keep the same inference rule for paramodulation, but we prefer another formulation which pinpoints the needed properties of the ordering in use. This is why we have renamed the paramodulation inference rule as *monotonic paramodulation*. We introduce now our version of paramodulation, *ordered paramodulation* and compare both rules by means of a few examples.

Ordered Paramodulation

$$\frac{C \vee l = r \qquad D \vee \pm A[u]}{C\sigma \vee D\sigma \vee \pm A\sigma[r\sigma]} \quad \begin{cases} \sigma = mgu(l = u), l\sigma = r\sigma \not\prec B \ \forall B \in C\sigma \\ A\sigma \not\prec A\sigma[r\sigma], A\sigma \not\prec B \ \forall B \in D\sigma \end{cases}$$

Fig. 5. Ordered Paramodulation Revisited

In ordered paramodulation, checking the rule instance has been replaced by checking the whole rewritten atom: ordered paramodulation coincides with monotonic paramodulation when the ordering is monotonic, total and well-founded. We call $\mathcal{ORP}$ the set of inference rules made of ordered resolution, ordered paramodulation, (ordered) factoring and (ordered) reflexivity.

Violating monotonicity. $\mathcal{ORP}$ is incomplete when the ordering on terms does not satisfy monotonicity. Consider the following unsatisfiable set of ground clauses

$$\{gb = b,\ fg^2b \neq fb\} \text{ with } fg^3b \succ fgb \succ fb \succ fg^2b \succ gb \succ b$$

Assuming that the ordering on terms is extended to atoms considered as multisets by taking its multiset extension, this set of ground unit clauses is closed under the inference rules in $\mathcal{ORP}$. Note that the ordering can be easily completed so as to satisfy the subterm property on the whole set of ground terms.

Using monotonic ordered paramodulation instead of ordered paramodulation yields the following set of clauses:

$$\{gb = b,\ fg^2b \neq fb\ , fgb \neq fb,\ fb \neq fb,\ \Box\}$$

and $\mathcal{ORMP}$ is indeed again complete [BG01b]. Note however that monotonic ordered paramodulation can be interpreted as ordered paramodulation with an ordering which is the monotonic extension of the ordering on ground instances of equality atoms. This ordering is therefore essentially monotonic.

Violating subterm. $\mathcal{ORP}$ turns out to be again incomplete when the ordering on terms does not satisfy the subterm property. Consider the following unsatisfiable set of ground clauses

$$\{a \neq fa,\ fb \neq fa,\ b = fb,\ a = fb\},\ \text{with } a \succ b \succ fa \succ fb$$

This set is closed under ordered paramodulation, resolution, factoring and reflexivity, assuming that the ordering on terms is extended to atoms considered as multisets by taking its multiset extension.

In [BGNR99], the authors show completeness of $\mathcal{ORMP}$ for Horn clauses when using a well-founded ordering which does not have the subterm property (with a proof which is quite intricate). To compute the set of clauses generated, we first need to extend the ordering into a well-founded ordering of the whole set of atoms:

$$f^n a \succ f^n b \succ \ldots \succ f^2 a \succ f^2 b \succ a \succ b \succ fa \succ fb$$

$\mathcal{ORMP}$ then yields the following infinite set of clauses:

$$\begin{array}{c}\{a \neq fa,\ fb \neq fa,\ a = fb\} \cup \\ \{f^n b = f^m b,\ a \neq f^m b,\ f^{n+1} b \neq f^{m+1} b \mid n \geq 0, m > 0\} \cup \\ \{\Box\}\end{array}$$

Indeed, any extension of the ordering would yield the same result, because the lefthand and righthand sides of equations are compared instead of the atoms themselves. Therefore, the equations $a = fb$ and $b = fb$ suffice for generating the whole set.

Subterm monotonicity does not suffice. We thought for a while that monotonicity could be restricted to the subterm relationship. Here is an example showing that this restriction of monotonicity does not ensure completeness:

$$\begin{array}{c}\{fa \neq b,\ a = b,\ gb = b, fga = b\} \\ \text{with} \\ f^2 b \succ f^2 a \succ fgb \succ fga \succ ga \succ gb \succ fb \succ fa \succ a \succ b\end{array}$$

Indeed, we need to paramodulate $fga = b$ by $a = b$ as if fga were bigger than fgb. In other words, the ordering $\succ$ must be monotonic on the rewrite relation induced by the equalitiy atoms $s = t$ generated from the clauses $s = t \vee C$ in which $s = t$ is maximal.

3.2 Ordering terms, atoms and clauses.

From now on, we assume that $\succeq$ is a stable, partial quasi-ordering on terms which restricts to a total strict ordering on ground terms which is monotonic and satisfies the subterm property. As a consequence, it is a simplification ordering, and is therefore well-founded on any set of terms which is generated from a finite signature. As another straightforward conseuqnece, ordered paramodulation and monotonic ordered paramodulation coincide.

We assume further that $\succ$ is extended to atoms so as to satisfy the following two properties:

(monotonicity) $s \succ t$ implies $A[s] \succ A[t]$ for any atom $A[s]$;

(*) $s \succ t$ implies $A[s] \succ s = t$ if A is not an equality atom.

Note that monotonicity extends monotonicity from terms to atoms. It also implies that $u[s] = u[t] \succ s = t$ if $u[] \neq []$ by the subterm property of $\succ$ applied twice and transitivity.

An example of ordering satisfying these properties can be obtained by extending the ordering $\succ$ from terms to atoms by letting

$$P(\overline{u}) \succ Q(\overline{u}) \text{ iff } (\max(\overline{t}), P, \overline{u})(\succ_{mul}, >_{\mathcal{P}}, \succ_{stat(P)})_{lex}(\max(\overline{v}), Q, \overline{v})$$

where the *precedence* $>_{\mathcal{P}}$ is a well-founded ordering on the set of predicate symbols in which the equality predicate is minimal and *stat* is a function from $\mathcal{P}$ to $\{lex, mul\}$ such that $stat(P) = mul$ iff P is the equality predicate.

3.3 Herbrand equality interpretations

Our goal is now to construct all Herbrand equality interpretations over a finite set $\mathcal{A}$ of ground atoms, which we suppose without loss of generality to be *closed under reflexivity*, that is, to contain all atoms $s = s$ such that $s = t \in \mathcal{A}$ for some t. The total well-founded ordering $\succ$ allows us to order the finite set of atoms, hence $\mathcal{A} = \{A_j\}_{j<n}$ such that $A_i \succ A_j$ if and only if $i > j$ (remember that we do not distinguish $s = t$ from $t = s$). The enumeration of the set of atoms based on the ordering $\succ$ provides us with a convenient characterization of Herbrand equality interpretations which are then organized as a finitely branching tree whose all nodes at a given depth assign a truth value to the same atom. Interpretations are in one-to-one corespondance with the branches of the tree.

Unlike the previous usual formulation of Herbrand interpretations, we assume here for convenience a set of three truth values $\{U, T, F\}$ where U stands for the *undefined* truth value and is used to consider partial interpretations as total functions over $\{U, T, F\}$.

Definition 7. *A (partial)* Herbrand interpretation *I of a finite set $\mathcal{A} = \{A_i\}_{i<n}$ of atoms is a mapping $[\_]_I$ from $\mathcal{A}$ to the set of truth values $\{U, T, F\}$. I is said to be* total *whenever its target is the subset $\{T, F\}$.*

Note that Herbrand interpretations are defined with respect to a given finite vocabulary of atoms closed under reflexivity. As usual, a *partial interpretation I of an initial segment* $\{A_i\}_{i<j\leq n}$ of $\mathcal{A}$ satisfies $[A_k]_I = U$ for all $j \leq k < n$. This is used in particular to represent all total interpretations assigning the same truth value among $\{T, F\}$ to the atoms in the initial segment, in the sense that if a formula ϕ takes value $x \in \{T, F\}$ in I, it takes the same value x in all total extensions of I. Here, undefined values may occur anywhere.

The logical connectives are classically extended to the third truth value by setting $T \vee U = T,\ F \vee U = U,\ T \wedge U = U,\ F \wedge U = F$ and $\neg U = U$. Interpretations are then extended to propositionnal formulae over $\mathcal{A}$ by taking their homomorphic extension. Let $U < T, U < F$ be the usual order on truth values, and $<$ be its natural pointwise extension to partial Herbrand interpretations. The intuition is that a partial Herbrand interpretation I of $\mathcal{A}$ stands for all total Herbrand interpretations H bigger than I in the order on interpretations.

We now turn our attention to Herbrand equality interpretations. Let E_I be the subset of equalities in $\mathcal{A}$ interpreted by T in some Herbrand interpretation I. Our goal is to define partial Herbrand equality interpretations in a way that specializes to the total case.

Definition 8. *A* Herbrand equality interpretation *is a Herbrand interpetation I that is compatible with the axioms of equality, that is:*

(i) for any term s, $[s = s]_I = T$;

(ii) for any two atoms A, B such that $A \longleftrightarrow^*_{E_I} B$, then $[A]_I = [B]_I$;

(iii) for any two terms s, t such that $s \longleftrightarrow^*_{E_I} t$ and any term u such that $u[s] = u[t] \in \mathcal{A}$, then $[u[s] = u[t]]_I = T$.

Note that the proof from A to B may involve atoms not in $\mathcal{A}$. A similar phenomenon may occur with the proof from s to t. Indeed, the first two conditions suffice to characterize Herbrand equality interpretations under our assumptions on $\succ$ and $\mathcal{A}$:

Lemma 5. *A Herbrand interpretation I of $\mathcal{A}$ is a Herbrand equality interpretation of $\mathcal{A}$ iff*

(i) for any atom $s = s \in \mathcal{A}$, $[s = s]_I = T$,

(ii) for any two different atoms $A, B \in \mathcal{A}$ such that $B \succ A$, $[A]_I, [B]_I \in \{T, F\}$ and $A \longleftrightarrow^*_{E_I} B$, then $[B]_I = [A]_I$.

Note that no constraint at all is imposed on A, B when $[A]_I = U$ or $[B]_I = U$. In case of a total interpretation, we obtain the usual characterization.

Proof. Clearly, if I is a partial Herbrand equality interpretation, (i) and (ii) must be satisfied. We need to show the converse.

Assume that $s \longleftrightarrow^*_{E_I} t$ and $u[s] = u[t] \in \mathcal{A}$ for some $u[]$. If s and t are identical, then $[u[s] = u[s]]_I = T$ by (i). Otherwise, let $s \succ t$. Then, $u[s] = u[t] \longleftrightarrow^*_{E_I} u[t] = u[t]$ which belongs to $\mathcal{A}$ by closure assumption and is smaller than $u[s] = u[t]$ by property of the ordering. By (ii) and (i), $[u[s] = u[t]]_I = [u[t] = u[t]]_I = T$. □

We now verify our intuition that partial Herbrand equality interpretations represent total ones:

Lemma 6. *Let ϕ be an arbitrary propositionnal formula over the vocabulary $\mathcal{A}$, I be a partial Herbrand equality interpretation, and $H > I$ a total Herbrand equality interpretation. Then $[\phi]_H = [\phi]_I$ iff $[\phi]_I \neq U$.*

We finally capture the idea that there are enough Herbrand equality interpretations on the one hand, and that a set of ground atoms becomes unsatisfiable in presence of the axioms of equality:

Definition 9. *A set S of Herbrand equality interpretations is* complete *if every Herbrand equality interpretation in $\{T, F\}^{\mathcal{A}}$ is smaller than some interpretation in S in the order of interpretations.*

Definition 10. *A set $\mathcal{C}$ of clauses is said to be* E-unsatisfiable *if $\mathcal{C}$ augmented with the axioms of equality is unsatisfiable.*

The following property of complete sets of Herbrand equality interpretations is the basis of our completeness proof:

Lemma 7. *A set $\mathcal{G}$ of ground clauses built from a set $\mathcal{A}$ of ground atoms closed under reflexivity is E-unsatisfiable iff $\mathcal{G}$ refutes a complete set of Herbrand equality interpretations over $\mathcal{A}$.*

Proof. Because the axioms of equality cannot refute Herbrand equality interpretations on the one hand, and a ground clause C refuting a partial interpretation I refutes all total interpretations bigger than I by Lemma 6 on the other hand. □

We now consider the problem of extending a complete set S of partial Herbrand equality interpretations over a finite set $\mathcal{A}$ of atoms into a complete set S' of partial Herbrand equality interpretations over $\mathcal{A} \cup \{B\}$. The new set of atom should of course contain the atoms $s = s$ and $t = t$ whenever B is the equality atom $s = t$. We will of course assume that $s = s$ and $t = t$ are added one by one before $s = t$. The flexibility of partial interpretetations allows us to extend each interpretation in S by exactly one interpretation in S':

Definition 11. *Given a partial Herbrand equality interpretation I over $\mathcal{A}$, we define its extension I' to $\mathcal{A} \cup \{B\}$ as follows:*

1. *If $B \in \mathcal{A}$, $I' = I$. Otherwise,*
2. *If B is an atom $s = s$, then $[B]_{I'} = T$.*
3. *If $B =_{E_I} A_i \in \mathcal{A}$ with $[A_i]_I \in \{T, F\}$, then $[B]_{I'} = [A_i]_I$.*
4. *If B is an atom $s = t$ such that there exists $A_i \neq A_j$ with $[A_i]_I = T$, $[A_j]_I = F$ and $A_i =_{E_I \cup \{s=t\}} A_j$, then $[s = t]_{I'} = F$.*
5. *Otherwise, $[B]_{I'} = U$.*

Note that Case 4 does not apply when B is strictly bigger than any atom in $\mathcal{A}$ since $\succ$ contains subterm.

Lemma 8. *Assume S is a complete set of partial Herbrand equality interpretations with respect to $\mathcal{A}$. Then the set S' obtained from S by replacing each partial Herbrand equality interpretation I by its extension I' to $\mathcal{A} \cup \{B\}$ is a complete set of partial Herbrand equality interpretations with respect to $\mathcal{A} \cup \{B\}$.*

Assume moreover that some interpretation $I \in S$ is refuted by a ground clause C. Then, its extensions I' in S' is refuted by the same clause C.

Proof. For the first statement, we need to show that every total Herbrand equality interpretation extending I extends I'. This follows from Definition 8 and Lemma 5. The second statement follows from Lemma 6. □

Example 1. Let $\mathcal{A}$ be the set $\{A(a), a = c, A(b), a = b, A(c)\}$ in increasing order, A being a predicate and a, b, c constants. We give from left to right: the 12 total Herbrand equality interpretations over the subset $\{A(a), a = c, A(b), A(c)\}$ of $\mathcal{A}$; a complete set of 4 partial Herbrand equality interpretations; its extension to $\mathcal{A}$.

A(a)	a=c	A(b)	A(c)
T	T	T	T
T	T	F	T
T	F	T	T
T	F	T	F
T	F	F	T
T	F	F	F
F	T	T	F
F	T	F	F
F	F	T	T
F	F	T	F
F	F	F	T
F	F	F	F

A(a)	a=c	A(b)	A(c)
T	U	U	U
F	T	U	F
F	F	T	U
U	F	F	U

A complete set of four partial Herbrand equality interpretations.

A(a)	a=c	A(b)	a=b	A(c)
T	U	U	U	U
F	T	U	U	F
F	F	T	F	U
U	F	F	U	U

Its extension with the atom $a = b$.

As usual, it is convenient to view a given set of Herbrand equality interpretations as a tree.

Definition 12. *Given a set S of partial Herbrand equality interpetations over the set of atoms $\mathcal{A} = \{A_i\}_{i<n}$ ordered by $\succ$, we construct the* tree of Herbrand equality interpretations *T_S by induction on $\succ$. Each node I in the tree defines a partial Herbrand equality interpretation I of an initial segment $\{A_i\}_{i<j<n}$ of atoms enumerated so far and a set E_I of equalities interpreted by true in I. The node I has*

1. *a single successor J such that $[A_j]_J = x$ in case all interpretations in S whose restriction coincide on $\{A_i\}_{i<j}$ assign the same value x to A_j;*
2. *two or three successors otherwise, depending on the different values assigned to A_i by the interpretations in S whose restriction coincide on $\{A_i\}_{i<j<n}$.*

Case 1 applies in particular when A_i is an atom of the form $s = s$ for some term s, in which case $[A_i]_J = T$, or when $A_j =_{E_I} A_k$ for some $k < j$, in which case $[A_j]_J = [A_k]_I$.

It is clear that the set of branches of T_S is in one-to-one correspondance with the set S. This property will be exploited without saying in the rest of the paper.

Definition 13. *The tree T_S of Herbrand equality interpretations over $\mathcal{A}$ is* narrow *if every internal node I has either one successor assigning a truth value among $\{U, T, F\}$ to the atom $A_{|I|+1}$, or else two assigning the truth values among T and F respectively to the atom $A_{|I|+1}$. The set S of interpretations will be called* narrow *as well.*

Lemma 9. *Every complete set S of Herbrand equality interpretations over $\mathcal{A}$ contains a narrow complete set S'.*

Proof. Let I a interal node of T_S with a son J such that $[A_{|I|+1}]_J = U$. Then, all other sons of I if any may be deleted without compromising completeness. □

Using narrow sets of interpretations makes the undefined truth value useless: if I has J for single successor assigning the truth value U to the atom $A_{|I|+1}$, then we can collapse the nodes I and J and omit this atom. We prefer however to keep undefined values because they allow us having a given atom interpreted at a given depth in the tree of Herbrand equality interpretations, all branches having therefore the same length. In other words, all branches of the tree give a truth value in $\{U, T, F\}$ to all atoms in $\mathcal{A}$, rather than a truth value in $\{T, F\}$ to a subset of atoms in $\mathcal{A}$ as it is the case in Section 2.4.

3.4 Semantic trees and generating interpretations

In this section, we assume given

- a finite set of atoms $\mathcal{A} = \{A_I\}_{i<n}$ closed under reflexivity such that $A_i \succ A_j$ iff $i > j$;
- an E-unsatisfiable set $\mathcal{G}$ of ground clauses built from the atoms in $\mathcal{A}$ which is closed under positive factoring;
- a complete narrow set S of partial Herbrand equality interpretations over $\mathcal{A}$, or equivalently, its associated narrow tree T_S.

We will say that the triple $(\mathcal{A}, \mathcal{G}, S)$ (or equivalently $(\mathcal{A}, \mathcal{G}, T_S)$ or even $(\mathcal{A}, \mathcal{G}, T_\mathcal{G})$ satisfies assumption (*). Note that both assumed closure properties do not need extending the set of terms, as would closure of $\mathcal{G}$ under ordered paramodulation.

Definition 14. *Given $(\mathcal{A}, \mathcal{G}, T_S)$ satisfying (*), we call* failure node *any node J of T_S for which there exists $C \in \mathcal{G}$ such that $[C]_J = F$ and $[C]_I = U$ for any ancestor I of J. We call* semantic tree *associated with $(\mathcal{A}, \mathcal{G}, T_S)$ the tree obtained from T_S by replacing a failure node J on each branch of the tree by a leaf labelled with the associated clause C. We denote it by $T_\mathcal{G}$.*

Note that $T_\mathcal{G}$ is not defined uniquely. This is on purpose, since it will be convenient to consider non-minimal semantic trees in our completeness proof. However, our definition forces the atom enumerated at a failure node to be either T or F.

Since C is a ground clause, $[C]_J$ is defined iff all its atoms are assigned a truth value in $\{T, F\}$ by J. Hence, the failure node cannot assign the undefined truth value to the last atom enumerated at a failure node. Another consequence, since $\mathcal{G}$ is E-unsatisfiable, is that the semantic tree is *closed*, that is, all its branches end up in a failure node. As usual, the only clause refuting the root of the tree is the empty clause.

We now define a specific interpretation G (actually, a class of interpretations) ending up in a failure node at which an ordered resolution or paramodulation will always be possible. The idea is that an equality atom $l = r$ should belong to E_G, that is, be interpreted in T by G, iff it corresponds to a clause $l = r \vee C$ which can be used to perform a ground ordered paramodulation. The generating interpretation is of course directly related to the notion of *generated equality* of Bachmair and Ganzinger. It pops up very naturally in the context of semantic trees.

Definition 15. *The set of* generating interpretation *G of a narrow closed semantic tree associated with the triple $(\mathcal{A}, \mathcal{G}, T_S)$ satisfying (*) is defined inductively as follows. Assume some node I in the semantic tree is the generating interpretation constructed so far. If I is a leaf, we are done. Otherwise, let A be $A_{|I|+1}$.*

1. *If I has a unique successor I' in the semantic tree, we choose I'.*
2. *If I has a successor L which is not a failure node and A is an equality atom such that $[A]_L = F$, we choose L.*

3. *If I has two successors K and L such that $[A]_L = F$ and L is a failure node, we choose K. If A is the equality atom $s = t$, the clause $s = t \vee D$ labelling L is called a* generating clause *and $s = t$ is a* generated equation.
4. *Otherwise we choose any successor of I.*

We denote by $\mathcal{G}en_G$ the set of generating clauses and by G an arbitrary generating interpretation.

Notice that we need not make any particular choice when the enumerated atom is not an equality atom and the right son is not a failure node, therefore leaving room for improvement. For example, we could superimpose a selection function as in Section 2. Note also that we could define generating interpretations for non-narrow trees. The above definition then shows that we would always need taking the successor J such that $[A]_J = U$ whenever there is such one.

In Bachmair and Ganzinger's work, the generating interpretation is unique, as well as the set of generating clauses. This is so because they encode predicates as Boolean functions. Here, the generating interpretation is not unique, but the set of generating clauses does not depend upon the choice of a particular generating interpretation: it is easy to see that a clause $s = t \vee C$ generates the equation $s = t$ with $s \succ t$ if $s = t$ is maximal in the clause and is irreducible by the previously generated equations. The definition by Bachmair and Ganzinger is slightly different, since they allow the right hand side t of the equation $s = t$ to be reducible. We could do that as well, since this becomes important for showing completeness of the superposition paramodulation strategy. This does not need changing the definition of generating interpretations, it suffices to collect more equations along them.

Lemma 10. *Assume that G is a generating interpretation of a narrow closed semantic tree. Then E_G is a canonical set of rewrite rules.*

Proof. Assume an equation $s = t$ is generated at the son L of some node I. By definition, I must have at least two successors, hence the newly atom $s = t$ is irreducible with respect to E_I.

Let now $u = v \in E_G \setminus (E_I \cup \{s = t\}$ and assume without loss of generality that $u \succ v$. By definition of the tree of Herbrand equality interpretations, $u = v \succ s = t$. By properties of $\succ$, $u \succ s$ and $u \succ t$, hence u is not a subterm of s or of t. It follows that $s = t$ cannot be reduced by $u \rightarrow v$.

Therefore, $s = t$ is irreducible with respect to $E_G \setminus \{s = t\}$. Since E_G is clearly terminating, the result follows. □

Lemma 11. *Assume that* G *is a generating interpretation of a narrow closed semantic tree associated with the triple* $(\mathcal{A}, \mathcal{G}, T_S)$ *satisfying (*). Assume further that* $A_i \longleftrightarrow^*_{E_G} A_j$ *for some* $j < i$*. Then, there exists a* generating clause $s = t \vee C$ *in* $\mathcal{G}$ *such that:*
(i) $A_i \longrightarrow_{s=t \in E_G} B$*, with* $s \succ t$ *and* $A_i \succ B$*,*
(ii) $s = t \succ A$ *for every atom* A *of* C*,*
(iii) $[C]_G = F$*.*

In case the ordering $\succ$ is not monotonic, the lemma does not hold anymore, and reducible atoms may not be reducible by (irreducible) generated equations. Our example violating subterm monotonicity shows this behaviour for the atom $fga = b$ which is reducible by $ga = a$ and $ga = b$, but not by $a = b$ although $a = b$ reduces ga. It is easy to see that monotonicity is only needed for equations reducing other equations, that is, for the equations in $\mathcal{E}$.

Proof. (i) follows from Lemma 10 and the assumption $A_i \succ A_j$. Note that B may not belong to $\mathcal{A}$. We are left with (ii) and (iii). Let I be the node of which G is a son.

Since $s = t$ is the last atom enumerated by G, it is maximal in the clause. Since $\mathcal{G}$ is closed under positive factoring, we can assume without loss of generality that $s = t \notin C$, hence $[C]_G = [C]_I = F$ and $s = t$ is strictly bigger than any atom in C. □

3.5 Refutational completeness of $\mathcal{ORP}$.

Let $\mathcal{C}$ be a set of clauses which is E-unsatisfiable. Our purpose is to show that $\mathcal{ORP}$ is refutationnaly complete, that is, the empty clause is generated in finite time from $\mathcal{C}$. To do this, we will as usual reason at the ground level, and use a lifting argument to relate both the ground and non-ground levels. Our lifting argument is indeed made simple because a ground instance $C\gamma$ of a clause is a multiset of ground atoms, therefore eliminating any need for contraction.

Theorem 2. *A set of clauses* $\mathcal{C}$ *is E-unsatisfiable iff the empty clause belongs to the closure of* $\mathcal{G}$ *under* $\mathcal{ORP}$*.*

Proof. By compactness and Lemma 7, we chose first a finite E-unsatisfiable set of ground instances of $\mathcal{C}$. Let $\mathcal{A}$ be the set of ground atoms occuring in $\mathcal{G}$. We add to $\mathcal{A}$ all atoms of the form $s = s$ whenever $s = t \in \mathcal{A}$, and close $\mathcal{G}$ under positive factoring. We then compute the set S of Herbrand equality interpretations over $\mathcal{A}$ and organize it as a narrow tree T_S. Therefore, the triple $(\mathcal{A}, \mathcal{G}, T_S)$ satisfies (*). We finally compute the narrow closed semantic tree T_G. This ends up the initialization phase.

We define the complexity of a semantic tree $T_{\mathcal{G}}$ to be the multiset of clauses $\in T_{\mathcal{G}}$ refuting its leaves. Complexities are compared in the multiset extension of $\succ$. Since the last atom enumerated at a failure node cannot be undefined, the smallest semantic tree in this order is therefore the empty tree refuted by the empty clause.

During the course of the proof, we will perform an operation on the current triple $(\mathcal{A}, \mathcal{G}, T_S)$ called *extension*, each time a new clause is added to $\mathcal{G}$, let us call $\mathcal{G}'$ the new set. First, we recompute the set of atoms, let us call it $\mathcal{A}'$, and complete it as before with the necessary atoms $s = s$. As before, we also close $\mathcal{G}$ under positive factoring. We then extend the complete set of interpretations S over $\mathcal{A}$ into a new complete S'' by adding the atoms in $\mathcal{A}' \setminus \mathcal{A}$ one by one, in increasing order, thanks to definition 11. By Lemma 8, S'' is complete. By Lemma 9, we now compute $S' \subseteq S''$ such that S' is narrow. Therefore, the new triple $(\mathcal{A}', \mathcal{G}', T_{S'})$ satisfies (*). By Lemma 8, the interpretations in S' are refuted by a subset of the clauses in $\mathcal{G} \subseteq \mathcal{G}'$ that refute the interpretations in S. It follows that extensions do not increase the complexity of the semantic tree.

We now reason by induction on the semantic tree $T_{\mathcal{G}}$. If $T_{\mathcal{G}}$ is empty, we are done. Otherwise, we choose an arbitrary generating interpretation ending up in a leaf J of $T_{\mathcal{G}}$. By non-emptyness assumption, J has a father node I. By definition of the semantic tree, J is refuted by a clause in $\mathcal{G}$ of the form $\pm P(\overline{u}\gamma) \vee C\gamma$, in which $A = P(\overline{u}\gamma)$ is the last atom enumerated by J, hence is bigger than or equal to any atom in C, and A is assigned either the value T or the value F in J. Let us assume that there exists some clause in $\mathcal{ORP}(\mathcal{G})$ that refutes some extension J' of J to be defined next, and is strictly smaller than $\pm P(\overline{u}\gamma) \vee C\gamma$. This clause may involve new atoms (because of paramodulation inferences). We therefore apply finitely many completion steps resulting in a set of clauses $\mathcal{G}'$ containing $\mathcal{G}$ and the new clause and a semantic tree $T_{\mathcal{G}'}$. By our assumption, we can replace the clause $\pm P(\overline{u}\gamma) \vee C\gamma$ refuting the node J' extending J by the infered clause which is strictly smaller, therefore decreasing the complexity of the semantic tree. We conclude by induction hypothesis.

We are left showing that our assumption can be fullfilled. By definition of the generated interpretation, there are four cases:

1. $P(\overline{u}\gamma)$ is of the form $s = s$, in which case I has J as single successor labelled by $\neg s = s \vee C\gamma \succ C\gamma$. By reflexivity, $C\gamma$ belongs to $\mathcal{ORP}(\mathcal{G})$ and refutes the interpretation J.
2. $P(\overline{u}\gamma)$ is irreducible by a rule in E_I. Then, I has two successors, J and K which are both failure nodes by definition of G, labelled by clauses in both of which the atom $P(\overline{u}\gamma)$ is maximal. Let these clauses be $+P(\overline{u}\gamma) \vee C\gamma$ and $-P(\overline{u}\gamma) \vee D\gamma$, in which $P(\overline{u}\gamma)$ is strictly bigger than any atom occuring in $C\gamma$. Therefore, $C\gamma \vee D\gamma$ refutes the interpretation J.
3. $A = P(\overline{u}\gamma)$ is reducible by E_I at a non-variable position p of $P(\overline{u})$ by an equation $s = t \in E_I$ such that $s \succ t$, yielding the atom $B[t]_p$. By Lemma 11, $s = t$ is generated by a clause $s = t \vee D\gamma$ such that $s = t$ is strictly bigger than any atom in $D\gamma$. Therefore, there is an ordered paramodulation between $s = t \vee D\gamma$ and the clause $\pm A \vee C\gamma$, yielding $B \vee C\gamma \vee D\gamma$, which therefore belongs to $\mathcal{ORP}(\mathcal{G})$. Consider now the tree of Herbrand equality interpretations extended from the previous one to the set of atoms $\mathcal{A} \cup \{B[t]_p$. Let I', J' be the respective extensions of I, J. Since $[s = t]_J = T$, then $[B[t]_p]_{J'} = [B[s]_p]_{J'} = F$, and since $B[s]_p \succ B[t]_p$, then $[B[t]_p]_{I'} = F$. By Lemma 8, $[C\gamma]_{J'} = [D\gamma]_{J'} = F$, hence $[C\gamma \vee D\gamma]_{J'} = F$, and by the same token as previously $[C\gamma \vee D\gamma]_J = F$, Therefore $[B[t]_p \vee C\gamma \vee D\gamma]_{I'} = F$.
4. $P(\overline{u}\gamma)$ is reducible by E_I at a position in γ, hence $\gamma \longrightarrow_{E_I} \gamma'$. We now consider the clause instance $+P(\overline{u}\gamma') \vee C\gamma'$, which is strictly smaller than the previous one. This case is similar to the previous one, except that there may be several new atoms in $+P(\overline{u}\gamma') \vee C\gamma'$. □

4 Conclusion

Recasting Ganzinger's work into the framework of finite semantic trees was an enriching experience which we plan to continue. Our next step will be to consider basic ordered resolution and paramodulation together with selection strategies via term selection as done in [BGLS92], as well as redundancy criteria. Then, we plan to blend various strategies, in particular, linear and ordering restrictions.

To conclude, we need comparing the model generation method with semantic trees. The implicit answer given here to that natural question is that there is no significant difference between both. The former does not construct all interpretations, but only a *relevant* one, while the latter describes the relevant one as a maximal branch in the tree of all interpretations. One main difference is the use of the compactness argument to make the semantic tree finite. The same could probably be done with model generation. A second diference is that semantic trees fit better with our own intuition.

Acknowledgments: The authors thank for their remarks Sergiu Bursuc and Yevgeny Kazarov from MPI, as well as Michaël Lienhardt, Bruno Marnette, Muriel Roger and Kumar Neeraj Verma from ENS-Cachan.

References

[Ake78] S. B. Akers. Functional testing with binary decision diagrams. In *Eighth Annual Conference on Fault-Tolerant Computing*, pages 75–82, 1978.

[BD94] Leo Bachmair and Nachum Dershowitz. Equational inference, canonical proofs and proof orderings. *Journal of the ACM*, 41(2):236–276, March 1994.

[BG01a] Leo Bachmair and Harald Ganzinger. *Resolution Theorem Proving*, volume I, chapter 2, pages 19–99. North-Holland, 2001. In [RV01b].

[BG01b] Miquel Bofill and Guillem Godoy. On the completeness of arbitrary selection strategies for paramodulation. *Lecture Notes in Computer Science*, 2076:951–??, 2001.

[BGLS92] Leo Bachmair, Harald Ganzinger, Christopher Lynch, and Wayne Snyder. Basic paramodulation and superposition. In Deepak Kapur, editor, *Proc. 11th Int. Conf. on Automated Deduction, Saratoga Springs, NY, LNAI 607*. Springer-Verlag, June 1992.

[BGNR99] M. Bofill, G. Godoy, R. Nieuwenhuis, and A. Rubio. Paramodulation with non-monotonic orderings, 1999.

[CL73] Chin-Liang Chang and Richard Char-Tung Lee. *Symbolic Logic and Mechanical Theorem Proving*. Computer Science Classics. Academic Press, 1973.

[dN95] Hans de Nivelle. *Ordering Refinements of Resolution*. PhD thesis, Technische Universiteit Delft, 1995.

[GL05] Jean Goubault-Larrecq. Deciding $\mathcal{H}_1$ by resolution. *Information Processing Letters*, 2005. To appear.

[GLRV04] Jean Goubault-Larrecq, Muriel Roger, and Kumar Neeraj Verma. Abstraction and resolution modulo AC: How to verify Diffie-Hellman-like protocols automatically. *J. Logic and Algebraic Programming*, 2004. To appear.

[HR86] Jieh Hsiang and Michaël Rusinowitch. A new method for establishing refutational completeness in theorem proving. In *Proc. 8th Int. Conf. on Automated Deduction, Oxford, England, LNCS 230*, pages 141–152. Springer-Verlag, 1986.

[HR91] Jieh Hsiang and Michal Rusinowitch. Proving refutational completeness of theorem-proving strategies: The transfinite semantic tree method. *Journal of the ACM*, 38(3):559–587, July 1991.

[Hue80] Gérard Huet. Confluent reductions: abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27(4):797–821, October 1980.

[JK86] Jean-Pierre Jouannaud and Hélène Kirchner. Completion of a set of rules modulo a set of equations. *SIAM Journal on Computing*, 15(4):1155–1194, 1986.

[Joy76] William H. Joyner Jr. Resolution strategies as decision procedures. *Journal of the ACM*, 23(3):398–417, July 1976.

[Kow69] Robert Kowalski. Semantic trees in automatic theorem-proving. *Machine Intelligence*, 4:86–101, 1969.

[RV01a] Alexander Riazanov and Andrei Voronkov. Splitting without backtracking. In B. Nebel, editor, *Proc. 17th Intl. Joint Conf. Artificial Intelligence*, volume 1, pages 611–617. Morgan Kaufmann, August 2001.

[RV01b] J. Alan Robinson and Andrei Voronkov, editors. *Handbook of Automated Reasoning*. North-Holland, 2001.

[SV05] Helmut Seidl and Kumar Neeraj Verma. Flat and one-variable clauses: Complexity of verifying cryptographic protocols with single blind copying. In *Proc. 11th Intl. Conf. Logic for Programming, Artificial Intelligence and Reasoning (LPAR'04)*, pages 79–94. Springer-Verlag LNCS 3452, 2005.

[WBH+02] Christoph Weidenbach, Uwe Brahm, Thomas Hillenbrand, Enno Keen, Christian Theobald, and Dalibor Topic. SPASS version 2.0. In *Proc. 18th Conference on Automated Deduction (CADE-18)*, pages 275–279. Springer-Verlag LNCS 2392, 2002.