ENS Paris-Saclay

Informatique, Logique L3

Resolution Strategies

Revision 4*

David Baelde

May 4th, 2020

Abstract

In this lecture we explore resolution *strategies*, i.e. restrictions of the resolution proof system. From a theoretical viewpoint, the main question one asks about a strategy is whether it is still refutationally complete, at least for a logical fragment of interest. Imposing strategies on resolution is crucial for many practical uses of resolution. We discuss below several interesting strategies, with different proofs of completeness and different applications.

We recall in section 1 the definition of resolution proof systems from the previous course. Section 2 also recalls Herbrand's theorem and gives an alternative formulation of it. In section 3, we go over the proof of completeness of resolution using semantic trees, and we adapt it to show the completeness of ordered resolution in section 4. Section 5 introduces Horn clauses and several resolution strategies that are refutationally complete for Horn clauses, and discusses their application.

*Modifications made since the original version of the document are indicated in margins with their corresponding revision.

1 Definitions and notations

In this document we assume some signature $\mathcal{F}$ and set of predicate symbols $\mathcal{P}$. We recall next the rules of resolution.

Definition 1.1 (Literal)**.** A literal (noted L) is either an atom (noted A) or the negation of an atom. We say that the literal is positive when it is an atom, negative otherwise. Given a literal L, we define the opposite literal $\overline{L}$ by $\overline{A} = \neg A$ and $\overline{\neg A} = A$.

Definition 1.2 (Clause)**.** A clause (noted C) is a universally quantified disjunction of literals, i.e. a formula of the form $\forall \vec{x}. \bigvee_{1 \leq i \leq n} L_i$ where the L_i are literals. We further require that clauses are closed, i.e. all their variables are bound by the universal quantifiers. We say that a clause is *ground* when it features no variable at all, i.e. it is a disjunction of closed literals without any universal quantification.

We identify clauses modulo the associativity and commutativity of disjunction. The empty disjunction is $\bot$. Clauses are often written with their universal quantifiers implicit, as in the following definition.

Definition 1.3 (Resolution, R)**.** The resolution proof system R is given by the following two rules, respectively called resolution and factorisation:

$$\frac{C \vee L \quad \overline{L'} \vee C'}{(C \vee C')\sigma}\ \sigma = \mathsf{mgu}(L, L') \qquad \frac{C \vee L \vee L'}{(C \vee L)\sigma}\ \sigma = \mathsf{mgu}(L, L')$$

We write $E \vdash_R C$ when the clause C can be derived from the clauses of E using the rules of R, in an arbitrary number of steps.

Definition 1.4 (Ground resolution, R_0)**.** We define the variant R_0 of R by the following three rules:

$$\frac{C \vee L \quad \overline{L} \vee C'}{C \vee C'} \qquad \frac{C \vee L \vee L}{C \vee L} \qquad \frac{C}{C\sigma}$$

System R_0 is convenient to work with in theory (e.g. in completeness arguments) but not appropriate for automated reasoning. Intuitively, system R is a "lazy" version of R_0 where the substitution rule is replaced by the automated discovery of interesting substitutions thanks to the use of unification. Formally, we immediately have that $E \vdash_R C$ implies $E \vdash_{R_0} C$. The converse is not true: R does not allow all the substitutions performed in R_0, but it can do *enough* in the following sense.

Lemma 1.5 (Lifting). If $E \vdash_{R_0} C$ then there exists C' and θ such that $E \vdash_R C'$ and $C'\theta = C$.

The refutational completeness of R immediately follows from the refutational completeness of R_0 and the lifting lemma, because the empty clause has no free variable to substitute.

2 Herbrand theorems

We have seen the following result, which can be proved by analyzing rule permutabilities in cut-free LK_1 derivations.

Theorem 2.1 (Herbrand's theorem)**.** Let $\phi_1, \ldots, \phi_n$ be quantifier-free formulas such that $\vdash \exists \vec{x_1}.\ \phi_1, \ldots, \exists \vec{x_n}.\ \phi_n$ is derivable in LK_1. Then, for each $i \in [1; n]$, there exist substitutions $(\theta^i_j)_{1 \leq j \in [1; k_i]}$ such that

$$\vdash \phi_1\theta^1_1, \ldots, \phi_1\theta^1_{k_i}, \ldots, \phi_n\theta^n_1, \ldots, \phi_n\theta^n_{k_n}$$

is also derivable.

Due to the symmetries of sequent calculus, the theorem can be equivalently stated with universal quantifications on the left hand-side of sequents.

Exercise 2.2. Show that this result does not hold if the formulas ϕ_i are allowed to contain quantifiers.

In order to introduce another "Herbrand theorem", we need to define *Herbrand structures.* This notion only makes sense under the assumption that the set of closed terms built from $\mathcal{F}$ is non-empty.

Exercise 2.3. Characterize the signatures $\mathcal{F}$ such that $\mathcal{T}(\mathcal{F}, \emptyset) \neq \emptyset$. Why is this a mild assumption when one is considering the satisfiability problem?

Definition 2.4. A Herbrand structure is an $\mathcal{F}, \mathcal{P}$-structure whose domain is the $\mathcal{F}$-algebra $\mathcal{T}(\mathcal{F}, \emptyset)$. In other words, it interprets terms by closed terms[1]. A Herbrand model can simply be represented as a set of closed atoms, i.e. a subset of $\mathbb{H}$ defined as follows:

$$\mathbb{H} \stackrel{def}{=} \left\{ P(t_1, \ldots, t_n) \mid P \in \mathcal{P} \text{ of arity } n, \text{ and } t_i \in \mathcal{T}(\mathcal{F}, \emptyset) \text{ for all } i \right\}$$

More precisely, the set $H \subseteq \mathbb{H}$ is seen as the structure $\mathcal{S}$ of domain $\mathcal{T}(\mathcal{F}, \emptyset)$ with, for all $P \in \mathcal{P}$ of arity n, $P_{\mathcal{S}} = \{(t_1, \ldots, t_n) \mid P(t_1, \ldots, t_n) \in H\}$.

Example 2.5. Assume $\mathcal{F} = \{0, s\}$. We are usually considering the axioms of elementary arithmetic with function symbols for addition and multiplication, but

[1] Recall that $\mathcal{T}(\mathcal{F}, \emptyset)$ stands for the $\mathcal{F}$-agebra $\mathcal{A}$ of domain $\mathcal{T}(\mathcal{F}, \emptyset)$ with, for all $f \in \mathcal{F}$ of arity n, $f_{\mathcal{A}}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$.

we can also do so without changing much by using addition and multiplication as (ternary) predicate symbols. For instance, the axioms for addition become

$$\forall x.\ \mathsf{add}(0,x,x) \quad \text{and} \quad \forall x,y.\ \mathsf{add}(x,y,z) \Rightarrow \mathsf{add}(s(x),y,s(z)).$$

Recall that elementary arithmetic admits a standard model of domain $\mathbb{N}$ but also non-standard ones featuring copies of $\mathbb{Z}$ or points at infinity, where e.g. addition may not be commutative.

- The canonical model, of domain $\mathbb{N}$, is not a Herbrand structure. However, it is isomorphic to a Herbrand structure of domain $\mathcal{T}(\mathcal{F},\emptyset) = \{s^k(0) \mid k \in \mathbb{N}\}$.
- Non-standard models of arithmetic (featuring copies of $\mathbb{Z}$ or points at infinity) are not isomorphic to any Herbrand structure over $\mathcal{F}$.

A Herbrand *model* wrt. some set S of formulas is simply a Herbrand *structure* that satisfies all formulas of S.

Theorem 2.6 (Herbrand's theorem for satisfiability)**.** Let E be a set of purely universal formulas, i.e. of the form $\forall\vec{x}.\ \phi$ with ϕ quantifier-free. Then E has a model iff it has a Herbrand model.

Exercise 2.7. Prove that theorem 2.6 does not hold if formulas ϕ are allowed to contain quantifiers.

Exercise 2.8. Prove theorem 2.6 by building a Herbrand model from an arbitrary model $\mathcal{S}$ of E.

We propose a proof that relies on sequent calculus, to better relate the two Herbrand theorems.

Proof. We show the non-trivial direction: assuming E admits no Herbrand model, we need to establish that it admits no model at all, i.e. that it is unsatisfiable.

Consider the set E^g of ground instances of E (i.e. the set of closed and quantifier-free formulas obtained by instantiating universal quantifiers with arbitrary closed terms): rev. 1

$$E^g \stackrel{def}{=} \{\phi\theta \mid \forall\vec{x}.\ \phi \in E \text{ and } \phi\theta \text{ is closed}\}$$

Like E, the set E^g has no Herbrand model: indeed, for every Herbrand structure there exists a formula $\forall\vec{x}.\ \phi$ of E such that $\mathcal{S} \not\models \forall\vec{x}.\ \phi$, so there exists a substitution[2] such that $\mathcal{S}, \theta \not\models \phi$, or equivalently $S \not\models \phi\theta$.

Now, E^g can be seen as a set of propositional formulas over the set of propositional variables $\mathcal{P}^g := \{P(t_1, \ldots, t_n) \mid P \in P, t_i \in \mathcal{T}(\mathcal{F}, \emptyset)\}$. If E^g had a propositional model, it would immediately yield a Herbrand model. Thus E^g is propositionally unsatisfiable and, by completeness of LK_0, we have a derivation of $F \vdash \bot$ for a finite $F \subseteq E^g$. A fortiori, we have a derivation of $F^g \vdash \bot$ for a finite $F \subseteq E$. From there we immediately obtain a derivation of $F \vdash \bot$. □

We finally show that the first Herbrand theorem is a consequence of the second. The direct proof of theorem 2.1 using rule permutations is however more informative (it tells us something about the size and structure or the resulting derivation) and also more versatile (rule permutations can be used in many situations).

Proof of theorem 2.1, assuming theorem 2.6. Assume $\forall\vec{x_1}.\phi_1, \ldots, \forall\vec{x_n}.\phi_n \vdash \bot$ is derivable. By soundness of LK_1, this means that $E := \{\forall\vec{x_i}.\phi_i \mid 1 \leq i \leq n\}$ is unsatisfiable. We cannot immediately conclude that the set of ground instances E^g is unsatisfiable: it could be that some structure $\mathcal{S}$ falsifies some $\forall x.\phi$ only because of some instantiation of x that is not the interpretation of any closed term. However, we have that E^g admits no Herbrand model. It is thus unsatisfiable by theorem 2.6. By completeness of LK_1 we thus have a derivation of $F \vdash \bot$ for a finite $F \subseteq E^g$, which allows us to conclude: F is a finite set of instantiations of E. □

[2] Semantic assignments are simply closed substitutions when working with Herbrand structures.

3 Semantic trees

We assume that $\mathcal{F}$ and $\mathcal{P}$ are countable, and consider an arbitrary enumeration of closed atoms, i.e. some $(A_i)_{i \in \mathbb{N}}$ such that $\mathbb{H} = \{A_i \mid i \in \mathbb{N}\}$.

Definition 3.1 (Partial interpretation)**.** A partial interpretation of size n is a set of literal $I = \{L_i\}_{0 \leq i < n}$ with $L_i \in \{A_i, \neg A_i\}$ for all $0 \leq i < n$. We say that a partial interpretation I falsifies a clause $\forall \vec{x}.C$, written $I \not\models \forall \vec{x}.C$, when there exists a ground instance $C\theta$ such that, for each literal L of $C\theta$, we have $\overline{L} \in I$.

Intuitively, a partial interpretation is a Herbrand structure where a truth value has been assigned to only a subset of atoms: $A \in I$ means that A is satisfied in I, $\neg A \in I$ means that A is not satisfied in I, but it is also possible that neither A nor $\neg A$ belongs to I, which means that the status of A is still undecided in the partial interpretation. We will not use a notion of satisfaction for partial interpretations (i.e. $I \models C$) because, as soon as a clause features universally quantified variables, we would have to consider all possible instantiations for these variables, and would thus need to know the status of infinitely many atoms in I, which is not possible as I is finite.

Partial interpretations can be ordered by inclusion, which yields an infinite directed tree, with the empty interpretation at its root and a path from I to I' iff $I \subseteq I'$. We make this formal in the next definition, where the *depth* of a node in a tree is the distance between the root and that node — in particular, the depth of the root is 0.

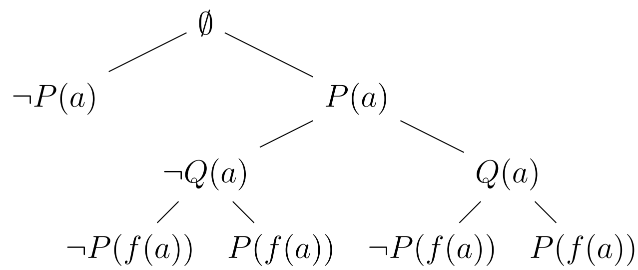
Definition 3.2 (Semantic tree)**.** A semantic tree is a possibly infinite tree where each node of depth n is labeled by a partial interpretation of size n. Moreover, a node of depth n with label I that is not a leaf must have exactly two child nodes respectively labeled $I \cup \{A_i\}$ and $I \cup \{\neg A_i\}$.

Definition 3.3 (Tree of a set of clauses, $T(E)$)**.** Given a set of clauses E, the semantic tree of E, noted $T(E)$, is the unique semantic tree such that, for every node $N \in T(E)$ and I such that I is the label of N, N is a leaf iff there exists a clause $C \in E$ such that $I \not\models C$.

Example 3.4. Assume $\mathcal{F}$ contains a constant symbol a and a unary function symbol f. Assume $\mathcal{P}$ contains two unary predicate symbols P and Q. Consider the following set of clauses:

$$E = \{ \quad \forall x.P(x), \quad \forall x.\neg P(f(x)) \vee \neg Q(a), \quad Q(a) \vee \neg P(f(a)) \quad \}$$

If the enumeration of closed atoms is $P(a), Q(a), P(f(a)), P(f(f(a)))\dots$ then $T(E)$ is the following tree, where we do not display the part of partial interpretations that is inherited from parent nodes:



Leaves labeled $\neg P(\_)$ falsify $\forall x.P(x)$. The leftmost leaf labeled $P(f(a))$ falsifies $Q(a) \vee \neg P(f(a))$, and the rightmost one falsifies $\forall x.\neg P(f(x)) \vee \neg Q(x)$. The nodes labeled $Q(a)$ or $\neg Q(a)$ do not falsify any clause, because they do not give an interpretation for any $P(f(\_))$ literal yet.

The clause $\neg Q(a)$ can be obtained by resolution from the first two clauses. In general, adding resolution consequences to a set of clauses does not change its satisfiability; here it obviously remains unsatisfiable. The semantic tree of the extended set of clauses is shorter: the node labeled $Q(a)$ becomes a leaf.

Proposition 3.5. Let E be a set of clauses. If $T(E)$ contains an infinite branch, then the set of positive literals along that branch forms a Herbrand model of E. If $T(E)$ is finite, then E is unsatisfiable.

Proof. Assume $T(E)$ has a infinite branch. Its positive labels define a Herbrand structure $H \subseteq \mathbb{H}$. Assume, by contradiction, that there is some clause $(\forall \vec{x}.\, C) \in E$ such that $H \not\models \forall \vec{x}.C$. Then there exists a substitution θ such that $H \not\models C\theta$. There is a node N with label I in our infinite branch such that all literals of $C\theta$ are determined in the partial interpretation I. Moreover, $H \not\models C\theta$ implies $I \not\models C\theta$ by definition of H and because I is in our branch. This contradicts the fact that the branch is infinite.

Conversely, any Herbrand model H of E yields an infinite branch: it is the Rev. 7
branch containing all the partial interpretations that agree with H, i.e. all interpretations I such that for all closed literals L we have $L \in I$ iff $H \models I$. Indeed, this branch cannot contradict any instance of a clause of E.

Finally, if $T(E)$ is finite then it does not have any infinite branch, thus E does not have a Herbrand model, and by theorem 2.6 it is unsatisfiable. □

Theorem 3.6 (Refutational completeness of R_0)**.** The system R_0 is complete.

Proof. Let E be a set of clauses that is unsatisfiable. Let $E^* := \{C \mid E \vdash_{R_0} C\}$ be the set of clauses obtained from E by applying the rules of R_0. Since E^* is unsatisfiable, $T(E^*)$ is finite. Assume by contradiction that this tree is not reduced to its root, and consider an internal node N of maximal depth, which must thus have two leaves as children.

Note that, if a node N of $T(E^*)$ falsifies some clause $C \in E^*$ then it also satisfies a ground clause, because E^* is closed under the substitution rule of R_0.

Let n be the depth of N and I its label. Its two children $N_\bot$ and $N_\top$ are respectively labeled $I_\bot = I \cup \{A_n\}$ and $I_\top = I \cup \{\neg A_n\}$. There is a ground clause $C_\bot \in E^*$ that is falsified by $N_\bot$ but not by N, it must thus be of the form $C'_\bot \vee \neg A_n \vee \ldots \vee \neg A_n$. Similarly, there exists a ground clause $C_\top \in E^*$ of the form $C'_\top \vee A_n \vee \ldots \vee A_n$. Because E^* is closed under resolution and factorisation, it must thus also contain $C'_\bot \vee C'_\top$. We have $I \not\models C'_\bot$ because $I_\bot \not\models C'_\bot$ and because this clause only contains literals over $\{A_i\}_{i<n}$, which have the same value in I and $I_\bot$. Similarly, $I \not\models C'_\top$ and thus $I \not\models C'_\bot \vee C'_\top$, so our node N should be a leaf of $T(E)$: contradiction. □

The refutational completeness of R follows from the previous result and the lifting lemma (lemma 1.5). Alternatively, the above proof can be adapted to use R instead of R_0 to obtain a direct proof of refutational completeness for R.

Another variant of the proof consists in considering the tree $T(E)$ instead of $T(E^*)$. That tree is not necessarily reduced to its root. However it is still finite and one can show, by induction over the height[3] of its nodes, that for every node N labeled I there exists a clause $C \in E^*$ that is falsified by I. Underlying that constructive proof we have an algorithm deriving the empty clause (for the root node) from the clauses that are falsified at the leaves of $T(E)$.

4 Ordered resolution

The proof of theorem 3.6 relies on an arbitrary enumeration $(A_i)_{i \in \mathbb{N}}$ of the atoms of $\mathbb{H}$. This actually yields a proof of completeness for a resolution strategy based on an ordering derived from this enumeration.

Define $A \leq A'$ when A appears before A' in the enumeration. This total order is lifted to a quasi-order[4] on closed literals by ignoring negations: $L \leq L'$ when $A \leq A'$ for the atoms A and A' respectively contained in L and L'.

[3] The height of a node is the height of the subtree rooted at that node.

[4] Antisymmetry is lost, we only have reflexivity and transitivity.

We abusively say that a closed literal L is maximal in a ground clause C when there is no literal L' in C such that $L < L'$. Note that we do not require that L actually belongs to C[5]. rev. 2

Example 4.1. Assume $A < A' < A''$. The literal $\neg A'$ is maximal in A and $A \vee \neg A'$ but not in $A \vee A''$.

We now define a restriction of resolution based on this ordering of literals.

Definition 4.2 (Ground ordered resolution, OR_0)**.** The system OR_0 is given by the following three rules, featuring the same substitution rule as R_0 but factorisation and resolution rules that are constrained to apply on *maximal* literals of *ground* clauses:

$$\frac{C}{C\sigma} \qquad \frac{C \vee L \vee L}{C \vee L}\ L \text{ maximal in } C$$

$$\frac{C \vee L \quad \overline{L} \vee C'}{C \vee C'}\ L \text{ maximal in } C, \text{ and } \overline{L} \text{ maximal in } C'$$

Theorem 4.3. System OR_0 is refutationally complete.

Proof. The argument for proving theorem 3.6 works here, since the factorisations and resolutions performed in the argument apply to maximal literals. □

To make this useful in practice, we need to lift the constraints of OR_0 to the resolution rule with unification of R.

Definition 4.4 ($A \prec A'$, $L \prec L'$)**.** Assume a relation $\prec$ on atoms with free variables such that:

- for any closed atoms A and A', $A \prec A'$ implies $A < A'$;
- for any atoms A and A', $A \prec A'$ implies $A\theta \prec A'\theta$ for any substitution θ.

This relation is lifted to literals as before by ignoring negations. We say that a literal L is maximal in C when this clause contains no literal L' such that $L \prec L'$.

Definition 4.5 (Ordered resolution, OR)**.** The system of ordered resolution, noted OR, is given by the two rules of R, i.e.

$$\frac{C \vee L \quad \overline{L'} \vee C'}{(C \vee C')\sigma}\ \sigma = \mathsf{mgu}(L, L') \qquad \frac{C \vee L \vee L'}{(C \vee L)\sigma}\ \sigma = \mathsf{mgu}(L, L')$$

with the following constraints:

[5] It would be more appropriate to say that L is an upper bound of C, but that terminology would not carry nicely to the next steps of our development.

- for resolution: L is maximal in C and $\overline{L'}$ is maximal in C';
- for factorisation: L and L' are maximal in C.

Example 4.6. Assume a constant symbol a and two unary predicate symbols P and Q. Consider the relation $\prec$ defined by:

- $P(a) \prec P(t)$ for all $t \neq a$;
- $Q(a) \prec P(t)$ for all t.

This relation satisfies the second condition of definition 4.4. It also satisfies the first one e.g. if the enumeration starts with $Q(a)$ immediately followed by $P(a)$.

Lemma 4.7 (Lifting for ordered resolution)**.** If $E \vdash_{OR_0} C$ then there exists C' and θ such that $E \vdash_{OR} C'$ and $C'\theta = C$.

Proof. We show that we can simulate in OR the application of an arbitrary number of substitution steps followed by either resolution or factorisation of OR_0. Since an arbitrary number (possibly zero) number of substitution steps can be combined in one, we can consider wlog. a situation with a single substitution step on each premise of the resolution or factorisation rule of OR_0.

Consider an OR_0 derivation of the following form, where $L\theta = L'\theta'$, and $L\theta$ is maximal in $C\theta$, and $L'\theta'$ is maximal in $C'\theta'$:

$$\frac{\dfrac{C \vee L}{C\theta \vee L\theta} \quad \dfrac{\overline{L'} \vee C'}{\overline{L'}\theta' \vee C'\theta'}}{C\theta \vee C'\theta'}$$

Because L and L' are unifiable, they have a most general unifier σ. We have that L is maximal in C: if there was some L' in C such that $L \prec L'$, then we would have $L'\theta$ in $C'\theta$ such that $L\theta \prec L'\theta$ and thus $L\theta < L'\theta$, contradicting the maximality of $L\theta$ in $C\theta$. Similarly, L' is maximal in C'. We can thus form the following derivation in OR, resulting in a more general clause than before:

$$\frac{C \vee L \quad \overline{L'} \vee C'}{C\sigma \vee C'\sigma}$$

For factorisation, we consider a derivation as follows, where $L\theta = L'\theta$ and this literal is maximal in $C\theta$:

$$\frac{\dfrac{C \vee L \vee L'}{C\theta \vee L\theta \vee L'\theta}}{C\theta \vee L\theta}$$

Since L and L' have a common instance, they have a most general unifier σ. We verify that both L and L' are maximal in C: if there were some L'' in C such that $L \prec L''$ we would have $L\theta < L''\theta$, contradicting the maximality of $L\theta$ in $C\theta$. We can thus form the following derivation in OR:

$$\frac{C \vee L \vee L'}{C\sigma \vee L\sigma}$$

□

As expected, this lemma allows to lift the refutational completeness of OR_O into that of OR.

Theorem 4.8. Ordered resolution (OR) is refutationally complete.

In the above presentation, we have started from an arbitrary enumeration of closed atoms, to obtain an ordering $<$ on closed atoms and literals, and we have defined ordered resolution based on some an ordering $\prec$ that needs to be compatible with $<$ and substitution. We can further refine our analysis to obtain a more convenient strategy: as we shall see, it suffices to consider a relation $\prec$ that is compatible with substitution and which yields a strict partial order on closed atoms.

It is not obvious how one can reconstruct a total order $<$ (and hence an enumeration) from the partial order induced by $\prec$ on closed atoms. For instance, one can simply take the relation defined by $P(t) \prec Q(t')$ for any t and t'. But we cannot enumerate all closed terms (using indices in $\mathbb{N}$) by putting *all* closed atoms $P(t)$ before closed atoms $Q(t')$…unless there are finitely many terms.

The last remark is the key to make things work. Consider an unsatisfiable set of clauses E. By compacity and Herbrand's theorem, there exists a finite subset of E^g that is still unsatisfiable. This finite subset induces a finite set of terms that is sufficient for our argument: If we consider an enumeration of the finitely many closed atoms obtained using these terms, we can still obtain a finite semantic tree for E, and obtain from it a refutation using resolution. Moreover, this refutation will satisfy the constraints of ordered resolution as long as the enumeration is compatible with the partial order induced by $\prec$ on the finitely many closed atoms, which can be obtained by taking any topological sorting of these atoms.

Exercise 4.9. Consider the order defined by $P(a) \prec Q(\_) \prec P(f(\_))$ and the set of clauses

$$P(x) \vee P(f(x)) \vee \neg Q(f(x)), \quad Q(f(a)), \quad \neg P(x) \vee P(a), \quad \neg P(a)$$

Identify maximal literals in each clause, then give a derivation of $\bot$ using ordered resolution.

Exercise 4.10. Consider the following set of clauses:

$$P(a) \vee \neg Q(f(x)), \quad P(f^3(x)) \vee \neg P(f(x)), \quad Q(a) \vee P(f(x))),$$

Show that it is satisfiable without exhibiting a model: it suffices to give an order $\prec$ for which the rules of ordered resolution do not apply to the clauses, hence $\bot$ is obviously not derivable.

5 Horn clauses

We introduce the class of Horn clauses. It has many practical applications, the most famous one being Prolog, the first logic programming language. Several resolution strategies that are in general incomplete turn out to be complete for Horn clauses.

Definition 5.1. A Horn clause is a clause featuring at most one positive literal. It is *definite* when it features exactly one positive literal.

Example 5.2. The clause $\forall x.P(x) \vee Q(s(x))$ is not a Horn clause. The clauses

$$\forall x.P(x), \quad A \vee \neg A \quad \text{and} \quad \forall x.\, \neg P(x) \vee P(s(x))$$

are definite Horn clauses. The empty clause, and $\forall x_1, x_2.\, \neg R(x_1, x_2) \vee \neg R(x_2, x_1)$ are Horn clauses but are not definite.

Horn clauses are better understood as implications. Accordingly, we will often write a clause $\forall \vec{x}.\, \neg A_1 \vee \ldots \neg A_n \vee A$ as $\forall \vec{x}.\, A_1 \wedge \ldots \wedge A_n \Rightarrow A$.

We will consider several resolution strategies in this section — examples will be given by the end of the section.

Definition 5.3 (Unit resolution, UR)**.** Restrict the resolution rule so that one clause is unitary, i.e. is restricted to a literal.

Definition 5.4 (Negative resolution, NR)**.** Restrict the resolution rule so that one clause is negative, i.e. contains only negative literals.

Definition 5.5 (Input strategy, IR)**.** Fix an initial set of clauses E, and restrict the resolution rule to have one of its premises in E.

Definition 5.6 (Selection strategy, SR)**.** Fix an arbitrary function f which, given a clause, returns one of its literals, called the selected literal of that clause. Restrict the resolution rule so that the literals on which resolution is performed are selected in their respective clauses.

Although we will not prove it, it is the case that all of the above strategies are refutationally complete for Horn clauses. Negative resolution is even refutationally complete in general.

Exercise 5.7. Show that the unit and input strategies are refutationally incomplete in general. We shall see that the selection strategy generalizes the unit strategy, it is thus also refutationally incomplete in general. rev. 3

5.1 Herbrand models of Horn clauses

To study the Herbrand models of a set of Horn clauses, it is useful to start by focusing on definite clauses and to consider the following operator.

Definition 5.8. Let C be a set of definite Horn clauses. We define $F_C : 2^{\mathbb{H}} \to 2^{\mathbb{H}}$ as follows: $F_X(H) = \{A\theta \in \mathbb{H} \mid \text{ there exists } (\forall \vec{x}.\ \bigwedge_{1 \leq i \leq n} A_i \Rightarrow A) \in C \text{ and } A_i\theta \in H \text{ for all } 1 \leq i \leq n\}$.

This operator yields a characterization of Herbrand models which helps to uncover some of their properties.

Proposition 5.9. Let C be a set of definite Horn clauses. A Herbrand structure $H \subseteq \mathbb{H}$ is a model of C iff $F_C(H) \subseteq H$.

Proof. Immediate by definition of F_C. □

Proposition 5.10. The set of Herbrand models of C is closed under arbitrary intersections, and admits a least element wrt. set inclusion.

Proof. A set $H \subseteq \mathbb{H}$ is a model of C iff $F_C(H) \subseteq H$. Let H_1 and H_2 be two Herbrand models of C. By monotonicity of F_C we have $F_C(H_1 \cap H_2) \subseteq F_C(H_1) \subset H_1$ and similarly with H_2, thus $F_C(H_1 \cap H_2) \subseteq H_1 \cap H_2$, i.e. $H_1 \cap H_2$ is a model.

This reasoning for binary intersection generalizes to arbitrary intersections. Thus the intersection of all Herbrand models of C is still a model of C. It is also included in all models of C by construction. □

Exercise 5.11. What is the greatest model of a set of definite Horn clauses ?

Exercise 5.12. Give a set of (non-Horn) clauses which admits models that are not closed under intersection, and do not admit a least model. Then give a set which admits a least model but whose models are not closed under intersection.

It can further be shown that the least model of a positive set of Horn clauses is the least fixed point of F_C, which can be obtained as the limit of its iterations.

Exercise 5.13. Give a model of $C = \{A \Rightarrow B\}$ that is not a fixed point of F_C.

Proposition 5.14. Let C be a set of definite Horn clauses and D be a set of negative Horn clauses. The set $C \cup D$ is unsatisfiable iff the least Herbrand model of D falsifies a clause of D.

Proof. Any Herbrand model H of C that is also a model of D is a model of $C \cup D$. Moreover, if H is a model of D and $H' \subseteq H$, then H' is still a model of D. Hence, $C \cup D$ admits a Herbrand model iff it the least Herbrand model of H is a model of D. □

5.2 Completeness of unit resolution

The previous results yield a proof of refutational completeness for unit resolution, a resolution strategy which closely mimicks F_C.

Proposition 5.15. Let C be a set of definite Horn clauses, and let H be its least Herbrand model. For any $A \in \mathbb{H}$ we have $H \models A$ iff $C \vdash_{UR} A$.

Proof. By the soundness of resolution, $C \vdash_{UR} A$ implies $H \models A$. We need to show the converse direction, which can be rephrased as $H \subseteq S$ with $S := \{A \mid C \vdash_{UR} A\}$. To conclude, it suffices to observe that $F_C(S) \subseteq S$. □

Theorem 5.16. Unit resolution is refutationally complete for Horn clauses.

Proof. Consider an unsatisfiable set of Horn clauses $C \cup D$ partitioned into definite and negative clauses. By proposition 5.14 and proposition 5.10 there is a clause $\neg A_1 \vee \ldots \vee \neg A_n$ in D and a substitution σ such that $A_1\sigma, \ldots, A_n\sigma$ all belong to the least Herbrand model of C. By proposition 5.15 we have $C \vdash_{UR} A_i\sigma$ for each i, and thus $C, D \vdash_{UR} \bot$. □

5.3 Discussion and applications

Unit resolution is a form of *forward* proof search, starting from the atoms initially known to derive more atoms that are consequences of Horn clauses. The derived atoms may eventually be used to contradict a negative Horn clause. This structure is shown on the following example, using the definite Horn clauses $\Rightarrow A$, $\Rightarrow B$ and $B \Rightarrow C$ and the negative Horn clause $A \wedge C \Rightarrow \bot$:

$$\dfrac{\Rightarrow A \qquad \dfrac{\dfrac{\Rightarrow B \quad B \Rightarrow C}{\Rightarrow C} \qquad A \wedge C \Rightarrow \bot}{A \Rightarrow \bot}}{\bot}$$

In general, performing unit resolution yields to the infinite enumeration of all true closed atoms, which is infeasilble or at least very expensive.

In contrast, negative resolution is a form of *backward* proof search: one first resolves a negative clause against a definite clause, obtaining a new negative clause, which can be resolved against another definite clause, etc. We show next how to obtain a refutation using negative resolution with the clauses from the previous example:

$$\dfrac{\Rightarrow A \qquad \dfrac{\Rightarrow B \qquad \dfrac{B \Rightarrow C \quad A \wedge C \Rightarrow \bot}{A \wedge B \Rightarrow \bot}}{A \Rightarrow \bot}}{\bot}$$

In logic programming, one sees a set of definite clauses as a program, and a single negative clause as a query. Searching for a refutation of all these clauses is the execution of the logic program, whose result is the resulting substitution. Negative resolution is used to obtain a simple, predictable execution strategy: at each step a part of the query (negative clause) is replaced by new subgoals, until none remains. For example, with the Horn clauses that specify addition in elementary arithmetic (see example 2.5) and the query $\neg\mathsf{add}(x, s(0), s(s(s(0))))$ the execution is successful and returns the substitution $x \mapsto s(s(0))$.

The resolution strategy allows to tune the direction of reasoning (either forward or backward) differently for each clause. Consider for instances the following clauses, where the selected literal is underlined:

$$\Rightarrow \underline{P(g(a))}, \quad P(x) \Rightarrow \underline{P(f(x))}, \quad \underline{P(g(x))} \Rightarrow P(x), \quad \underline{P(f(a))} \wedge P(a) \Rightarrow \bot$$

On this example, unit resolution can derive all unit clauses $P(f^i(g(a)))$ and $P(f^i(a))$. Negative resolution can generate all negative clauses $P(g^i(f(a))) \wedge P(a) \Rightarrow \bot$. However, resolution with selection can only generate $\underline{P(a)}$ and $\underline{P(a)} \wedge P(a) \Rightarrow \bot$, and then $\underline{P(a)} \Rightarrow \bot$, and finally $\bot$.

Exercise 5.17. Show that both negative resolution and unit resolution are particular cases of resolution with selection, by exhibiting an appropriate selection function in each case.