Logic ch 2.14

Sherri Shulman
TESC, CS

Mar 9, 2011

Chapter 2.14: Esercises p 157: 2,3

2 State and prove the soundness theorem for ordered resolution for predicate logic.

Well first we have to give a definition of ordered resolution which is really problem 1! Sorry about that. OR we can use the definition in the next section on p 160:

Definition 1.3:

i If $G = \{\neg A_0, \cdots, \neg A_n\}$ and $C = \{B, \neg B_0, \cdots, \neg B_m\}$ are ordered clauses and θ is an mgu for A_i and B, then we can perform an ordered resolution of G and C on the literal A_i. The (ordered) resolvent of this resolution is the ordered clause $\{\neg A_0, \cdots \neg A_{i-1}, \neg B_0, \cdots, \neg B_m, \neg A_{i+1}, \cdots, \neg A_n\}\theta$.

ii This part of the definition is for linear definite or LD-refutation, so I'll leave it out.

(Note that this is a more restrictive version of an ordered clause than I need to have, since this requires the clauses to have a particular form vis a vis the positive, negative status. If we use the model of I.9.6, then we do not need to restrict ourselves to these clauses but can use any clauses, and provide an order by assigning an index to each of the propositional letters.)

So I'll give a new definition: Assume we have indexed all the propositional letters. We define $\mathcal{R}^<$ or ordered resolution as usual, except that we only allow resolutions of $C_1 \sqcup \{p\}$ and $C_2 \sqcup \{\overline{p}\}$ when p has a higher index than any propositional letter in C_1 or C_2.

So, now we want to state the soundness theorem for ordered resolution. Soundness (in its most simple form) says that if we have a proof of some clause then that clause is valid.

Now, we already have a proof of soundness of resolution (THeorem 13.6): If $\Box \in \mathcal{R}(S)$, then S is unsatisfiable. Recall that a proof is a refutation, so that we add the negation of what we want to prove to a set of clauses that themselves are satisfiable. So that if the addition of the negation of what we want to prove yields $\Box$ then the unnegated goal must be a consequence of the set of clauses.

In general, what we are saying here is that if $S \in UNSAT$ (see Thm 8.21), then after a resolution step we have the set of clauses S', the resulting set of clauses $S' \in UNSAT$. This may seem to be an indirect way to establish soundness. However, if it were not the case, then a set $\in UNSAT$ could (after the resolution step) be found to be $\in SAT$. So we would produce a proof of something for which there was no interpretation (the essence of unsoundness.)

In Chapter I.9 we have Lemma 9.2, which states that any restriction of a sound method, ie one that allows fewer deductions than the sound method, is itself sound. The intuition here is that if resolution is sound, then if we restrict resolution, the method is still sound (I may introduce incompleteness, but not unsoundness.)

By Theorem 13.6 we also know that if $\Box \in \mathcal{R}(S)$ then S in unsatisfiable. Since we know that resolution is sound, the restriction of resolution to an ordered clause is also sound. However, it is possible that if we do a resolution step then we may no longer have ordered resolution (ie after doing a resolution step as above, the required deduction would require us to violate the above order requirement.

Following on the hint in the textbook, we follow Definition I.9.6 and Theorem I.9.7 and give an ordering to all the predicate symbols occuring in our ordered clauses. We're going to restrict $\mathcal{U}$ (the set UNSAT) to $\mathcal{U}^<$ be the set defined inductively by

i $\Box \in S \Rightarrow S \in \mathcal{U}^<$

ii If no predicate letters with index strictly smaller than that of p occurs in S, $S^p \in \mathcal{U}^<$ and $S^{\overline{p}} \in \mathcal{U}^<$, then $S \in \mathcal{U}^<$.

(This is the same set as before, only with the order requirement.)

Now this puts an order on the formulas in $\mathcal{U}^<$. And the sets S^p and $S^{\overline{p}}$ capture what is unsatisfiable: if set of clauses require that p and $\overline{p}$ be true, then it must be unsatisfiable. This definition is based on Thm I.9.7 which itself is based on Thm I.8.21. What we've done here is restrict the resolution so we want to know the relation of $\mathcal{U}^<$ to $\mathcal{U}$. Since our inductive step is a restriction of the full definition of $\mathcal{U}$, just restricting the set of clauses to those that can be constructed by choosing resolvents that obey the order requirement. So this clearly means that $\mathcal{U}^< \subseteq \mathcal{U}$.

In the other direction, we must show that $UNSAT \subseteq \mathcal{U}$. We can list the $\{p_i\}$ occurring in S in ascending order of their indices and suppose that $S \not\in \mathcal{U}^<$ then we want to show that S is satisfiable. Let $\{p_i\}$ list the propositional letters in their assigned order, such that p_i or $\overline{p_i}$ occurs in a clause of S, and define by induction the sequence $\{l_i\}$ such that $l_i = p_i$ or $l_i = \overline{p_i}$, and $S^{l_1,\cdots,l_i} \not\in \mathcal{U}^<$. (Property ii guarantees we can always find such an l_i.) Now let $\mathcal{A} = \{l_i | i \in \mathcal{N}\}$, then A satisfies S. ($\mathcal{A}$ is the assignment of truth values to the predicate symbols, or since we are looking at predicate logic, $\mathcal{A}$ is a structure based on the Herbrand model). We assumed $S \not\in \mathcal{U}^<$, so we have to show that S is satisfiable. If it is satisfiable, then $A \models C$ for $C \in S$. C is a finite clause, so there is some n such that $i < n$ the clause C is composed of a sequence of literals in $S^{l_1,\cdots,l_i}$. At each resolution step, we remove l_i (by the order requirement of $\mathcal{R}^<$). Since all the literals in C are in the sequence of resolvents $\overline{l_i}$, C must eventually become $\Box$ in $S^{l_1,\cdots,l_n}$. So that contradicts our assumption that $S \not\in \mathcal{U}^<$, so that is our desired contradiction.

3 State and prove the completeness theorem for ordered resolution for predicate logic.

If S is unsatisfiable, then there is an ordered resolution refutation of S, ie $\Box \in \mathcal{R}^<(S)$.

We already know from Theorem 14.4 that if $S \in UNSAT$ and if U is a set of support for S, then there is a linear refuation of S with support U.

According to definition 14.3, the support for a set of clauses are the clauses that are actually used in a (linear) resolution proof.

The issue here is that if $S \in UNSAT$, is there an ordered clause resolution proof? If $P \cup \{G\}$ is an unsatisfiable set of clauses, and U is a set of support for S, then there is a linear refutation of S with support U. (Thm 14.4). A linear refutation of $\Box$ from

S is a sequence $\langle C_0, B_0 \rangle, \cdots, \langle C_n, B_n \rangle$ of pairs of clauses such that C_0 and each B_i are either renaming substitutions of elements of S or some C_j for $j < i$; each $C_{i+1}, i \leq n$ is a resolvent of C_i and B_i; and $C_{n+1} = \square$. The only difference between this resolution and resolution with ordered clauses is that ordered clauses specify an order on the predicate symbols and we perform the resolution step on the least literal in this order.

Assume there were no such order and that S is minimally unsatisfiable (see definition 14.5). Then there must be two resolvent pairs in the sequence such that $\langle C_i, B_i \rangle$ and $\langle C_k, B_k \rangle$ and C_i and C_k with $i < k$ differing at some literal m and n, and $m < n$ (ie C_i is resolved before C_k on literal l_m, and subsequent to this C_k is resolved on literal l_n. Because we assumed that S is minimally unsatisfiable, both these resolvent steps are required. But if that is the case, then we can just reorder the resolvent steps so that C_k is resolved before C_i.

What if C_k is a clause that results from the resolution step? If C_i and B_i resolved on literal l_n then the order requirement is met and the subsequent resolution of l_m also obeys the order requirement.

If C_i and B_i resolve on literal l_m, resulting in C_k with literal l_n to be resolved on next with B_k then it must be the case that $C_i = \{l_1, \cdots, l_m, \cdots, l_n\}$ and $B_i = \{\cdots, \overline{l_m}, l_{m+1}, \cdots\}$. The result of the resolution will yield $C_k = \{l_1, \cdots, \cdots, l_{m+1}, \cdots, l_n\}$. And now we resolve on l_n with some B_k. But if that were the case, then we could just resolve on l_n first with clauses C_k, B_k, and obey the order requirement.