Worksheet 10

Soundness and Completeness of Resolution Refutation

The aim of this worksheet is that

- you should note that resolution refutation is sound and complete;
- you should get some idea of the 'story' that proves these properties.

I present a number of theorems. Most I do not prove. I merely sketch proofs or give examples.

Our first theorem, which we will make use of in proving our second theorem, is the Deduction Theorem, which you might have encountered in TOC.

10.1 Deduction Theorem

$\alpha_1, \ldots, \alpha_n \models \beta$ **iff** $(\alpha_1 \wedge \ldots \wedge \alpha_n) \rightarrow \beta$ **is a tautology.**

Proof:

• *Proving If* $\alpha_1, \ldots, \alpha_n \models \beta$ *then* $(\alpha_1 \wedge \ldots \wedge \alpha_n) \rightarrow \beta$ *is a tautology.*

Suppose $\alpha_1, \ldots, \alpha_n \models \beta$. Let M be an arbitrary model. If $\alpha_1, \ldots, \alpha_n$ are true in M, then, by the definition of entailment, β is true in M. Hence, $(\alpha_1 \wedge \ldots \wedge \alpha_n) \rightarrow \beta$ is true in M (the antecedent and consequent are true so the implication is true). On the other hand, if $\alpha_1, \ldots, \alpha_n$ are false in M, then $(\alpha_1 \wedge \ldots \wedge \alpha_n) \rightarrow \beta$ is true in M (the antecedent is false so the implication is true). Thus, we have shown that if $\alpha_1, \ldots, \alpha_n \models \beta$, then $(\alpha_1 \wedge \ldots \wedge \alpha_n) \rightarrow \beta$ is true in any model, M.

- *Proving If* $(\alpha_1 \wedge \ldots \wedge \alpha_n) \rightarrow \beta$ *is a tautology, then* $\alpha_1, \ldots, \alpha_n \models \beta$.

Suppose $(\alpha_1 \wedge \ldots \wedge \alpha_n) \rightarrow \beta$ is a tautology. For any model, M, if $(\alpha_1 \wedge \ldots \wedge \alpha_n)$ is true in M, then for the implication to be true as a whole, β must be true in M. Therefore, β is an entailment of $\alpha_1, \ldots, \alpha_n$. That is, $\alpha_1, \ldots, \alpha_n \models \beta$.

We use this to prove the refutation theorem.

10.2 Refutation Theorem

$\alpha_1, \ldots, \alpha_n \models \beta$ iff $\alpha_1 \wedge \ldots \wedge \alpha_n \wedge \neg\beta$ is inconsistent

Proving this theorem obviously justifies our use of refutation proofs (proofs by contradiction) in place of deduction proofs.

Proof:

By the Deduction Theorem, β is an entailment of $\alpha_1, \ldots, \alpha_n$ iff $(\alpha_1 \wedge \ldots \wedge \alpha_n) \rightarrow \beta$ is a tautology. Hence, $\alpha_1, \ldots, \alpha_n \models \beta$ iff the negation of $(\alpha_1 \wedge \ldots \wedge \alpha_n) \rightarrow \beta$ is inconsistent.

Since

$$\begin{aligned} \neg((\alpha_1 \wedge \ldots \wedge \alpha_n) \rightarrow \beta) &= \neg(\neg(\alpha_1 \wedge \ldots \wedge \alpha_n) \vee \beta) \\ &= \neg\neg(\alpha_1 \wedge \ldots \wedge \alpha_n) \wedge \neg\beta \\ &= \alpha_1 \wedge \ldots \wedge \alpha_n \wedge \neg\beta, \end{aligned}$$

we conclude that $\alpha_1, \ldots, \alpha_n \models \beta$ iff $\alpha_1 \wedge \ldots \wedge \alpha_n \wedge \neg\beta$ is inconsistent.

Here is an *alternative* proof that doesn't make use of the Deduction Theorem.

(Alternative) Proof:

If $\alpha_1, \ldots, \alpha_n \models \beta$, then all the models that satisfy $\alpha_1, \ldots, \alpha_n$ also satisfy β. Hence, none of these can satisfy $\neg\beta$, and thus $\alpha_1 \wedge \ldots \wedge \alpha_n \wedge \neg\beta$ is unsatisfiable. Conversely, suppose $\alpha_1 \wedge \ldots \wedge \alpha_n \wedge \neg\beta$ is unsatisfiable but that $\alpha_1 \wedge \ldots \wedge \alpha_n$ is satisfiable. Let M be a model that satisfies $\alpha_1 \wedge \ldots \wedge \alpha_n$; M does not satisfy $\neg\beta$, because, if it did, $\alpha_1 \wedge \ldots \wedge \alpha_n \wedge \neg\beta$ would be satisfiable. Therefore, M satisfies β. (A model must satisfy one of either β or $\neg\beta$). Since this holds for arbitrary M satisfying $\alpha_1 \wedge \ldots \wedge \alpha_n$, it holds for all M satisfying $\alpha_1 \wedge \ldots \wedge \alpha_n$. Thus, all models that satisfy $\alpha_1 \wedge \ldots \wedge \alpha_n$ also satisfy β, and so β logically follows from $\alpha_1, \ldots, \alpha_n$.

10.3 Skolem Theorem

Let Φ be a set of clauses that represent the normal form of a formula of FOPC α. Then α is unsatisfiable iff Φ is unsatisfiable.

In other words, this theorem claims that our conversion to CFL preserves (un)satisfiability, as I said it did. (We know that all the other steps are fine: they use established equivalences or are mere notational variations. Only the

Skolemisation step requires more justification.) What follows is not a proof, just a sketch of relevant ideas.

10.3.1 An Informal Justification of Skolemisation

Example 1

Suppose our original wff is $\exists X$elephant(X). This is satisfiable. Its Skolemised form, elephant(clyde), is also satisfiable.

It is obvious that in models that make elephant(clyde) true, then $\exists X$elephant(X) is true.

It is also clear that if we have a model that makes $\exists X$elephant(X) true, then we can easily construct one that makes elephant(clyde) true (simply by tweaking the original interpretation function so that the constant clyde denotes (one of) the object(s) that satisfied the elephant predicate).

We see that the Skolemised form is in fact true in a subset of the models in which the original is true.

Example 2

Suppose our original wff is $\exists X$(elephant$(X) \wedge \neg$elephant(X)). This is unsatisfiable. The Skolemised form is elephant(clyde) $\wedge\, \neg$elephant(clyde). This is also unsatisfiable.

What I haven't explained so far is why Skolemisation is so complicated when the existentially quantified variable is in the scope of universally quantified variables. We informally justify this with reference to two further examples.

Example 3

Consider a case where the existential is not in the scope of a universal. For example, $\exists X \forall Y$loves(X, Y). This says that there is at least one thing/person who loves everybody. Skolemisation in this case is like giving this person a name: $\forall Y$loves(god, Y), i.e. god loves everybody (where god is the constant I've put in place of the existentially quantified variable).

Example 4

By contrast, consider a case where the existential is in the scope of a universal. For example, $\forall Y \exists X$loves(X, Y). This says that for everybody there is at least one person who loves them. In this case, we are still all loved but by possibly different people. If we simply replaced the existentially quantified variable X with a constant, then this would not capture the idea that in this case we may be loved by different people. We have to capture the idea that the person who loves us depends on who we are, i.e. can differ from one of us to another. We do this by introducing a function of Y which can be thought of as a function that takes in a person and returns the person who loves them: $\forall Y$loves(mother(Y), Y) (where mother is the function I've used to replace the existentially quantified variable).

10.3.2 Summary

A wff and its corresponding Skolemised form are not equivalent: it is not the case that whenever one is true the other is true. In particular, there are cases where the original will be true when the Skolemised form is not. However, if the original wff is *satisfiable*, its Skolemised form is consistent (and *vice versa*); if the original is *unsatisfiable*, its Skolemised form is unsatisfiable (and *vice versa*). In other words, *Skolemisation preserves (un)satisfiability.* As Skolemisation preserves (un)satisfiability and our proof theory is in terms of proof by contradiction (i.e. determining whether a wff is inconsistent (unsatisfiable)) then this is OK.

10.4 Soundness of Resolution

For clauses α and β, if the resolvent of α and β is γ, then $\{\alpha, \beta\} \models \gamma$.

If we prove this we show that resolution produces only clauses that are licensed by the model theory.

We give the proof for the propositional case. The proof for the case with variables is the same but with a lot of messy notation to deal with the possibility of multiple complementary pairs and to deal with the mgu of the complementary pairs.

Proof:

Let $\alpha = P \vee \alpha'$ and $\beta = \neg P \vee \beta'$, hence $\gamma = \alpha' \vee \beta'$, where α' and β' are disjunctions of literals.

Suppose α and β are true in model M. We want to prove that γ is also true in M. To prove this, note that either P or $\neg P$ is false in M.

Assume P is false in M. Then α' must not be a unit clause, for otherwise α would be false in M. Therefore, α' must be true in M. Thus, the resolvent $\alpha' \vee \beta'$ is true in M.

Similarly, assume $\neg P$ is false in M. Then β' must not be a unit clause, for otherwise β would be false in M. Therefore, β' must be true in M. Thus, the resolvent $\alpha' \vee \beta'$ is true in M.

Either way, whenever $\{\alpha, \beta\}$ are true, $\alpha' \vee \beta'$ is true.

10.5 Incompleteness of Resolution Deduction

In worksheet 9 we said that resolution deduction is not complete, i.e. if you just start with some proper axioms in CFL and keep applying resolution as your only inference rule there are entailments that cannot be proved. (This motivated us to move to using resolution refutation, which we show in the next section *is* complete.)

The proof that resolution deduction is not complete is easily shown by means of a counter-example. Here are two counter-examples.

Example 1

$\{\forall X(\text{elephant}(X) \rightarrow \text{grey}(X)), \text{elephant}(\text{clyde})\} \models \text{elephant}(\text{clyde}) \vee \text{mouse}(\text{martie})$

If we convert the premisses to clausal form, we get $\{\neg\text{elephant}(X) \vee \text{grey}(X)$, $\text{elephant}(\text{clyde})\}$. There is clearly no resolution of these two clauses that will give us elephant(clyde) $\vee$ mouse(martie).

Example 2

$\{\forall X(\text{elephant}(X) \rightarrow \text{grey}(X)), \text{elephant}(\text{clyde})\} \models \text{mouse}(\text{martie}) \vee \neg\text{mouse}(\text{martie})$

The clausal form of the premisses is again $\{\neg\text{elephant}(X) \vee \text{grey}(X)$, elephant(clyde)$\}$. And, again, there is clearly no resolution of these two clauses that will give us mouse(martie) $\vee$ $\neg$mouse(martie).

However, if we were to use resolution refutation (i.e. negate the things we're trying to prove, convert them to clausal form and then use resolution to find a refutation tree), then a refutation would be found. (YOU SHOULD TRY THIS OUT TO CONFIRM IT!!!). Hence resolution refutation does better than resolution deduction. Which leads us into our final theorem...

10.6 Completeness of Resolution Refutation

Again I give the proof only for the propositional case. There is an awful lot of hard work needed when exploiting this result to give a proof for the more general case with variables.

First some terminology. The *number of literal occurrences* in a set of clauses is the sum of the number of literal occurrences in each clause. The *number of excess literals* in a set of clauses is the number of literal occurrences minus the number of clauses. Thus, the number of excess literals is an indication of the number of clauses in the set of clauses with more than one literal.

A *unit clause* is one with only one literal. A *nonunit* clause has more than one literal

Here's a statement of the theorem.

If a set Φ of clauses is unsatisfiable, then there is a resolution deduction of the empty clause from Φ.

Proof:

If the empty clause is in Φ, then there is a trivial resolution deduction of the empty clause from Φ.

We prove the case in which Φ does not contain the empty clause by induction on the number of excess literals in Φ.

If $n = 0$, then there are no nonunit clauses in Φ. (All are unit clauses.) So, if Φ is unsatisfiable, there must be at least one pair of complementary literals, and this pair can be resolved to yield the empty clause.

Suppose now that the theorem is true for all sets of clauses with fewer than n excess literals (there are up to $n-1$ nonunit clauses).

Since $n > 0$, and Φ does not contain the empty clause, there must be at least one nonunit clause, say α. We select a literal P from α and form a new clause $\alpha' = \alpha - \{P\}$. α' is stronger than α is, so the set $(\Phi - \{\alpha\}) \cup \{\alpha'\}$ also must be unsatisfiable.

But this set contains one fewer excess literals; so, by the induction hypothesis, there is a resolution deduction of the empty clause from this set.

Similarly, the set $(\Phi - \{\alpha\}) \cup \{\{P\}\}$ is unsatisfiable; so, by the induction hypothesis, there must be a deduction of the empty clause from this set as well. If α' is not used in the former refutation, then this refutation also works for Φ. Otherwise, we can construct a refutation for Φ as follows. First we add P back into α' and all its descendants in the refutation, so that the sequence is a refutation from Φ. If the empty clause is still a member of this sequence, then we are done. Otherwise, the addition of P to the empty clause must have created the singleton P. Now, we can form a deduction of the empty clause from Φ by appending the definition of the empty clause from $(\Phi - \{\alpha\}) \cup \{\{P\}\}$ onto the end of this modified deduction.

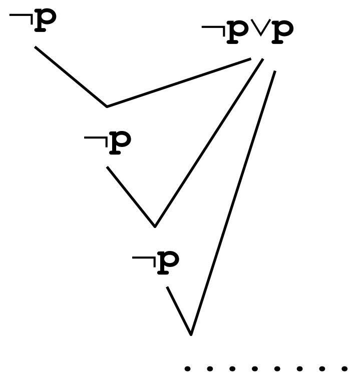
And that's it. (Remember this is the propositional case only.)

10.7 Decidability of Propositional Logic and Semi-Decidability of FOPC

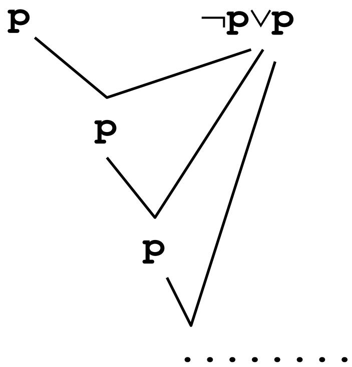
This is a highly informal presentation of these issues.

Consider showing that neither p nor ¬p follow from ¬p ∨ p.

To show ¬p ∨ p $\not\vdash$ p. We negate the goal to give ¬p.



To show $\neg p \vee p \nvdash \neg p$. We negate the goal to give p.



Both sequences of resolutions are infinite. But you can see it would be easy to add to a theorem-proving program a test to check whether the same resolvents are being produced time and time again (as is happening here), and to abandon such branches. If this means that all branches are abandoned, then the program can return ***no***. (Hence, in both of the above examples, the answer is ***no***.) This intuition shows why propositional logic is decidable: it will always return an answer.

However, such a check is not possible in general for FOPC: some problems will produce an infinite number of resolvents but detection of this will not be possible. (This is related to the Halting Problem: you cannot write a program that will tell you whether, for all other possible programs, those other programs will halt.)

(By the way, the above examples, also answer a common student question: if I can't prove a problem (can't find $\Box$) one way, can I try the other (i.e. if p won't prove, can I try $\neg$p)? The answer is: you *can* do this, but you can see that it won't always help.)