Worksheet 9

Resolution

Resolution is the inference rule that we will be using. The aims of this worksheet are that

- you should appreciate the power of the resolution rule (note how it 'encompasses' other rules and logical axiom schemata)
- learn how to use it in refutation proofs, both to return answers to yes-or-no questions and to return answers to fill-in-the-blank questions.

9.1 The Proof Theory of Clausal Form Logic

The proof theory that we present for CFL is based on the inference rule of *resolution*. Resolution is an inference rule that derives a clause from two clauses.

We will present the inference rule in stages.

9.1.1 Resolution without variables

Resolution without regard to variables is the simplest case.

- Let α, β and γ be clauses. Then γ is a resolvent of α and β iff

  - There is a non-empty set of positive literals $A \subseteq \alpha$ and a non-empty set of negative literals $B \subseteq \beta$ such that when we complement the literals in B, to give B', then each literal in A is equal to each in B'.
  - $\gamma = ((\alpha - A) \cup (\beta - B))$

More roughly, if p is a literal in one of the clauses and in the other clause there is the literal $\neg p$, these are a complementary pair. Resolution allows us to infer a new clause that consists of all the literals of both clauses without the complementary pair.

In a given step of resolution α and β are said to be the *parents*, the literals A are said to be the *selected literals* of α, and those in B are said to be the selected literals of β.

Example 1

1. p ∨ q
2. ¬p ∨ r
3. q ∨ r (from resolution of 1 and 2)

The rationale of this is clear. Since the original clauses are implicitly conjoined, both must be true. But if p is true then r must be true; if ¬p is true then q must be true. So we can deduce q ∨ r.

Example 2

Remember that clauses can be thought of as sets, so there can be no duplicates in the resolvent.

1. p ∨ q
2. ¬p ∨ q
3. q (from resolution of 1 and 2)

Note how the two occurrences of q have been 'merged'.

Example 3

Notice how Modus Ponens is a special case of resolution. In a particular instance of Modus Ponens we would have from p and p → q and we would deduce q. Converting the premisses to clausal form, we get:

1. p
2. ¬p ∨ q
3. q (from resolution of 1 and 2)

Example 4

Notice how Modus Tollens is also a special case of resolution. In a particular instance of Modus Tollens we would have ¬q and p → q and we would deduce ¬p. Converting the premisses to clausal form, we get:

1. ¬q
2. ¬p ∨ q
3. ¬p (from resolution of 1 and 2)

Example 5

Resolving two *unit clauses* leads to the *empty clause*, which is denoted □:

1. p
2. ¬p
3. □ (from resolution of 1 and 2)

The derivation of the empty clause means that the original clauses are contradictory.

Example 6

The following two clauses ((1) and (2)) have two resolvents ((3) and (4)):

1. p ∨ q
2. ¬p ∨ ¬q
3. q ∨ ¬q (from resolution of 1 and 2)
4. p ∨ ¬p (also from resolution of 1 and 2)

(Similarly p ∨ ¬q and ¬p ∨ q would have two resolvents.)

Note crucially about this example that the resolvent is *not* the empty clause: it is not the case that each of the literals in the set of positives from (1), i.e. {p, q} is equal to each of those in the set of complemented negatives from (2), i.e. {p, q} (e.g. p from the first set is not equal to q from the second set).

9.1.2 Resolution with variables

So far we've said nothing about variables. This will make use of unification.

• Let α, β and γ be clauses. Then γ is a resolvent of α and β iff

- There is a non-empty set of positive literals $A \subseteq \alpha$ and a non-empty set of negative literals $B \subseteq \beta$ such that, when we complement the literals in B, to give B', then each of the literals in A unifies with each in B' with some mgu θ
- $\gamma = ((\alpha - A) \cup (\beta - B))\theta$

More roughly, again we want clauses that have complementary pairs. But such a pair is now defined in terms of the atoms unifying. If they do unify, producing a mgu θ, then θ is applied to the resolvent.

Example 1

1. $\mathrm{p}(X) \vee \mathrm{q}(X, Y)$
2. $\neg\mathrm{p}(\mathrm{a}) \vee \mathrm{r}(\mathrm{b}, Z)$
3. $\mathrm{q}(\mathrm{a}, Y) \vee \mathrm{r}(\mathrm{b}, Z)$ (from resolution of 1 and 2 with mgu $\{X \mapsto \mathrm{a}\}$).

Example 2

Here's an example which shows that it is important to be able to resolve upon multiple literals (i.e. there is more than one complementary pair).[1]

1. $\mathrm{p}(U) \vee \mathrm{p}(V)$
2. $\neg\mathrm{p}(X) \vee \neg\mathrm{p}(Y)$
3. $\Box$ (resolution from 1 and 2 with mgu $\{X \mapsto U, Y \mapsto V\}$)

(Note that, in this example, each of the positive literals from (1), i.e. $\{\mathrm{p}(U), \mathrm{p}(V)\}$ unifies with each of the literals that are the complements of the negatives from (2), i.e. $\{\mathrm{p}(X), \mathrm{p}(Y)\}$. Hence, there is a resolvent: this is different from example 6 of the previous section.)

[1] Some textbooks don't allow this. In this case, they are presenting the inference rule of *binary* resolution, and they need to introduce an additional rule, called *factoring*, to compensate. We are presenting what is called *general resolution*, which is like binary resolution and factoring in a single step.

9.2 Resolution deduction

The simplest approach might now seem to be to start with the proper axioms, and by applying resolution to try to derive new clauses.

Example

All elephants are grey. Clyde is an elephant. Show that Clyde is grey.

The proper axioms in FOPC are:

$$\forall X(\text{elephant}(X) \rightarrow \text{grey}(X))$$
$$\text{elephant}(\text{clyde})$$

Converting these to clausal form and using them in a resolution deduction we get:

1.	$\neg\text{elephant}(X) \vee \text{grey}(X)$	(proper axiom)
2.	elephant(clyde)	(proper axiom)
3.	grey(clyde)	(from resolution of 1 and 2 with mgu $\{X \mapsto \text{clyde}\}$)

Thus we have shown that Clyde is grey.

Unfortunately, resolution deduction is not complete. There are entailments that cannot be proved. See Worksheet 10 for a demonstration of this. To make resolution deduction complete, we would need a number of logical axiom schemata. One of the logical axiom schemata is $\neg P \vee P$ for any atom P. We also need some logical axiom schemata to treat `true` and `false` properly. Once we have these we have a complete proof theory, but we have moved to a system which has the horror of logical axioms. Therefore, we have increased the amount of search that will need to be done. Fortunately, there is an alternative.

9.3 Resolution refutation

Resolution is very good at demonstrating inconsistency (unsatisfiability). If a set of clauses is unsatisfiable, then it is always possible by resolution alone to derive a contradiction from the clauses in the set, where in clausal form a contradiction takes the form of the empty clause, $\Box$.

Demonstrating that a set of clauses is unsatisfiable can also be used to demonstrate that a wff is an entailment of a set of wffs. For, example, suppose we want to show that $\alpha_1, \ldots, \alpha_n \models \beta$. Instead of using deduction (trying to derive β from $\alpha_1, \ldots, \alpha_n$), we negate β and try to show that $\alpha_1 \wedge \ldots \wedge \alpha_n \wedge \neg\beta$ is inconsistent, i.e. we show that this enables us to prove a contradiction (proof by contradiction). This way of doing proofs is known as *refutation* theorem proving. It involves denying the conclusion we are trying to reach, and applying inference rules from that denial until we reach some contradiction with the premisses. If we find a contradiction then the augmented set is inconsistent. This must have been caused by the addition of the negated goal[2] and so the goal itself (unnegated) must follow; if, on the other hand, we don't find a contradiction, the negated goal is consistent and so the goal itself doesn't follow. *Resolution refutation* theorem proving is refutation theorem proving where the inference rule we use is resolution.

Resolution is refutation complete, i.e. if the set of wffs is inconsistent then you will be able to derive □ from it using resolution alone (no other inference rules, no logical axioms).[3]

The Algorithm

```
1. Convert all the premisses to clausal form.
2. Negate the goal and convert to clausal form.
   Add it to the clauses obtained in (1).
3. Repeat until either a contradiction is found, no progress can be made,
          or a predetermined amount of effort has been expended:
    (1) Select two clauses. Call them the parent clauses.
    (2) Resolve them together. (There might be several resolvents).
    (3) If one of the resolvents is the empty clause, then
          a contradiction has been found.
        If none is, then
          add the resolvents to the set of clauses available to the procedure.
          Before doing so, standardise their variables apart from all those in
          existing clauses.
```

(Notice that the variables in the resolvents should be renamed so that they don't clash with those in other clauses. An example will be given in lectures to illustrate this.)

Of course we need a systematic way to choose parents. If we get this right, then *if there is a contradiction* we will find it (if there isn't, we might not stop). All in all this might take a long time. If a contradiction exists then eventually it will be found. Of course, if no contradiction exists, it is possible that the procedure will never terminate. This is due to the semi-decidability of FOPC (see TOC).

It is conventional to show the resolutions that lead to the derivation of □ as a *(resolution) refutation tree*.

[2]Assuming the premisses were consistent to start with; if they were not, then anything follows from them anyway.

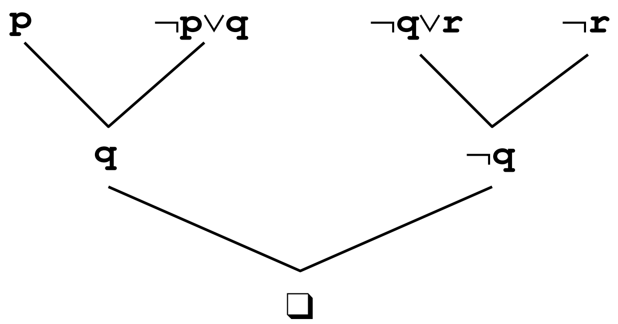
[3]Advanced note: if we want to use equality as a predicate, resolution is no longer refutation-complete. We must either axiomatise the properties of equality (reflexivity, symmetry, transitivity and the substitution of equals), change the way resolution works, or add another inference rule. If we do not do one of these, then from p(a) and a = b, we would not be able to infer p(b).

9.3.1 Some Examples

We can use resolution refutation to answer questions and solve problems.

9.3.2 A Simple Example

Show that clauses p, ¬p ∨ q, ¬q ∨ r and ¬r are inconsistent.



The empty clause has been derived, hence the axioms are inconsistent. Of course, I've done my usual trick of just giving the answer as if it's obvious. In practice, our dumb machine needs to *search* for the solution. We discuss this in a future worksheet.

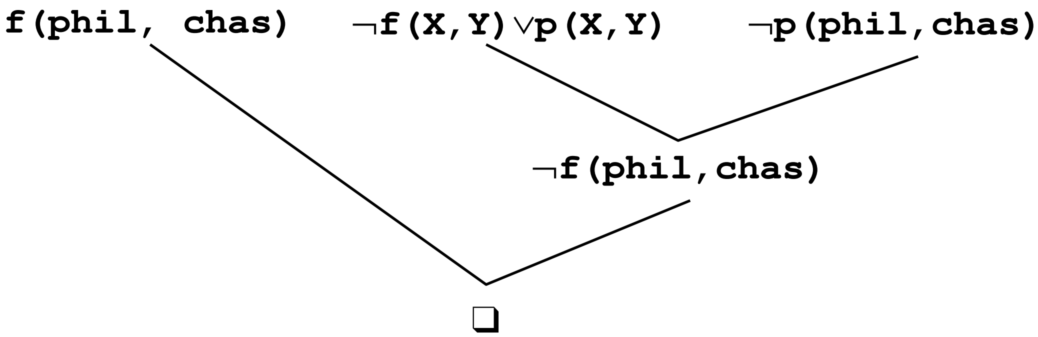
9.3.3 Answering Yes-No Questions

Example 1

Phil is the father of Chas. Fathers are parents. Is Phil a parent of Chas?

English	FOPC	CFL
1. Phil is the father of Chas	f(phil, chas)	f(phil, chas)
2. Fathers are parents	$\forall X \forall Y(\mathrm{f}(X, Y) \rightarrow \mathrm{p}(X, Y))$	$\neg \mathrm{f}(X, Y) \vee \mathrm{p}(X, Y)$

We negate the goal to give: It is not the case that Phil is a parent of Chas. In FOPC this is ¬p(phil, chas), which in CFL happens also to be ¬p(phil, chas).

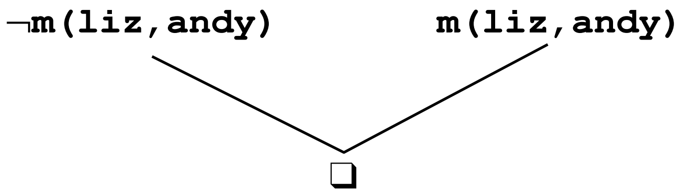


Since □ has been derived, the answer is "yes".

Example 2

Here's another example of a yes-no question. Suppose I know nothing about Liz or Andy and I want to prove that Liz is the mother of Andy or she is not the mother of Andy.

I have no proper axioms, just a goal. I negate the goal to give: It's not the case that (Liz is the mother of Andy or Liz is not the mother of Andy). In FOPC this is: ¬(m(liz, andy) ∨ ¬m(liz, andy)). In CFL we get two clauses: ¬m(liz, andy) and m(liz, andy).[4]



The answer is "yes": Liz either is or isn't the mother of Andy.

This example might give you some clue as to why resolution refutation is complete. Just by starting with the negated goal, we get somewhere. If we were doing resolution deduction instead of resolution refutation we would get nowhere with this example unless we had some logical axiom schemata to start us off.

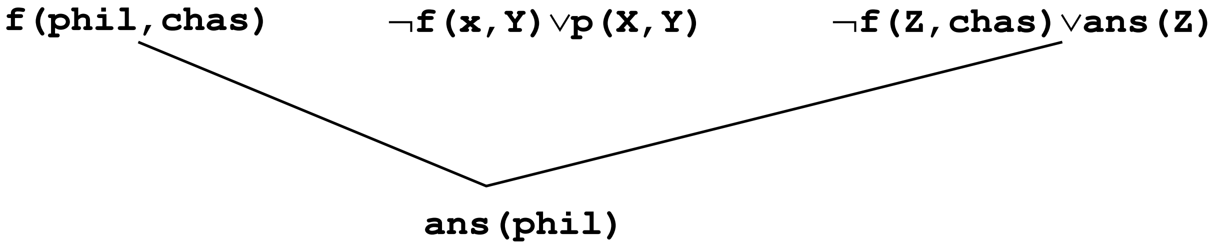
9.3.4 Fill-in-the-blank question

Here we look at questions where we do not just want a yes or no answer. Rather we want some term(s) as the answer(s).

[4]This is one of those examples where, if you negate the CFL rather than negating the FOPC and then converting to CFL, you'd be completely wrong.

Example 1

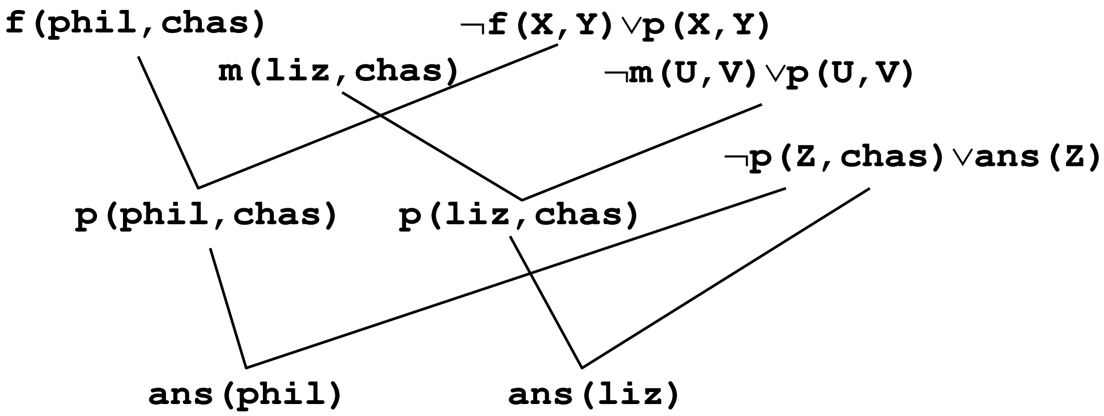
For our earlier axioms, let us ask: who is Chas' father? In FOPC this goal is $\exists Z \mathrm{f}(Z, \text{chas})$. I negate this to get $\neg\exists Z \mathrm{f}(Z, \text{chas})$ and convert to clausal form to get $\neg\mathrm{f}(Z, \text{chas})$. But there is now an excellent 'trick' that can be used. We add $\mathrm{ans}(Z)$ to this negated clausal form goal to give $\neg\mathrm{f}(Z, \text{chas}) \vee \mathrm{ans}(Z)$. (You can see that this is the same as $\forall Z(\mathrm{f}(Z, \text{chas}) \rightarrow \mathrm{ans}(Z)))$. Now we prove as usual, but instead of looking for the empty clause, we look for a clause containing only uses of the predicate ans.



The 'blank' has been filled in with Phil.

Example 2

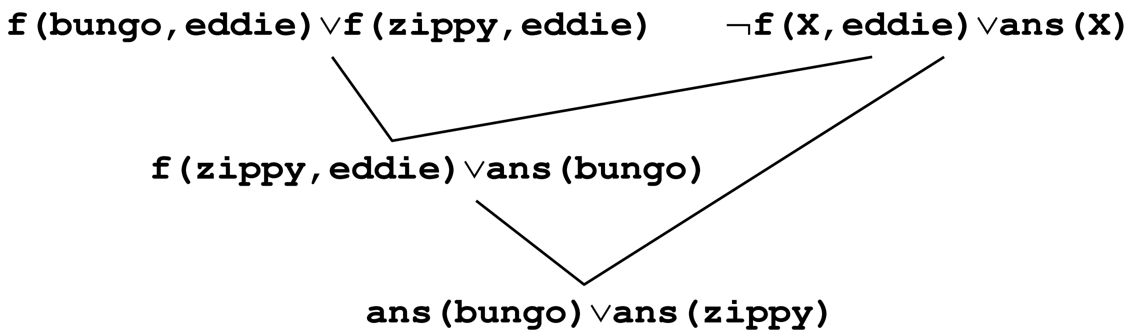
Here's another example. For the same axioms plus the axioms that Liz is Chas' mother and mothers are parents, who are Chas' parents? The new axioms are $\mathrm{m}(\text{liz}, \text{chas})$ and $\neg\mathrm{m}(U, V) \vee \mathrm{p}(U, V)$. The goal is $\neg\mathrm{p}(Z, \text{chas}) \vee \mathrm{ans}(Z)$.



The blank has been filled in once by Phil and again by Liz. So here there are two answers. Fortunately, the search space for this example is not infinite and is in fact very small and so we were able to search it exhaustively. Unfortunately, since these nice properties need not hold, there is no general way of knowing whether we have all the answers. We can continue to search. But due to the semi-decidability of FOPC and CFL, we can never know in general whether we have found all the possible answers.

Example 3

Here's a more tricky example. Bungo or Zippy is Eddie's father. Who is Eddie's father? The proper axiom is f(bungo, eddie) $\vee$ f(zippy, eddie) and this is already in clausal form. The goal in FOPC is $\exists X$f(X, eddie); we negate this to get $\neg\exists X$f(X, eddie); in clausal form this is f(X, eddie).



The answer remains a disjunction which is correct: Bungo or Zippy is Eddie's father.

9.4 Exercises

Exercise 9.1 *For each of the following wffs (which happen to be instances of logical axiom schemata), negate them, convert them to clausal form and derive $\Box$ using resolution refutation (thus proving that the instances are tautologies).*

1. $p \to (q \to p)$
2. $(p \to (q \to r)) \to ((p \to q) \to (p \to r))$
3. $(\forall X\,(p(X) \to q(X))) \to ((\forall X\,p(X)) \to (\forall X\,q(X)))$

In the following exercises, you will need to convert the English to FOPC, negate goals, convert FOPC into CFL and then use resolution refutation. The answers come in two parts: a sheet of FOPC and then a sheet of CFL and refutation trees. I suggest that you check your FOPC against the FOPC answer sheet before going on to goal negation, conversion to CFL and resolution refutation.

Exercise 9.2 *Chemical Synthesis. Suppose we can perform the following chemical reactions:*

$$M_gO + H_2 \rightarrow M_g + H_2O$$
$$C + O_2 \rightarrow CO_2$$
$$CO_2 + H_2O \rightarrow H_2CO_3$$

Suppose we have some quantities of M_gO, H_2, O_2 *and* C. *Show that we can make* H_2CO_3.
(Hint: This is 'easy' by virtue of the fact that it can be solved using propositional logic, i.e. to translate the 7 (yes, seven) premisses and the goal into logic does not require the use of quantifiers and variables.)

Exercise 9.3 *Whoever can read is literate. Dolphins are not literate. Some dolphins are intelligent. Show that some who are intelligent cannot read.*

Use the following predicates: $r(X)$ *for* X *can read,* $l(X)$ *for* X *is literate,* $d(X)$ *for* X *is a dolphin, and* $i(X)$ *for* X *is intelligent.*

Exercise 9.4 *All jokes are meant to amuse. No act of Parliament is a joke. Can you answer: is it true that every act of Parliament is not meant to amuse? If you can answer it, give the answer; if not, explain.*

Use the following predicates: $aop(X)$ *for* X *is an act of Parliament,* $mta(X)$ *for* X *is meant to amuse, and* $j(X)$ *for* X *is a joke.*

Exercise 9.5 *Any robot who can solve a logic problem is intelligent. All the Japanese robots can solve every logic problem. There's at least one logic problem. Arnie is a Japanese robot. Who is intelligent?*

Use the following predicates and constants: $r(X)$ *for* X *is a robot,* $s(X,Y)$ *for* X *can solve* Y, $lp(X)$ *for* X *is a logic problem,* $i(X)$ *for* X *is intelligent,* $j(X)$ *for* X *is Japanese, and* a *for Arnie.*

Exercise 9.6 *Every tabloid is owned by a press baron. It is not the case that every newspaper is owned by a press baron. All broadsheets are newspapers. All tabloids are newspapers. Newspapers are either broadsheets or tabloids but not both. Show that there are broadsheets that are not owned by any of the press barons.*

Use the following predicates: $n(X)$ *for* X *is a newspaper,* $t(X)$ *for* X *is tabloid,* $b(X)$ *for* X *is broadsheet,* $p(X)$ *for* X *is a press baron, and* $o(X,Y)$ *for* X *owns* Y.

Exercise 9.7 *The police computer recorded that Mr. Smallfry had not paid his parking fine. When he did pay it, the computer recorded the fact, but due to poor program design, did not wipe the statement that he had not. Show how the computer concluded that the Prime Minister was a spy.*

Use the following predicates and constants: paid(X) for X has paid his parking fine, spy(X) for X was a spy, smf for Mr. Smallfry, and pm for the Prime Minister.

Exercise 9.8 *Victor has been murdered, and Arthur, Bertram and Carleton are suspects. Arthur says he did not do it. He says that Bertram was the victim's friend but that Carleton hated the victim. Bertram says he was out of town the day of the murder, and besides he didn't even know the guy. Carleton says he is innocent and he saw Arthur and Bertram with the victim just before the murder. Assuming that everyone (except possibly for the murderer) is telling the truth, who committed the crime?*

Use the following predicates and constants: i(X) for X is innocent, friend(X,Y) for X was Y's friend, in-town(X) for X was in town, know(X,Y) for X knew Y, saw-with(X,Y) for X was seen with Y, a for Arthur, b for Bertram, c for Carleton, and v for Victor.

Hints: You will need to add some axioms to capture world knowledge not given in the question. In particular, you will need to add the fact that only one person is guilty, i.e. $(i(a) \vee i(b)) \wedge (i(a) \vee i(c)) \wedge (i(b) \vee i(c))$. *Anyone who was with Victor must have been in town. And friends know each other.*

You also want to make each person's statements conditional on their innocence.

Acknowledgements: These exercises are based on exercises and example to be found in Chang & Lee, Nilsson, and Genesereth & Nilsson.