Resolution Refutation

1 Resolution deduction

Now we have explained our only inference rule (resolution), we can start to use it to derive conclusions from sets of wffs.

Example

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

(1) English: All elephants are grey

FOPL:

CFL:

(2) English: Clyde is an elephant

FOPL:

CFL:

To show

(3) English: Clyde is grey

FOPL:

CFL

More complicated examples might involve using the resolution inference rule several times.

Unfortunately, resolution deduction is not complete. In other words, there are logical consequences that cannot be derived using resolution. There are cases where $\Phi \models W$ but using only resolution $\Phi \not\vdash W$.

Example

If you know that 'anything follows from a contradiction', then you won't be surprised by the following logical consequence:

$$\{elephant(clyde), \neg elephant(clyde)\} \models mouse(martie)$$

But, resolution deduction does not derive $mouse(martie)$:

It turns out that, if we want to make resolution deduction complete, we have to add some *logical axiom schemata* to our proof theory. This is undesirable: as we saw before, they make the search space infinite.

Fortunately, there is an alternative.

2 Resolution refutation

Instead of using resolution to show

$$\{W_1, W_2, \ldots, W_n\} \vdash W$$

we can use resolution to show that

$$W_1 \wedge W_2 \wedge \ldots \wedge W_n \wedge \neg W$$

is *inconsistent*.

In other words, we can do a *proof by contradiction*. Proof by contradiction also goes by the name of *refutation proof*.

In the case of clausal form logic, we will try to derive $\Box$ from

$$\{W_1, W_2, \ldots, W_n, \neg W\}$$

It turns out (although the proof of this is quite involved) that resolution refutation on clausal form is complete.

3 Resolution Refutation on CFL: Summary

Here, in summary, is what we have to do to show that some query W follows from some premisses Φ using resolution refutation.

1. Convert all the *premisses* to clausal form.
2. Negate the query ***and then*** convert to clausal form.
3. Repeat until either a contradiction is found, no progress can be made or a predetermined amount of effort has been expended
   - Select two clauses (the parents)
   - Resolve them together

- If one of the resolvents is the empty clause, then a contradiction has been found
  If not, standardise the variables apart in these new clauses and then add them to the set of clauses available to the procedure.

I want you to show these proofs in the form of *refutation trees*.

The following examples/exercises will be completed during the lecture.

Example 1

1. Every elephant is grey.
2. Clyde is an elephant.
3. Is Clyde grey?

Example 2

1. If one is in Paris, then one is not in Moscow.
2. Flopsy is in Paris.
3. Is Flopsy in Moscow?

Example 3

1. Everyone who saves money earns interest.
2. Show if there is no interest earned, then nobody saves money.

Use $s(x)$ for x saves money and $e(x)$ for x earns interest.

Example 4

1. Flopsy is Clyde's friend.
2. Who is Clyde's friend?

This example is different from the previous ones. In the previous examples, the query was a *yes/no-question* (Is Clyde grey? Is Flopsy in Moscow?) But here we have what is called a *wh-question* (who, what, why, when, where, how). The answer is not just yes or no. We need to find some term, and this will be the answer.

There is an excellent 'trick' we can use. We add, e.g., $\mathit{ans}(x)$ to the negated clausal form query. (Use an appropriate variable in place of x.) We then use resolution refutation as normal, but, instead of searching for the empty clause, we search for a unit clause whose predicate is *ans*. The argument to that predicate symbol is the answer to the question.

Example 5

1. For all x, y and z, if x is the father of y and z is the father of x, then z is the grandfather of y.
2. Everyone has a father.
3. For some x, who is the grandfather of x?

Exercises

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 valid wffs).

   (a) $p \Rightarrow (q \Rightarrow p)$

   (b) $(p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r))$

   (c) $(\forall x(p(x) \Rightarrow q(x))) \Rightarrow ((\forall x p(x)) \Rightarrow (\forall x q(x)))$

2. 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: $\mathit{paid}(x)$ for x has paid his parking fine, $\mathit{spy}(x)$ for x was a spy, *smf* for Mr. Smallfry, and *pm* for the Prime Minister.

3. (Past exam question) This question uses the following 'key' for the unary predicate symbols *lecturer*, *student* and *csdept* and the binary predicate symbol *supervises*:

$\mathit{lecturer}(x)$	:	x is a lecturer
$\mathit{student}(x)$	:	x is a student
$\mathit{incsdept}(x)$	:	x is a member of the Computer Science Department
$\mathit{supervises}(x, y)$	:	x supervises y

   (x, y, z and subscripted versions of these will be used as variables.)

   (a) Indicate, by writing *Correct* or *Incorrect*, whether the following wffs of FOPL are correct representations of the corresponding English sentences. Where you think they are *incorrect*, briefly explain why.

      i. Every Computer Science student is supervised by a Computer Science lecturer.
      $$\forall x((\mathit{student}(x) \wedge \mathit{incsdept}(x)) \Rightarrow \exists y(\mathit{lecturer}(y) \wedge \mathit{incsdept}(y) \wedge \mathit{supervises}(y, x)))$$
      ii. Computer Science students do not supervise Computer Science lecturers.
      $$\forall x((\mathit{student}(x) \wedge \mathit{incsdept}(x)) \Rightarrow \neg\forall y((\mathit{lecturer}(y) \wedge \mathit{incsdept}(y)) \Rightarrow \mathit{supervises}(x, y)))$$
      iii. If there's at least one Computer Science student then there's at least one Computer Science lecturer.
      $$\exists x \exists y((\mathit{student}(x) \wedge \mathit{incsdept}(x)) \Rightarrow (\mathit{lecturer}(y) \wedge \mathit{incsdept}(y)))$$

   (b) Convert the following wff of FOPL into *Clausal Form Logic*. Show your working.
   $$(\exists x(\mathit{lecturer}(x) \wedge \mathit{incsdept}(x))) \Rightarrow (\exists y(\mathit{student}(y) \wedge \mathit{incsdept}(y)))$$

   (c) You are given the following **four** *clauses*:

   All members of the Computer Science Department are either lecturers or students.
   $$\neg\mathit{incsdept}(x_1) \vee \mathit{lecturer}(x_1) \vee \mathit{student}(x_1)$$
   Computer Science students have at least one Computer Science lecturer.
   $$\neg\mathit{student}(x_2) \vee \neg\mathit{incsdept}(x_2) \vee \mathit{lecturer}(f(x_2))$$
   $$\neg\mathit{student}(x_3) \vee \neg\mathit{incsdept}(x_3) \vee \mathit{incsdept}(f(x_3))$$
   There is at least one member of the Computer Science department.
   $$\mathit{incsdept}(\mathit{sk})$$
   (f is a Skolem function and *sk* is a Skolem constant.)

   From these clauses, use *resolution refutation* theorem-proving to show that there is at least one Computer Science lecturer, i.e. in FOPL:
   $$\exists z(\mathit{lecturer}(z) \wedge \mathit{incsdept}(z))$$
   Show your working, presenting your proof in the form of a *refutation tree*.

4. (Past exam question) This question uses the following 'key' for the unary predicate symbols *irish* and *scot*, the binary predicate symbols *in* and *likes*, the unary function symbol *kitchenOf*, and the constant symbol *b*:

$\mathit{irish}(x)$	:	x is Irish
$\mathit{scot}(x)$	:	x is Scottish
$\mathit{in}(x, y)$	:	x is in y
$\mathit{likes}(x, y)$	:	x likes y
$\mathit{kitchenOf}(x)$	:	the kitchen of x
b	:	the Big Brother House

(x, y, z and subscripted versions of these will be used as variables.)

(a) Give a natural English paraphrase of the following wff of FOPL:

$$\forall x \forall y((\mathit{irish}(x) \land \mathit{in}(x, b) \land \mathit{scot}(y) \land \mathit{in}(y, b)) \Rightarrow \mathit{likes}(x, y))$$

(b) Translate the following sentence of English into FOPL:

There are Irish people who are outside the Big Brother House who do not like any of the Irish people who are inside the Big Brother House.

(c) Determine whether the members of the following pairs of atoms unify with each other. If they do, give their *most general unifier* (mgu); if they do not, give a brief explanation.

i. $\mathit{in}(x, x)$ and $\mathit{in}(\mathit{kitchenOf}(y), y)$
ii. $\mathit{in}(\mathit{kitchenOf}(x), x)$ and $\mathit{in}(b, \mathit{kitchenOf}(b))$
iii. $\mathit{in}(x, \mathit{kitchenOf}(x))$ and $\mathit{in}(y, \mathit{kitchenOf}(\mathit{kitchenOf}(b)))$

(d) Convert the following wff of FOPL into *Clausal Form Logic*. Show your working.

$$(\forall x((\mathit{irish}(x) \land \mathit{in}(x, b)) \Rightarrow \mathit{likes}(x, x))) \Rightarrow \exists y(\mathit{scot}(y) \land \mathit{likes}(y, y))$$

(e) You are given the following five *clauses*:

Everyone in the Big Brother House is either Irish or Scottish.

$$\neg \mathit{in}(x_1, b) \lor \mathit{irish}(x_1) \lor \mathit{scot}(x_1)$$

If a Scot is in the House, then there is some Irish person in the House whom the Scot likes.

$$\neg \mathit{scot}(x_2) \lor \neg \mathit{in}(x_2, b) \lor \mathit{irish}(f(x_2))$$
$$\neg \mathit{scot}(x_3) \lor \neg \mathit{in}(x_3, b) \lor \mathit{in}(f(x_3), b)$$
$$\neg \mathit{scot}(x_4) \lor \neg \mathit{in}(x_4, b) \lor \mathit{likes}(x_4, f(x_4))$$

Someone is in the Big Brother House.

$$\mathit{in}(\mathit{sk}, b)$$

(f is a Skolem function and *sk* is a Skolem constant.)

From these clauses, use *resolution refutation* theorem-proving to show that an Irish person is in the House, i.e. in FOPL:

$$\exists y(\mathit{irish}(y) \land \mathit{in}(y, b))$$

Show your working, presenting your proof in the form of a *refutation tree*.