Resolution

The proof theory that we present for Clausal Form Logic is based on only one inference rule, which is called resolution. Resolution is an inference rule that derives a new clause from two other clauses.

We present the rule in stages: first we ignore arguments to predicates (i.e. we deal with propositional logic), then we bring in the case where predicate have arguments.

1 Resolution for Propositions

So, to begin with there are no variables or function symbols in our logic. Informally, the resolution inference rule says that if one clause contains a literal P, and another clause contains a literal $\neg P$, then we can derive a new clause that contains all the literals of both clauses excluding the literals P and $\neg P$.

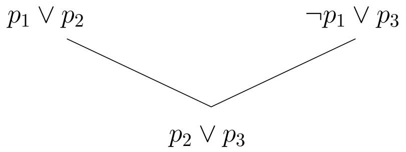
Example 1

1. $p_1 \vee p_2$
2. $\neg p_1 \vee p_3$
3. $p_2 \vee p_3$ (from resolution of 1 and 2)

In this example,

- (1) and (2) are the *parents*;
- p_1 and $\neg p_1$ are the *selected literals*;
- p_1 and $\neg p_1$ are a *complementary pair*; and
- (3) is the *resolvent*.

It will be more common for us to show a use of the resolution inference rule in a tree diagram:



Resolution is a *sound* rule of inference. In other words, if we can derive a clause from two other clauses using resolution, then that clause is a logical consequence of those other clauses. We won't prove this, but we will demonstrate it.

We know that, using resolution,

$$\{p_1 \vee p_2, \neg p_1 \vee p_3\} \vdash p_2 \vee p_3$$

So let's show that

$$\{p_1 \vee p_2, \neg p_1 \vee p_3\} \models p_2 \vee p_3$$

We can do this using a truth-table:

p_1	p_2	p_3	$p_1 \vee p_2$	$\neg p_1 \vee p_3$	$p_2 \vee p_3$	
T	T	T	T	T	T	*
T	T	F	T	F	T	
T	F	T	T	T	T	*
T	F	F	T	F	F	
F	T	T	T	T	T	*
F	T	F	T	T	T	*
F	F	T	F	T	T	
F	F	F	F	T	F	

We see that, in all interpretations in which $p_1 \vee p_2$ and $\neg p_1 \vee p_3$ are true (rows 1, 3, 5 and 6), the conclusion is also true.

Here's another informal attempt to show that resolution produces sensible results: if p_1 is true, then for the other clause to be true, p_3 must be true; but if $\neg p_1$ is true, then for the first clause to be true, p_2 must be true. So we've shown that either p_3 is true or p_2 is true: $p_2 \vee p_3$.

We'll look at some more examples of resolution.

Example 2

Remember that clauses can be thought of as sets, so duplicates can be removed from resolvents:

1. $p \vee q$
2. $\neg p \vee q$
3.

Example 3

The inference rule of Modus Ponens (or $\Rightarrow$-ELIM), i.e. $\frac{W_1, W_1 \Rightarrow W_2}{W_2}$, is a special case of resolution. Here's an example showing a use of Modus Ponens. Alongside it, we'll convert to clausal form and use resolution:

1.	p	1.
2.	$p \Rightarrow q$	2.
3.	q	3.

Example 4

There is another inference rule which is called Modus Tollens, $\frac{\neg W_2, W_1 \Rightarrow W_2}{\neg W_1}$. This too is a special case of resolution:

1.	$\neg q$	1.
2.	$p \Rightarrow q$	2.
3.	$\neg p$	3.

Example 5

Recall that, if a clause contains only one literal, it is called a *unit clause*. If we resolve two unit clauses, we get the empty clause:

1. p
2. $\neg p$
3.

Some examples, like the next two, require real care.

Example 6

1. $p \vee q$
2. $\neg p \vee \neg q$
3. (from resolution of 1 and 2)
4. (from resolution of 1 and 2)

Example 7

1. $p \vee \neg q$
2. $\neg p \vee q$
3. (from resolution of 1 and 2)
4. (from resolution of 1 and 2)

Despite what you might think, the empty clause is *not* a resolvent in either example 6 or 7. In each example, there are *two* resolvents, both simplifying to true.

Finally, try these next two.

Example 8

1. $p \vee \neg q \vee r$
2. $p \vee \neg r \vee q \vee s$
3.
4.

Example 9

1. $p \vee p$
2. $\neg p \vee \neg p$
3.

2 Resolution for Predicates

In the previous section, we used propositional logic (i.e. predicate symbols had no arguments): we were just using p, q, p_1, p_2, etc. When we introduce arguments to predicates, the resolution inference rule becomes a little more complicated. It now needs to use *unification*.

Recall that our unification algorithm allows us to see whether two atoms are structurally identical, although one may have variables in places the other does not. And, if they do match, the algorithm gives us a substitution (a set of bindings) called the mgu (most general unifier).

So resolution now states: Two parent clauses can be resolved if there is some pair of literals in the parents that is a complementary pair when some *mgu* is applied to them. The mgu must also be applied to the resolvent.

Example 1

1. $p(x) \vee q(x, y)$
2. $\neg p(a) \vee r(b, z)$
3.

Example 2

1. $p(x) \vee q(x, y)$
2. $\neg p(f(z)) \vee \neg r(f(a), b) \vee s(z)$
3.

Example 3

1. $p(x, a) \vee q(x, y)$
2. $\neg p(f(z), z) \vee r(z)$
3.

3 Resolution deduction

So now we have explained our only inference rule, 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:

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

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.

4 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.