Resolution as Search

1 Proof systems

The proof system resolution:

- All formulae are clauses, i.e. disjunctions of literals
- One rule of inference, the resolution rule

We may assume that our knowledge base KB is a set of clauses as follows:

- Think of KB as a conjunction of all its formulae
- Find an equivalent formula in conjunctive normal form
- Split the c.n.f conjunction into a set of clauses, so we have moved back to the KB being a set of formula

2 The resolution inference algorithm

Suppose that we want to decide whether $KB \models \phi$

- Convert $\neg\phi$ to cnf
- Add the resulting clauses to (the set of clauses) KB
- Iteratively apply the resolution rule to produce new clauses which are added to the set of clauses if they are not already present
- This iterative process continues until either
  - There are no new clauses to be added - in which case our algorithm answers that KB **does not** entail ϕ
  - Two clauses resolve to yield the empty clause, so these clauses must be X and $\neg X$, for some boolean variable X, in which case our algorithm answers that KB **does** entail ϕ
- We can also factor, that is, replace $\alpha \vee \alpha$ with α (α is some literal)

The resolution inference algorithm is both sound and complete, so in the worst case it is exponential

2.1 Example

Recall our wumpus world KB from earlier

$$\{\neg P_{1,1}, B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1}), B_{1,2} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{1,3}), \neg B_{1,1}, B_{1,2}\}$$

Rewrite these formula so that KB is in c.n.f and take the clauses

$$\begin{aligned}&\{\neg P_{1,1}, -B_{1,1} \vee P_{1,2} \vee P_{2,1}, \neg P_{1,2} \vee B_{1,1}, \neg P_{2,1} \vee B_{1,1}\ ,\\ &\neg B_{1,2} \vee P_{1,1} \vee P_{2,2} \vee P_{1,3}, \neg P_{1,1} \vee B_{1,2}, \neg P_{2,2} \vee B_{1,2},\\ &\neg P_{1,3} \vee B_{1,2}, \neg B_{1,1}, B_{1,2}\}\end{aligned}$$

Suppose that the agent returns from room [1,2] to room [1,1] and then moves to room [2,1]
As a consequence, suppose the following formulae are added to KB

$$B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1}) \text{ and } \neg B_{2,1}$$

Which converted to c.n.f is:

$$B_{2,1} \vee P_{1,1} \vee P_{2,2} \vee P_{3,1}, \neg P_{1,1} \vee B_{2,1}, \neg P_{2,2} \vee B_{2,1}, \neg P_{3,1} \vee B_{2,1}, \neg B_{2,1}$$

Suppose we want to know whether $KB \models P_{1,3}$ (which is finding if there is a pit in $P_{1,3}$)
Add $\neg P_{1,3}$ to our set of clauses and apply Resolution

We get the empty set, so there is definitely a pit in room [1,3]

This is a bit of a cheat as the resolution is just being applied, where in reality it is difficult for an algorithm to do this. It is especially difficult as the set of clauses could get very large-

3 Realising Resolution via ”global path-based” search

- In our first illustration we ”magically” found the derivation of the empty clause
- In general, what we’ll need to do is apply a search algorithm in order to try and ”find” the empty clause

State	Set of clauses
Initial state	The clauses of $KB \vee \neg\phi$
Actions	Resolve and Factor
Goal state	Any set of clauses containing the empty clause
State Transition	(Σ, action, Σ') such that • **Resolve**: the target of clauses Σ' is the result of applying the resolution rule of inference (once) to the source set of clauses Σ • **Factor**: we factor a clause Σ to obtain the set of clauses Σ'
Step cost	1

The path from the initial state to a goal state is a ”proof” with an optimal path being a ”shortest proof”

4 Realising Resolution via ”local state-based” search

Here is a ”local state-based” search formulation of Resolution

State	Set of clauses
Initial state	The clauses of $KB \vee \neg\phi$
State transition	(Σ, Σ') is such that the target set of clauses Σ' is • The result of applying the resolution rule of inference (once) to the source set of clauses Σ, or • The result of factoring a clause Σ
Objective function	f such that $f(\Sigma)$ is • The number of clauses in Σ if the empty clause $\varnothing$ is in Σ • ∞ otherwise, with ∞ a number bigger than the total number of clauses • Or a better objective function might be the size of the smallest clause in Σ