COMP3702/7702 Artificial Intelligence
Semester 2, 2020
Tutorial 7 - Sample Solutions

Exercise 7.1

$$(P \wedge \neg P) \vDash R$$

$$[axioms] \vDash [conclusion]$$

This might seem strange. The conclusion has nothing to do with the axioms! Intuitively, this hints at the answer – but let's prove it.

The first step of resolution refutation is to convert all sentences in our axioms to conjunctive normal form (CNF), i.e. "ANDs of ORs". We have just one sentence in this case, which is straightforward to convert:

$$P \wedge \neg P = (F \vee P) \wedge (\neg P \vee F)$$

Since OR-ing something with false is itself (why we choose this way of writing it becomes apparent later). Next we negate the conclusion, giving

$$\neg R$$

Finally, we apply the resolution rule iteratively. This is

$$\begin{array}{r} A \vee B \\ \neg B \vee C \\ --- \\ A \vee C \end{array}$$

Let's write out the clauses appropriately:

$$\begin{array}{rl} F \vee P & (1) \\ \neg P \vee F & (2) \\ ---- & \\ \neg R & (3) \end{array}$$

Applying the resolution rule on (1) and (2) yields:

$$4. F \vee F = F$$

We have derived false, a contradiction. This means the conclusion follows from the axioms. Note that this is an odd case where it doesn't matter what the conclusion is. We can see this just by deriving $P \wedge \neg P = F$ directly from the original axiom. False entails anything, including R in this case, so it is valid.

Exercise 7.2

Conjunctive normal form (CNF) of the problem is:

$$(B \lor C) \land (B \lor R) \land (\neg C \lor B \lor H) \land (\neg H \lor F) \land (\neg B \lor \neg F) \land (\neg R \lor F) \land (R \lor H \lor F)$$

We can see if this is satisfiable using DPLL. The search tree is not included in this solution, but you may find it a useful learning experience to draw it as we step through the algorithm.

1. Count each occurrence of each variable: B: 4, C: 2, H: 3, R: 3, F: 4.
   Pick B, and set B to true. We have:

   $$(\neg H \lor F) \land (\neg F) \land (\neg R \lor F) \land (R \lor H \lor F)$$

2. Count occurrences again: F: 4, H: 2, R: 2.
   - Set F to true. We have false since $(\neg F)$ is false.
   - Set F to false:

   $$(\neg H) \land (\neg R) \land (R \lor H)$$

3. Count occurrences again: H: 2 R: 2.
   - Set H to true. We have false since $(\neg H)$ is false.
   - Set H to false. We have $(\neg R) \land (R)$ which is false.

   There is no point exploring this branch any further.

4. Now backtrack to B.
   Set B to false. We have:

   $$(C) \land (R) \land (\neg C \lor H) \land (\neg H \lor F) \land (\neg R \lor F) \land (R \lor H \lor F)$$

5. Count occurrences again: C: 2, H: 3, F: 3, R: 3
   Set H to true:

   $$(C) \land (R) \land (F) \land (\neg R \lor F)$$

6. Count occurrences again: C: 1, F: 2, R: 2
   Set F to true:
7. There is one sentence remaining:

   $$(C) \land (R)$$

   Set C to true: R
8. Set R to true, we get true so stop here. This sentence is satisfiable.

To figure out whether it is satisfiable with only 3 variables being true, you will need to traverse the DPLL search tree.

Exercise 7.3

An AND-OR tree is used to find a guaranteed solution in a non-deterministic search problem. In this case, we know that an agent can fully control where it goes but not how long it takes since traffic is uncertain. So each edge under an *or* node can represent direction which is deterministic, and each edge under an *and* node can represent time it would take which are discretised into 15-minutes intervals.

If the agent cannot find a solution, it only means that there is no guaranteed solution. The agent could still move to the goal point in time if the traffic happens to be good.