Introduction to AI: Solutions to Logic Exercises week 09

April 6th, 2021

Ex. 1 Consider the following knowledge base $KB = \{\neg p \to q, q \to p, p \to r \wedge s\}$. Decide if the formula $p \wedge r \wedge s$ follows from KB. Try using each of the following methods: truth tables, Modus Ponens only, resolution.

Answer: **Truth tables** approach draw a truth table for four variables p, q, r, s, compute the truth values under all assignments fot the formulas in KB and for the goal formula. Check if in all rows in which all the formulas in KB are true, the goal formula is also true. If that is so, the goal formula follows from KB, otherwise it does not. **Modus Ponens** method is not applicable since the goal formula does not appear as a consequent of an implication formula in any of the elements of KB. **Resolution** method is used in the following way. Proof by refutation:

1. $A \vee B$ premise
2. $\neg B \vee A$ premise
3. $\neg A \vee C$ premise
4. $\neg A \vee D$ premise
5. $\neg A \vee \neg C \vee \neg D$ negated thesis
6. A resolution 1, 2
7. C resolution 3, 6
8. D resolution 4, 6
9. $\neg C \vee \neg D$ resolution 5, 6
10. $\neg D$ resolution 7, 9
11. $\{\}$ resolution 8, 10

Ex. 2 Come up with a propositional logic formula which uses only implication and negation, and which can not be represented as a Horn clause.

A possible answer: $(\neg p \to q) \to \neg r$.

Ex. 3 (7.11 and 7.18 a from the handbook) Any propositional logic sentence is logically equivalent to the assertion that:

- each possible world in which it would be false is not the case;
- some possible world in which it would be true is in fact the case.

From those prove that any sentence can be written in CNF and in DNF. Then keep app

Answer: Each possible world can be written as a conjunction of literals, e.g. $(A \wedge B \wedge \neg C)$. Asserting that a possible world is not the case can be written by negating that, e.g. $\neg(A \wedge B \wedge \neg C)$, which can be rewritten as $(\neg A \vee \neg B \vee C)$. This is the form of a clause; a conjunction of these clauses is a CNF sentence, and can list the negations of all the possible worlds that would make the sentence false.

Each possible world can be expressed as the conjunction of all the literals that hold in the model. The sentence is then equivalent to the disjunction of all these conjunctions, i.e., a DNF expression.

Ex. 4 Construct an algorithm that converts any sentence in propositional logic into DNF.

Short answer: convert to negation normal form with De Morgan laws then distribute AND over OR.

Ex. 5 There are three suspects for murder: Jørgen, Thomas, and Nina. Jørgen, says 'I didn't do it. The victim was old acquaintance of Thomas'. But Nina hated him.' Thomas states 'I didn't do it. I didn't know the guy. Besides I was out of town all the week.' Nina says 'I didn't do it. I saw both Jørgen and Thomas downtown with the victim that day; one of them must have done it.' Assume that the two innocent people are telling the truth, but that the guilty might not be. Write out the facts as sentences in Propositional Logic, and use propositional resolution to solve the crime.

Answer: Convert to propositional logic and look for contradictions:

- Jørgen says: $\neg M_J \wedge K_T \wedge H_N$
- Thomas says: $\neg M_T \wedge \neg K_T \wedge \neg I_T$
- Nina says: $\neg M_N \wedge I_J \wedge I_T$

where J, T and N denote Jørgen, Thomas and Nina respectively. M_X denotes that X is the murderer, K_X that X knew the victim, H_X that X hated the victim and I_X that X was in town.

Jørgen says that Thomas knew the victim while Thomas says he did not (contradiction), thus one of them must be lying. Similarly, Nina says that Thomas was in town while Thomas says he wasn't (contradiction), and one of them must be lying as well. Assuming that the two innocent people are telling the truth, Thomas must be the murderer.

Ex. 6 (7.19 from the book) Convert the following set of sentences to clausal form.

- $A \leftrightarrow (C \vee E)$
- $E \rightarrow D$
- $B \wedge F \rightarrow \neg C$
- $E \rightarrow C$
- $C \rightarrow F$
- $C \rightarrow B$

Give a trace of the execution of DPLL on the conjunction of these clauses.

Answer:

We have following clauses that we should convert to CNF:

1. $A \leftrightarrow (C \vee E)$
2. $E \rightarrow D$
3. $(B \wedge F) \rightarrow \neg C$
4. $E \rightarrow C$
5. $C \rightarrow F$
6. $C \rightarrow B$

1: $A \leftrightarrow (C \vee E)$

$$\begin{aligned} A \leftrightarrow (C \vee E) &\Leftrightarrow ((A \rightarrow C) \vee (A \rightarrow E)) \wedge (C \rightarrow A) \wedge (E \rightarrow A) \\ &\Leftrightarrow ((\neg A \vee C) \vee (\neg A \vee E)) \wedge (\neg C \vee A) \wedge (\neg E \vee A) \\ &\Leftrightarrow (\neg A \vee C \vee E) \wedge (\neg C \vee A) \wedge (\neg E \vee A) \end{aligned} \tag{1}$$

2: $E \rightarrow D$

$$E \rightarrow D \Leftrightarrow \neg E \vee D \tag{2}$$

3: $(B \wedge F) \rightarrow \neg C$

$$\begin{aligned} (B \wedge F) \rightarrow \neg C &\Leftrightarrow \neg(B \wedge F) \vee C \\ &\Leftrightarrow \neg B \vee \neg F \vee C \end{aligned} \tag{3}$$

4: $E \rightarrow C$

$$E \rightarrow C \Leftrightarrow \neg E \vee C \tag{4}$$

5: $C \rightarrow F$

$$C \rightarrow F \Leftrightarrow \neg C \vee F \tag{5}$$

6: $C \rightarrow B$

$$C \rightarrow B \Leftrightarrow \neg C \vee B \tag{6}$$

DPLL Trace The DPLL algorithm is shown on Figure 1. Above all clauses have been converted to CNF. Clause 1 is splitted into three clauses. The following clauses are used in DPLL:

1. $\neg A \vee C \vee E$
2. $\neg C \vee A$
3. $\neg E \vee A$
4. $\neg E \vee D$
5. $\neg B \vee \neg F \vee C$
6. $\neg E \vee C$
7. $\neg C \vee F$
8. $\neg C \vee B$

function DPLL-SATISFIABLE?(s) **returns** *true* or *false*
  inputs: s, a sentence in propositional logic

  clauses ← the set of clauses in the CNF representation of s
  symbols ← a list of the proposition symbols in s
  return DPLL(*clauses*, *symbols*, { })

---

function DPLL(*clauses*, *symbols*, *model*) **returns** *true* or *false*

  if every clause in *clauses* is true in *model* **then return** *true*
  if some clause in *clauses* is false in *model* **then return** *false*
  P, *value* ← FIND-PURE-SYMBOL(*symbols*, *clauses*, *model*)
  if *P* is non-null **then return** DPLL(*clauses*, *symbols* – *P*, *model* ∪ {*P*=*value*})
  P, *value* ← FIND-UNIT-CLAUSE(*clauses*, *model*)
  if *P* is non-null **then return** DPLL(*clauses*, *symbols* – *P*, *model* ∪ {*P*=*value*})
  P ← FIRST(*symbols*); *rest* ← REST(*symbols*)
  return DPLL(*clauses*, *rest*, *model* ∪ {*P*=*true*}) **or**
      DPLL(*clauses*, *rest*, *model* ∪ {*P*=*false*}))

Figure 1: DPLL algorithm from *Artificial Intelligence:A Modern Approach, Russell & Norvig, 2010*, page 261

By following the algorithm with the symbols set $\{C, A, B, D, E, F\}$ the following can be a trace:

1. (Pure symbol) D is True.
    - Clause 4 is True no matter the value of E
    - The symbols set are updated to: $\{C, A, B, E, F\}$
    - The model set are updated to: $\{D\}$
    - DPLL is performed again.
2. (No Pure Symbol and No Unit Clause) The search branches here. The first branch continues at 3, the second on 9.
3. C is True
    - Clause 1 is True no matter the value of A and E
    - Clause 5 is True no matter the value of B and F
    - Clause 6 is True no matter the value of E
    - Clause 2 is now a unit clause.
    - Clause 7 is now a unit clause.
    - Clause 8 is now a unit clause.
    - A, B, E, and F are now pure symbols
    - The symbols set are updated to: $\{A, B, E, F\}$
    - The model set are updated to: $\{D, C\}$
    - DPLL is performed again.
4. (Pure Symbol) A is True
    - Clause 2 is True.

- Clause 3 is True no matter the value of E
- The symbols set are updated to: $\{B, E, F\}$
- The model set are updated to: $\{D, C, A\}$
- DPLL is performed again.

5. (Pure Symbol) B is True
   - Clause 8 is True
   - The symbols set are updated to: $\{E, F\}$
   - The model set are updated to: $\{D, C, A, B\}$
   - DPLL is performed again.

6. (Pure Symbol) E is False
   - The value of E doesn't have any effect.
   - The symbols set are updated to: $\{F\}$
   - The model set are updated to: $\{D, C, A, B, \neg E\}$
   - DPLL is performed again.

7. (Pure Symbol) F is True
   - Clause 7 is True
   - The symbols set are updated to: $\{\}$
   - The model set are updated to: $\{D, C, A, B, \neg E, F\}$
   - DPLL is performed again.

8. All clauses are true in the model: $\{D, C, A, B, \neg E, F\}$
   - DPLL stops here and returns true because it has found a model.

9. C is False
   - The search never reaches this branch.

   This is due to the first branch (3) evaluating to True.