CS 188 Spring 2019 — Introduction to Artificial Intelligence — Written HW 6 Sol.

Self-assessment due: Monday 3/18/2019 at 11:59pm (submit via Gradescope)

Instructions for self-assessment: Take your original submission and annotate any differences from the provided solutions. For **each subpart** where your original answer was correct, **write "correct"** to demonstrate that you have checked your work. For each subpart where your original answer was incorrect, write out the correct answer and comment on the difference between your answer and the explanation provided in the solutions. You should complete your self-assessment using a **different color** of ink from your original work. If you need to, you can download a PDF copy of your submission from Gradescope.

Your submission must be a PDF that follows the template (page 1 has name/collaborators, question 1 begins on page 2, etc.). Do not reorder, split, combine, or add extra pages. If your original homework submission did not follow the correct format, **you must fix the format to receive credit on your self-assessment**.

If you did not complete some questions in your original submission, first complete those questions without consulting the solutions and then use a different color of ink to conduct a self-assessment.

Q1. Minesweeper

Minesweeper, the well-known computer game, is played on a rectangular grid of N squares with M invisible mines scattered among them. Any square may be probed by the agent; instant death follows if a mine is probed. Minesweeper indicates the presence of mines by revealing, in each probed square, the number of mines that are directly or diagonally adjacent. The goal is to probe every unmined square.

(a) Let $X_{i,j}$ be true iff square $[i,j]$ contains a mine. Write down the assertion that exactly two mines are adjacent to $[1,1]$ as a sentence involving some logical combination of $X_{i,j}$ propositions. (The upper left most corner is $[0,0]$).

Correct Solution: To say that exactly two mines are adjacent is equivalent to saying at least two mines are adjacent and at most two mines are adjacent. To formulate at most 2 mines are equivalent, there will be a conjunction of $\binom{8}{3}$ clauses, saying that every triplet of neighboring squares cannot all be mines. One clause is:

$\neg(X_{0,0} \wedge X_{1,0} \wedge X_{2,0}) \equiv (\neg X_{0,0} \vee \neg X_{1,0} \vee \neg X_{2,0})$

To say that at least 2 mines are adjacent, is equivalent to saying that at most 6 adjacent squares are not hiding mines, which we write as a conjunction of $\binom{8}{7}$ clauses, saying that for every group of 7 neighboring mines, they cannot all not be hiding mines. One clause would be:

$\neg(\neg X_{2,2} \wedge \neg X_{1,2} \wedge \neg X_{0,2} \wedge \neg X_{0,1} \wedge \neg X_{2,1} \wedge \neg X_{0,0} \wedge \neg X_{1,0}) \equiv (X_{2,2} \vee X_{1,2} \vee X_{0,2} \vee X_{0,1} \vee X_{2,1} \vee X_{0,0} \vee X_{1,0})$

Alternate Solution, but difficult to convert to CNF: In order to say "exactly two mines" are adjacent to $[1,1]$ say "at most two mines are adjacent" and "at least two mines are adjacent"

"At most two mines are adjacent" is equivalent to saying not "at least three squares are mines" The assertion will be a disjunction of $\binom{8}{2} = 28$ clauses, with each clause asserting that exactly two neighboring squares have a mine. One clause is: $X_{2,2} \wedge X_{1,2} \wedge \neg X_{0,2} \wedge \neg X_{0,1} \wedge \neg X_{2,1} \wedge \neg X_{0,0} \wedge \neg X_{1,0} \wedge \neg X_{2,0}$.

(b) Generalize your assertion from (a) by explaining how to construct a CNF sentence asserting that k of n neighbors contain mines. (It is fine once you have a logical assertion to say "convert this to CNF").

Correct Solution: Analagous to part a to say that exactly k mines are adjacent is equivalent to saying at least k mines are adjacent and at most k mines are adjacent. To formulate at most k mines are equivalent, there will be a conjunction of $\binom{n}{k+1}$ clauses, saying that every group of $k+1$ neighboring squares cannot all be mines.

To say that at least k mines are adjacent, is equivalent to saying that at most $n-k$ adjacent squares are not hiding mines, which we write as a conjunction of $\binom{n}{n-k+1}$ clauses, saying that for every group of $n-k+1$ neighboring mines, they cannot all not be hiding mines.

Alternate Solution (acceptable for self-assessment, given the way part (a) was initially phrased): The original assertion will be a disjunction of $\binom{n}{k}$ clauses, with each clause asserting that exactly k neighboring squares have a mine. Convert this to CNF.

(c) Say you have successfully probed l squares, each of which is separated by a Manhattan Distance of at least 2. Each square has n_i neighbors and the game reveals that the square is surrounded by k_i mines ($i = 1 \dots l$). How can an agent use DPLL to infer whether a given square $[i,j]$ contains a mine, ignoring the global constraint that there are exactly M mines in all? Explain

(i) the query

The query is $X_{i,j}$. The goal is to find out if $KB \models X_{i,j}$.

(ii) the knowledge base

Contains a sentence for each probed square, for a total of l sentences, each a disjunct of $\binom{n_i}{k_i}$ literals.

(iii) how to combine the sentences of the knowledge base into CNF

The conjunction of all the sentences already written in CNF.

(iv) (optional) the number of disjuncts in the CNF.

Each "at least k_i mines in n_i neighbors" statement has $\binom{n_i}{k_i+1} \times (k_i + 1)$ disjuncts, and each "at most k_i mines in n_i neighbors" has $\binom{n_i}{n_i-k_i+1} \times (n_i - k_i + 1)$ disjuncts. Therefore the total number of disjuncts is: $\sum_i \binom{n_i}{k_i+1} \times (k_i + 1) + \binom{n_i}{n_i-k_i+1} \times (n_i - k_i + 1)$.

(d) Explain how to write the global constraint using the notation from part (a). How does the number of clauses in the constraint depend on M and N? Suggest a way to modify DPLL so that the global constraint does not need to be represented explicitly.

To encode the global constraint that there are M mines altogether, we can construct a disjunction of $\binom{N}{M}$ conjunctions, each of size N. However, we can represent the global constraint within the DPLL algorithm itself. We add the parameter min and max to the DPLL function; these indicate the minimum and maximum number of unassigned symbols that must be true in the model. For an unconstrained problem the values 0 and N will be used for these parameters. For a minesweeper problem the value M will be used for both min and max. Within DPLL, we fail (return false) immediately if min is less than the number of remaining symbols, or if max is less than 0. For each recursive call to DPLL, we update min and max by subtracting one when we assign a true value to a symbol.

(e) Are any conclusions derived by the method in part (c) invalidated when the global constraint is taken into account?

No.

Q2. DPLL

Convert the following set of sentences to clausal form.

(a) S1: $A \Leftrightarrow (B \wedge E)$.

$(\neg A \vee B) \wedge (\neg A \vee E) \wedge (A \vee \neg B \vee \neg E)$

(b) S2: $E \Rightarrow D$.

$(\neg E \vee D)$

(c) S3: $C \wedge F \Rightarrow \neg B$.

$(\neg C \vee \neg F \vee \neg B)$

(d) S4: $E \Rightarrow B$.

$(\neg E \vee B)$

(e) S5: $B \Rightarrow F$.

$(\neg B \vee F)$

(f) S6: $B \Rightarrow C$.

$(\neg B \vee C)$

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

1. $(\neg A \vee B) \wedge (\neg A \vee E) \wedge (A \vee \neg B \vee \neg E) \wedge (\neg E \vee D) \wedge (\neg C \vee \neg F \vee \neg B) \wedge (\neg E \vee B) \wedge (\neg B \vee F) \wedge (\neg B \vee C)$
2. Assign pure symbol $D = true$: $(\neg A \vee B) \wedge (\neg A \vee E) \wedge (A \vee \neg B \vee \neg E) \wedge (\neg C \vee \neg F \vee \neg B) \wedge (\neg E \vee B) \wedge (\neg B \vee F) \wedge (\neg B \vee C)$
3. No pure symbols or unit clauses so assign $A = true$: $(B) \wedge (E) \wedge (\neg C \vee \neg F \vee \neg B) \wedge (\neg E \vee B) \wedge (\neg B \vee F) \wedge (\neg B \vee C)$
4. Assign unit clauses to be true $B = true, E = true$: $(\neg C \vee \neg F) \wedge (F) \wedge (C)$
5. Assign unit clauses to be true $C = true, F = true$: $false$.
6. Since the assignment was unsatisfactory we need to backtrack to last free assignment and assign $A = false$: $(\neg B \vee \neg E) \wedge (\neg C \vee \neg F \vee \neg B) \wedge (\neg E \vee B) \wedge (\neg B \vee F) \wedge (\neg B \vee C)$
7. Assign pure symbols $E = false$: $(\neg C \vee \neg F \vee \neg B) \wedge (\neg B \vee F) \wedge (\neg B \vee C)$

8. Assign pure symbol $B = false$: $true$
9. Algorithm terminates. Found a satisfying model $\{A = false, B = false, D = true, E = false\}$. C and F can be anything.

Q3. Inference with First Order Logic

Suppose you are given the following axioms:

1. $0 \leq 3$.
2. $7 \leq 9$.
3. $\forall x,\ x \leq x$.
4. $\forall x,\ x \leq x + 0$.
5. $\forall x,\ x + 0 \leq x$.
6. $\forall x, y,\ x + y \leq y + x$.
7. $\forall w, x, y, z,\ w \leq y \wedge x \leq z \Rightarrow w + x \leq y + z$.
8. $\forall x, y, z,\ x \leq y \wedge y \leq z \Rightarrow x \leq z$.

(a) Give a backward-chaining proof of the sentence $7 \leq 3 + 9$. (Be sure, of course, to use only the axioms given here, not anything else you may know about arithmetic.) Show only the steps that leads to success, not the irrelevant steps.

Goal: $7 \leq 3 + 9$. From (8) and {x/7, z/3+9} derive two subgoals: $7 \leq y1$, $y1 \leq 3 + 9$.

Goal: $7 \leq y1$. Resolve with (4) and substitution {y1/7+0}.

Goal: $7 + 0 \leq 3 + 9$. From (8) and {x2/7+0, z2/3+9} derive two subgoals: $7 + 0 \leq y2$, $y2 \leq 3 + 9$.

Goal: $7 + 0 \leq y2$. Resolve with (6) and substitution {y2/0+7, x3/7, y3/0}.

Goal: $0 + 7 \leq 3 + 9$. From (7) and substitution {w4/0, x4/7, y4/3, z4/9} derive two subgoals: $0 \leq 3$, $7 \leq 9$

Goal: $0 \leq 3$. Resolve with (1).

Goal: $7 \leq 9$. Resolve with (2).

(b) Give a forward-chaining proof of the sentence $7 \leq 3 + 9$. Again, show only the steps that lead to success.

i. From (7) {w/0, y/3, x/7, z/9} infer that $0 + 7 \leq 3 + 9$.
ii. From (6) {y1/0, x1/7} infer that $7 + 0 \leq 0 + 7$.
iii. From (4) {x2/7} infer that $7 \leq 7 + 0$.
iv. From (8), (ii), (iii) {x3/7, y3/7+0, z3/0+7} infer that $7 \leq 0 + 7$.
v. From (8), (i), (iv) {x4/7, y4/0+7, z4/3+9} infer that $7 \leq 3 + 9$.