| | | |
|---|---|---|
| CS 188<br>Fall 2002 | Introduction to AI<br>Stuart Russell | Midterm |

You have 1 hour and 20 minutes. The exam is open-book, open-notes. 100 points total.

You will not necessarily finish all questions, so do your best ones first.
Write your answers in blue books. Check you haven't skipped any by accident. Hand them all in. Panic not.

**1. (18 pts.) True/False**
Decide if each of the following is true or false. If you are not sure you may wish to provide a *brief* explanation to follow your answer.

(a) (3) Breadth-first search is complete even if zero step-costs are allowed.

(b) (3) Every valid sentence is satisfiable.

(c) (3) $(A \wedge B) \Rightarrow C$ entails $(A \Rightarrow C) \vee (B \Rightarrow C)$.

(d) (3) Some pruning is possible in game trees with chance nodes.

(e) (3) Assume that a rook can move on a chessboard any number of squares in a straight line, vertically or horizontally, but cannot jump over other pieces. Manhattan distance is an admissible heuristic for the problem of moving the rook from square A to square B in the smallest number of moves.

(f) (3) Every CSP with only unary constraints is soluble.

**2. (20+5 pts.) Constraint satisfaction**
Consider the problem of tiling a surface (completely and exactly covering it) with $n$ dominoes ($2\times 1$ rectangles). The surface is an arbitrary edge-connected (i.e., adjacent along an edge, not just a corner) collection of $2n$ $1\times 1$ squares (e.g., a checkerboard, a checkerboard with some squares missing, a $10 \times 1$ row of squares, etc.).

(a) (8) Formulate this problem precisely as a CSP where the dominoes are the variables (i.e, define the variable domain and the constraints).

(b) (8) Formulate this problem precisely as a CSP where the squares are the variables, keeping the state space as small as possible. [*Hint: does it matter which particular domino goes on a given pair of squares?*]

(c) (4) Construct a surface consisting of 6 squares such that your CSP formulation from part (b) has a *tree-structured* constraint graph.

(d) (5 extra credit) Describe exactly the set of solvable instances that have a tree-structured constraint graph.

**3. (24 pts.) Propositional Logic**
A propositional *2-CNF* expression is a conjunction of clauses, each containing *exactly 2* literals, e.g.,

$$(A \vee B) \wedge (\neg A \vee C) \wedge (\neg B \vee D) \wedge (\neg C \vee G) \wedge (\neg D \vee G)\ .$$

(a) (8) Prove using resolution that the above sentence entails $G$.

(b) (8) Two clauses are *semantically distinct* if they are not logically equivalent. How many semantically distinct 2-CNF clauses can be constructed from $n$ proposition symbols?

(c) (6) Using your answer to (b), prove that propositional resolution always terminates in time polynomial in $n$ given a 2-CNF sentence as input containing no more than $n$ distinct symbols.

(d) (2) Explain why your argument in (c) does not apply to 3-CNF.

**4. (20 pts.) First-order logic**

For each of the following sentences in English, decide if the accompanying first-order logic sentence is a good translation. If not, explain why not and correct it. (Some sentences may have more than one error!)

(a) "No two people have the same social security number."

$$\neg\exists\, x, y, n \;\; Person(x) \land Person(y) \;\Rightarrow\; [HasSS\#(x,n) \land HasSS\#(y,n)]$$

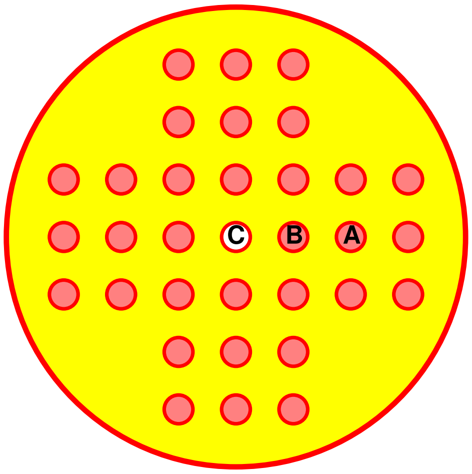
(b) "John's social security number is the same as Mary's."

$$\exists\, n \;\; HasSS\#(John,n) \land HasSS\#(Mary,n)$$

(c) "Everyone's social security number has nine digits."

$$\forall\, x, n \;\; Person(x) \;\Rightarrow\; [HasSS\#(x,n) \land Digits(n,9)]$$

(d) (6) Rewrite each of the above (uncorrected) sentences using a function symbol $SS\#$ instead of the predicate $HasSS\#$.




**5. (18 pts.) Planning**

Consider the problem of *peg solitaire*, as shown in the figure above. The goal is to remove all the pieces but one, which must be left in the center hole C. A piece is removed by hopping an adjacent piece over the piece in question into an empty adjacent hole. (For example, Piece A can hop over piece B into hole C, removing piece B.)

We will formulate peg solitaire as a partial-order planning problem using STRIPS operators.

(a) (4) Describe the start and finish steps for this problem (no need to list EVERY literal one by one, but say what literals are needed). You will need to choose a vocabulary. You may assume that the predicate $Line(u,v,w)$ (positions $u$, $v$, $w$ are consecutive locations along a line) is available and that the constant $C$ refers to the center location.

(b) (8) Now write the STRIPS operator for making a move. Be sure to strictly observe the syntactic restrictions on STRIPS operators.

(c) (6) Describe how a clobbering conflict can occur in the course of partial-order planning with this particular formulation, or explain why one cannot occur.