CS 188 Fall 2005 — Introduction to AI — Stuart Russell — Midterm Solution

1. (12 pts.) True/False

(a) (2) True; both because it is often possible to make changes to an environment without affecting optimal action choices (e.g., small changes to outcome probabilities, monotonic transformations in deterministic game payoffs) and because a single agent (e.g., a minimax game player) can act optimally in two environments by responding appropriately to the different percepts the two environments supply.

(b) (2) False; Ch. 3 gives examples of solving the sensorless vacuum world. Solutions to sensorless problems are actions sequences, just as for fully observable deterministic problems.

(c) (2) False; although the circuit-based agent does not have to be replicated over time, its circuit may be exponentially larger than the propositional KB.

(d) (2) False; e.g., $\exists\, x \;\; P(x) \land \neg P(x)$.

(e) (2) False; it might be unlucky with the dice. (A perfectly rational agent never loses at chess when playing White, we assume.)

(f) (2) False; a lucky DFS might expand exactly d nodes to reach the goal. A* largely dominates any graph-search algorithm that is *guaranteed to find optimal solutions*.

2. (16 pts.) Problem solving

(a) (6) Initial state: two arbitrary 8-puzzle states. Successor function: one move on an unsolved puzzle. (You could also have actions that change both puzzles at the same time; this is OK but technically you have to say what happens when one is solved but not the other.) Goal test: both puzzles in goal state. Path cost: 1 per move.

(b) (4) Each puzzle has $9!/2$ reachable states (remember that half the states are unreachable). The joint state space has $(9!)^2/4$ states.

(c) (2) This is like backgammon; expectiminimax works.

(d) (4) Actually the statement in the question is not true (it applies to a previous version of part (c) in which the opponent is just trying to prevent you from winning—in that case, the coin tosses will eventually allow you to solve one puzzle without interruptions). For the game described in (c), consider a state in which the coin has come up heads, say, and you get to work on a puzzle that is 2 steps from the goal. Should you move one step closer? If you do, your opponent wins if he tosses heads; or if he tosses tails, you toss tails, and he tosses heads; or any sequence where both toss tails n times and then he tosses heads. So his probability of winning is *at least* $1/2 + 1/8 + 1/32 + \cdots = 2/3$. So it seems you're better off moving *away* from the goal. (There's no way to stay the same distance from the goal.) This problem unintentionally seems to have the same kind of solution as suicide tictactoe with passing. So everyone got full credit for this question, and extra credit if they made progress towards the right answer.

3. (16 pts.) Propositional logic

(a) (5) If the clauses have no complementary literals, they have no resolvents. If they have one pair of complementary literals, they have one resolvent, which is logically equivalent to itself. If they have more than one pair, then one pair resolves away and the other pair appears in the resolvent as $(\ldots A \lor \neg A \ldots)$ which renders the resolvent logically equivalent to $True$.

(b) (3) True. $(C \lor (\neg A \land \neg B)) \equiv (C \lor \neg A) \land (C \lor \neg B)) \equiv ((A \;\Rightarrow\; C) \land (B \;\Rightarrow\; C))$.

(c) (4) True (simple argument from models).

(d) (4) False. α can entail the disjunction without committing to either disjunct. Consider the trivial case where α is just $\beta \lor \gamma$.

4. (16 pts.) Logical knowledge representation

(a) (8) (b) and (c). (a) makes no sense because it uses $Child$ as a function. (d) uses $\Rightarrow$ with $\exists$.

(b) (8) "Everyone's DNA is unique and is derived from their parents' DNA."
$DNA(x)$ is the string of DNA characters of person x. (Notice that English is a bit loose: if two people "have the same DNA," it means shared character strings, not shared molecules!)
$DerivedFrom(u, v, w)$ means string u is derived from v and w. (No need to go deeper here.)
$\forall x, y \;\; (\neg(x = y) \Rightarrow \neg(DNA(x) = DNA(y))) \wedge DerivedFrom(DNA(x), DNA(Mother(x)), DNA(Father(x)))$

5. (18 pts.) Logical inference

(a) (12)

i. (3) $Ancestor(Mother(y), John)$: Yes, $\{y/John\}$ (immediate).

ii. (3) $Ancestor(Mother(Mother(y)), John)$: Yes, $\{y/John\}$ (second iteration).

iii. (3) $Ancestor(Mother(Mother(Mother(y))), Mother(y))$: Yes, $\{\}$ (second iteration).

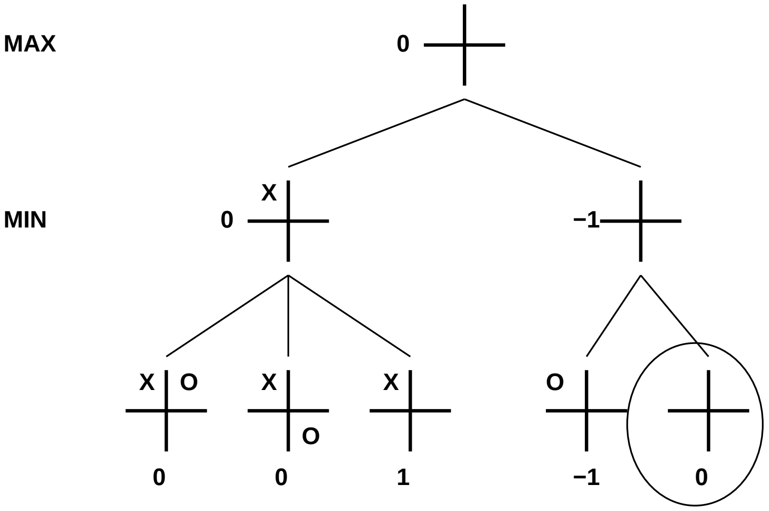
iv. (3) $Ancestor(Mother(John), Mother(Mother(John)))$: Does not terminate.

(b) (3) Although resolution is complete, it cannot prove this because it does not follow. Nothing in the axioms rules out the possibility of everything being the ancestor of everything else.

(c) (3) Same answer.

6. (22 pts.) Game playing

(a) (6)



(b) (4) See tree.

(c) (4) See tree. (If your tree is in a different order, it might have no pruned leaf.)

(d) (4) Minimax will loop forever. Because alpha-beta, with the right move ordering, prunes the no-move node as soon as it finds a sure win for X, it avoids the loop.

(e) (4) In the suicide case, the no-move solution is optimal for both players. Minimax cannot return this is a solution because it *requires* going into the infinite loop! We can avoid the infinite loop in minimax by recognizing that the current node is identical to an earlier node. But we need a way to give it a "value" so we can choose a move. We could assign 0 for "draw", but this is not right in cases where the game is winnable by one player or another from the repeated position. Instead, we can assign "?" and use the fact that a win is better than or equal to "?" which is better than or equal to a loss. This can be encoded directly into the inequality tests in the Min-Value and Max-Value functions.