CS 188 Spring 2004 — Introduction to AI — Stuart Russell

Final Solution

1. (10 pts.) Some Easy Questions to Start With

(a) (2) *True*: in every model, *Rumpelstiltskin* maps to an object, and the sentence is true when x maps to that object also.

(b) (2) *False*: "context" in CFG refers to the syntactic context of a nonterminal.

(c) (2) *True*: trivially, the output variable can have as parents all the input variables and then has a CPT of size 2^n.

(d) (2) *False*: A* search is often used in robotics; the space can be discretized or skeletonized.

(e) (2) *False*: probabilistic inference is often used in robotics; continuous variables are very common in probability theory.

2. (10 pts.) Propositional Logic

(a) (1) *True*: in fact, *every* sentence is true in all models where $False$ is true (none).

(b) (1) *False*: $False$ is not true in any model, let alone every model.

(c) (2) *True*: $A \Leftrightarrow B$ is true when both A and B are true.

(d) (2) *True*: A true and B false together satisfy the sentence.

(e) (2) *True*: this is simply the satisfiability of the conjunction.

(f) (2) *True*: for it to resolve with itself, the clause must contain a symbol and its negation, which renders the whole clause equivalent to $True$.

3. (10 pts.) First-Order Logic

(a) (4) $\forall\, d, a, s\;\; Locked(d, Result(a, s)) \;\Leftrightarrow\; [a = Lock(d) \lor (Locked(d, s) \land a \neq Unlock(d))]$

(b) (4) $\forall\, x\;\; Key(x) \;\Rightarrow\; \exists\, t_1\;\; Lost(x, t_1) \land \forall\, t_2\;\; Before(t_1, t_2) \;\Rightarrow\; Lost(x, t_2)$

(c) (2) *False*: consider the clause $\neg P(c) \lor P(x)$.

4. (22 pts.) Probabilistic inference, decision theory

(a) (3) (i) and (ii).
(i) is true because L is a nondescendant and I has no parents.
(ii) is true—both are equal to $\mathbf{P}$(E—P).
(iii) cannot be asserted by the network structure—the CPTs may imply equality.

(b) (3) $P(i, h, \neg l, p, \neg e) = P(i)P(h)P(\neg l|h)P(p|i, h, \neg l)P(\neg e|p)$
$= .5 \times .1 \times .7 \times .4 \times .4 = .056$

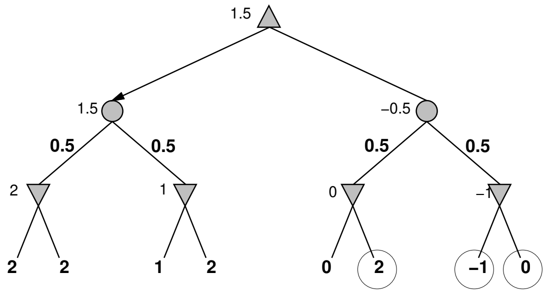
(c) (4) $\mathbf{P}(I|h, \neg l, e) = \alpha\mathbf{P}(I, h, \neg l, e)$
$= \alpha(\mathbf{P}(I, h, \neg l, p, e) + \mathbf{P}(I, h, \neg l, \neg p, e))$
$= \alpha(\langle .084, .063\rangle + \langle .021, .0245\rangle) = \alpha\langle .105, .0875\rangle \approx \langle .545, .455\rangle$

(d) (6) R is a parent of E and a child of H, possibly L, possibly P (because someone who is popular has no need to rig the election). The CPT for E is the same as before given $R = false$ but when $R = true$ it should show a high probability for E perhaps with less dependence on P. The CPT for R should show $R = true$ is less likely when $H = true$.

(e) (2) *False*: such a model asserts, among other things, that both candidates can be elected.

(f) (4) Add a *Donation* action node as a parent of L and a value node as a child of E.

5. (13 pts.) Game playing



(a) (3) See figure.

(b) (2) *False*: if they were both $+\infty$ then the values of the min node and chance node above would also be $+\infty$ and the best move would change.

(c) (2) *True*: even if it is $+\infty$, the min node cannot be worth more than -1, so the chance nod above cannot be worth more than -0.5, so the best move won't change.

(d) (2) (iii): the worst case is if the third and fourth leaves are -2, in which case the chance node above is 0.

(e) (4) See figure.

6. (20 pts.) Language and statistical learning

(a) (2) (iii): we can have infinitely long *AdjString*s, although the probability decreases exponentially with length.

(b) (3) (i) and (ii).

(c) (5) "The green green" can be generated two ways, so we add the probabilities. If the first green is an adjective, the probability is $.6 \times .5 \times .5 \times .5 \times .5 \times .4 = 0.015$. If the first green is a noun, the probability is $.4 \times .5 \times 1.0 \times .5 \times .5 = 0.05$. So the total probability is 0.065.

(d) (2) (i) and (ii).

(e) (2) *True*: this simply requires summing over all possible parses, and there are finitely many of them.

(f) (1) (iii).

(g) (5) If we consider the simplest kind of maximum likelihood learning, for grammars viewed as generative models, the argument fails. The grammar that accepts every sentence must put a very small probability on each, so the likelihood for such a grammar will be much lower than for the correct grammar. Another way to say this is that if the grammar that accepts every sentence were the true grammar, we would see many more sentences than we actually do!

7. (15 pts.) Robotics

(a) (5) (iii) is correct.
(i) is wrong because it shows that the forearm can rotate more as the robot moves into the gap, which is not the case. (It was generated by ignoring the horizontal edges of the square obstacles.) (ii) ignores the self-intersection constraint. (iv) is missing the discontinuity for the corners of the obstacle, and shows that the forearm can rotate less as the robot moves into the gap, which is not the case. (It was generated with two diagonal walls instead of the squares.)

(b) (5) S is at $d = 3$ in the upper gap, G is at $d = 3$ in the center, and the path goes all the way to the left, around the obstacle, and back to the right.

(c) (5) There are several solutions. Perhaps the simplest is as follows:
– reduce d to 1;
– increase θ to $+\pi/2$;
– decrease θ to $-\pi/2$, measuring the time t to do this.
– increase θ for time $t/2$; – increase d to 3.