CS 188 Fall 1997	Introduction to AI Stuart Russell	Solutions to Final

1. **(12 pts.) True/False**
Decide if each of the following is true or false. If you are not sure you may wish to provide some explanation to follow your answer.
   (a) (2) True
   (b) (2) True
   (c) (2) False
   (d) (2) True
   (e) (2) False
   (f) (2) False

2. **(10 pts.) Logic *True/false*:**
   (a) (2) True
   (b) (2) True
   (c) (3) True
   (d) (3) False

3. **(5 pts.) Logic contd.**
In addition to $Q(f(a,a))$ and $Q(x)$, we have $Q(f(x,a))$, $Q(f(a,x))$, $Q(f(x,x))$, $Q(f(x,y))$, $Q(x)$, and the empty clause.

4. **(18+5 pts.) Neural networks**
   (a) (3) Weights are all 1 and threshold is 2.5.
   (b) (3) A linear tree, with "true" branches pointing to "true" leaves and the last "false" branch pointing to "false".
   (c) (4) Why: Simulated annealing can converge to global minima, whereas simple gradient descent reaches only local minima. Also, other global search algorithms such as discrete best-first will hae huge memory problems.
   How: Successor chosen randomly, eg uniformly from ϵ-sphere in weight space. The energy function should be the error function. The annealing schedule may need to be adjusted based on progress.
   (d) (4) $E = \sum_e |T^e - O^e| = Np(1-O_e) + N(1-p)(O_e) = Np + N(1-2p)O_e$
   This is minimized by making O_e as large as possible, i.e., $O_e = 1$.
   (e) (4)
$$\begin{aligned} E &= \sum_e f(T^e - O^e) \\ &= Npf(1-O^e) + N(1-p)f(-O_e) \\ &= Np(1-O^e)^n + N(1-p)(-O_e)^n \end{aligned}$$

We want to have $\partial E/\partial O_e$ to be zero where $O_e = p$:

$$\begin{aligned}\frac{\partial E}{\partial O_e} &= -Npn(1-O^e)^{n-1} - N(1-p)n(-O_e)^{n-1}\\ 0 &= Npn(1-p)^{n-1} + N(1-p)n(-p)^{n-1}\\ (1-p)^{n-2} &= (-p)^{n-2}\end{aligned}$$

which can only hold for general p if $n = 2$.

(f) (5, extra credit) TBD

5. (20+5 pts.) Search and MDPs

(a) (3) A $\rightarrow$B,C; B $\rightarrow$A,C; C at depth 1 is a goal so search ends.

(b) (2) Depth-first fails.

(c) (2) BFS and IDS find the suboptimal solution at depth 1; only uniform-cost search finds the optimal path.

(d) (3) The new equation is just

$$U(i) = \max_a [R(a,i) + \sum_j M^a_{ij} U(j)]$$

(e) (5) Yes: a finite search problem is a deterministic MDP:

- states $\rightarrow$states;
- $M^a_{ij} = 1$ if $a.j \in S(i)$ for some a and 0 otherwise;
- $R(a,i)$ = MINUS the step cost from i to the successor reached by a;
- goal nodes are terminal nodes with no possible actions.

To translate an optimal policy back to a search path, simply start from the start state and follow the optimal action at each state until the goal is reached.

(f) (5) The process goes as follows:

State	Iteration							
	0	1	2	3	4	5	6	...
A	0,B	-1,B	-2,B	-3,B	-4,B	-5,B		
B	0,A	-1,A	-2,A	-3,A	-4,A/C	-4,C		
C	0	0	0	0	0	0		

(g) (5, EXTRA CREDIT) Policy iteration will NOT work because value determination will fail: the initial policy has a loop between A and B whose value is $-\infty$. The failure arises from the deterministic nature of search problems. Essentially any loop in a policy will cause failure. MDPs must be ergodic for success: any policy eventually visits all states. Value iteration also fails if, eg, the step cost of AB goes to zero, since the values of A and B never decrease to -4. Iterative deepening algorithms handle such problems in search representations.

6. (16 pts.) Natural language

(a) (6) (i), (ii). Sentence (iii) fails because modifiers are not allowed for ProperNouns.

(b) (4) The only tricky bit is seeing that "green golf club" has to be Noun* Noun; green cannot be an adjective in this sentence.

(c) (2) 4, lexical ambiguity of red and green.

(d) (4) S $\rightarrow$Either NP VP Alternative$^+$
Alternative $\rightarrow$or NP VP

7. (19 pts.) Probabilistic inference

(a) (3) (i) only.

(b) (2) (i)

(c) (3) $P(C|L,S) = \langle 0.1, 0.7, 0.6, 0.8\rangle$ for FF,FT,TF,TT respectively. FF should be appreciable and TT should be more than FT and TF.

(d) (3) $P(B|S) = \sum_l P(B|l)P(l|S)$

(e) (4) $P(L|B) = \alpha P(B|L)P(L) = \alpha P(B|L)\sum_s P(L|s)P(s)$

(f) (4) $P(Y|X)$ by conditioning on Y's parents, and on their parents, and so on. We will need to condition on all Y's ancestors except the ancestors of X.