2006 Comprehensive Examination Solutions
Artificial Intelligence

1. Propositional Constraint Satisfaction. (20 points) Taken from a Final exam in CS188 at UCB.

(a) $n + 1$ solutions. Once any X_i is true, all subsequent X_js must be true. Hence, each solution consists of *I* falses followed by $n - i$ trues for $i = 0, \ldots, n$.

(b) Quadratic in n. Consider what part of the complete binary tree is explored during the search. The algorithm must follow all solutions sequences, which themselves cover a quadratic –sized portion of the tree. Failing branches are those trying a false after a preceding variable is assigned true. Such conflicts are detected immediately.

(c) True. Use the Forward Chaining Algorithm in Russell and Norvig.

(d) True. Directed arc consistency in Russell and Norvig.

2. Logic. (20 points) Adapted from a problem in CS157 at Stanford.

(a) True
(b) False
(c) True
(d) False
(e) False
(f) True
(g) False
(h) False
(i) False
(j) True

3. Resolution. (20 points) Adapted from a problem in CS157 at Stanford.

1. $\{\neg p(x, y), q(x, y, f(x,y))\}$	Premise a
2. $\{\neg r(y, z), q(a, y, z)\}$	Premise b
3. $\{r(y, z), \neg q(a, y, z)\}$	Premise b
4. $\{p(x, g(x)), q(x, g(x), z)\}$	Premise c
5. $\{\neg r(x,y), \neg q(x, w, z)\}$	Negated Goal
6. $\{\neg q(a, x, y), \neg q(x, w, z)\}$	3, 5
7. $\{q(x, g(x), f(x, g(x))), q(x, g(x), z)\}$	1, 4
8. $\{\neg q(g(a), w, z)\}$	6, 7 (factoring 7)
9. $\{\}$	7, 8 (factoring 7)

4. Bayes Nets. Taken from a Final exam in CS188 at UCB.

(a) Assertions (2) and (3) are implied by the structure of the net; assertion (1) is not.

(b) $p(b, i, \neg m, g, j) = p(b) * p(\neg m) * p(i \mid b, -m) * p(g \mid b, i, m) * p(j \mid g)$
$= 0.9 * 0.9 * 0.5 * 0.8 * 0.9 = 0.2916$

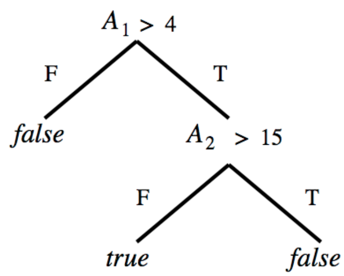
(c) Since B, I, M are true in the evidence, we can treat G as having a prior of 0.9 and look at the submodel with just G and J.

$$p(j \mid b, g, m) = p(j \mid g) * p(g) = 0.9 * 0.9 = 0.81$$

That is, the probability of going to jail is 0.81.

5. Learning. Taken from a Final exam in CS188 at UCB.

(a) One possibility follows.



(b) With 2 examples of each kind, the initial entropy is 1 bit. After the test, we have one subset with counts 0, 1 and one subset with counts 2, 1. Hence the information gain is as follows.

$$
\begin{aligned}
&1- ((1/4 * 0) + (3/4) * (-1/3 \log(1/3) - 2/3 \log(2/3)) \\
&\quad = 1+ (1/4) * \log(1/3) + (1/2) * \log(2/3) \\
&\quad \sim 0.3113 \text{ bits}
\end{aligned}
$$

(c) False. A test-once tree can with one attribute creates exactly two regions on the real line, whereas the data may alternate along the line.

(d) True. A test-many tree can define arbitrarily small hyper-rectangles, each containing exactly one example.