6.825 Midterm Solutions, Fall 2006

November 7, 2006

1 Search (16 points)

Let's formulate resolution theorem proving as a search problem.

- A state is a set of clauses.
- The descendants of a state are all the ways that the state can be augmented by the result of a possible resolution in that state (which yields a new clause not in the state).
- The goal is a state that includes a contradiction (null clause).

Answer the following questions and provide a brief explanation justifying your answers.

1. Assuming we are dealing with propositional logic clauses, which search should you use: Depth First, Breadth First, Uniform Cost or A^*?

   We know the space is bounded so all the methods will eventually find an answer, the only question is how much work they will do. Using depth-first means that we will only resolve a particular pair of clauses only once, but we might find a proof that is very long. Breadth-first will find the shortest proof, but it will end up resolving the same pair of clauses potentially many times as it considers different orders of doing resolution along different branches. Uniform cost will behave like breadth-first (with additional overhead. A^* *with a heuristic related to size of clauses so that it prefers to do unit resolutions could be faster on average than uniform cost or breadth first, but it would still repeat resolutions on different branches.*

   The key point is that the way this search has been set up is wasteful because it's searching over different orders of doing the resolutions, but that doesn't affect the answer; proofs don't depend on the order of the resolutions. In fact, one should do all the possible resolutions at each step so we don't have to search over order. This is called a "level-saturation" strategy.

2. Is your search guaranteed to terminate? Will it find a contradiction if one exists?

   We know that repeatedly applying propositional resolution (without duplication) will eventually either run out of resolutions or find a contradiction if one exists. So, any systematic search method will eventually terminate and find a contradiction if one exists.

3. Assuming we are dealing with first order logic clauses, would you need to change the formulation of the problem and/or your choice of search to get a theorem prover that is guaranteed to find a contradiction if one exists? If so, explain how.

One part of the answer is that we need factoring and paramodulation (unless we axiomatize equality) in addition to resolution to get completeness. The other part is that now the search space can be infinite. Consider the following clauses: $\neg P(x) \vee P(f(x))$ $P(f(y))$ *Note that the resolvent is* $P(f(f(y)))$ *and then you can use it again and again... So, depth-first is out as a possible search strategy.*

4. Is your search guaranteed to terminate? Will it find a contradiction if one exists?

   If a contradiction does not exist, the search may not terminate (semi-decidability of first-order logic). If a contradiction exists, the search will terminate due to the refutation-completeness of (resolution + factoring + paramodulation).

2 Entailment (8 points)

Let S_1 and S_2 be sentences in propositional logic, let I_1 be the set of interpretations that make S_1 true, and let I_2 be the set of interpretations that make S_2 true. Assume that I_1 is a subset of I_2. Mark all of the statements which *must* be true in this case:

1. S_1 entails S_2. *Yes*
2. S_2 entails S_1. *No*
3. S_2 can be proven from S_1 with any sound proof system. *Not necessarily*
4. S_2 can be proven from S_1 with any complete proof system. *Yes*
5. The sentence $(S_1 \rightarrow S_2)$ is satisfiable. *Yes*
6. The sentence $(S_1 \rightarrow S_2)$ is valid. *Yes*
7. The sentence $(S_2 \rightarrow S_1)$ is satisfiable. *Not necessarily.* This one is tricky. This sentence might not be satisfiable, if, for instance S_2 is true and S_1 is false.
8. The sentence $(S_2 \rightarrow S_1)$ is valid. *No, if it's not satisfiable, it can't possibly be valid.*

3 Writing FOL (6 points)

Write the following sentence in standard First Order Logic notation. Use only the following one-place predicates: $Dog(.)$, $Cat(.)$, and $Bird(.)$ and the following two-place predicates $Eat(.,.)$, and $Hate(.,.)$ where the first argument is the subject and the second is the object, that is, $Hate(x,y)$ means that x hates y.

"Some dogs hate all cats who eat birds"

$$\exists d.Dog(d) \wedge \forall c.Cat(c) \wedge (\exists b.Bird(b) \wedge Eat(c,b)) \Rightarrow Hate(d,c)$$

4 Logical Inference (9 points)

1. Write a possible result of applying resolution on $\neg P(y, f(y)) \vee P(x,y)$ and $P(A,w)$.

$$P(x, A)$$

2. Write a possible result of applying paramodulation on $P(f(x)) \vee (f(y) = g(x))$ and $Q(z) \vee P(g(h(z)))$.

   For example,

$$P(f(h(z))) \vee Q(z) \vee P(f(y))$$

3. Write the clause form for $\forall x.((\forall y.P(x,y)) \Leftrightarrow (\exists y.Q(x,y)))$

   Two clauses:

$$\neg P(x, sk_1(x)) \vee Q(x, sk_2(x))$$

$$\neg Q(x,y) \vee P(x,z)$$

5 First-Order (21 points)

Here are some English sentences and their translation into clausal form.

1. Every state has a governor.
$$G(f(x_1), x_1)$$
2. The governor of a state is in the state.
$$\neg G(x_2, y_2) \vee In(x_2, y_2)$$
3. "In" is transitive.
$$\neg In(x_3, y_3) \vee \neg In(y_3, z_3) \vee In(x_3, z_3)$$
4. Governors are Republicans.
$$\neg G(x_4, y_4) \vee R(x_4)$$
5. MA (a state) is in the US.
$$In(MA, US)$$
6. Prove that: There is a Republican in the US. (Write the clause you would need to include to prove this.)
$$\neg R(x_6) \vee \neg In(x_6, US)$$

We'd like to prove the conclusion using resolution refutation. This proof is kind of tricky, so we're going to tell you, in English, what the steps should be. For each step, say which of the previous clauses (P1 and P2 in the table) it can be derived from using resolution, what the resulting clause is and what the unifier is.

Step	P1	P2	Clause	Unifier
7	Every governor is in their state. (one term)			
	1	2	$In(f(x_1), x_1)$	$\{x_2/f(x_1), y_2/x_1\}$
8	If a state is in some location, then its governor is in that location. (two terms)			
	3	7	$\neg In(x_1, z_3) \vee In(f(x_1), z_3)$	$\{x_3/f(x_1), y_3/x_1\}$
9	The governor of a state is a Republican. (one term)			
	1	4	$R(f(x_1))$	$\{x_4/f(x_1), y_4/x_1\}$
10	The governor of MA is in the US. (one term)			
	5	8	$In(f(MA), US)$	$\{x_1/MA, z_3/US\}$
11	There is no state whose governor is in the US. (one term)			
	6	9	$\neg In(f(x_1), US)$	$\{x_6/f(x_1)\}$
12	False			
	10	11	nil	$\{x_1/MA\}$

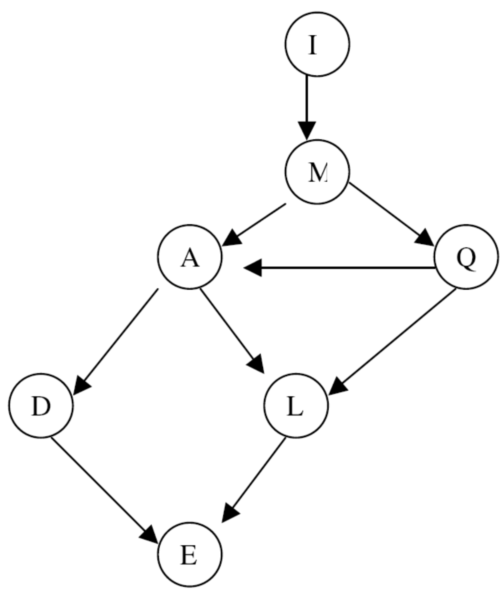
6 Bayesian networks (20 points)

1. (8 pts) Draw a Bayesian network among the following binary variables that model the outcome of an election:

   - I : candidate is Incumbent
   - M : has lots of Money for advertising
   - A : uses advertisements that focus on Attacking the candidate's opponent
   - Q : uses advertisements that focus on the candidate's Qualifications
   - L : candidate is Liked
   - D : opponent is Distrusted
   - E : candidate is Elected

   Your network should encode the following beliefs:

   - Incumbents tend to raise lots of money.
   - Money can be used to buy advertising that either focuses on the candidate's qualifications or that attacks the candidate's opponent. But if one does one, there is less money to do the other.
   - Attack advertisement tend to make voters distrust the opponent but they also make the voters tend not to like the candidate.
   - Advertisement focusing on qualifications tends to make the voters like the candidate.
   - Candidates that people like tend to get elected.
   - Candidates whose opponent people distrust tend to get elected.



   The arrow from Q to A could go either way (but not both, since that would create a loop).

2. (9 pts) For each of the following, say whether it is or is not asserted by the network structure you drew (without assuming anything about the numerical entries in the CPTs)? Explain your answers briefly.

   (a) $P(L|A, Q, D) = P(L|A, Q)$
   Asserted. L and D are d-separated given A and Q and E not given.

   (b) $P(A|M, Q) = P(A|M)$
   Not asserted. Q is a direct ancestor (or descendant, depending on how you draw the arrow between Q and A) of A.

   (c) $P(L, D|A, Q) = P(L|A, Q)P(D|A, Q)$
   This is simply another way of writing that L and D are conditionally independent given A, Q (and not E).

3. (3 pts) What variables, if any, can be ignored if we want to answer the query $P(D|I)$ using variable elimination? Why?

   In the network as we drew it, neither E or L are ancestors of D and I, so they can be ignored.

7 Temporal Models (20 points)

1. For the Rain-Umbrella HMM model (from class and Figure 15.2 in AIMA) with 2 time periods, show how **variable elimination** operates to answer the following query $P(R_1|U_1 = T, U_2 = T)$. You do not need to do numerical calculations.

   We need to eliminate R_0 and R_2, the initial factors are

$$P(R_0), P(R_1|R_0), P(R_2|R_1), P(U_1 = T|R_1), P(U_2 = T|R_2)$$

   Eliminating R_0

$$F_1(R_1) = \sum_{R_0} P(R_0)P(R_1|R_0)$$

   Eliminate R_2

$$F_2(R_1, U_2 = T) = \sum_{R_2} P(U_2 = T|R_2)P(R_2|R_1)$$

   The final answer is the product of these three factors:

$$F_1(R_1)P(U_1 = T|R_1)F_2(R_1, U_2 = T)$$

2. Explain the relationship between the variable elimination computation and the **forward/backward algorithm**. Be specific, referring to the factors obtained during variable elimination.

   The forward message in a smoothing computation would be: $P(R_1|U_1 = T)$ which is $F_1(R_1)P(U_1 = T|R_1)$, the backward message would be $P(U_2 = T|R_1)$ which is $F_2(R_1, U2 = T)$