COS 470/570

Prelim II

Spring 2017

Put your name on the back of all the pages. Answer the questions in the space provided. If you need additional space, answer on the back of the test, but make sure that you clearly label your answers. There are 85 total points on 7 pages. Good luck!

1. Represent each of the following in first-order logic (FOL, predicate calculus) **as well as** conjunctive normal form (CNF). If a statement cannot be represented in FOL, say why not. Each part lists the predicates you are allowed to use; don't use any additional ones unless you cannot solve the problem otherwise, and you clearly explain why you did. You may create constants as needed. (3 pts each)

   (a) Joe likes everyone Mary likes. (Predicate allowed: Likes(x,y).)

   Ans:

   $\forall x\ Likes(Mary, x) \Rightarrow Likes(John, x)$

   $\neg Likes(Mary, x)\ \vee\ Likes(John, x)$

   (b) I have three cats. (Predicates allowed: Owns(x,y), Cat(x), NotSame(x,y).)

   Ans:

   $\exists x, y, z\, Owns(Me, x) \wedge Owns(Me, y) \wedge Owns(Me, z) \wedge Cat(x) \wedge \cdots \wedge NotSame(x, y) \cdots$

   Each will be a different clause, with each variable replaced by a Skolem constant

   (c) I like anyone who owns a gorilla. (Predicates allowed: Likes(x,y), Gorilla(x), Owns(x,y).)

   Ans:

   $\forall p \exists g\ Owns(p, g)\ \wedge\ Gorilla(g)\ \Rightarrow\ Likes(Me, p)$

   $\neg Owns(p, sk(p)\ \vee\ \neg Gorilla(sk(p))\ \vee\ Likes(Me, p)$

(d) Mary owns a gorilla that is smaller than any chimpanzee. (Predicates allowed: Owns(x,y), Gorilla(x), Chimp(x), SmallerThan(x,y).

Ans:

$\forall c \exists g \, Gorilla(g) \wedge Owns(Mary, g) \wedge Chimp(c) \Rightarrow SmallerThan(g, c)$ or

$\exists g \, Gorilla(g) \wedge Owns(Mary, g) \, (\forall c \, Chimp(c) \Rightarrow (g, c)$

The first really says that for every chimp in the world, Mary owns a gorilla that is smaller than it; thus, Mary could own multiple gorillas, each smaller than some group of chimps, with all chimps in one of the groups. The second is the more correct version.

First one: $\neg Gorilla(sk(c)) \vee \neg Owns(Mary, sk(c)) \vee \neg Chimp(c) \vee SmallerThan(sk(c), c)$

Second: $\neg Gorilla(sk)) \vee \neg Owns(Mary, sk) \vee (\neg Chimp(c) \vee SmallerThan(sk, c)$

As you can see, they differ in that in the second, the *same* gorilla is smaller than all chimps.

(e) I have a cat that has a favorite toy. (Predicates allowed: Cat(x), Owns(x,y), LikesMore(x,y,z) [where x is the thing that likes y more then z], NotSame(x,y), Toy(x).)

Ans:

$\forall z \exists x, y \, Cat(x) \wedge Owns(Me, x) \wedge Toy(y) \wedge Toy(z) \wedge NotSame(y, z) \Rightarrow LikesMore(x, y, z)$

$\neg Cat(sk_1(z) \vee \neg Owns(Me(sk_1(z)) \vee \neg Toy(sk_2(z)) \vee \neg Toy(z) \vee \neg NotSame(sk_2(z), z) \vee LikesMore(sk_1(z), sk_2(z), z)$

2. What is a *frame*, and how are frames used in knowledge representation? (5 pts)

   Ans: A frame is a slot–filler knowledge structure that includes support for inheritance and (usually) for some procedural attachment in its slots. It is used to represent objects in the world (and their properties and relationships with objects) or relationships between objects.

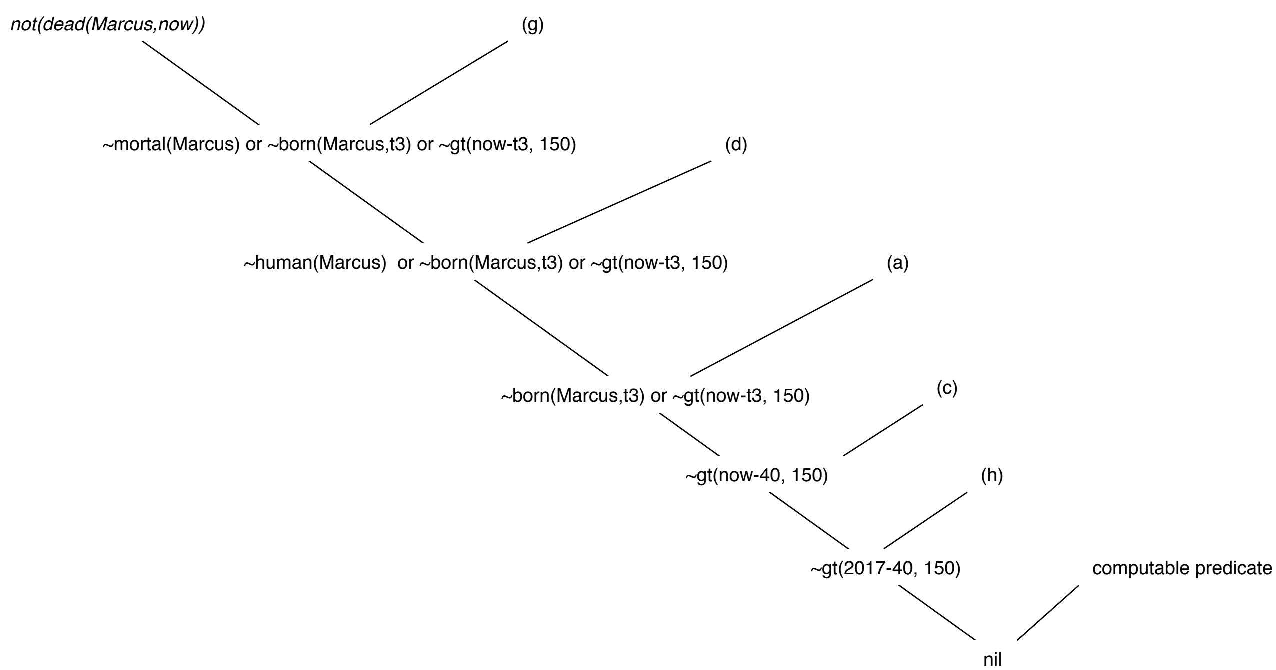
3. Compare and contrast backward- and forward-chaining expert systems in terms of how they work and what they are good for. (10 pts)

Ans: Forward-chaining expert systems apply their rules in a forward direction by comparing their antecedents to the world (i.e, the working memory) and triggering those rules that match; then conflict resolution strategies or rules come into play to select a single rule to fire, which changes the world (and/or working memory). Forward chaining is most often used for synthesis tasks, and often no uncertainty management is needed. Backward chaining expert systems are like foward systems in that they use rules of the same form, but they are primarily used in a hypothetico-deductive manner in which rules that *could* conclude something of interest are used to set up subgoals (from their antecedent clauses) to conclude. Ultimately, a backward chain of reasoning will or will not be supported by facts in the working memory or from the user; if not suported, the chain of reasoning is abandoned, but if it is supported, then evidence propagates from the facts through the reasoning chain to support the conclusion(s). Backward chainers are used for analysis tasks (e.g., diagnosis), and they usually use uncertainty management, often in the form of certainty factors.

4. Given this axiom set:
   (a) $human(Marcus)$
   (b) $Pompeian(Marcus)$
   (c) $born(Marcus, 40)$
   (d) $\neg human(x_1) \vee mortal(x_1)$
   (e) $\neg Pompeian(x_2) \vee died(x_2, 79)$
   (f) $erupted(volcano, 79)$
   (g) $\neg mortal(x_3) \vee \neg born(x_3, t_1) \vee \neg gt(t_2 - t_1, 150) \vee dead(x_3, t2)$
   (h) $now = 2017$
   (i) $\neg alive(x_4, t_3) \vee \neg dead(x_4, t_3)$
   (j) $dead(x_5, t_4) \vee alive(x_5, t_4)$
   (k) $\neg died(x_6, t_5) \vee \neg gt(t_6, t_5) \vee dead(x_6, t_6)$

   - Prove that Marcus is dead now. Show your proof in the form of a proof tree, as discussed in class. (5 pts)

     Ans:



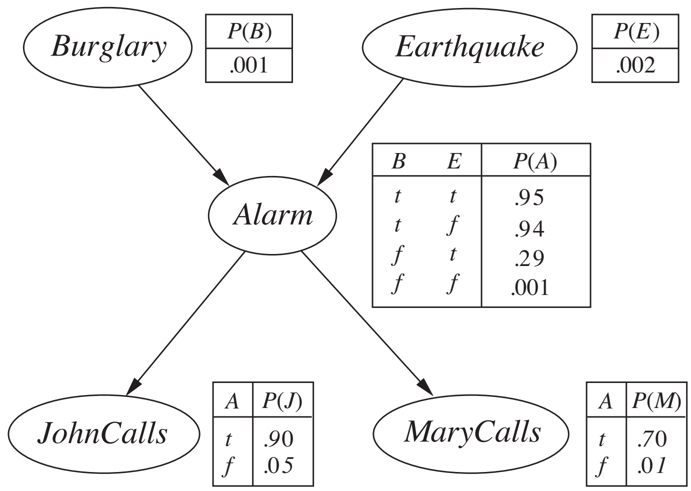
- Suppose that we replace axiom 4b with one that states that Marcus is dead now. Can we now prove that he was Pompeian? If so, show the proof tree; if not, explain why not. (5 pts)

  Ans: We cannot, since $\neg Pompeian(Marcus)$ would now not resolve with any axiom.

5. Compare and contrast these **three** knowledge representation formalisms: first-order logic; frames; and description logic. (10 pts)

   Ans: FOL is a "flat" knowledge representation that has no inherent hierarchical structure, although such a structure can be created by the user via special predicates and rules. It allows quantification (existential as well as universal) and, thus, variables; it can express anything a Turing machine can compute/recognize. Frames are described above; they cluster related knowledge together, include inheritance, etc. DL is similar to both frames and FOL in some ways (esp. frames, from which it arose). It is based on classes of things and reasoning based on subsumption and class membership.

6. Given this Bayesian network:



Just to refresh your memory of how to do things with this net, recall the formula used to compute the probability of a burglary given thatJohn and Mary call:

$$\mathbf{P}(b \mid j, m) = \alpha P(b) \sum_{e} P(e) \sum_{a} P(a \mid b, e) P(j \mid a) P(m \mid a)$$

where a means alarm goes off, b means a burglary occurred, etc. (You won't necessarily use this exact formula below, of course!) (5 pts each)

(a) What is the probability that there has been an earthquake, given that both John and Mary call?

Ans:
First, recall Figure 14.8 in your book, which shows how to compute the probability distribution for burglary given that John and Mary call:

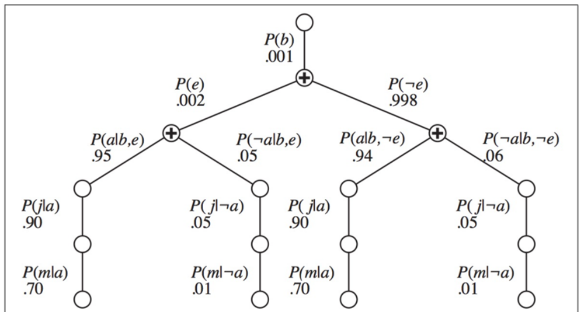
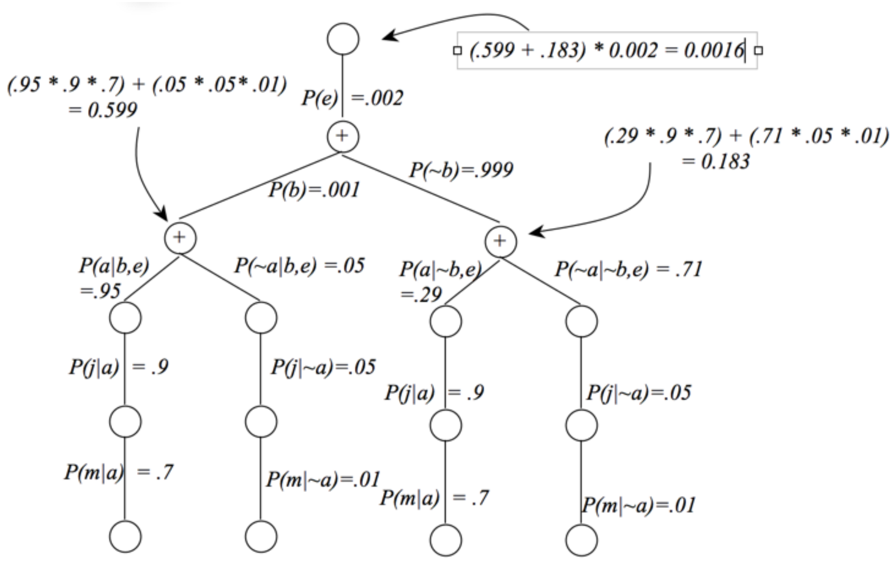


Figure 14.8 The structure of the expression shown in Equation (14.4). The evaluation proceeds top down, multiplying values along each path and summing at the "+" nodes. Notice the repetition of the paths for j and m.

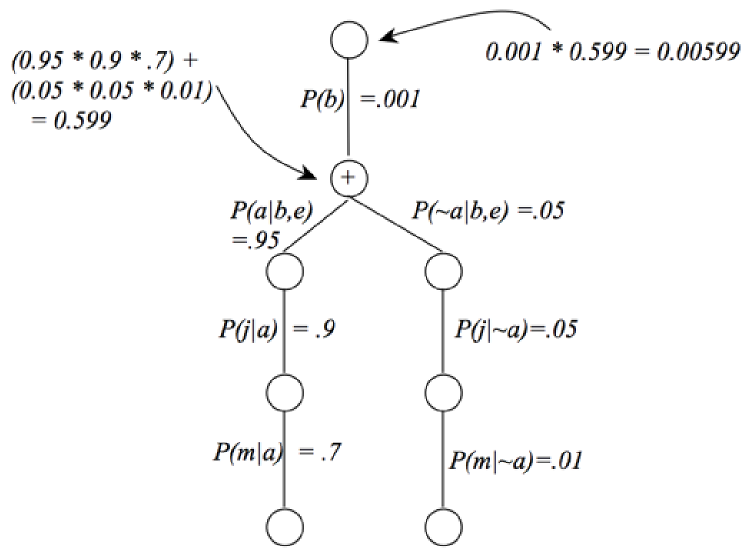
The process for this question is similar:



So $P(e|j,m) = 0.0016$.

(b) Suppose that both John and Mary call and we know that there has been an earthquake; what is the probability that there has also been a burglary?

Ans:
Here, you *know* that there has been an earthquake. Consequently, $P(e) = 1$ and $P(\neg e) = 0$:



So $P(b|e, j, m) = 0.00599$

7. Compare and contrast linear planning (as done, e.g., in STRIPS) with nonlinear planning (as done, e.g., in POP). (5 pts)

   Ans: Linear planners work on one goal at a time, and so if there are dependencies between goals, extra steps are needed in the plan. They in essence over-commit to which goal they are going to work on at any given time. Non-linear planners, on the other hand, do not overcommit and are free to work on multiple goals at once. Dependencies are detected and avoided during planning for any goals, and so more optimal plans can be constructed, even when goals interact.

8. Describe in detail how a graph plan-based planner (e.g., the forward planner GRAPHPLAN) finds a plan for a set of goals. (10 pts)

   Ans: First, all operators (e.g., $Stack(x, y)$) are instantiated against all objects (e.g., A, B), yielding ground-state (instantiated) operators (e.g., $Stack(A, B)$, $Stack(B, A)$, $Stack(A, A)$, $Stack(B, B)$). The initial state is considered the state part of level 0, e.g., S_0. All actions that can be applied to this state, including the persistence action, are applied; these actions constitute the action part of level 0, or A_0. The results of A_0 is S_1, which contains all results from the actions. These are then marked with mutex conditions indicating which are mutually exclusive. The actions in A_0 are marked with mutex conditions, too, indicating which produce conflicting results or which rely on conflicting preconditions. Then the process is repeated to yield A_1 and S_2, etc. When all goal conditions occur and are non-mutex in a level, then the planner searches back through the prior levels to see if a plan can be created. For the state S_n, this entails finding actions in A_{n-1} that produce the goal conditions and that are themselves mutex. Then their preconditions are considered goal conditions, and the process continues. If the initial state is reached, a plan has been found. If not, and there are no such plans, then the planner expands the plan another level.

9. Suppose you are given the assignment of getting a robot to deliver mail on one floor of Boardman Hall. In general: (a) what reasoning technique(s) (e.g., planning, search, RTP, etc.) would you use and why? and (b) what knowledge representation(s) would you choose and why? (10pt)

   Ans: There are a wealth of good answers for this, so I won't choose one to put here. I will say, however, that the knowledge representations you choose need to be consistent with the reasoning techniques you choose! There were quite a few answers that didn't do this.