First Examination
Intelligent Systems

Jan Broersen

Code : INFOIS Date : March 7th, 2013 Time : 08:30-10:30

This exam has 6 questions. Your answers should be given in Dutch or English. With the answers to the 6 questions you can earn 90 points. You get 10 points for free. 100 points yields a 10. This examination only contributes to your end mark if your (weighted) average benefits from it.

Question1	Question2	Question3	Question4	Question5	Question6
15pt	20pt	10pt	15pt	15pt	15pt

FOL = First Order Logic KB = Knowledge Base

1. Determine for each of the following FOL structures whether or not it satisfies the formula: $\forall x \exists y \exists z : P(x,y) \wedge P(x,z) \wedge \neg P(y,z) \wedge \neg P(z,y)$. Explain your answer!

   (a) Domain $= \mathbb{Z}$ (i.e., the integers),
   Relation $P = \{(m, m+1) \mid m \in \mathbb{N}\}$

   answer No. But only because of the $\mathbb{N}$, otherwise it would have been Yes.

   (b) Domain $= \mathbb{Q}$ (i.e., the rational numbers),
   Relation $P = \{(m,n) \mid m,n \in \mathbb{Q} \text{ en } m \leq n\}$

   answer No. Does not work; incompatible order requirements.

   (c) Domain $= 2^{\mathbb{N}}$ (i.e., the set of subsets of the natural numbers),
   Relation $P = \{(A,B) \mid A,B \in 2^{\mathbb{N}} \text{ en } A \subseteq B\}$

   answer No. Does not work. But only because it fails in case we take the whole set.

2. Translate the following sentences into FOL. Use a member predicate to denote that points belong to lines or circles. Use the same predicate to denote that lines or circles belong to a certain class of lines or circles. So points, lines, circles, sets of points, sets of lines and sets of circles are all seen as objects and membership relations are denoted using a membership predicate relating these objects.

   (a) Parallel lines do not have a point in common.

   answer $\forall x,y : M(x,Lines) \wedge M(y,Lines) \wedge x \neq y \wedge PL(x,y) \rightarrow \neg \exists p : M(p,Points) \wedge M(p,x) \wedge M(p,y)$

   (b) Three points uniquely determine a circle.

   answer $\forall x,y,z : M(x,Points) \wedge M(y,Points) \wedge M(z,Points) \rightarrow \exists c : M(c,Circles) \wedge M(x,c) \wedge M(y,c) \wedge M(z,c) \wedge \neg \exists k : M(k,Circles) \wedge k \neq c \wedge M(x,k) \wedge M(y,k) \wedge M(z,k)$

3. Which of the following pairs of predicates can be unified? Give the substitutions in case unification succeeds. In these formulas $Brother$, $Sister$ and $Mother$ are functions, and $WorksFor$, $Loves$ and Old are predicates.

   (a) $WorksFor(Sister(Jan), y)$ and $WorksFor(x, Jan)$

answer simple substitutions

   (b) $Old(Mother(Mother(x))$ and $Old(Mother(x))$

answer fail

   (c) $Loves(Brother(Jan), Piet)$ and $Loves(x, Brother(y))$

answer fail

4. We assume that to move in the wumpus world, an agent only has the possibility to GoEast and GoNorth. Apart from that, the agent can perform the actions Grab-Gold and Shoot, both with the effect suggested by their name. The formula below represents an initial attempt to formulate a successor state axioma based on the fluent $At(Agent, x, y, s)$, where the x and the y are place coordinates (as usual for the situation calculus, all variables are implicitly universally quantified). Finish the formula by substituting a correct formula for the '...'.

$$Poss(a, s) \rightarrow [At(Agent, x, y, Result(a, s)) \Leftrightarrow (\ldots)]$$

answer Successor State Axiom:

$$\begin{aligned} \forall a, s, x, y\ [Poss(a, s) \quad &\rightarrow \quad At(Agent, x, y, Result(a, s))] \\ &\leftrightarrow \quad [(a = \texttt{GoEast} \wedge At(Agent, x - 1, y, s)) \\ &\qquad \vee\ (a = \texttt{GoNorth} \wedge At(Agent, x, y - 1, s)) \\ &\qquad \vee\ (a \neq \texttt{GoEast} \wedge a \neq \texttt{GoNorth} \wedge At(Agent, x, y, s))] \end{aligned}$$

5. If there is something wrong with the following reasoning, then explain where the mistake is: "FOL is complete. So, for every formula that follows from a FOL knowledge base there is a derivation. This means that if we ask a FOL knowledge base to find an answer to the question if a given formula follows from it, we will always get an answer. "

6. We consider the following Prolog program:

```
parent(5,3).
parent(1,2).
parent(2,3).
parent(2,4).
ancestor(X,Y) :- parent(X,Y).
ancestor(X,Y) :- ancestor(Z,Y), parent(X,Z).
```

   What will be the output of the query '`?- ancestor(X,3).`' after repeatedly pushing the "`;`"?