CS 440: Introduction to AI

Homework 2 - Part A (Part B will be available soon.)

Due: September 23 11:59PM, 2010

Your answers must be concise and clear. Explain sufficiently that we can easily determine what you understand. We will give more points for a brief interesting discussion with no answer than for a bluffing answer.

1 Validity and Satisfiability

Determine whether each formula is valid, satisfiable (but not valid), or unsatisfiable.

1. $P(x)$
2. $\forall x\ P(x)$
3. $\exists x \forall y\ P(x, y)$
4. $\forall x\ (P(x) \vee \neg P(x))$
5. $\exists x \forall y\ (P(x, y) \Rightarrow \forall w \exists z\ P(w, z))$
6. $\exists x\ (P(x) \Rightarrow \neg P(x))$
7. $\exists x\ \neg(\neg P(x) \vee P(x))$
8. $\forall x \forall y\ (\neg P(x) \vee P(y)) \Rightarrow (P(x) \Rightarrow P(y))$

2 Sentence Translation

This problem has two parts. In the first part, you are asked to translate some first order predicate calculus (FOPC) sentences into English sentences. The meanings of the predicates and functions in the FOPC sentences are given. You should give as natural the English translations as possible. Few points will be given for an English sentence like "For all x there exists a y such that either P of x and y or Q of y." In the second part, you are asked to translate English sentences into FOPC sentences. If the meaning of the English sentence is ambiguous, give all possible readings. Note that sometimes *and*, *or*, and *if* should not be directly translated into $\wedge$, $\vee$, and $\Rightarrow$ in FOPC. Please define every

predicate and function you use in your FOPC sentences whenever they are not clearly self-explanatory.

If there is a sentence that cannot be translated, please state so and clearly explain why not.

If you do nat have access to the symbols used in FOPC, make the following six substitutions in part 2:

$\wedge$ /\ (slash, backslash)

$\vee$ \/ (backslash, slash)

$\forall$ *forall*

$\exists$ *exists*

$\neg$ $-$

$\Rightarrow$ => (equals, greater than)

Predicate symbol definitions for part 1:

Thief(x**)** x is a thief

Cop(x**)** x is a cop

Student(x**)** x is a student

Bike(x**)** x is a bike

Owns(x**,** y**)** x is the owner of y

Stolen_by(x**,** y**)** x is stolen by y

Caught_by(x**,** y**)** x is caught by y

Accomplice(x**,** y**)** x and y are accomplices

Function symbol definition for part 1:

CarOf(x**)** A car of x.

1. FOPC to English
   (a) $\forall x(\neg\text{Student}(x) \vee \neg\text{Cop}(x))$
   (b) $\exists x \exists y(\text{Cop}(x) \wedge \text{Thief}(y) \wedge \text{Stolen\_by}(\text{CarOf}(x), y))$
   (c) $\forall x \forall y((\exists z(\text{Bike}(z) \wedge \text{Stolen\_by}(z, x) \wedge \text{Accomplice}(x, y))) \Rightarrow \text{Thief}(x \wedge y))$
   (d) $\forall x(\text{Student}(x) \Rightarrow \exists y \exists z(\text{Bike}(y) \wedge \text{Thief}(z) \wedge \text{Owns}(x, y) \wedge \text{Stolen\_by}(y, z)))$
   (e) $\neg\forall x \forall y(\text{Bike}(x) \wedge \text{Thief}(y) \wedge \text{Stolen\_by}(x, y) \Rightarrow \exists z(\text{Caught\_by}(y, \text{Cop}(z))))$
2. English to FOPC

(a) Evey student who takes Korean passes it.

(b) The Fighting Illini is the team that won the most of games in the Big Ten conference.

(c) There is a shef who cooks for all people in town who do not cook for themselves.

3 Unification

For each set of the following atomic sentences, find the *most general unifier* (MGU) and the corresponding unification instance. If certain pair is not unifiable, state so and explain why.

In the sentences below: Is_wizard, Are_brothers, Teaches, and P are predicates; *BestFriendOf*, *FatherOf*, and *SonOf* are functions; *Fred*, *George*, *Ron*, *Harry*, and *James* are constants; x, y, z, u, and v are variables which have already been standardized apart.

1.
   - $P(Fred, x, Ron)$
   - $P(y, y, x)$
2.
   - Is_wizard(x)
   - Is_wizard($BestFriendOf(y)$)
3.
   - Are_brothers(*Fred*, *George*)
   - Are_brothers(x, *Fred*)
4.
   - $P(FatherOf(x), SonOf(x, y), x)$
   - $P(z, u, SonOf(z, u))$
5.
   - $P(SonOf(x, y), FatherOf(Ron), u)$
   - $P(v, FatherOf(u), x)$
6.
   - Teaches($FatherOf(x), x$)
   - Teaches($y, SonOf(z, u)$)
   - Teaches($w, SonOf(v, v)$)
7.
   - Teaches(x, *James*)
   - Teaches($y, FatherOf(Harry)$)

   (We know that Harry's father is James)

4 Soundness and Completeness

For each part below, give an inference rule that has the specified properties. Your inference rules must be different from all of those that appear in the lecture slides.

1. Sound but not complete
2. Complete but not sound
3. Neither sound nor complete