COMP329 KNOWLEDGE SYSTEMS

Assignment 1 — KRR and Prolog (Worth 20%)

Due: 11:55 pm, Friday, 23 September, 2016 (First week of break)

This assignment will be marked out of a maximum of 20. You may want to use LaTeX, or its wysiwyg version Lyx. For the former a good start will be using Texniccentre (www.texniccenter.org/) together with MikTex (miktex.org/). Your lecturers have also heard good things about Texmaker (http://www.xm1math.net/texmaker/), but have not tried.

You should upload soft copy of your solution on iLearn in the form of two files:

1. `answer1.pdf`, *containing your response to the first part of the assignment (Questions 1-4). This document should incorporate the appropriate cover page. Make sure that the document is meant to be printed on A4 size paper with adequate margin all all four sides.*
2. `berlin.pl` *and* `wildlife.pl` *containing the Prolog programs for the second part of the assignment (Questions 5-6).*

Late submissions will not be accepted after 11:55 pm Sunday September 25. No request for extension will be considered unless request for special consideration is formally submitted online.

Knowledge Representation and Reasoning

1. [2.5 marks]

(a) A proposed translation is given of items (i) to Propositional Language, and (ii) to the First Order Language, using the scheme of abbreviation provided below. For each of these two translations, (1) state whether or not it is a correct translation, and (2) provide a short justification (about five to six lines) behind your judgment. If the translation in question is incorrect, give the correct translation.

i. Australia will win only if Italy forfeits, unless Japan refuses to lose.

ABBREVIATION SCHEME USED
A: Australia will win
I: Italy will forfeit
J: Japan will lose

PROPOSED TRANSLATION
$(J \wedge A) \rightarrow I$

ii. Those under seventeen but not accompanied by a parent are not admitted.

ABBREVIATION SCHEME USED
$S(x)$: x is under seventeen
$A(x)$: x is admitted

$P(x, y)$: x is a parent of y
$C(x, y)$: x accompanies y

PROPOSED TRANSLATION
$\forall x((Sx \wedge \exists y(P(yx) \rightarrow \neg C(yx))) \rightarrow \neg A(x))$

(b) The English sentence *Some student is pulled over by a cop every Friday* has **at least** four readings. Translate them into First Order Language, providing with each such sentence how it is to be read off in English. Use the following translation scheme for this purpose:

$F(x)$: x is Friday
$C(x)$: x is a cop
$S(x)$: x is a student
$P(x, y, z)$: x pulls over y on z

2. [2.5 marks]

An attempt is made to prove the following theorem by using the resolution technique:
$\emptyset \vdash (P \vee (\neg P \wedge Q)) \rightarrow (P \vee Q)$

The attempted proof is given below. Is this proof correct? Explain why it is or is not correct. If it is not correct, provide a correct proof.

<u>ATTEMPTED PROOF</u>

OBTAIN CNF OF NEGATION OF CONCLUSION:

(i) $\neg((P \vee (\neg P \wedge Q)) \rightarrow (P \vee Q))$

(ii) $\neg(\neg(P \vee (\neg P \wedge Q)) \vee (P \vee Q))$

(iii) $\neg\neg(P \vee (\neg P \wedge Q)) \wedge \neg(P \vee Q)$

(iv) $P \wedge (\neg P \vee Q)) \wedge \neg P \wedge \neg Q$

RESOLUTION:

(v)	$\neg P \vee Q$	[$\neg$ Conclusion]
(vi)	$\neg P$	[$\neg$ Conclusion]
(vii)	Q	[(v, vi), Resolution]
(viii)	$\neg Q$	[$\neg$ Conclusion]
(ix)	$\square$	[(vii), (viii), Resolution]

3. [2.5 marks]

Your professor made some mistakes when converting the following formula to CNF:

$$\forall x \exists y (H(x) \to M(y)) \leftrightarrow \exists y \forall x (H(x) \to M(y)).$$

Apart from dropping some parentheses for readability, compressing some steps, and leaving out the annotation, he committed two major (types of) errors. You need to:

(a) Identify those errors. Tell in which line(s) the errors occurred, and explain the nature of those errors.

(b) Correct those errors. (You need to provide only those lines, with line numbers, corrected.)

THE CONVERSION DONE BY YOUR PROFESSOR

(a) $\forall x \exists y (H(x) \to M(y)) \leftrightarrow \exists y \forall x (H(x) \to M(y))$

(b) $\forall x \exists y (H(x) \to M(y)) \to \exists y \forall x (H(x) \to M(y)) \;\wedge$
$\exists y \forall x (H(x) \to M(y)) \to \forall x \exists y (H(x) \to M(y))$

(c) $\neg \forall x \exists y (H(x) \to M(y)) \vee \exists y \forall x (H(x) \to M(y)) \;\wedge$
$\neg \exists y \forall x (H(x) \to M(y)) \vee \forall x \exists y (H(x) \to M(y))$

(d) $\neg \forall x \exists y (\neg H(x) \vee M(y)) \vee \exists y \forall x (\neg H(x) \vee M(y)) \;\wedge$
$\neg \exists y \forall x (\neg H(x) \vee M(y)) \vee \forall x \exists y (\neg H(x) \vee M(y))$

(e) $\exists x \forall y (H(x) \wedge \neg M(y)) \vee \exists y \forall x (\neg H(x) \vee M(y)) \;\wedge$
$\forall y \exists x (H(x) \wedge \neg M(y)) \vee \forall x \exists y (\neg H(x) \vee M(y))$

(f) $\exists x_1 \forall y_1 (H(x_1) \wedge \neg M(y_1)) \vee \exists y_2 \forall x_2 (\neg H(x_2) \vee M(y_2)) \;\wedge$
$\forall y_3 \exists x_3 (H(x_3) \wedge \neg M(y_3)) \vee \forall x_4 \exists y_4 (\neg H(x_4) \vee M(y_4))$

(g) $\forall y_1 (H(a_1) \wedge \neg M(y_1)) \vee \forall x_2 (\neg H(x_2) \vee M(b_2)) \;\wedge$
$\forall y_3 (H(c) \wedge \neg M(y_3)) \vee \forall x_4 (\neg H(x_4) \vee M(c))$

(h) $(H(a_1) \wedge \neg M(y_1)) \vee (\neg H(x_2) \vee M(b_2)) \;\wedge$
$(H(c) \wedge \neg M(y_3)) \vee (\neg H(x_4) \vee M(c))$

(i) $(\, H(a_1) \vee \neg M(y_1) \vee \neg H(x_2) \,) \;\wedge\; (\, H(a_1) \vee \neg M(y_1) \vee \neg M(b_2) \,) \;\wedge$
$(\, H(c) \vee \neg M(y_3) \vee \neg H(x_4) \,) \;\wedge\; (\, H(c) \vee \neg M(y_3) \vee \neg M(c) \,)$

4. [2.5 marks]

Consider the argument:

> Those who love their partners don't love themselves. Therefore nobody's partner loves everybody.

Using the abbreviation scheme: $L(x, y)$: x loves y, and $p(x)$: the partner of x, we translate this argument as:

$\forall x \; (L(x, p(x)) \to \neg L(x, x)) \vdash \forall x \; \exists y \neg L(p(x), y).$

Show that this argument holds, by using First Order Resolution.

Warning: *Remember to negate the conclusion before converting it to CNF.*

Prolog Programming

The next two questions, 5 and 6, assess your Prolog Programming and problem solving skills by asking you to solve two logic puzzles.

5. **Berlin Marathon** [4 marks]
   Write a Prolog program that solves the following puzzle.

   > Sam, Pedro and Yusuf live in different countries and have different jobs. They meet every year at the Berlin marathon in Germany where one of them lives. This year Yusuf arrived before the Australian in the race. Pedro was faster than the cook. Also this year the winner of the race was the university professor. By the way, Yusuf is an engineer and Pedro lives in France.
   > Who lives where, has what job, and came first, second, and third?

   The Prolog program should return for the query:

   ```
   ?- solve_marathon(S).
   ```

   a list that contains three predicates of the form:

   ```
   rank(Name, Job, CountryName, Rank).
   ```

6. **At the Wildlife Park** [6 marks]

   Write a Prolog program that solve the following puzzle.

   > There are nine cages in the local wildlife park, in the positions shown below. From the clues given below, can you state which animal is in which cage? Throughout the clues, east means due east, north-west means due north-west etc.

   Clues:

   (a) The lion is north of the platypus, but south of the zebra.
   (b) The ape is west of the tiger.
   (c) The giraffe is immediately south-east of the camel.
   (d) The koala is east of the lion.
   (e) The elephant is immediately south-west of the koala.

```
----------    ----------     -----------
|        |    |        |     |         |
|   1    |    |   2    |     |    3    |
|        |    |        |     |         |
----------    ----------     -----------

----------    ----------     -----------
|        |    |        |     |         |
|   4    |    |   5    |     |    6    |
|        |    |        |     |         |
----------    ----------     -----------

----------    ----------     -----------
|        |    |        |     |         |
|   7    |    |   8    |     |    9    |
|        |    |        |     |         |
----------    ----------     -----------
```

The Prolog program should print out for the query:

```
?- solve_wildlife.
```

the following information:

```
[[1, Animal1], [2, Animal2], [3, Animal3]]

[[4, Animal4], [5, Animal5], [6, Animal6]]

[[7, Animal7], [8, Animal8], [9, Animal9]]
```

where `AnimalX` is one of the 9 animals.