UNIVERSITY OF ABERDEEN **SESSION 2011-2012**

Examination in CS3019 (Knowledge-based Systems)

23 January 2012 **(12:00 – 14:00)**

Candidates are not permitted to leave the Examination Room during the first or last half hours of the examination.

Answer TWO out of the three questions.
Appendix (including some tables and algorithms used in lectures) are available after the questions.

Each question is worth 25 marks; the marks for each part of a question are shown in brackets.

1. (a) What is a consistent ontology? What is a satisfiable class? Use some example to illustrate your points. [6]

(b) Given the following interpretation

1. Δ^{I} ={a,b,c,d,e,f,g,h,i}
2. $Plant^{I}$= {a,c,f}
3. Animal ={d,e,i}
4. eat^{I}={<d,a>,<d,b>,<e,c>,<e,i>}
5. $partOf^{I}$= {<c,a>,<a,f>};

Write down the interpretations of the following class descriptions and show your working. [4]

- $\neg$Plant
- $\forall$eat.(Plant $\sqcup$ $\exists$partOf.Plant)

(c) How does explicit representation of knowledge benefit software engineering (as of in the Alive EU project) in terms of openness and flexibility? [3]

(d) Given the following OWL axioms, (1) write down the corresponding RDF statements in N3, (2) write down these RDF statements in rules (3) use OWL axioms (in DL syntax) to express the part of the semantic net that is not expressible in RDF. [12]

1. Class (Lecturer partial Staff)
2. SubClassOf (Staff restriction(workIn someValueFrom(Organisation)))
3. EquivalantClasses (Lecturer AsistentProf)
4. DisjointClasses (Male Female)
5. ObjectProperty (primarilyTeach super(teach) domain (Lecturer) range (Course))
6. SubClasssOf (L3 restriction (teach allValuesFrom (Level3Course)))
7. SameIndividual (cs3019 kbs)

2.

(a) Is the following class equivalence valid? Prove your answer. [6]

$$C \sqcap (D \sqcup E) \equiv (C \sqcap D) \sqcup E$$

(b) Briefly explain the five problems in ontology reuse discussed in this course. [5]

(c) Consider the ontology O2 consisting of the following axioms:

Class(Cow partial Herbivore)
Class(MadCow partial (intersectionOf(Cow restriction(eat someValuesFrom(AnimalComponent)))))
Class(Herbivore partial restriction(eat allValuesFrom(Vegetable)))
DisjointClasses(AnimalComponent Vegetable)
Individual(Mary type(MadCow))

Write down the above ontology in DL syntax and use the tableaux algorithm to check if O2 is consistent. [14]

TURN OVER

3.

(a) The Jess expert system shell uses two main data structures to execute Jess programs: the so-called Working Memory (holding a list of facts) and the Agenda, holding a list of currently activated Jess rules. List the means provided by Jess that allow to influence the ordering of activated rules on the agenda and describe how they work. [5]

(b) Assume that the Working Memory of Jess contains the following facts:

Fact-1: (week monday tuesday wednesday thursday friday)
Fact-2: (shopping eggs bread beans milk salt soap)

Assume that the following patterns are condition elements of the left-hand side (LHS) of rules. Describe whether they will match the facts in Working Memory and what values are bound to any variable because of this match:

1. **(week $? ?x ?y&:(neq ?y thursday))**
2. **(shopping eggs $? ?x $?)**
3. **(shopping $?x milk ?y)**
4. **(week $? ?x ?y&:(neq ?y wednesday) $?)**
5. **(shopping $? $?groceries&:(> (length$?groceries) 3) ?)**

[10]

(c) Write a small Jess program that simulates a room thermostat. A thermostat can be set to a particular temperature and will turn on the heating (opens the radiator valve) when the temperature measured by a sensor falls below the thermostat's setting and turns off the heating when the temperature rises above this setting. The "tolerance" of the thermostat determines how far below or above the thermostat's setting the room temperature can fall or rise, before the thermostat will act.

We assume that a thermostat has a "location", a "setting" and a "tolerance": the thermostat is situated in a room, is set to a particular temperature and has a particular tolerance.
We assume that there are sensors that are associated with thermostats, each sensor has therefore a "location" and provides an actual "temperature" reading.

The thermostat will use measured temperature, setting and tolerance to determine when to turn on and off the heating:
- If it is true that "temperature" <= ("setting" – "tolerance") then turn on the heating
- If it is true that "temperature" >= ("setting" + "tolerance") then turn off the heating

The template for the thermostat is defined in the following way:

```
(deftemplate thermostat
      (slot location    (type SYMBOL))
      (slot setting     (type INTEGER))
      (slot tolerance   (type INTEGER))
)
```

(i) Define a template that represents a sensor (a sensor has a "location" and a "temperature" reading). Add type specifications for slots.

[4]

(ii) Specify two rules: a rule "open-valve" and a rule "close-valve".

Rule "open-valve" should check whether "temperature" <= ("setting" – "tolerance"). If that is the case it should print out on the RHS that the heating at the thermostat's "location" is turned on.

Rule "close-valve" should check whether "temperature" >= ("setting" + "tolerance"). If that is the case it should print out on the RHS that the heating at the thermostat's "location" is turned off.

Use constraints or tests to encode these conditions. [6]

TURN OVER (APPENDIX AVAILABLE)