Answers G53KRR 2009-10

1. (a) Express the following sentences in first-order logic, using unary predicates `Fragile`, `Break`, `Fall`, `TennisBall`.

**S1** Fragile things break if they fall (2 marks)

**S2** Tennis balls are not fragile (2 marks)

**S3** Tennis balls don't break if they fall (2 marks)

*Answer*

**S1** $\forall x(\texttt{Fragile}(x) \land \texttt{Fall}(x) \supset \texttt{Break}(x))$

**S2** $\forall x(\texttt{TennisBall}(x) \supset \neg\texttt{Fragile}(x))$

**S3** $\forall x(\texttt{TennisBall}(x) \land \texttt{Fall}(x) \supset \neg\texttt{Break}(x))$

(b) Give a definition of logical entailment for first-order logic, that is, what it means for a set of sentences $S$ to logically entail a sentence $A$. (4 marks)

*Answer* $S$ logically entails $A$ if for every interpretation $I$, if $I$ makes true all sentences in $S$, it also makes $A$ true.

(c) Do sentences **S1** and **S2** logically entail **S3**? Justify your answer: if you answered yes, show why the entailment holds, if you answered no, give a counterexample interpretation. (10 marks)

*Answer* No, **S1** and **S2** do not logically entail **S3**. An interpretation which satisfies **S1** and **S2** but not **S3** would be for example where *all* things break when they fall: the domain is $D = \{d_1, d_2\}$, $I(\texttt{Fragile}) = \{d_1\}$, $I(\texttt{TennisBall}) = \{d_2\}$, $I(\texttt{Fall}) = \{d_1, d_2\}$, and $I(\texttt{Break}) = \{d_1, d_2\}$.

[3 marks for a correct answer without a counterexample, complete marks for any correct counterexample, pro-rata for partial answers.]

(d) Is there a fully automatic terminating procedure to establish, for any finite set of sentences $S$ and a sentence $A$, whether $S$ logically entails $A$ in either propositional or first-order logic? If the answer is yes to either of these, then describe that procedure. (5 marks)

*Answer* There is no such procedure for first-order logic. For propositional logic, one could for example construct a truth table for $S$ and $A$ and check whether all assignments satisfying $S$ also satisfy $A$.

[2 marks for correct answers without describing the procedure, full marks for a complete answer (the answer can use other procedures, e.g. resolution rather than truth tables).]

2. (a) Reduce the following sentences to clausal form:

**S1** $\forall x \exists y \forall z \exists u P(x, y, z, u)$ (2 marks)

**S2** $\forall x \forall y (R(x, y) \vee \exists z R(y, z))$ (2 marks)

**S3** $\forall x \forall y ((R(x, y) \vee Q(x, y)) \supset (R(y, x) \vee Q(y, x)))$ (2 marks)

*Answer*

**S1** $[P(x, f(x), z, g(x, z))$

**S2** $[R(x, y), R(y, f(x, y))]$

**S3** $[\neg R(x, y), R(y, x), Q(y, x)]$, $[\neg Q(x, y), R(y, x), Q(y, x)]$

(b) Show by resolution that clauses **C1**–**C3** below entail $P(a, f(f(a)))$.

**C1** $[\neg P(x, y), \neg P(y, z), P(x, z)]$

**C2** $[\neg P(x, f(x)), P(f(x), f(f(x)))]$

**C3** $[P(a, f(a))]$ (10 marks)

*Answer*

**1** $[\neg P(x, y), \neg P(y, z), P(x, z)]$ C1

**2** $[\neg P(a, f(f(a)))]$ C4

**3** $[\neg P(a, y), \neg P(y, f(f(a)))]$ res. 1,2, $x/a, z/f(f(a))$

**4** $[P(a, f(a))]$ C3

**5** $[\neg P(f(a)), f(f(a)))]$ res. 3,4, $y/f(a)$

**6** $[\neg P(x, f(x)), P(f(x), f(f(x)))]$ C2

**7** $[P(f(a)), f(f(a)))]$ res. 4,6, $x/a$

**8** [] res. 5,7.

[Clearly other correct derivations are possible and will receive full marks. Incorrect applications of the resolution rule will be penalised.]

(c) Give an SLD resolution of $C$ from the following clauses: (9 marks)

**A1** $[A_1]$

**A2** $[A_2]$

**A3** $[A_3]$

**A4** $[A_4]$

**B1** $[\neg A_1, \neg A_2, B_1]$

**B1** $[\neg A_3, \neg A_4, B_2]$

**C** $[\neg B_1, \neg B_2, C]$

*Answer* $[\neg B_1, \neg B_2]$, $[\neg A_1, \neg A_2, \neg B_2]$, $[\neg A_2, \neg B_2]$, $[\neg B_2]$, $[\neg A_3, \neg A_4]$, $[\neg A_4]$, []. [Clearly other correct derivations are possible and will receive full marks. Steps which do not follow by the resolution rule will be penalised.]

3. (a) Define a Horn clause, a unit clause, a positive Horn clause and a negative Horn clause. Is the empty clause positive or negative? (5 marks)

*Answer* A Horn clause is a clause with at most one positive literal. A unit clause is a clause with one literal. A positive Horn clause contains one positive literal. A negative Horn clause has no positive literals. The empty clause is negative. [1 mark each.]

(b) State the forward chaining procedure for checking whether a set of positive unit clauses follows from a set of positive Horn clauses (for propositional case). (5 marks)

*Answer*

**Forward chaining** .
**input:** an atomic sentence $q$
**output:** YES if KB entails $q$, NO otherwise

1. if $q$ is marked as solved, return YES
2. if there is $[\neg p_1, \ldots, \neg p_n, q_1]$ in KB such that $p_1, \ldots, p_n$ are marked as solved and $q_1$ is not marked solved: mark $q_1$ as solved and go to 1; else return NO.

(c) State the backward chaining procedure for checking whether a set of positive unit clauses follows from a set of positive Horn clauses (for propositional case). (5 marks)

*Answer*

**Backward chaining** .
**input:** a finite set of atomic sentences $q_1, \ldots, q_n$
**output:** YES if KB entails all of $q_i$, NO otherwise

```
procedure: SOLVE[q1, ..., qn ]
if n = 0 then return YES
for each clause c in KB do
   if  c = [not p1,...,not pm, q1] and SOLVE [p1,...,pm,q2,...,qn]
   then return YES
end for
return NO
```

(d) Explain how the order of attempting clauses (rules) to match and the order of solving subgoals may affect efficiency of the backward chaining procedure. (10 marks)

*Answer* Depending on the order in which clauses are used to generate subgoals and on the in which subgoals are attempted, the search may or may not terminate. For example, assuming Prolog clause and subgoal ordering (clauses in the order they are listed, subgoals left-to-right), the set of clauses $[Ancestor(x, y), \neg Ancestor(z, y), \neg Parent(x, z)]$, $[Ancestor(x, y), \neg Parent(x, y)]$, corresponding to the Prolog program

$$ancestor(X, Y) : -ancestor(Z, Y), parent(X, Z)$$

$$ancestor(X, Y) : -parent(X, Y)$$

will go into an infinite loop if given a query $ancestor(anne, X)$. On the other hand, for a different clause ordering

$$ancestor(X, Y) : -parent(X, Y)$$

$$ancestor(X, Y) : -parent(X, Z), ancestor(Z, Y)$$

will not.

Sometimes one order of subgoals gives rise to a more efficient search than another:

$$americanCousin(X, Y) : -american(X), cousin(X, Y)$$

vs

$$americanCousin(X, Y) : -cousin(X, Y), american(X)$$

Consider a goal $americanCousin(X, sally)$: first program will need to check for all Americans if any of them is Sally's cousin; the second one checks for all cousins of Sally if they are an American. [All things being equal, 5 marks for showing how the order of clause processing matters, and 5 marks for how the order of subgoal processing matters, although a very careful discussion of just subgoal ordering may merit more than 5 points.]

4. (a) What is the point of description logics and other ontology languages? Why don't knowledge representation professionals use first-order logic for everything? (5 marks)

*Answer* The main reason for using resticted ontology languages is that restriction in expressive power makes reasoning in them much more efficient than in first order logic. Another reason is that hierarchies of concepts, and entity-relationship diagrams are familiar to many users and are considered more intutitive. [Full marks can be obtained by making only the first point properly.]

(b) Recall the description logic $\mathcal{DL}$ given in the textbook:

`******************definition of DL******************************`

**Concepts**:

- atomic concept is a concept
- if $r$ is a role and $b$ is a concept, then [**ALL** $r$ $b$] is a concept (e.g. [**ALL** : *Child Girl*] describes someone all of whose children are girls).
- if $r$ is a role and $n$ is a positive integer, then [**EXISTS** $n$ $r$] is a concept (e.g. [**EXISTS** 2 : *Child*] describes someone who has at least 2 children)
- if $r$ is a role and $c$ is a constant, then [**FILLS** $r$ $c$] is a concept (e.g. [**FILLS** : *Child john*] describes someone whose child is John).
- if $b_1, \ldots, b_n$ are concepts, [**AND** $b_1 \ldots b_n$] is a concept.

**Sentences**:

- if $b_1$ and $b_2$ are concepts then $b_1 \sqsubseteq b_2$ is a sentence (all $b_1$s are $b_2$s, $b_1$ is *subsumed* by $b_2$)
- if $b_1$ and $b_2$ are concepts then $b_1 \doteq b_2$ is a sentence ($b_1$ is equivalent to $b_2$)
- if $c$ is a constant and $b$ a concept then $c \rightarrow b$ is a sentence (the individual denoted by $c$ satisfies the description expressed by $b$).

`******************end definition of DL***************************`

Express the following concepts and sentences in $\mathcal{DL}$ using constants `john`, `g51prg`, roles `Module` and `Supervision` and atomic concepts `Academic`, `Lecturer`, `Compulsory`:

**C1** concept of an academic who has some project students (2 marks)

**C2** concept of an academic who teaches at least two modules (2 marks)

**C3** concept of an academic who teaches only compulsory modules (2 marks)

**C4** concept of someone who teaches G51PRG (2 marks)

**S1** a lecturer is an academic who has at least 8 project students and teaches at least 2 modules (4 marks)

**S2** John teaches at least 3 modules and they are all compulsory (3 marks)

*Answer*

**C1** [**AND** `Academic` [**Exists** 1 `Supervision`]]

**C2** [**AND** `Academic` [**Exists** 2 `Module`]]

**C3** [**AND** `Academic` [**ALL** `Module Compulsory`]]

**C4** [**FILLS** : `Module g51prg`]

**S1** `Lecturer` $\doteq$ [**AND** `Academic` [**Exists** 8 `Supervision`][**Exists** 2 `Module`]]

**S2** `john` $\rightarrow$ [**AND** [**Exists** 3 `Module`] [**ALL** `Module` `Compulsory`]]

(c) At the moment the logic does not contain concept negation **NOT**. It also cannot say that there exists some individual connected by a role which is in a concept $b$ (namely, we have [**ALL** $r$ $b$] but no [**EXISTS** $r$ $b$]). If we add concept negation **NOT**, with the obvious meaning that **NOT** $b$ is a concept containing all individuals which are not in $b$, explain how we can then define [**EXISTS** $r$ $b$]. (5 marks)

*Answer* [**EXISTS** $r$ $b$] = [**NOT** [**ALL** $r$ **NOT** $b$]]

5. (a) Define closed world assumption and the entailment relation $\models_{CWA}$. (3 marks)

*Answer* Closed-world assumption (CWA): if an atomic sentence is not classically entailed by the knowledge base, it is assumed to be false. The corresponding entailment $\models_{CWA}$:

$$KB \models_{CWA} \phi \;\Leftrightarrow\; KB^+ \models \phi$$

where $KB^+ = KB \cup \{\neg p : p \text{ is atomic and } KB \not\models p\}$.

(b) Consider the following knowledge base:

$KB = \{$
$\mathtt{NorthOf}(\mathtt{york}, \mathtt{edinburgh})$,
$\mathtt{NorthOf}(\mathtt{london}, \mathtt{nottingham})$,
$\mathtt{NorthOf}(\mathtt{york}, \mathtt{durham})$,
$\mathtt{NorthOf}(\mathtt{london}, \mathtt{york})$,
$\forall x \forall y \forall z (\mathtt{NorthOf}(x, y) \wedge \mathtt{NorthOf}(y, z) \supset \mathtt{NorthOf}(x, z))$
$\}$

i. Does it hold that $KB \models_{CWA} \mathtt{NorthOf}(\mathtt{london}, \mathtt{edinburgh})$? Explain why. (3 marks)

ii. Does it hold that $KB \models_{CWA} \neg\mathtt{NorthOf}(\mathtt{london}, \mathtt{edinburgh})$? Explain why. (3 marks)

iii. Does it hold that $KB \models_{CWA} \mathtt{NorthOf}(\mathtt{nottingham}, \mathtt{edinburgh})$? Explain why. (3 marks)

iv. Does it hold that $KB \models_{CWA} \neg\mathtt{NorthOf}(\mathtt{nottingham}, \mathtt{edinburgh})$? Explain why. (3 marks)

*Answer*

i. $KB \models_{CWA} \mathtt{NorthOf}(\mathtt{london}, \mathtt{edinburgh})$ because $\mathtt{NorthOf}(\mathtt{london}, \mathtt{york})$, $\mathtt{NorthOf}(\mathtt{york}, \mathtt{edinburgh})$ and $\forall x \forall y \forall z (\mathtt{NorthOf}(x, y) \wedge \mathtt{NorthOf}(y, z) \supset \mathtt{NorthOf}(x, z))$ classically entail $\mathtt{NorthOf}(\mathtt{london}, \mathtt{edinburgh})$.

ii. $KB \not\models_{CWA} \neg\mathtt{NorthOf}(\mathtt{london}, \mathtt{edinburgh})$ because there is an interpretation satisfying $KB^+$ and $\mathtt{NorthOf}(\mathtt{london}, \mathtt{edinburgh})$ (in fact all interpretations of $KB^+$ satisfy $\mathtt{NorthOf}(\mathtt{london}, \mathtt{edinburgh})$).

iii. $KB \not\models_{CWA} \mathtt{NorthOf}(\mathtt{nottingham}, \mathtt{edinburgh})$ because $\mathtt{NorthOf}(\mathtt{nottingham}, \mathtt{edinburgh})$ does not follow classically from $KB$ (a consistent interpretation for $KB$ which does not satisfy $\mathtt{NorthOf}(\mathtt{nottingham}, \mathtt{edinburgh})$ can be obtained by interpreting $\mathtt{nottingham}$ as St.Andrews), so its negation is in $KB^+$ and since $KB^+$ is consistent, $\mathtt{NorthOf}(\mathtt{nottingham}, \mathtt{edinburgh})$ is not entailed by it.

iv. $KB \models_{CWA} \neg\mathtt{NorthOf}(\mathtt{nottingham}, \mathtt{edinburgh})$ because $\mathtt{NorthOf}(\mathtt{nottingham}, \mathtt{edinburgh})$ does not follow classically from $KB$, so its negation is in $KB^+$.

(c) Give the definitions of minimal models and minimal entailment in circumscription theory. (5 marks)

*Answer.*

Minimal entailment considers not all models of the knowledge base but only those where the set of exceptions is made as small as possible. Namely, consider a predicate $Ab$ (for abnormal) and the formulation of a default rule as

$$\forall x (Bird(x) \wedge \neg Ab(x) \supset Flies(x))$$

and say that a conclusion follows by default if it is entailed on all interpretations where the extension of $Ab$ is as small as possible. (This is called *circumscribing* $Ab$ and the approach is called *circumscription*.) We need one $Ab$ for every default rule, because a bird which is abnormal with respect to flying may be normal with respect to having two legs etc.

Let $A$ be the set of $Ab$ predicates we want to minimise. Let $M_1 = (D, I_1)$ and $M_2 = (D, I_2)$ be two interpretations over the same domain such that every constant and function are interpreted the same way.

$$M_1 \leq M_2 \;\Leftrightarrow\; \forall Ab \in A\ (I_1(Ab) \subseteq I_2(Ab))$$

$M_1 < M_2$ if $M_1 \leq M_2$ but not $M_2 \leq M_1$. (There are strictly fewer abnormal things in $M_1$). *Minimal entailment:* $KB \models_{\leq} \phi$ iff for all interpretations $M$ which make $KB$ true, either $M \models \phi$ or $M$ is not minimal (there exists an $M'$ such that $M' < M$ and $M' \models KB$).

(d) For the following KB:

$KB = \{$
$\mathtt{NorthOf}(\mathtt{milan}, \mathtt{glasgow})$,
$\mathtt{NorthOf}(\mathtt{milan}, \mathtt{london})$,
$\mathtt{NorthOf}(\mathtt{milan}, \mathtt{moscow})$,
$\mathtt{glasgow} \neq \mathtt{london}$,
$\mathtt{london} \neq \mathtt{moscow}$,
$\mathtt{glasgow} \neq \mathtt{moscow}$,
$\neg\mathtt{ColderThan}(\mathtt{milan}, \mathtt{glasgow}) \vee \neg\mathtt{ColderThan}(\mathtt{milan}, \mathtt{london})$,
$\forall x(\mathtt{NorthOf}(\mathtt{milan}, x) \wedge \neg\mathtt{Ab}(x) \supset \mathtt{ColderThan}(\mathtt{milan}, x))$
$\}$

state whether the sentence $\mathtt{ColderThan}(\mathtt{milan}, \mathtt{moscow})$ is minimally entailed, and explain why. (5 marks)

*Answer.* The minimal models of KB are those where the extension $I(Ab)$ of $Ab$ is either $\{I(\mathtt{glasgow})\}$ or $\{I(\mathtt{london})\}$, but not both. In all of them $(I(\mathtt{milan}), I(\mathtt{moscow}))$ is in $I(\mathtt{NorthOf})$ and $I(\mathtt{moscow})$ is not in $I(Ab)$, so $(I(\mathtt{milan}), I(\mathtt{moscow}))$ is in $I(\mathtt{ColderThan})$.

6. (a) Describe the development cycle of a knowledge-based system. (10 marks)

*Answer* Development cycle of a knowledge based system:

```
                   1. Plan knowledge base
                          |
                          v
                   2. Select domain experts
                   and knowledge sources
                          |
                          v
                   3. Elicit knowledge
                          |
                          v
                   4. Formulate and represent
                   knowledge
Continue                  |
knowledge                 v
acquisition        5. Implement knowledge base
                          |
    ^                     v
    |              6. Test knowledge base
    |                     |
    |       no            v
    -------------------- OK?
                          | yes
                          v
                   7. Systems test
```

In 1, the content of the knowledge base, relevant inputs and outputs, strategy for testing, knowledge dictionary, concepts etc. are identified.

In 2, sources of knowledge are selected (domain experts, corpora, etc.).

In 3, the process of knowledge elicitation takes place.

In 4, the knowledge is formulated in the form suitable for inference.

In 5, it is encoded in machine-readable form. [A complete answer (listing all relevant stages) gets full marks, pro-rata for partial answers.]

(b) Describe the process of knowledge acquisition for a production rule system using decision tables. Show a decision table for a production rule system making medical insurance reimbursement decisions based on the following policy:

*No charges are reimbursed to the patient until the deductible has been met. After the deductible has been met, the amount to be reimbursed depends on whether or not the doctor or hospital is a "Preferred Provider". For preferred providers Doctor's office visits are reimbursed at 65% and Hospital visits are reimbursed at 95%. For other providers reimburse 50% for Doctor's Office visits or 80% for Hospital visits.*
[Example from `http://web.sxu.edu/rogers/sys/decision_tables.html`.]

A deductible is met if the claim amount is larger than some fixed value, and for each case the deductible is either met or is not. (15 marks)

*Answer*

First we need to decide on all conditions relevant for making a decision, and all possible actions. A decision table consists of a list of condition stubs (possible relevant conditions), action stubs (progress), condition entries (yes, no, and - for not relevant, for each condition ) and action entries (which action is to be taken if specified conditions hold).

The relevant conditions are:

i. whether deductible has been met (possible values Y,N)
ii. whether the provider is preferred (possible values Y,N)
iii. type of visit (D for doctor, H for hospital).

The possible actions are:

i. reimburse 65%
ii. reimburse 95%
iii. reimburse 50%
iv. reimburse 80%
v. no reimbursement

The resulting table is

```
Conditions
Deductible met?          Y   Y   Y   Y  N
Preferred Provider?      Y   Y   N   N  -
Type of visit            D   H   D   H  -

Actions
Reimburse 65/%           X
Reimburse 95/%               X
Reimburse 50/%                   X
Reimburse 80/%                       X
No reimbursement                        X
```

The rules correspond to columns in the table, for example (if deductible met and preferred provider and doctor, then reimburse 65%).

[Approximately 7 marks for the general description of the procedure, and 8 for the decision table.]