UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2002

MEng Honours Degrees in Computing Part IV
MSc in Advanced Computing
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute

PAPER C491

KNOWLEDGE REPRESENTATION

Friday 26 April 2002, 10:00
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators not required

1 Th(X) denotes the classical truth-functional consequences of a set of formulas X. $X \vdash \alpha$ is shorthand for $\alpha \in \mathrm{Th}(X)$.

a In general terms the content of a logic database is a set D of formulas of some language $\mathcal{L}$, together with a notion of consequence, represented by the consequence operator Cn(D).

i) What are the three possible answers for a query α to such a database?

ii) Three different definitions of integrity constraint satisfaction are common:

- 'consistency',
- 'entailment' or 'theoremhood',
- 'metalevel' or 'epistemic'.

Express each of these definitions in terms of Cn(D). Use the notation $\alpha \Rightarrow \beta$ for the metalevel/epistemic constraint 'if α is in the database then β is in the database.'

iii) By reference to the databases Th($\{a \vee b\}$), Th($\{\neg a\}$), Th($\{a\}$), demonstrate that the satisfaction definition for a metalevel/epistemic constraint $\alpha \Rightarrow \beta$ is not the same as the 'consistency' or 'entailment' definitions of satisfaction of the constraint $\alpha \rightarrow \beta$.
$\alpha \rightarrow \beta$ is material implication ($\neg\alpha \vee \beta$).

b Let D be a *normal logic program.* Consider the consequence operator Cn_{NBF} defined as follows:

$$\mathrm{Cn}_{NBF}(D) = \mathrm{Th}(\,comp(D)\,)$$

where $comp(D)$ is the Clark completion of the program D.
Show that the consequence operator Cn_{NBF}:

i) has the property of 'inclusion',

ii) is 'supraclassical', i.e., satisfies $\mathrm{Th}(D) \subseteq \mathrm{Cn}_{NBF}(D)$,

iii) has the property of 'left absorption', which for any consequence operator Cn is defined as follows:

$$\mathrm{Th}(\mathrm{Cn}(A)) = \mathrm{Cn}(A).$$

iv) By means of a simple example, or otherwise, show that Cn_{NBF} is non-monotonic.

c Let Cn be any consequence operator that satisfies 'left absorption' defined in part b(iii). $A \mid\!\sim \alpha$ is shorthand for $\alpha \in \mathrm{Cn}(A)$. Show that the following hold.

i) If $B \subseteq \mathrm{Cn}(A)$ and $C \subseteq \mathrm{Th}(B)$ then $C \subseteq \mathrm{Cn}(A)$.

ii) $A \mid\!\sim \alpha$ and $A \mid\!\sim \beta$ iff $A \mid\!\sim (\alpha \wedge \beta)$.

iii) Hence, or otherwise: if $(\alpha \rightarrow \beta) \in D$, then $\mathrm{Cn}_{NBF}(D)$ satisfies the metalevel/epistemic integrity constraint $\alpha \Rightarrow \beta$.

The four parts carry, respectively, 30%, 35%, 35% of the marks.

2a i) Define the *answer set* semantics of an *extended logic program.*

ii) Explain how an extended logic program can be translated to a *normal* logic program, and state (without proof) in what sense the two programs are equivalent.

iii) Write down the normal logic program obtained by translating the following extended logic program P:

$$
\begin{aligned}
&h(x) \leftarrow l(x), not\, \neg h(x)\\
&\neg h(x) \leftarrow t(x,y), \neg h(y)\\
&s(x) \leftarrow l(x), not\, d(x)\\
&d(x) \leftarrow t(x,y), not\, h(x)\\
&l(a)\\
&l(b)\\
&t(a,b)\\
&t(a,c)\\
&\neg h(a)
\end{aligned}
$$

b i) The extended logic program P in part (a) has exactly one answer set. Why? You may state standard results without proof.

ii) Hence or otherwise, compute the answer set for P.

iii) Hence, what are the answers to the following three queries on the extended logic program P?

$$h(a)?, \quad h(b)?, \quad h(c)?$$

c In Answer Set Programming (ASP), solutions (answer sets) containing the atom r can be eliminated by adding the following 'constraint' to the program:

$$p \leftarrow not\, p, r$$

where p is a new atom not appearing elsewhere in the program.

Explain why this device eliminates solutions containing r in the case of *normal* logic programs, where the answer sets are *stable models.*

Hint: Consider two cases: a stable model of the form $\{r, p, \ldots\}$ and a stable model of the form $\{r, \ldots\}$ (no p).

The three parts carry, respectively, 40%, 40%, 20% of the marks.

3a Formulate the following as a Reiter default theory:

- Students are typically lazy and poor.
- Postgraduate students (PG students for short) are a type of student, and are typically not lazy.
- Business students (BS students for short) are typically PG students and are typically not poor.

The defaults should be read so as to give the following conclusions:

- a student, Alan, is lazy and poor;
- a PG student, Bill, is not lazy and poor;
- Colin, a BS student who is not a PG student, is lazy and not poor.

Demonstrate that your formulation does give these conclusions. (It is not necessary to show the detailed computation of the extensions.)

b Define what is meant by a *stable expansion* in autoepistemic logic.

State, without proof, how Reiter default theories can be translated into autoepistemic logic. In what sense are the two formulations equivalent?

Write down the translation into autoepistemic logic of the default theory from part (a).

c Derek is either a PG student, or a BS student who is not a PG student. What conclusion does the default theory from part (a) reach in this case?

How does the autoepistemic formulation of part (b) deal with this case?

The three parts carry, respectively, 50%, 30%, 20% of the marks.

4 Consider the following variant YSP_{aim} of the 'Yale Shooting Problem' (YSP) scenario. There are actions of loading a gun, aiming the gun at a person X, who then becomes the 'target', and shooting (i.e., pulling the trigger). Shooting a loaded gun when it is aimed at X results in X being dead (not alive).

a Formulate the YSP_{aim} scenario in the event calculus.
Include the general event calculus axioms for *holds_at* in your answer.

b Formulate the YSP_{aim} scenario as an action description in the language $\mathcal{C}/\mathcal{C}+$.
In order to illustrate the use of multi-valued fluents, let the result of aiming the gun at X result in the fluent $target = X$ being true. In order to demonstrate how 'statically determined' fluents are treated, let the action of shooting a loaded gun at the target X result in fluent $shot(X)$ being true, and then define $\neg alive(X)$ in terms of $shot(X)$.
Some further constraints: a gun cannot be loaded and aimed at the same time; a gun cannot be loaded and shot at the same time.

c Describe how the 'Causal Calculator' CCALC is used to perform computations on action descriptions in $\mathcal{C}/\mathcal{C}+$.
Structure your answer as follows:

i) How is an action description in $\mathcal{C}/\mathcal{C}+$ translated to a 'causal theory'?

ii) What is a *model* of a causal theory Γ?

iii) What is the *literal completion* of a definite causal theory Γ and how is it related to the models of Γ? There is no need to define the term 'definite'.

iv) How does one formulate a query to determine whether the sequence of actions *load*, *aim at* x_1, *aim at* x_2, *shoot*, *shoot* results in fluent *alive*(x_2) being false when executed in an initial state where the gun is not loaded?
How does the 'Causal Calculator' CCALC obtain answers to the query?
(It is not necessary to compute the answers.)

Concentrate on explaining the main ideas. Mistakes in points of detail will not be penalized.

The three parts carry, respectively, 25%, 25%, 50% of the marks.