UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2004

MSc in Advanced Computing
for Internal Students of the Imperial College of Science, Technology and Medicine

PAPER C329

COMPUTATIONAL LOGIC

Friday 14 May 2004, 14:30
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators not required

1 Given a first-order sentence **S**, **gcl(S)** denotes a clause-set obtained by converting **S** to general clausal form in the standard way.

a Let **S** and **T** be first-order sentences. If $\mathbf{S} \models \mathbf{T}$ then this can be shown by a resolution refutation. Using the **gcl** notation, express the clause-set from which such a refutation would be derived.

b For the following case, in which $\mathbf{S} \models \mathbf{T}$ holds,

$\mathbf{S} = (\forall X \forall Y)(\text{path}(X, Y) \text{ iff } (\text{go}(Y) \text{ if go}(X)))$
$\mathbf{T} = (\forall X \forall Y \forall Z)(\text{path}(X, Z) \text{ if path}(X, Y), \text{path}(Y, Z))$

i) construct the appropriate clause-set for showing $\mathbf{S} \models \mathbf{T}$ by refutation (you are not required to show the conversion details);

ii) construct the refutation, taking care to identify the parents in each step.

Parts a, b(i) and b(ii) carry, respectively, 10%, 40% and 50% of the marks.

2a For a definite program **P**, define the $T_{\mathbf{P}}$ function and explain how the limits $T_{\mathbf{P}} \uparrow \omega$ and $T_{\mathbf{P}} \downarrow \omega$ are used to determine which atoms in **B(P)** succeed or fail finitely.

b For the following program **P** on domain **H** = {chris, bob, logic}

likes(chris, X) if likes(X, logic)
likes(bob, logic)

construct $T_{\mathbf{P}} \uparrow \omega$ and explain why $T_{\mathbf{P}} \downarrow \omega$ must be identical to it.

c For the same program **P**, identify

i) the smallest reflexive **H**-model;

ii) the largest anti-symmetric **H**-model.

Note: a reflexive model M is such that, for any X in **H**, likes(X, X) is in M;

an anti-symmetric model M is such that, for any X, Y, if likes(X, Y) and likes(Y, X) are both in M then X=Y.

The three parts carry, respectively, 15%, 15% and 70% of the marks.

3a Describe how, in general, one constructs the local dependency graph for a normal-clause program. Sketch the graph for the following program **P** on domain **H** = {a, f(a), f(f(a)), ...}:

p(X) if fail q(a)
q(a) if fail p(f(X))

b Use this graph to determine, with explanation, whether **P** is locally stratified and whether it is locally call-consistent. What would such properties (if they held) imply about **P** having a unique stable model and about **comp(P)** having a model?

c Show that {q(a)} is a stable model for **P** and is also a model for **comp(P)**.

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

4 The following interpreter **V**

demo(*true*)
demo(X) if clause(X, Y), demo(Y)
demo(X∧Y) if demo(X), demo(Y)

can be used to evaluate by SLD a given conjunction of atomic calls with respect to a given definite program **P**, where **P** contains no clauses defining demo.
Note — clause(X, Y) succeeds for atom X if **P** contains a clause X if Y; if that clause has an empty body then Y is returned as *true*.

a Justify carefully the following two statements:

given **P**, X and Y,
?- demo(X∧Y) succeeds using **V** iff ?- demo(Y∧X) succeeds using **V**

given **P** and X,
?- demo(X∧X) succeeds using **V** iff ?- demo(X) succeeds using **V**

b **V** can be extended to form a new interpreter **M** able to evaluate a conjunction of atomic or negated calls with respect to a given program **P** in which clause bodies may contain atomic or negated calls. The extension adds the clause

demo(¬X) if undemo(X)

together with a set **U** of definite clauses defining undemo(X), whose intended meaning is "demo(X) does not succeed using **M**". The given program **P** is assumed to contain no clauses defining demo or undemo.

Write down a suitable definition **U** of undemo, using only the predicate symbols undemo, demo, forall and clause.
Note — forall(A, B) succeeds when every way of solving A also solves B.

c Show, by sketching the evaluation tree, a case in which ?- demo(q) does not succeed using **M** when given a program **P** that logically implies q.

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