UNIVERSITY OF LONDON

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2001

BEng Honours Degree in Computing Part III
BSc Honours Degree in Mathematics and Computer Science Part III
MSci Honours Degree in Mathematics and Computer Science Part III
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
This paper is also taken for the relevant examinations for the Associateship of the Royal College of Science

PAPER C329

COMPUTATIONAL LOGIC

Tuesday 1 May 2001, 14:00
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators not required

1a Describe briefly but carefully the mechanism of an SLD inference step. Discuss the statement "the choice of computation rule influences only the efficiency of execution".

b Given a program consisting of the single clause

```
p(X, f(X)) :- p(f(X), X).
```

discuss the evaluations by Prolog of the queries ?p(a, V) and ?p(U, V).

c A definite program is written in a language **L** whose Herbrand universe is **H** and whose clauses define a 2-ary relation q. The call q(X, Y) succeeds from this program without binding X or Y. How many ground calls q(...) written in **L** would also succeed? Justify your answer.

d A definite program contains just one clause defining a 2-ary relation p. The three SLD branches shown below all make use of this program.

```
¦                       ¦
? p(a, f(a))            ? p(b, f(c))            ¦
|                       |                       ? p(X, a)
? q(a), p(a, f(a))      ? q(c), p(a, f(b))      |
¦                       ¦                       ■
```

By considering these, infer the structure of the clause defining p.

e Sketch one successful computation (that is, one branch in the SLD tree) obtained by evaluating with Prolog the query

```
?append(_, X, [a, b, c]), append(_, X, [a, c]).
```

using the program

```
append([], Z, Z).
append([U | X], Y, [U | Z]) :- append(X, Y, Z).
```

and show how the computed answer substitution is formed in that branch.

The five parts carry, respectively, 20%, 20%, 10%, 30% and 20% of the marks.

2a Describe carefully the safe SLDNF inference system.

b Explain how the completion **comp(P)** of a normal program **P** is constructed, and state how it relates to the answers computable from **P** using safe SLDNF inference.

c Given the following normal program **Q**

```
resigns(X)  :-  fail credible(X).
credible(X)  :-  recalls(X, Z), fail doubts(Y, Z).

recalls(peter, phonecall).
doubts(alistair, phonecall).
doubts(tony, X)  :-  doubts(alistair, X).
```

i) construct **comp(Q)** and determine (showing your reasoning) whether it implies `resigns(peter)`.
Note—the equality theory need not be written down explicitly, and the Herbrand domain is {peter, alistair, tony, phonecall}.

ii) sketch the safe and unsafe SLDNF evaluations of the query `?resigns(peter)` and comment upon the results obtained.

The three parts carry, respectively, 15%, 20% and 65% of the marks.

3a Define precisely the following terms in respect of a clausal-form program **P**:

i) a Herbrand interpretation
ii) a Herbrand model
iii) a minimal Herbrand model

b Show that if **I** and **J** are any two Herbrand models for a definite program then so is their intersection.

c The *immediate consequence function* T_P for a definite program **P** is defined by

$$T_P(\mathbf{I}) = \{ q \mid (q \text{ if body}) \in \mathbf{G}(\mathbf{P}),\ \text{body} \subseteq \mathbf{I} \}$$

Show that **I** is a Herbrand model of **P** if and only if $T_P(\mathbf{I}) \subseteq \mathbf{I}$.

d The program **P** below is written in a language whose Herbrand domain **H** comprises just a, b, [] and all the lists constructible from them:

```
atom(a).
atom(b).

end([U], U)  :-  atom(U).
end([U | X], U)  :-  end(X, U).
```

Show all the iterates $T_P{}^k(\phi)$ up to k=3. Explain how they approximate the meaning of **P** as characterized by the least fixpoint of T_P.

The four parts carry, respectively, 20%, 30%, 15% and 35% of the marks.

4a Compose a meta-program **M** which, when used by Prolog to evaluate a query ?demo(**P**, **Q**), simulates Prolog's evaluation of the query ?**Q** using the program named **P**. **M** accesses **P** via the predicate has_clause(**P**, Head, Body) which holds when **P** contains a clause unifying with Head :- Body. Allow calls in **Q** and **P** to be either positive or negative. Represent a negative call by (fail C) where C is the atomic part of the call.

Hint —only three demo clauses are needed in **M**: one for the case that **Q** is empty, one for the case where **Q**'s first call is positive and one for the case where **Q**'s first call is negative.

b Explain the connection between this statement about the subset relation

S: $(\forall X \forall Y)(\text{subset}(X, Y)$ if $(\forall U)(U \in Y$ if $U \in X))$

and this normal program

```
subset(X, Y) :- fail e(X, Y).
e(X, Y) :- mem(U, X), fail mem(U, Y).
```

The statement **S** implies subset(X, X) for any set X, including the infinite set n={0, s(0), s(s(0)), ...}. However, the normal program extended with

```
mem(0, n).
mem(s(U), n) :- mem(U, n).
```

cannot solve the query ?subset(n, n). Explain why this is so.

c Modify your meta-program from part a so that it can overcome the weakness that prevented it, in part b, from being able to solve ?subset(n, n).

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