UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1997

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
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 Royal College of Science
Associateship of the City and Guilds of London Institute

PAPER 3.27

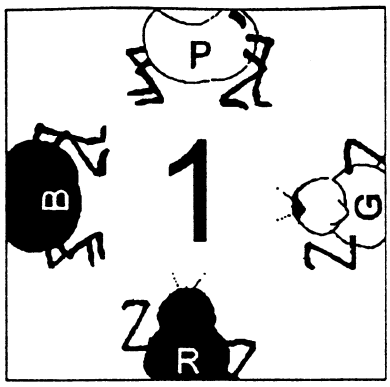
LOGIC PROGRAMMING - APPLICATIONS

Monday, April 28th 1997, 2.00 - 4.00

Answer THREE questions

For admin. only: paper contains 4 questions

1 **The Bug Puzzle comprises four square cards, numbered 1 to 4. Along each side of a card is a picture of either a bug's head or its body. Bugs are coloured blue, green, pink or red.**



Reading along the sides of each of the cards, in a clockwise direction starting at the top, they contain the following pictures:

Card 1:	pink body,	green head,	red head,	blue body.
Card 2:	green head,	pink head,	blue body,	red body.
Card 3:	blue head,	pink head,	green body,	pink body.
Card 4:	pink head,	green body,	red body,	blue head.

The aim of the puzzle is to orient the cards and place them in a 2 x 2 array, so that along each common edge between two cards a bug's head on one card fits onto a body of the same colour on the other card.

The solution to the puzzle can be expressed as a list:

[[C1, R1], [C2, R2], [C3, R3], [C4, R4]]

where C1 to C4 are the numbers of the cards that form the array:

C1 C2

C3 C4

and R1 to R4 are the number of right-angles (0, 1, 2 or 3) through which cards C1 to C4 need to be rotated in an anti-clockwise direction from the upright.

a Write the Prolog code for a program that implements the predicate

```
image(C, R, S, P)
```

meaning that when card C has been rotated through R right-angles, picture P appears on side S. Assume that S can take values n, e, s or w (representing the four main compass directions).

b Write the Prolog code for

```
matches(P1, P2)
```

meaning that picture P1 and picture P2 fit together.

c Write a program in ElipSys for finding a solution to the bug problem.

The three parts carry, respectively, 40%, 10%, 50% of the marks.

2a Write a Prolog program for the predicate

```
insert(A, L1, L2)
```

meaning that ordered list L2 results from inserting element A into ordered list L1 that does not already contain A. You may assume that the binary infix operator @< is able to check whether one term is less than another.

b Using part (i), write a Prolog program for the predicate

```
union(L1, L2, L3)
```

meaning that L3 is the ordered list that contains precisely the elements of the two disjoint ordered lists L1 and L2.

c Write a Prolog program for the predicate

```
delete(X, L1, L2)
```

meaning that list L2 results from deleting the sole occurrence of element X in list L1.

d Using parts a, b and c, write a Prolog program for

```
coalesce(P, C)
```

which takes a list P of pairs of equivalent elements and produces a list C of ordered equivalence classes.

For example, the query

```
?- coalesce([[e,a],[b,d],[c,d],[e,g],[s,r],[g,b],[p,r]], C).
```

should result in the answer

```
C = [[a, b, c, d, e, g], [p, r, s]]
```

The four parts carry, respectively, 10%, 10%, 10%, 70% of the marks.

Turn over ...

3 A particular expert systems shell deals with propositional facts and rules, i.e., ones that do not contain variables. Each fact and rule has a certainty factor (CNF) between 0 and 100. A CNF of 100 means definitely true, whilst 0 means definitely false.

Rules contain conjunctions of conditions but not disjunctions, although there may be several rules with the same conclusion. In a rule, a condition may be a negation, but the conclusion may not be.

Examples of a fact with a CNF of 60 and a rule with a CNF of 70 are

```
good_reference CNF 60.

IF good_reference
    AND smart_appearance
    AND NOT criminal_conviction
THEN offer_job CNF 70
```

The inference engine calculates the CNF of a conclusion by combining the CNFs of conditions and rules.

The CNF of the overall premise (i.e., the conditions taken as a whole) of a rule is calculated as the **lowest** CNF of the conditions.

The CNF of the concusion of the rule is computed by multiplying the CNF of the overall premise by the CNF of the rule itself (and then dividing by 100 to normalise the result).

If several rules reach the same conclusion, then the combined CNF of that conclusion is taken to be the **highest** CNF computed for the conclusions of those individual rules.

The CNF of a negated condition "NOT cond" is taken to be 100 - CNF1 , where CNF1 is the CNF of condition "cond". The inference engine does not employ negation as failure.

a Show how the example fact and rule and their CNFs can be written in Prolog.

b Write a Prolog program for the predicate

```
lower(X, Y, Z)
```

meaning that number Z is the lower of numbers X and Y.

c Write a Prolog program for the predicate

```
max(L, M)
```

meaning that number M is the greatest of all the elements in the non-empty list of non-negative numbers L.

d Write the top-level "demo" Prolog code for the expert system's inference engine, that carries out the computation of CNFs. You need not be concerned with explanatory facilities. However, you should include the option of querying the user for his CNF for information not derivable from the knowledge base.

e Write the lower-level Prolog code that carries out interaction with the user.

The five parts carry, respectively, 20%, 10%, 10%, 50%, 10% of the marks.

4a Write a tail-recursive program in Prolog that implements the predicate

```
fact1(N, F)
```

meaning that F is the factorial of non-negative integer N. The program should be able to calculate F when given N.

b Explain what happens when the query

```
?- fact1(N, 6)
```

is posed in Prolog.

c Write a program in CLP(R) that implements the predicate

```
fact2(N, F)
```

again meaning that F is the factorial of non-negative integer N. This time, the program should be able not only to calculate F when given N, but also to find N when given an exact factorial number F.

d Carefully describe the execution steps involved when the query

```
?- fact2(N, 6)
```

is posed in CLP(R).

End of paper