UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2000

BEng Honours Degree in Computing Part III
BEng Honours Degree in Mathematics and Computer Science Part III
MEng 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 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 C327

THE PRACTICE OF LOGIC PROGRAMMING

Thursday 4 May 2000, 10:00
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions

1 a i) Write a Prolog program for: `posint(I)`

for generating each of the positive integers, I, in turn, from 1 upwards.

ii) Write a Prolog program for: `integer(J, K, I)`

for generating each of the positive integers I, from given integer J to given integer K inclusive, where $J \leq K$.

iii) Write a Prolog program for: `intlist(J, K, L)`

for generating the list, L, of the positive integers running from given integer J to given integer K inclusive, where $J \leq K$.

b Positive integers X, Y and Z form a Pythagorean Triple if $X^2 + Y^2 = Z^2$.

Write a Prolog program for: `pythagorean(X, Y, Z)`

for generating all the sets of Pythagorean triples.

c Write a Prolog program for: `modulo(X, Y, Z)`

meaning that Z is the remainder when integer X is divided by integer Y. It should be possible to use your program both for checking and for finding Z when given two integers X and Y. *Do **NOT** use Prolog's built-in mod operator.*

d The Sieve of Erastosthenes generates all prime numbers up to a given integer N.

It works as follows:
1. Put all the numbers between 2 and N into the "sieve" (i.e., a list).
2. Select and remove the smallest number remaining in the sieve.
3. Include this number in the primes.
4. Step through the sieve, removing all multiples of this number.
5. If the sieve is not empty, repeat steps 2 to 5.

Write a Prolog program for: `primes(N, P)`

for generating the list P of all prime up to integer N, using this method.

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

2a Describe how Prolog's built-in predicate `Clause` behaves.

b i) Write a Prolog program for: `conjList(C, L)`

that is able to translate between a conjunction, C, of items and a list, L, of the same items.

ii) Write a Prolog Program for: condL(Conds, N)

that finds the number, N, of a clause's conditions, Conds. You may assume that Conds is in the standard representation of a clause's conditions.

iii) Write a Prolog program for: myClause(Goal, N, Conds)

that finds a clause in the workspace with N conditions, Conds, whose conclusion matches Goal.

iv) Write a Prolog program for: condNumbers(Goal, L)

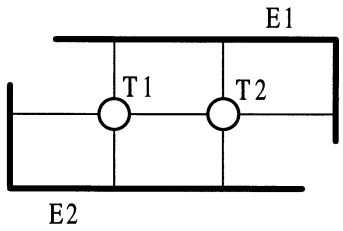
that generates the list, L, of the numbers of the conditions of all the clauses in the workspace whose conclusion matches Goal. List L is to be in increasing order and without duplicates. (You may use Prolog's built-in predicate `Sort(L1, L2)` that sorts a list and removes duplicates.)

c Write the top-level `demo` code for a Prolog interpreter that tries matching clauses in ascending order of the number of conditions in the clause. (It will try facts first). You may ignore explanatory and interactive features.

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

3 Provided you know the boundary point temperatures you can approximate the steady-state temperatures of points in the interior of a metal plate using the finite-elements method. You divide the sheet into a grid of discrete points and take the temperature at each interior point to be the average of the temperatures at the point's four orthogonal neighbours (i.e., the nearest points above, left, right and below). This produces a set of simultaneous equations whose solutions represent the interior point temperatures.

3a Consider the small plate below, with two interior points and two boundary strips with uniform temperatures E1 and E2:



i) Write a simple clause in CLP(R) for: `solve(E1, E2, T1, T2)`

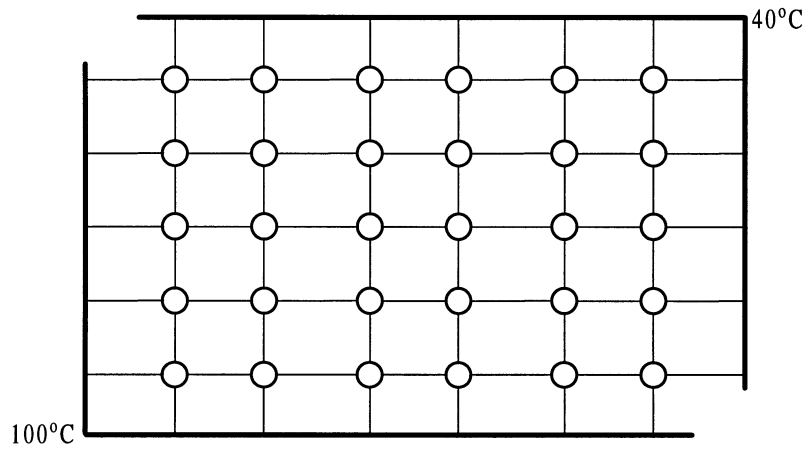
that relates the interior and boundary point temperatures.

ii) Describe how CLP(R) handles the query:

```
?- solve(40, 100, X, Y).
```

iii) What would happen if the same query were posed in Prolog?

b With larger plates divided into grids with far more interior points it is no longer practical to list out explicitly a constraint for each point.



i) How could you represent the above plate in CLP(R). (Only the boundary point temperatures are known).

ii) Define a CLP(R) procedure: `solveRow(Above, Row, Below)`

which expresses how the temperatures of the points along a row are related to those of the points on the rows immediately above and below it.

iii) Define a CLP(R) procedure: `solvePlate(Plate)`

which generates the values of the temperatures of all the interior points, given only the boundary point temperatures.

The six parts and subparts carry, respectively, 10%, 15%, 15%, 10%, 25%, 25% of the marks.

4a Describe the two types of consistency checking technique employed in CLP(FD).

b What is the role of labeling in CLP(FD)?

c Write a CLP(FD) program for: `sumList(L, S)`

meaning that S is the sum of a non-empty list, L, of integers.

d Fleets of battlecruisers from Jupiter and Mars are engaged in battle just off asteroid Zog. There are eight sectors around Zog and, at present, there are three battlecruisers in each sector.

- Sector 1 contains two Jupiter battlecruisers.
- Each side has an equal number of battlecruisers.
- Sector 6 contains battlecruisers from both Jupiter and Mars.
- There are more battlecruisers from Mars than from Jupiter in sector 7.
- Sector 2 holds two fewer Jupiter battlecruisers than Sector 1.
- The balance of forces in Sectors 5 and 4 are identical.
- There are an equal number of battlecruisers from both sides in Sectors 1 plus 7.
- There are two Mars battlecruisers in Sector 8.
- There are two Jupiter battlecruisers in sector 3.
- There are twice as many Jupiter battlecruisers as those from Mars in Sectors 3 plus 6.

i) Write a program in CLP(FD) to determine the number of battlecruisers from each side in each of the eight sectors. *You do **NOT** have to solve the problem yourself - simply formulate it in CLP(FD).*

ii) To win the battle one side must have three battlecruisers in some sector. Modify your program from part i) so that it indicates whether Jupiter has won outright, Mars has won outright, whether there is a draw (with each side having a sector with three of their battlecruisers) or whether the battle is unfinished (with neither side having three battlecruisers in any sector).

The five parts and subparts carry, respectively, 20%, 15%, 15%, 30%, 20% of the marks.