Coláiste na Tríonóide, Baile Átha Cliath
Trinity College Dublin
Ollscoil Átha Cliath | The University of Dublin

Faculty of Engineering, Mathematics and Science

School of Computer Science & Statistics

Integrated Computer Science Programme
B.A. (Mod.) Computer Science & Business
B.A. (Mod.) Computer Science & Language
Mathematics
Year 3 Annual Examinations

Trinity Term 2018

Symbolic Programming

Wed, 16 May 2018 EXAM HALL 14:00 - 16:00

Dr Tim Fernando

Instructions to Candidates:

Attempt *two* questions. All questions carry equal marks. Each question is scored out of a total of 50 marks.

You may not start this examination until you are instructed to do so by the Invigilator.

Exam paper is not to be removed from the venue.

Materials permitted for this examination:

Non-programmable calculators are permitted for this examination — please indicate the make and model of your calculator on each answer book used.

1. (a) State how a Prolog interpreter responds to the following queries, assuming no Prolog program has been consulted.

(i) `?- X = 1+1.` X = 1+1.

(ii) `?- X is 1+1.` X = 2.

(iii) `?- love(john,mary).` Error. (love has not been defined yet)

(iv) `?- assert(love(john,mary)).` true.

(v) `?- .([],.(a,Y)) = [X,a].` Y = X, X = [].

(vi) `?- setof(X,X=X,L).` L = [x]

(vii) `?- findall(X,X=f(X),L).` L = [X] where X = f(x)

(viii) `?- X=1,X<2.` X=1

(ix) `?- X\=2,X=1.` False

(x) `?- X < 2.` Arguments not sufficiently instantiated.

[20 marks]

(b) Suppose we have the following Prolog program.

```
q(a).
q(X) :- X=b,!.
q(c).
```

Write all of Prolog's answers to the following queries.

(i) `?- q(X).` X = a ;
X = b.

X = Y, Y = a ;
X = a, Y = b ;
X = b, Y = a ;
X = Y, Y = b.

(ii) `?- q(X), q(Y).`

(iii) `?- q(X),!,q(Y).`

X = Y, Y = a ;
X = a,
Y = b.

(iv) `?- q(c).` true.

[8 marks]

(c) Given unary predicates `bird`, `fly`, `penguin`, write a Prolog program saying all birds fly except for penguins (which do not). fly(X) :- bird(X), \+ penguin(X).

[10 marks]

(d) Recall that positive integers can be encoded as successors of 0 (with, for example, `succ(succ(0))` encoding 2), and similarly, negative integers can be encoded as predecessors of 0 (with, for example, pred(pred(0)) encoding −2).

```
pos(succ(0)).
pos(succ(X)) :- pos(X).

neg(pred(0)).
neg(pred(X)) :- neg(X).
```

These definitions suggest two ways to represent all integers, given by the predicates pure and mixed below.

```
pure(0).
pure(X) :- pos(X); neg(X).

mixed(0).
mixed(succ(X)):- mixed(X).
mixed(pred(X)):- mixed(X).
```

(i) Give a term t without variables such that Prolog answers yes/true to the following query

?- mixed(t), \+ pure(t).

[2 marks]

(ii) State Prolog's response to the query

```
?- mixed(X), \+ pure(X).
```

[2 marks]

(iii) Define a binary predicate convert(Mixed,Pure) that converts a mixed representation to its pure counterpart (representing the same integer).

[8 marks]

2. (a) Define the binary predicate `member(X,L)` that is true when `X` is a member of the list `L`.

member(X, [H|T]) :- X == H; member(X, T).

State how Prolog responds to the query

`?- member(X,[Y]).`

False because X is not in the list containing only Y. i.e. X /= Y

[6 marks]

(b) Define the binary predicate `nonmember(X,L)` that is true when `X` is not a member of the list `L`.

nonmember(_, []).
nonmember(X, [H|T]) :- X \= H, nonmember(X, T).

State how Prolog responds to the query

`?- nonmember(X,[Y]).`

It answers false, not sure why.

[7 marks]

(c) Define the 3-ary predicate `diff(X,L1,L2)` that is true when `X` is a member of `L1` but not a member of the list `L2`.

diff(X, L1, L2) :- member(X, L1),
nonmember(X,L2).

State how Prolog responds to the query

`?- diff(X,L,L).`

Error.

[7 marks]

(d) Define a 4-ary predicate `sublist(+L,+Begin,+End,?SubL)` that given a list `L`, and positive integers `Begin` and `End` returns the list `SubL` consisting of the members of `L` between list positions `Begin` and `End`, both included.

d

[15 marks]

(e) The *n*th *Harmonic number* H_n is the sum

$$\sum_{k=1}^{n} \frac{1}{k} = 1 + \cdots + \frac{1}{n}$$

(e.g. $H_1 = 1, H_2 = \frac{3}{2}$). Define a binary predicate `harmonic(+N,?H)` that given a positive integer N sets H to the Nth Harmonic number. For full marks, be sure to memoize (i.e., store results of computations for reuse).

[15 marks]

3. (a) What are difference lists and how are they useful.

[10 marks]

(b) Define a Definite Clause Grammar (DCG) for the set of strings $a^n b^{n+m} c^m$ of length $2n + 2m$ for $n, m \geq 0$. For full marks, avoid the use of extra arguments in the DCG.

[15 marks]

(c) Write out the DCG you have in part (b) as ordinary Prolog clauses, making the difference lists explicit.

[10 marks]

(d) Sharpen the DCG you have in part (a) to specify the length of the string (as an extra argument to the start predicate `s`) so that for example, for length 6, we can get all strings in this language of length 6 via the query

```
| ?- setof(L,s(6,L,[]),All6).
All6 = [[a,a,a,b,b,b],[a,a,b,b,b,c],[a,b,b,b,c,c],[b,b,b,c,c,c]]
```

[15 marks]