

## CM20214/CM20221 ADVANCED PROGRAMMING PRINCIPLES COURSEWORK 2

ALESSIO GUGLIELMI  
1 FEBRUARY 2016 (REVISED 7 MARCH 2016)

This is an individual assignment worth 20% of the total marks for the course. It is due by 5pm, 16 March 2016, and failure to submit by the deadline will result in a mark of 0. Extensions may only be granted by the Director of Studies.

### SPECIFICATION

Use Prolog to solve the following problem.

*Let  $X$  and  $Y$  be two integers with  $1 < X < Y$  and  $X + Y \leq 100$ . The mathematician  $S$  is given their sum  $X + Y$  and the mathematician  $P$  is given their product  $XY$ . The following conversation takes place:*

- (a) P: I do not know the two numbers.*
- (b) S: I knew you didn't know. I don't know either.*
- (c) P: Now I know the two numbers.*
- (d) S: Now I know the two numbers.*

*What are the numbers?*

The solution to the problem will be a Prolog program that accomplishes the following tasks:

- (1) The goal `s1(Q, 100)` will bind  $Q$  with a list of quadruples  $[X, Y, S, P]$ , where  $S = X + Y$  and  $P = XY$  and  $X$  and  $Y$  are possible solutions after sentence (a) is pronounced.
- (2) The goal `s2(Q, 100)` will bind  $Q$  with a list of quadruples  $[X, Y, S, P]$ , where  $S = X + Y$  and  $P = XY$  and  $X$  and  $Y$  are possible solutions after sentence (b) is pronounced.
- (3) The goal `s3(Q, 100)` will bind  $Q$  with a list of quadruples  $[X, Y, S, P]$ , where  $S = X + Y$  and  $P = XY$  and  $X$  and  $Y$  are possible solutions after sentence (c) is pronounced.
- (4) The goal `s4(Q, 100)` will bind  $Q$  with a value  $[[X, Y, S, P]]$ , where  $S = X + Y$  and  $P = XY$  and  $X$  and  $Y$  are the solution of the problem.
- (5) The goal `s4(Q, 500)` will bind  $Q$  with a list of quadruples  $[X, Y, S, P]$ , where  $S = X + Y$  and  $P = XY$  and  $X$  and  $Y$  are the solution of the problem above, when the constraint on the sum is changed into  $X + Y \leq 500$ .

The solution provided by the program must be mathematically correct for any value of the upper bound  $N$  of the sum – therefore, not only for  $N = 100$  or  $N = 500$ .

### ASSESSMENT CRITERIA

- The total of marks is 100, and each task contributes for at most 20 marks.
- The program must not contain any Prolog built-in functions and predicates, with the exception of arithmetic functions, comparison and assignment predicates, and the cut '!'. If built-ins are used, such as `findall`, `forall`, `bagof`, `\+`, `not`, `assert`, `retract`, `->`, etc., the final grade is decreased by 20. Parallel and multi-core processing are not allowed, under the same penalty.
- The grade for tasks 4 and 5 strongly depends on the number of inferences, as indicated by the time Prolog primitive.

- The program should not loop and should not admit multiple answers. ‘Multiple answers’ means getting further answers to any query by pressing the ‘;’ key. The penalty for loops or multiple answers is 30% of the final mark.
- Every predicate definition should be very briefly commented, in such a way that who marks can understand the idea without trying to interpret the code.
- No credit is given for partial solutions.
- The instructor might modify the grade assigned by the software, for example when the solution provided is not mathematically correct for every  $N$ .
- At the top of the program file there should be a comment **exactly like this**:

```
/* <username and name of the author>
THIS WORK IS ENTIRELY MY OWN.
The program does <or does not> produce multiple answers.
I have <or I have not, choose which is appropriate> used built-ins.
1. <Number of elements in the list binding Q after executing s1(Q,100)>
<At most 300 characters of clear text on the main idea for the definition of s1.>
2. <Number of elements in the list binding Q after executing s2(Q,100)>
<At most 300 characters of clear text on the main idea for the definition of s2.>
3. <Number of elements in the list binding Q after executing s3(Q,100)>
<At most 300 characters of clear text on the main idea for the definition of s3.>
4.
<At most 300 characters of clear text on the main idea for the definition of s4.>
5. <Number of elements in the list binding Q after executing s4(Q,500)>
s4(Q,500) uses <number> inferences. */
```

**Please adhere strictly to the structure above** (where `<username>` is your BUCS ID).

- At the bottom of the program file **there must be** a comment containing a capture from the terminal of the execution of the five goals required, including their running time, *i.e.*:

```
/*
?- consult(<username>).
true.
?- time(s1(Q,100)).
%<...> inferences, <...> CPU in <...> seconds (100% CPU, <...> Lips)
Q = [[3,4,7,12],[2,6,8,12], <rest of complete list of quadruples>]
?- time(s2(Q,100)).
Q = <...>
<...>
*/
```

If a complete capture is not possible (*e.g.*, limitations of the buffer) say so.

## HANDING IN

Upload on Moodle one **uncompressed** ASCII file named `<username>.p1`. SWI Prolog should be able to `consult` the file as is.

## FEEDBACK

Feedback will be sent by email within three weeks. Each solution will be assessed by software and feedback will be automatically generated in most cases. In special cases the instructor might provide feedback in person and modify the automatically generated grade. The marking software is made available on the web page of the course so that students can self-assess their Prolog program before submitting it.

**Please read very carefully the above instructions and adhere to them strictly.** The penalty for not following strictly the format suggested will be 10 marks.