CM20214/CM20221A ADVANCED PROGRAMMING PRINCIPLES COURSEWORK 2

ALESSIO GUGLIELMI
7 FEBRUARY 2014

This assignment is worth 20% of the total marks for the course. It is due by 5pm, 21 March 2014. Exceptions 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. The list will be ordered by ascending values of S.

(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. The list will be ordered by ascending values of P.

(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. The list will be ordered by ascending values of S.

(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 query is to be answered in less than ten minutes by an ordinary computer.

ASSESSMENT CRITERIA

- The total of marks is 100, and each task contributes for at most 20 marks.
- The maximum grade for each task can only be obtained if the relative part of the program does 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, \+, etc., the maximum grade for each task that employs them is 10. Parallel and multi-core processing are not allowed.
- 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 admit multiple answers, under penalty of 20 marks. 'Multiple answers' means getting further answers to any query by pressing the ';' key.
- 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.
- At the top of the program file there should be a comment **exactly like this**:

  /* *<username and name of the author>*

  `The program does` *<or* `does not`*> * `produce multiple answers.`

  `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.>`

  `I have` *(or* `I have not`*, choose which is appropriate)* `used built-ins.`

  `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.>`

  `I have` *(or* `I have not`*, choose which is appropriate)* `used built-ins.`

  `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.>`

  `I have` *(or* `I have not`*, choose which is appropriate)* `used built-ins.`

  `4.`

  <At most 300 characters of clear text on the main idea for the definition of `s4.>`

  `I have` *(or* `I have not`*, choose which is appropriate)* `used built-ins.`

  `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 complete capture from the terminal of the execution of the five goals required, including their running time, *i.e.*:

  `/*`

  `?- consult(`*<username>*`).`

  `%` *<username>* `compiled 0.00 sec,` *< · · · >* `bytes`

  `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 =` *< · · · >*

  < · · · >

  `*/`

Handing In

Upload on Moodle one uncompressed ASCII file named *<username>*`.pl`. 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. The instructor, in special cases, might modify the grade assigned by the software (this is where the program comments might be useful). The marking software will be made available so that students will be able to assess their Prolog program by themselves 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.