ICS1015 – Logic Programming exercise #3

This is the third of a series of exercises you will have to complete as part of your project (worth 50% of your final mark) for this study unit. These exercises will generally be focused on (though not limited to) the material covered in the previous lecture. They are not optional, so please ensure that you do them, as otherwise this will affect your final mark.

Preparation

You will be using the Classtime online assessment tool for these exercises (instructions for this will follow). It is important you ensure that you are logged in to the university website prior to starting the exercise, and that, when Classtime asks you to log in using a Google ID, you use your @um.edu.mt ID so that you can be identified correctly. Logging in with your Google ID (ie, your @um.edu.mt ID) will also enable you to interrupt your session and log in later to complete it if this is necessary – although I would recommend that you complete the exercise all at once.

VERY IMPORTANT: You should **develop your answers using the prolog interpreter** and **only insert them** into the Classtime answer fields **when you are sure of them**, as you will not be able to change your answers once you have submitted them.

Exercise 3

For this exercise, open your prolog interpreter and type your programs directly into the editor. Compile and run/test them there before responding to the questions in Classtime.

3.1 Implement Stein's Algorithm in prolog

In class we covered a method of computing the Greatest Common Divisor (GCD) of two non-negative integers using Euclid's algorithm.

A more efficient way of calculating the GCD is using Stein's algorithm. This shortcuts the subtractions done in the conventional Euclidean algorithm, replacing the larger of the two numbers by its remainder when divided by the smaller. The stopping condition is now when one reaches a zero remainder – when this happens, the GCD is the (non-zero) number that is left.

Write clauses to compute the GCD of two numbers using Stein's algorithm.

3.2 A to the power of B

Write prolog clauses to implement a function, power/3, that takes two numbers A and B, and computes C = A to the power of B.

When you have finished testing your programs, open www.classtime.com and use the code RVP42 *to access ex 3.1 and the code* ER3YM *to access ex 3.2.* ***Answer all questions.***

VERY IMPORTANT: Retain copies of your answers in case there should be any issue with the Classtime system and you should need to input them again, or I should need to see evidence of the work you have done.

Please write to me at peter.xuereb@um.edu.mt should you have any questions.