

CM20019 – Prolog Lab –part 4

This exercise will give you useful practice for the next assessed Prolog lab exercise. Test your programs with representative queries, in each case finding all solutions.

Preliminary

Enter `:- use_module(library(lists)).` at the start of your Prolog script to enable access to the Sicstus library of list-processing predicates.

1. Write a program for the relation `pairs(X, Y)` which, given a list `X` of numbers, constructs a list `Y` of the same length such that

each member of `Y` is `(U, V)` iff the corresponding member of `X` is `N`
and `U` is `N-1` and `V` is `N+1`.

2. Write a program for the relation `arbpairs(X, Y)` which, given a list `X` of numbers, constructs a list `Y` of the same length such that

each member of `Y` is `(N, L)` iff the corresponding member of `X` is `N`
and `L` is either `N` or `2N`.

3. An arithmetic term is defined to be

	a number	
else	a term <code>a(X, Y)</code>	where <code>X</code> and <code>Y</code> are both arithmetic terms
else	a term <code>m(X, Y)</code>	where <code>X</code> and <code>Y</code> are both arithmetic terms

A term `a(X, Y)` represents `X+Y`, whilst `m(X, Y)` represents `X*Y`.

Write a program for the relation `numval(A, V)` which, given the arithmetic term `A`, computes the numerical value `V` that it represents.

For example,

if	<code>A=a(m(3, a(2, 5)), a(2, 3))</code>
then	<code>A</code> represents <code>(3*(2+5))+(2+3)</code>
so	<code>V=26</code> .

The only predicates you need to use are `numval`, `number` and `is`.