CEng 242 Homework 6

Due: 29^{th} May 2005

In this homework you will implement some prolog clauses to make computations on polynomial expressions. A polynomial expression in this case is defined by infix expressions composed of:

- numeric constants (prolog integer and real valued terms)
- The variable '`x`' which is the only variable expressions can have.
- The infix expressions of '`*,+,-,^`' operators. '`^`' denotes the power operation and left handside should be '`x`' and right handside should be a non-negative integer. Also unary minus '`-`' will be handled. All other expressions and terms will be considered invalid and ignored. No division or other operators are defined in our expressions.

For example, the following are valid polynomial expressions:
`(x-2)* -(x^2-1)`
`x+x+x^1+x^2*x^3*1+2*x^2`
In order to represent polynomials in a normal form we will keep them in a vector like representation. A list of constant numbers in the form $[c_0, c_1, c_2, ..., c_n]$ denotes the polynomial $c_0 + c_1x + c_2x^2 + ... + c_nx^n$. We will call this list as the *normal form polynomial expression.*
Write the following prolog clauses: ,

`add(`*`ne1`*`, `*`ne2`*`, `*`ne3`*`)`
Adds two normal form polynomials and gives the resulting polynomial in normal form in *`ne3`*.
```
?- add([1,0,1,2,1],[0,1,0,0,3,2,2],R).
R= [1,1,1,2,4,2,2].
```

`multiply(`*`ne1`*`, `*`ne2`*`, `*`ne3`*`)`
Multiplies two polinomials in normal form and gives the resulting polynomial in third parameter.
```
?- multiply([1,2,1],[1,0,3],R).
R= [1,2,4,6,3].
```

`subtract(`*`ne1`*`, `*`ne2`*`, `*`ne3`*`)`
Similar to `add/3` but subtracts the second term from the first term
```
?- subtract([1,0,1,2,1],[0,1,0,0,3,2,2],R).
R= [1,-1,1,2,-2,-2,-2].
```

`normalize(`*`pexp`*`, `*`nexp`*`)`
Gets a polynomial expression *`pexp`* in arbitrary infix form and gives the normal form in the *`nexp`*.
```
?- normalize(x+x+x^1+x^2*x^3*1+2*x^2,R).
R=[0, 3, 2, 0, 0, 1].
```

`evalpoly(`*`pexp`*`,`*`xval`*`,`*`result`*`)`
Gets a polynomial expression *`pexp`* in infix form, substitutes all occurences of '`x`' with evaluatable numerical expression *`xval`* and calculates a single numerical value in *`result`*. This works like a function application. The evaluation order (eager or normal order) is not relevant, you may choose to normalize or evaluate directly.
```
?- evalpoly(x+x+x^1+x^2*x^3*1+2*x^2,2,R).
R= 46.
```

`prettypoly(`*`nexp`*`,`*`pexp`*`)`
This clause will be bonus (+20 points) for this homework. It will get a normal form polynomial in *`nexp`* and gives a infix polynomial expression in a form like `1+2*x+x^5+2*x^6` where 0 terms removed, 0 and 1 powers removed, constant 1 multipliers removed, terms are ordered from minimum to maximum power.
```
?- prettypoly([1, 2, 0, 0, 0, 1, 2],R).
R= 1+2*x+x^5+2*x^6.
```
(*it is not* `1+(2*x+(x^5+2*x^6))`).

You can have any other clause definitions in your program. Put everything in `hw6.pl` and submit as a single file.
We like to remind you that our cheating policy is to give 0 to all participants for all 6 previous and following homeworks.