Homework Assignment #8: Another ML Programming Exercise

Note

This assignment constitutes 4% of your grade and **may be completed by a team of two members or by yourself alone**. It is due 2:10PM Wednesday, December 12, 2012. Please write/type your answers/code on A4 (or similar size) paper. Drop your homework (one copy per team) by the due time in Yih-Kuen Tsay's mail box on the first floor of Management College Building 2. Late submission will be penalized by 20% for each working day overdue. You may discuss the problems with other teams or individuals, but copying answers/code is strictly forbidden.

Your work will be graded according to its thoroughness, correctness, and presentation. Specifically, you should provide evidences showing that your program is correct. You should also organize and document your program in such a way that other programmers, for example your classmates, can understand it. **Some of you may be requested to demonstrate your program.**

Problem

Write an ML program to simplify multi-variable polynomials in the infix notation. Below are a few examples of input and its expected output.

- "`2x + 0 + (1+0)y`" $\mapsto$ "`2x + y`"
- "`x + y + 2*3 + 2x + x^2 + y^2`" $\mapsto$ "`x^2 + 3x + (y^2+y+6)`"
- "`(2x + 1)y + x + x^2`" $\mapsto$ "`x^2 + (2y+1)x + y`"
- "`1 + 2yx + x^2 + x^2*y + x^3*2y + y`" $\mapsto$ "`2yx^3 + (y+1)x^2 + 2yx + (y+1)`" (or "`2x^3y + (y+1)x^2 + 2xy + (y+1)`")

(Note: the multiplication operator `*` is omitted in some places; e.g., `yx` is intended to be `y*x` rather than a variable with two characters.)

In general, you should simplify a polynomial as a sum of terms, while grouping terms that are "similar", e.g., having the same exponent for `x`. You may choose your own measure of similarity between terms, as long as the produced output looks reasonably simpler than the input.

It is satisfactory if you deal only with two-variable polynomials and single-digit numbers (as coefficients or constants). You will earn extra credits if you can handle more than two variables and/or multi-digit numbers.

Below are a number of functions that you may find useful:

```
# string_of_int 123;;
- : string = "123"
# int_of_string "123";;
- : int = 123
# String.length "abcde";;
- : int = 5
```

```
# String.get "abcde" 0;;
- : char = 'a'
# String.get "abcde" 2;;
- : char = 'c'
# Char.escaped 'c';;
- : string = "c"
# String.sub "abcde" 2 3;;
- : string = "cde"
# "ab" ^ "cde";;
- : string = "abcde"
# String.concat "+" ["a";"bc"; "de"];;
- : string = "a+bc+de"
```