Programming Project 4

Due date: Thu 3/30 11:00 pm

Introduction

In this assignment you will implement functions to handle addition and multiplication of arbitrary size integers in ML. We will call these *big integers*. To simplify matters you may assume that the numbers are greater or equal to 1. These operations (and others) for long integers are needed for various applications, for example encryption and decryption with the RSA system.

You should represent a big integer as list of integers where each element of the list represents one digit. It is convenient to represent the integer lists reversed. Thus 123 should be represented by the list [3,2,1]. (Of course we could use a larger base for a digit to save space and time but for simplicity and uniformity of projects we ask you to use base 10 as in the example.)

Your Task

You should implement the following 4 functions where \( t \) stands for \texttt{int list}

- \texttt{fromInt: int -> t;}
- \texttt{toString: t -> string;}
- \texttt{plus: t * t -> t;}
- \texttt{mult: t * t -> t;}

\texttt{fromInt} takes an integer and converts it into the int list representation. \texttt{toString} takes an int list representing a number and returns a string representation of the number in correct order. So for the number above the string “123” should be returned (this will basically be our mechanism to check your results.)

The functions \texttt{plus} and \texttt{mult} perform addition and multiplication on integers in this format. You should use the standard “school algorithms” for these functions. Thus \texttt{plus} should add the numbers digit by digit passing carry as needed. The function \texttt{mult} should multiply the first number by the digits of the second and add them, as in

\[
\begin{array}{c}
1 & 2 & 3 \\
2 & 5 \\
\hline
6 & 1 & 5 \\
2 & 4 & 6 \\
\hline
3 & 0 & 7 & 5
\end{array}
\]

You must use this algorithm or an algorithm as fast as this one. Slow implementations that multiply, e.g. \( a \times b \) by adding \( a \) to a sum \( b \) times are not acceptable.

Here are two examples of using your functions. The code

\[
\begin{align*}
&\text{val a = fromInt(123);} \\
&\text{val b = fromInt(25);} \\
&\text{val c = mult(a,b);} \\
&\text{val d = toString(c);} \\
\end{align*}
\]

produces the following response:
val a = [3,2,1] : int list
val b = [5,2] : int list
val c = [5,7,0,3] : int list
val d = "3075" : string

and the code:

val a = fromInt(1234567);
val b = fromInt(654321);
val c = mult(a,b);
val d = mult(c,c);
val e = toString(d);

produces the following response:

val a = [7,6,5,4,3,2,1] : int list
val b = [1,2,3,4,5,6] : int list
val c = [7,0,0,4,1,1,3,0,8,7,0,8] : int list
val d = [9,4,0,6,9,5,9,3,2,6,0,4,...] : int list
val e = "652545870999406239596049" : string

Summary of Requirements

1. Implement all 4 functions as described above. You may use as many “helper functions” as you need.

2. Use an accumulator in at least one of these functions.

3. Put all your code including your functions and at least 2 example runs (different from the above) in a file pp4.sml. Make sure that the code in pp4.sml runs smoothly in the system (that is, no errors should occur if we try sml < pp4.sml).

4. Make sure that your code is easily readable and well documented. The grade will be based both on correctness and on readability.

Submitting your program

Assuming your code is in the current directory on one of the local Sun machines, you should submit by typing provide comp80 pp4 pp4.sml.