Homework 4: Due Monday, February 13

Problem 1: Write a function optionMax, that takes two int options, and returns the int option that is obtained by taking the max of the two underlying inputs. For example, optionMax(NONE, NONE) = NONE, optionMax(NONE, SOME(3)) = SOME(3), and optionMax(SOME(5), SOME(3)) = SOME(5).

Problem 2: In class and on Homework 3, we coded sets in ML by using lists without repetition. Suppose instead that we code sets of integers as *ordered* lists (in increasing order) without repetition. Using this approach, write ML functions that perform each of the following. In each case, write the most efficient procedure that you can by exploiting the ordered nature of the lists.
a. setAddElem.
b. setUnion: Your function should only walk through each list at most once (unlike our corresponding function on unordered lists, which would walk through the second list many times, once for each element of the first list). In particular, you should avoid calling the function from part a, as each such call might walk through the corresponding list.
c. setIntersection: As above, your function should walk through each list at most once.
d. setMax.

Problem 3:
a. Let A be a finite set, and suppose that $b \notin A$. Using our discussion in class, together with the function powerSet that we wrote, explain why $|\mathcal{P}(\{b\} \cup A)| = 2 \cdot |\mathcal{P}(A)|$.
b. Using part a, explain why $|\mathcal{P}(A)| = 2^n$ whenever $|A| = n$.

Problem 4:
a. Determine the set $\mathcal{P}(\mathcal{P}(\{1, 2\}))$ explicitly. What is its cardinality?
b. Is $\mathbb{Z} \in \mathcal{P}(\mathbb{R})$? Is $\mathbb{Z} \subseteq \mathcal{P}(\mathbb{R})$? Explain.
c. Does $\mathcal{P}(A) \cap \mathcal{P}(B) = \mathcal{P}(A \cap B)$ for all sets A and B? Explain.
d. Does $\mathcal{P}(A) \cup \mathcal{P}(B) = \mathcal{P}(A \cup B)$ for all sets A and B? Explain.

Problem 5: In class, we introduced the recursive type natural that was defined by

datatype natural = Zero | Successor of natural.

Without converting anything to integers, write the following ML functions:
a. Write a function isEven that takes an input of type natural, and outputs the boolean that says whether the corresponding element is even. Note that 0 is defined to be even (as we will see later). Thus, isEven(Zero) = true, isEven(Successor(Zero)) = false, etc.
b. In class, we wrote a function plus for natural. Now write a function times for natural. Thus times takes two inputs of type natural, and outputs the product of the inputs (as an object of type natural).