CS3311 Homework 3 Due date: Wednesday, September 26, 2012, in class

---

The answers must be the original work of the author. While discussion with others is permitted and encouraged, the final work should be done individually. You are not allowed to work in groups. You are allowed to build on the material supplied in the class. Any other source must be specified clearly.

---

1. *(20 points)* Give a recursive definition of the following sets. You may only use the successor operator to generate new elements.

Part a. $S_1 = \{k \mid k \in \mathbb{N} \text{ and } (k \text{ is even or } k \geq 10)\}$

Part b. $S_2 = \{0, 2, 4, 6, \ldots\} \cup \{[0,0], [0,1], [0,2], \ldots\}$

2. *(10 points)* Give a recursive definition the relation R_1 on $\mathbb{N}$ where xR_1y iff $y = x^n$ and $x, y, n \in \mathbb{N}$. You may assume that a multiplication operator ($\times$) is defined.

3. *(10 points)* Subtraction on the set of natural numbers is defined by

$$n \dot{-} m = \begin{cases} n - m & \text{if } n > m \\ 0 & otherwise \end{cases}$$

This operation is often called *proper subtraction* ($n \dot{-} m$). Give a recursive definition of proper subtraction. You may use the successor ($s(n)$) or the predecessor ($pred(n)$) operations, but no other operations are permitted. Checking for equality is also permitted, e.g., "if $n = 0$".

4. *(10 points)* Let X be a finite set. Give a recursive definition of the set of subsets of X. Use set union as the operator in the definition.

5. *(20 points)* Use induction on the size of X to show that if X is a finite set, then $card(P(X)) = 2^{card(X)}$. Clearly label the **basis**, **inductive hypothesis**, and **inductive step**.

6. *(10 points)* Use induction to prove that $\sum_{i=1}^{n} 2i - 1 = n^2$. Clearly label the **basis**, **inductive hypothesis**, and **inductive step**.

7. *(20 points)* Consider the following program segment and use induction on the number of iterations of the `for` loop to prove that the value printed out for `Y` is n^2. You must present the proof based on the pseudocode and on the number of iterations of the `for` loop. Clearly label the **basis**, **inductive hypothesis**, and **inductive step**. The loop is an implementation of: $1 + 3 + 5 + \ldots + (2n - 1) = n^2$.

```
Y = 0;
for I = 1 to n
  {
    Z = (2 * I - 1);
    Y = Y + Z;
  }
print (Y);
```