CS3311 Homework 3

Due date: Wednesday, September 22, 2010, in class

The answers, comments, and programs (if any) 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. If you use any other source than the class notes and the textbook, specify it clearly.

---

1. *(10 points)* Give a recursive definition of the *equal to* relation on $\mathcal{N} \times \mathcal{N}$. Use the successor operator s.

2. *(15 points)* Give a recursive definition of the *equal to* relation on $(\mathcal{N} \times \mathcal{N}) \times (\mathcal{N} \times \mathcal{N})$. The objective is to compare pairs and put them into the relation if and only if they are equal. For example, $[[1,2],[1,2]]$ is a member of the relation, but $[[1,2],[1,3]]$ is not. Use the successor operator s.

3. *(15 points)* Give a recursive definition of the predecessor operation shown below. Use the successor operator s.

$$pred(n) = \begin{cases} 0 & \text{if } n = 0 \\ n-1 & \text{otherwise} \end{cases}$$

4. *(15 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. *(15 points)* Use induction to prove that $(7^{2n} - 48n - 1) \bmod 2304 = 0$. Clearly label the **basis**, **inductive hypothesis**, and **inductive step**.

6. *(15 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**.

7. *(15 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);
```