University of Washington
Department of Computer Science and Engineering
CSE 311, Autumn 2012

November 6, 2012

Homework 7, Due Wednesday, November 14, 2012

Problem 1:

Define $T(n)$ for $n \geq 1$ by $T(1) = 0$, $T(n+1) = T((n+1)/2)+1$ if $n \geq 1$ is odd and $T(n+1) = T(n)$ if $n \geq 1$ is even. Prove that $2^{T(n)} \leq n$ for all $n \geq 1$,

Problem 2:

Consider the following one-player game: The player starts with an integer $n \geq 1$.
If $n = 1$ the game stops and the player has not earned any points.
If $n > 1$ the player gets to split n into two positive integers r and $n - r$. For this move, the player earns $r \cdot (n - r)$ points. After this, the player plays the game both on r and on $n - r$, adding the points earned from those games to the points already earned.

Prove that no matter how the player plays on input $n \geq 1$, the player earns exactly $n(n - 1)/2$ points.

Problem 3:

In class we gave the following recursive definition of a set S:
Basis: $[1, 1, 0] \in S$ and $[0, 1, 1] \in S$.
Recursive Step: If $[x_1, y_1, z_1] \in S$ and $[x_2, y_2, z_2] \in S$ then $[x_1 + x_2, y_1 + y_2, z_1 + z_2] \in S$ and $[\alpha x_1, \alpha y_1, \alpha z_1] \in S$ for every $\alpha \in \mathbb{R}$.
Prove that for every $[x, y, z] \in S$ we have $y = x + z$.

Problem 4:

The set of *almost balanced* binary trees is a subset of all rooted binary trees and is defined in the same way as rooted binary trees except that the recursive step has an extra restriction:
In an almost balanced binary tree, one can only join trees T_1 and T_2 as in the rooted binary tree definition if either **height**(T_1) = **height**(T_2) or **height**(T_1) and **height**(T_2) differ by 1. The functions **size** and **height** are defined exactly as for rooted binary trees.

Prove by induction that every almost balanced binary tree T satisfies $\textbf{size}(T) \geq f_{\textbf{height}(T)+1}$ where f_m denotes the m-th Fibonacci number. (As usual $f_0 = 0$, $f_1 = 1$, and $f_{m+1} = f_m + f_{m-1}$ for $m \geq 1$.)

Problem 5:

Construct regular expressions that match (generate) each of the following sets of strings.

a) The set of all binary strings that start with 0 and have even length, or start with 1 and have odd length.

b) The set of all binary strings that have a 1 in every odd-numbered position counting from the start of the string.

Problem 6:

Construct regular expressions that match (generate) each of the following sets of strings.

a) The set of all binary strings that contain at least two 0's and at most one 1.

b) The set of all binary strings that don't contain 110.

Problem 7:

Construct context-free grammars that generate the following sets of strings. For each of your constructions write a sentence or two to explain why your construction is correct. If you use more than one variable, as documentation explain what sets of strings you expect each variable to generate.

a) The set of all binary strings that contain at least two 0's and at most one 1.

b) The set of all binary strings that are of odd length and have 0 as their middle character.

Problem 8:

If $a \in \Sigma$ is a symbol then the string a^n for $n \geq 0$ is the string consisting of n copies of a, one after another. Construct context-free grammars that generate the following sets of strings. For each of your constructions write a sentence or two to explain why your construction is correct. If you use more than one variable, as documentation explain what sets of strings you expect each variable to generate.

a) $\{0^m 1^n 0^{m+n} : m, n \geq 0\}$.

b) $\{0^m 1^n 0^p : m, n, p \geq 0 \text{ and } m = n \text{ or } n = p\}$.

Extra Credit 9:

Consider the set S_3 of strings in $\{0, 1, 2\}^*$ such that the sum of the values is congruent to 0 modulo 3, so

$$S_3 = \{\lambda, 0, 00, 12, 21, 000, 012, 021, 102, 111, 120, 201, 210, 222, \cdots\}.$$

a) Design a context-free grammar that generates S_3.

b) Design a regular expression that generates S_3.