CSE 311  Foundations of Computing I
Lecture 16
Functions on Recursively Defined Sets and Structural Induction
Autumn 2011

Highlights from last lecture

• Recursively defined sets
  – Basis step: Some specific elements are in S
  – Recursive step: Given some existing named elements in S some new objects constructed from these named elements are also in S
• Structural Induction:
  1. By induction we will show that P(x) is true for every x in S
  2. Base Case: Show that P is true for all elements of S mentioned in the Basis step
  3. Inductive Hypothesis: Assume that P is true for some arbitrary values of each of the existing named elements mentioned in the Recursive step
  4. Inductive Step: Prove that P holds for each new element constructed in the Recursive step using the elements mentioned in the Inductive Hypothesis
  5. Conclusion: Result follows by induction

Strings

• An alphabet \( \Sigma \) is any finite set of characters.
• The set \( \Sigma^* \) of strings over the alphabet \( \Sigma \) is defined by
  – Basis: \( \lambda \in \Sigma^* \) (\( \lambda \) is the empty string)
  – Recursive: If \( w \in \Sigma^* \), \( x \in \Sigma \), then \( wx \in \Sigma^* \)

Palindromes

• Palindromes are strings that are the same backwards and forwards
• Basis: \( \lambda \) is a palindrome and any \( a \in \Sigma \) is a palindrome
• Recursive step: If \( p \) is a palindrome then \( apa \) is a palindrome for every \( a \in \Sigma \)

Function definitions on recursively defined sets

\[ \text{len}(\lambda) = 0; \]
\[ \text{len}(wa) = 1 + \text{len}(w); \text{ for } w \in \Sigma^*, a \in \Sigma \]

Reversal:
\[ \lambda^R = \lambda, \]
\[ (wa)^R = aw^R \text{ for } w \in \Sigma^*, a \in \Sigma \]

Concatenation:
\[ w \cdot \lambda = w \text{ for } w \in \Sigma^* \]
\[ w_1 \cdot w_2 a = (w_1 \cdot w_2)a \text{ for } w_1, w_2 \in \Sigma^*, a \in \Sigma \]
len(x•y)=len(x)+len(y) for all strings x and y

Rooted Binary trees

• Basis: is a rooted binary tree
• Recursive Step: If \( T_1 \) and \( T_2 \) are rooted binary trees then so is:

Functions defined on rooted binary trees

• \( \text{size}(●)=1 \)
• \( \text{size}(\text{tree}) = 1+\text{size}(T_1)+\text{size}(T_2) \)
• \( \text{height}(●)=0 \)
• \( \text{height}(\text{tree}) = 1+\max\{\text{height}(T_1),\text{height}(T_2)\} \)

For every rooted binary tree T

\[ \text{size}(T) \leq 2^{\text{height}(T)+1} - 1 \]