CSE 390Z: Mathematics for Computing Workshop

Week 8 Workshop

0. Structural Induction: Strings

Recursive Definition of a String:

- Basis Step: ϵ is a string
- Recursive Step: If w is a string and a is a character, $w \bullet a$ is a string (the string w with the character a appended on to the end)

Recursive functions on String:

Length:

$$\begin{aligned} \text{len}(\epsilon) &= 0 \\ \text{len}(w \bullet a) &= \text{len}(a \bullet w) = 1 + \text{len}(w) \end{aligned}$$

Reverse:

$$\begin{aligned} \text{rev}(\epsilon) &= \epsilon \\ \text{rev}(w \bullet a) &= a \bullet \text{rev}(w) \end{aligned}$$

Prove that for any string x, len(rev(x)) = len(x).

Complete the proof below:

Let P(x) be "______________________________". We will prove P(x) for all strings x by structural induction on the set of strings.

Base Case : ($x = \epsilon$)

Let s be an arbitrary string not covered by the base case. Then by the exclusion rule, $s = w \bullet a$ for some string w and some character a.

Inductive Hypothesis: Suppose $P($_____$)$ holds.

Inductive Step: $\boxed{\text{Goal: Show that } P(\_\_\_\_\_\_\_\_\_\_) \text{ holds}}$

This proves $P($__________$)$.

Conclusion: P(x) holds for all strings x by structural induction.

1. Structural Induction: CharTrees

Recursive Definition of CharTrees:

- Basis Step: `Null` is a **CharTree**
- Recursive Step: If L, R are **CharTree**s and $c \in \Sigma$, then $\texttt{CharTree}(L, c, R)$ is also a **CharTree**

Intuitively, a **CharTree** is a tree where the non-null nodes store a `char` data element.

Recursive functions on CharTrees:

- The preorder function returns the preorder traversal of all elements in a **CharTree**.

$$\begin{aligned} \text{preorder}(\texttt{Null}) &= \varepsilon \\ \text{preorder}(\texttt{CharTree}(L, c, R)) &= c \cdot \text{preorder}(L) \cdot \text{preorder}(R) \end{aligned}$$

- The postorder function returns the postorder traversal of all elements in a **CharTree**.

$$\begin{aligned} \text{postorder}(\texttt{Null}) &= \varepsilon \\ \text{postorder}(\texttt{CharTree}(L, c, R)) &= \text{postorder}(L) \cdot \text{postorder}(R) \cdot c \end{aligned}$$

- The mirror function produces the mirror image of a **CharTree**.

$$\begin{aligned} \text{mirror}(\texttt{Null}) &= \texttt{Null} \\ \text{mirror}(\texttt{CharTree}(L, c, R)) &= \texttt{CharTree}(\text{mirror}(R), c, \text{mirror}(L)) \end{aligned}$$

- Finally, for all strings x, let the "reversal" of x (in symbols x^R) produce the string in reverse order.

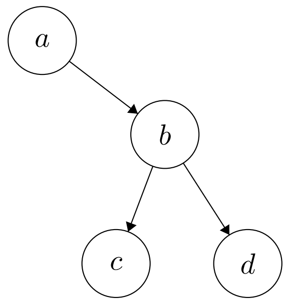
Additional Facts:
You may use the following facts:

- For any strings $x_1, ..., x_k$: $(x_1 \cdot ... \cdot x_k)^R = x_k^R \cdot ... \cdot x_1^R$
- For any character c, $c^R = c$

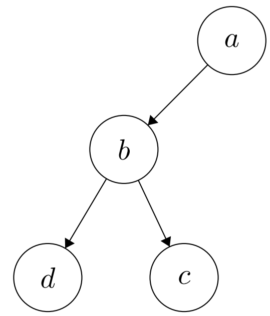
Statement to Prove:
Show that for every **CharTree** T, the reversal of the preorder traversal of T is the same as the postorder traversal of the mirror of T. In notation, you should prove that for every **CharTree**, T: $[\text{preorder}(T)]^R = \text{postorder}(\text{mirror}(T))$.

There is an example and space to work on the next page.

Example for Intuition:



Let T_i be the tree above.
preorder(T_i) ="abcd".
T_i is built as $(\texttt{null}, a, U)$
Where U is (V, b, W),
$V = (\texttt{null}, c, \texttt{null}), W = (\texttt{null}, d, \texttt{null})$.



This tree is mirror(T_i).
postorder(mirror(T_i)) ="dcba",
"dcba" is the reversal of "abcd" so
[preorder(T_i)]^R = postorder(mirror(T_i)) holds for T_i

2. Structural Induction: Dictionaries

Recursive definition of a Dictionary (i.e. a Map):

- Basis Case: [] is the empty dictionary
- Recursive Case: If D is a dictionary, and a and b are elements of the universe, then $(a \to b)$ :: D is a dictionary that maps a to b (in addition to the content of D).

Recursive functions on Dictionaries:

$$\begin{aligned} \text{AllKeys}([\,]) &= [\,] \\ \text{AllKeys}((a \to b) :: \text{D}) &= a :: \text{AllKeys}(\text{D}) \end{aligned}$$

$$\begin{aligned} \text{len}([\,]) &= 0 \\ \text{len}((a \to b) :: \text{D}) &= 1 + \text{len}(\text{D}) \end{aligned}$$

Recursive functions on Sets:

$$\begin{aligned} \text{len}([\,]) &= 0 \\ \text{len}(a :: \text{C}) &= 1 + \text{len}(\text{C}) \end{aligned}$$

Statement to prove:

Prove that len(D) = len(AllKeys(D)).

3. Structural Induction: CFGs

Consider the following CFG:

$$S \to SS \mid 0S1 \mid 1S0 \mid \epsilon$$

Prove that every string generated by this CFG has an equal number of 1's and 0's.

Hint 1: Start by converting this CFG to a recursively defined set.
Hint 2: You may wish to define the functions $\#_0(x), \#_1(x)$ on a string x.