CSE 390Z: Mathematics for Computation Workshop

Week 7 Workshop

0. Conceptual Review

(a) Regular expression rules:
Basis: ε, a for $a \in \Sigma$
Recursive: If A, B are regular expressions, $(A \cup B)$, AB, and A^* are regular expressions.

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**.

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

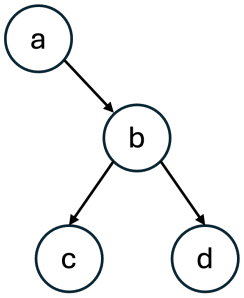
- Finally, for all strings x, the "reversal" of x, denoted x^R, produces the string in reverse order.

Additional Facts:
You may use the following facts:

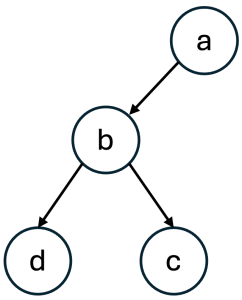
- **Fact 1:** For any strings $x_1, ..., x_k$: $(x_1 \cdot ... \cdot x_k)^R = x_k^R \cdot ... \cdot x_1^R$
- **Fact 2:** For any character c, $c^R = c$

It turns out that for any CharTree T, the reversal of the preorder traversal of T is the same as the postorder traversal of the mirror of T.

Example for Intuition:



Let T be the tree above.
T is built as follows: the two leaf nodes are
$C = (\texttt{Null}, c, \texttt{Null})$ and $D = (\texttt{Null}, d, \texttt{Null})$
The tree rooted at b is $B = (C, b, D)$
Finally, T is $T = (\texttt{Null}, a, B)$
preorder(T) =“abcd”.



This tree is mirror(T).
postorder(mirror(T)) =“dcba”,
“dcba” is the reversal of “abcd” so
$[\text{preorder}(T)]^R = \text{postorder}(\text{mirror}(T))$ holds for T

Use structural induction to prove the following claim:

For every **CharTree**, T: $[\text{preorder}(T)]^R = \text{postorder}(\text{mirror}(T))$

2. More Induction...Literally

Define a set S as follows:
Basis: $6 \in S$; $15 \in S$
Recursive: if $x, y \in S$ then $x + y \in S$

Define a set T as follows:
Basis: $6 \in T$; $15 \in T$
Recursive: if $x \in T$ then $x+6 \in T$ and $x+15 \in T$

In lecture you proved that every element of T is an element of S.
Now we're going to prove that every element of S is an element of T.

(a) First, use structural induction to prove the following lemma:
The sum of any two elements in T is also in T. Formally this is: $\forall a, b \in T(a + b \in T)$

(b) Now, use structural induction to prove the main claim: Every element of S is also in T.
You can use the Lemma from part (a) by citing "part (a) lemma".

3. Regular Expressions Warmup

(a) Consider the following Regular Expression (RegEx):

$$1(45 \cup 54)^\star 1$$

List 5 strings that are accepted by the RegEx and 5 strings that are rejected. The strings should be over the alphabet $\Sigma := \{1, 4, 5\}$. After listing the strings, summarize the RegEx in your own words.

(b) Consider the following Regular Expression (RegEx):

$$a(aaa)^*(bb)^*$$

List 5 strings that are accepted by the RegEx and 5 strings that are rejected. The strings should be over the alphabet $\Sigma := \{a, b\}$. After listing the strings, summarize the RegEx in your own words.

4. Constructing RegExs

For each of the following, construct a regular expression for the specified language.

(a) Strings over the alphabet $\Sigma := \{a, b\}$ with odd length.

(b) Strings over the alphabet $\Sigma := \{a\}$ with an even number of a's.

(c) Strings over the alphabet $\Sigma := \{a, b\}$ with an even number of a's.

(d) Strings over the alphabet $\Sigma := \{a, b\}$ with alternating a's and b's (i.e., not containing aa or bb).

(e) Strings over the alphabet $\Sigma := \{a, b\}$ where the second to last character is a b.

(f) Strings over the alphabet $\Sigma := \{a, b\}$ not ending in aa.

(g) Strings over the alphabet $\Sigma := \{a, b\}$ with an even number of a's or an odd number of b's.