# CSE 390Z: Mathematics for Computation Workshop

## Week 8 Workshop

## 0. Conceptual Review

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

## 1. 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.

## 2. Constructing RegExes

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 $aa$.

## Relations Conceptual Review

Relations definitions: Let $R$ be a relation on $A$. Then:

- $R$ is reflexive iff for all $a \in A$, $(a, a) \in R$.
- $R$ is symmetric iff for all $a, b$, if $(a, b) \in R$, then $(b, a) \in R$.
- $R$ is antisymmetric iff for all $a, b$, if $(a, b) \in R$ and $a \neq b$, then $(b, a) \notin R$.
- $R$ is transitive iff for all $a, b$, if $(a, b) \in R$ and $(b, c) \in R$, then $(a, c) \in R$.

**Note:** Instead of saying $R$ is a relation on $A$, we might say $R \subseteq A \times A$. They mean the same thing.

Let $R, S$ be relations on $A$. Then:

- $R \circ S = \{(a, c) : \exists b \text{ such that } (a, b) \in R \text{ and } (b, c) \in S\}$

## 4. Relations Examples

(a) Suppose that $R, S$ are relations on $\mathbb{Z}$, where $R = \{(1, 2), (4, 3), (5, 5)\}$ and $S = \{(2, 5), (2, 7), (3, 3)\}$. What is $R \circ S$? What is $S \circ R$?

(b) Consider the relation $R \subseteq \mathbb{Z} \times \mathbb{Z}$ defined by $(a, b) \in R$ iff $a \leq b + 1$. List 3 pairs of integers that are in $R$, and 3 pairs of integers that are not.

(c) Consider the relation $R \subseteq \mathbb{Z} \times \mathbb{Z}$ defined by $(a, b) \in R$ iff $a \leq b + 1$. Determine if $R$ is reflexive, symmetric, antisymmetric, and/or transitive. If the relation has a property, explain why. If not, state a counterexample.

## 5. Relations Proofs

Suppose that $R, S \subseteq \mathbb{Z} \times \mathbb{Z}$ are relations.

Recall that we can disprove a statement by providing a counterexample.

(a) Prove or disprove: If $R$ and $S$ are transitive, $R \cup S$ is transitive.

(b) Prove or disprove: If $R$ and $S$ are reflexive, then $R \circ S$ is reflexive.

(c) Prove or disprove: If $R \circ S$ is reflexive, then $R$ and $S$ are reflexive.

(d) Prove or disprove: If $R$ is symmetric, $\overline{R}$ (the complement of $R$) is symmetric.

## 6. Context Free Grammars Warmup

Consider the following CFG that generates strings over the alphabet $\Sigma := \{0, 1, 2, 3, 4\}$

$$\mathbf{S} \to 0\mathbf{X}4$$
$$\mathbf{X} \to 1\mathbf{X}3 \mid 2$$

List 5 strings that are accepted by the CFG and 5 strings that are rejected. The strings should be over the alphabet $\Sigma := \{0, 1, 2, 3, 4\}$. After listing the strings, summarize the CFG in your own words.

## 7. Constructing CFGs

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

**Note:** The regular expressions for each, which we constructed in last week's workshop, have been provided.

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

**RegEx:** $(aa)^*$

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

**RegEx:** $b^*(ab^*ab^*)^*$ **Alternate RegEx:** $(b \cup (ab^*a))^*$

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

**RegEx:** $(a \cup b)(aa \cup ab \cup ba \cup bb)^*$

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

**RegEx:** $(b^*(ab^*ab^*)^*) \cup (a^*ba^*(ba^*ba^*)^*)$

## 8. Constructing DFAs

For each of the following, construct a DFA for the specified language over the alphabet $\Sigma = \{a, b\}$.

(a) Strings with odd length.

(b) Strings with an even number of $a$'s.

(c) Strings with an odd number of $b$'s.

(d) Strings with an even number of $a$'s **or** an odd number of $b$'s.

## 9. All the Things

Let $\Sigma := \{0, 1, 2, 3, 4, 5\}$. For an arbitrary string $x$ over $\Sigma$, we can write $x = x_0 x_1 \cdots x_n$, where $x_0, x_1, ..., x_n \in \Sigma$. Define a language $L$ over $\Sigma$ as follows:
$x \in L$ iff for every position $i$ from $0$ to $n$, if the value of $x_i$ is odd, then every digit (character) that comes after $x_i$ must be **greater** than $x_i$.

For example, the string $2124 \in L$ because $1$ is the only odd digit and every digit after 1 is greater than 1.
The string $21254 \notin L$ because 5 is an odd digit, 4 comes after 5, and $4 < 5$.
The string $211 \notin L$ because 1 comes after 1 and $1 \not> 1$.

(a) List 3 strings in $L$ and 3 strings not in $L$. The strings should be over the alphabet $\Sigma$.

(b) Construct a regular expression for the language $L$.

(c) Construct a CFG for the language $L$.

(d) Construct a DFA for the language $L$.

## 10. Structural Induction: CFGs

Consider the following CFG:

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

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

You may assume that the functions $\#_0(x), \#_1(x)$ are defined and return the number of zeros in a string $x$ and the number of $1$ in a string $x$, respectively.

**Hint:** Start by converting this CFG to a recursively defined set.