CSE 390Z: Mathematics for Computation Workshop

Week 8 Workshop Solutions

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. 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.

Solution:

Accepted:

- 1451
- 1541
- 145541
- 1454545451
- 11

Rejected:

- 1
- 1441
- 45
- 14451
- 111

This RegEx accepts exactly those strings that start with a 1 and end with a 1, and have zero or more copies of 45 or 54 in the middle.

(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.

Solution:

Accepted:

- a
- aaaa
- abb
- abbbb
- aaaaaaabbbbbb

Rejected:

- ε
- aa
- aaa
- ab
- abba

This RegEx accepts exactly those strings that start with a run of 'a's with length equivalent to 1 mod 3 followed by an even number of 'b's.

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.

Solution:

RegEx:

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

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

Solution:

RegEx:

$$(aa)^*$$

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

Solution:

RegEx:

$$b^*(b^*ab^*ab^*)^*$$

Alternate solution:

$$b^*(ab^*ab^*)^*$$

The extra b^* isn't necessary because we can add any number of b's we want before the very first a using the b^* before the parentheses. Then, we can use the b^* after the second a to add as many b's as we want before the next pair of a's.

Another solution!

$$(b \cup (ab^*a))^*$$

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

Solution:

RegEx:

$$(a \cup \varepsilon)(ba)^*(b \cup \varepsilon)$$

Alternate solution:

$$(b \cup \varepsilon)(ab)^*(a \cup \varepsilon)$$

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

Solution:

RegEx:

$$(a \cup b)^*(bb \cup ba)$$

Alternate solution:

$$(a \cup b)^* b(b \cup a)$$

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

Solution:

RegEx:

$$\varepsilon \cup a \cup b \cup ((a \cup b)^*(bb \cup ab \cup ba))$$

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 $S \circ R$?

Solution:

$R \circ S = \{(1, 5), (1, 7), (4, 3)\}$
$S \circ R = \{(2, 5)\}$

(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.

Solution:

In R: $(3, 3), (3, 2), (-5, -6)$
Not in R: $(3, 1), (5, 1), (-5, -7)$

(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.

Solution:

- Reflexive: Yes. For any integer a, it is true that $a \leq a+1$. So $(a,a) \in R$.
- Symmetric: No. For example, $(2,20) \in R$ but $(20,2) \notin R$. ($2 \leq 21$, but $20 \nleq 3$)
- Antisymmetric: No. For example $(0,1) \in R$ and $(1,0) \in R$. ($0 \leq 2$ and $1 \leq 1$)
- Transitive: No. For example $(2,1) \in R$ and $(1,0) \in R$, but $(2,0) \notin R$. ($2 \leq 2$ and $1 \leq 1$, but $2 \nleq 1$).

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.

Solution:

False. Consider the following counter example. Let R be $<$ on $\mathbb{Z}$ and let S be $>$ on $\mathbb{Z}$ (i.e., $R = \{(a,b) : a < b \text{ and } a,b \in \mathbb{Z}\}$ and $S = \{(a,b) : a > b \text{ and } a,b \in \mathbb{Z}\}$). R and S are both transitive. Since $(1,2) \in R$ and $(2,1) \in S$, by definition of union, $(1,2)$ and $(2,1)$ are both in $R \cup S$. But, $(1,1) \notin R$ and $(1,1) \notin S$, so $(1,1) \notin R \cup S$. Therefore, the claim is false.

For a similar but smaller example, consider the sets $R = \{(4,5),(5,6),(4,6)\}$ and $S = \{(6,5),(5,4),(6,4)\}$. Both are transitive. $R \cup S = \{(4,5),(5,6),(4,6),(6,5),(5,4),(6,4)\}$. Specifically, $(4,6)$ and $(6,4)$ are both in $R \cup S$, but $(4,4) \notin R \cup S$, so $R \cup S$ is not transitive.

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

Solution:

True. We need to show that $\forall a \in \mathbb{Z}, (a,a) \in R \circ S$. Let $a \in \mathbb{Z}$ be arbitrary. Since R and S are both reflexive, by definition of reflexive, $(a,a) \in R$ and $(a,a) \in S$. Then, by definition of composition, $(a,a) \in R \circ S$. Since a was arbitrary, we have shown that $R \circ S$ is reflexive.

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

Solution:

False. Let $R = \{(a, a+1) : a \in \mathbb{Z}\}$. In other words, $R = \{...(-2,-1),(-1,0),(0,1),(1,2)...\}$. Let $S = \{(a, a-1) : a \in \mathbb{Z}\}$. In other words, $S = \{...(-1,-2),(0,-1),(1,0),(2,1)...\}$. R and S are both not reflexive. For any arbitrary $a \in \mathbb{Z}$, we have $(a,a+1) \in R$ and $(a+1,a) \in S$. So $(a,a) \in R \circ S$, and $R \circ S$ is reflexive. This is a counterexample, so the claim is false.

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

Solution:

True. Since R is symmetric, we know the following.

$$\forall a \forall b \ [(a,b) \in R \to (b,a) \in R]$$

Taking the contrapositive, this is equivalent to:

$$\forall a \forall b \ [(b,a) \notin R \to (a,b) \notin R]$$

By the definition of complement, this is equivalent to:

$$\forall a \forall b\ [(b,a) \in \overline{R} \rightarrow (a,b) \in \overline{R}]$$

This is the definition of $\overline{R}$ being symmetric.

You could also prove this by contradiction as follows: Suppose for contradiction that there is some relation R that is symmetric, but $\overline{R}$ is not symmetric. Since $\overline{R}$ is not symmetric, by definition of symmetric, $\exists (a,b) \in \overline{R}$ such that $(b,a) \notin \overline{R}$. Since $(b,a) \notin \overline{R}$, by definition of complement, $(b,a) \in R$. Since R is symmetric, we must have $(a,b) \in R$. This contradicts our earlier statement that $(a,b) \in \overline{R}$. Therefore, the original claim holds.

6. Context Free Grammars Warmup

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

$$\textbf{S} \rightarrow 0\textbf{X}4$$
$$\textbf{X} \rightarrow 1\textbf{X}3 \mid 2$$

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.

Solution:

Accepted:

- 024
- 01234
- 0112334
- 011123334
- 01111233334

Rejected:

- ε
- 2
- 0244
- 011234
- 10234

This CFG generates all strings of the form $0\ 1^m\ 2\ 3^m\ 4$, where $m \geq 0$. That is, it's all strings made of one 0, followed by zero or more 1's, followed by a 2, followed by the same number of 3's as 1's, followed by one 4.

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: $(aa)^*$

Solution:

CFG:

$$\textbf{S} \rightarrow \textbf{SS} | aa | \varepsilon$$

Alternate CFG:

$$\textbf{S} \rightarrow aa\textbf{S} \mid \varepsilon$$

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

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

Solution:

CFG: Working off of the RegEx, we can create a rule **B** that is equivalent to b^* (allows us to add 0 or more b's) and a separate rule called **A** that is equivalent to $(ab^*ab^*)^*$ (allows us to add an a, then any number of b's, then another a, then any number of b's, then repeat).

$$\mathbf{S} \to \mathbf{BA}$$
$$\mathbf{B} \to \mathbf{BB} \mid b \mid \varepsilon$$
$$\mathbf{A} \to \mathbf{AA} \mid a\mathbf{B}a\mathbf{B} \mid \varepsilon$$

Alternate CFG: This solution corresponds to the alternate RegEx solution.

$$\mathbf{S} \to \mathbf{SS} \mid a\mathbf{S}a \mid b \mid \varepsilon$$

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

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

Solution:

CFG:

$$\mathbf{S} \to a\mathbf{A} \mid b\mathbf{A}$$
$$\mathbf{A} \to \mathbf{AA} \mid aa \mid ab \mid ba \mid bb \mid \varepsilon$$

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

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

Solution:

CFG:

$$\mathbf{S} \to \mathbf{E} \mid \mathbf{O}$$
$$\mathbf{E} \to \mathbf{BC}$$
$$\mathbf{B} \to \mathbf{BB} \mid b \mid \varepsilon$$
$$\mathbf{C} \to \mathbf{CC} \mid a\mathbf{B}a\mathbf{B} \mid \varepsilon$$
$$\mathbf{O} \to \mathbf{A}b\mathbf{AD}$$
$$\mathbf{A} \to \mathbf{AA} \mid a \mid \varepsilon$$
$$\mathbf{D} \to \mathbf{DD} \mid b\mathbf{A}b\mathbf{A} \mid \varepsilon$$

Alternate CFG:

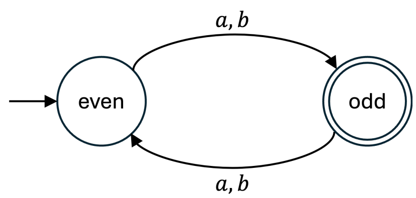
$$\mathbf{S} \to \mathbf{E} \mid \mathbf{O}b\mathbf{O}$$
$$\mathbf{E} \to \mathbf{EE} \mid a\mathbf{E}a \mid b \mid \varepsilon$$
$$\mathbf{O} \to \mathbf{OO} \mid b\mathbf{O}b \mid a \mid \varepsilon$$

8. Constructing DFAs

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

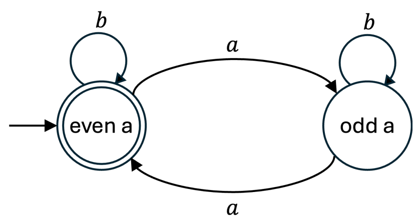
(a) Strings with odd length.

Solution:



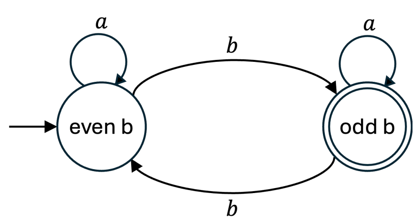
(b) Strings with an even number of a's.

Solution:



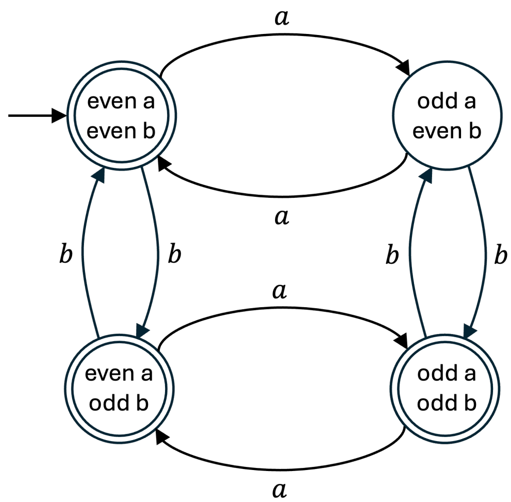
(c) Strings with an odd number of b's.

Solution:



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

Solution:



9. All the Things

Let $\Sigma := \{0, 1, 2, 3, 4, 5\}$. For an arbitrary string x over Σ, we can write $x = x_0x_1 \cdots x_n$, where $x_0, x_1, ..., x_n \in \Sigma$. Define a language L over Σ 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 Σ.

Solution:

Accepted:

- 145
- 135
- 12425
- 2004
- 2034

Rejected:

- 321
- 11
- 455
- 452
- 2010

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

Solution:

$(0 \cup 2 \cup 4)^*(\varepsilon \cup 1)(2 \cup 4)^*(\varepsilon \cup 3)4^*(\varepsilon \cup 5)$

(c) Construct a CFG for the language L.

Solution:

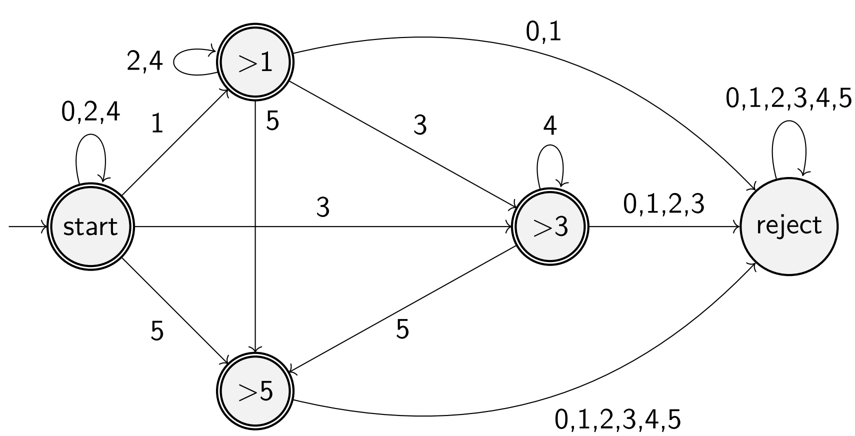
$$\begin{aligned}
\textbf{S} &\rightarrow \textbf{ABCDEF} \\
\textbf{A} &\rightarrow \textbf{AA} \mid 0 \mid 2 \mid 4 \mid \varepsilon \\
\textbf{B} &\rightarrow 1 \mid \varepsilon \\
\textbf{C} &\rightarrow \textbf{CC} \mid 2 \mid 4 \mid \varepsilon \\
\textbf{D} &\rightarrow 3 \mid \varepsilon \\
\textbf{E} &\rightarrow \textbf{EE} \mid 4 \mid \varepsilon \\
\textbf{F} &\rightarrow 5 \mid \varepsilon
\end{aligned}$$

Alternate Solution:

$$\begin{aligned}
\textbf{S} &\rightarrow 0\textbf{S}|2\textbf{S}|4\textbf{S}|\textbf{A} \\
\textbf{A} &\rightarrow 1\textbf{B}|\textbf{B} \\
\textbf{B} &\rightarrow 2\textbf{B}|4\textbf{B}|\textbf{C} \\
\textbf{C} &\rightarrow 3\textbf{D}|\textbf{D} \\
\textbf{D} &\rightarrow 4\textbf{D}|\textbf{E} \\
\textbf{E} &\rightarrow 5|\varepsilon
\end{aligned}$$

(d) Construct a DFA for the language L.

Solution:



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.

Solution:

First we observe that the language defined by this CFG can be represented by the following recursively defined set S:

Basis Rule: $\varepsilon \in S$

Recursive Rule: If $x, y \in S$, then $xy, 0x1, 1x0 \in S$.

Proof. For a string t, let P(t) be defined as "$\#_0(t) = \#_1(t)$". We will prove P(t) is true for all strings $t \in S$ by structural induction.

Base Case $(t = \varepsilon)$: By definition, the empty string contains no characters, so $\#_0(t) = 0 = \#_1(t)$

Inductive Hypothesis: Suppose P(x) and P(y) hold for arbitrary strings $x, y \in S$.

Inductive Step:

Case 1: Goal: show $P(0x1)$.

$$\begin{aligned} \#_0(0x1) &= \#_0(x) + 1 && \text{[Def of } \#_0(x)] \\ &= \#_1(x) + 1 && \text{[By IH]} \\ &= \#_1(0x1) && \text{[Def of } \#_1(x)] \end{aligned}$$

Therefore $\#_0(0x1) = \#_1(0x1)$. This proves P$(0x1)$.

Case 2: Goal: show $P(1x0)$

$$\begin{aligned} \#_0(1x0) &= \#_0(x) + 1 && \text{[Def of } \#_0(x)] \\ &= \#_1(x) + 1 && \text{[By IH]} \\ &= \#_1(1x0) && \text{[Def of } \#_1(x)] \end{aligned}$$

Therefore $\#_0(1x0) = \#_1(1x0)$. This proves P$(1x0)$.

Case 3: Goal: show $P(xy)$

$$\begin{aligned} \#_0(xy) &= \#_0(x) + \#_0(y) && \text{[Def of } \#_0(x)] \\ &= \#_1(x) + \#_1(y) && \text{[By IH]} \\ &= \#_1(xy) && \text{[Def of } \#_1(x)] \end{aligned}$$

Therefore $\#_0(xy) = \#_1(xy)$. This proves P(xy).

So by structural induction, $\mathsf{P}(t)$ is true for all strings $t \in S$. $\square$

Since the recursively defined set, S, is exactly the set of strings generated by the CFG, we have proved that the statement is true for every string generated by the CFG too.