Istanbul Technical University
Department of Computer Engineering
BLG 311E Formal Languages and Automata
Spring 2024

Assignment 3

For any questions, please contact:
Res. Assist. Meral Kuyucu korkmazmer@itu.edu.tr
Res. Assist. Batuhan Can canb18@itu.edu.tr

May 11, 2024

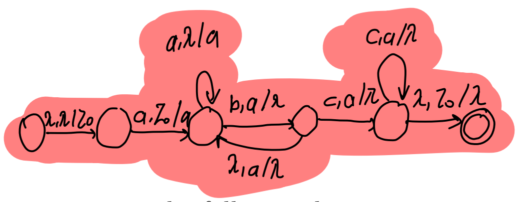
Problem 1

Either prove or disprove the following languages are regular or irregular.

(a) $L = \{a^n b^m : n \neq m\}$

(b) $L = \{0^n 1^m : n > m\}$

(c) $L = \{cc^r : c \in \{0, 1\}^*\}$



Problem 2

Design a pushdown automaton (PDA) that recognizes the following language:

$$L(G) = \{a^k b^m c^n \mid k, m, n > 0 \text{ and } k = 2m + n\}$$

$S \to aAA \to abSa$

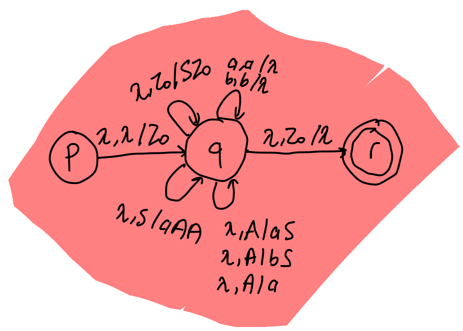
abaAAa

abaaaa

Problem 3

Convert the given CFG below to PDA.

$$S \to aAA$$
$$A \to aS \mid bS \mid a$$



Problem 4

Convert the given PDA below to CFG.

$$\delta(q, 1, Z) = (q, XZ)$$
$$\delta(q, 1, X) = (q, XX)$$
$$\delta(q, 0, X) = (q, X)$$
$$\delta(q, \lambda, X) = (q, \lambda)$$
$$\delta(q, 0, Z) = (q, Z)$$

$[qZq] \to 1\,[qXq]\,[qZq]$

$[qXq] \to 1\,[qXq]\,[qXq]$

$[qXq] \to 0\,[qXq]$

$[qXq] \to \lambda$

$[qZq] \to 0\,[qZq]$

$[qZq] = S$

$[qXq] = A$

$S \to 1AS \mid 0S$

$A \to 1AA \mid 0A \mid \lambda$

Problem 5

Claim if the following language can be recognized by a pushdown automata (PDA). Prove/rationalize your claim.

$$L(G) = \{a^i b^k a^i b^k \mid i, k > 0\}$$

No

Problem 6

Prove the language NO below is not context-free.

$$L = \{0^n 1^n 2^n \mid n \geq 0\}$$

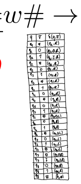
Problem 7

Consider a Turing Machine (TM) designed to operate on inputs from the alphabet $\{0, 1\}$. This TM is tasked with duplicating the input string. An example execution is provided below, where w represents any string in $\{0, 1\}^*$:

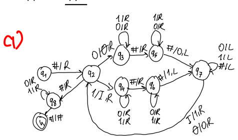
$$\underline{\#}w\# \rightarrow \#w\#w\underline{\#}$$

Compute the following:

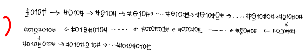
b)



a)



c)



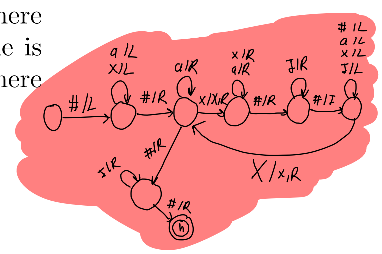
(a) State transition diagram.

(b) State transition table.

(c) Demonstrate the steps for the following execution: $\underline{\#}010\# \rightarrow \#010\#010\underline{\#}$.
Note: You may introduce additional characters to the working alphabet if deemed necessary for your solution.

Problem 8

Design a Turing Machine (TM) that counts the number of a's in a given input string. It should do this by writing a string containing as many I's as there are a's following the input string. The initial configuration of the machine is $\#xax \ldots xax \ldots \underline{\#}$ where x's represent symbols other than a's. Assuming there are a total of n a's, the transformation can be represented as:



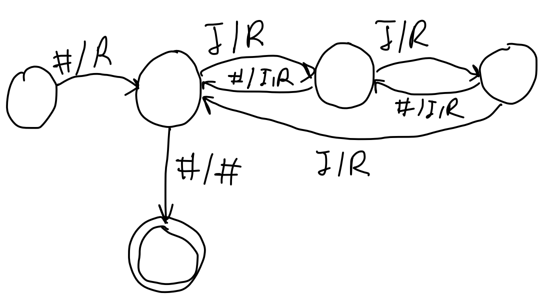
$$\#xax \ldots xax \ldots \underline{\#} \rightarrow \#xax \ldots xax \ldots \# \underbrace{\text{IIIIIII}}_{\text{n}} \underline{\#}$$

Note: You may assume that $\forall x : x \neq \gamma$

Problem 9

Design a Turing machine that calculates the modulo 3 equivalent of a given number represented using I's, where the count of I's indicates the number itself. Initially, the tape head will be positioned on the blank symbol # before the number. At the completion of execution, the machine will append I's to the number to represent the result.
The following example demonstrates the initial and final configurations for $5 \equiv 2 \pmod 3$.

$$\underline{\#}\underbrace{\text{IIIII}}_{5}\#\underbrace{\text{IIIIIII}}_{5\,+\,2}\underline{\#}$$



Eliminate Lambda Productions

$S \to AB \mid BC \mid Bc \mid B$
$A \to a$
$B \to b$
$C \to c$

Eliminate Unit Productions

$S \to AB \mid BC \mid b$
$A \to a$
$B \to b$
$C \to c$

There is no at least two non-terminal on the right side.

Problem 10

Consider the following context-free grammar (CFG) G, with start symbol S:

$$S \to AB|BC$$

$$A \to a|\lambda$$

$$B \to b$$

$$C \to c|\lambda$$

Convert the given CFG G into Chomsky Normal Form (CNF). Provide the resulting grammar with all necessary productions. Explain how you applied the CNF conversion rules. Show how the string "ab" can be derived using the CNF grammar, if possible.

$S_1 = S_2$ is possible.

only 1 production each line.

$S_1 \to$
$S_2 \to$
⋮
$S_n \to$

Problem 11

Suppose we have a grammar G with n productions, none of them λ-productions, and convert this grammar to CNF. Show that the CNF grammar has at most $O(n^2)$ productions.

Problem 12

Design a context-free language L such that any string in L consists of a balanced combination of parentheses, square brackets, and curly braces. Then, demonstrate how to apply the Pumping Lemma for Context-Free Languages to prove that L is indeed context-free. Additionally, provide an explanation of how the Pumping Lemma could be used to show that a language containing strings of the form $\{a^n\ b^n\ c^n \mid n \geq 0\}$ is not context-free.

Problem 13

Design a Turing decider M that accepts the language L defined as follows:

L={w | w is a binary string representing a number divisible by 3}

Your Turing decider should take as input a binary string and halt in an accepting state if the input represents a number divisible by 3, and halt in a rejecting state otherwise.

You may assume that the input binary string will be given on the tape, with the head initially positioned at the leftmost bit of the input.