# CS 205 - Problem Set 6 - Context-Free Grammar, FSA, NFA, Boolean Algebra

Sections 01-03 and 07-09

Due Date, Sunday December 7, 2022 in gradescope

Instructions

- Read Rosen Chapter 13, cubits and related lecture notes prior to completing this problem set
- This is an individual worksheet. NO group work is permitted.
- You are authorized to seek help from course staff ONLY
- This handout is also available on gradescope
- Submit solutions to gradescope
- No late submissions accepted

your work may not be graded without you signing below
I certify that this paper represents my own work and I have read RU academic integrity policies
https://www.cs.rutgers.edu/academic-integrity/introduction

**Sign and PRINT your name :** Akshaj Kammari

**Your recitation section (1-3,7-9) and/or day of the week :** 7 (tuesday)

**NetID :** ak1990

## Problem 1 - Context-Free Grammar

A palindrome is a string that reads the same backward as it does forward. For example, abaaaba is a palindrome. Suppose that we need to define a language that generates palindromes.

1. Define a context-free grammar that generates the set of all palindromes over the alphabet $\{a, b\}$ clearly describing the recursive rules that generates palindromes. Use the notation $Symbol \to rule$. The empty set is denoted by $\lambda$.

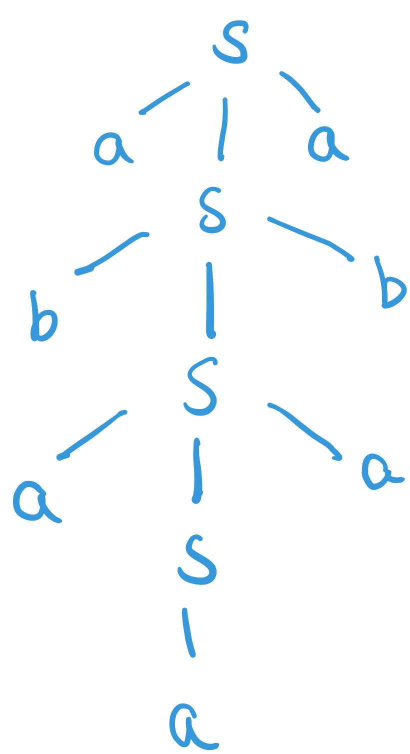
$S \to \lambda, a, b, asa, bsb$

2. Show the set of terminal symbols and the set of non-terminal symbols in your grammar.

terminal : $\{a, b, \lambda\}$

non-terminal : $\{S\}$

3. Show that the palindrome *abaaaba* can be recognized by your grammar. To show this, show all steps of parsing the expression *abaaaba* using the rules you defined above.



## Problem 2 - FSM

Suppose you are constructing a finite-state machine for entering a security code into an automatic teller machine (ATM). The following rules are implemented.

- A user enters a string of four digits. You can consider PIN as a 4-digit number that is valid or invalid. (there is no need to consider one digit at a time. Just consider the input as code or nocode )
- If the user enters the correct four digit PIN, the ATM displays a welcome screen.
- When the user enters an incorrect PIN, the ATM displays a screen that informs the user that an incorrect PIN was entered.
- If a user enters the incorrect PIN three times, the account is locked.

Answer the following questions.

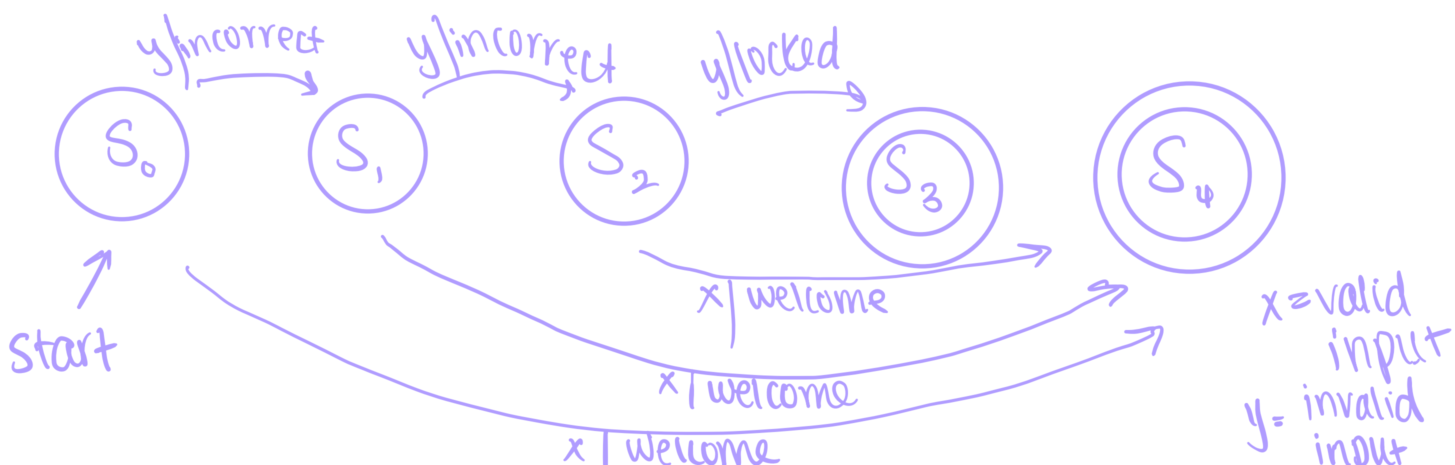
1. What is the input alphabet for this machine?

$\{n, n, n, n\}$ $n = \{0|1|2|3|4|5|6|7|8|9\}$

2. What is the output alphabet for this machine?

{ welcome screen, incorrect, locked }

3. Draw the FSM clearly identifying all states, including start state and end state(s).

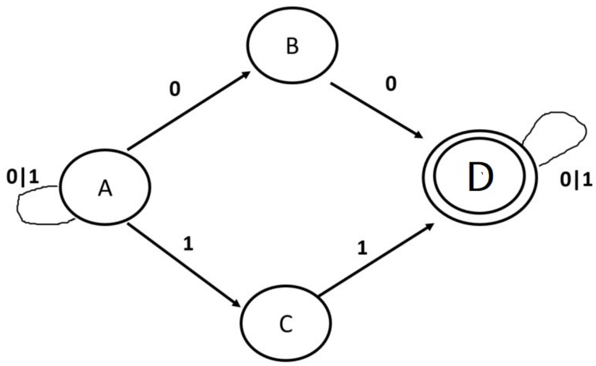



4. Complete the transition table in the format discussed in lectures.That is a table of states (rows) and inputs (columns) and (state, output) pairs as table entries.

| | valid input | invalid input |
|---|---|---|
| $S_0$ | $S_4$, welcome | $S_1$, incorrect |
| $S_1$ | $S_4$, welcome | $S_2$, incorrect |
| $S_2$ | $S_4$, welcome | $S_3$, locked |
| $S_3$ | — | — |
| $S_4$ | — | — |

## Problem 3 - NFA to DFA

Consider the following NFA



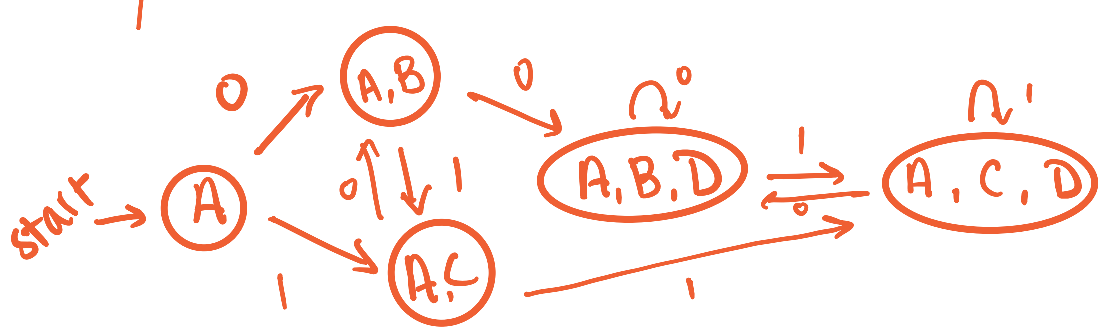

1. Draw the transition table for this NFA

| | 0 | 1 |
|---|---|---|
| A | {A,B} | {A,C} |
| B | {D} | – |
| C | – | {D} |
| D | {D} | {D} |

2. Find a DFA equivalent to this NFA

| | 0 | 1 |
|---|---|---|
| {A} | {A,B} | {A,C} |
| {A,B} | {A,B,D} | {A,C} |
| {A,C} | {A,B} | {A,C,D} |
| {A,B,D} | {A,B,D} | {A,C,D} |
| {A,C,D} | {A,B,D} | {A,C,D} |

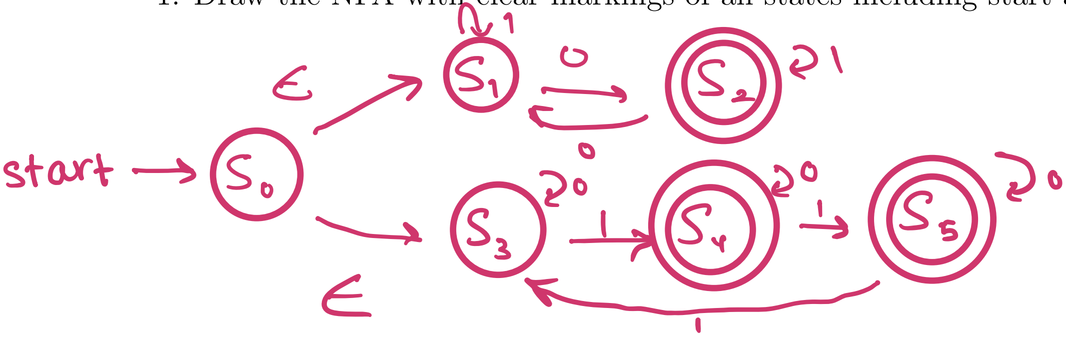
DFA transition table



## Problem 4 - NFA construction and Simulation

Consider an NFA that recognize the set of all binary strings that have either the number of 0's odd, or the number of 1's not a multiple of 3, or both.

1. Draw the NFA with clear markings of all states including start and acceptance state(s).




2. Simulate the NFA to show that string 01001 will be accepted by the NFA

| string | possible states |
|---|---|
| $\lambda$ | $\{S_1, S_3\}$ |
| $\lambda 0$ | $\{S_2, S_3\}$ |
| $\lambda 01$ | $\{S_2, S_4\}$ |
| $\lambda 010$ | $\{S_1, S_4\}$ |
| $\lambda 0100$ | $\{S_2, S_4\}$ |
| $\lambda 01001$ | $\{S_2, S_5\}$ |

Since the last accepted string results in the possible states $\{S_2, S_5\}$ 01001 must be accepted by the NFA.

3. Simulate the NFA to show that string 10101 will not be accepted by the NFA

| string | possible states |
|---|---|
| $\lambda$ | $\{S_1, S_3\}$ |
| $\lambda 1$ | $\{S_1, S_4\}$ |
| $\lambda 10$ | $\{S_2, S_4\}$ |
| $\lambda 101$ | $\{S_2, S_5\}$ |
| $\lambda 1010$ | $\{S_1, S_5\}$ |
| $\lambda 10101$ | $\{S_1, S_3\}$ |

The possible states $\{S_1, S_3\}$ are not accepted as the last string. therefore 10101 is not accepted by the NFA.

## Problem 5 - Boolean Algebra

1. Suppose there are 3-sensors and each sends a binary signal 0 or 1. Design a function that receives 1 if at least 2 of the sensors send 1. Provide a sum-of-the-products representation of the function $F(x, y, z)$ where $x, y, z$ represents the three sensors. Show all work to receive full credit.

$F(x,y,z) = xy\bar{z} + x\bar{y}z + \bar{x}yz + xyz$

| $x$ | $y$ | $z$ | $xy\bar{z}$ | $x\bar{y}z$ | $\bar{x}yz$ | $xyz$ | $xy\bar{z} + x\bar{y}z + \bar{x}yz + xyz$ |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

2. Find the sum of product representation of the function $F(x, y, z)$ that is equal to 1 if and only if x + z = 1 (where + is the OR operator)

$F(x,y,z) = x\bar{z} + \bar{x}z + xz$

$$\boxed{F(x,y,z) = xy\bar{z} + \bar{x}yz + xyz + x\bar{y}\bar{z} + x\bar{y}z + x\bar{y}z}$$

3. The nor operator, denoted by $\downarrow$, is defined by $0 \downarrow 0 = 1$, and $1 \downarrow 0 = 0 \downarrow 1 = 1 \downarrow 1 = 0$. Prove that the set consisting of just the one operator nor $\downarrow$ is functionally complete. That is show that operators, $., +, \overline{x}$ can be obtained from $\downarrow$

| $x$ | $y$ | $x \cdot y$ | $(x\downarrow x)\downarrow(y\downarrow y)$ |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 |

+

| $x$ | $y$ | $x+y$ | $(x\downarrow y)\downarrow(y\downarrow x)$ |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 |

— complement

| $x$ | $\bar{x}$ | $x \downarrow x$ |
| --- | --- | --- |
| 1 | 0 | 0 |
| 0 | 1 | 1 |

$\therefore \; x \cdot y = (x \downarrow x) \downarrow (y \downarrow y)$

$x + y = (x \downarrow y) \downarrow (y \downarrow x)$

$\bar{x} = x \downarrow x$

So, $\downarrow$ is functionally complete