

2202 COL 352 Minor2

CHINMAY MITTAL

TOTAL POINTS

32 / 35

QUESTION 1

15 pts

1.1 NPDA 5 / 5

+ 0 pts Incorrect/Not Attempted

✓ + 5 pts *Correct NPDA*

+ 3 pts Partially correct NPDA

1.2 Middle letter 5 / 5

+ 0 pts Incorrect / not attempted

+ 2 pts Partially correct.

✓ + 5 pts *Correct*

1.3 Turing decidable 5 / 5

✓ + 5 pts *correct answer with proper proof*

+ 3 pts correct answer with partial proof

+ 1 pts correct answer with no proof

+ 0 pts incorrect answer/ unanswered

QUESTION 2

2 Minimal DFA 10 / 10

+ 0 pts Incorrect

+ 1 pts Only NFA

+ 2.5 pts Only DFA

+ 4 pts Correct DFA and correct minimization

steps

+ 7 pts Use of Myhill Nerode

Theorem/equivalence classes with complete

arguments

+ 5 pts Use of Myhill Nerode

Theorem/equivalence classes but some

incomplete/incorrect arguments

✓ + 10 pts *Correct with proof*

QUESTION 3

3 CFL quotient Regular 7 / 10

+ 7 pts Correct NPDA

✓ + 4 pts *Partially correct NPDA*

+ 0 pts Incorrect NPDA / not attempted

✓ + 3 pts *Correct proof / explanation for given NPDA*

+ 0 pts Incorrect explanation / not attempted

💬 You needed to properly define the NPDA.
Here, you have just described it.

Name: CHINNAY MITTAL

Roll No: 2020CS10336

(COL 352) Introduction to Automata and Theory of Computation

Mar 23, 2023

Minor 2

Duration: 60 minutes

(35 points)

Beware: Be clear in your writing. If you use a statement proved in class or in the problem set, then write down the entire statement before using it. You will not get a new sheet, so make sure you are certain when you write something (maybe use a dark pencil). Make a judicious decision of which tool(s) to use to get a clean and short answer that fits in the space. If you cheat, you will surely get an F in this course.

1. ($3 \times 5 = 15$ points) Each of the following questions carry 5 points.

- (a) Give an equivalent NPDA for the grammar $G = (\{S, A\}, \{a\}, \{S \rightarrow aAA; A \rightarrow aS \mid bS \mid a\}, \{S\})$. Is $L(G)$ regular?

$\{a, b\}^*$

We can use the construction mentioned in class to create the NPDA. The initial stack will have only the start symbol.

Whenever we read the input, if it matches with the stack top we can simply pop it from the stack.

If the top of stack is a non-terminal we can non-deterministically read ϵ , pop the non-terminal from the stack top and push a production.

$$\Gamma = \{a, b\} \cup \{S, A\}$$

$$\Sigma = \{a, b\}$$

$$Q = \{q\} \text{ (single state)}$$

$$\delta \rightarrow \delta(q, a, a) \rightarrow (q, \epsilon)$$

$$\delta(q, b, b) \rightarrow (q, \epsilon)$$

$$\delta(q, \epsilon, S) \rightarrow (q, aAA)$$

$$\delta(q, \epsilon, A) \rightarrow (q, aS) \text{ or } (q, bS) \text{ or } (q, a)$$

Acceptance by empty stack.

$L(G)$ is regular we can create a DFA to simulate the non-terminals

(b) Prove or disprove: $L = \{w \in \{a, b\}^* \mid w \text{ has odd length and the middle letter is } a\}$ is a CFL.

True. we will create a NPDA for this language which will try to guess the position of the middle letter.

In the initial state, all letters will be pushed to stack.

if in the initial state and we encounter an a then we non-deterministically assume that it was the middle letter. whether it was part of first part of the string either we remain in the state or we change the state to q_2 (we will read the second part of the string here). When we are in q_2 for every input letter we will pop from the stack, if stack becomes empty before input is read \Rightarrow reject. if stack becomes empty and all input is read \Rightarrow accept. if stack non empty and all input is read \Rightarrow reject.

(c) Prove or disprove: Every CFL is Turing-decidable.

True. given a string to check whether it is part of the CFL or not we can use the CYK algorithm. The CYK algorithm deterministically terminates after some steps (not infinite) on every input and produces a YES/NO result. The CYK Algorithm essentially uses memory, for loops and if conditions (independent of the input) all of which can be encoded into a Turing Machine. thus TM will halt on every input and produce a YES/NO answer.
 \Rightarrow CFLs are Turing decidable

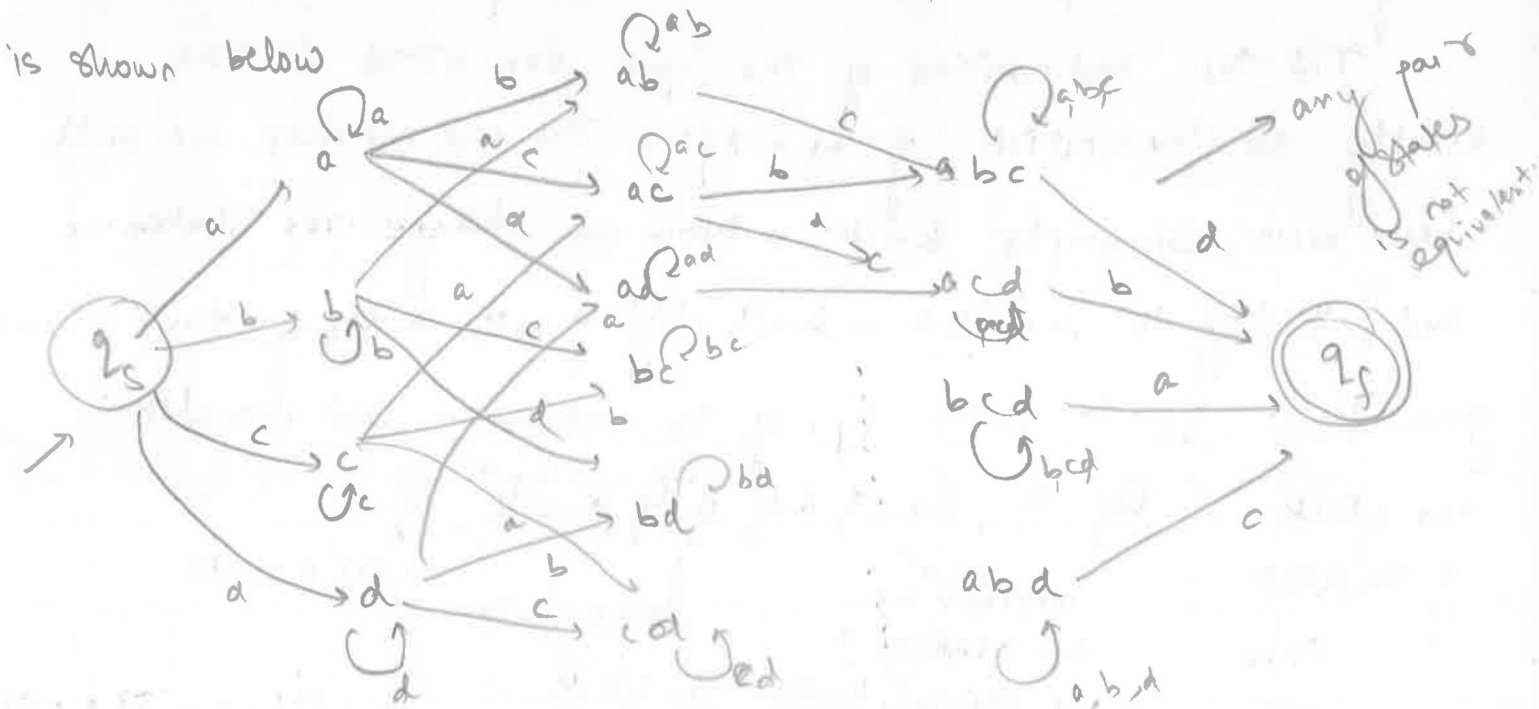
Name: CHINMAY MITTAL

Roll No: 2020010336

2. (10 points) Find the minimal DFA for

$L = \{w \in \{a, b, c, d\}^* \mid |w| > 1 \text{ and the last letter in } w \text{ does not appear anywhere else in } w\}$

We will create a DFA and prove that it is minimum. There is one start state and one reject state. The other states encode the letters that we read till now. The DFA is shown below



essentially $Q \xrightarrow{\alpha} Q \cup \{ \alpha \}$ and $Q \xrightarrow{\alpha} F$ only if $\alpha \cap Q = \{ \alpha \}$ and $Q \cap \alpha = \{ \alpha \}$. $Q_s = \{ \alpha \}$

Consider any two states. We show that a string exists such that one state goes to accept and one state goes to reject. if both the states are non empty then there exists a symbol α which WLOG $\alpha \in q_1$ and $\alpha \notin q_2 \Rightarrow$ on reading α q_2 accepts and q_1 rejects \Rightarrow not equivalent. if $q_1 = q_s$ and $q_2 \neq q_s$ ready any letter not in $q_2 \Rightarrow q_1$ rejects and q_2 accepts \Rightarrow not equivalent.

3. (10 points) Let $L_1, L_2 \subseteq \Sigma^*$. Let $\frac{L_1}{L_2} = \{x \mid \exists y \in L_2, xy \in L_1\}$. Show that if L_1 is a CFL and L_2 is a regular language then $\frac{L_1}{L_2}$ is a CFL.

he will get a string x as input essentially we want to show that can we find a string $y \in L_2$ such that $xy \in L_1$. We create a NPDA for $\frac{L_1}{L_2}$ using the NPDA for L_1 and the DFA for L_2 .

Till the end marker of the input our NPDA functions exactly as the NPDA of L_1 . After the end marker we will have some non empty stack. Now we change our state and will try to non deterministically guess a string that goes from q_0 to some q_f of the automaton and also clears the stack.

$$Q = \{q_0, 1\} \times \{q' \mid q' \in Q\} \times \{q'' \mid q'' \in Q\}$$

of the NPDA. whether we are reading x or trying to clear the stack. NPDA state DFA state

When we are reading x , transitions are according to NPDA and DFA state does not change. When we are reading y , at all states we can create transitions for all possible input letters.

The accepting criterion will be the following \rightarrow We are reading y and stack becomes empty and we are in the final state of the DFA.