# 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: CHINMAY 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 × 5 = 15 points)** Each of the following questions carry 5 points.

   (a) Give an equivalent NPDA for the grammar $G = (\{S, A\}, \{a\}, \{S \to aAA; A \to aS \mid bS \mid a\}, \{S\})$. 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 $\epsilon$, pop the non terminal from the stack top and push a production.

$\Gamma = \{a, b\} \cup \{S, A\}$. $\Sigma = \{a, b\}$. $Q = \{q\}$ → single state

$\delta \to$
$$\delta(q, a, a) \to (q, \epsilon)$$
$$\delta(q, b, b) \to (q, \epsilon)$$
$$\delta(q, \epsilon, S) \to (q, aAA)$$
$$\delta(q, \epsilon, A) \to (q, aS) \text{ or } (q, bS) \text{ or } (q, a)$$

Acceptancy by empty stack.

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

(b) Prove or disprove: $L = \{w \in \{a,b\}^* \mid w$ 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 than we non determenestically assume that it was the middle | 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 $q2$ for every input letter we will pop from the stack, if stack becomes empty before input is read ⇒ reject.

if stack becomes empty and all input is read ⇒ accept

if stack non empty and all input is read ⇒ reject.

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

True. given all string to check whether it is part of the CFL or not we can use the CKY algorithm. The CKY algorithm detemenistically terminates after some steps (not infinite) on every input and produces a YES|NO result. The CKY 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 correctly ⇒ CFLs are Turing decidable

↓ ∈L ↳ ∉L

Name: CHINMAY MITTAL Roll No: 2020CS10336.

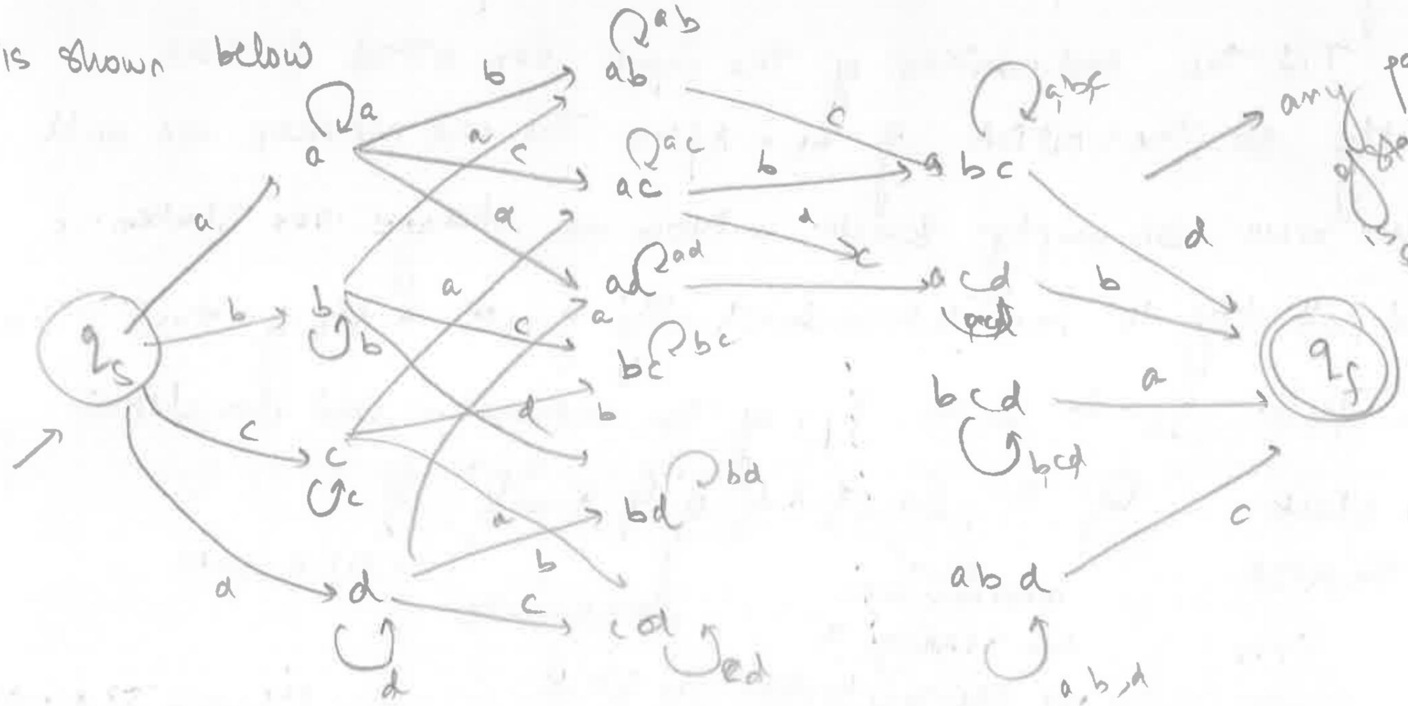
2. (10 points) Find the minimal DFA for

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

We will create a DFA and prove that it is minimum → no two states are equivalent +

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

→ any pair of states is not equivalent




essentially $Q \xrightarrow{\sigma} Q \cup \{\sigma\}$ $\sigma \in \{a, b, c, d\}$

and $Q \xrightarrow{\sigma} F$ $q_s = \{\}$

only if $\sigma \cap Q = \{\}$ and $Q \,!= \{\}$.

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 $(q_1, q_2)$ are non empty then there exists a symbol $\sigma$ which WLOG $\sigma \in q_1$ and $\sigma \notin q_2$ ⇒ on reading $\sigma$ $q_2$ accepts and $q_1$ rejects ⇒ not equivalent

if $q_1 = q_s$ and $q_2 \neq q_s$ reading any letter not in $q_2$ ⇒ $q$ rejects and $q_2$ accepts ⇒ 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.

We will get a string $x$ as input essentially we want to show that can we find a string $\in L2$ such that $xy \in L1$. We create a NPDA for $\frac{L1}{L2}$ using the NPDA for L1 and the DFA for L2.

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 of the NPDA.

$Q = \{0,1\} \times \{Q'\} \times \{Q^2\}$

$\{0,1\}$ → whether we are reading $x$ or trying to clear the stack.

$\{Q'\}$ → NPDA state

$\{Q^2\}$ → 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 → We are reading $y$ and stack becomes empty and we are in the final state of the DFA.