Practice Final Exam

(8:45 AM – 10:45 AM : 120 Minutes)

- This exam will account for 30% of your overall grade.
- There are seven (7) questions, worth 120 points in total. Please answer all of them.
- This is a *closed book*, *closed notes* exam. *No cheat sheets* are allowed.
- You are allowed to *use scratch papers* for your calculations.
- You are *not allowed to use your own calculator*. A scientific calculator will be available inside the Respondus Lockdown Browser.

Good Luck!

Question	Parts	Points
1. Construct CFG	–	15
2. Construct CFG from DFA	–	10
3. Convert CFG to CNS	–	15
4. Non-CFL	–	20
5. Construct TM	–	20
6. Easy or Hard?	(a)–(b)	$5 + 10 = 15$
7. True or False?	(a)–(e)	$5 \times 5 = 25$
Total		120

Question 1. [15 Points] Construct CFG. Write down a Context-free Grammar (CFG) to accept the following language.

$$L = \{w \mid w \in \Sigma^* \text{ and } n_a(w) = n_b(w) = n_c(w)\}, \quad \Sigma = \{a, b, c\},$$

where, $n_x(w)$ represents the number of occurrences of symbol x in string w.

This language is not context-free.We already know that

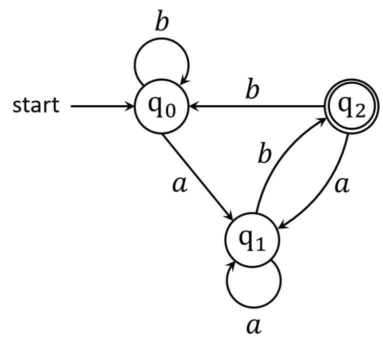
$$L1 = \{w \mid w \in \Sigma^* \text{ and } a^n b^n c^n\}, \quad \Sigma = \{a, b, c\},$$

is not a CFL. We also know that

$$L2 = \{w \mid w \in \Sigma^* \text{ and } a^* b^* c^*\}, \quad \Sigma = \{a, b, c\},$$

is a regular language and therefore is a CFL. Now, we assume that L is a CFL. So, $L1 = L \cap L2$ must be a CFL. Contradiction! Therefore, L is not a CFL

QUESTION 2. [10 Points] Construct CFG from DFA. Convert the following Deterministic Finite Automata (DFA) to an equivalent CFG assuming $\Sigma = \{a, b\}$.



$$S \rightarrow aA \mid bS$$

$$A \rightarrow aA \mid bB$$

$$B \rightarrow aA \mid bS \mid \epsilon$$

QUESTION 3. [**15 Points**] **Convert CFG to CNS.** Convert the following CFG to Chomsky Normal Form (CNF). Please show the transformation step by step (as shown in the class).

$$\begin{aligned} S &\rightarrow ASA \mid A \mid \epsilon \\ A &\rightarrow aa \mid \epsilon \end{aligned}$$

Step 1: Remove start nonterminal from RHS:

$$\begin{aligned} S_0 &\rightarrow S \\ S &\rightarrow ASA \mid A \mid \epsilon \\ A &\rightarrow aa \mid \epsilon \end{aligned}$$

Step 2: Remove any productions that lead to ϵ:

$$\begin{aligned} S_0 &\rightarrow S \mid \epsilon \\ S &\rightarrow ASA \mid AS \mid SA \mid AA \mid A \\ A &\rightarrow aa \end{aligned}$$

Step 3: Add non-terminals for terminals

$$\begin{aligned} S_0 &\rightarrow S \mid \epsilon \\ S &\rightarrow ASA \mid AS \mid SA \mid AA \mid A \\ A &\rightarrow MM \\ M &\rightarrow a \end{aligned}$$

Step 4: Drop unit rules

$$\begin{aligned} S_0 &\rightarrow ASA \mid AS \mid SA \mid AA \mid MM \mid \epsilon \\ S &\rightarrow ASA \mid AS \mid SA \mid AA \mid MM \\ A &\rightarrow MM \\ M &\rightarrow a \end{aligned}$$

Step 5: CNF rules

$$\begin{aligned} S_0 &\rightarrow AN \mid AS \mid SA \mid AA \mid MM \mid \epsilon \\ S &\rightarrow AN \mid AS \mid SA \mid AA \mid MM \\ A &\rightarrow MM \\ N &\rightarrow SA \\ M &\rightarrow a \end{aligned}$$

Question 4. [**20 Points**] **Non-CFL.** Use the pumping lemma to show that the following is not a Context-free Language (CFL).

$$L = \{a^{\triangle n} \mid n \geq 0,\ \triangle n = 1 + 2 + 3 + \ldots + n,\ \triangle 0 = 0\}, \quad \Sigma = \{a\}$$

Solution:
- Assume L is CFL. Then it must satisfy the pumping property.
- Let $s =$ Pumping Length
- Let $w = a^{\triangle s}$
- Let $w = uvxyz$, $u = \epsilon$, $v = a^p$, $x = \epsilon$, $y = \epsilon$, $z = a^{\triangle s - p}$ where $0 < p < s$
- $\mid vy \mid \; = p \geq 1$ and $|vxy| = p \leq s$
- Then $uv^i xy^i z$ must belongs to L for all integer $i \geq 0$.
- However$uxz = a^{\triangle s - p}$, which is not in L

This is a contradiction to our assumption that L is CFL! Hence, L is not CFL.

QUESTION 5. **[20 Points] Construct TM.** Construct a Turing Machine (TM) for identifying the following language. Write down the TM in the 6-tuple format.

$$L = \{\text{strings containing the substring } baba\}, \quad \Sigma = \{a, b\}$$

You do not need to draw the state transition diagram.

6-Tuple: $M = (Q, \Sigma, \Gamma, \delta, q_0, H)$, where,

Set of states is	$Q = \{Q_0, Q_1, Q_2, Q_3, Q_{acc}\}$
Set of strings' symbols is	$\Sigma = \{a, b\}$
Set of tape's symbols is	$\Gamma = \{a, b, \triangleright, \square\}$
Start state is	$q_0 = Q_0$
Set of halting states is	$H = \{Q_{acc}\}$
Transition function is	

δ:

	a	b	$\triangleright$	$\square$
Q_0	$(Q_0, \rightarrow)$	$(Q_1, \rightarrow)$	$(Q_0, \rightarrow)$	$-$
Q_1	$(Q_2, \rightarrow)$	$(Q_1, \rightarrow)$	$-$	$-$
Q_2	$(Q_0, \rightarrow)$	$(Q_3, \rightarrow)$	$-$	$-$
Q_3	$(Q_{acc}, \rightarrow)$	$(Q_1, \rightarrow)$	$-$	$-$

Question 6. [**15 Points**] **Easy or Hard?** From a CSE 303 class of n students we want to select the largest possible number of students such that the SBU IDs of no two chosen students have a common substring of length more than three. For example, if 112252235 and 121225703 are the SBU IDs of two students then both of them cannot be chosen simultaneously as their SBU IDs share a substring of length 4 (i.e., 1225).

(a) [**5 Points**] Which problem that we discussed in the class does this problem remind you of?

(b) [**10 Points**] Is this problem NP-hard or solvable in polynomial time (**5 points**)? In one or two sentences write down the main idea supporting your answer (**5 points**).

a. This is nothing but the *Independent Set* problem. Every SBU ID represents a node and there is an edge between two nodes provided the corresponding SBU IDs share a substring of length larger than 3. The goal is to select the largest possible set of nodes such that no two selected nodes share an edge. Clearly, the selected nodes correspond to a maximum independent set in that graph.

b This problem is NP-hard. This is because one can show that if one has a polynomial time algorithm for solving this problem one can also solve the independent set problem in polynomial time which is already known to be NP-complete. Given an instance of the independent set problem one can transform that into an instance of the problem given in the question in polynomial time [1]

[1] [**You do not need to add the following part**]. *We can do this by starting with an empty string (SBU ID) for each node and then adding a unique common string of length 4 (e.g., xxxx for some $x \in \Sigma$, where $|\Sigma|$ = number of edges in the graph corresponding to the given instance of the independent set problem) to the endpoints of every edge.*

Question 7. [**25 Points**] **True or False?** State if the following statements are true or false (**2 points** each). Justify each of your answers in one or two sentences (**3 points** each).

(a) [**5 Points**] The following language is Turing-decidable: $L = \{\text{programs that do not halt}\}$.

(b) [**5 Points**] Context-free Grammar is Turing-complete.

(c) [**5 Points**] A TM never halts when deciding a Turing-semidecidable language.

(d) [**5 Points**] The set of all algorithms is countable.

(e) [**5 Points**] At least two problems are known that are in NP but not in P.

a. False. This is because, we know that If L is a Turing-decidable language, then $\overline{L}$ is a Turing-decidable language, too. So, if the given problem is Turing-decidable then its complement, i.e., the *Halting Problem*, will also be Turing-decidable which is a contradiction

b. False. There are languages that a TM can handle but CFG cannot.

c. False. Turing Semidecidable Languages can halt when accepting or rejecting w.

d. True. Any algorithm can be represented as a string of zeros and ones of finite length, which is countable.

e. False. By definition, every problem in P also belongs to NP. So, there cannot be a problem that is in NP but not in P.