Final Exam

COSE215: Theory of Computation

2023 Spring

Instructor: Jihyeok Park

June 14, 2023. 14:00-15:15

- **If you are not good at English, please write your answers in Korean.**
  (영어가 익숙하지 않은 경우, 답안을 한글로 작성해 주세요.)
- **Write answers in good handwriting.**
  If we cannot recognize your answers, you will not get any points.
  (글씨를 알아보기 힘들면 점수를 드릴 수 없습니다. 답안을 읽기 좋게 작성해주세요.)
- **Write your answers in the boxes provided.**
  (답안을 제공된 박스 안에 작성해 주세요.)

Student ID	
Student Name	

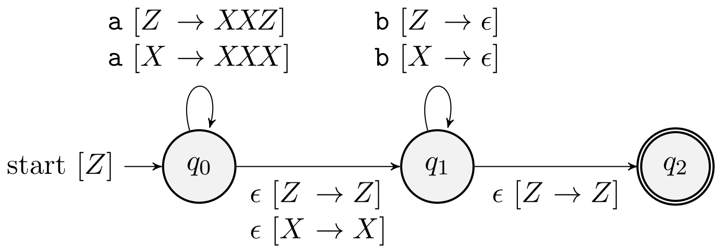
Question:	1	2	3	4	5	6	7	8	Total
Points:	15	20	7	8	10	15	10	15	100
Score:									

1. A **pushdown automaton (PDA)** $P = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$ is a 7-tuple where:

   - Q is a finite set of **states**
   - Σ is a finite set of **symbols**
   - Γ is a finite set of **stack alphabets**
   - $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma^*)$ is a **transition function**
   - $q_0 \in Q$ is the **initial state**
   - $Z \in \Gamma$ is the **initial stack alphabet** (the stack is initially Z)
   - $F \subseteq Q$ is a set of **final states**

   Consider the following PDA P_1 described by a **transition diagram**:

   P_1 =



   A **configuration** (q, w, α) represents the current status of a PDA, and a **one-step move** ($\vdash$) of a PDA P is a transition from a configuration to another one.

   (a) 3 points Fill in the blanks:

   The language accepted by **final states** of a PDA P is generally defined as:

   $$L_F(P) = \{w \in \Sigma^* \mid (q_0, w, Z) \vdash^* (q, \epsilon, \alpha) \text{ for some } q \in F, \alpha \in \Gamma^*\}$$

   Therefore, the language accepted by **final states** of PDA P_1 is:

   $L_F(P_1) = \{$ ______ $\}$

   (b) 6 points Fill in the blanks:

   The language accepted by **empty stacks** of PDA P is generally defined as:

   $L_E(P) = \{w \in \Sigma^* \mid$ ______ $\}$

   Therefore, the language accepted by **empty stacks** of PDA P_1 is:

   $L_E(P_1) = \{$ ______ $\}$

   (c) 6 points Fill in the blanks with the execution for $w =$ `abb` in P_1 that explains why the word w is accepted by final states of P_1 (i.e., $w \in L_F(P_1)$):

______	$\vdash$	______	$\vdash$	______	$\vdash$
______	$\vdash$	______	$\vdash$	______	

2. Design each **PDA** P_i using a **transition diagram** whose language $L_E(P_i)$ accepted by **empty stacks** of the PDA P_i is equal to each of the following languages.

   (a) [5 points] $L_E(P_2) = \{\mathtt{a}^n\mathtt{b}\mathtt{a}^n \mid n \geq 0\}$

$P_2 =$

   (b) [5 points] $L_E(P_3) = L(G)$ where G is the following **context-free grammar (CFG)**:

$$S \rightarrow \mathtt{c} \mid \mathtt{a}S\mathtt{a} \mid \mathtt{b}S\mathtt{b}$$

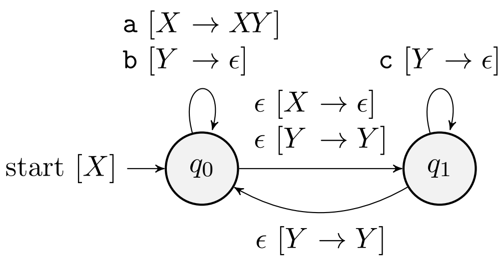
$P_3 =$

(c) 10 points $L_E(P_4) = \{a_1 a_2 \cdots a_{2n} \in \{\mathtt{a}, \mathtt{b}\}^* \mid n \geq 1 \wedge a_i = a_{n+i} \text{ for some } 1 \leq i \leq n\}$

$P_4 =$ [ ]

3. 7 points Consider the following PDA P_5:

$P_5 =$



Fill in the blanks in the following **context-free grammar (CFG)** converted from PDA P_5 whose language is equivalent to the language of **empty stacks** of the PDA P_5.

$S \to$ [ ] | [ ]

$A^X_{0,0} \to$ [ ] | [ ]

$A^X_{0,1} \to$ [ ] | [ ] | [ ]

$A^Y_{0,0} \to A^Y_{1,0} \mid \mathtt{b}$ $\qquad$ $A^Y_{0,1} \to A^Y_{1,1}$

$A^Y_{1,0} \to A^Y_{0,0}$ $\qquad$ $A^Y_{1,1} \to A^Y_{0,1} \mid \mathtt{c}$

Note that each variable $A^X_{i,j}$ should generate all words that cause the PDA to go from the state q_i to the state q_j by popping the alphabet X:

$$A^X_{i,j} \Rightarrow^* w \qquad \text{if and only if} \qquad (q_i, w, X) \vdash^* (q_j, \epsilon, \epsilon)$$

4. 8 points Draw a Venn diagram to illustrate the relationships among the following classes of languages:

- **CFL**: the class of **context-free languages**.
- **DCFL**: the class of **deterministic context-free languages**.
- **DCFL$_{\mathbf{ES}}$**: the class of **deterministic context-free languages by empty stacks**.
- **RL**: the class of **regular languages**.

and place the following languages in the Venn diagram:

① $\{wcw^R \mid w \in \{a, b\}^*\}$

② $\{a^n \mid n \geq 0\}$

③ $\{a^n b^n \mid n \geq 0\}$

④ $\{a^i b^j c^k \mid i, j, k \geq 0 \wedge (i = j \vee j = k)\}$

⑤ $\{ww^R \mid w \in \{a, b\}^*\}$

⑥ $\{0^n 1 \mid n \geq 0\}$

⑦ $\{1(01)^n 0 \mid n \geq 0\}$

⑧ $\{ww \mid w \in \{a, b\}^*\}$

5. Consider the following **context-free grammar (CFG)** G_0:

$$G_0 = \begin{cases} S \to AB \mid BA \mid C \\ A \to \mathtt{a}S \\ B \to \mathtt{b}S \\ C \to \epsilon \mid \mathtt{c} \mid D \\ D \to \mathtt{d}D \end{cases}$$

(a) [3 points] Construct a new CFG G_1 consisting of productions produced by replacing **nullable variables** with ϵ in all combinations, except for the ϵ-production, in G_0:

$G_1 =$

(b) [3 points] Construct a new CFG G_2 by removing all **unit productions** and adding all possible **non-unit productions** of B to A for each **unit pair** (A, B) in G_1:

$G_2 =$

(c) [3 points] Construct a new CFG G_3 by removing all productions that contain **non-generating variables** or come from **unreachable variables** in G_2:

$G_3 =$

(d) [1 point] What is the difference between the languages $L(G_0)$ and $L(G_3)$?

$L(G_0) \setminus L(G_3) =$

6. Fill in the blanks in the **proof** showing that each language L_i is **not** a **context-free language (CFL)** in two different ways.

(a) [10 points] $L_1 = \{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^i\mathtt{d}^j \mid i, j \geq 0\}$.

We can prove L_1 is **not** a **CFL** by using the **pumping lemma** for **CFLs**.

1. Assume that any positive integer n is given. (i.e., $n \geq 1$)
2. Pick a word $L_1 \ni z =$ [ ] .
3. $|z| =$ [ ] $\geq n$.
4. Assume that any split $z = uvwxy$ satisfying ① $|vx| > 0$ and ② $|vwx| \leq n$ is given.
5. Let $i =$ [ ] . We need to show that ¬③ $uv^iwx^iy \notin L_1$:

[ ]

(b) [5 points] $L_2 = \{w \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}^* \mid N_\mathtt{a}(w) = N_\mathtt{c}(w) \land N_\mathtt{b}(w) = N_\mathtt{d}(w)\}$ where $N_x(w)$ denotes the number of x's in w for each $x \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$.

We can prove L_2 is **not** a **CFL** by using the **closure properties** of **CFLs** and L_1.

1. Let's prove by contradiction.
2. Assume that L_2 is a **CFL**.
3. Consider the following language L_3:

$$L_3 = \{ \quad \boxed{} \quad \}$$

such that L_2 [ ] $L_3 = L_1$.

4. Since **CFLs** are closed under [ ] ,

L_1 must be [ ] .

5. However, we know that [ ] .
6. Therefore, it is a contradiction, and L_2 is **not** a **CFL**. □

7. Fill in the blanks about **Turing machines (TMs)**.

(a) 5 points Draw a TM M_1 that accepts the following language:

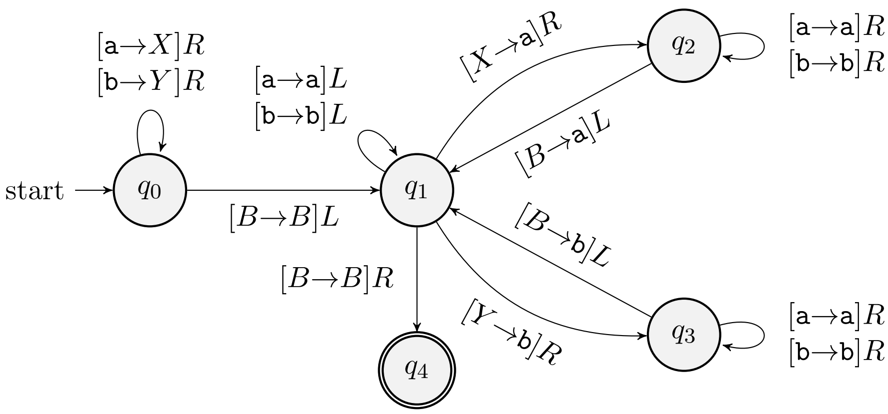
$$L(M_1) = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \geq 0\}$$

$M_1 =$

(b) 5 points The TM M_2 represents a **computable function** $f : \{\mathtt{a}, \mathtt{b}\}^* \to \{\mathtt{a}, \mathtt{b}\}^*$ s.t.:

$f(w) =$

$M_2 =$



8. 15 points **True/False questions.** Answer O for True and X for False.
(Note that each question is worth **1 points**, but you will get **-1 points** for each wrong answer. So, if you are unsure about the answer, leave it blank.)

1. A language is decidable if and only if there is a TM that accepts it. ☐
2. There is no polynomial-time algorithm for all NP problems. ☐
3. If a DCFL L has a prefix property, then L is DCFL_{ES}. ☐
4. All DCFLs always have unambiguous grammars. ☐
5. A language accepted by an unambiguous grammar is always DCFL. ☐
6. The complement of a CFL is always non-CFL. ☐
7. CFLs are closed under difference with regular languages. ☐
8. There is no abstract machine having more expressive power than a TM. ☐
9. NP-hard problems are always NP-complete. ☐
10. If there exists an NP-complete problem that is in P, then P = NP. ☐
11. A Boolean formula $x_1 \land (\neg x_1 \lor x_2) \land \neg x_2$ is satisfiable. ☐
12. The Boolean satisfiability problem is NP-hard. ☐
13. If there is a polynomial-time TM as a verifier for a search problem π, π is in NP. ☐
14. TMs cannot solve NP-complete problems. ☐
15. If there is a polynomial-time reduction from π_1 to π_2, then π_1 is harder than π_2. ☐

This is the last page.
I hope that your tests went well!