CS 301: Languages and Automata **Spring 2018**

Homework 3

Due: Sunday, March 11, 2018

Instructions

This assignment is due Sunday, March 11, 2018 at 11:59PM (Central Time). Solutions for Part I must be submitted on Blackboard and solutions for Part II must be submitted on Gradescope.

Late submissions will be accepted within 24 hours after the deadline with a penalty of 25% of the assignment grade. No late submissions will be accepted more than 24 hours after the deadline.

Part I: JFLAP constructions

For both of the problems below, construct the required context free grammars (CFG) and pushdown automata (PDA) in JFLAP version 7 for the given languages. In addition, for each language, create two text files, one that contains 5 strings (one string per line) that are in the language and one that contains 5 strings (one string per line) that are *not* in the language. **Turn in all 15 files on Blackboard.**

In order to test if strings are in the language or not, first convert the CFG to Chomsky normal form (CNF) by selecting "Transform Grammar" from the "Convert" menu and selecting "Do all" and then "Proceed" until the CFG is in CNF and then click "Export." You can test the resulting CFG by selecting "Multiple CYK Parse" from the "Input" menu. **If you do not first convert to CNF, the CYK algorithm will fail and JFLAP will give you incorrect results.** The grammars you turn in should *not* be the ones you converted to CNF.

Problem 1 Create a CFG that generates each of the languages below.

- **a.** [**10 points**] $A = \{w \mid w \in \{\mathtt{a}, \mathtt{b}\}^* \text{ has more } \mathtt{a}\text{s than } \mathtt{b}\text{s}\}$.
- **b.** [**10 points**] $B = \{w\#x \mid w, x \in \{\mathtt{a}, \mathtt{b}\}^* \text{ and } w^{\mathcal{R}} \text{ is a substring of } x\}$. Remember, $w^{\mathcal{R}}$ is a substring of x if there are strings $y, z \in \{\mathtt{a}, \mathtt{b}\}^*$ such that $x = yw^{\mathcal{R}}z$.
- **c.** [**10 points**] $C = \{\mathtt{a}^m\mathtt{b}^n\mathtt{c}^k \mid m, n > 0 \text{ and } k = m + n\}$.

Problem 2 Create a PDA that recognizes each of the languages below.

a. **[10 points]** $D = \{w \mid \text{every prefix of } w \in \{\texttt{a}, \texttt{b}\}^* \text{ has at least as many } \texttt{a}\text{s as } \texttt{b}\text{s}\}$. [*Hint: As the PDA reads its input, think about the number of* as *minus the number of* bs*. What does it mean if this number is positive, zero, or negative?*]

b. **[10 points]** $E = \{x\texttt{c}^n \mid n \geqslant 0 \text{ and } x \in \{\texttt{a}, \texttt{b}\}^* \text{ has } n \text{ } \texttt{a}\text{s or } n \text{ } \texttt{b}\text{s}\}$.

Part II: Proofs

Remember, your solutions to Part II must be typeset. Handwritten solutions will not be graded and will receive a 0.

Problem 1 **[5 points]** Convert the following CFG to CNF. You may use either the procedure given in class (START, BIN, DEL-ε, UNIT, TERM) or the procedure given in Sipser (START, DEL-ε, UNIT, BIN, TERM). Show each step of the conversion. When deleting rules in the BIN, DEL-ε, UNIT, and BIN steps, state which rules are being deleted. For example, "Remove $T \to \varepsilon$."

$$
\begin{aligned}
T &\to \texttt{aa}T\texttt{b} \mid U \mid \varepsilon \\
U &\to \texttt{bb}U\texttt{a} \mid T \mid \varepsilon
\end{aligned}
$$

Problem 2 **[15 points]** Show that if G is a CFG in Chomsky normal form (CNF), then for any string $w \in L(G)$ of length $n \geqslant 1$, exactly $2n - 1$ steps are required for any derivation of w.

For example, consider the following CFG.

$$
\begin{aligned}
S &\to XU \mid XY \mid \varepsilon \\
T &\to XU \mid XY \\
U &\to TY \\
X &\to \texttt{a} \\
Y &\to \texttt{b}
\end{aligned}
$$

Deriving the string $w = \texttt{aabb}$ takes 7 steps. Here's one derivation.

$$S \Rightarrow XU \Rightarrow \texttt{a}U \Rightarrow \texttt{a}TY \Rightarrow \texttt{a}XYY \Rightarrow \texttt{aa}YY \Rightarrow \texttt{aab}Y \Rightarrow \texttt{aabb}$$

This is just an example to illustrate what you have to prove. Your proof should be about an arbitrary CFG $G = (V, \Sigma, R, S)$ in CNF, *not* about this particular example.

Problem 3 **[15 points]** If A and B are languages, define

$$A \diamond B = \{xy \mid x \in A,\ y \in B, \text{ and } |x| = |y|\}.[1]$$

[1] In LaTeX, you can produce $\diamond$ by using `\diamond` in math mode.

Prove that if A and B are regular languages, then $A \diamond B$ is context-free. [*Hint: Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ be DFAs that recognize A and B, respectively. Construct a PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ to recognize $A \diamond B$. You may assume that $Q_1 \cap Q_2 = \varnothing$. Alternatively, you can consider a right regular grammar for A and a left regular grammar for B and construct a new CFG for $A \diamond B$.*]

Problem 4 [15 points] Given a language A over the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}^*$, define

$$\textsc{Sort}(A) = \{\mathtt{a}^m\mathtt{b}^n \mid \text{there exists some } w \in A \text{ with exactly } m \text{ } \mathtt{a}\text{s and exactly } n \text{ } \mathtt{b}\text{s}\}.$$

For example, $\textsc{Sort}\big(\{\varepsilon, \mathtt{a}, \mathtt{bab}, \mathtt{bba}, \mathtt{abba}\}\big) = \{\varepsilon, \mathtt{a}, \mathtt{abb}, \mathtt{aabb}\}$. We can apply $\textsc{Sort}$ to infinite languages as well. For example, $\textsc{Sort}\big(\underline{(\mathtt{ab})^*}\big) = \{\mathtt{a}^n\mathtt{b}^n \mid n \geqslant 0\}$.

Prove that if A is a regular language, then $\textsc{Sort}(A)$ is a context-free language. [*Hint: Let $M = (Q, Sigma, delta, q_0, F)$ be a DFA that recognizes A and construct a CFG $G = (V, Sigma, R, S)$ that generates $\textsc{Sort}(A)$. Review the DFA/NFA to CFG construction presented in class and modify it to produce a CFG that generates $\textsc{Sort}(A)$ rather than A.*]