Student ID: 805167986

Collaborators: David Tran, Sashwath Sundher, Dylan Gunn

CS181 Winter 2021 – Problem Set 3

Due Tuesday, February 9, 11:59 pm

- Please write your student ID **and the names of anyone you collaborated with** in the spaces provided and attach this sheet to the front of your solutions. **Please do not include your name anywhere since the homework will be blind graded.**
- An extra credit of **5%** will be granted to solutions written using LaTeX. Here is one place where you can create LaTeX documents for free: `https://www.overleaf.com/`. The link also has tutorials to get you started. There are several other editors you can use. We have also posted a short LaTeX tutorial on CCLE under References.
- If you are writing solutions by hand, please write your answers in a neat and readable hand-writing.
- Always explain your answers. When a proof is requested, you should provide a rigorous proof.
- If you don't know the answer, write "I don't know" along with a clear explanation of what you tried. For example: "I couldn't figure this out. I think the following is a start, that is correct, but I couldn't figure out what to do next. [[Write down a start to the answer that you are sure makes sense.]] Also, I had the following vague idea, but I couldn't figure out how to make it work. [[Write down vague ideas.]]" At least 20% will be given for such an answer. Note that if you write things that do not make any sense, no points will be given.
- The homework is expected to take anywhere between 8 to 14 hours. You are advised to start early.
- Submit your homework online on Gradescope. The Gradescope code is 5V7GW5.
- Homework points will be scaled according to the number of homework assignments. All assignments will be weighted equally. As per the syllabus, your homework assignments will together comprise 25% of your final grade.

Note: *Suggested practice problems from the book: 2.4 and 2.5. Please, do not turn in solutions to problems from the book.*

1. **(20 points).** Consider a binary operation ∇ defined as follows: if A and B are two languages, then $A \nabla B = \{xy \mid x \in A,\ y \in B, \text{ and } |x| = |y|\}$. Prove that if A and B are regular languages, then $A \nabla B$ is a context-free language.

Claim:

If A and B are regular languages then $A \nabla B$ if a context-free language.

Proof:

Let A and B be regular languages where $M_A = (Q_A, q_{0_A}, \Sigma_A, \delta_A, F_A)$ and $M_B = (Q_B, q_{0_B}, \Sigma_B, \delta_B, F_B)$ are two DFAs which solve languages A and B respectively. Where $F_A = \{f_{A_1}, \dots, f_{A_n}\}$ and $F_B = \{f_{B_1}, \dots, f_{B_x}\}$

To prove that $A \nabla B$ is a context-free language we need to make a PDA $P = (Q, q_0, \Sigma, \Gamma, \delta, F)$ which accepts the language of $A \nabla B$.

See Figure 1.

$$Q = Q_A \cup Q_B \cup \{q_{new}, q_f\}$$

$$q_0 = q_{new}$$

$$\Sigma = \Sigma_A \cup \Sigma_B$$

$$\Gamma = \{\#\} \cup \{\$\}$$

$$\delta_1(a, c, \varepsilon) = \{(\delta_A(a, c), \#)\} \text{ where } a \in A, c \in \Sigma_A$$

$$\delta_2(b, d, \#) = \{(\delta_B(b, d), \varepsilon)\} \text{ where } b \in B, d \in \Sigma_B$$

$$\delta(q \in Q, \sigma \in \Sigma, g \in \Gamma) = \delta_1 \cup \delta_2$$

$$F = \{q_f\}$$

In our PDA P we will have states Q which consist of the two new states q_{new}, q_f and the original states in both DFAs Q_A, Q_B.

We will have a new starting state q_{new} which has transitions out of it as seen in Figure 1.

Our new alphabet Σ will be the union of the original alphabets Σ_A, Σ_B.

Our stack alphabet will consist of just the bottom of the stack character \$ and an arbitrary symbol # which we will use to "remember" how many states are in x, $|x|$.

Our transition function δ will be defined as the transitions defined in *Figure 1* as well as the union of the transitions δ_1, δ_2. δ_1 is just the transition function δ_A but every move pushes the symbol # onto the stack and the δ_2 is just the transition function δ_B but every move pops the # off until we reach \$ then if we are in an accepting state of M_B we can declare that the string we processing is in $A \nabla B$. This method makes sure that $|x| = |y|$ if $x \in A$ and $y \in B$.
$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \rightarrow \mathbb{P}(Q \times (\Gamma \cup \{\varepsilon\}))$

The accepting state is if we are in an accepting state in M_B and the top of our stack is \$ then we can accept the string because this means that $|x| = |y|$.

If we had a string $s \in A \nabla B$ when we process it in P we will have an $x \in A$ and $y \in B$ where $s = xy$ and $|x| = |y|$. This means the x will be processed in M_A where each step pushes a # onto the stack. Then when we reach an accepting state in M_A an epsilon transition will be taken from

M_A to M_B and each step processing y in M_B will pop of a # from the stack. Then we will reach an accepting state in M_B, and there will be a \$ at the top of the stack so we can epsilon transition to the accepting state of P.

If $s \notin A\nabla B$ we will either

(a) not reach an accepting state in M_A.

(b) reach an accepting state in M_A but will not reach an accepting state in M_B.

(c) reach an accepting state in M_A and M_B but $|x| \neq |y|$ and we will know this because we will be at an accepting state in M_B but the top of the stack will have a # not a \$.

Therefore s will be rejected by P.

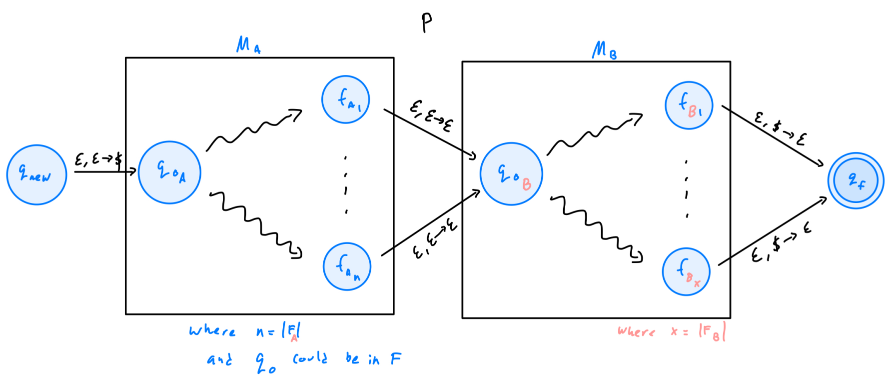

Figure 1: PDA P made from M_A and M_B

Because there exists a PDA P which accepts $A\nabla B$ we know that a PDA can be converted into a Context-Free Grammar, which we proved in class, and that means that there is CFG which represents $A\nabla B$. Therefore we can say that $A\nabla B$ is a context-free language by proof of construction.

$\square$

2. **(45 points).** This problem explores two related languages. Remember to use the ideas from part (a) in part (b).

   (a) **(20 points).** Show that the language

$$L_1 = \{x\$y \mid x, y \in \{0,1\}^* \text{ and } x \neq y\}$$

   over the alphabet $\Sigma = \{\$, 0, 1\}$ is a context-free language.

Claim:

L_1 is a context-free language.

Proof:

Let a CFG $G = (V, \Sigma, S \in V, R \subseteq V \times (V \cup \Sigma \cup \{\varepsilon\})^*)$.

$$V = \{S, U, V, W, X, Y, Z\}$$

$$\Sigma = \{0, 1, \$\}$$

$$S \to U|V$$

$$U \to Y1W|Z0W$$

$$V \to XVX|\$XW|XW\$$$

$$W \to XW|\varepsilon$$

$$X \to 1|0$$

$$Y \to XYX|0W\$$$

$$Z \to XZX|1W\$$$

We have our non terminal variables V, alphabet Σ, starting variable S, and rules R.
We start by going from S to U or V.
Then we will have W which will allow us to make any string x where $x \in \{0,1\}^*$.

If we have a string $s \in L_1$ where $s = x\$y$ and $|x| = |y|$. There must be an index i s.t. $x_i \neq y_i$. We will catch this in our terminal variable U. Those rules will make a string where $s = x\$y$ where $x = j0k$ and $y = l1h$

or

$x = j1k$ and $y = l0h$

but for either case $|j| = |l|$ which means we will be able to find the difference between x and y at index $|j| + 1$.

If we have a different string $s \in L_1$ where $s = x\$y$ and $|x| \neq |y|$. This means our CFG must accept if and only if $x \neq y$. Our rule V does this because either $|x| > |y|$ or $|y| > |x|$.

Because we were able to construct a CFG G which accepts the language L_1 we can say that L_1 is infact a context-free language by proof by construction.

$\square$

(b) **(25 points).** Show that the language

$$L_2 = \{xy \mid x, y \in \{0,1\}^*,\ |x| = |y|, \text{ and } x \neq y\}$$

is a context-free language.

Hint: Have non-determinism on your mind.

Claim:

The language L_2 is a context-free language.

Proof:

Consider a CFG $G = (V, \Sigma, S \in V, R \subseteq V \times (V \cup \Sigma \cup \{\varepsilon\})^*)$

$$V = \{S, X, Y, Z\}$$
$$\Sigma = \{1, 0\}$$
$$S \to XY | YX$$
$$Z \to 1 | 0$$
$$X \to ZXZ | 1$$
$$Y \to ZYZ | 0$$

Our CFG G will be able to make sure that the first half x of a string s and the second half y have the same magnitude $|x| = |y|$ and have at least one difference s.t. $x \neq y$.

We have two cases.

Let $q, w, e, r \in \Sigma^*$

(a) $s = q0we1r \in L_2$

(b) $s = q1we0r \in L_2$

From our rules R we can see that $|q| = |w|$ and $|e| = |r|$.

The length of our string s is $2 \times |q| + 2 \times |e| + 2$, and we can split this string into an x and y s.t. the first half x and the second half y have at least one difference at some index. This means we can say that $x \neq y$ where $|x| = |y|$.

Because we can create a CFG G which solves L_2 we can say that L_2 is a context-free language by proof of construction.

$\square$
