Student ID: Terry Ye

Collaborators: 004757414

CS181 Winter 2019 – Problem Set 3

Due Monday, February 11, 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 LaTeXdocuments for free: `https://www.overleaf.com/`. The link also has tutorials to get you started. There are several other editors you can use.
- 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.

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.

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.

   (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.

1 Problem1

If A and B are regular, then let M_A be an NFA with $(Q_A, \Sigma_A, \delta_A, q_{0A}, F_)A$ which recognizes A. Let M_B be an NFA with $(Q_B, \Sigma_B, \delta_B, q_{0B}, F_B)$. We can construct a PDA P with:

Q = $Q_A \cup Q_B \cup q_0 \cup q_F$
$\Sigma = \Sigma_A \cup \Sigma_B$
$\Gamma = \{\$, 0\}$
$q_0 = q_0$
$F = \{q_F\}$
Transition function:
$\delta(q_0, \epsilon, \epsilon) = \{(q_{0A}, \$)\}$ (from start state ,push \$ to stack)
$\delta(q, \epsilon, \epsilon | q \in F_A) = \{(q_{0B}, \epsilon)\}$ (any accepting state in A will trasit to start state in B without modifying stack)
$\delta(q, \epsilon, \$ | q \in F_B) = \{(q_F, \epsilon)\}$
(reach accepting state if it reaches accepting states in B and stack is empty)
$\delta(q, a, \epsilon | q \in Q_A) = \{(\delta_A(q, a), 0)\}$ (if q in Q_A, add 0 to stack)
$\delta(q, a, 0 | q \in Q_B) = \{(\delta_B(q, a), \epsilon)\}$ (if q in Q_B, pop 0 from stack)

$A \nabla B \subseteq L(P)$
Proof: If x $\in A\nabla B$, for the first half of x x_1, $M(x_1)$ will reach the state $(q_{0B}, 0)$ with $|x_1|$ 0 in the stack as $x_1 \in A$. And then for the second half x_2, it will reach $(q, |q \in F_B)$ as $x_2 \in B$ and $|x_2| = |x_1|$, so stack will pop all $|x_1|$ 0 out and reach empty state. So M(x) will end in (q_F, ϵ) and be accepted.
$x \in L(P) \rightarrow A\nabla B \subseteq L(P)$.

$L(P) \subseteq A\nabla B$
Proof: If x $\in L(P)$, then x can be divided into two substrings that first part $x_1 \in A$, second part $x_2 \in B$. And $|x_1| = |x_2|$ as the stack has to be empty at last, so $x \in A\nabla B \rightarrow L(P) \subseteq A\nabla B$.

2 Problem2

(a) For L_1, there is a simple PDA P
$Q = \{q_0, q_1, q_{20}, q_{21}, q_3, q_{F1}, q_{F2}\}$
$\Sigma = \{0, 1, \$, \#\}$
$\Gamma = \{\$, 0, 1, x\}$
$q_0 = q_0$
$F = \{q_{F1}, q_{F2}\}$
Transition function:
$\delta(q_0, \epsilon, \epsilon) = \{(q_1, \$)\}$ (start by pushing \$ to stack)
$\delta(q_1, 0, \epsilon) = \{(q_{20}, a)\}$
$\delta(q_1, 1, \epsilon) = \{(q_{21}, a)\}$ (go into corresponding state for the current input in x)
$\delta(q_1, a, \epsilon) = \{(q_1, x)\}$ (push x to stack for every input in x)
$\delta(q_{20}, \$, \epsilon) = \{(q_{30}, \epsilon)\}$
$\delta(q_{21}, \$, \epsilon) = \{(q_{31}, \epsilon)\}$ (if encounter \$,go to corres state in q_3
$\delta(q_{30}, a, x) = \{(q_{30}, \epsilon)\}$
$\delta(q_{31}, a, x) = \{(q_{31}, \epsilon)\}$ (pop x for each input symbol in y)
$\delta(q_{31}, 0, \$) = \{(q_{F1}, \epsilon)\}$
$\delta(q_{30}, 1, \$) = \{(q_{F1}, \epsilon)\}$ (if reach the point of x_i different from y_i, accept)

$\delta(q_{F1}, \epsilon, \epsilon) = \{(q_{F1}, \epsilon)\}$
(F_1 is the accept state that self loop so any string enter F1 is automatically accepted)
$\delta(q_{30}, \epsilon, x) = \{(q_{F2}, \epsilon)\}$
$\delta(q_{31}, \epsilon, x) = \{(q_{F2}, \epsilon)\}$ (if end of input, and stack not empty, accept because size different)
(F_2 is the accept state that has no outward function so it can only accept if comes in at the end of input)

$L(P) \subseteq L_1$
Proof: If a string s $\in L(P)$, then it either has unequal size of strings before and after \$ or there is an i that x_i and y_i are different for $x\$y$. so $x \neq y$. $s \in L_1$.

$L_1 \subseteq L(P)$ Proof: If a string s $\in L_1$, then there are two case for s: it is either that $|x| \neq |y|$ or $|x| = |y|$ and $x \neq y$. For the first case, it will be accepted by P as there will be a point where the input is empty while stack is not and vice versa. For the second case,there will be an i that $x_i \neq$, so it can be accepted bt P as well.

(b)There is a grammar G with:
V = $\{A, B, S, C\}$
$\Sigma = \{0, 1\}$
S = S
R = $S \rightarrow AB|BA$
$A \rightarrow 0 \mid CAC$
$B \rightarrow 1 \mid CBC$
$C \rightarrow 0 \mid 1$

L(G) $\subseteq L_2$:
Proof: If a string s is generated by G, then it is even length as A and B can only be odd. So it can be divided into two halves x and y with equal length $|x| = |y|$. x and y cannot be equal because there will be an i that $x_i \neq y_i$ as A and B has the terminal case $0 \neq 1$ and it fills the rest. So x $\in L_2$

$L_2 \subseteq L(G)$:
Proof: If a string s is in L_2, then there is x and y with an i such that $x_i \neq y_i$. So this is the case when $A \rightarrow 0$ and $B \rightarrow 1$. And G can generate all the rest characters in x and y because it can generate all possible strings for the rest as C can be either 0 or 1. So s $\in L(G)$.