CpSc 421 Homework 3 **Due: Oct. 6, 4pm**

**Solution Set**

1. **(40 points)** Use the pumping lemma to prove that each language listed below is not regular. For each language, I state $\Sigma$ the input alphabet.

   (a) $\{w \mid$ the number of zeros in $w$ is less than the number of ones$\}$. $\Sigma = \{0, 1\}$.

   **Solution:**

   i. Call this language $A$. Let $p$ be a proposed pumping lemma constant $A$.
   ii. Let $w = 0^p 1^{p+1}$. $w \in A$.
   iii. Let $xyz = w$ with $0 < |y|$ and $|xy| \leq p$. Note that $x$ and $y$ are in $0^*$.
   iv. $xy^2z = 0^{p+|y|}1^{p+1} \notin A$.
   v. $A$ does not satisfy the pumping lemma; therefore it is not regular.

   (b) $\mathtt{1}^{n^2}$. $\Sigma = \{\mathtt{1}\}$.

   **Solution:**

   i. Call this language $B$. Let $p$ be a proposed pumping lemma constant $B$.
   ii. Let $w = 1^{p^2}$. $w \in B$.
   iii. Let $xyz = w$ with $0 < |y|$ and $|xy| \leq p$.
   iv. $xy^2z = 1^{p^2+|y|}$. We observe that

   $$\begin{array}{rcl} p^2 & < & p^2 + |y| \\ & \leq & p^2 + p \\ & < & p^2 + 2p + 1 \\ & = & (p+1)^2 \end{array}$$

   Thus, $|xy^2z|$ is not a perfect square; therefore, $xy^2z \notin B$.
   v. $B$ does not satisfy the pumping lemma; therefore it is not regular.

   (c) $\{w \cdot \mathtt{c} \cdot w^{\mathcal{R}} \mid w \in \{\mathtt{a}, \mathtt{b}\}^*, w^R$ is the reverse of $w\}$. $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$.

   **Solution:**

   i. Call this language $C$. Let $p$ be a proposed pumping lemma constant $C$.
   ii. Let $w = \mathtt{a}^p\mathtt{c}\mathtt{a}^p$. $w \in C$.
   iii. Let $xyz = w$ with $0 < |y|$ and $|xy| \leq p$. Note that $x$ and $y$ are in $\mathtt{a}^*$.
   iv. $xy^2z = \mathtt{a}^{p+|y|}\mathtt{c}\mathtt{a}^p \notin C$.
   v. $C$ does not satisfy the pumping lemma; therefore it is not regular.

   (d) $\{w \mid$ the number of left parentheses in any prefix of $w$ is greater than or equal to the number of right parentheses, and the number of left parentheses in $w$ is equal to the number of right parentheses $\}$. $\Sigma = \{(, )\}$.

   **Solution:**

   i. Call this language $D$. Let $p$ be a proposed pumping lemma constant $D$.
   ii. Let $w = (^p)^p$. $w \in D$.
   iii. Let $xyz = w$ with $0 < |y|$ and $|xy| \leq p$. Note that $x$ and $y$ are in $(^*$.
   iv. $xy^2z = (^{p+|y|})^p \notin D$.
   v. $D$ does not satisfy the pumping lemma; therefore it is not regular.

2. **(20 points)** Let

$$\Sigma_3 = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \cdots \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}.$$

$\Sigma_3$ contains all size 3 columns of $0$s and $1$s. A string of symbols in $\Sigma_3$ gives three rows of $0$s and $1$s. Consider each row to be a binary number with the most significant bit first. For example, let

$$w = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

The first row of $w$ is the binary representation of $7$, the second row corresponds to $5$, and the third row corresponds to $12$.

Let

$$B = \{w \in \Sigma_3^* \mid \text{the bottom row of } w \text{ is the product of the top two rows}\}.$$

Show that $B$ is not regular.

**Solution:**

(a) Let $p$ be a proposed pumping lemma constant for $B$.

(b) Let

$$w = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^p \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^{p+1}$$

$w$ is the string for $2^{p+1} * 2^{p+1} = 2^{2p+2}$.

(c) Let $xyz = w$ with $0 < |y|$ and $|xy| \leq p$.

(d) Consider $xy^2z$. Pumping $y$ changes the value of the third component (the product) but just puts more zeros into the upper bits of the factors and therefore doesn't change their values. Thus, the supposed product changes, but the factors remain the same. Therefore $xy^2z \notin B$.

(e) $B$ does not satisfy the pumping lemma therefore it is not regular.

3. **(20 points)** Consider the two languages described below:

- $\{w \in \{\mathsf{a}, \mathsf{b}\}^* \mid \exists x, y \in \Sigma^*. \ (w = xy) \wedge \#a(x) = \#b(y)\}$

- $\{w \in \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}^* \mid \exists x, y \in \Sigma^*. \ (w = x \cdot \mathsf{c} \cdot y) \wedge \#a(x) = \#b(y)\}$

One of these languages is regular and the other is not. Determine which is which and give short proofs for your conclusions.

**Solution:** Let

$$\begin{array}{rcl} A_1 & = & \{w \in \{\mathsf{a}, \mathsf{b}\}^* \mid \exists x, y \in \Sigma^*. \ (w = xy) \wedge \#a(x) = \#b(y)\}, \\ A_2 & = & \{w \in \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}^* \mid \exists x, y \in \Sigma^*. \ (w = x \cdot \mathsf{c} \cdot y) \wedge \#a(x) = \#b(y)\} \end{array}$$

$A_1$ is regular.

In fact $A_1 = \Sigma^*$. This is a problem where it can really help to try a few examples. Try to find a string that is not in $A_1$. After a few attempts, you'll see what is happening.

Let $\mathit{diff}(x, y) = \#a(x) - \#b(y)$. Given a string $w$, let $x_i y_i = w$ with $|x_i| = i$. Clearly $\#a(x_0) = 0$ (because $x_0 = \epsilon$, and $\#b(y_0) = \#b(w)$ (because $y_0 = w$). Thus $\mathit{diff}(x_0, y_0) = -\#b(w)$. Furthermore, $\mathit{diff}(x_{i+1}, y_{i+1}) = \mathit{diff}(x_i, y_i) + 1$. To see this, let $x_{i+1} = x_i \cdot c$ (thus $y_i = c \cdot y_{i+1}$). If $c = \mathsf{a}$, then $\#a(x_{i+1} = \#a(x_i) + 1$, and $\#b(y_{i+1}) = \#b(y_i)$; thus $\mathit{diff}(x_{i+1}, y_{i+1}) = \mathit{diff}(x_i, y_i) + 1$ as claimed. On the other hand, if $c = \mathsf{b}$, then $\#a(x_{i+1} = \#a(x_i)$, and $\#b(y_{i+1}) = \#b(y_i) - 1$; and again $\mathit{diff}(x_{i+1}, y_{i+1}) = \mathit{diff}(x_i, y_i) + 1$. We've now shown the base case and induction step to prove that $\mathit{diff}(x_i, y_i) = i - \#b(w)$.

Let $i = \#b(w)$. We conclude that $\mathit{diff}(x_i, y_i) = 0$, which means that $\#a(x_i) = \#a(y_i)$, and we also have that $x_i y_i = w$. Thus, $w \in A_1$. Our choice of $w$ was arbitarary. Thus, $A_1$ contains all strings; in other words, $A_1 = \Sigma^*$ which is regular. Therefore, $A_1$ is regular.

$A_2$ is not regular.

(a) Let $p$ be a proposed pumping lemma constant $A_2$.

(b) Let $w = \mathsf{a}^p \mathsf{c} \mathsf{b}^p$. $w \in A_2$.

(c) Let $xyz = w$ with $0 < |y|$ and $|xy| \leq p$. Note that $x$ and $y$ are in $\mathsf{a}^*$.

(d) $xy^2z = \mathsf{a}a^{p+|y|}\mathsf{c}\mathsf{b}^p \notin D$.

(e) $D$ does not satisfy the pumping lemma therefore it is not regular.

Note that in this case, the $\mathsf{c}$ in string $w$ (or $xy^iz$ forced the choice of prefix in which the $\mathsf{a}$'s were counted and the suffix in which the $\mathsf{b}$'s were counted. Thus, we couldn't divide the string at any arbitrary location (as we could with langauge $A_1$). This extra restrictiveness of $A_2$ is what allowed us to violate the pumping lemma.

4. **(30 points)** (Sipser problem 1.47)

If $A$ is any language over alphabet $\Sigma$, let $A_{\frac{1}{2}-}$ be the set of all first halves of strings in $A$ so that

$$A_{\frac{1}{2}-} \quad = \quad \{u \mid \exists v \in \Sigma^{|u|}.\ uv \in A\}$$

Show that if $A$ is regular, then so is $A_{\frac{1}{2}-}$.

**Solution:** Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA that recognizes $A$. We will construct an NFA that recognizes $A_{\frac{1}{2}-}$. The basic idea is to use $M$ go read the input string. In parallel, we will have an NFA, $N$, that runs backwards, similar to the NFA that we used to show that $A^{\mathcal{R}}$ is regular in HW2. Rather than processing the input string, $N$ will determine all possible reachable states given all possible inputs strings of the specificied lenght. If after reading a string $w$, $N$ can reach the state that $M$ is in, then we accept.

Here's the formal definition of $N$:

$$\begin{aligned} N^{\mathcal{R}} \quad &= \quad (Q \cup \{q_x\}, \Sigma, \delta^{\mathcal{R}}, q_x, \emptyset) \\ \delta^{\mathcal{R}}(q, c) \quad &= \quad \{p \mid \exists d \in \Sigma.\ \delta(p, d) = q\} \end{aligned}$$

Note that $\delta_N(q, c)$ includes $p$ if $M$ there is any input symbol for which $M$ can move from $p$ to $q$ – it doesn't have to be $c$. This is what allows $N$ to simulate an arbitrary string of the same length as the input.

Now, we build a product machine with $M$ and $N^{\mathcal{R}}$:

$$\begin{aligned} N_{\frac{1}{2}-} \quad &= \quad Q \times (Q \cup \{q_x\}, \Sigma, \delta_{\frac{1}{2}-}, (q_0, q_x), F_{\frac{1}{2}-}) \\ \delta_{\frac{1}{2}-}((q_1, q_2), c) \quad &= \quad \{(p_1, p_2) \mid (p_1 = \delta(q_1, c)) \wedge (p_2 \in \delta^{\mathcal{R}}(q_2, c))\} \\ F_{\frac{1}{2}-} \quad &= \quad \{(q, q) \in Q \times Q\} \end{aligned}$$

The DFA part of $N_{\frac{1}{2}-}$ says that $M$ reaches state $q$ after reading $w$. The NFA part says that there is a $v$ with $|v| = |w$ that would take $M$ from state $q$ to an accepting state. Thus, $N_{\frac{1}{2}-}$ accepts $w$ iff $w \in A_{\frac{1}{2}-}$. This shows that $A_{\frac{1}{2}-}$ is recognized by an NFA; therefore, $A_{\frac{1}{2}-}$ is regular.