# Homework 03

Aaron Wang

February 17 2025

1. **Regular expressions vs. Unix regular expressions.** Regular expressions and Unix regular expressions have some superficial differences, but also some deeper ones that affect the class of languages recognized.

   (a) Unix regular expressions have quantifiers: if $\alpha$ is a regular expression, $\alpha^{\{m,n\}}$ is a regular expression that matches at least $m$ and no more than $n$ strings that match $\alpha$. More formally, it matches all strings $w^{(1)} \cdots w^{(l)}$ where $m \leq l \leq n$, and for all $i$ such that $1 \leq i \leq l$, $w^{(i)}$ matches $\alpha$. Prove that for any regular expression with quantifiers, there is an equivalent regular expression without quantifiers.

   Let $f(l, \alpha) = w^1 w^2 \cdots w^l$ s.t. $w^{(i)} = \alpha$. For any Unix regular expression that has quantifiers, we can convert it to a regular expression without quantifiers with this process.

   $$\alpha^{\{m,n\}} = \bigcup_{m \leq l \leq n} \alpha^{\{l,l\}} = \bigcup_{m \leq l \leq n} f(l, \alpha)$$

   Thus, we have a Unix regular expressions that have quantifiers with an equivalent regular expression without quantifiers.

   (b) Unix regular expressions have backreferences: for an explanation, please see

   `http://www.regular-expressions.info/backref.html`.

   Give an example of a Unix regular expression that uses backreferences to describe a nonregular language, and prove that this language is not regular. We want you to get practice writing a non-regularity proof, so although you may use Examples 1.73–77, do not simply cite one of them; please write out a full proof.

   An example of a Unix Regular Expression that uses backreferences: ([01]*)\1

   Let this language be represented as $L = \{ww | w \in \{0, 1\}^*\}$

   Proof: Assume $L$ is a regular language. Let $p$ be the pumping length from the Pumping Lemma. Let $s = 1^p 0 1^p 0$. Observe that $s \in L$. The Pumping Lemma says that for some $xyz$, $s = xyz$, $|y| > 0$, $|xy| \leq p$ and $xyyz \in L$. Let $s' = xyyz$. Since $|xy| \leq p$, we know that $y$ contains all 1s so $s' = 1^r 0 1^p 0$ s.t. $r > p$ which means that $s' \notin L$ $\lightning$. Thus, by contradiction we have shown that $L$ can not be a regular language.

2. **Binary addition.** This problem is about two ways of representating addition of binary natural numbers. We consider 0 to be a natural number. We allow binary representations of natural numbers to have leading 0s, and we consider $\varepsilon$ to be a binary representation of 0. When adding numbers, we do not allow overflow, so, for example, $1111 + 0001 = 0000$ is false.

   (a) [Problem 1.32] Let

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

   that is, an alphabet of eight symbols, each of which is a column of three bits. Thus, a string over $\Sigma_3$ gives three rows of bits. Show that the following is regular:

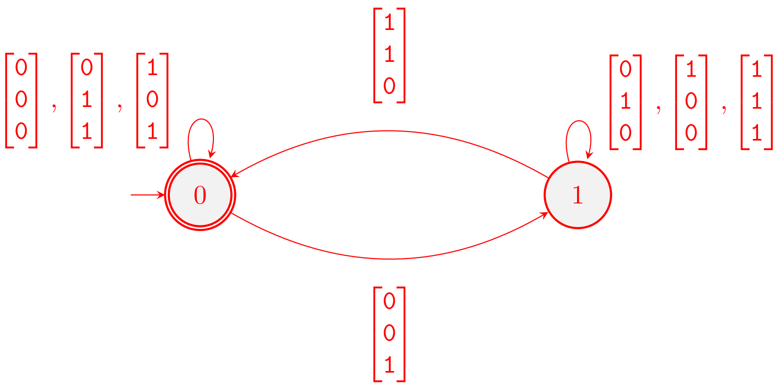
$$B = \{w \in \Sigma_3^* | \text{ the bottom row of w is the sum of the top two rows}\}.$$

   For example, because 011+001 = 100,

$$\begin{bmatrix} \texttt{0} \\ \texttt{0} \\ \texttt{1} \end{bmatrix} \begin{bmatrix} \texttt{1} \\ \texttt{0} \\ \texttt{0} \end{bmatrix} \begin{bmatrix} \texttt{1} \\ \texttt{1} \\ \texttt{0} \end{bmatrix} \in B$$

   Hint: Since it's easier to think about addition from right to left, design an automaton for $B^R$ first, then convert it into an automaton for $B$.

   The following is an NFA for $B$. The intuition is this. State 0 does not expect a carry in and state 1 expects a carry in. Thus, for state 0 if the next column would give a carry out, it automatically rejects. Likewise, for state 1, if the next column does not give a carry out, it automatically rejects. Further, if the addition of the top two matches the bottom, a carry out will not be expected for the next turn so it will transition to state 0; otherwise, it will transition to state 1.




   (b) [Problem 1.53] Let $\Sigma = \{\texttt{0}, \texttt{1}, +, =\}$, and prove that the following is not regular:

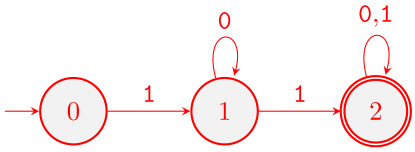
$$ADD = \{x = y + z | x, y, z \in \{\texttt{0}, \texttt{1}\}^* \text{ and } x = y + z \text{ is true }\}.$$

   Proof: Assume $ADD$ is a regular language. Let $p$ be the pumping length from the Pumping Lemma. Let $s = \texttt{1}^p\texttt{=1}^p\texttt{+0}$. Observe that $s \in ADD$. The Pumping Lemma says that for some $xyz$, $s = xyz$, $|y| > 0$, $|xy| \leq p$ and $xyyz \in ADD$. Let $s' = xyyz$. Since $|xy| \leq p$, we know that $y$ contains all 1s so $s' = \texttt{1}^r\texttt{=1}^p\texttt{+0}$ s.t. $r > p$. $s'$ is not true as a number + 0 should be itself but it is not in this expression. Since $s'$ is not a true expression, $s' \notin ADD$ $\lightning$. Thus, by contradiction we have shown that $ADD$ can not be a regular language.

3. **Similar but different** [Problem 1.49].

(a) Let $B = \{1^k w | w \in \{0,1\}^*$ and $w$ contains at least $k$ 1s, for $k \geq 1\}$. Show that $B$ is a regular language. Hint: Try out some strings to see what does and doesn't belong to $B$, in order to find another simpler way of thinking about $B$.

The following is an NFA for B. The Intuition is this. First let's consider $k = 1$. In this case, all we need is a $w$ that contains one 1. Now consider, what if we thought $k$ was 2 or $s = 1^2 w$. Let us rewrite it as $s = 1w'$ s.t. $w' = 1w$. Now we have the same case as before. For any $k > 1$, we could do this such that $s = 1^k w = 1^1 w'$ s.t. $w' = 1^{k-1} w$. In essence, we now realize that we are looking only for a string that starts with 1 and contains at least 2 1s.




(b) Let $C = \{1^k w | w \in \{0,1\}^*$ and $w$ contains at most $k$ 1s, for $k \geq 1\}$. Prove that $C$ is not a regular language.

Proof: Assume $C$ is a regular language. Let $p$ be the pumping length from the Pumping Lemma. Let $s = 1^p 0 1^p$. Observe that $s \in C$. The Pumping Lemma says that for some $xyz$, $s = xyz$, $|y| > 0$, $|xy| \leq p$ and $xz \in C$. Let $s' = xz$. Since $|xy| \leq p$, we know that $y$ contains all 1s so $s' = 1^r 0 1^p$ s.t. $r < p$. Thus $k \leq r$ and $w$ contains more than $p - 1 = r$ 1s so $s' \notin L$ $\lightning$. Thus, by contradiction we have shown that $C$ can not be a regular language.