Homework 3: Regular expressions and non-regular languages

Theory of Computing (CSE 30151), Spring 2025

Due: **Monday** 2024-02-17 5pm

Instructions

- Create a PDF file (or files) containing your solutions. You can write your solutions by hand, but please scan them into a PDF.
- Please name your PDF file(s) as follows to ensure that the graders give you credit for all of your work:
  - If you're making a complete submission, name it `netid-hw3.pdf`, where *netid* is replaced with your NetID.
  - If you're submitting some problems now and want to submit other problems later, name it `netid-hw3-part123.pdf`, where `123` is replaced with the problem number(s) you are submitting at this time.
- Submit your PDF file(s) in Canvas.

Problems (10 points each)

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 α is a regular expression, $\alpha^{\{m,n\}}$ is a regular expression that matches at least m and no more than n strings that match α. 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 α. Prove that for any regular expression with quantifiers, there is 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.

Solution Grader: Renee Shi

(a) (5 points) Define

$$\alpha^i = \underbrace{\alpha \circ \cdots \circ \alpha}_{i \text{ copies}}.$$

Note that quantifiers can be used on any subexpression of a regular expression. We didn't take off points if a solution didn't take this into account, but it should.

Short answer: Given a regular expression containing quantifiers, we can go through it and change every subexpression of the form $\alpha^{\{m,n\}}$ to

$$\bigcup_{i=m}^{n} \alpha^i = \alpha^m \cup \alpha^{m+1} \cup \cdots \alpha^{n-1} \cup \alpha^n.$$

Long answer: We define a function $\mathsf{dequantify}(\alpha)$ that inputs a regular expression with quantifiers α and returns an equivalent regular expression without quantifiers.

- If $\alpha = a \in \Sigma$ or $\alpha = \emptyset$ or $\alpha = \varepsilon$, let $\mathsf{dequantify}(\alpha) = \alpha$.
- If $\alpha = \beta \cup \gamma$, let $\mathsf{dequantify}(\alpha) = \mathsf{dequantify}(\beta) \cup \mathsf{dequantify}(\gamma)$.
- If $\alpha = \beta \circ \gamma$, let $\mathsf{dequantify}(\alpha) = \mathsf{dequantify}(\beta) \circ \mathsf{dequantify}(\gamma)$.
- If $\alpha = \beta^*$, let $\mathsf{dequantify}(\alpha) = \mathsf{dequantify}(\beta)^*$.
- If $\alpha = \beta^{\{m,n\}}$, let $\mathsf{dequantify}(\alpha) = \bigcup_{i=m}^{n} \mathsf{dequantify}(\beta)^i$.

(b) (5 points: 2 points for the language, 3 points for the proof) `(a+)b\1` will match the language $L = \{a^n b a^n | n > 0\}$. We can prove that this language is not regular using proof by contradiction and the pumping lemma: Suppose, for the sake of contradiction, that L is regular. Let p be the pumping length given by the pumping lemma. Let $s = \mathtt{a}^p\mathtt{b}\mathtt{a}^p$. Then the pumping lemma writes s as xyz, where $|xy| \leq p$ and $|y| > 0$, which means that y consists of only `a`s from the first set of `a`'s. Let $i = 2$. The pumping lemma says that $xy^iz \in B$, but xy^iz contains more `a`s in the first set than in the second set, which is a contradiction.

---

2. **Binary addition.** This problem is about two ways of representing 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 ε 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

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

that is, an alphabet of eight symbols, each of which is a column of three bits. Thus, a string over Σ_3 gives three rows of bits. Show that the following is regular by writing an NFA for it:

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

For example, because $011 + 001 = 100$,

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

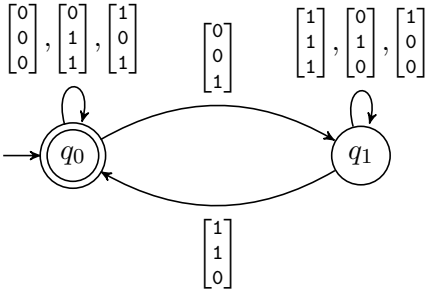
Hint: Since it's easier to think about addition from right to left, design an automaton for B^R first. Then reverse the direction of every transition to get an automaton for B.

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

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

Solution Grader:

(a) (5 points) Some variations are possible; this version allows the empty string but doesn't allow overflow.



Since Σ_3 can be represented by a finite automaton, then it must be a regular language.

(b) (5 points) There are many choices of s that will work here.

Suppose that ADD is a regular language; then there is a DFA M such that $\mathcal{L}(M) = L$. Let p be the pumping length given by the pumping lemma. Then let $s = \mathtt{1}^p\mathtt{=}\mathtt{1}^p\mathtt{+}\mathtt{0}^p$. Since s belongs to ADD, the pumping lemma says that $s = xyz$ for some x, y, z such that $|y| > 0$, $|xy| \leq p$, and $xy^iz \in ADD$ for all $i \geq 0$.

Since $|xy| \leq p$, it must be that y is made only of 1's. Then take $s' = xy^2z$, which must also be accepted by M. However, $s' = \mathtt{1}^{(p+k)}\mathtt{=}\mathtt{1}^p\mathtt{+}\mathtt{0}^p$, for some $k = |y| > 0$, which is not a true binary addition fact, so s' does not belong to ADD. This contradicts the claim that s' is accepted by M. So ADD is not regular.

---

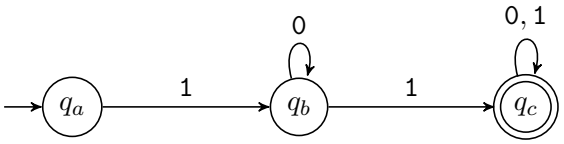
3. **Similar but different** [Problem 1.49].

(a) Let $B = \{1^kw \mid w \in \{0,1\}^* \text{ and } w \text{ contains at least } k \text{ 1s, for } k \geq 1\}$. 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.

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

Solution Grader: Abby Kleist

(a) The case when $k = 1$ is a superset of all other cases. A simpler way of thinking about B is that it describes the language of all strings that start with a 1 and have at least two 1s. This can be described by the regular expression $\mathtt{1}\mathtt{0}^*\mathtt{1}(\mathtt{0} \cup \mathtt{1})^*$, or by the following finite automaton:



(b) Let the pumping length given by the pumping lemma be p. Consider the string $s = \mathtt{1}^p\mathtt{0}\mathtt{1}^p$. The pumping lemma says that s can written as xy^iz with $|xy| \leq p$ and $|y| > 0$. Since $|xy| \leq p$, $y = \mathtt{1}^j$ for some $j > 0$. Then the pumping lemma says that $xy^iz \in C$ for all i, but $xy^0z = \mathtt{1}^{p-j}\mathtt{0}\mathtt{1}^p$ and this has more 1s after the 0 than before, which is a contradiction.
