

CSCE 355, Spring 2016, Assignment 9

Due: December 15, 2019

Instructions

Make sure to submit your homework on **SINGLE-SIDED 8.5x11 INCH PAPER**. Write your **NAME ON EVERY PAGE** and **DO NOT STAPLE OR FOLD** the sheets.

In this and future homeworks, questions marked as optional will not be used for quizzes or final exam questions.

The Exercises are taken out of Hopcroft, with their difficulty annotations preserved.

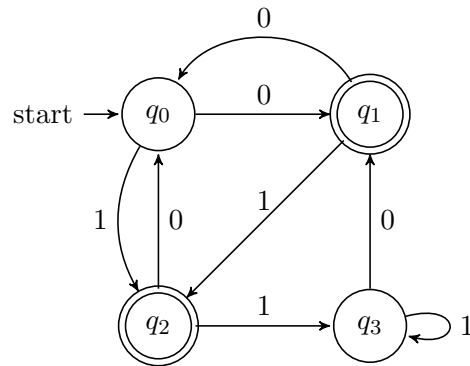
Exercise 4.2.1: Suppose h is the homomorphism from the alphabet $\{0, 1, 2\}$ to the alphabet $\{a, b\}$ defined by $h(0) = a$; $h(1) = ab$, and $h(2) = ba$.

- What is $h(0120)$?
- What is $h(21120)$?
- If L is the language $L(\mathbf{01^*2})$, what is $h(L)$?
- If L is the language $L(\mathbf{0 + 12})$, what is $h(L)$?
- Suppose L is the language $\{ababa\}$, that is, the language consisting of only the one string $ababa$. What is $h^{-1}(L)$?
- If L is the language $L(\mathbf{a(ba)^*})$, what is $h^{-1}(L)$?

(not in text): Let h be the homomorphism mapping strings in $\{a, b, c\}^*$ to strings in $\{0, 1\}^*$ such that

$$\begin{aligned}h(a) &= 011, \\h(b) &= 0, \\h(c) &= 110.\end{aligned}$$

- Find $h^{-1}(L)$, where $L = \{00110, 000111\}$.
- Using the transformation described in class, give a regular expression for $h(M)$, where M is the regular language denoted by $a^*(bc + ca)^*$.
- Using the construction described in class, construct a DFA for $h^{-1}(N)$, where N is recognized by the following DFA:



! Exercise 4.2.6: (optional) Show that the regular languages are closed under the following operations:

- $\min(L) = \{w \mid w \text{ is in } L, \text{ but no proper prefix of } w \text{ is in } L\}$.
- $\max(L) = \{w \mid w \text{ is in } L \text{ and for no string } x \text{ other than } \varepsilon \text{ is } wx \text{ in } L\}$.
- $\text{init}(L) = \{w \mid \text{for some string } x, wx \text{ is in } L\}$.

Hint: Like Exercise 4.2.2, it is easiest to start with a DFA for L and perform a construction to get the desired language.

***!! Exercise 4.2.8 (optional):** Let L be a language. Define $\text{half}(L)$ to be the set of first halves of strings in L , that is, $\{w \mid \text{for some } x \text{ such that } |x| = |w|, \text{ we have } wx \text{ in } L\}$. For example, if $L = \{\varepsilon, 0010, 011, 010110\}$ then $\text{half}(L) = \{\varepsilon, 00, 010\}$. Notice that odd-length strings do not contribute to $\text{half}(L)$. Prove that if L is a regular language, so is $\text{half}(L)$.

!! (not in text; optional): Let L be a language. Define \sqrt{L} to be the set $\{w \mid ww \text{ is in } L\}$. Prove that if L is regular, then so is \sqrt{L} . *Hint:* Given a DFA for L with n many states, one can build a DFA for \sqrt{L} with n^n many states.

More generally, for $k \geq 1$, define $\sqrt[k]{L}$ to be the set $\{w \mid w^k \text{ is in } L\}$. Also define $\sqrt[*]{L}$ to be the set $\{w \mid w \text{ is in } \sqrt[k]{L} \text{ for some } k \geq 1\}$. Similar constructions prove that if L is regular, then all these other languages are regular.

! Exercise 4.2.13: We can use closure properties to help prove certain languages are not regular. Start with the fact that the language

$$L_{0^n 1^n} = \{0^n 1^n \mid n \geq 0\}$$

is not regular. Prove the following languages not to be regular by transforming them, using operations known to preserve regularity, to $L_{0^n 1^n}$:

- a) $\{0^i 1^j \mid i \neq j\}$.
- b) $\{0^n 1^m 2^{n-m} \mid n \geq m \geq 0\}$.

Exercise 4.2.14: In Theorem 4.8, we described the “product construction” that took two DFAs and constructed one DFA whose language is the intersection of the languages of the first two.

- 3. Show how to modify the product construction so the resulting DFA recognizes the difference of the languages of the two given DFAs.
- 4. Show how to modify the product construction so the resulting DFA recognizes the union of the languages of the two given DFAs.

(not in text): Let L be a language. Define $subseq(L)$ to be the set of all strings obtained by removing zero or more symbols (anywhere) from a string in L and closing up the gaps. For example, if $L = \{abc, cab\}$, then

$$subseq(L) = \{\varepsilon, a, b, c, aa, ab, ac, bc, aab, aac, abc, ca, cb, cab\}.$$

Prove that if L is regular, then so is $subseq(L)$. (*subseq* is short for “subsequence.” Actually, it is known that if L is *any language whatsoever*, then $subseq(L)$ is regular.) *Hint:* You can do this either by transforming regular expressions or by transforming ε -NFAs.

! (not in the text; optional) Let x and y be any two strings. A *merge* of x and y is any string obtained by merging the symbols of x with those of y in some arbitrary way, maintaining the order of the symbols from each string. More exactly, if $|x| = m$ and $|y| = n$, then a string $z = z_1 \cdots z_k$ is a merge of x and y if and only if

- $k = m + n$,
- there exist $1 \leq i_1 < i_2 < \dots < i_m \leq k$ such that $x = z_{i_1} z_{i_2} \dots z_{i_m}$,
- there exist $1 \leq j_1 < j_2 < \dots < j_n \leq k$ such that $y = z_{j_1} z_{j_2} \dots z_{j_n}$,
and
- $\{i_1, \dots, i_m\} \cap \{j_1, \dots, j_n\} = \emptyset$.

For example, there are five different merges of the strings ab and bc :

abbc abcb babc bacb bcab

Let A and B be any languages over the same input alphabet Σ . Define

$A \text{ merge } B := \{z \in \Sigma^* \mid z \text{ is a merge of some } x \in A \text{ and some } y \in B\}$.

Show that if A and B are regular, then so is $A \text{ merge } B$. *Hint:* Given a DFA for A with r many states and an DFA for B with s many states, you can construct an *NFA* for $A \text{ merge } B$ with rs many states.