The Australian National University
Research School of Computer Science

Semester 1, 2022
Assignment 1 of 3

Theory of Computation

Release Date. Monday Mar 7 2022
Due Date. Monday March 21 2022, 23.50 Canberra time
Maximum credit. 50

Exercise 1 — Revisiting the Myhill-Nerode Theorem (5+5 credits)

Let L be a language over an alphabet Σ. Define the relation $R_\mathsf{L} \subseteq \Sigma^* \times \Sigma^*$ by $(u, w) \in R_\mathsf{L}$ if the following holds:

$$\forall x \in \Sigma^*(ux \in \mathsf{L} \iff wx \in \mathsf{L}).$$

We have seen in the tutorials that R_L is in fact an equivalence relation. We denote the equivalence class of $x \in \Sigma^*$ under this relation by $[x] = \{y \in \Sigma^* \mid (x, y) \in R_\mathsf{L}\}$. The Myhill-Nerode theorem states the following:

> **Myhill-Nerode Theorem.** *For any language L we have:*
>
> L *is regular if and only if R_L has a finite number of equivalence classes.*

In Tutorial 2 we have proved the direction from left to right. We will now prove the converse direction.

(a) Let $\mathsf{L} \subseteq \Sigma^*$ be a language and suppose that R_L has finitely many equivalence classes. Define the DFA $(Q, \Sigma, \delta, q_0, F)$ by

$$Q = \{[x] \mid x \in \Sigma^*\} \qquad \delta([x], a) = [xa]$$
$$q_0 = [\epsilon] \qquad F = \{[x] \mid x \in \mathsf{L}\}$$

and note that Q is finite by assumption.

  (i) Show that δ is well defined, i.e. if $[x] = [y]$, then $[xa] = [ya]$ for all $a \in \Sigma$.

  (ii) Prove by induction that $\widehat{\delta}([\epsilon], x) \in F$ if and only if $x \in L$.

(b) Prove that the DFA obtained in Part (a) is minimal. That is, given regular language L and a DFA A that accepts precisely L, the state set Q of A has at least as many states as the automaton constucted from R_L in Part (a).

Exercise 2 — Failure of Pumping Lemma (5+5 credits)

Unlike the Pumping Lemma, the Myhill-Nerode theorem exactly characterises the regular languages. As an illustration, consider the following language:

$$\mathsf{L} = \{a^j c^k a^\ell b^m \mid 0 \le j \le 5 \text{ and } k, \ell, m \in \mathbb{N} \text{ and } (k \ge 2 \Rightarrow \ell \ne m)\}.$$

(a) Prove that the Pumping Lemma fails to demonstrate that L is not regular. In detail, demonstrate that there exists $n \in \mathbb{N}$ such that any string $w \in \mathsf{L}$ with $|w| \ge n$ can be written as $w = xyz$ with $|xy| \le n$, $|y| > 0$ and $xy^iz \in \mathsf{L}$ for all $i \ge 0$.

(b) Use the Myhill-Nerode theorem to prove that L is not regular.

Exercise 3 — DFA-homomorphisms (8 + 2 credits)

In the lectures, we have talked about homomorphisms between *languages*. Here, we discuss homomorphisms of *automata*.

Definition. *Let $P = (Q, \Sigma, \delta, q_0, F)$ and $P' = (Q', \Sigma, \delta', q_0', F')$ be two DFAs with the same alphabet Σ. A* DFA*-homomorphism f from P to P' is a function $f : Q \to Q'$ which satisfies*

(1) $f(q_0) = q_0'$;

(2) $q \in F$ if and only if $f(q) \in F'$, for all $q \in Q$;

(3) $f(\delta(q, a)) = \delta'(f(q), a)$ for all $q \in Q$ and all $a \in \Sigma$.

(a) For a given automaton P as above, let $\mathscr{L}(P, q) = \{w \in \Sigma^* \mid \widehat{\delta}(q, w) \in F\}$ denote the words that P accepts when started in state $q \in Q$.

Show that if f is a DFA-homomorphism as above, then $\mathscr{L}(P, q) = \mathscr{L}(P', f(q))$ for all $q \in Q$.

(b) Hence, or otherwise, conclude that $L(P) = L(P')$ if there exists a DFA-homomorphism from P to P', where $L(P)$ denotes the language of P.

Exercise 4 **Algebra of Regular Expressions** (2 + 5 + 3 credits)

We say that an equation $r = s$ between regular expressions is *valid*, if $L(r) = L(s)$. Show the validity of the following equations between regular expressions.

- $(r^*)^* = r^*$
- $(r + s)^* = (r^* s)^* r^*$
- $(rs)^* = \epsilon + r(sr)^* s$

You may use the fact that $L^k \cdot L^* \subseteq L^*$ for any $k \geq 0$ and all $L \subseteq \Sigma^*$ as well as associativity, i.e. $(L \cdot K) \cdot M = L \cdot (K \cdot M)$ and the fact that $\{\epsilon\}$ is neutrial with respect to concatenation, i.e. $\{\epsilon\} \cdot L = L = L \cdot \{\epsilon\}$, where $L, K, M \subseteq \Sigma^*$ are languages. You should state and prove any other laws that you might require.

Exercise 5 **Buffalo** (3 + 4 + 3 credits)

This question is based on the fun fact that the English word "buffalo" has three meanings:

- A proper noun (always capitalised as "Buffalo"), referring to a location in New York
- A transitive verb, meaning "to bully"
- A common noun, meaning the animal, bison (or a group of such animals)

Because of its versatility, the following string is a grammatically correct English sentence:

"Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo."

It means: "New York bison, that other New York bison bully, also bully New York bison."

In the following, we are considering a context free grammar G over the alphabet $\{a, b\}$ that derives the above sentence by writing "Buffalo" for a and "buffalo" for b, initially without paying attention to the fact that the first word of a sentence is capitalised, which you are asked to remedy in part (c).

Formally, the grammar G is given by the following productions:

$$\begin{aligned} S &\to NVN \\ N &\to NNV \\ N &\to ab \\ N &\to b \\ V &\to b \end{aligned}$$

(a) Construct parse trees for the following string:

(i) $bbab$

(ii) $ababbabbbab$

(iii) $ababbbab$

(b) Is the above grammar ambiguous? Either prove that every $w \in L(G)$ only has one parse tree, or demonstrate that G is ambiguous by giving a string $w \in L(G)$ and two different parse trees with yield w.

(c) Define a grammar G' such that $L(G') = \{aw \mid aw \in L(G) \text{ or } bw \in L(G)\}$. That is, $L(G')$ arises from $L(G)$ by replacing the first letter of a string in $L(G)$ by the letter 'a'. You are just required to state the grammar, and a proof of $L(G) = L(G')$ is not required.