Transducers

Session 3: Logic of Transductions

Aliaume LOPEZ
TA mail* Course page† Exercises page‡

March 19, 2024

1 Previously, in Transducers

Exercise 1 (Aperiodicity and Counters). Let L be a regular language. Prove the equivalence between the following properties.

1. The minimal DFA of L is counter-free.
2. The syntactic monoid of L is aperiodic.

Assume that L is recognised by a counter-free automaton (that may not be minimal), is L aperiodic? What about a non-deterministic counter-free automaton?
▷ Hint 1
▷ Hint 2
▷ Solution 1 (From aperiodicity to counter-freeness)
▷ Solution 2 (From counter-freeness to aperiodicity)
▷ Solution 3 (Non-minimal counter-free automaton)

Exercise 2 (Fixed points). A fixed point of a function f is a value x such that $f(x) = x$. For the following models of computation, can we decide if f has a fixed point?

- Mealy Machines?
- Rational Transductions?
- Two-way Deterministic Transducers with outputs?

▷ Hint 3
▷ Solution 4 (Solution)

2 Logic

Exercise 3 (Kleene Star Stability). Are languages definable in first-order logic closed under kleene star?

*ad.lopez@uw.edu.pl
†https://www.mimuw.edu.pl/~bojan/2023-2024/przeksztalcenia-automatowe-transducers
‡https://aliaumel.github.io/transducer-exercices/

2.1 Examples of aperiodic languages

Write a star-free expression that defines the language $(ab)^*$.

2.2 A single existential quantifier is enough

Show that regular languages are definable by $\mathbb{MSO}$ formulas using a single existential monadic second order quantifier.
▷ Hint 4

A Hints

Hint 1 (Exercise 1 Use the transition monoid)**.** Prove that the transition monoid of the minimal DFA of L is the syntactic monoid of L.

Hint 2 (Exercise 1 Non-deterministic counter-free automaton)**.** Use the transition monoid to define what a counter should be.

Hint 3 (Exercise 2 What do you want to prove)**.**

- Mealy Machines: Yes, because the collection of fixed points is a regular language.
- Rational Transductions: **no**.

Hint 4 (Section 2.2 Encode the states with padding)**.** If the automaton has n states, then represent the state of the automaton for positions that are multiple of n using a unary encoding of the state plus a separator. How can you then recover the intermediate transitions?

B Solutions

Solution 1 (Solution to Exercise 1). Let us assume that $A = (Q, q_0, \delta, F)$ is the minimal DFA of L and that the syntactic monoid of L is aperiodic. Using the hint, we know that δ_w^n is eventually constant for all $w \in \Sigma^*$. As a consequence, if $q \in Q$ is such that $\delta(q, w^n) = q$, then $(\delta_w)^{kn}(q) = q$ for all $k \in \mathbb{N}$. If k is large enough, then $\delta_w^{kn} = \delta_w^{kn+1}$, and therefore $\delta_w(q) = \delta_w(\delta_w^{kn})(q) = \delta_w^{kn}(q) = q$. We have proven that A has no counters.

Solution 2 (Solution to Exercise 1). Assume that the minimal DFA $A = (Q, q_0, \delta, F)$ recognising L is counter-free. Let $w \in \Sigma^*$. We will prove that the sequence δ_w^n is eventually constant. Let $q \in Q$, there exists $i < j$ such that $\delta_w^i(q) = \delta_w^j(q)$. Let $q' := \delta_w^i(q)$, then $\delta_w^{j-i}(q') = q'$. Since A is counter-free, we conclude that $\delta_w(q') = q'$. In particular, the sequence $\delta_w^n(q)$ is eventually constant. Now, because Q is finite, the sequence δ_w^n is itself eventually constant. And because there are finitely many functions δ_w there exists a uniform bound N_0 such that $\delta_w^n = \delta_w^m$ for all $n, m \geq N_0$ and all $w \in \Sigma^*$.

Solution 3 (Solution to Exercise 1). If A is a counter-free automaton that recognises L, then the minimal DFA recognising L is also counter-free.

Solution 4 (Solution to Exercise 2). For Mealy Machines, the output is letter-to-letter, so if a fixed point exists, it must start with a transition that produces exactly the letter that is read. This means that it has a fixed point if and only if it has a fixed point of length 1.

For rational transductions, the problem is undecidable because it is equivalent to the halting problem for Turing Machines. Let M be a Turing Machine, such that a configuration of M terminates.

Consider the function $s_M\colon \Sigma^* \to \Sigma^*$ that maps an encoding of a configuration of M to the encoding of the successor configuration. Let $f_M\colon \Sigma^* \to \Sigma^*$ be the rational function that maps a sequence of configurations to the sequence of **successor** configurations, prepending to the result the initial configuration of M.

A terminating run of M is a fixed point of M. Conversely, if f_M has a fixed point, then it must be a valid run of M (successor configurations are correctly computed), and this run cannot be continued (otherwise it would not be a fixed point). Therefore, the problem of deciding whether f_M has a fixed point is equivalent to the halting problem for M.