Introduction to Language Theory and Compilation: Exercises

Session 1: Regular languages

ULB

About these sessions

- Assistants: Jean-Sébastien
  E-mail: `Jean-Sebastien.Lerat@ulb.ac.be`
  Office: P.NO.2.O8.212
- Resources on *Université Virtuelle*:
  `webctapp.ulb.ac.bel`
- Exercises originally by Gilles Geeraerts, Sébastien Collette, Anthony Piron, Joël Goossens, etc.
- Slides based on former slides by Gilles Geeraerts, translated by Markus Lindström

Let Σ[1] be a (finite) alphabet. A *language* is a set of *words* defined on a given alphabet. Let L, L_1 and L_2 be languages, we can then define some operations:

Definition (Union)

$L_1 \cup L_2 = \{w \mid w \in L_1 \text{ or } w \in L_2\}$

Definition (Concatenation)

$L_1 \cdot L_2 = \{w_1 w_2 \mid w_1 \in L_1 \text{ and } w_2 \in L_2\}$

Definition (Kleene closure)

$L^* = \{\varepsilon\} \cup \{w \mid w \in L\} \cup \{w_1 w_2 \mid w_1, w_2 \in L\} \cup \cdots$

[1] Sigma

Regular languages

Regular languages are defined inductively:

Definition (Regular language)

- $\emptyset$ is a regular language
- $\{\varepsilon\}$ is a regular language
- For all $a \in \Sigma$, $\{a\}$ is a regular language

If L, L_1, L_2 are regular languages, then:

- $L_1 \cup L_2$ is a regular language
- $L_1 \cdot L_2$ is a regular language
- L^* is a regular language

Finite automata (FA)

$M = \langle Q, \Sigma, \delta, q_0, F \rangle$ where:

- Q is a *finite* set of *states*
- Σ is the *input alphabet*
- δ is the *transition function*
- $q_0 \in Q$ is the *start state*
- $F \subseteq Q$ is the set of *accepting states*

M is a *deterministic* finite automaton (DFA) if the transition function $\delta : Q \times \Sigma \to Q$ is total. In other words, on each input, there is *one and one only* state to which the automaton can transition from its current state.

Exercise 1

Consider the alphabet $\Sigma = \{0, 1\}$. Using the inductive definition of regular languages, prove that the following languages are regular:

1. The set of words made of an arbitrary number of ones, followed by 01, followed by an arbitrary number of zeroes.
2. The set of odd binary numbers.

Exercise 2

1. Prove that any finite language is regular.
2. Is the language $L = \{0^n 1^n \mid n \in \mathbb{N}\}$ regular? Explain.

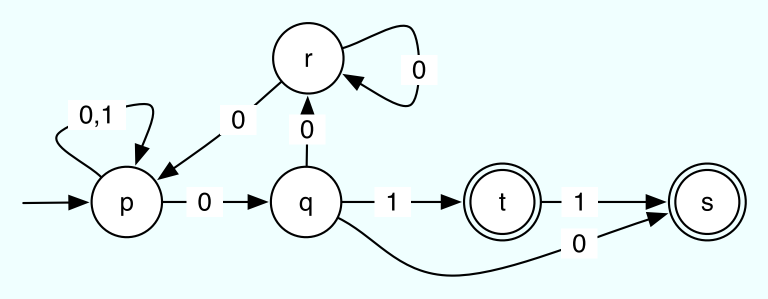
Exercise 3

For each of the following languages (defined on the alphabet $\Sigma = \{0, 1\}$), design a nondeterministic finite automaton (NFA) that accepts it.

1. The set of strings ending with 00.
2. The set of strings whose 10^{th} symbol, counted from the end of the string, is a 1.
3. The set of strings where each pair of zeroes is followed by a pair of ones.
4. The set of strings not containing 101.
5. The set of binary numbers divisible by 4.

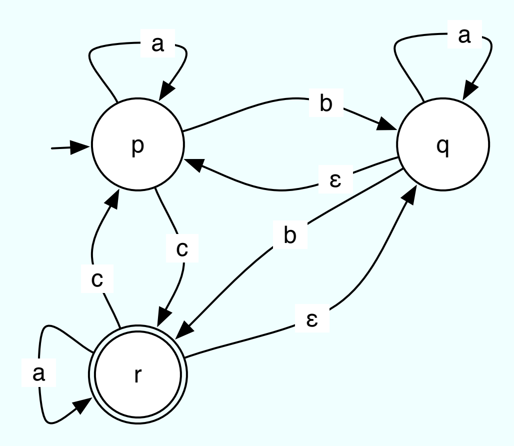
Exercise 4.1

Transform the following NFA into a DFA:



Exercise 4.3

Transform the following ε-NFA into a DFA:



Exercise 5

Write a C function that implements the following automaton and returns the accepting state number.