Finite automata

Nikita

1 DFAs

This week, we will study a simple computational device called a *discrete finite automaton*, or DFA. Given a sequence of letters (a "string"), this device will either accept or reject it.

A DFA is defined as an abstract mathematical concept, but is often implemented in hardware and software for solving various specific problems such as lexical analysis and pattern matching. For example, a DFA can model software that decides whether or not online user input such as email addresses are syntactically valid. Your favourite text editor creates a DFA each time when you run the "find and replase" function.

Consider the automaton A shown below:

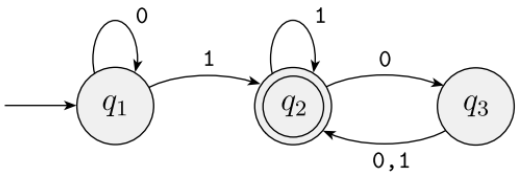

Figure 1: Automaton A

A always starts in the state q_1. This is called the *start state*. It takes strings using letters in the alphabet $\{0, 1\}$ and reads them left to right, moving between states along the edges marked by each letter.
For example, consider the string `1011`. Processing this string, A will go through the states $q_1 - q_2 - q_3 - q_2 - q_2$.
Note that q_2 has a circle in the diagram above. This means that the state q_2 is *accepting*, and that all the strings which end up in it are *accepted*. Similarly, states q_1 and q_3 are *rejecting* and the strings which end up there are *rejected*.

Problem 1.
Which of the following strings are accepted by A?
(a) 1

(b) 1010

(c) 1110010

(d) 1000100?

Problem 2.
Describe the general form of a string accepted by A. (Hint: work backwards from the

accepting state, and decide what all the strings must look like at the end in order to be accepted. Where can they come from?)

Now consider the automaton B, which uses the alphabet $\{a, b\}$, starts in the state s, and has two accepting states q_1 and r_1.

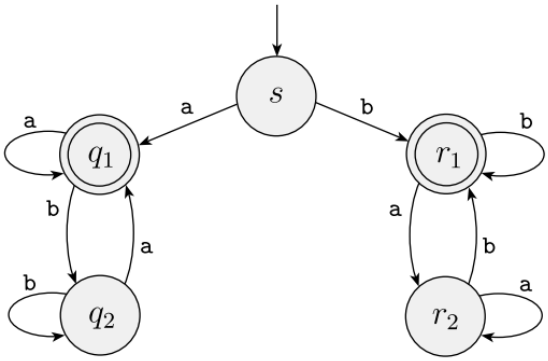

Figure 2: Automaton B

Problem 3.
Which of the following strings are accepted by B:

(a) *aa*

(b) *abba*

(c) *abbba*

(d) *baabab*?

Problem 4.
Describe the strings accepted by B.

Definition 1.
An *alphabet* is any finite set of symbols.
A *string* over an alphabet Q is any finite sequence of symbols from Q. This includes the empty string, which is denoted by ε.
We denote by Q^* the set of all possible strings over Q. For example $\{0, 1\}^*$ is the set of all binary strings and $\{a, b, \ldots, z\}^*$ is the set of all strings containing only English letters.
A *language* over an alphabet Q is a set of strings. For example, all strings in $\{0, 1, \ldots, 9\}^*$ which are decimal representations of prime numbers form a language.
A language L is *recognized* by a DFA if this DFA accepts all the strings from L and only them.

Remark 1.
A machine, such as DFA or Turing machine, may accept several strings, but it always recognizes only one language. If the machine accepts no strings, it still recognizes one language — namely, the empty language $\emptyset$.

Problem 5.
How many strings of length n are accepted by the automaton C?

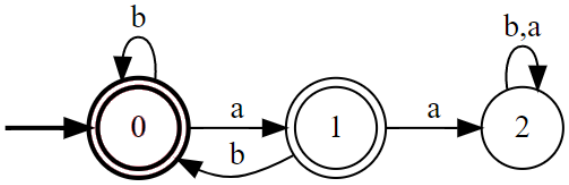

Figure 3: Automaton C

Remark 2.
Note that all the states in our DFAs A, B and C from figures 1, 2, 3 have outgoing symbols for each letter of the alphabet. Do the same for your DFAs.

Problem 6.
Draw state diagrams of DFAs recognizing the following languages. In all parts, the alphabet is $\{0, 1\}$:

(a) $\{w|w$ begins with a 1 and ends with a $0\}$

(b) $\{w|w$ contains at least three 1s$\}$

(c) $\{w|w$ contains the substring 0101 (i.e., $w = x0101y$ for some x and $y)\}$

(d) $\{w|w$ has length at least 3 and its third symbol is a $0\}$

(e) $\{w|w$ starts with 0 and has odd length, or starts with 1 and has even length$\}$

(f) $\{w|w$ doesn't contain the substring 110$\}$

Problem 7.
Draw a DFA over an alphabet $\{a, b, @, .\}$ recognizing the language of strings of the form *user*@*website*.*domain*, where *user*, *website* and *domain* are nonempty strings over $\{a, b\}$ and *domain* has length 2 or 3.

Problem 8.
Draw a state diagram for a DFA over an alphabet of your choice which recognizes exactly $f(n)$ strings of length n if

(a) $f(n) = n$

(b) $f(n) = n + 1$

(c) $f(n) = 3^n$

(d) $f(n)$ is a *Tribonacci number* defined by the rules $f(0) = 0$, $f(1) = 1$, $f(2) = 1$ and $f(n) = f(n-1) + f(n-2) + f(n-3)$ for $n \geq 3$.

(e) $f(n) = n^2$.

Problem 9.
(a) Draw a DFA recognizing the language of strings over $\{0, 1\}$, which has 0 as a third digit from the end.

(b*) Prove that any such DFA would have ≥ 8 states.

2 Regular languages

Definition 2.
A language is called *regular* if it is recognized by some DFA.

Problem 10.
(a) Draw a DFA over an alphabet $\{A, B\}$ accepting strings which do not start and end with the same letter.

(b) Prove that for any regular language L over an alphabet Q its complement $\overline{L} = Q^* \setminus L$ is also regular.

Problem 11.
(a) Draw a DFA over an alphabet $\{A, B\}$ accepting strings which do not start and end with the same letter AND have an even length.

(b) Prove that for any regular languages L_1, L_2 over an alphabet Q their union and intersection are also regular.

However, not all the languages are regular. You will later see that the language consisting of palindromes is not regular. DFAs have some special properties, and if we have a language that does not satisfy the conclusion of the theorem below, we can stop our search to try to construct a DFA for it since it is not regular.

Theorem 1 (Pumping lemma)**.**
If A is a regular language, then there is a number p (the pumping length) where if s is any string in A of length at least p, then s may be divided into three pieces, $s = xyz$, satisfying the following conditions:

1. for each $i \geq 0$, $xy^iz \in A$,
2. $|y| > 0$, and
3. $|xy| \leq p$.

Here $|s|$ represents the length of string s (assuming ε has length 0), y^i means that i copies of y are concatenated together, and y^0 equals ε. When s is divided into xyz, either x or z may be ε, but condition 2 says that $y \neq \varepsilon$. Observe that without condition 2 the theorem would be trivially true.

Problem 12.
(a) Check that pumping lemma holds for the language recognized by the automaton C from Figure 3 and pumping length $p = 2$.

(b) Prove the pumping lemma. *Hint: look at the first cycle in the DFA you get while reading the word.*

Problem 13.
Show that the following languages are not regular:
(a) $\{0^n1^n\}$ over $\{0, 1\}$;

(b) Let $\Sigma = \{0, 1, +, =\}$ and

$ADD = \{"x = y + z" \mid x, y, z$ are binary integers, and x is the sum of y and $z\}$;

(c) Language of all palindromes over Latin alphabet.

Problem 14.
For a word w over an alphabet $\{a, b\}$ denote by $|w|_a$ and $|w|_b$ the amount of letters a and b respectively inside w.

(a) Prove that the language $L_p = \{w, \text{ s.t. } p \text{ divides } |w|_a - |w|_b\}$ is regular for any prime p.

(b) Prove that $L = \{w, \text{ s.t. } |w|_a - |w|_b = \pm 1\}$ is not regular.

(c) Prove the infinitude of primes.