Finite Automata

Prepared by Mark and Nikita on August 16, 2023

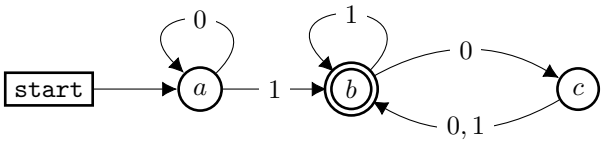
Instructor's Handout

This file contains solutions and notes.
Compile with the "nosolutions" flag before distributing.

Part 1: DFAs

This week, we will study computational devices called *deterministic finite automata*.
A DFA has a simple job: it will either "accept" or "reject" a string of letters.

Consider the automaton A shown below:



A takes strings of letters in the alphabet $\{0, 1\}$ and reads them left to right, one letter at a time. Starting in the state a, the automaton A will move between states along the edge marked by each letter.

Note that node b has a "double edge" in the diagram above. This means that the state b is *accepting*. Any string that makes A end in state b is *accepted*. Similarly, strings that end in states a or c are *rejected*.

For example, consider the string `1011`.
A will go through the states $a - b - c - b - b$ while processing this string.

Problem 1:
Which of the following strings are accepted by A?

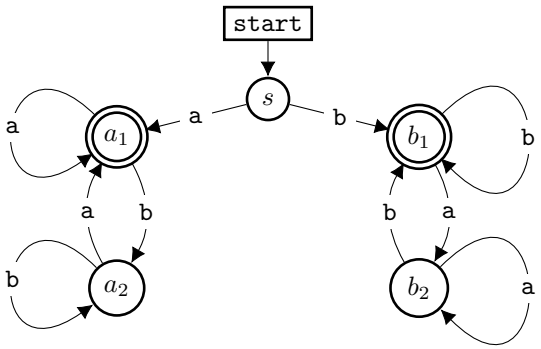
- `1`
- `1010`
- `1110010`
- `1000100`

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.

Solution

A will accept strings that contain at least one 1 and end with an even (possibly 0) number of zeroes.

Now consider the automaton B, which uses the alphabet $\{a, b\}$.
It starts in the state s and has two accepting states a_1 and b_1.



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

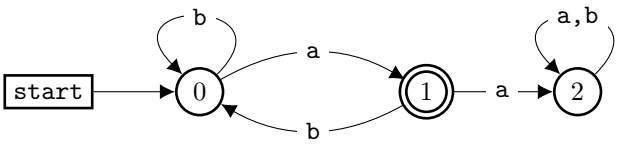
- `aa`
- `abba`
- `abbba`
- `baabab`

Problem 4:
Describe the strings accepted by B.

Solution

B accepts strings that start and end with the same letter.

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



Solution

If A_n is the number of accepted strings of length n, then $A_n = A_{n-1} + A_{n-2}$.
Computing initial conditions, we see that A_n is an $n+2$-th Fibonacci number.

Definition 1:
An *alphabet* is a finite set of symbols.

Definition 2:
A *string* over an alphabet Q is a finite sequence of symbols from Q.
We denote the empty string ε.

Q^* is the set of all possible strings over Q.
For example, $\{0,1\}^*$ is the set $\{\varepsilon, 0, 1, 00, 01, 10, 11, 000, ...\}$
Note that this set contains the empty string.

Definition 3:
A *language* over an alphabet Q is a subset of Q^*.
For example, the language "strings of length 2" over $\{0,1\}$ is $\{00, 01, 10, 11\}$

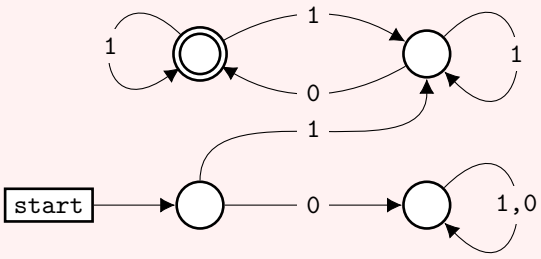
Definition 4:
We say a language L is *recognized* by a DFA if that DFA accepts a string w if and only if $w \in L$.

Problem 6:
Draw DFAs that recognize the following languages. In all parts, the alphabet is $\{0, 1\}$:

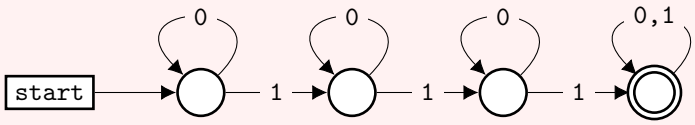
- $\{w \mid w$ begins with a `1` and ends with a `0`$\}$
- $\{w \mid w$ contains at least three `1`s$\}$
- $\{w \mid w$ contains the substring `0101` (i.e, $w = x$`0101`y for some x and y)$\}$
- $\{w \mid w$ has length at least three and its third symbol is a `0`$\}$
- $\{w \mid w$ starts with `0` and has odd length, or starts with `1` and has even length$\}$
- $\{w \mid w$ doesn't contain the substring `110`$\}$

Solution

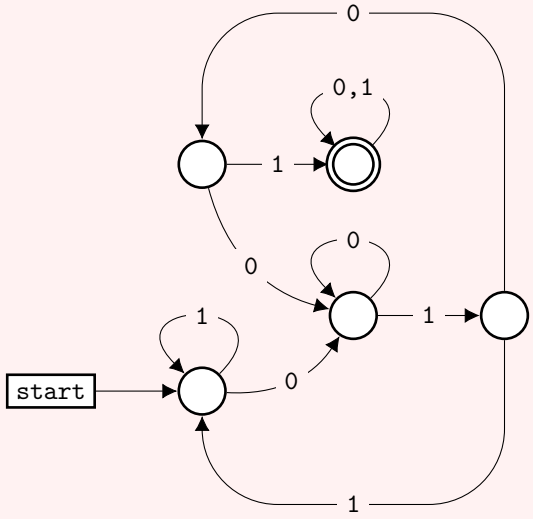
$\{w \mid w$ begins with a `1` and ends with a `0`$\}$



$\{w \mid w$ contains at least three `1`s$\}$

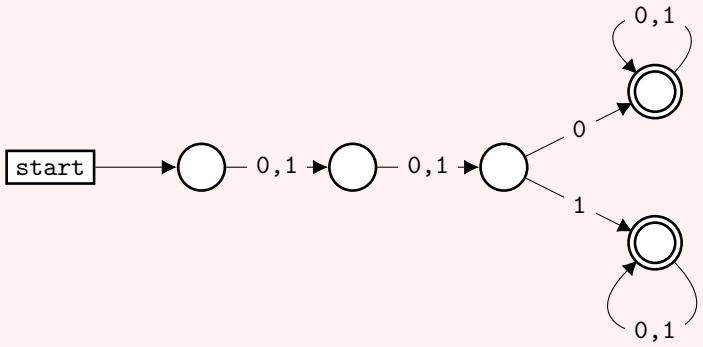


$\{w \mid w$ contains the substring `0101`$\}$

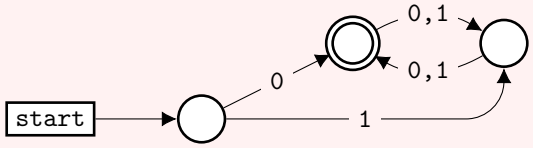


Notice that after getting two 0's in a row we don't reset to the initial state.

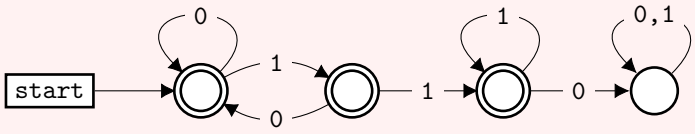
$\{w \mid w$ has length at least three and its third symbol is a 0$\}$



$\{w \mid w$ starts with 0 and has odd length, or starts with 1 and has even length$\}$



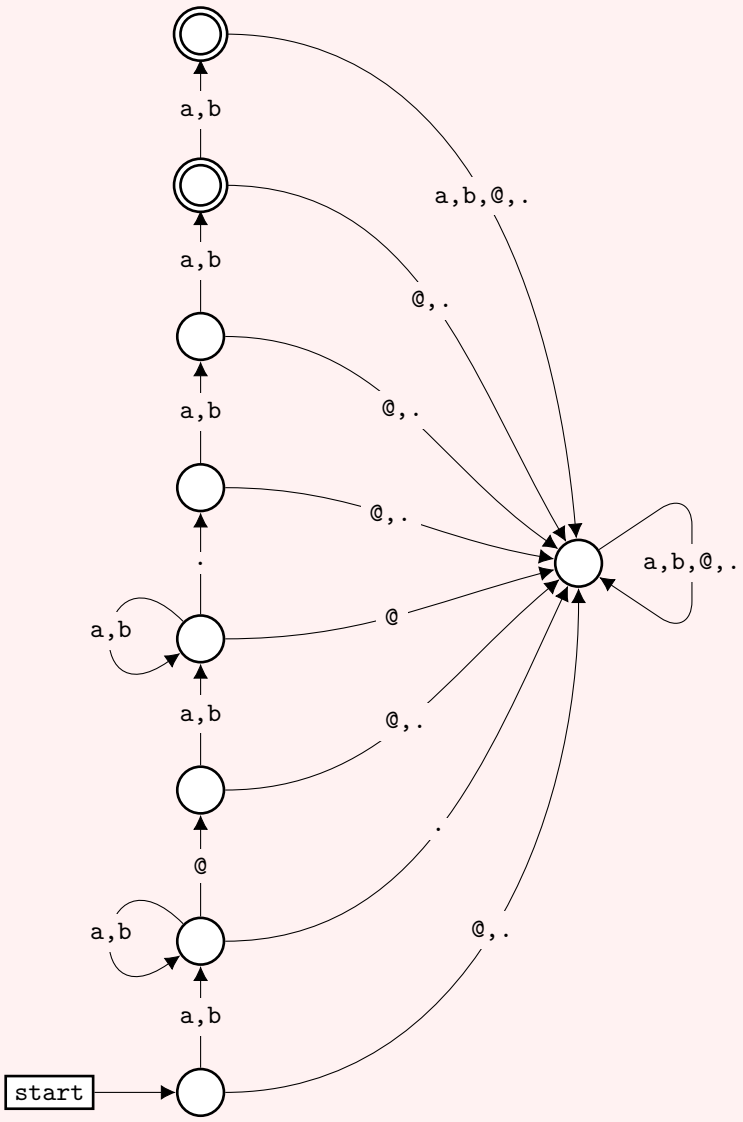
$\{w \mid w$ doesn't contain the substring 110$\}$



Notice that after getting three 1's in a row we don't reset to the initial state.

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.

Solution

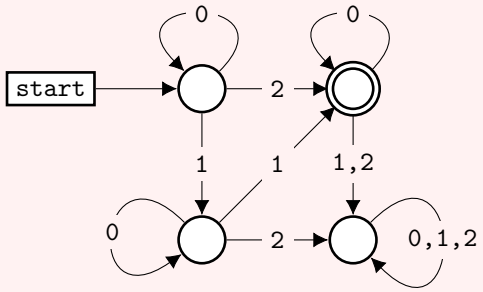


Problem 8:
Draw a state diagram for a DFA over an alphabet of your choice that accepts exactly $f(n)$ strings of length n if

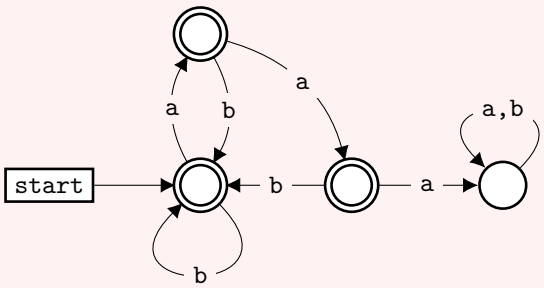
- $f(n) = n$
- $f(n) = n + 1$
- $f(n) = 3^n$
- $f(n) = n^2$
- $f(n)$ is a Tribonacci number.
  Tribonacci numbers are defined by the sequence $f(0) = 0$, $f(1) = 1$, $f(2) = 1$, and $f(n) = f(n-1) + f(n-2) + f(n-3)$ for $n \geq 3$
  Hint: Fibonacci numbers are given by the automaton prohibiting two `'a'`s in a row.

Solution

Part 4: $f(n) = n^2$
Consider the language of words over $\{0, 1, 2\}$ that have the sum of their digits equal to 2.
Such words must contain two ones or one two:

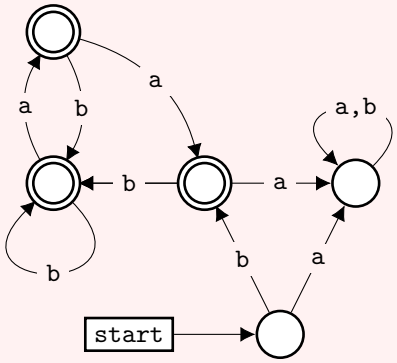


Part 5: Tribonacci numbers
Using the hint, we get the following automaton.
It rejects all strings with three `'a'`s in a row.



This automaton rejects all strings with three `'a'`s in a row. If we count accepted strings, we get the Tribonacci numbers with an offest: $f(0) = 1$, $f(1) = 2$, $f(2) = 4$, ...

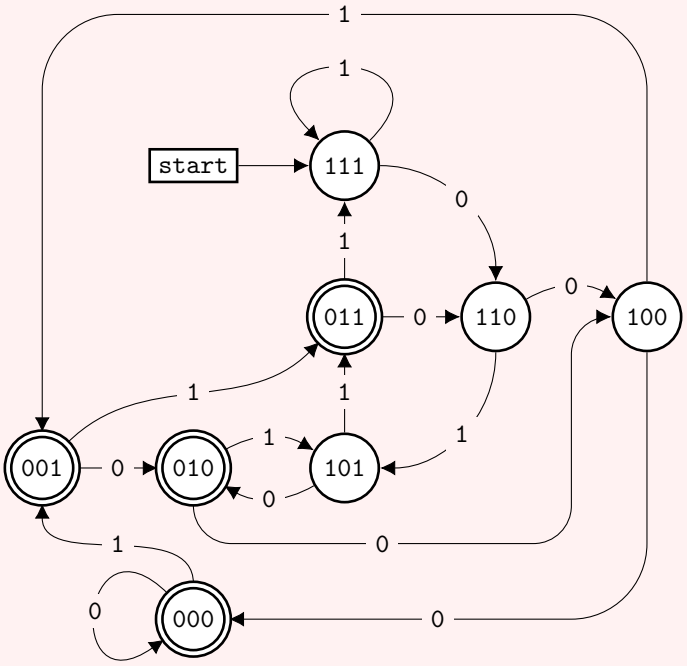
We can fix this by adding a node and changing the start state:



Problem 9:
Draw a DFA recognizing the language of strings over $\{0, 1\}$ in which 0 is the third digit from the end. Prove that any such DFA must have at least 8 states.

Solution

Part 1:
Index the states by three-letter suffixes 000, 001, ..., 111. All strings that end with letters $d_1d_2d_3$ will end up in the state $d_1d_2d_3$. We accept all states that start with a 0.
Note that we can start at any node if we ignore strings with fewer than three letters.



Part 2:
Strings 000, 001, ..., 111 must lead to pairwise different states.

Assume 101 and 010 lead to the same state. Append a 1 to the end of the string.
101 will become 011, and 010 will become 101. These must be different states, since we accept 011 and reject 101. We now have a contradiction: one edge cannot lead to two states!

101 and 010 must thus correspond to distinct states.
We can repeat this argument for any other pair of strings.

Part 2: Regular languages

Definition 5:
We say a language is *regular* if it is recognized by some DFA.

Problem 10:
Draw a DFA over $\{A, B\}$ that accepts strings which do not start and end with the same letter.
Hint: Modify the DFA in Problem 4.

Problem 11:
Let L be a regular language over an alphabet Q.
Show that $Q^* - L$ is also regular.
Hint: $Q^* - L$ is the set of objects in Q^* but not in L. This is often called the *complement* of L.

Solution

Invert accepting and rejecting states.

Problem 12:
Draw a DFA over the alphabet $\{A, B\}$ that accepts strings which have even length and do not start and end with the same letter.

Problem 13:
Let L_1, L_2 be two regular languages over an alphabet Q.
Show that their union and intersection are also regular.

Solution

Consider a product of automatons where each state is a pair of states in the first and second automaton and every transition works if it was applied to both elements in pair.

For union, we call the state (s_1, s_2) accepting if s_1 OR s_2 is accepting in their respective automaton.

For intersection, we call it accepting if s_1 AND s_2 are accepting in their respective automaton.

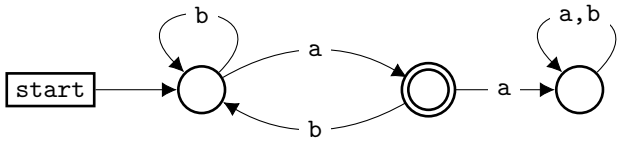
Theorem 1: Pumping Lemma
Let A be a regular language.
There then exists a number p, called the *pumping length*, so that any string $s \in A$ of length at least p may be divided into three pieces $s = xyz$ satisfying the following:

- $|y| > 0$ *Hint:* In other words, the segment y is not the empty string.
- $|xy| \leq p$. *Hint:* $|s|$ is the length of a string.
- $\forall i > 0$, $xy^iz \in A$ *Hint:* y^i means that y is repeated i times. y^0 is the empty string.

When s is divided into xyz, either x or z may be the empty string, but y must not be empty.
Notice that without the first condition, this theorem is trivially true.

In english, the pumping lemma states that in any regular language, any string of sufficient length contains a substring that can be "pumped" (or repeated) to generate more strings in that language.

Problem 14:
Check that the pumping lemma holds with $p = 3$ for the following DFA.
Hint: This is the same DFA as in Problem 5. What kind of strings does it accept?



Problem 15:
How can we use the pumping lemma to show that a language is **not** regular?

Problem 16:
Prove the pumping lemma.
Hint: Look at the first cycle in the DFA you get while reading s.

Solution

Look at the first place where we come to an already visited state while reading the word. Say the first time we came to this state after reading x and the second time after reading xy. Then y doesn't move us from this state and we can omit it or repeat any number of times we want.

Problem 17:
Show that the following languages are not regular:

A: $\{0^n 1^n \mid n \in \mathbb{Z}_0^+\}$ over $\{0, 1\}$, which is the shorthand for the set $\{\varepsilon, 01, 0011, \dots\}$

B: The language ADD over the alphabet $\Sigma = \{0, 1, +, =\}$ where
ADD = { "x=y+z" | x, y, z are binary integers, and x is the sum of y and z}

C: The language of all palindromes over the english alphabet

Solution

Part A:
Assume this language is regular. Let p be the pumping length. The string $0^p 1^p$ must then be accepted, implying that the string $0^{p-|y|}1^p$ (or $0^{p+|y|}1^p$) is also accepted.

Part B:
By pumping $10^{p+1} = 10^p + 10^p$

Part C:
By pumping $a^p b a^p$

Definition 6:
Let w be a string over an alphabet A.
If $a \in A$, $|w|_a$ is the number of times the letter a occurs in w.

For the following problems, we will use the alphabet $\{a, b\}$.

Problem 18:
Show that the language $L_p = \left\{ w \;\middle|\; p \text{ divides } |w|_a - |w|_b \right\}$ is regular for any prime p.

Problem 19:
Show that $L = \left\{ w \;\middle|\; |w|_a - |w|_b = \pm 1 \right\}$ is not regular.

Problem 20:
Prove that there are infinitely many primes.

Solution

`https://www.jstor.org/stable/48661886`