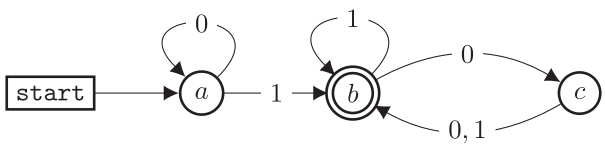
Finite Automata

Prepared by Mark and Nikita for Winter 2023 Advanced 2
Adapted by Sunny and Curtis for Winter 2024 Advanced 1

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*.

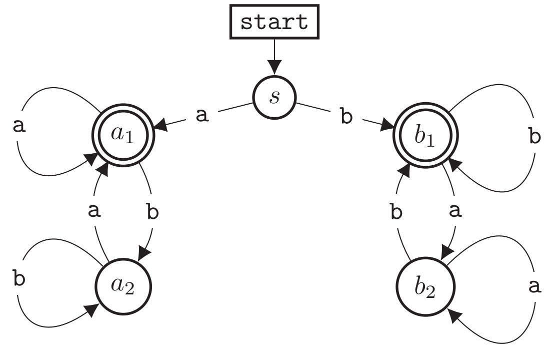
For example, consider the string `1011`.
A will go through the states $a - b - c - b - b$ while processing this string.

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.

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.

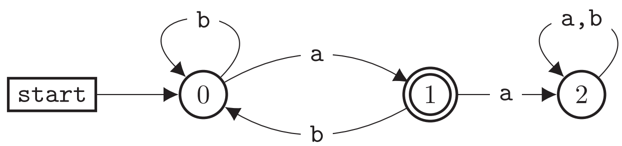


Problem 3:
Which of the following strings are accepted by B?

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

Problem 4:
Describe the strings accepted by B.

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



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, $w = x\texttt{0101}y$ 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`$\}$

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

Problem 8:
Construct a DFA to check whether a number written in binary is divisible by 2.

Problem 9:
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.

Problem 10:
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.

Problem 11:

Build a DFA that takes binary strings and accepts it iff it is divisible by 6 when interpreted as a binary whole number (e.g. 10010 is 18 in decimal so it should be accepted. 0 and 0010010 should also be accepted). Read the string from most to least significant digit.