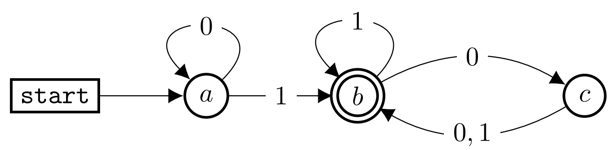
Finite Automata: Regular Languages

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

Quick review of *deterministic finite automata* (DFA):
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*.

Here are some definitions from last week that may be useful:

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'll denote the empty string ε.

Definition 3:
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 4:
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 5:
The language *recognized* by a DFA is the set of strings that the DFA accepts.

Here is a new definition which will be the focus of this handout:

Definition 6:
We say a language is *regular* if it is recognized by some DFA.

Problem 1:
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. Most of you should remember this from last week.

Problem 2:
Draw a DFA over $\{A, B\}$ that accepts strings which do not start and end with the same letter.
Hint: It may help to refer to last week's packet. Do you remember a very similar problem?

Problem 3:
Let L be a regular language over an alphabet Q.
Show that $Q^* - L$ is also regular.
Hint: Consider some DFA and its corresponding regular language. How might you modify it to get the *complement* of that language?

Problem 4:
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 5:
Let L_1, L_2 be two regular languages over an alphabet Q.
Explain why their union and intersection are also regular.
Hint: Consider a Cartesian product of two automata, that is, a set of ordered pairs where each coordinate corresponds to a different automaton.

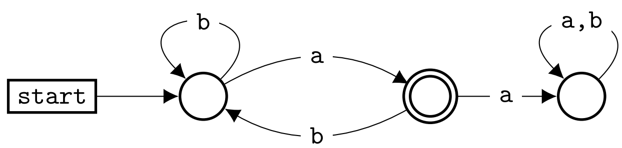
Theorem 1: Pumping Lemma
Let A be a regular language.
There then exists a number $p \geq 1$, called the *pumping length*, so that any string $s \in A$ of length at least p may be divided into three pieces $s = xyz$ satisfying the following:

- $|y| > 1$ *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 6:
Check that the pumping lemma holds with $p = 3$ for the following DFA.
Hint: This was our Fibonacci DFA from last week. What kind of strings does it accept?



Problem 7:
Consider the language of strings of length 1 on the typical English alphabet. Show that the pumping lemma holds by choosing an appropriate p.
Hint: The pumping lemma may sometimes be hold only *vacuously*. Ask if you don't know what that means.

Problem 8:
Prove the pumping lemma, or at least explain intuitively why it works.
Hint: Think about this in terms of DFAs. How does the structure of some DFAs allow repeated substrings?

Problem 9:

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

Hint: You can think of it like a game where your opponent is trying to prove that a given language is regular, and you are trying to prove that it is not. Use the next problem as practice. Ask your instructors for a general strategy if you cannot come up with one.

Problem 10:

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 over $\{0, 1\}$ such that there are an equal number of 0s and 1s.

C: $\{w \mid w = 1^n$, n is a perfect square$\}$

D: The language of all palindromes over the english alphabet

Definition 7:
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 11:
Show that the language $L_p = \left\{ w \;\middle|\; p \text{ divides } |w|_a - |w|_b \right\}$ is regular for any prime p.

Problem 12:
Show that $L = \left\{ w \;\middle|\; |w|_a - |w|_b = \pm 1 \right\}$ is not regular.

Problem 13:
Use the previous 2 problems to prove that there are infinitely many primes.