Recognizing Languages

Discrete Mathematics
Evgeny Skvortsov

Previous Lecture

- Grammars
- Finite automata

Finite Automata: Definition

- A finite automaton consists of:
  - a set of internal states S
  - the input alphabet Σ
  - the transition function ν
  - the initial state s_0
  - the set of accepting states $F \subseteq S$
- The transition function is a collection of rules of the form

$$(s,a) \to s'$$

where s is the current state of the automaton, $a \in \Sigma$ is an input symbol, and s' is the new state

The transition function contains such a rule for every pair from $S \times \Sigma$

Finite Automata: Definition (cntd)

- Input of the automaton is a string w
- The automaton starts in the initial state
- The automaton reads w from left to right
- After reading each symbol it changes its state accordingly to the transition function
- It stops after reaching the end of w

- Example: S = { s_0, s_1 }, Σ = {a,b}, s_0 is the initial state

  v =
  { (s_0,a) ® s_1, (s_0,b) ® s_0,
  (s_1,a) ® s_0, (s_1,b) ® s_1 }

  Input: aabaa

Recognizing Languages

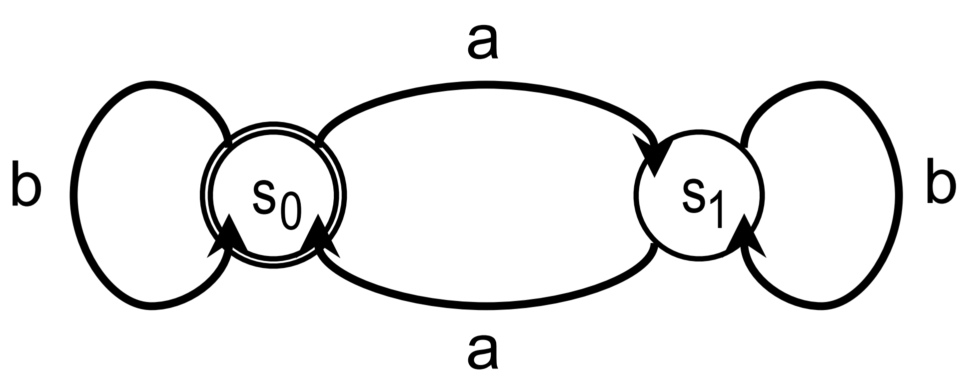
- A finite automaton accepts a string w, if its work with input w ends up in an accepting state
- Example (see previous slide): $F = \{ s_0 \}$
  Inputs: aabaa, abbaba

- Let M be a finite automaton
  $$L(M) = \{w \mid M \text{ accepts } w\}$$
  is the language accepted by M

- What is the language accepted by the automaton from the previous slide?

Graph of a Finite Automaton

- It is convenient to represent finite automata as directed graphs
- States correspond to vertices of the graph
- Every rule $(s,a) \to s'$ of the transition function is represented by an arc (or edge) from s to s' labeled by the symbol a
- The initial state is always s_0
- Final states we encircle twice



Examples

- Draw the graph of the automaton:

  $S = \{s_0, s_1, s_2, s_3\}$, $\Sigma = \{a,b\}$, $F = \{ s_1 \}$

  $v = \{$ (s_0,a) ® s_1, (s_0,b) ® s_3,
  (s_1,a) ® s_2, (s_1,b) ® s_3,
  (s_2,a) ® s_3, (s_2,b) ® s_1,
  (s_3,a) ® s_3, (s_3,b) ® s_3 $\}$

  What language does it accept?

- Construct a finite automaton that accepts the language $a^* b a^* b a^*$

Transducers

- Sometimes we want a finite automaton to output something
- A transducer is a finite automaton with two extra features
  - an output alphabet, Δ
  - the transition function is a collection of rules of the form

$$(s,a) \to (s',a')$$

  where s is the current state of the automaton, $a \in \Sigma$ is an input symbol, s' is the new state, and $a' \in \Delta$ is an output symbol

- At each step, when applying a rule $(s,a) \to (s',a')$ the automaton not only changes the current state, but also outputs a', so producing a string over the output alphabet

Example

- Describe the work of the automaton:

  $S = \{ s_0, s_1, s_2 \}$, $\Sigma = \{a,b\}$, $\Delta = \{a,b\}$, s_0 is the initial state

  $v = \{ (s_0,a) ® (s_0,a), \quad (s_0,b) ® (s_1,a),$
  $(s_1,a) ® (s_2,a), \quad (s_1,b) ® (s_1,a),$
  $(s_2,a) ® (s_0,a), \quad (s_2,b) ® (s_1,b) \}$

  on input babaab

Another Example

- Serial binary adder: an automaton that given binary expansions of two integers outputs the expansion of their sum. Assume that the least significant bits of the expansions go first.

What Finite Automata Cannot Do

- Consider the language $L = \{ a^k b^k \mid k \in N\}$.
- If we try to construct a recognizing finite automaton for this language we find the most difficult thing is to synchronize the powers of a and b

Theorem

L is not recognizable by any finite automaton.

Proof

By contradiction.

Suppose that L is recognized by a finite automaton

$M = (S, \Sigma, v, s_0, F)$, and $|S| = n$.

Proof (cntd)

- Run the automaton on input $a^{n+1}b^{n+1}$

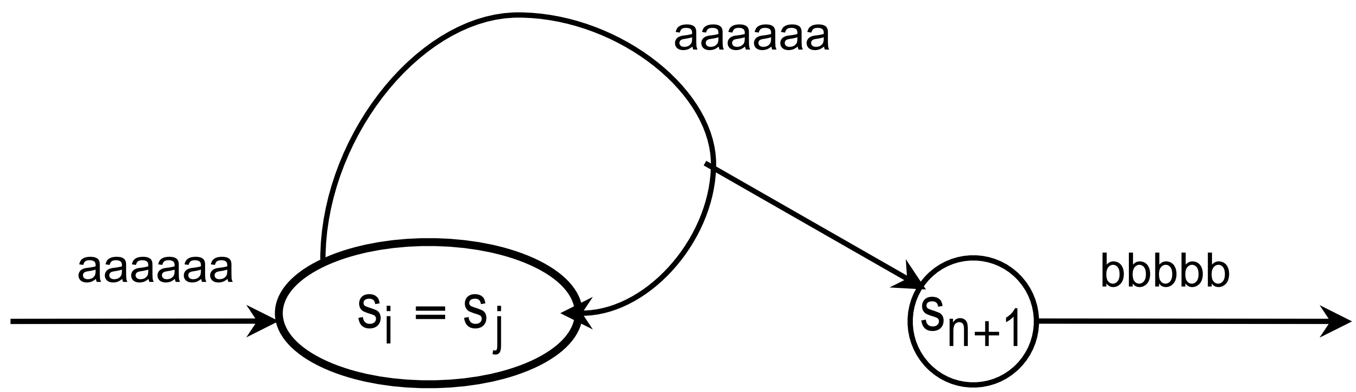
  The automaton accepts this input

  While working on the input it goes through some states. Let us denote the sequence of states by

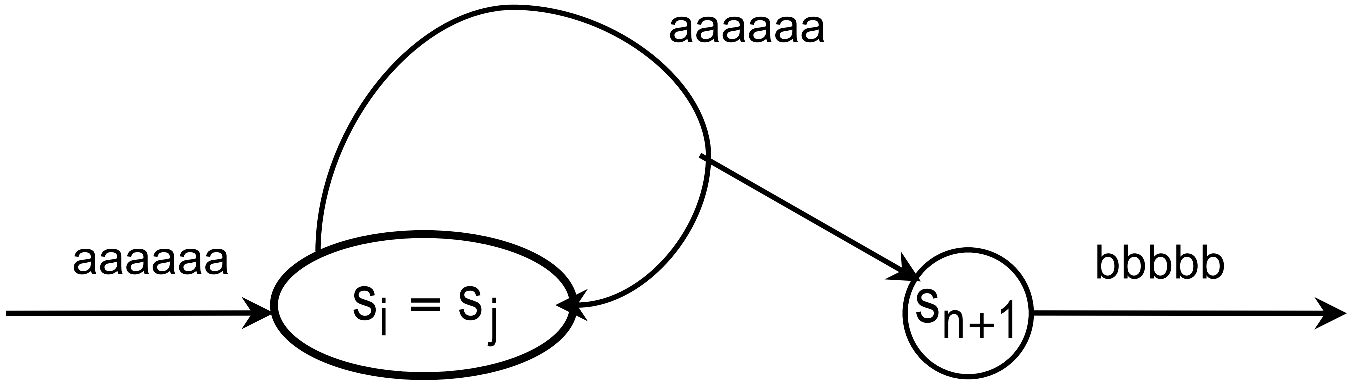
  input a a a a a … a b …

  states s_1 s_2 s_3 s_4 s_5 … s_{n+1} …

  Since $|S| = n$, by the pigeonhole principle $s_i = s_j$ for some $i \neq j$



Proof (cntd)



- The loop in the picture contains $j - i$ transitions

  Consider the automaton on input $a^{n+1+(j-i)}b^{n+1}$

  This time it goes around the loop one time more, but still arrives to the same state s_{n+1} with the same input

  Thus the automaton accepts.

Kleene Theorem

Theorem

A language is recognizable by a finite automaton if and only if it can be represented by a regular expression

Homework

Exercises from the Book:

No. 1, 3, 5 (page 324 – 325)