

# **Autmaton Minimization**

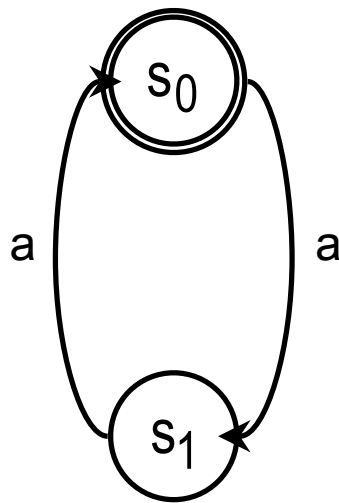
Discrete Mathematics  
Evgeny Skvortsov

## Previous Lecture

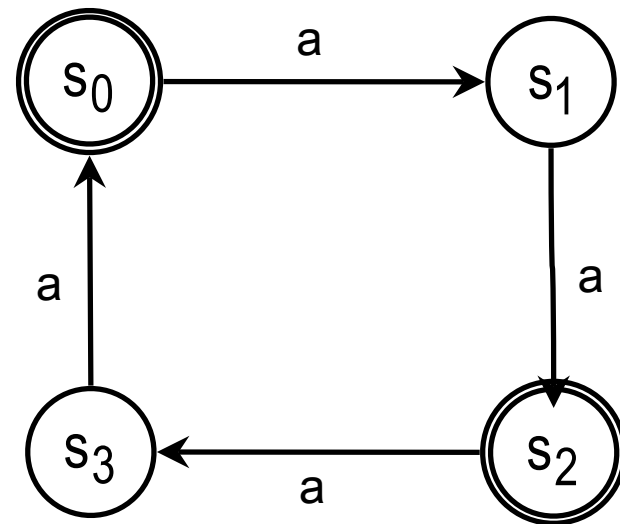
- Transducers
- Language not recognizable by finite automata
- Kleene Theorem

## Equivalent Automata

- Two automata are said to be **equivalent** if they accept the same language
- Example



$$L = (aa)^*$$



$$L = (aa)^*$$

## Minimal Automaton

- A **minimal** finite automaton is an automaton that has the least number of states among automata accepting the same language
- Given a finite automaton, how to construct a minimal automaton accepting the same language?
- The process is known as the **minimalization process**
- It works inductively, by identifying states

## Minimization Process

- Let  $M$  be an automaton with the set of internal states  $S$ , input alphabet  $\Sigma$ , transition function  $v$ , and the set of accepting states  $F$ . We assume that for any state  $s$  there is a string that takes initial state to  $s$ .
- On each step will construct a partition of  $S$  (or an equivalence relation  $E(k)$  on  $S$ )
- Basis Step: states  $s$  and  $t$  are 0-equivalent if they both accepting, or both not accepting. Partition with classes  $F$  and  $S - F$
- Inductive step: states  $s$  and  $t$  are  $k + 1$  – equivalent ( $(s, t) \in E_{k+1}$ ) if they are  $k$ -equivalent, and for any  $a \in \Sigma$  the states  $s'$  and  $t'$  are  $k$ -equivalent, where  $(s, a) \rightarrow s'$  and  $(t, a) \rightarrow t'$  are rules from the transition function
- The process stops when every pair of  $k$ -equivalent states is also  $k + 1$  – equivalent

## Minimization Process (cntd)

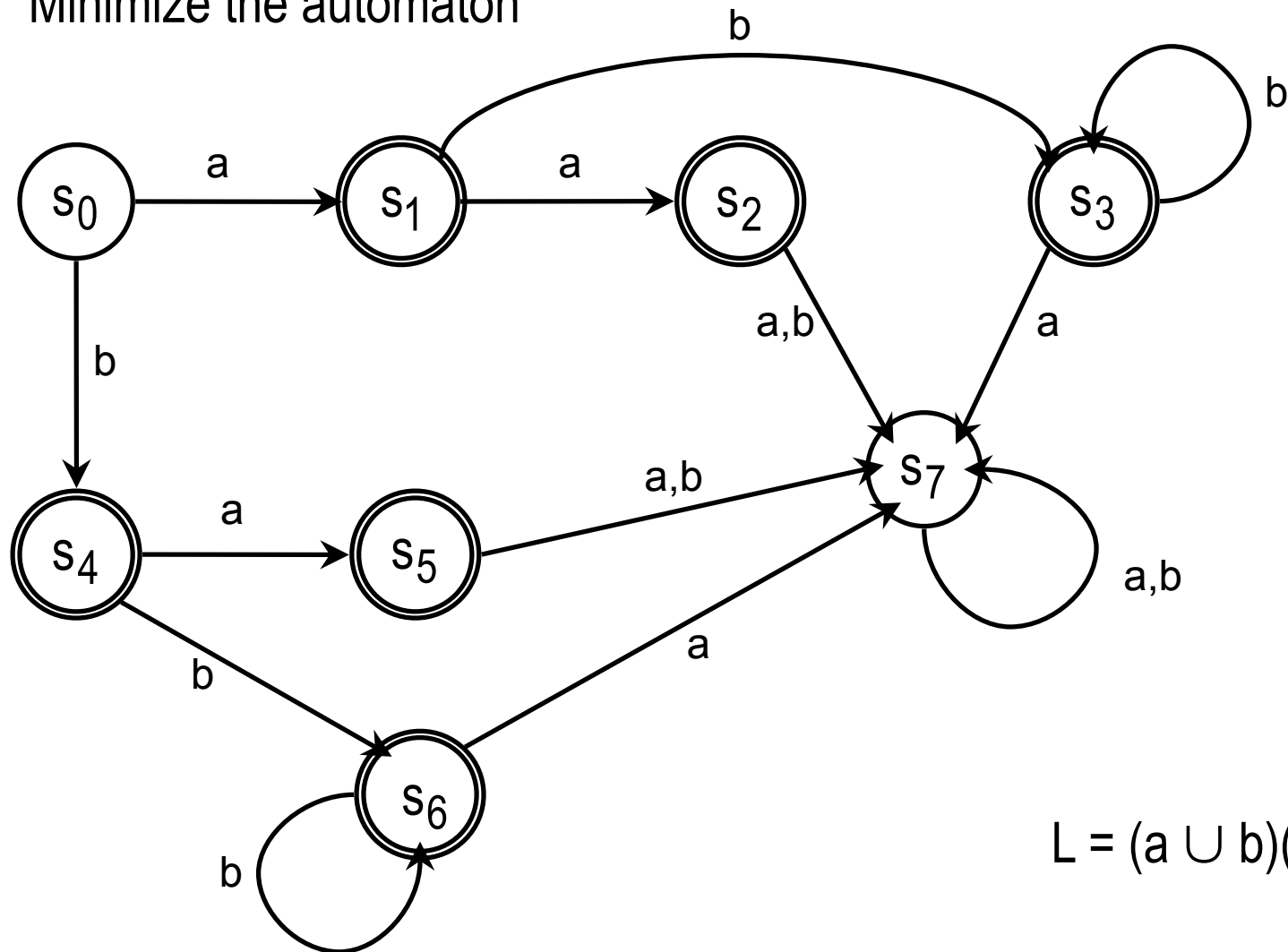
- Two states  $a$  and  $b$  are equivalent  $((a, b) \in E)$  if they are  $k$ -equivalent at the last step of the algorithm.
- Let  $S'$  be the set of classes of equivalence of  $E$ .
- If  $U$  is a class of equivalence of state  $u$ ,  $T$  is a class of equivalence of state  $t$  and  $v(u, a) = t$  then we define the transition function  $v'$  as  $v'(U, a) = T$ .
- The definition of the transition function is correct, since  $E = E_{k+1} = E_k$
- We set  $F'$  to be classes of equivalence of all  $s \in F$ .
- **Theorem:**
  - 1) The automaton  $(S', \Sigma, v', s'_0, F')$  is equivalent to  $(S, \Sigma, v, s_0, F)$ .
  - 2) The automaton  $(S', \Sigma, v', s'_0, F')$  is minimal.

## Ideas of the proof

- We have  $(u, v) \in E_k$  if and only if for any string  $w$ ,  $|w| \leq k$ , we have  $uw \in F$  if and only if  $v \in F$ .
- Consequently  $(u, v) \in E$  if and only if for any string  $w$  we have  $uw \in F$  if and only if  $v \in F$ .
- Since  $s_0'w$  is the class of equivalence of  $s_0w$  we conclude that automata are equivalent.
- Let  $w(u)$  be a string such that  $s_0w = u$ . If  $u$  and  $v$  are not equivalent then for any automata recognizing the same language as  $(S', \Sigma, v', s_0', F')$  states  $s_0w(u)$  and  $s_0w(v)$  must be distinct.

## Example

- Minimize the automaton





## Homework

Minimize the automaton:

$$S = \{s_0, \dots, s_6\}, \Sigma = \{a, b\}, F = \{s_1, s_3, s_5, s_6\}$$

$$\begin{aligned} v = \{ & (s_0, a) \rightarrow s_1, (s_0, b) \rightarrow s_3, \\ & (s_1, a) \rightarrow s_2, (s_1, b) \rightarrow s_4, \\ & (s_2, a) \rightarrow s_5, (s_2, b) \rightarrow s_5, \\ & (s_3, a) \rightarrow s_4, (s_3, b) \rightarrow s_2, \\ & (s_4, a) \rightarrow s_5, (s_4, b) \rightarrow s_5, \\ & (s_5, a) \rightarrow s_6, (s_5, b) \rightarrow s_5, \\ & (s_6, a) \rightarrow s_6, (s_6, b) \rightarrow s_6 \} \end{aligned}$$