CS170 Computation Theory

Lecture 7

September 26, 2023

Megumi Ando

Tufts University | School of Engineering Computer Science

Review of Last Lecture

- Robustness of TMs
- Church-Turing Thesis

Today’s Topics

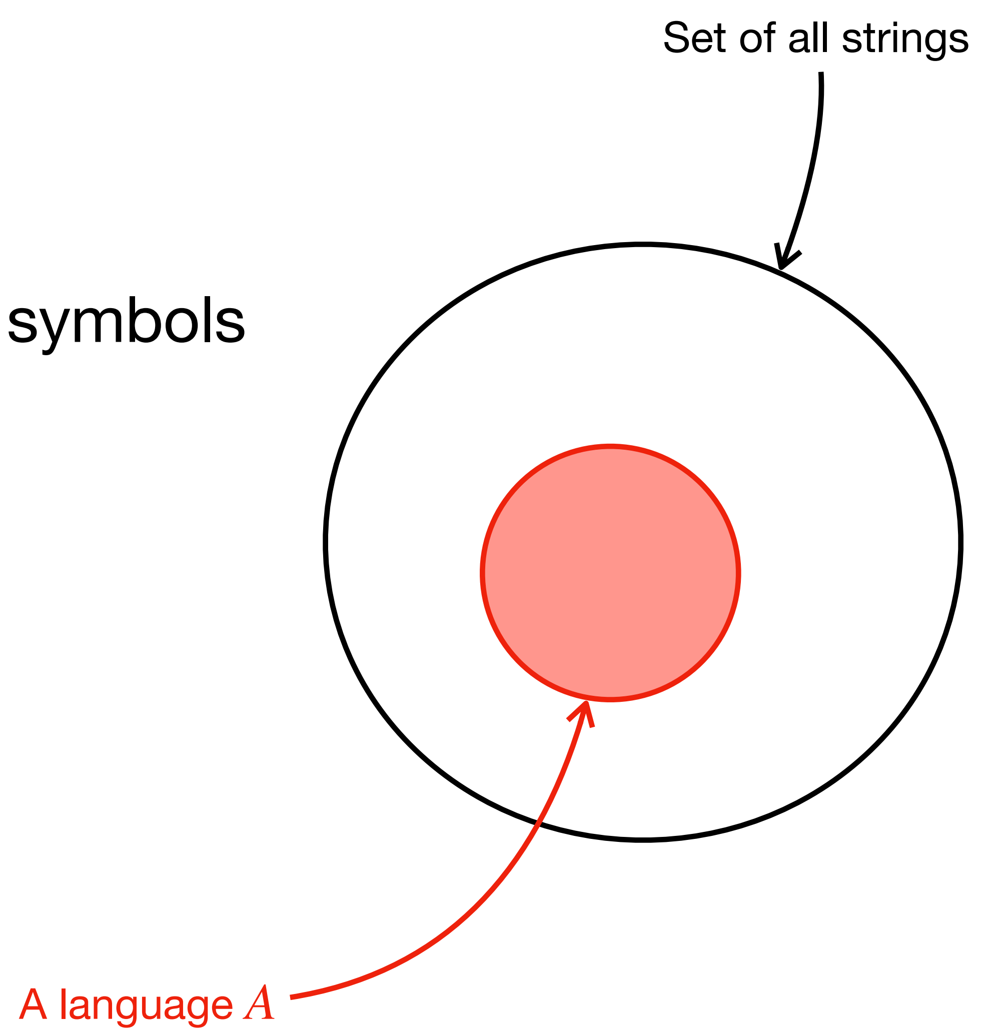
- ~~Robustness of TMs~~
- ~~Church-Turing Thesis~~

- Notation for Encodings and TMs
- Decision procedures for DFAs

Recall Definition of a Language

Definitions:

- A string is a finite sequence of symbols
- A language is a set of strings



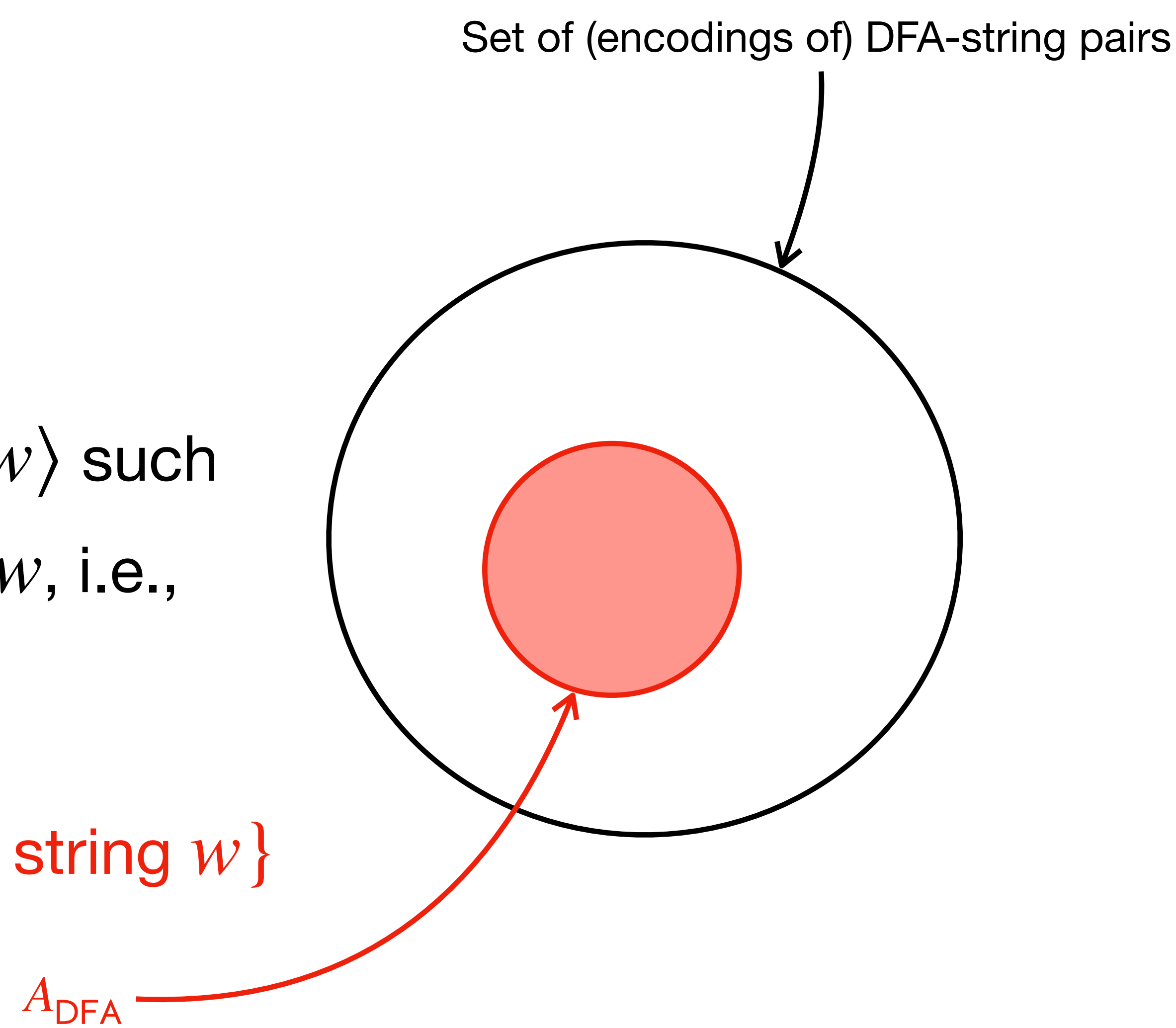
Language Can Represent a Computational Problem

E.g.,

Let B be a Deterministic Finite Automaton.

Let A_{DFA} be the set of pairs $\langle B, w \rangle$ such that w is a string and B accepts w, i.e.,

$$A_{\text{DFA}} = \{\langle B, w \rangle \mid B \text{ is a DFA that accepts input string } w\}$$



Language Can Represent a Computational Problem

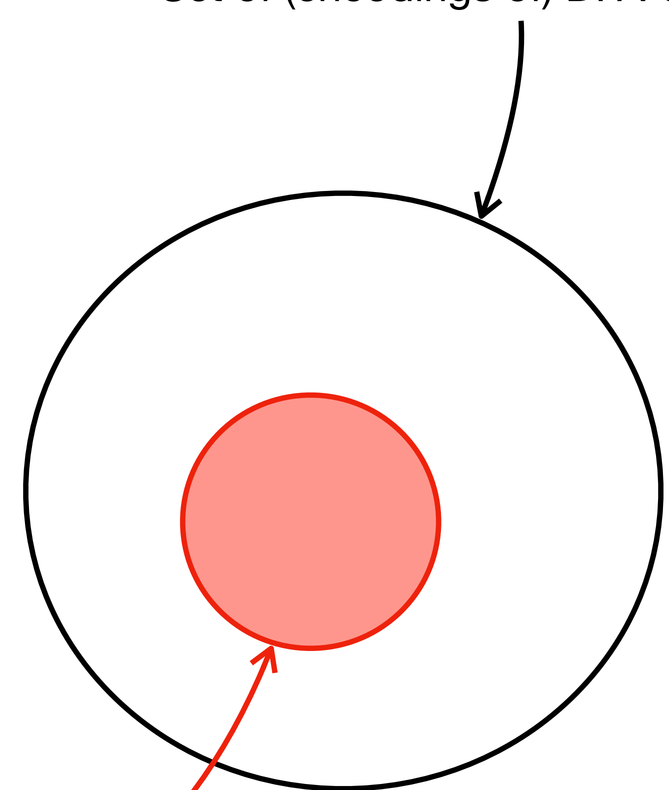
E.g.,

Let B be a Deterministic Finite Automaton.

Let A_{DFA} be the set of pairs $\langle B, w \rangle$ such that w is a string and B accepts w, i.e.,

$A_{\text{DFA}} = \{\langle B, w \rangle \mid$

B is a DFA that accepts input string $w\}$



Is there a TM that recognizes this language A_{DFA}?

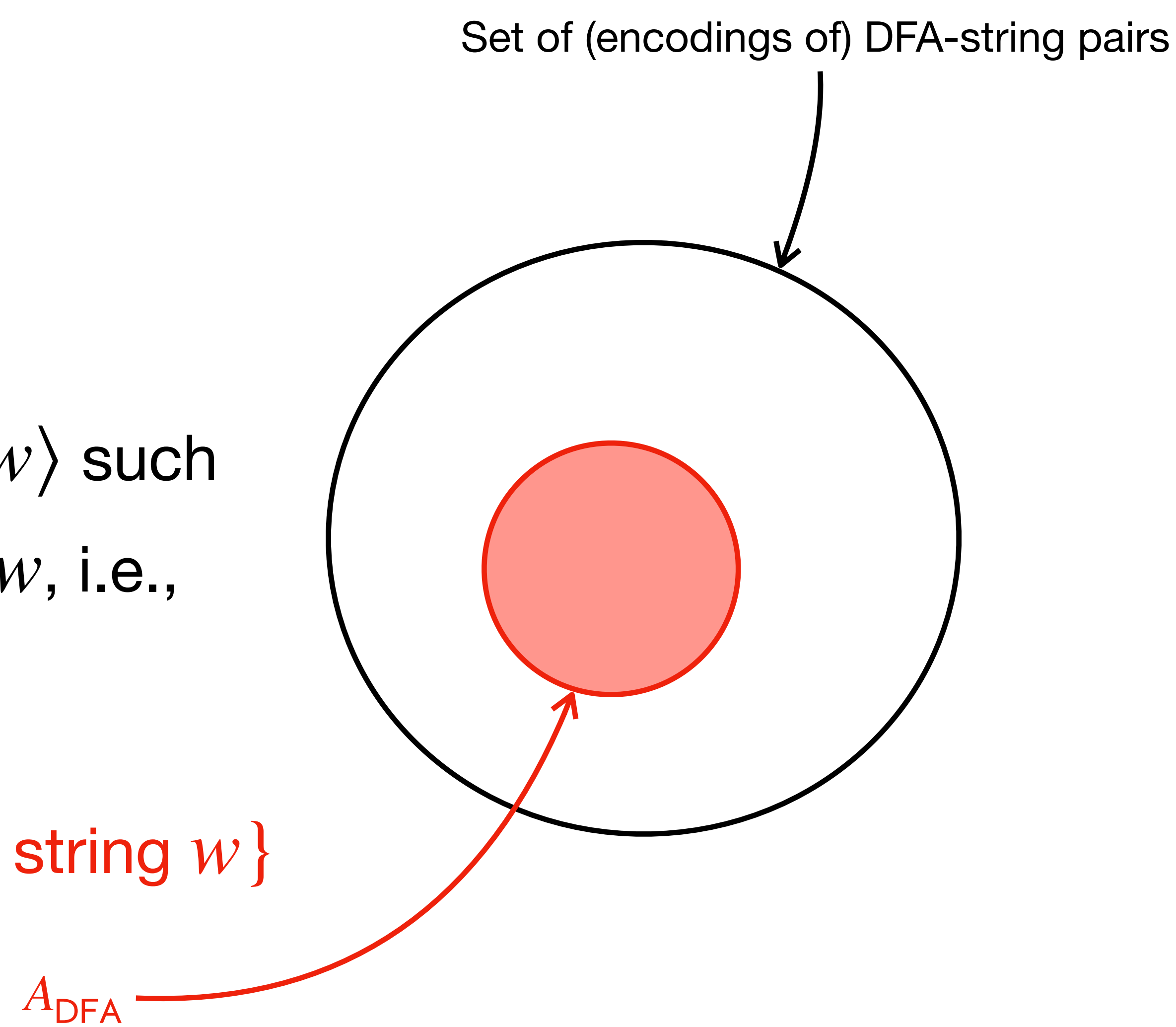
Language Can Represent a Computational Problem

E.g.,

Let B be a Deterministic Finite Automaton.

Let A_{DFA} be the set of pairs $\langle B, w \rangle$ such that w is a string and B accepts w, i.e.,

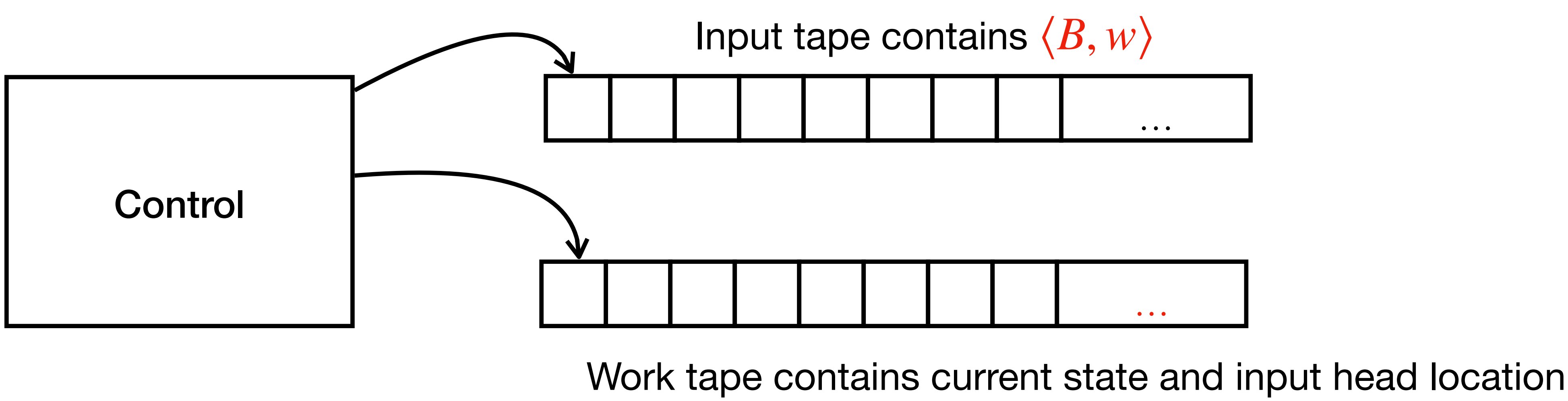
$$A_{\text{DFA}} = \{\langle B, w \rangle \mid B \text{ is a DFA that accepts input string } w\}$$



Is there a TM that recognizes this language A_{DFA}? that <u>decides</u> this language A_{DFA}?

Notation for Encodings

- Let $O_1, O_2, \ldots, O_k$ be “objects,” e.g., TMs, graphs, etc.
- We denote the encoding of these objects as $\langle O_1, O_2, \ldots, O_k \rangle$.
- In example in previous slide, $\langle B, w \rangle$ is the encoding of the pair, consisting of the DFA B and the string w.



Acceptance Problem for DFAs

Theorem (p.194): The language A_{DFA} is decidable.

Acceptance Problem for DFAs

Theorem (p.194): The language A_{DFA} is decidable.

Proof: Let $D_{A_{\text{DFA}}}$ be the following TM.

Acceptance Problem for DFAs

Theorem (p.194): The language A_{DFA} is decidable.

Proof: Let $D_{A_{\text{DFA}}}$ be the following TM.

$D_{A_{\text{DFA}}}$= "On input s,

Acceptance Problem for DFAs

Theorem (p.194): The language A_{DFA} is decidable.

Proof: Let $D_{A_{\text{DFA}}}$ be the following TM.

$D_{A_{\text{DFA}}}$= "On input s,

1. Check that s has correct form, i.e., $s = \langle B, w \rangle$ such that B is a DFA and w is a sequence of symbols from B's input alphabet; if not, *reject*.

Acceptance Problem for DFAs

Theorem (p.194): The language A_{DFA} is decidable.

Proof: Let $D_{A_{\text{DFA}}}$ be the following TM.

$D_{A_{\text{DFA}}}$ = "On input s,

1. Check that s has correct form, i.e., $s = \langle B, w \rangle$ such that B is a DFA and w is a sequence of symbols from B's input alphabet; if not, *reject*.
2. Simulate the computation of B on w.

Acceptance Problem for DFAs

Theorem (p.194): The language A_{DFA} is decidable.

Proof: Let $D_{A_{\text{DFA}}}$ be the following TM.

$D_{A_{\text{DFA}}}$ = "On input s,

1. Check that s has correct form, i.e., $s = \langle B, w \rangle$ such that B is a DFA and w is a sequence of symbols from B's input alphabet; if not, *reject*.
2. Simulate the computation of B on w.
3. If B ends in an accept state, *accept*. Otherwise, *reject."*

Acceptance Problem for DFAs

Theorem (p.194): The language A_{DFA} is decidable.

Proof: Let $D_{A_{\mathsf{DFA}}}$ be the following TM.

$D_{A_{\mathsf{DFA}}}$ = "On input s,

1. Check that s has correct form, i.e., $s = \langle B, w \rangle$ such that B is a DFA and w is a sequence of symbols from B's input alphabet; if not, *reject*.
2. Simulate the computation of B on w.
3. If B ends in an accept state, *accept*. Otherwise, *reject."*

$D_{A_{\mathsf{DFA}}}$ decides A_{DFA}.

Acceptance Problem for DFAs

Theorem (p.194): The language A_{DFA} is decidable.

Proof: Let $D_{A_{\text{DFA}}}$ be the following TM.

$D_{A_{\text{DFA}}}$ = "On input s,

Verbose; no need to include in future.

1. ~~Check that s has correct form, i.e., $s = \langle B, w \rangle$ such that B is a DFA and w is a sequence of symbols from B's input alphabet; if not, *reject*.~~
2. Simulate the computation of B on w.
3. If B ends in an accept state, *accept*. Otherwise, *reject.*"

$D_{A_{\text{DFA}}}$ decides A_{DFA}.

Acceptance Problem for DFAs

Theorem (p.194): The language A_{DFA} is decidable.

Proof: Let $D_{A_{\text{DFA}}}$ be the following TM.

$D_{A_{\text{DFA}}}$ = "On input s,

Verbose; no need to include in future.

1. ~~Check that s has correct form, i.e., $s = \langle B, w \rangle$ such that B is a DFA and w is a sequence of symbols from B's input alphabet; if not, *reject*.~~
2. Simulate the computation of B on w.
3. If B ends in an accept state, *accept*. Otherwise, *reject.*"

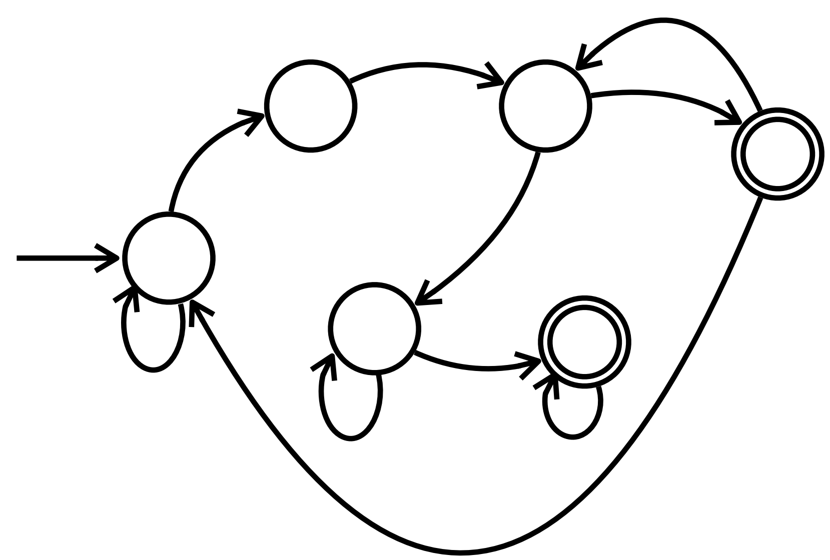
$D_{A_{\text{DFA}}}$ decides A_{DFA}.

Why does $D_{A_{\text{DFA}}}$ always halt?

$D_{A_{DFA}}$ Always Halts

- Let $r_0 = q_{\text{start}}$ (in the DFA).

- The simulation (step 2) always completes in $|w|$ sub-steps because each transition “consumes” an input symbol w_i and takes us to the next state r_{i+1}.

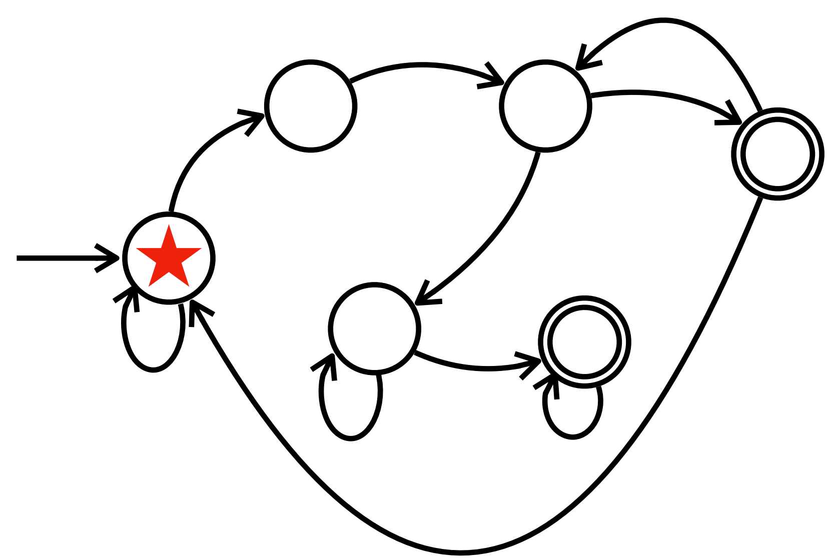


$D_{A_{DFA}}$ Always Halts

- Let $r_0 = q_{\mathsf{start}}$ (in the DFA).

- The simulation (step 2) always completes in $|w|$ sub-steps because each transition "consumes" an input symbol w_i and takes us to the next state r_{i+1}.

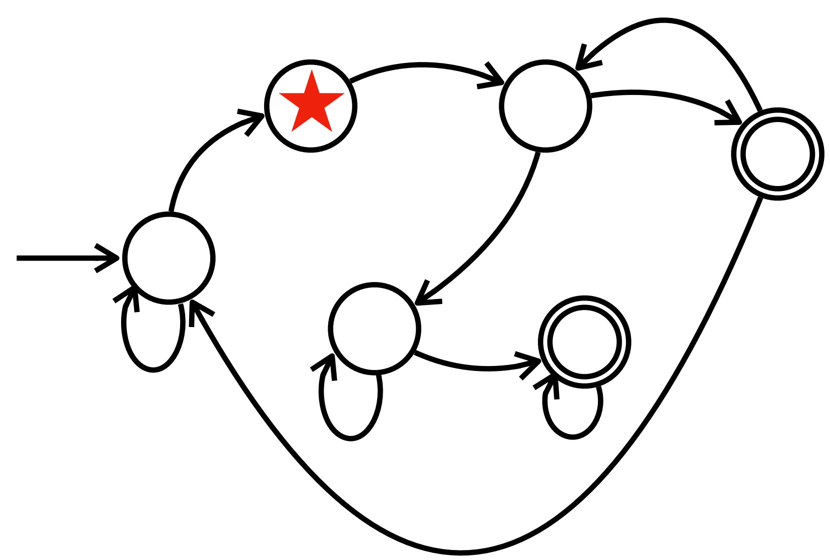
Input: w_1, w_2, w_3, w_k



$D_{A_{DFA}}$ Always Halts

- Let $r_0 = q_{\text{start}}$ (in the DFA).
- The simulation (step 2) always completes in $|w|$ sub-steps because each transition "consumes" an input symbol w_i and takes us to the next state r_{i+1}.

Input: $\cancel{w}_1, w_2, w_3, w_k$

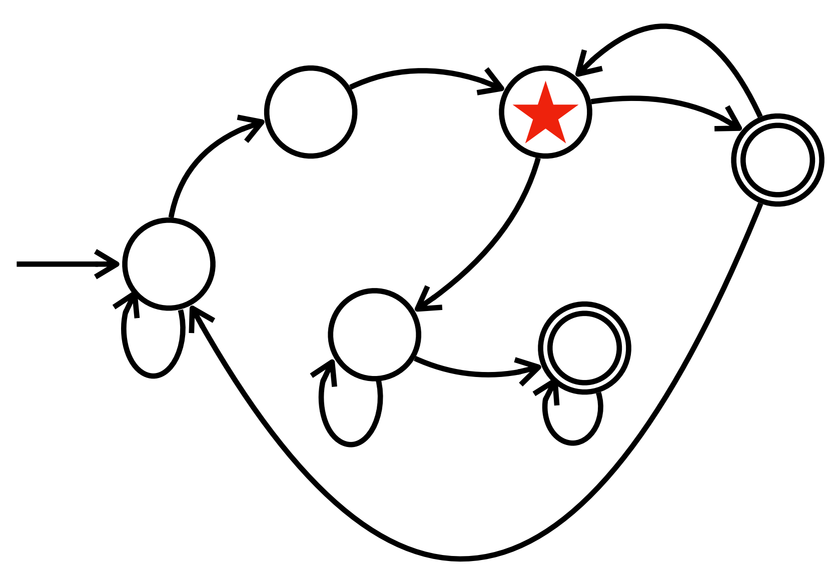


$D_{A_{DFA}}$ Always Halts

- Let $r_0 = q_{\mathsf{start}}$ (in the DFA).

- The simulation (step 2) always completes in $|w|$ sub-steps because each transition "consumes" an input symbol w_i and takes us to the next state r_{i+1}.

Input: $\cancel{w_1}, \cancel{w_2}, w_3, w_k$

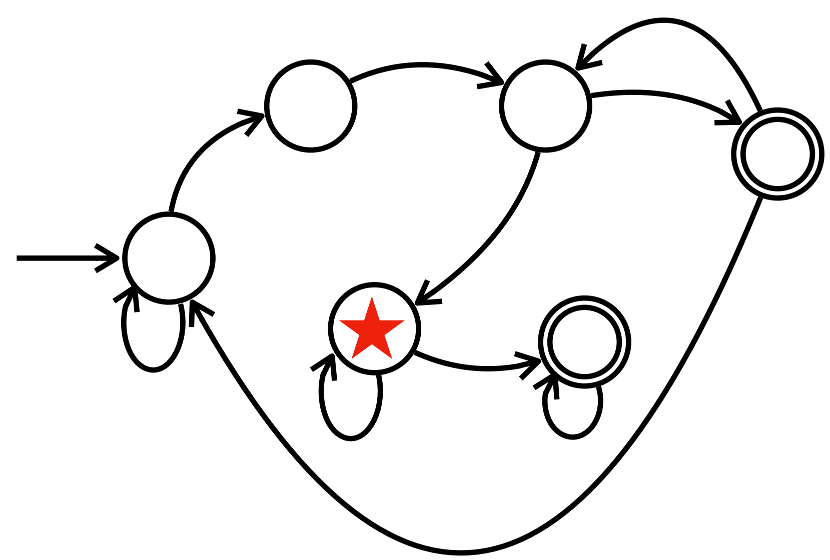


$D_{A_{DFA}}$ Always Halts

- Let $r_0 = q_{\mathsf{start}}$ (in the DFA).

- The simulation (step 2) always completes in $|w|$ sub-steps because each transition "consumes" an input symbol w_i and takes us to the next state r_{i+1}.

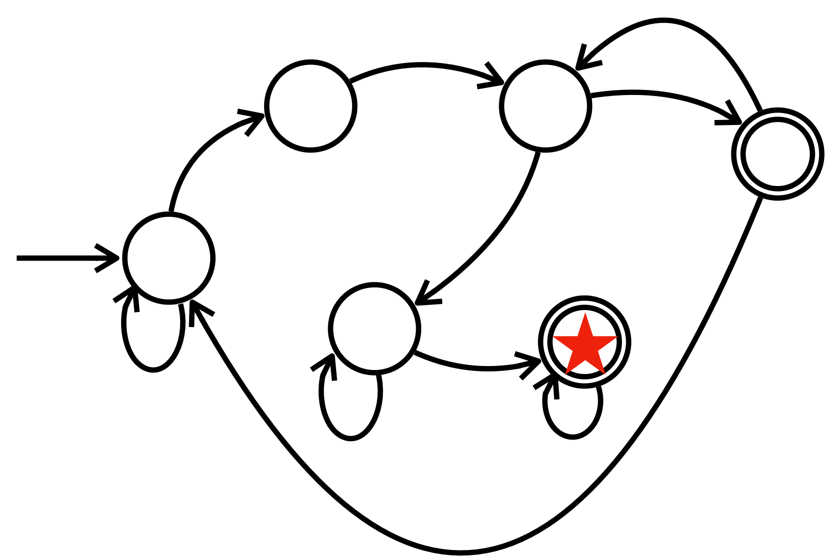
Input: $\cancel{w_1}, \cancel{w_2}, \cancel{w_3}, w_k$



$D_{A_{DFA}}$ Always Halts

- Let $r_0 = q_{\text{start}}$ (in the DFA).
- The simulation (step 2) always completes in $|w|$ sub-steps because each transition "consumes" an input symbol w_i and takes us to the next state r_{i+1}.

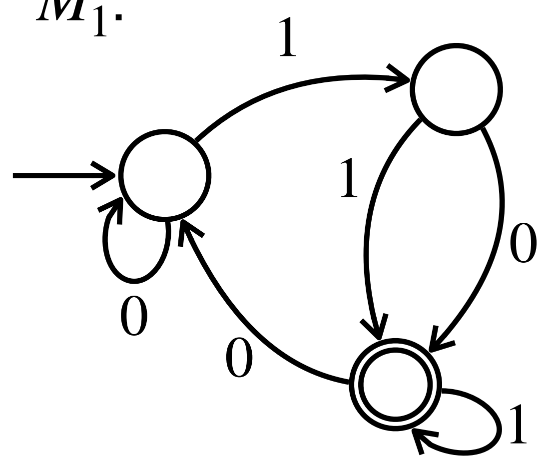
Input: $\cancel{w}_1, \cancel{w}_2, \cancel{w}_3, \cancel{w}_k$



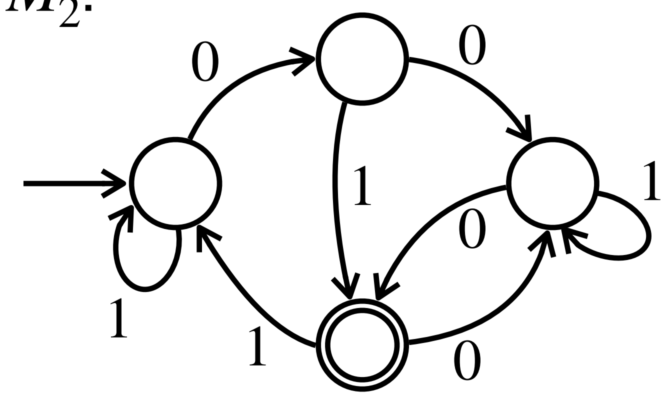
Check-In 1 (Break)

Consider the following DFAs.

M_1:



M_2:



1. Is $\langle M_1 \rangle$ in A_{DFA}?
2. Is $\langle 0100 \rangle$ in A_{DFA}?
3. Is $\langle M_1, 0101 \rangle$ in A_{DFA}?
4. Is $\langle M_2, 0101 \rangle$ in A_{DFA}?
5. Is $\langle M_2, 0011111111111110010 \rangle$ in A_{DFA}?
6. Is $\langle M_2, 00111111111111110011111111101 \rangle$ in A_{DFA}?

Acceptance Problem for NFAs

Theorem (p.195): The language
$A_{\text{NFA}} = \{\langle B, w\rangle \mid B \text{ is a NFA that accepts input string } w\}$ is decidable.

Proof Attempt 1: Let $D'_{A_{\text{NFA}}}$ be the following TM.

$D'_{A_{\text{NFA}}}$= "On input $\langle B, w\rangle$ where B is a NFA and w is a string,

1. Simulate the computation of B on w.
2. If B ends in an accept state, *accept*. Otherwise, *reject.*"

$D'_{A_{\text{NFA}}}$ decides A_{NFA}.

Acceptance Problem for NFAs

Theorem (p.195): The language
$A_{\text{NFA}} = \{\langle B, w\rangle \,|\, B$ is a NFA that accepts input string $w\}$ is decidable.

Proof Attempt 1: Let $D'_{A_{\text{NFA}}}$ be the following TM.

$D'_{A_{\text{NFA}}}$ = "On input $\langle B, w\rangle$ where B is a NFA and w is a string,

1. Simulate the computation of B on w.
2. If B ends in an accept state, *accept*. Otherwise, *reject.*"

$D'_{A_{\text{NFA}}}$ decides A_{NFA}.

(ϵ-transitions make argument tricky.)

Acceptance Problem for NFAs

Theorem (p.195): The language $A_{\text{NFA}} = \{\langle B, w\rangle \mid B$ is a NFA that accepts input string $w\}$ is decidable.

Acceptance Problem for NFAs

Theorem (p.195): The language
$A_{\text{NFA}} = \{\langle B, w\rangle \mid B$ is a NFA that accepts input string $w\}$ is decidable.

Proof: Let $D_{A_{\text{NFA}}}$ be the following TM.

Acceptance Problem for NFAs

Theorem (p.195): The language
$A_{\text{NFA}} = \{\langle B, w\rangle \mid B$ is a NFA that accepts input string $w\}$ is decidable.

Proof: Let $D_{A_{\text{NFA}}}$ be the following TM.

$D_{A_{\text{NFA}}}$= "On input $\langle B, w\rangle$ where B is a NFA and w is a string,

Acceptance Problem for NFAs

Theorem (p.195): The language
$A_{\text{NFA}} = \{\langle B, w\rangle \mid B$ is a NFA that accepts input string $w\}$ is decidable.

Proof: Let $D_{A_{\text{NFA}}}$ be the following TM.

$D_{A_{\text{NFA}}}$= "On input $\langle B, w\rangle$ where B is a NFA and w is a string,

1. Convert B to an equivalent DFA, B', using the procedure from Lecture 2.

Acceptance Problem for NFAs

Theorem (p.195): The language
$A_{\text{NFA}} = \{\langle B, w\rangle \mid B$ is a NFA that accepts input string $w\}$ is decidable.

Proof: Let $D_{A_{\text{NFA}}}$ be the following TM.

$D_{A_{\text{NFA}}}$= "On input $\langle B, w\rangle$ where B is a NFA and w is a string,

1. Convert B to an equivalent DFA, B', using the procedure from Lecture 2.
2. Run $D_{A_{\text{DFA}}}$ (from previous slide) on $\langle B', w\rangle$.

Acceptance Problem for NFAs

Theorem (p.195): The language
$A_{\text{NFA}} = \{\langle B, w \rangle \mid B$ is a NFA that accepts input string $w\}$ is decidable.

Proof: Let $D_{A_{\text{NFA}}}$ be the following TM.

$D_{A_{\text{NFA}}}$= "On input $\langle B, w \rangle$ where B is a NFA and w is a string,

1. Convert B to an equivalent DFA, B', using the procedure from Lecture 2.
2. Run $D_{A_{\text{DFA}}}$ (from previous slide) on $\langle B', w \rangle$.
3. If $D_{A_{\text{DFA}}}$ accepts, accept. Else, reject."

Acceptance Problem for NFAs

Theorem (p.195): The language $A_{\text{NFA}} = \{\langle B, w\rangle \mid B$ is a NFA that accepts input string $w\}$ is decidable.

Proof: Let $D_{A_{\text{NFA}}}$ be the following TM.

$D_{A_{\text{NFA}}}$= "On input $\langle B, w\rangle$ where B is a NFA and w is a string,

1. Convert B to an equivalent DFA, B', using the procedure from Lecture 2.
2. Run $D_{A_{\text{DFA}}}$ (from previous slide) on $\langle B', w\rangle$.
3. If $D_{A_{\text{DFA}}}$ accepts, accept. Else, reject."

$D_{A_{\text{NFA}}}$ decides A_{NFA}.

Acceptance Problem for NFAs

Theorem (p.195): The language

$A_{\text{NFA}} = \{\langle B, w\rangle \mid B$ is a NFA that accepts input string $w\}$ is decidable.

Proof: Let $D_{A_{\text{NFA}}}$ be the following TM.

$D_{A_{\text{NFA}}}$= "On input $\langle B, w\rangle$ where B is a NFA and w is a string,

1. Convert B to an equivalent DFA, B', using the procedure from Lecture 2.
2. Run $D_{A_{\text{DFA}}}$ (from previous slide) on $\langle B', w\rangle$.
3. If $D_{A_{\text{DFA}}}$ accepts, accept. Else, reject."

$D_{A_{\text{NFA}}}$ decides A_{NFA}.

Why does $D_{A_{\text{NFA}}}$ always halt?

Acceptance Problem for NFAs

Theorem (p.195): The language
$A_{\text{NFA}} = \{\langle B, w\rangle \mid B \text{ is a NFA that accepts input string } w\}$ is decidable.

Proof: Let $D_{A_{\text{NFA}}}$ be the following TM.

$D_{A_{\text{NFA}}}$= "On input $\langle B, w\rangle$ where B is a NFA and w is a string,

1. Convert B to an equivalent DFA, B', using the procedure from Lecture 2.
2. Run $D_{A_{\text{DFA}}}$ (from previous slide) on $\langle B', w\rangle$.
3. If $D_{A_{\text{DFA}}}$ accepts, accept. Else, reject."

$D_{A_{\text{NFA}}}$ decides A_{NFA}.

Why does $D_{A_{\text{NFA}}}$ always halt?

(Because $D_{A_{\text{DFA}}}$ does.)

Q.E.D.

Acceptance Problem for Regular Expressions

Theorem (p.196): The language
$A_{\text{REG}} = \{\langle R, w \rangle \mid R$ is a regular expression that generates string $w\}$ is decidable.

Acceptance Problem for Regular Expressions

Theorem (p.196): The language
$A_{\text{REG}} = \{\langle R, w\rangle \mid R$ is a regular expression that generates string $w\}$ is decidable.

Proof: Let $D_{A_{\text{REG}}}$ be the following TM.

Acceptance Problem for Regular Expressions

Theorem (p.196): The language
$A_{\text{REG}} = \{\langle R, w\rangle \,|\, R$ is a regular expression that generates string $w\}$ is decidable.

Proof: Let $D_{A_{\text{REG}}}$ be the following TM.

$D_{A_{\text{REG}}}$= "On input $\langle R, w\rangle$ where R is a regular expression and w is a string,

Acceptance Problem for Regular Expressions

Theorem (p.196): The language
$A_{\text{REG}} = \{\langle R, w\rangle \mid R$ is a regular expression that generates string $w\}$ is decidable.

Proof: Let $D_{A_{\text{REG}}}$ be the following TM.

$D_{A_{\text{REG}}}$= "On input $\langle R, w\rangle$ where R is a regular expression and w is a string,

1. Convert R to an equivalent NFA, R', using the procedure from Lecture 2 (closure properties).

Acceptance Problem for Regular Expressions

Theorem (p.196): The language
$A_{\text{REG}} = \{\langle R, w\rangle \mid R$ is a regular expression that generates string $w\}$ is decidable.

Proof: Let $D_{A_{\text{REG}}}$ be the following TM.

$D_{A_{\text{REG}}}$= "On input $\langle R, w\rangle$ where R is a regular expression and w is a string,

1. Convert R to an equivalent NFA, R', using the procedure from Lecture 2 (closure properties).
2. Run $D_{A_{\text{NFA}}}$ (from previous slide) on $\langle R', w\rangle$.

Acceptance Problem for Regular Expressions

Theorem (p.196): The language
$A_{\text{REG}} = \{\langle R, w \rangle \mid R$ is a regular expression that generates string $w\}$ is decidable.

Proof: Let $D_{A_{\text{REG}}}$ be the following TM.

$D_{A_{\text{REG}}}$= "On input $\langle R, w \rangle$ where R is a regular expression and w is a string,

1. Convert R to an equivalent NFA, R', using the procedure from Lecture 2 (closure properties).
2. Run $D_{A_{\text{NFA}}}$ (from previous slide) on $\langle R', w \rangle$.
3. If $D_{A_{\text{NFA}}}$ accepts, accept. Else, reject."

Acceptance Problem for Regular Expressions

Theorem (p.196): The language
$A_{\text{REG}} = \{\langle R, w \rangle \mid R$ is a regular expression that generates string $w\}$ is decidable.

Proof: Let $D_{A_{\text{REG}}}$ be the following TM.

$D_{A_{\text{REG}}}$= "On input $\langle R, w \rangle$ where R is a regular expression and w is a string,

1. Convert R to an equivalent NFA, R', using the procedure from Lecture 2 (closure properties).
2. Run $D_{A_{\text{NFA}}}$ (from previous slide) on $\langle R', w \rangle$.
3. If $D_{A_{\text{NFA}}}$ accepts, accept. Else, reject."

$D_{A_{\text{REG}}}$ decides A_{REG}.

Acceptance Problem for Regular Expressions

Theorem (p.196): The language
$A_{\text{REG}} = \{\langle R, w\rangle \mid R$ is a regular expression that generates string $w\}$ is decidable.

Proof: Let $D_{A_{\text{REG}}}$ be the following TM.

$D_{A_{\text{REG}}}$= "On input $\langle R, w\rangle$ where R is a regular expression and w is a string,

1. Convert R to an equivalent NFA, R', using the procedure from Lecture 2 (closure properties).
2. Run $D_{A_{\text{NFA}}}$ (from previous slide) on $\langle R', w\rangle$.
3. If $D_{A_{\text{NFA}}}$ accepts, accept. Else, reject."

$D_{A_{\text{REG}}}$ decides A_{REG}.

$D_{A_{\text{REG}}}$ always halts because $D_{A_{\text{NFA}}}$ does.

Q.E.D.

Emptiness Problem for DFAs

Theorem (p.196): The language $E_{\mathsf{DFA}} = \{\langle B \rangle \mid B \text{ is a DFA and } L(B) = \emptyset\}$ is decidable.

Emptiness Problem for DFAs

Theorem (p.196): The language $E_{\text{DFA}} = \{\langle B \rangle \mid B \text{ is a DFA and } L(B) = \emptyset\}$ is decidable.

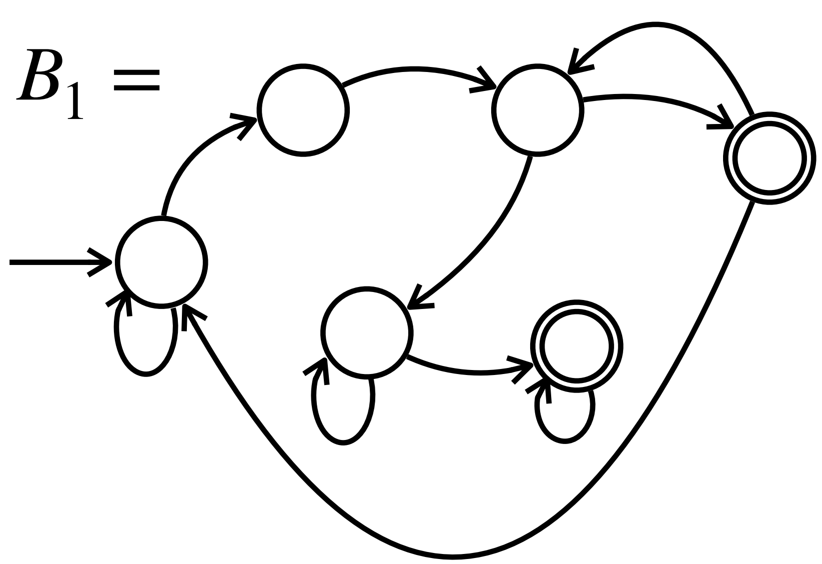
Let's see some examples to build intuition.

Emptiness Problem for DFAs

Theorem (p.196): The language $E_{\text{DFA}} = \{\langle B \rangle \mid B \text{ is a DFA and } L(B) = \emptyset\}$ is decidable.

Let's see some examples to build intuition.

Let $B_1 =$

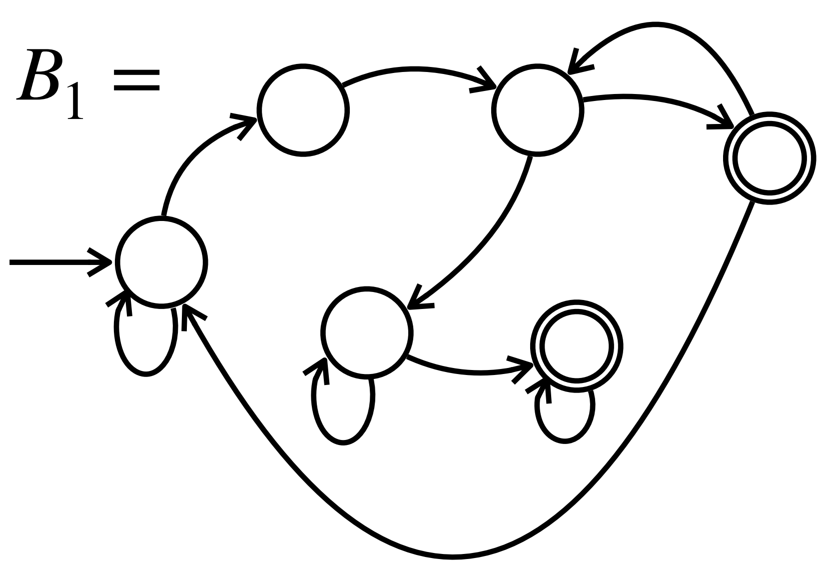


Emptiness Problem for DFAs

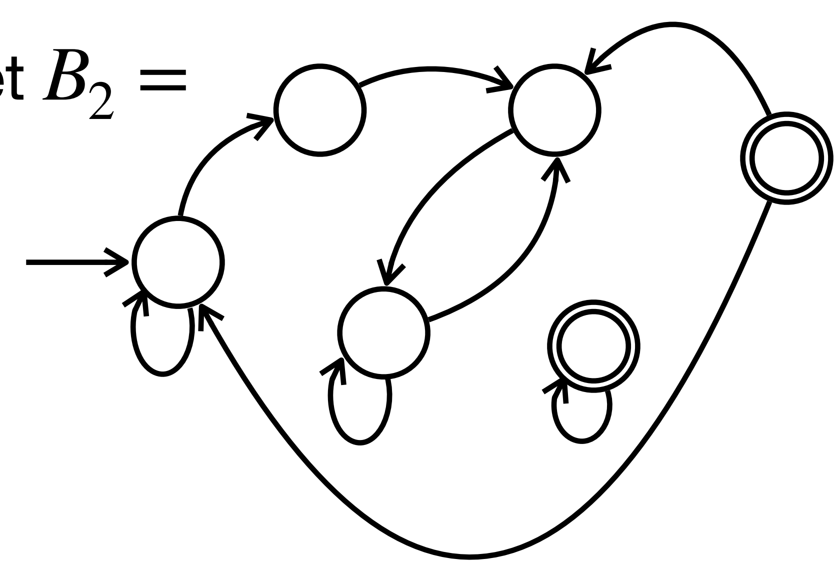
Theorem (p.196): The language $E_{\text{DFA}} = \{\langle B \rangle \mid B \text{ is a DFA and } L(B) = \emptyset\}$ is decidable.

Let's see some examples to build intuition.

Let $B_1 =$



Let $B_2 =$

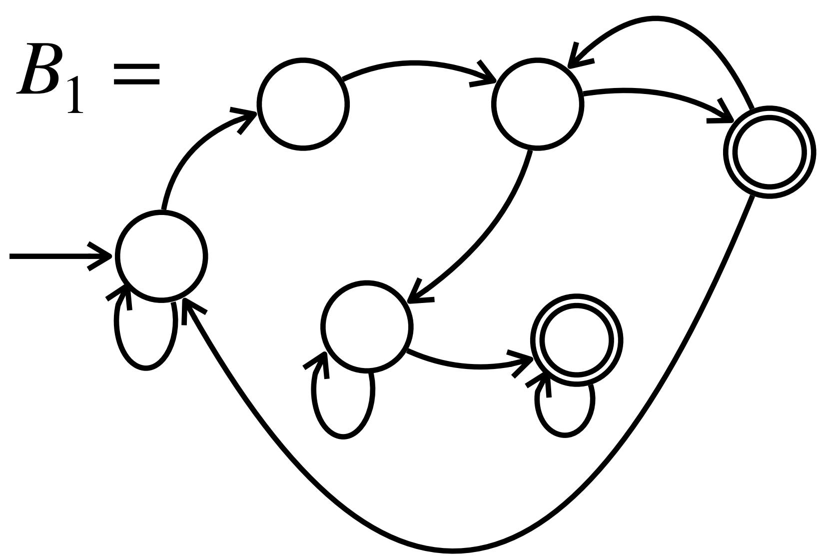


Emptiness Problem for DFAs

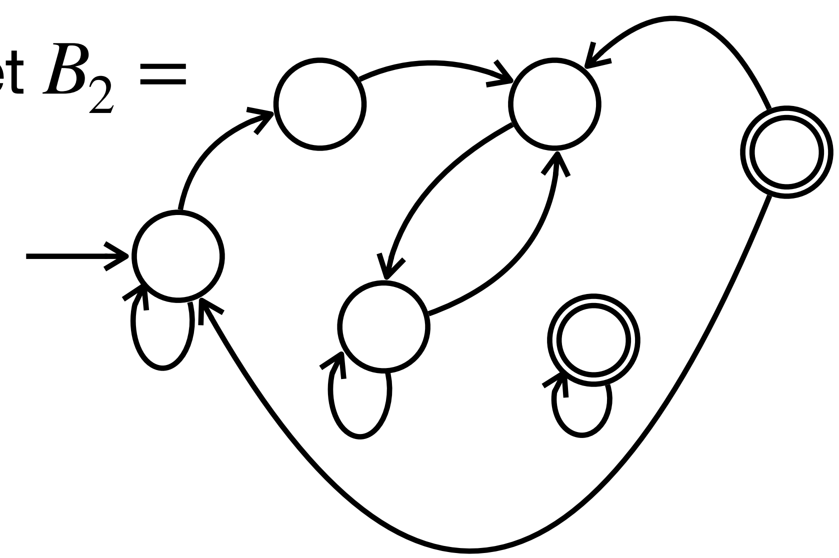
Theorem (p.196): The language $E_{\text{DFA}} = \{\langle B \rangle \mid B \text{ is a DFA and } L(B) = \emptyset\}$ is decidable.

Let's see some examples to build intuition.

Let $B_1 =$



Let $B_2 =$



Is $B_1 \in E_{\text{DFA}}$? Is $B_2 \in E_{\text{DFA}}$?

Emptiness Problem for DFAs

Theorem (p.196): The language $E_{\text{DFA}} = \{\langle B \rangle \mid B \text{ is a DFA and } L(B) = \emptyset\}$ is decidable.

Proof: Let $D_{E_{\text{DFA}}}$ be the following TM that tests if there is a path from the start state to an accept state.

Emptiness Problem for DFAs

Theorem (p.196): The language $E_{\text{DFA}} = \{\langle B \rangle \mid B \text{ is a DFA and } L(B) = \emptyset\}$ is decidable.

Proof: Let $D_{E_{\text{DFA}}}$ be the following TM that tests if there is a path from the start state to an accept state.

Emptiness Problem for DFAs

Theorem (p.196): The language $E_{\text{DFA}} = \{\langle B \rangle \mid B \text{ is a DFA and } L(B) = \emptyset\}$ is decidable.

Proof: Let $D_{E_{\text{DFA}}}$ be the following TM that tests if there is a path from the start state to an accept state.

$D_{E_{\text{DFA}}}$ = "On input $\langle B \rangle$ where B is a DFA,

Emptiness Problem for DFAs

Theorem (p.196): The language $E_{\text{DFA}} = \{\langle B \rangle \mid B \text{ is a DFA and } L(B) = \emptyset\}$ is decidable.

Proof: Let $D_{E_{\text{DFA}}}$ be the following TM that tests if there is a path from the start state to an accept state.

$D_{E_{\text{DFA}}}$= "On input $\langle B \rangle$ where B is a DFA,

1. Mark the start state of B.

Emptiness Problem for DFAs

Theorem (p.196): The language $E_{\text{DFA}} = \{\langle B \rangle \mid B \text{ is a DFA and } L(B) = \emptyset\}$ is decidable.

Proof: Let $D_{E_{\text{DFA}}}$ be the following TM that tests if there is a path from the start state to an accept state.

$D_{E_{\text{DFA}}}$ = "On input $\langle B \rangle$ where B is a DFA,

1. Mark the start state of B.
2. Repeat until no new states are marked: mark any unmarked state with a transition from a marked state.

Emptiness Problem for DFAs

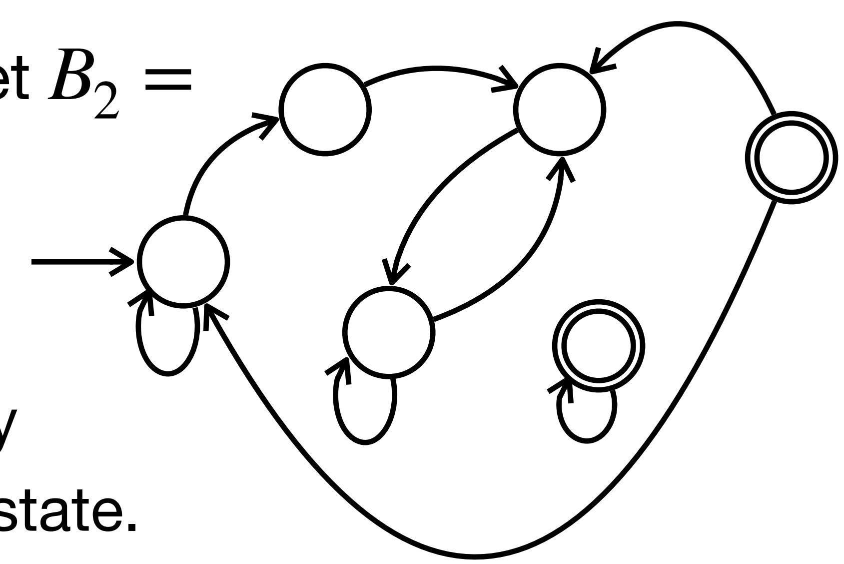
Theorem (p.196): The language $E_{\text{DFA}} = \{\langle B \rangle \mid B \text{ is a DFA and } L(B) = \emptyset\}$ is decidable.

Proof: Let $D_{E_{\text{DFA}}}$ be the following TM that tests if there is a path from the start state to an accept state.

$D_{E_{\text{DFA}}}$ = "On input $\langle B \rangle$ where B is a DFA,

1. Mark the start state of B.
2. Repeat until no new states are marked: mark any unmarked state with a transition from a marked state.

Let $B_2 =$

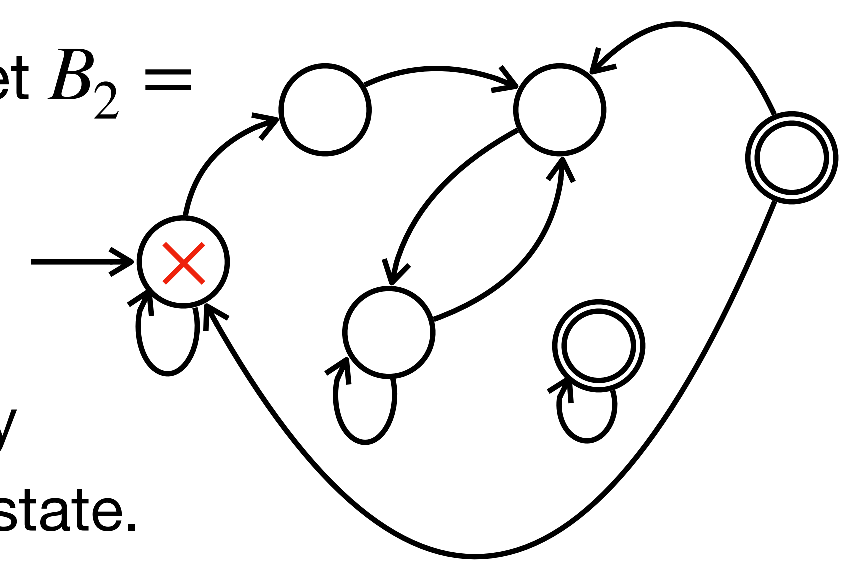


Emptiness Problem for DFAs

Theorem (p.196): The language $E_{\text{DFA}} = \{\langle B \rangle \mid B \text{ is a DFA and } L(B) = \varnothing\}$ is decidable.

Proof: Let $D_{E_{\text{DFA}}}$ be the following TM that tests if there is a path from the start state to an accept state.

Let $B_2 =$



$D_{E_{\text{DFA}}}$= "On input $\langle B \rangle$ where B is a DFA,

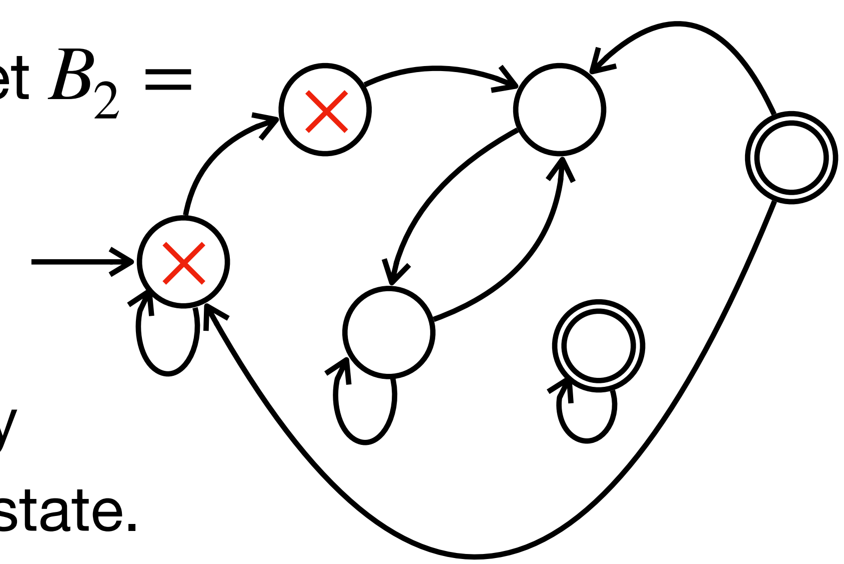
1. Mark the start state of B.
2. Repeat until no new states are marked: mark any unmarked state with a transition from a marked state.

Emptiness Problem for DFAs

Theorem (p.196): The language $E_{\text{DFA}} = \{\langle B \rangle \mid B \text{ is a DFA and } L(B) = \emptyset\}$ is decidable.

Proof: Let $D_{E_{\text{DFA}}}$ be the following TM that tests if there is a path from the start state to an accept state.

Let $B_2 =$



$D_{E_{\text{DFA}}}$ = "On input $\langle B \rangle$ where B is a DFA,

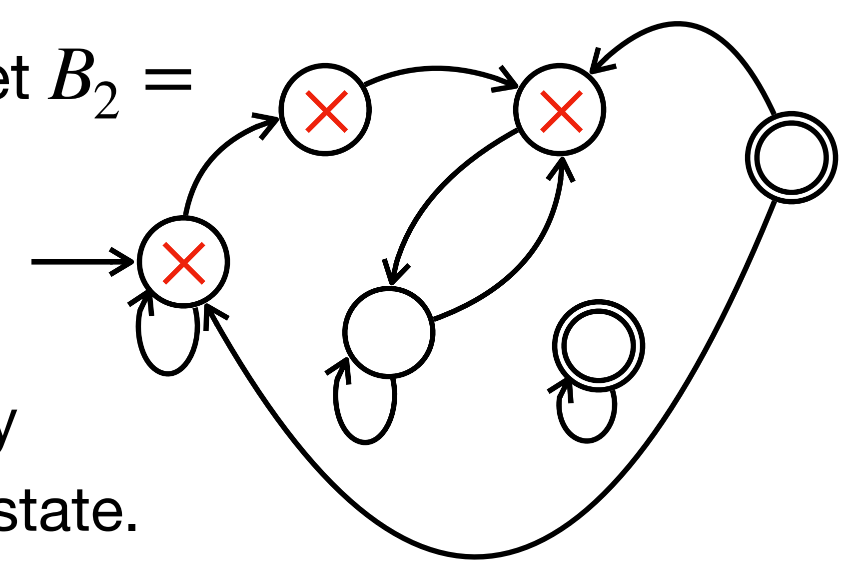
1. Mark the start state of B.
2. Repeat until no new states are marked: mark any unmarked state with a transition from a marked state.

Emptiness Problem for DFAs

Theorem (p.196): The language $E_{\text{DFA}} = \{\langle B \rangle \mid B \text{ is a DFA and } L(B) = \emptyset\}$ is decidable.

Proof: Let $D_{E_{\text{DFA}}}$ be the following TM that tests if there is a path from the start state to an accept state.

Let $B_2 =$



$D_{E_{\text{DFA}}}$ = "On input $\langle B \rangle$ where B is a DFA,

1. Mark the start state of B.
2. Repeat until no new states are marked: mark any unmarked state with a transition from a marked state.

Emptiness Problem for DFAs

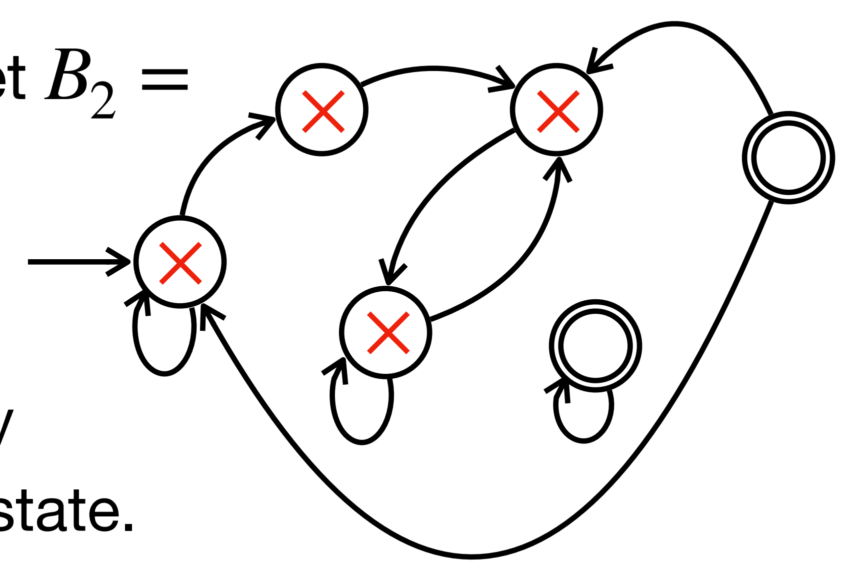
Theorem (p.196): The language $E_{\text{DFA}} = \{\langle B \rangle \mid B \text{ is a DFA and } L(B) = \emptyset\}$ is decidable.

Proof: Let $D_{E_{\text{DFA}}}$ be the following TM that tests if there is a path from the start state to an accept state.

$D_{E_{\text{DFA}}}$ = "On input $\langle B \rangle$ where B is a DFA,

1. Mark the start state of B.
2. Repeat until no new states are marked: mark any unmarked state with a transition from a marked state.

Let $B_2 =$

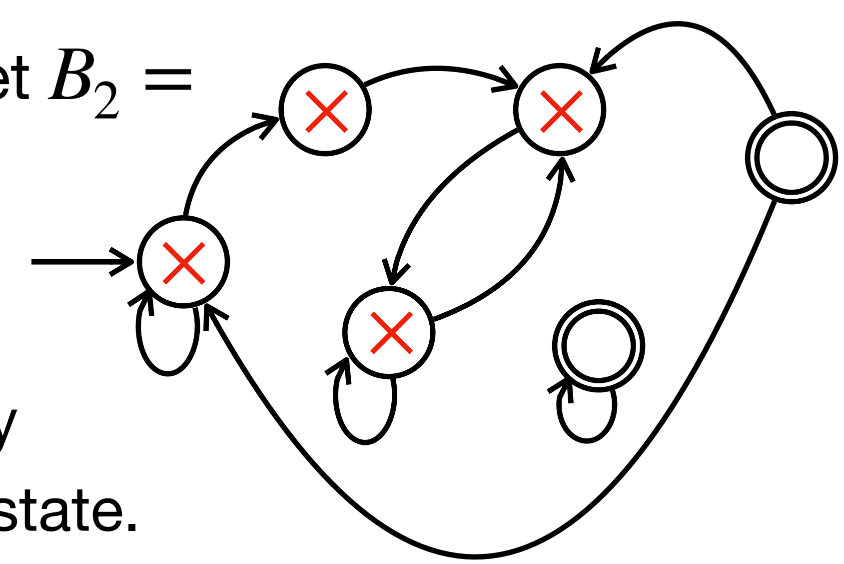


Emptiness Problem for DFAs

Theorem (p.196): The language $E_{\text{DFA}} = \{\langle B \rangle \mid B \text{ is a DFA and } L(B) = \emptyset\}$ is decidable.

Proof: Let $D_{E_{\text{DFA}}}$ be the following TM that tests if there is a path from the start state to an accept state.

Let $B_2 =$



$D_{E_{\text{DFA}}}$ = "On input $\langle B \rangle$ where B is a DFA,

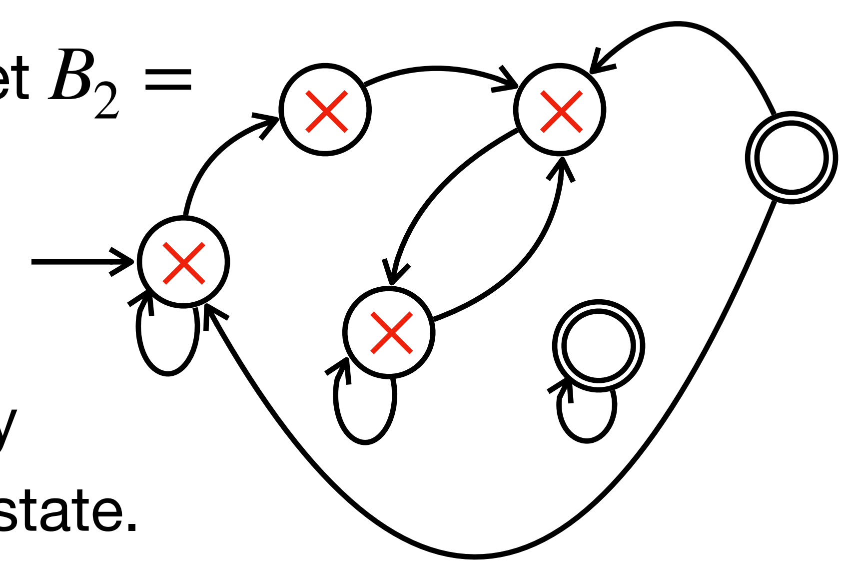
1. Mark the start state of B.
2. Repeat until no new states are marked: mark any unmarked state with a transition from a marked state.
3. If no accept state is marked, accept. Else, reject."

Emptiness Problem for DFAs

Theorem (p.196): The language $E_{\text{DFA}} = \{\langle B \rangle \mid B \text{ is a DFA and } L(B) = \emptyset\}$ is decidable.

Proof: Let $D_{E_{\text{DFA}}}$ be the following TM that tests if there is a path from the start state to an accept state.

Let $B_2 =$



$D_{E_{\text{DFA}}}$ = "On input $\langle B \rangle$ where B is a DFA,

1. Mark the start state of B.
2. Repeat until no new states are marked: mark any unmarked state with a transition from a marked state.
3. If no accept state is marked, accept. Else, reject."

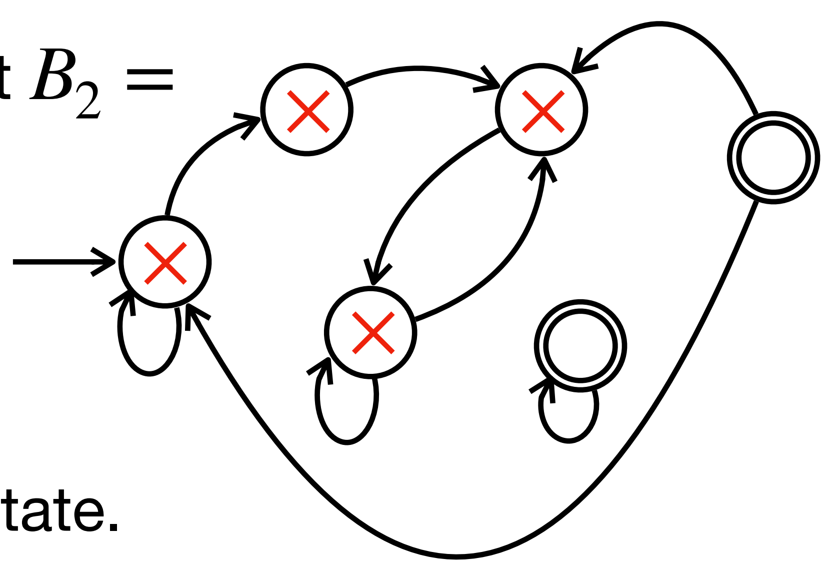
$D_{E_{\text{DFA}}}$ decides E_{DFA}.

Emptiness Problem for DFAs

Theorem (p.196): The language $E_{\text{DFA}} = \{\langle B \rangle \mid B \text{ is a DFA and } L(B) = \emptyset\}$ is decidable.

Proof: Let $D_{E_{\text{DFA}}}$ be the following TM that tests if there is a path from the start state to an accept state.

Let $B_2 =$



$D_{E_{\text{DFA}}}$ = "On input $\langle B \rangle$ where B is a DFA,

1. Mark the start state of B.
2. Repeat until no new states are marked: mark any unmarked state with a transition from a marked state.
3. If no accept state is marked, accept. Else, reject."

$D_{E_{\text{DFA}}}$ decides E_{DFA}.

$D_{E_{\text{DFA}}}$ always halts within $|Q|$ sub-steps.

Q.E.D.

Check-In 2 (Break)

Let $ALL_{\text{DFA}} = \{\langle M \rangle \mid M \text{ is a DFA and } L(M) = \Sigma^*\}$.

Prove that ALL_{DFA} is decidable.

(*Hint*: Consider what we just saw in the last slide: E_{DFA}.)

Equivalence Problem for DFAs

Theorem (p.197): The language $EQ_{\text{DFA}} = \{\langle A, B\rangle \mid A$ and B are DFAs and $L(A) = L(B)\}$ is decidable.

On the whiteboard.

Summary of Today's Lecture

- Notation for Encodings and TMs
- Decision procedures for DFAs

Acknowledgements

- These slides are based on lecture notes on Theory of Computation from other universities, namely Michael Sipser (MIT), Lorenzo De Stefani (Brown).

- **Errata:** If you let us know of any errors in the slides, we'll fix them and acknowledge you here!