21.3.2
NFAs/DFAs and Universality

DFA Universality

A DFA M is universal if it accepts every string.
That is, $L(M) = \Sigma^*$, the set of all strings.

Problem 21.2 (DFA universality).

Input: *A DFA M.*
Goal: *Is M universal?*

How do we solve **DFA Universality**?
We check if M has any reachable non-final state.

DFA Universality

A DFA M is universal if it accepts every string.
That is, $L(M) = \Sigma^*$, the set of all strings.

Problem 21.2 (DFA universality).

Input: *A DFA M.*
Goal: *Is M universal?*

How do we solve **DFA Universality**?
We check if M has <u>any</u> reachable non-final state.

DFA Universality

A DFA M is universal if it accepts every string.
That is, $L(M) = \Sigma^*$, the set of all strings.

Problem 21.2 (DFA universality).

Input: *A DFA M.*
Goal: *Is M universal?*

How do we solve **DFA Universality**?
We check if M has <u>any</u> reachable non-final state.

DFA Universality

A DFA M is universal if it accepts every string.
That is, $L(M) = \Sigma^*$, the set of all strings.

Problem 21.2 (DFA universality).

Input: *A DFA M.*
Goal: *Is M universal?*

How do we solve **DFA Universality**?
We check if M has any reachable non-final state.

NFA Universality

An NFA $\boldsymbol{N}$ is said to be universal if it accepts every string. That is, $\boldsymbol{L}(\boldsymbol{N}) = \Sigma^*$, the set of all strings.

Problem 21.3 (NFA universality).

Input: *A NFA $\boldsymbol{M}$.*
Goal: *Is $\boldsymbol{M}$ universal?*

How do we solve **NFA Universality**?

Reduce it to **DFA Universality**?

Given an NFA $\boldsymbol{N}$, convert it to an equivalent DFA $\boldsymbol{M}$, and use the **DFA Universality** Algorithm.

The reduction takes exponential time!

NFA Universality is known to be PSPACE-Complete and we do not expect a polynomial-time algorithm.

NFA Universality

An NFA N is said to be universal if it accepts every string. That is, $L(N) = \Sigma^*$, the set of all strings.

Problem 21.3 (NFA universality).

Input: *A NFA M.*
Goal: *Is M universal?*

How do we solve **NFA Universality**?
Reduce it to **DFA Universality**?
Given an NFA N, convert it to an equivalent DFA M, and use the **DFA Universality** Algorithm.
The reduction takes exponential time!
NFA Universality is known to be PSPACE-Complete and we do not expect a polynomial-time algorithm.

NFA Universality

An NFA N is said to be universal if it accepts every string. That is, $L(N) = \Sigma^*$, the set of all strings.

Problem 21.3 (NFA universality).

Input: *A NFA M.*
Goal: *Is M universal?*

How do we solve **NFA Universality**?
Reduce it to **DFA Universality**?
Given an NFA N, convert it to an equivalent DFA M, and use the **DFA Universality** Algorithm.
The reduction takes exponential time!
NFA Universality is known to be PSPACE-Complete and we do not expect a polynomial-time algorithm.

NFA Universality

An NFA N is said to be universal if it accepts every string. That is, $L(N) = \Sigma^*$, the set of all strings.

Problem 21.3 (NFA universality).

Input: *A NFA M.*
Goal: *Is M universal?*

How do we solve **NFA Universality**?
Reduce it to **DFA Universality**?
Given an NFA N, convert it to an equivalent DFA M, and use the **DFA Universality** Algorithm.
The reduction takes exponential time!
NFA Universality is known to be PSPACE-Complete and we do not expect a polynomial-time algorithm.

THE END

...

(for now)