


# 01204211 Discrete Mathematics

## Lecture 9b: Nonregular languages<sup>1</sup>

Jittat Fakcharoenphol

August 26, 2024

---

<sup>1</sup>Based on lecture notes of *Models of Computation* course by Jeff Erickson. 

# DFA: Formal definitions

A **finite-state machine** or a **deterministic finite-state automaton** (DFA) has five components:

- ▶ the input alphabet  $\Sigma$ ,
- ▶ a finite set of states  $Q$ ,
- ▶ a transition function  $\delta : Q \times \Sigma \longrightarrow Q$
- ▶ a start state  $s \in Q$ , and
- ▶ a subset  $A \subseteq Q$  of accepting states.

## Acceptance

**One step move:** from state  $q$  with input symbol  $a$ , the machine changes its state to  $\delta(q, a)$ .

**Extension:** from state  $q$  with input string  $w$ , the machine changes its state to  $\delta^*(q, w)$  defined as

$$\delta^*(q, w) = \begin{cases} q & \text{if } w = \varepsilon, \\ \delta^*(\delta(q, a), x) & \text{if } w = ax. \end{cases}$$

The signature of  $\delta^*$  is  $Q \times \Sigma^* \longrightarrow Q$ .

accepting  $w$

For a finite-state machine with starting state  $s$  and accepting states  $A$ , it accepts string  $w$  iff

$$\delta^*(s, w) \in A.$$

# Language of a DFA

## $L(M)$

For a DFA  $M$ , let  $L(M)$  be the set of all strings that  $M$  accepts. More formally, for  $M = (\Sigma, Q, \delta, s, A)$ ,

$$L(M) = \{w \in \Sigma^* \mid \delta^*(s, w) \in A\}.$$

We refer to  $L(M)$  as the language of  $M$ .

# Automatic languages<sup>2</sup>

## Definition (for now)

A language  $L$  is “**automatic**” if there is a DFA  $M$  such that  $L(M) = L$ .

## Lemma 1

*If  $L_1$  and  $L_2$  are automatic languages over alphabet  $\Sigma$ , then*

- ▶  $L_1 \cap L_2$ ,
- ▶  $L_1 \cup L_2$ ,
- ▶  $L_1 \setminus L_2$ , *and*
- ▶  $\Sigma^* \setminus L_1$ ,

*are also automatic.*

The set of automatic languages is closed under these boolean operations.

---

<sup>2</sup>Taken directly from Erikson's lecture notes

## Other ways to combine DFAs?

Given two languages  $L_1$  and  $L_2$ , we can combine them in various ways using Boolean operations (i.e.,  $\cap$ ,  $\cup$ , etc.).

What else can we do?

## Other ways to combine DFAs?

Given two languages  $L_1$  and  $L_2$ , we can combine them in various ways using Boolean operations (i.e.,  $\cap$ ,  $\cup$ , etc.).

What else can we do?

- Concatenation:  $L_1 \cdot L_2$ , defined as

## Other ways to combine DFAs?

Given two languages  $L_1$  and  $L_2$ , we can combine them in various ways using Boolean operations (i.e.,  $\cap$ ,  $\cup$ , etc.).

What else can we do?

- Concatenation:  $L_1 \cdot L_2$ , defined as

$$\{x \cdot y \mid x \in L_1, y \in L_2\}$$



## Other ways to combine DFAs?

Given two languages  $L_1$  and  $L_2$ , we can combine them in various ways using Boolean operations (i.e.,  $\cap$ ,  $\cup$ , etc.).

What else can we do?

- Concatenation:  $L_1 \cdot L_2$ , defined as

$$\{x \cdot y \mid x \in L_1, y \in L_2\}$$

- Kleene closure:  $L_1^*$ .

# Interesting questions

We know that the set of automatic languages is closed under Boolean operations.

## Questions

- ▶ Is it closed under concatenation?
- ▶ Is it closed under taking Kleene closure?

# Interesting questions

We know that the set of automatic languages is closed under Boolean operations.

## Questions

- ▶ Is it closed under concatenation?
- ▶ Is it closed under taking Kleene closure?

**Spoiler:**

# Interesting questions

We know that the set of automatic languages is closed under Boolean operations.

## Questions

- ▶ Is it closed under concatenation?
- ▶ Is it closed under taking Kleene closure?

**Spoiler:** Yes, it is (for both operations). We will see the proof, after we learn a required new concept.

# Closure

## Lemma 2

*Given two automatic languages  $L_1$  and  $L_2$ , the following languages are automatic:*

- ▶  $L_1 \cup L_2$ ,
- ▶  $L_1 \cdot L_2$ , and
- ▶  $L_1^*$ .

# Closure

## Lemma 2

*Given two automatic languages  $L_1$  and  $L_2$ , the following languages are automatic:*

- ▶  $L_1 \cup L_2$ ,
- ▶  $L_1 \cdot L_2$ , and
- ▶  $L_1^*$ .

More over,  $\emptyset$  and a language containing a single string are also automatic.

# Closure

More over,  $\emptyset$  and a language containing a single string are also automatic.

## Lemma 2

*These are automatic languages*

- ▶ *The empty set,*
- ▶ *A language containing one string,*
- ▶  *$L_1 \cup L_2$  for automatic languages  $L_1$  and  $L_2$ ,*
- ▶  *$L_1 \cdot L_2$  for automatic languages  $L_1$  and  $L_2$ , and*
- ▶  *$L^*$  for automatic languages  $L$ .*

Doese this look familiar?

# Regular languages

## Definition: regular languages

A language  $L$  is **regular** if and only if it satisfies one of the following conditions:

- ▶  $L$  is empty;
- ▶  $L$  contains one string (can be the empty string  $\varepsilon$ );
- ▶  $L$  is a union of two regular languages;
- ▶  $L$  is the concatenation of two regular languages; or
- ▶  $L$  is the Kleene closure of a regular language.





Every regular language is automatic

Big question:

$\Rightarrow$

Every regular language is automatic

Big question:

$\Leftarrow$

Is every automatic language regular?

**Spoiler:**

$\Rightarrow$

Every regular language is automatic

Big question:

$\Leftarrow$

Is every automatic language regular?

**Spoiler:** Yes, it is. We will see some idea on how to prove this.

⇒

Every regular language is automatic

Big question:

⇐

Is every automatic language regular?

**Spoiler:** Yes, it is. We will see some idea on how to prove this.

Theorem 3

*A language  $L$  is regular if and only if there exists a DFA  $M$  such that  $L(M) = L$ .*

# Nonregular languages

Can you design a DFA that accepts strings from language

$$\{0^n 1^n \mid n \geq 0\}$$

## Key idea

If you have finite states, you can't possibly distinguish between strings in the language and strings not in the language.

## Basic question

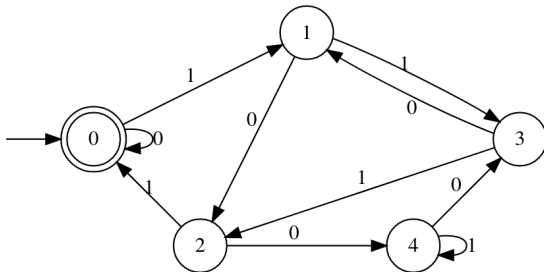
How can you show that you need at least two states?

## Basic question

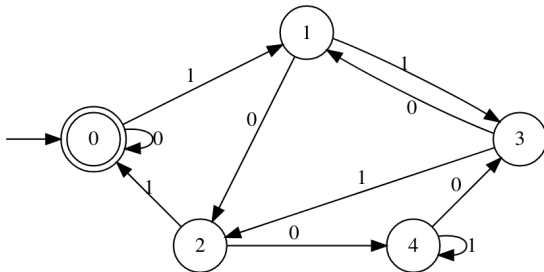
How can you show that you need at least two states?  
Let's see how a DFA works.



## Another example

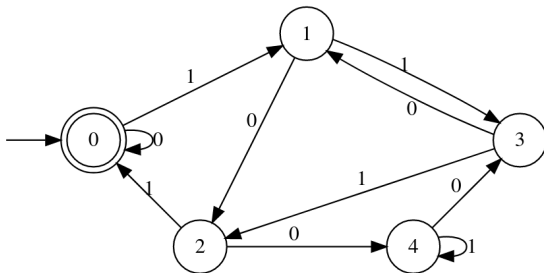


## Another example



If string  $x$  and  $y$  reach the same state in a DFA, for any string  $z$ , both  $xz$  and  $yz$  must reach the same state.

## Another example



If string  $x$  and  $y$  reach the same state in a DFA, for any string  $z$ , both  $xz$  and  $yz$  must reach the same state.

In other words, a DFA accepts  $xz$  iff it accepts  $yz$ .

## Distinguishing suffixes

Consider language  $L = \{0^n 1^n \mid n \geq 0\}$ .

Consider  $x = 0$  and  $y = 00$ . Consider suffix  $z = 11$ .

## Distinguishing suffixes

Consider language  $L = \{0^n 1^n \mid n \geq 0\}$ .

Consider  $x = 0$  and  $y = 00$ . Consider suffix  $z = 11$ .

We have that

$$xz = 0\mathbf{11} \notin L,$$

but

$$yz = 00\mathbf{11} \in L.$$

## Distinguishing suffixes

Consider language  $L = \{0^n 1^n \mid n \geq 0\}$ .

Consider  $x = 0$  and  $y = 00$ . Consider suffix  $z = 11$ .

We have that

$$xz = 0\mathbf{11} \notin L,$$

but

$$yz = 00\mathbf{11} \in L.$$

What can you say about a DFA  $M$  such that  $L(M) = L$ ?

## Distinguishing suffixes

Consider language  $L = \{0^n 1^n \mid n \geq 0\}$ .

Consider  $x = 0$  and  $y = 00$ . Consider suffix  $z = 11$ .

We have that

$$xz = 0\mathbf{11} \notin L,$$

but

$$yz = 00\mathbf{11} \in L.$$

What can you say about a DFA  $M$  such that  $L(M) = L$ ?

### Definition

For strings  $x$  and  $y$ , string  $z$  is a **distinguishing suffix** with respect to  $L$  if exactly one of  $xz$  and  $yz$  is in  $L$ .

## Fooling sets

A **fooling set** for a language  $L$  is set  $F$  of strings such that every pair of strings in  $F$  has a distinguishing suffix.

**Example:** The set  $\{0, 00, 000\}$  is a fooling set for  $L = \{0^n 1^n \mid n \geq 0\}$ .



# A large fooling set

## Lemma 4

*The set  $\{0^n \mid n \geq 0\}$  is a fooling set for  $L = \{0^n 1^n \mid n \geq 0\}$ .*

Proof.



## Observation

If language  $L$  has an infinite fooling set,  $L$  is not regular

## Observation

If language  $L$  has an infinite fooling set,  $L$  is not regular

## Lemma 5

*Language  $L = \{0^n 1^n \mid n \geq 0\}$  is not regular.*

## Proof.

We previously establish that the set  $F = \{0^n \mid n \geq 0\}$  is a fooling set for  $L$ . Since  $F$  has infinite size, from the observation, we know that  $L$  is not regular. □

## Lemma 6

*For  $\Sigma = \{0, 1\}$ , the language  $L = \{ww^R \mid w \in \Sigma^*\}$  is not regular.*

Proof.



$L = \{0^{2^n} \mid n \geq 0\}$ : Proof 1

### Lemma 7

*For  $\Sigma = \{0\}$ , the language  $L = \{0^{2^n} \mid n \geq 0\}$  is not regular.*

Proof.



$L = \{0^{2^n} \mid n \geq 0\}$ : Proof 2

### Lemma 8

*For  $\Sigma = \{0\}$ , the language  $L = \{0^{2^n} \mid n \geq 0\}$  is not regular.*

Proof.



$L = \{0^{2^n} \mid n \geq 0\}$ : Proof 3

### Lemma 9

*For  $\Sigma = \{0\}$ , the language  $L = \{0^{2^n} \mid n \geq 0\}$  is not regular.*

Proof.



$L = \{0^p \mid p \text{ is prime}\}$ : Proof 1

### Lemma 10

*For  $\Sigma = \{0\}$ , the language  $L = \{0^p \mid p \text{ is prime}\}$  is not regular.*

Proof.





$L = \{0^p \mid p \text{ is prime}\}$ : Proof 2

### Lemma 11

*For  $\Sigma = \{0\}$ , the language  $L = \{0^p \mid p \text{ is prime}\}$  is not regular.*

Proof.

