San José State University
Department of Computer Science

Ahmad Yazdankhah
ahmad.yazdankhah@sjsu.edu
www.cs.sjsu.edu/~yazdankhah

Nondeterministic Finite Automata

(Part 3)

Lecture 11

Day 12/31

CS 154

Formal Languages and Computability

Fall 2019

Agenda of Day 11

- Announcement
- Summary of Lecture 10
- Lecture 11: Teaching ...
  - Nondeterministic Finite Automata (Part 3)
- Solution and Feedback of Quiz +
- Solution of HW1

Announcement

- Wrong file name was ignored only this time!
- We need to read the requirements at least 10 times!

- HW1 was individual assignment but looks like some students forgot!
- Because I found some obvious similarities in some students' solutions, but I ignored!

- It's your job to check Canvas!

- HW2 is posted.

Solution and Feedback of Quiz + (Out of 60)

Section	Average	High Score	Low Score
01 (TR 3:00 PM)	51	59	34
02 (TR 4:30 PM)	51	59	42
03 (TR 6:00 PM)	54	58.5	37

Summary of Lecture 10: We learned ...

λ-transition

- Short circuit is ...

  ... an edge with no input symbol.

- We represent it with symbol λ.
- The transition is called λ-transition.
  - In fact λ means "NO symbol".
- λ-transition in automata theory is a transition that ...

  ... the machine can unconditionally transit.

- This is a general definition for all types of automata.

- The sub-rule of the following transition is ...

$$\delta\ (q_3\ ,\ a) = \{q_9\ ,\ q_{21}\}$$

- As a general rule, when NFAs encounter multiple choices, they start parallel processing.

Any question?

Summary of Lecture 10: We learned ...

NFAs Formal Definition

- An NFA M is defined by a quintuple:

$$M = (Q, \Sigma, \delta, q_0, F)$$

- Except δ, the rest items are similar to DFAs'.

$$\delta : Q \times \Sigma \rightarrow 2^Q$$

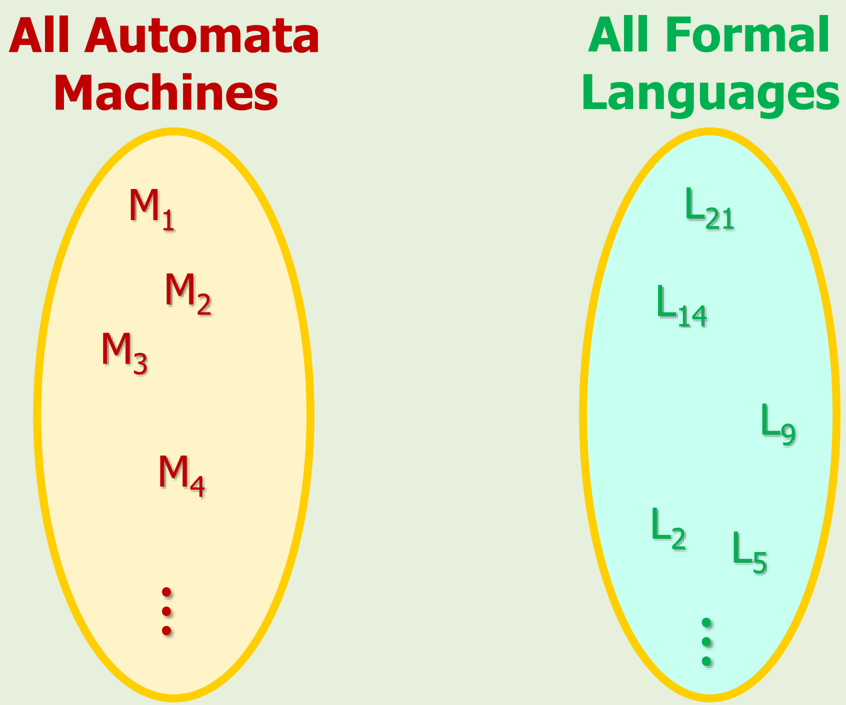
δ is total function.

Any question?

Machines and Languages Association

Machines and Languages Association

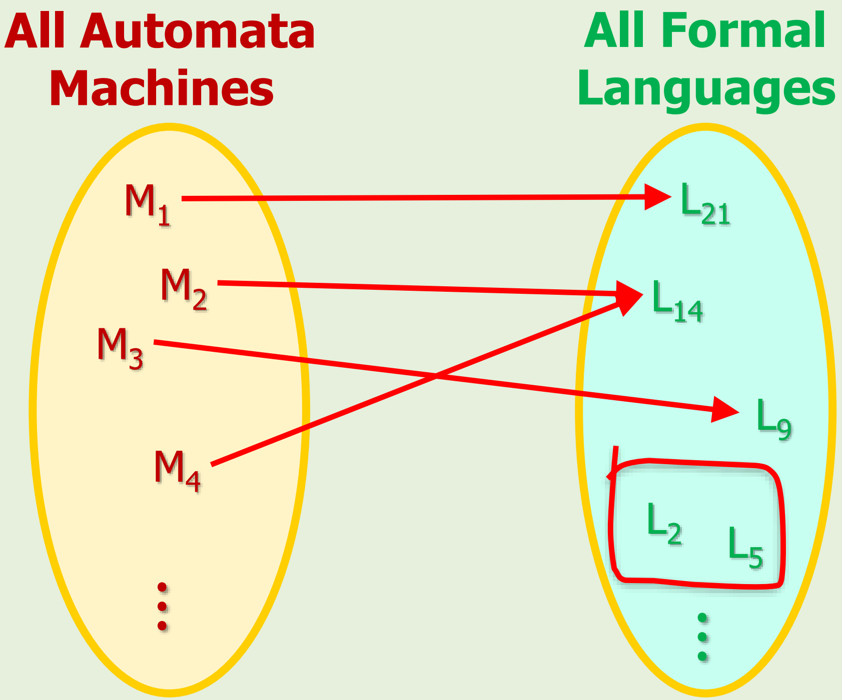
- What is the relationship between ...
- ... the set of all automata machines and ...
- ... the set of all formal languages?



One of the most interesting topics of computer science

Machines and Languages Association

- We've learned that "every machine has an associated language".
- BUT we don't know yet whether for every language, we can construct a machine!
  - Our knowledge is not enough yet.



A Side Note: Computer Scientists Mission

- Why should we be interested in the relationship between machines and languages?
- We'll see later that we can encode all problems into formal languages.

Formal Language ≡ Problem

Accepting (understanding, recognizing) a language
≡ Solving the problem

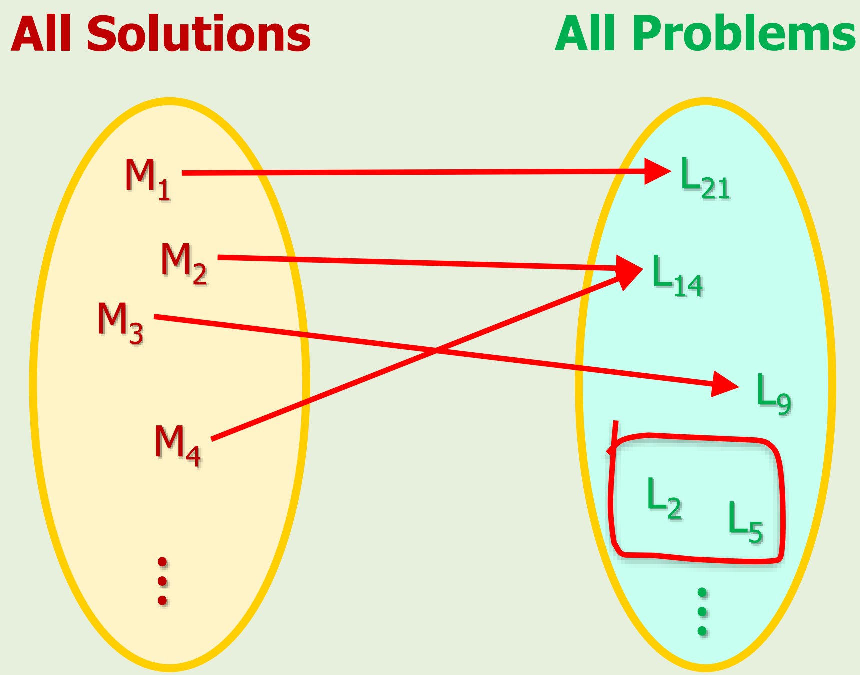
- So, as computer scientists, our mission is:

To design a machine for every language ≡ To solve the problems

- In other words, when you design a machine, it means you are solving the problem.

Machines and Languages Association

- Now, with this background, let's look at the association again.
- Let's rename them to "Solutions" and "Problems".
- Obviously, it's true that every solution is related to a problem!
- But, is this the case that for every problem, there is a solution?



7. DFAs vs NFAs

Objective

- The goal of this section is to compare DFAs and NFAs.
- To compare two classes of automata,
  we'd need some "metrics".
- We'll use the concept of "power"
  as the metrics for comparison.
- So, first we need to define "power".

Power of Automata Classes

- Let's assume we have two classes of automata:
  - Class A (e.g. NFAs)
  - Class B (e.g. DFAs)

Question

- What is the best criteria to state that:

Class A is "more powerful" than class B?

Answer

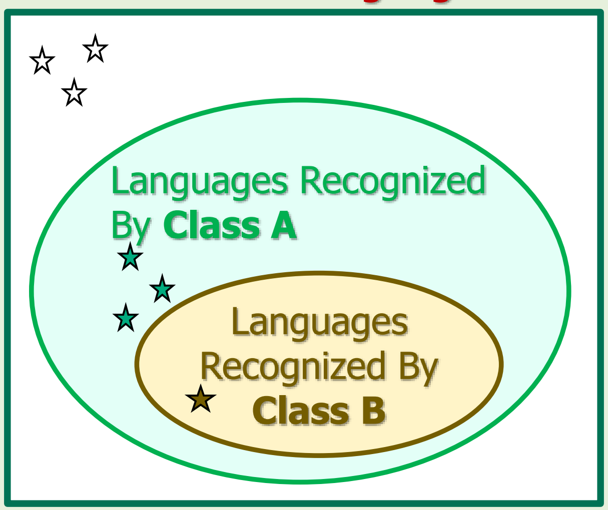
- If class A can solve (accept, understand, recognize) more problems, then it is more powerful.

Power of Automata Classes

Definition

- The (automata) class A is "more powerful" than class B if the set of languages recognized by class B is a proper subset of the set of the languages recognized by class A.

U = All Formal Languages



DFAs and NFAs Relationship

- Let's get back to our topic in this section:

  DFAs vs. NFAs

- If the universal set is the set of all formal languages:

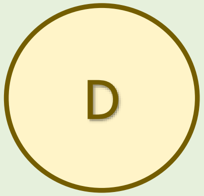
  1. What portion of the formal languages can be recognized by DFAs?
  2. What portion can be recognized by NFAs?

DFAs and NFAs Relationship

- Let's use the following definitions and notations:

U = {x : x is a formal language}

D = {d : d ∈ U, d is recognized by a DFA}

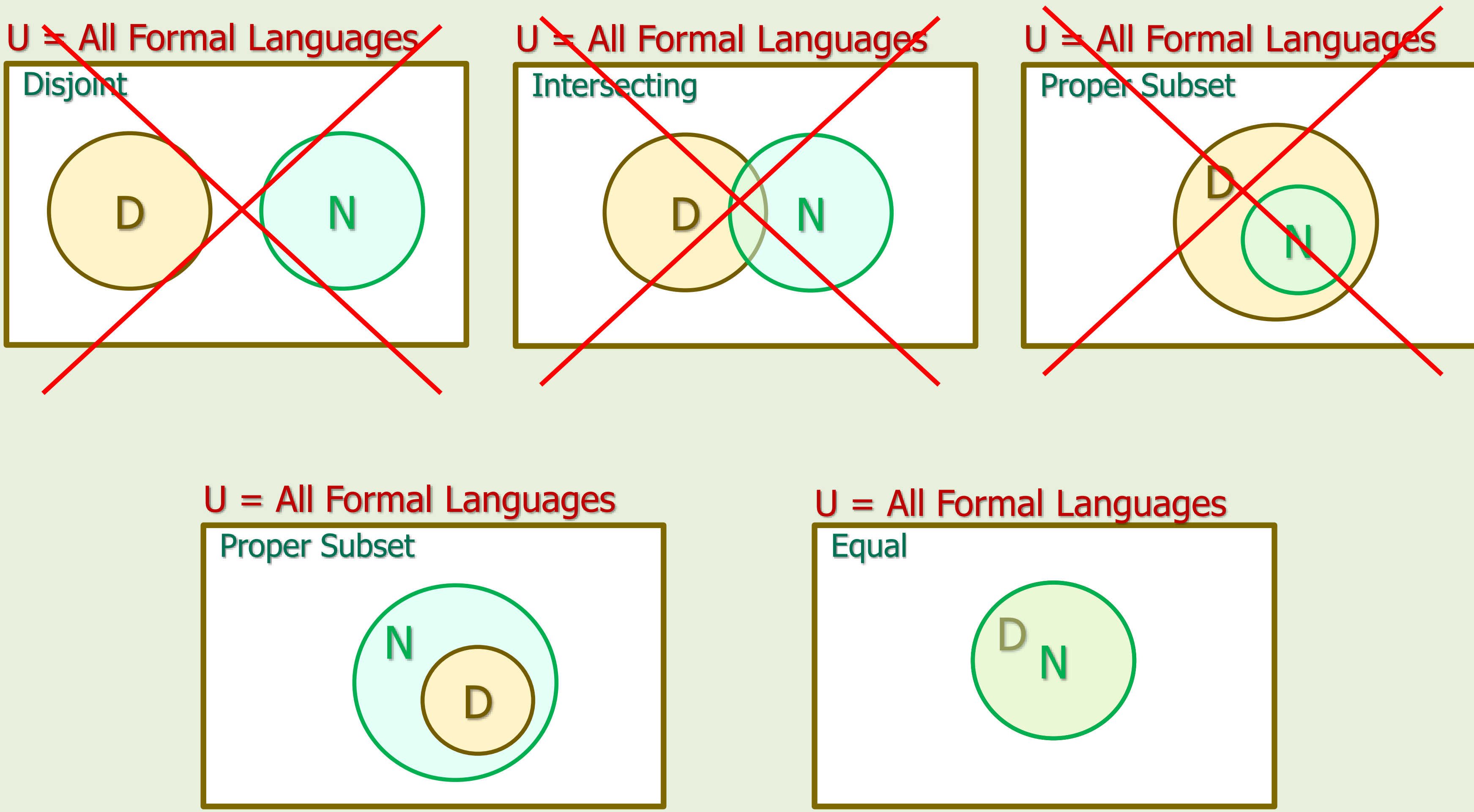


N = {n : n ∈ U, n is recognized by a NFA}

- What is the relationship between set D and set N?

DFAs and NFAs Relationship

- Which one is reasonable relationship between D and N?



Can NFAs Simulate DFAs?

- Let's assume that we've constructed a DFA for an arbitrary language L.
- Can we always construct an NFA for L?
- Yes! How?
- Mathematically speaking, the only difference between the definition of NFAs and DFAs is their transition function.
- So, we should prove that we can always convert a DFA's definition to an NFA's definition.
- Let's show this through an example.

Can NFAs Simulate DFAs?

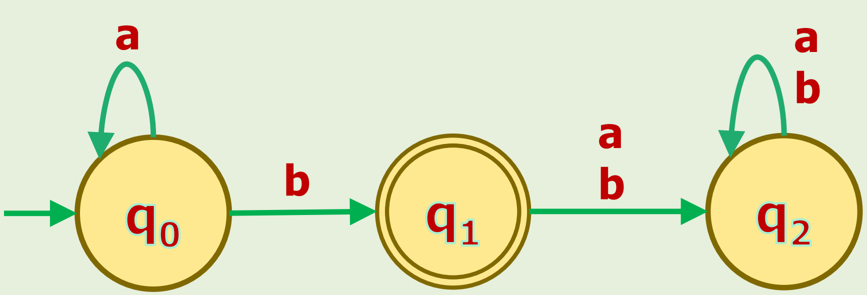
Example 19

- Convert the following DFA's definition to an NFA's.
- q_0 is the initial state, and q_1 is the final state.

$$\begin{cases} \delta(q_0, a) = q_0 \\ \delta(q_0, b) = q_1 \\ \delta(q_1, a) = q_2 \\ \delta(q_1, b) = q_2 \\ \delta(q_2, a) = q_2 \\ \delta(q_2, b) = q_2 \end{cases} \quad \Rightarrow \quad \begin{cases} \delta(q_0, a) = \{q_0\} \\ \delta(q_0, b) = \{q_1\} \\ \delta(q_1, a) = \{q_2\} \\ \delta(q_1, b) = \{q_2\} \\ \delta(q_2, a) = \{q_2\} \\ \delta(q_2, b) = \{q_2\} \end{cases}$$

DFA **NFA**

- Just convert the δ.
- The rest items are the same.



DFAs Can be Converted Directly to NFAs

	DFA	NFA
States	$Q = \{q_0, q_1, q_2\}$	$Q = \{q_0, q_1, q_2\}$
Alphabet	$\Sigma = \{a, b\}$	$\Sigma = \{a, b\}$
Sub-rule	$\delta(q_i, a) = q_j$	$\delta(q_i, a) = \{q_j\}$
Initial state	q_0	q_0
Final states	$F = \{q_1\}$	$F = \{q_1\}$

Can NFAs Simulate DFAs?

- As the previous example showed, there is a simple algorithm to convert a DFA to an NFA.

Algorithm: Converting DFAs' Formal Definition to NFAs'

- Change all DFAs' sub-rules to NFAs format by enclosing the range element with a pair of curly brackets. i.e.:

$$\delta(q_i , x) = q_j$$

changes to

$$\delta(q_i , x) = \{q_j\}$$

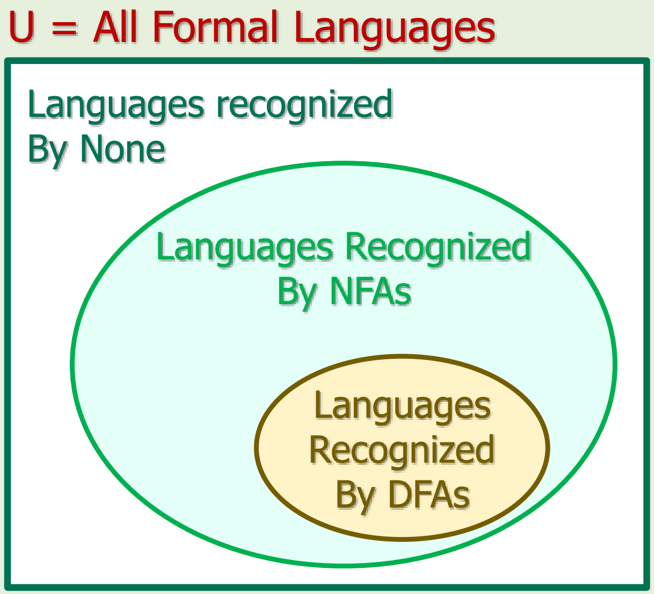
- The rest of the items, (i.e. Q, Σ, q_0, F) don't need to be touched.

- Finding an algorithm is equivalent to mathematical proof.

Can NFAs Simulate DFAs?

Conclusion

- Can NFAs simulate DFAs?
- Yes, the set of all languages recognized by DFAs can be recognized by NFAs too.



- Now, let's ask another question ...

Can DFAs Simulate NFAs?

- Let's assume that we've constructed an NFA for an arbitrary language L.
- Can we always construct a DFA for L?
- The answer of this question is not so obvious.
- Let's take an example to make it clear.

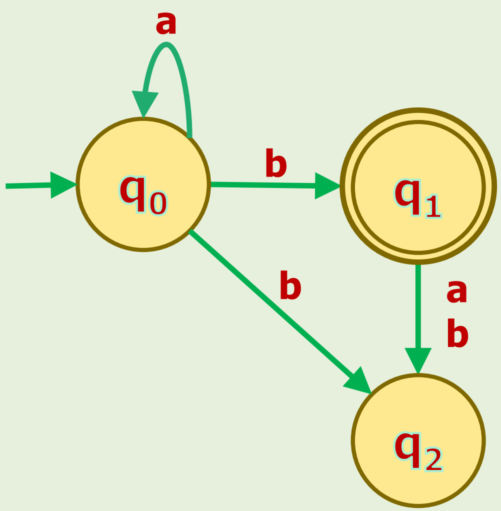
Can DFAs Simulate NFAs?

Example 20

- Can we convert the following NFA to a DFA?
- q_0 is the initial state, and q_1 is the final state.

$$\begin{cases} \delta(q_0, a) = \{q_0\} \\ \delta(q_0, b) = \{q_1, q_2\} \\ \delta(q_1, a) = \{q_2\} \\ \delta(q_1, b) = \{q_2\} \\ \delta(q_2, a) = \{\,\} \\ \delta(q_2, b) = \{\,\} \end{cases}$$

NFA ⇒ ? DFA



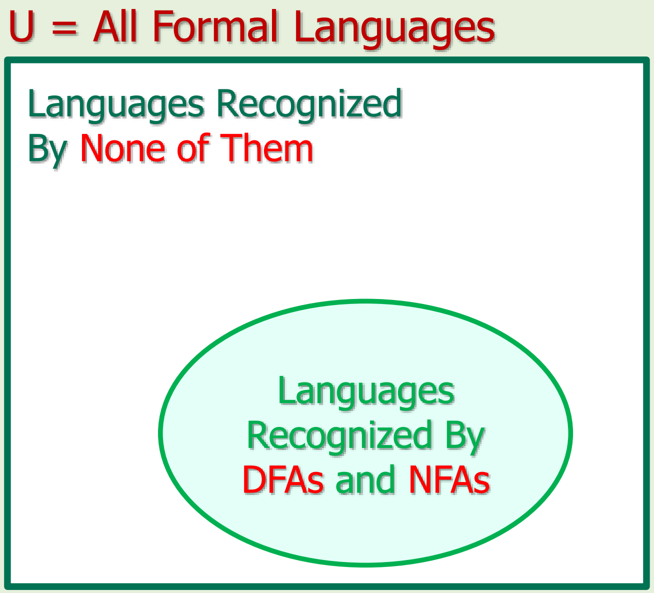
- Yes, but it needs a special technique to convert an NFA to a DFA.
- We might cover it later if we have time!

DFAs and NFAs are Equivalent

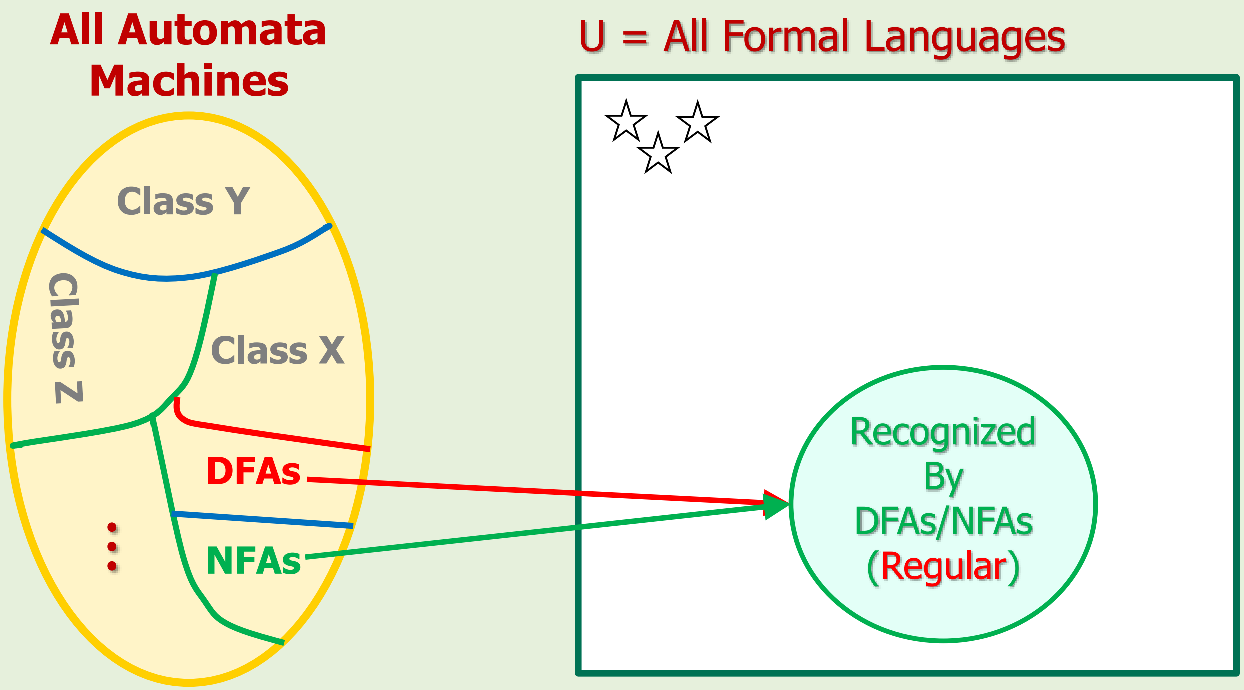
- DFAs and NFAs are equivalent as the following theorem states.

Theorem

- The set of languages recognized by NFAs is equal to the set of languages recognized by DFAs.

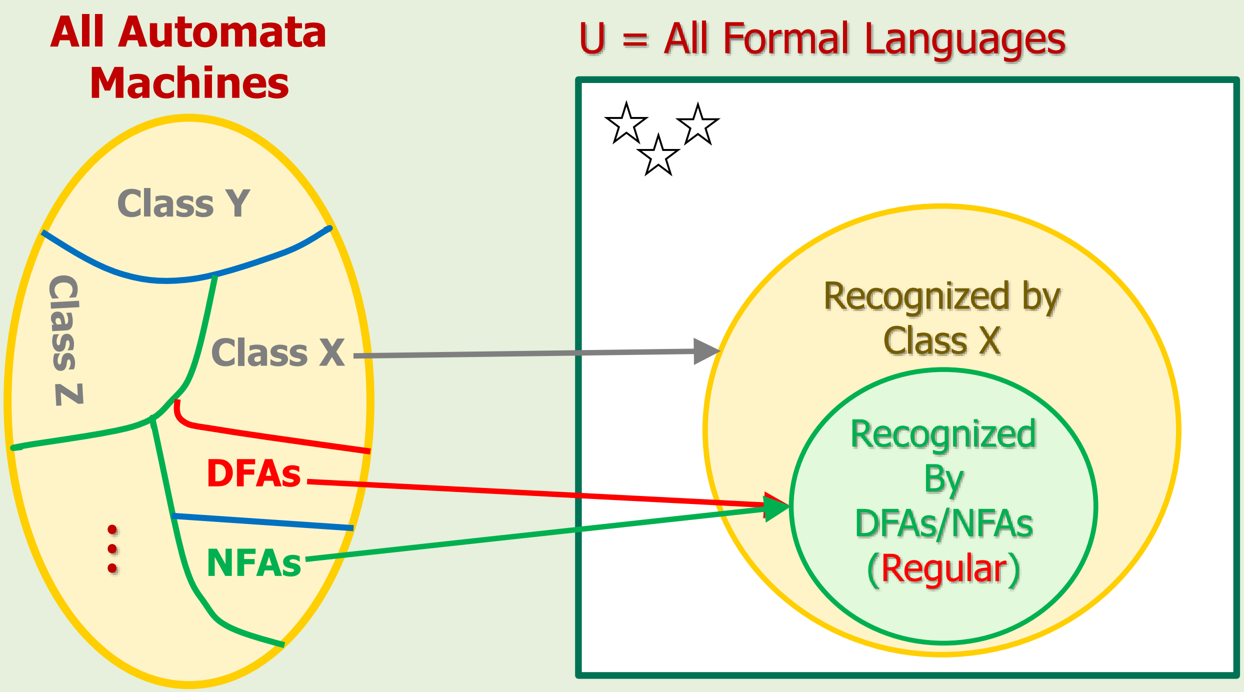


Machines and Languages Association



- DFAs and NFAs have the same power because both recognize the same portion of languages.

Machines and Languages Association



- Later, we'll define other classes of automata (i.e. Class X, Y, Z, etc.) to recognize more languages.
- But note that those new classes must recognize NFAs' portion and more!

References

1. Linz, Peter, "An Introduction to Formal Languages and Automata, 5th ed.," Jones & Bartlett Learning, LLC, Canada, 2012
2. Michael Sipser, "Introduction to the Theory of Computation, 3rd ed.," CENGAGE Learning, United States, 2013
   ISBN-13: 978-1133187790