

# Introduction

Tutorial - 1



# Who am I?

- Name: Reza Gholizadeh
- Call me: Reza
- Email: [reza.gholizadeh@ucalgary.ca](mailto:reza.gholizadeh@ucalgary.ca)
- Master of Science in Computer Networks



# Be Patient and Give Feedback

- I'm a Computer Networks guys
- I'm learning some parts of the course, but ahead of you
- I'll try my best to be helpful
- I need you to be patient with my possible mistakes



# Course Information

- We do everything on **D2L**.
- Recommended Textbook: **Compiler Construction: Principles and Practice (Louden)**



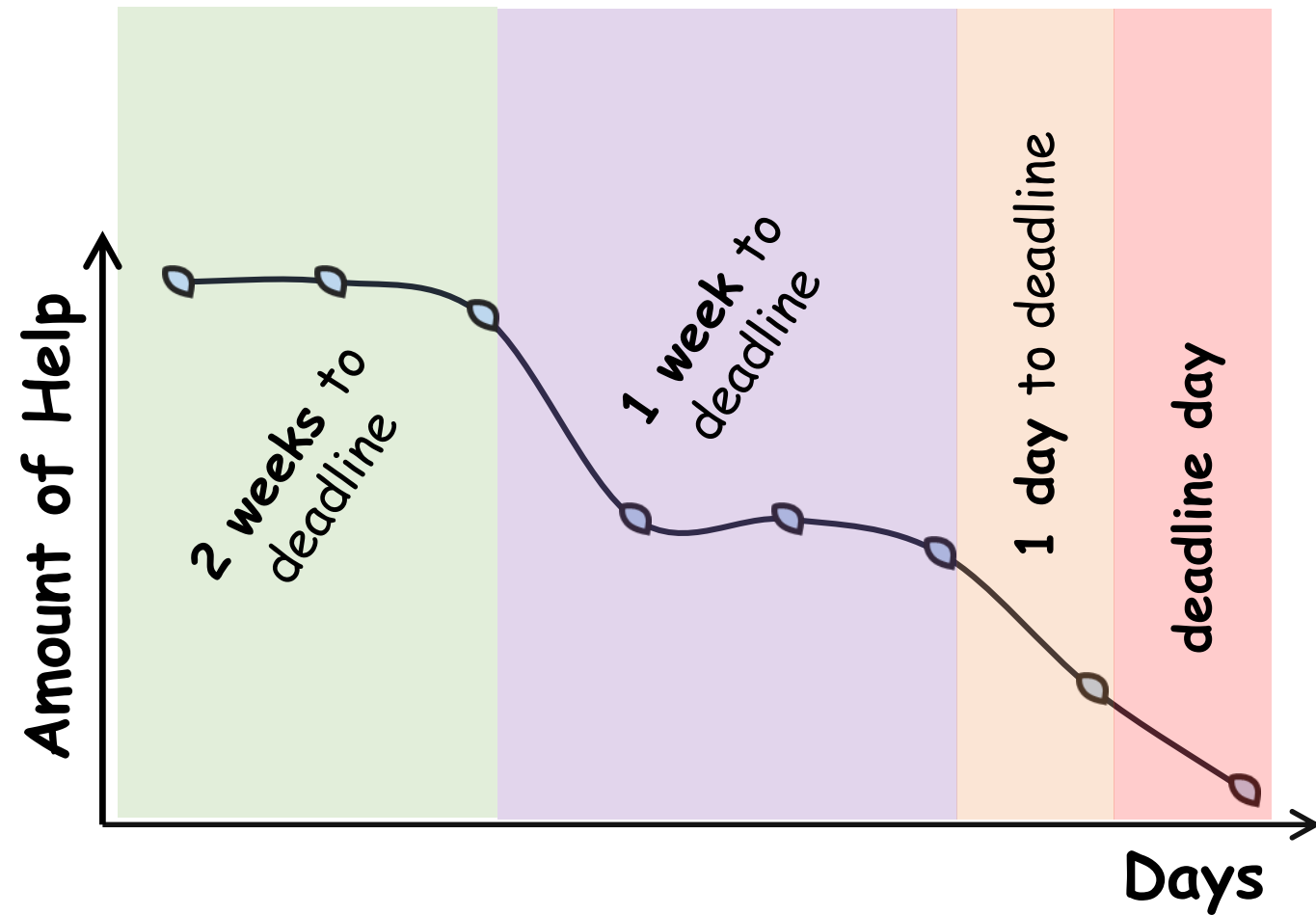
# Ground Rules

- You must read the tutorials at least once before asking a question via email
  - Password Mechanism: I'll put a password in each tutorial slide deck and you have to put the most recent one in the subject of your email (no password, no answer)
- Office hours: If you attend the tutorials, I'll try my best to give you some (Although it is not in my contract)



# Help you can get from TA

The amount of help you can get from me decreases as you get closer to the deadline



# Assignments

- There will be 4 assignments
- Each of them 10% of your total mark
- The score is out of 40 (+ 4 marks bonus)



# Assignment Sections

## 1. Written Questions:

- Concepts and ideas in the course. Must be answered in your own word. Easy to get marks
- Exercises

## 2. Project

- Creating a compiler in 4 phases
- Can be done in groups of two
- Each components will interface with the next one

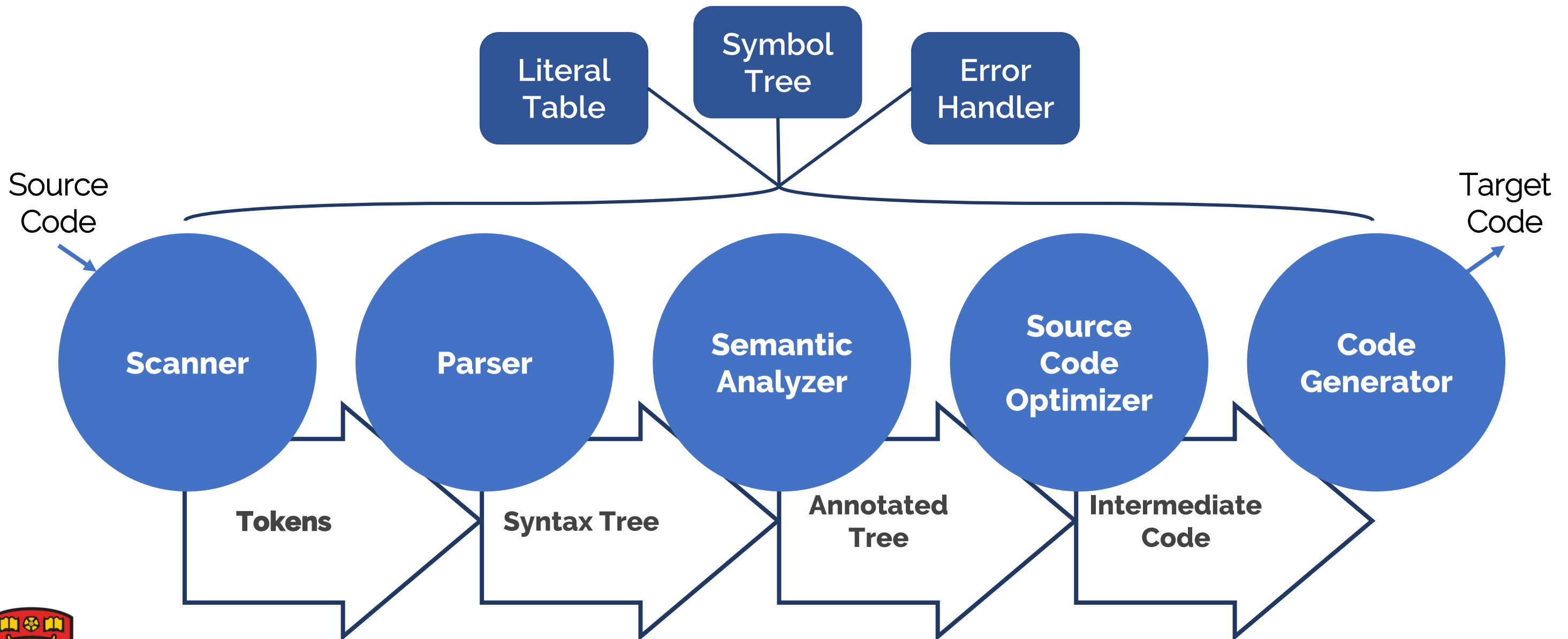




# What is a Compiler?

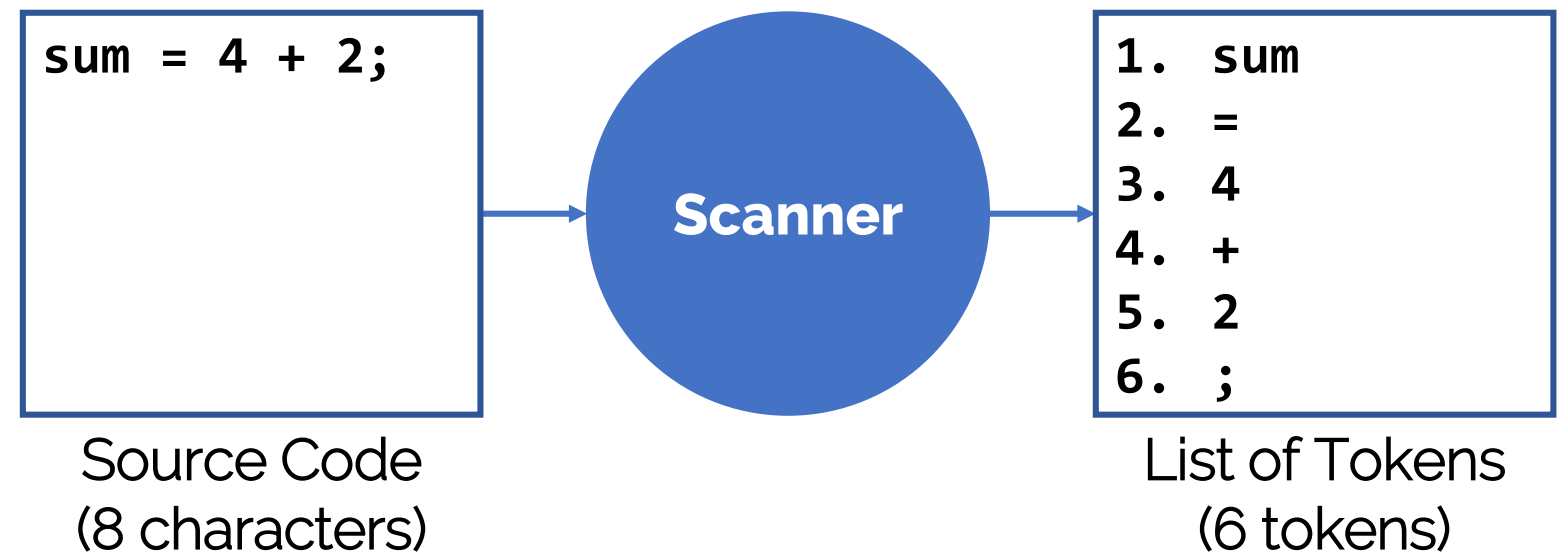


# Compilation Process



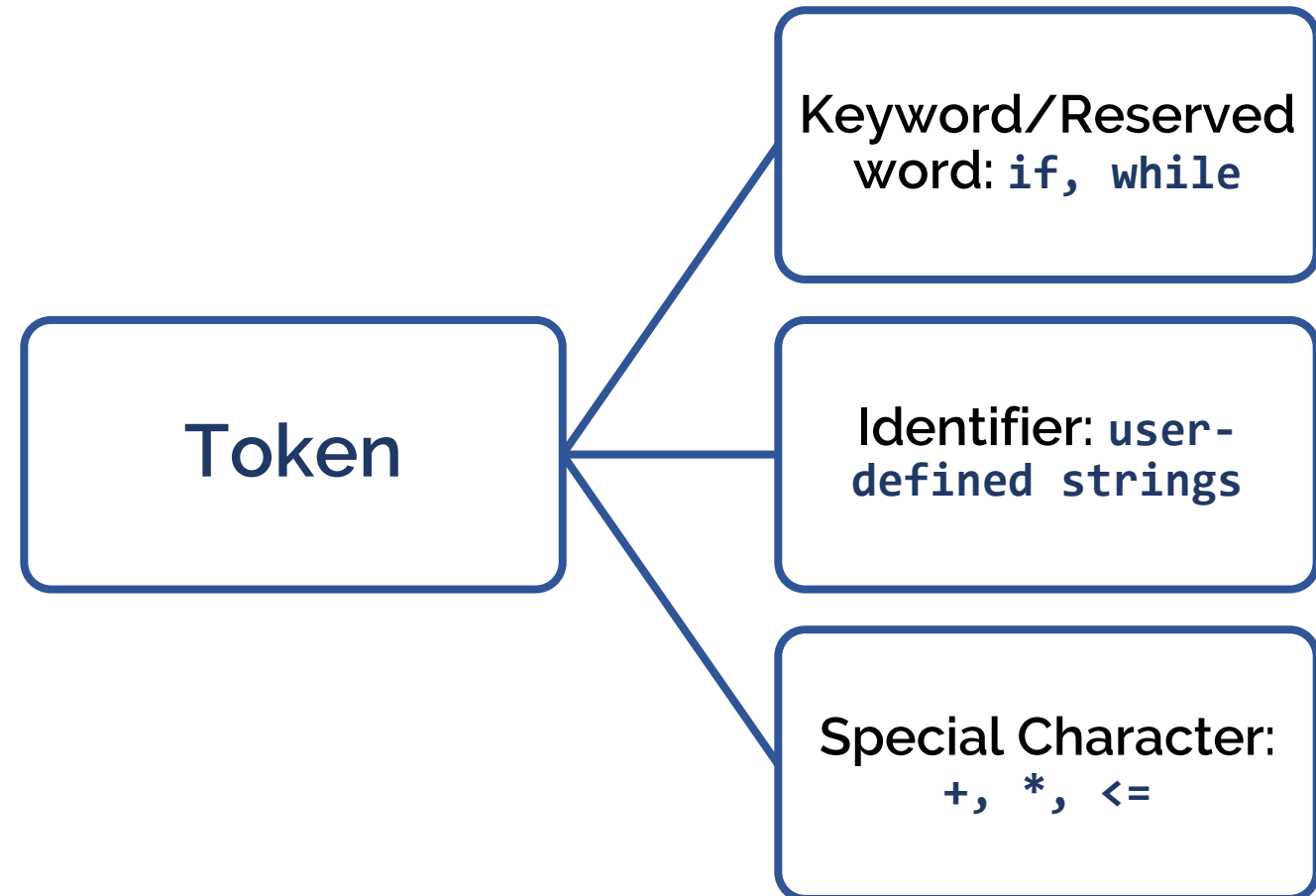
# Scanner

- Transforms **source code** to **tokens**  
(Lexical Analysis)



# Token

- Has a name and optional attributes value
- Meaningful word for to parser



# Regular Expression

- Character set or alphabet:

$$\Sigma = [a-zA-Z], \Sigma = [0-9]$$

- Basic regular expression: single characters from the alphabet
- Name for regular expr:

$$\text{digit} = 0|1|\dots|9, \text{NAT} = \text{digit digit}^*$$



# Operations

- Various operations help to define specific regular expressions

Operator	Definition	Example
	Choice among alternatives	<b>a b</b>
<b>Concatenation</b>	-	<b>ab</b>
*	Repetition or "Closure"	<b>a*</b>
+	One or more repetition	<b>(a b)+</b>
.	Any character	<b>.*b</b>
[ ]	Range of characters	<b>[a-z]</b>
~ or ^	Any character not in a given set	<b>~(a b) or [^ab]</b>
?	Optional character	<b>ab?</b>



# Example

The regular expressions for the following languages over the alphabet  $\Sigma = \{0, 1\}$ :

- The set of all strings which start and end with the same digit.

Answer:

$$(1(0^*1)^*) \mid (0(1^*0)^*)$$


# Example

The regular expressions for the following languages over the alphabet  $\Sigma = \{0, 1\}$ :

- The set of all strings representing a binary number where the sum of its digits is even.

Answer:

$$(0^*10^*1)^*0^*$$





# Example

The regular expressions for the following languages over the alphabet  $\Sigma = \{0, 1\}$ :

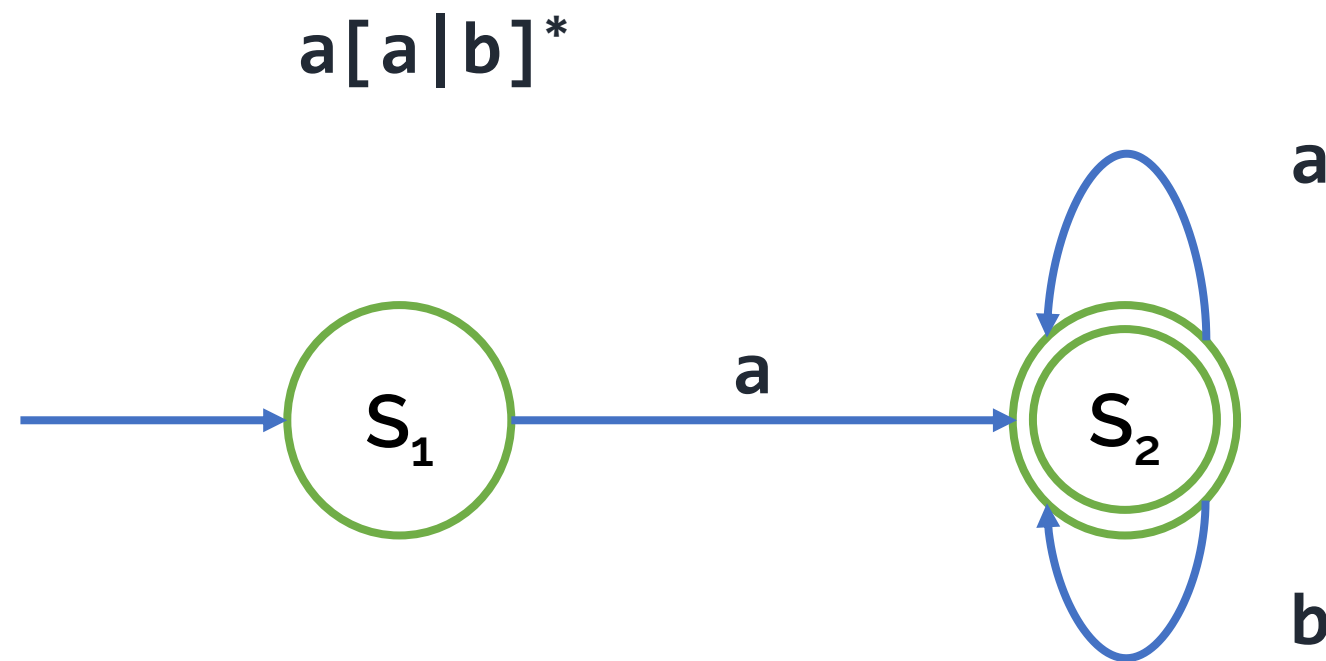
- The set of all strings that contain the substring 10100.

Answer:

$(0|1)^*10100(0|1)^*$

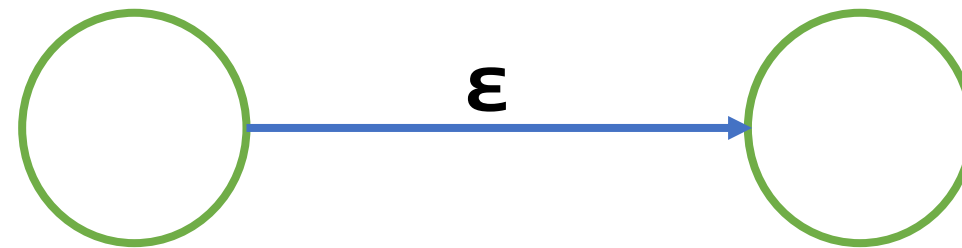


# What is a DFA?



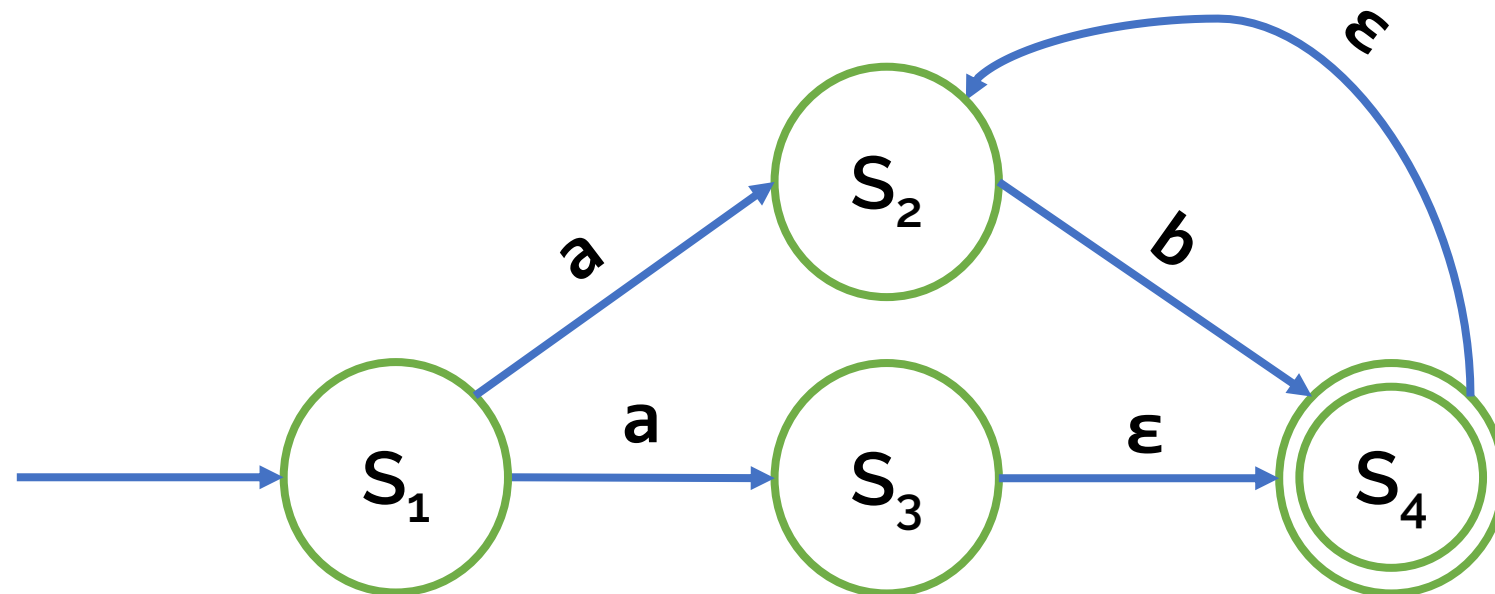
# $\epsilon$ -transition

- A transition without consuming characters

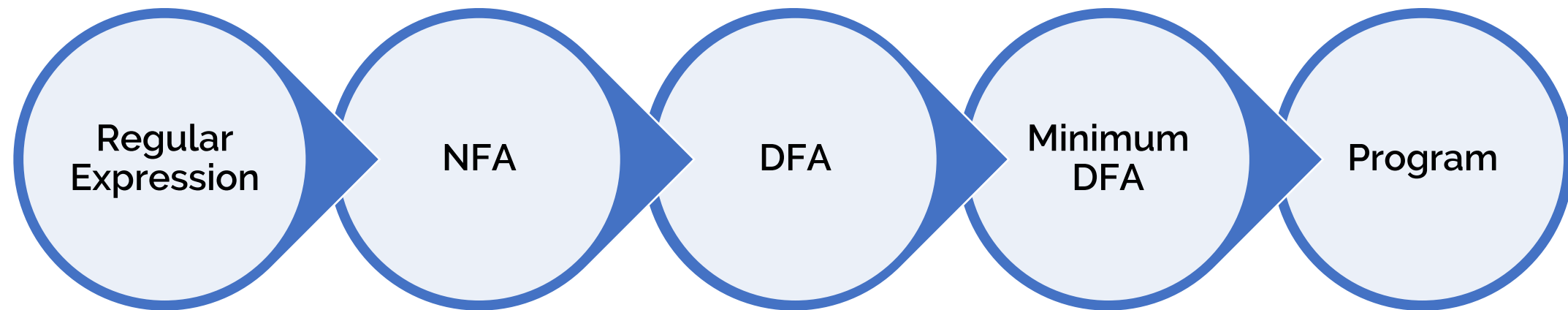


# What is an NFA?

- May includes cases with more than one transition from a state for a particular character
- May includes  $\epsilon$ -transition

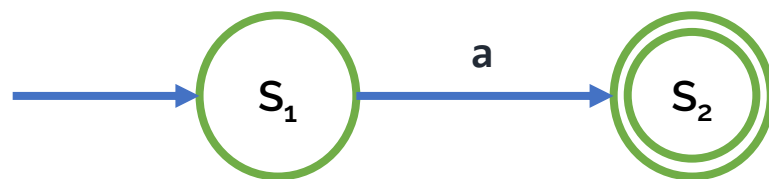


# Implementation of scanner

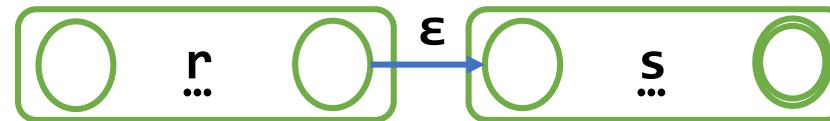


# Regular Expression to NFA

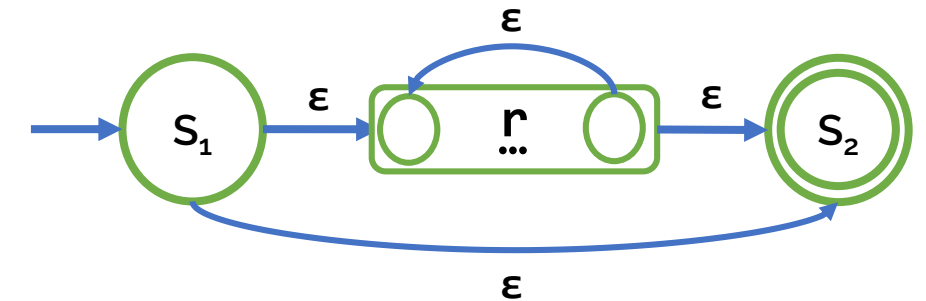
- Thompson's Construction method
- 4 construction cases:



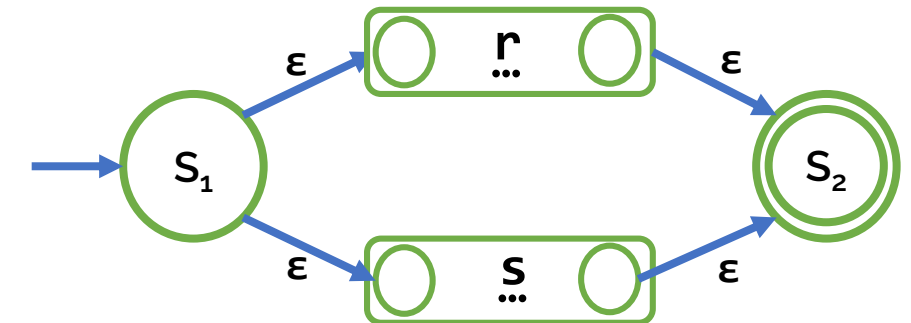
Basic Regular Expression



Concatenation



Repetition



Choice Among Alternatives



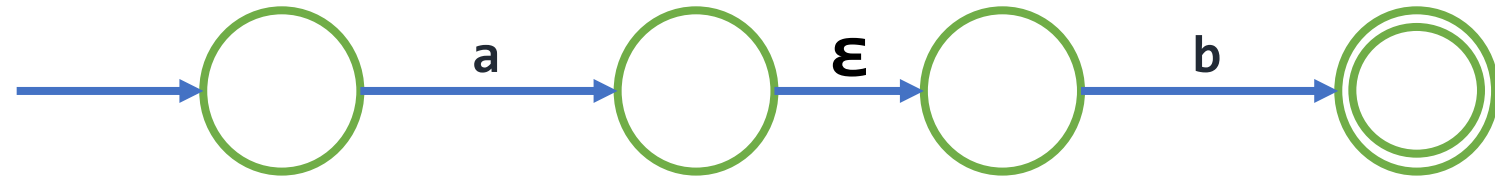
# Example

$ab \mid a$

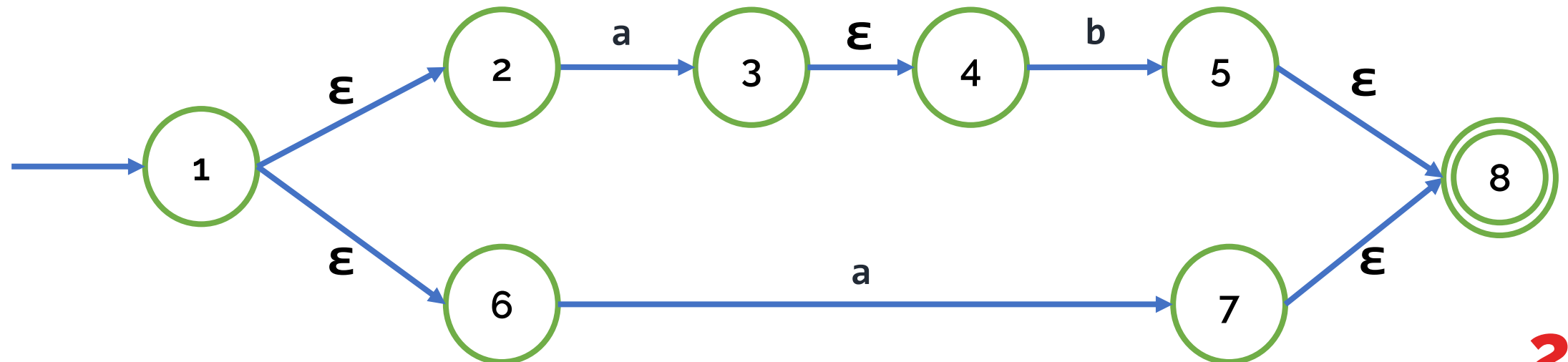
## Basic Regular Expressions



## Concatenation



## Choice Among Alternatives



# NFA to DFA

## The Subset Construction Method:

1. Compute  **$\epsilon$ -closure** of start state of NFA, this becomes the start state of DFA
2. For this set, and each subsequent set , compute transitions on character **a** in the alphabet:

$$S'_a = \{t \mid \text{for some } s \text{ in } S \text{ there is a transition from } s \text{ to } t \text{ on } a\}$$





# NFA to DFA

## The Subset Construction Method:

3. Compute  $\overline{S'_a}$ , this defines a new state and transition in DFA

$$S \xrightarrow{a} \overline{S'_a}$$

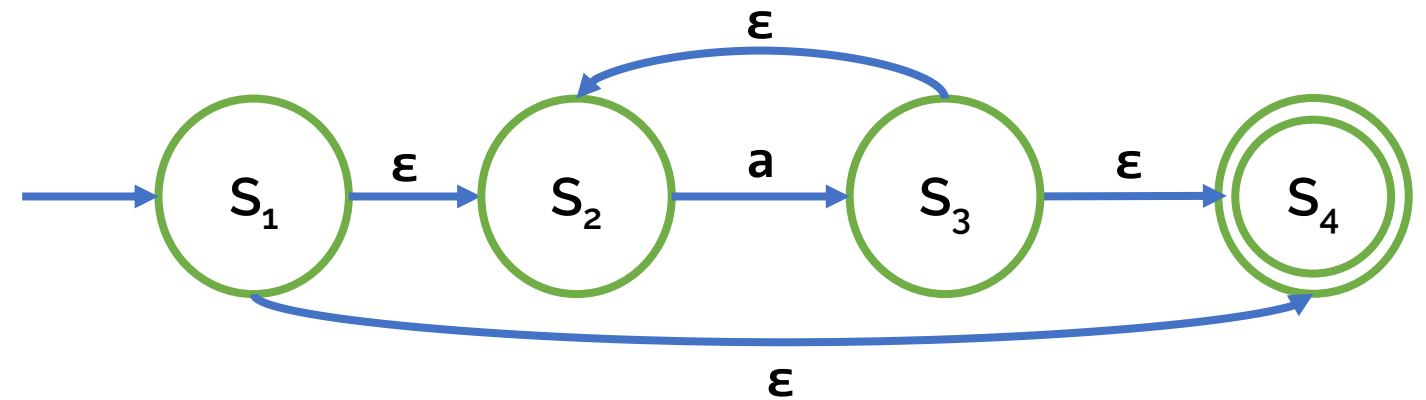
4. Continue this process until no new states or transitions are created

5. States that include accepting the state of NFA will be accepting states in DFA



# The $\epsilon$ -closure of a set of states

- The set of states reachable by a series of zero or more  $\epsilon$ -transitions



$$\overline{S1} = \{S1, S2, S4\}$$

$$\overline{S2} = \{S2\}$$

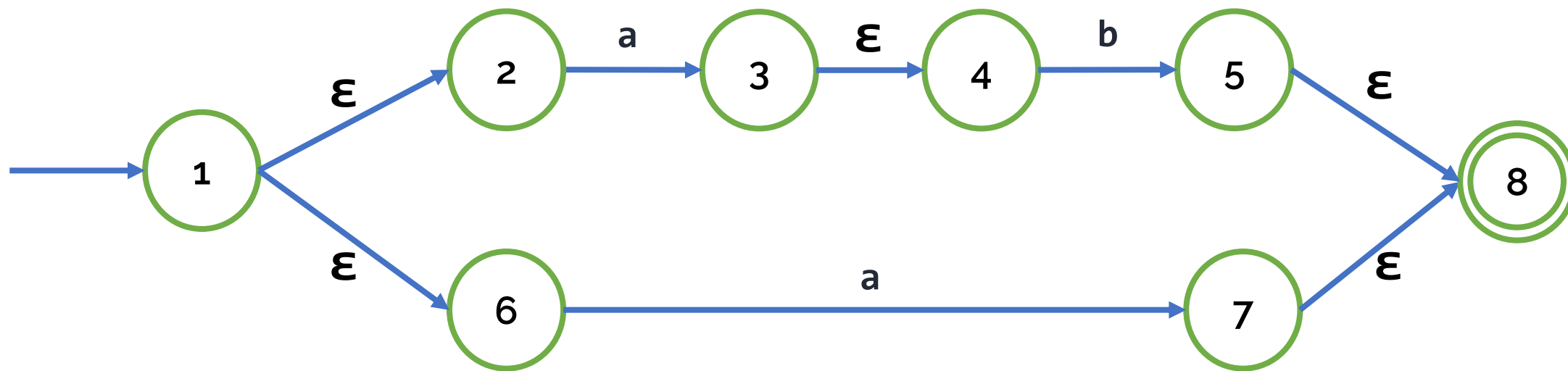
$$\overline{S3} = \{S2, S3, S4\}$$

$$\overline{S4} = \{S4\}$$



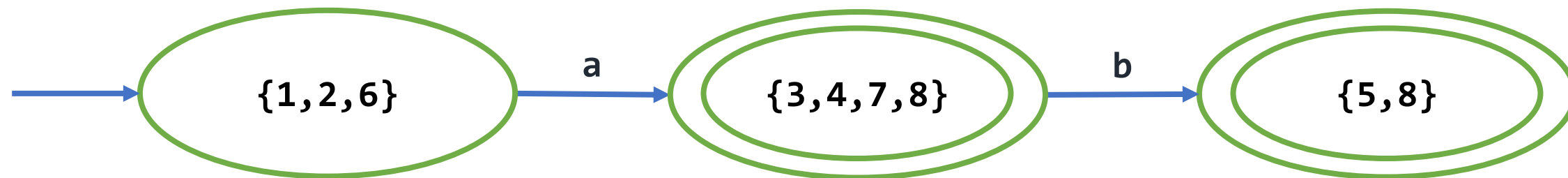
# Example

## NFA



# Example

**DFA**



# DFA to minimum-state DFA

- Create two sets, one including all the accepting states and one with non-accepting states
- Consider the transitions on each character **a** of the alphabet, if all accepting states have transitions on **a** to accepting states, then this defines an **a-transition** from the new accepting state to itself.
- If all accepting states have transitions on **a** to non-accepting states, then this defines an **a-transition** from the new accepting state to the new non-accepting state.



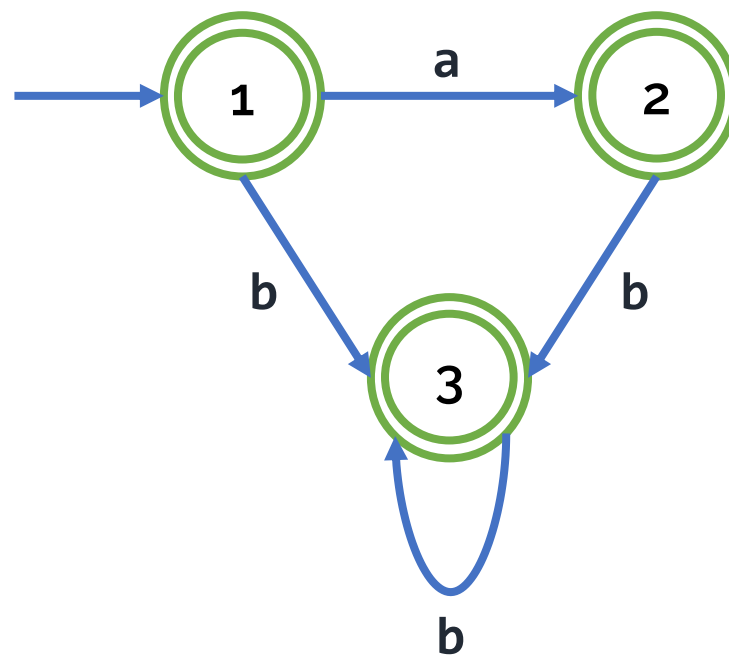
# DFA to minimum-state DFA

- If there are two states **s** and **t** that have transitions on **a** that land in different states, then **a-distinguishes** the states **s** and **t**. In this case, the set of states must be split according to where their **a-transitions** land
- Repeat this process until no further split happens or each state only consists of one element

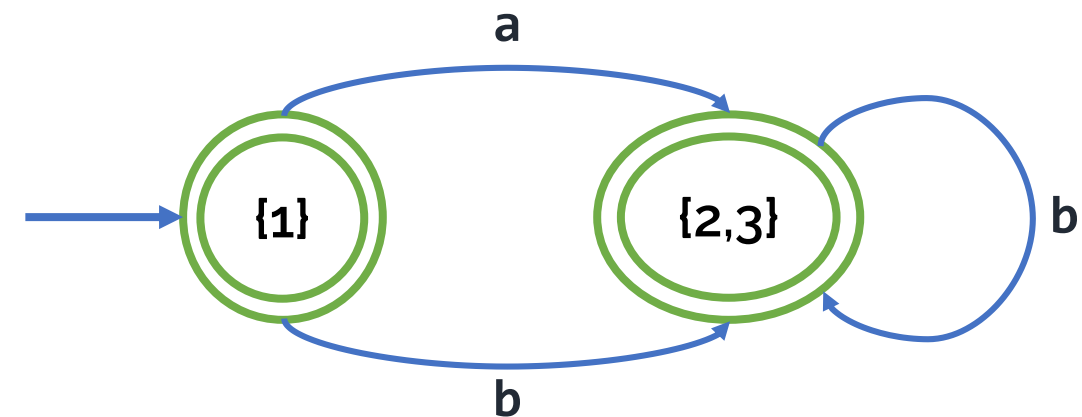


# Example

**DFA**



**minimum-state DFA**



# Assignment 1

- 6 written questions (10 marks)
  - Please do all of them. They are easy.
- Phase 1 of the project (30 marks)
  - Using Flex to generate the scanner module
  - Next 3 tutorials will teach to how to use Flex
  - You have to use Flex (not other Lex tools)

