

CYDF315 - Compiler

Lab 1: NFA to DFA



Objective

- The goal of labs 1 and 2 is to implement the subset construction algorithm (NFA to DFA).
- The goal of lab1 is to implement ϵ -closure and move functions

$$\epsilon\text{-closure}(s) = \{s\} \cup \{t \mid s \rightarrow_{\epsilon} \dots \rightarrow_{\epsilon} t\}$$

$$\epsilon\text{-closure}(T) = \bigcup_{s \in T} \epsilon\text{-closure}(s)$$

$$\text{move}(T, a) = \{t \mid s \rightarrow_a t \text{ and } s \in T\}$$

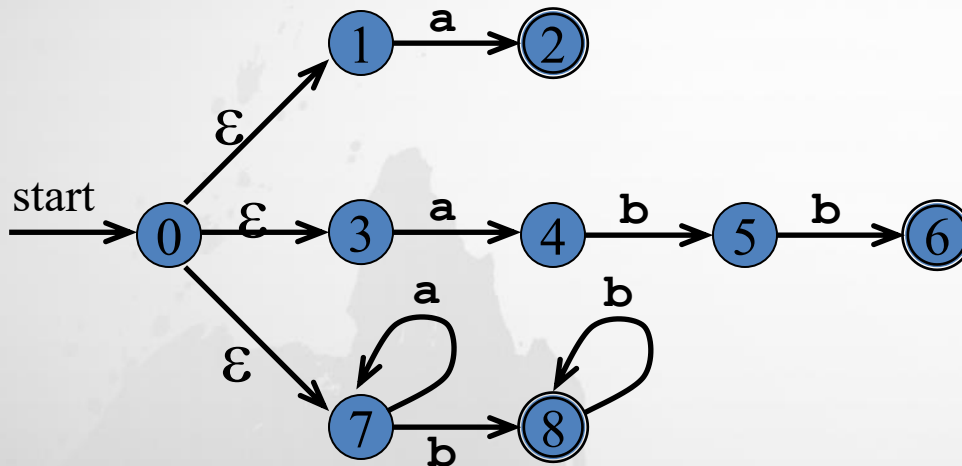
Getting Started

- Download the template file (lab1.py) and the input file (input.txt) from Blackboard

Description of Closure

- $\varepsilon\text{-closure}(s) = \{s\} \cup \{t \mid s \rightarrow_{\varepsilon} \dots \rightarrow_{\varepsilon} t\}$
 - A set of NFA states that can be reached from s only by ε transitions
- $\varepsilon\text{-closure}(T) = \bigcup_{s \in T} \varepsilon\text{-closure}(s)$
 - A set of NFA states that can be reached from the states in T only by ε transitions
- $\text{move}(T, a) = \{t \mid s \rightarrow_a t \text{ and } s \in T\}$
 - A set of NFA states that can be reached from the states in T by input a

ϵ -closure and move Examples



$$\epsilon\text{-closure}(\{0\}) = \{0, 1, 3, 7\}$$

$$\text{move}(\{0, 1, 3, 7\}, \mathbf{a}) = \{2, 4, 7\}$$

$$\epsilon\text{-closure}(\{2, 4, 7\}) = \{2, 4, 7\}$$

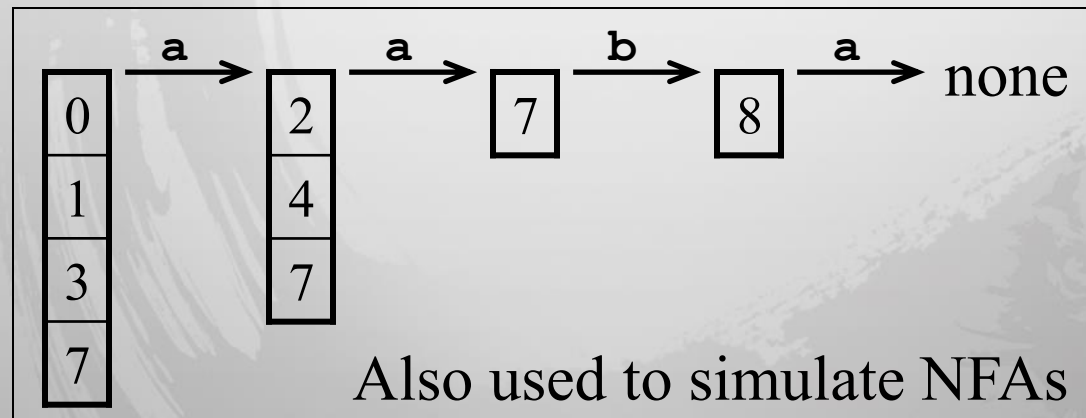
$$\text{move}(\{2, 4, 7\}, \mathbf{a}) = \{7\}$$

$$\epsilon\text{-closure}(\{7\}) = \{7\}$$

$$\text{move}(\{7\}, \mathbf{b}) = \{8\}$$

$$\epsilon\text{-closure}(\{8\}) = \{8\}$$

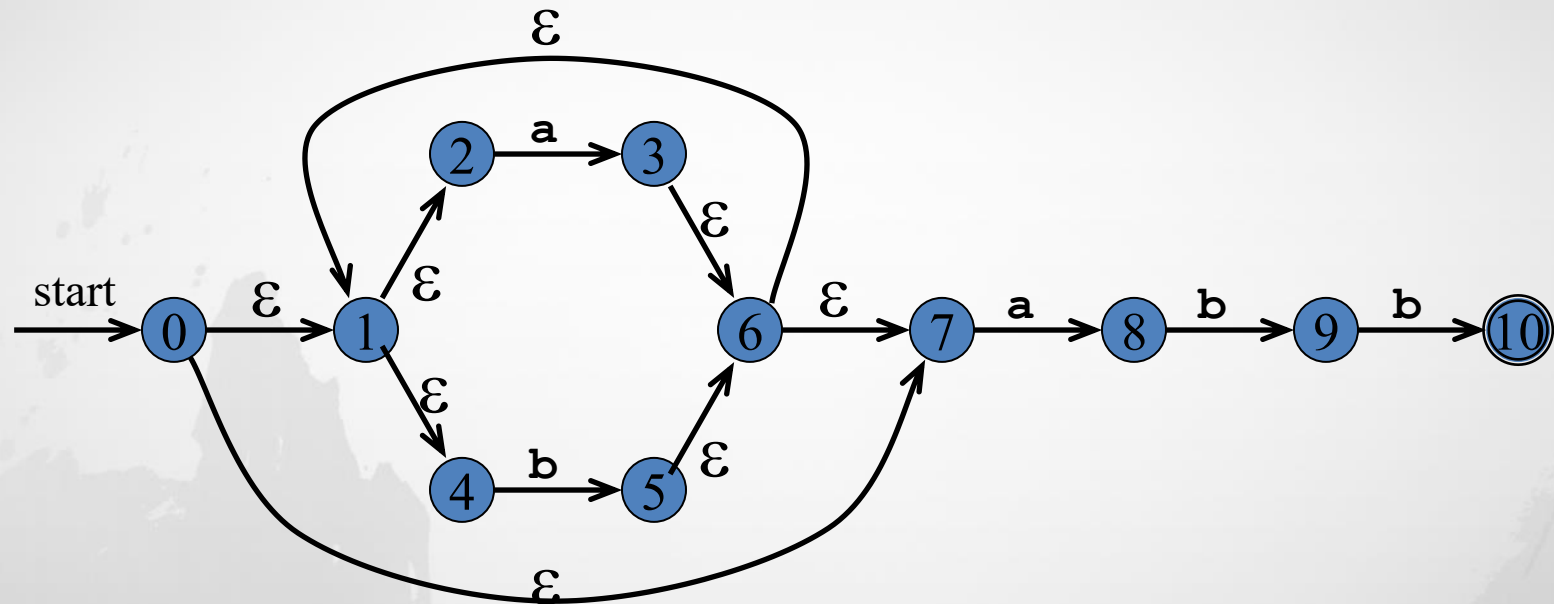
$$\text{move}(\{8\}, \mathbf{a}) = \emptyset$$



Given Classes

- Edge
 - Represents an edge in the graph by (from, to, input)
- Graph
 - Represents the graph by a collection of edges
 - Edges are stored in a list
 - Read an input file
 - Each line of the input file represents an edge
 - The format of one line is "from,to,input"
 - The input is optional. If it is not present, it means an empty string

Test Input



Lab Assignment

- Implement the following three functions of the Graph class (marked with #TODO)

```
getEClosures(self, s_)
```

```
getEClosureT(self, t_)
```

```
getMove(self, t_, a_)
```


Expected Results

```
$ python main.py input.txt  
{0, 1, 2, 4, 7}  
{8, 3}  
{1, 2, 3, 4, 6, 7, 8}  
{5}  
{1, 2, 4, 5, 6, 7}
```