Lexical Analysis
Programming Languages

Sujit Kumar Chakrabarti

IIITB

Non-Deterministic FSA (NFA)

- Finite set of states – (S)
- Alphabet - $(\sum)$
- Transition function $(T : S \times \sum \rightarrow 2^S)$
- Initial state (S_0)
- Final/accepting states $(F \subseteq S)$

Non-Deterministic FSA (NFA)

- Finite set of states – (S)
- Alphabet - $(\sum)$
- Transition function $(T : S \times \sum \to 2^S)$
- Initial state (S_0)
- Final/accepting states $(F \subseteq S)$
- **Acceptance of a string:** When there exists a path corresponding to the input leading to an accepting state.

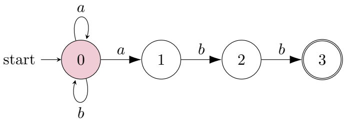
Simulation of NFAs

Salient Points

- Possibly more than one outgoing transitions with the same label.
- ϵ-transitions
- More than one paths can be traced during the same run.
- All the possible traces have to be tracked.
- Multiple states can be active at the same time.

Simulation of NFAs

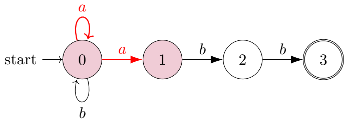
Example 1

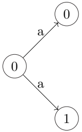


Input: *aabbabb*

Simulation of NFAs

Example 1

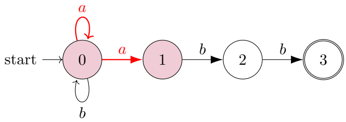


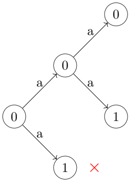


Input: **a***abbabb*

Simulation of NFAs

Example 1

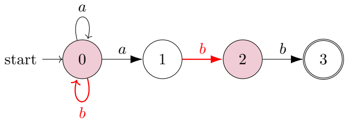


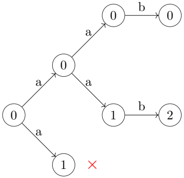


Input: *aabbabb*

Simulation of NFAs

Example 1

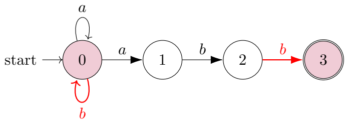


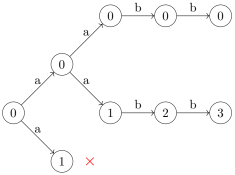


Input: $aa\textbf{b}babb$

Simulation of NFAs

Example 1

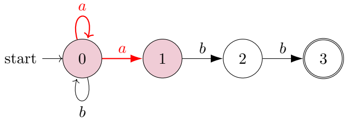


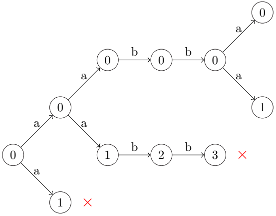


Input: $aabbabb$

Simulation of NFAs

Example 1

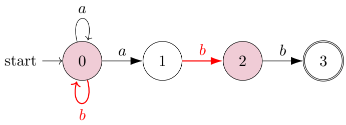


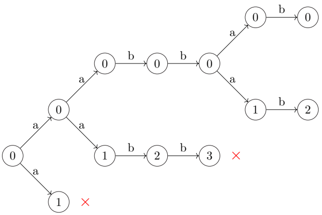


Input: $aabb\mathbf{a}bb$

Simulation of NFAs

Example 1

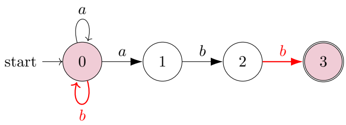


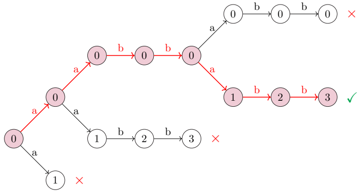


Input: *aabbabb*

Simulation of NFAs

Example 1

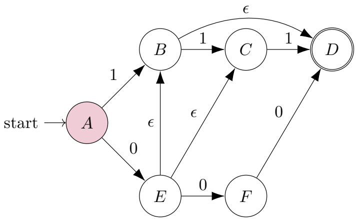




Input: $aabbab$**b**

Simulation of NFAs

Example 2.1

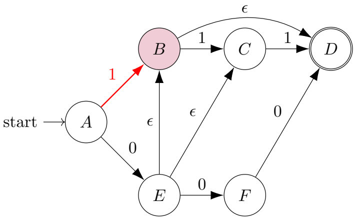


Input: 1...

Simulation of NFAs

ϵ-closure

Example 2.1

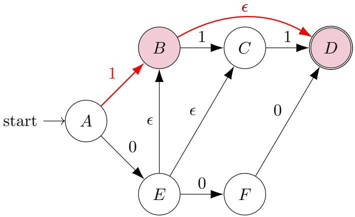


Input: 1...

Simulation of NFAs

ϵ-closure

Example 2.1

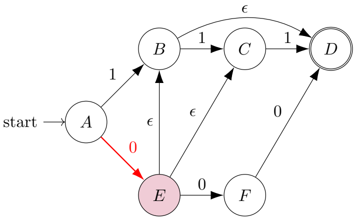


Input: 1...

Simulation of NFAs

ϵ-closure

Example 2.2

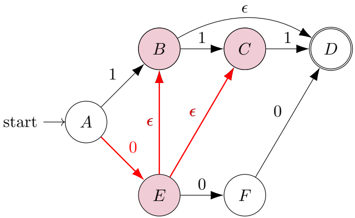


Input: 0...

Simulation of NFAs

ϵ-closure

Example 2.2

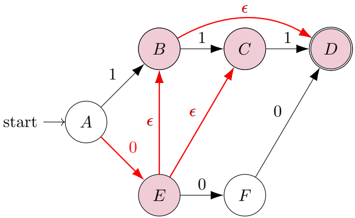


Input: 0...

Simulation of NFAs

ϵ-closure

Example 2.2



Input: 0...

Simulation of NFAs

ϵ-closure

- ϵ-closure: computed on a set of states
- Transitive closure of all states reachable through ϵ-transitions
- From a source state set S_1, on an input symbol a, the destination state set S_2 is computed as:

$$U = \bigcup_{s \in S_1} Trans[s, a]$$
$$S_2 = \epsilon\text{-}closure(U)$$

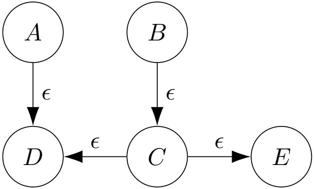
- ϵ-closure – a reflexive relation

Computing ϵ-closure

procedure ϵ-CLOSURE(s)
    $stack$.PUSH(s)
    ep.ADD(s)
    while $stack$ is not empty **do**
        $t \leftarrow stack$.POP
        $U \leftarrow \{u : u \in M.Trans[t, \epsilon]\}$
        for $u \in U$ **do**
            if $u \notin ep$ **then**
                $stack$.PUSH(u)
                ep.ADD(u)
            end if
        end for
    end while
    return ep
end procedure

Computing ϵ-closure

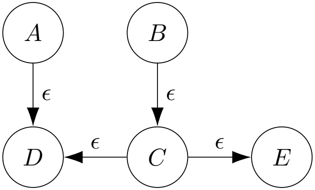
Example



ep	*stack*

Computing ϵ-closure

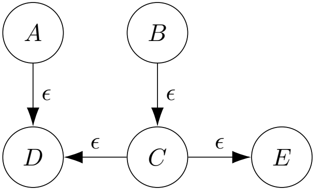
Example



ep	*stack*
$A,\ B$	$B,\ A$

Computing ϵ-closure

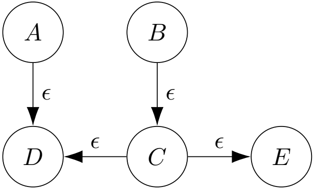
Example



ep	*stack*
$A,\ B$	$B,\ A$
$A,\ B,\ D$	$B,\ D$

Computing ϵ-closure

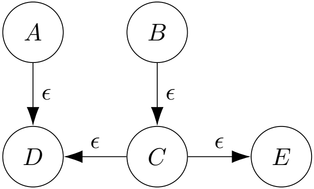
Example



ep	$stack$
$A,\ B$	$B,\ A$
$A,\ B,\ D$	$B,\ D$
$A,\ B,\ D$	B

Computing ϵ-closure

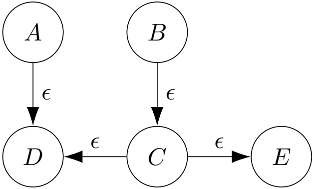
Example



ep	*stack*
A, B	B, A
A, B, D	B, D
A, B, D	B
A, B, D, C	C

Computing ϵ-closure

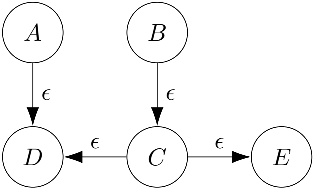
Example



ep	*stack*
A, B	B, A
A, B, D	B, D
A, B, D	B
A, B, D, C	C
A, B, D, C, E	E

Computing ϵ-closure

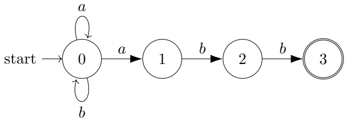
Example



ep	*stack*
A, B	B, A
A, B, D	B, D
A, B, D	B
A, B, D, C	C
A, B, D, C, E	E
A, B, D, C, E	

Simulating FSAs

Representating transition function using transition tables

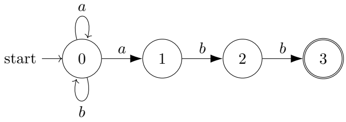


Transition Table:

State	a	b
0		
1		
2		
3		

Simulating FSAs

Representating transition function using transition tables



Transition Table:

State	a	b
0	{0, 1}	{0}
1	{}	{2}
2	{}	{3}
3	{}	{}

Simulation of NFA

procedure SIMNFA(N, inp)

Simulation of NFA

```
procedure simNFA(N, inp)
    S ← ϵ-closure({N.s_0})
    while there is input left do
        c ← nextChar
        T′ ← move(S, c)
        S ← ϵ-closure(T′)
    end while
    if S ∩ N.F ≠ {} then
        return true
    else
        return false
    end if
end procedure
```

Simulation of NFA

procedure SIMNFA(N, inp)
    $S \leftarrow \epsilon$-CLOSURE($\{N.s_0\}$)
    while there is input left **do**
        $c \leftarrow$ NEXTCHAR
        $T' \leftarrow$ MOVE(S, c)
        $S \leftarrow \epsilon$-CLOSURE(T')
    end while
    if $S \cap N.F \neq \{\}$ **then**
        return true
    else
        return false
    end if
end procedure

Next

Conversion of NFA to DFA