Deterministic Finite Automata (DFA)

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

Q = a finite set (of states)

Σ = a finite (input alphabet) set

δ = the transition function (full function) where :

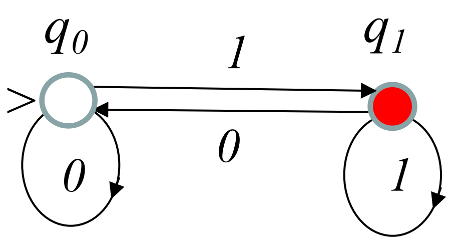
$$\delta : Q \times \Sigma \rightarrow Q ; (q, \sigma) \rightarrow \delta(q,\sigma) \in Q$$

q_0 = initial state, $q_0 \in Q$

F = final state set , $F \subseteq Q$

*Simple Representations of **DFA***

(1) Visual (Graphical) : Transition Diagrams



strings (event sequences) that end up in

colored (final) state

(2) Tabular : Transition Tables

state	*input*	*state'*
q_0	0	q_0
q_0	1	q_1
q_1	0	q_0
q_1	1	q_1

no. of columns in transition table = 3

in general how many rows are there ?

answer $\rightarrow |\Sigma| \times |Q|$ ***rows***

δE = Extended Transition Function

$\delta E : Q \times \Sigma^* \rightarrow Q \,;\, (q, s) \rightarrow \delta E(q,s) \in Q$

*Inductive Definition (**e** =empty string)*

$\delta E(q, e) := q$, *Basis*

$\delta E(q\,,\, s.a) = \delta(\delta E(q, s)\,,\, a)$, *Induction*

Complement Language

L(A) := *the language **accepted** by **A***

$s \in L(A) \leftrightarrow$ (*if and only if*) $\delta E(q_0, s) \in F$; **or :**

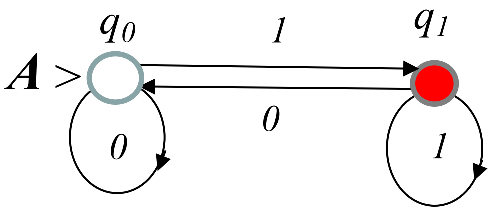
Non-final states

$L(A) = (s \in \Sigma^* \mid \delta E(q_0\,,\, s) \in F\,)$

$\Sigma^* - L(A) = (s \in \Sigma^* \mid \delta E(q_0\,,\, s) \in Q\text{-}F\,)$

Note that ***'-'*** *sign is used for* ***set subtraction*** *hence for sets* ***X*** *and* ***Y*** *:* $X\text{-}Y := X \cap Y^c$

Examples **1-** *Describe in simple natural language* ***L(A)*** *= the language accepted by* ***A***

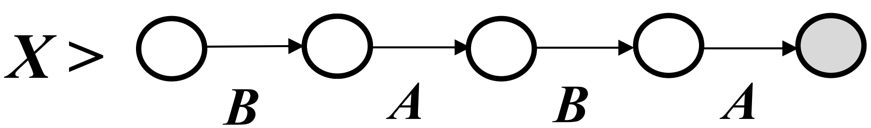


$$L(A) = (s \in \{0,1\}^* \mid \delta E(q_0 , s) \in \{q_1\})$$

Answer : all strings in {0,1}* that terminate with a 1

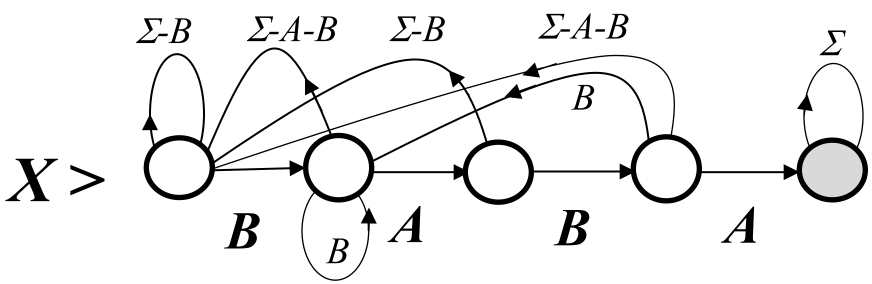
2- *Design a DFA* ***X*** *that accepts the string of letters in Turkish alphabet in which the substring* ***'BABA'*** *occurs at least once !*

Let $\boldsymbol{\Sigma}$ *denote the set of all capital letters in the Turkish alphabet*



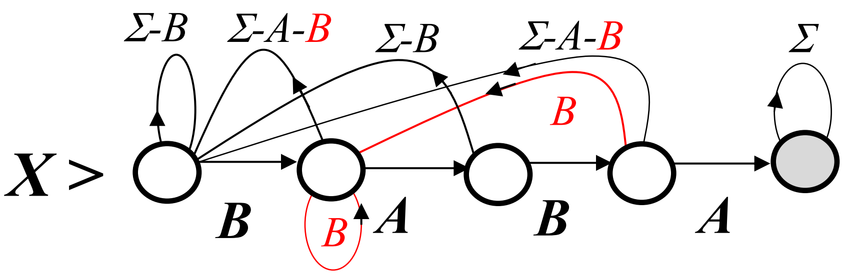
NOT *a DFA ! ! Why ?*

δ ***NOT*** *a full function ; missing transitions !*



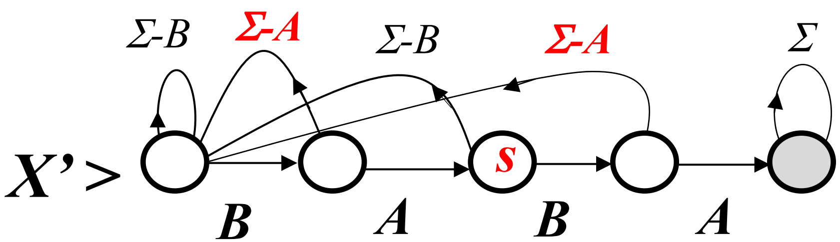
This is a DFA !

Discussion slide on Example 2



Is this or the next DFA

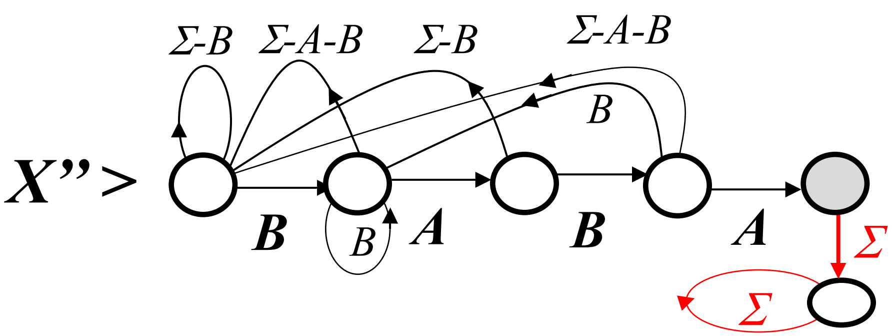
the true solution ?



Does not accept strings

'BBABA'** or **; 'BABBABA'

*Both strings get stuck in state **s***
which is not a final state



*What language does **X"** accept ?*

*All strings where **'BABA'** occurs as a substring*

precisely once** and only as a **postfix

Nondeterministic Finite Automata (NFA)

Same as ***DFA*** *except :*

set of all subsets of Q

(1) $\delta : Q \times \Sigma \to 2^Q$ (where $2^Q := P(Q)$ = *power set of* Q)

(2) *initial state (is a set !)* $Q_0 \subseteq Q$ *(differs from main text !)*

Distinction in graphical representation (transition diagram) :

In ***DFA*** *for every* $\sigma \in \Sigma$ *there is* ***exactly one*** *outgoing transition edge from every state* $q \in Q$

In ***NFA*** *for every* $\sigma \in \Sigma$ *there may be* ***multiple*** *(including* ***none*** *!) outgoing transition edges from every state* $q \in Q$

Extended Transition Function for NFA

$\delta E : 2^Q \times \Sigma^* \rightarrow 2^Q ; (X, s) \rightarrow \delta E(X,s) \in 2^Q$

Inductive Definition

$\delta E(X , e) := X$, *Basis*

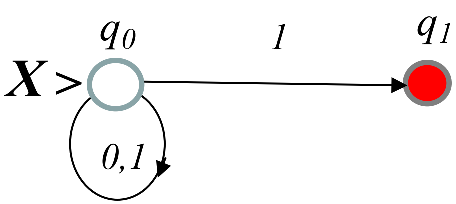
$\delta E(X, s.a) = \cup_{q \in \delta E(X, s)} \delta(q , a)$, *Induction*

L(A) := the language ***accepted*** by ***A***

$s \in L(A) \leftrightarrow$ (if and only if) $\delta E\big(Q_0, s) \cap F \neq \varnothing$; *or* :

$L(A) := \{ s \in \Sigma^* \mid \delta E\big(Q_0, s) \cap F \neq \varnothing \}$; $\varnothing :=$ *null set*

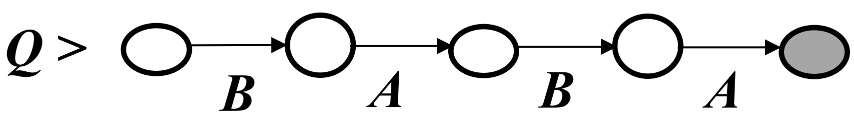
Examples 1 - *Describe in simple natural language* $L(X)$ = *the language accepted by* X



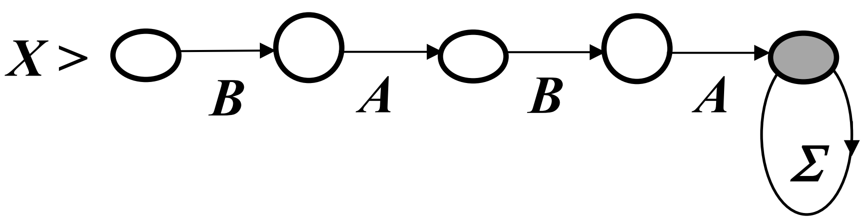
$$L(X) = (s \in \{0,1\}^* \mid \delta E(q_0, s) \cap \{q_1\} \neq \varnothing)$$

Answer : all strings in {0,1}* that terminate with a 1

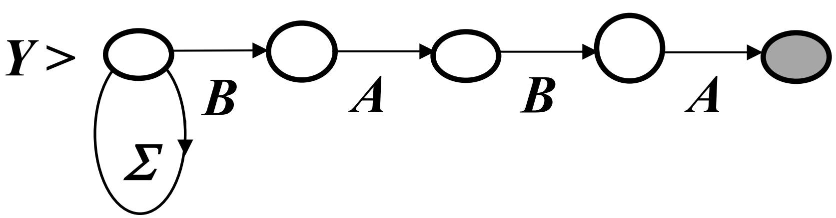
2 - *Design an NFA* ***W*** *that accepts only the string* ***BABA*.**



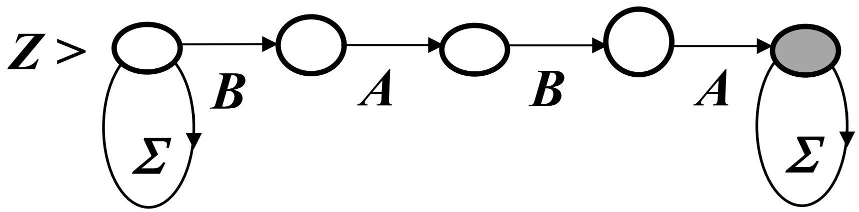
3 - *Design an NFA* ***X*** *that accepts* ***all strings*** *in which* ***BABA*** *is a* ***prefix*.**



4 - *Design an NFA* ***Y*** *that accepts* ***all strings*** *in which* ***BABA*** *is a* ***postfix*.**



*5 - Design an NFA **Z** that accepts **all strings** in which **BABA** is a **substring .***



*6 - Design NFAs that accepts **all strings** in which **BABA** is **NOT** a : (i) **prefix,** (ii) **postfix** (iii) **substring.***

*Convert respectively **X,Y, Z** into their DFA equivalents (see slides 11+etc)*

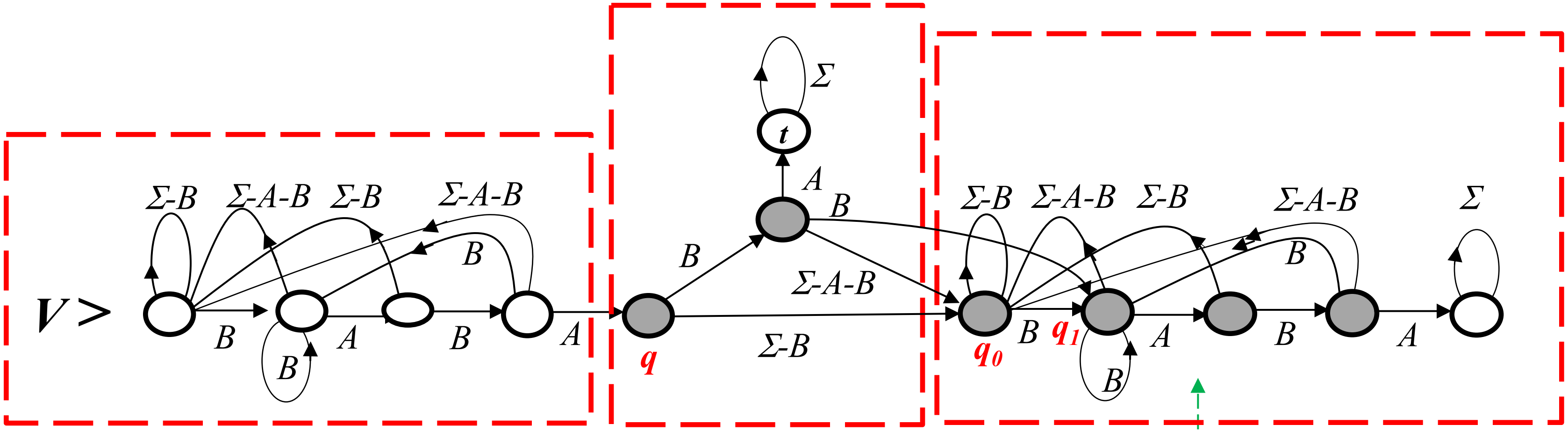
*and convert each into its corresponding **complement** DFA by interchanging its final*

and nonfinal state sets .

*7 - Design an NFA **V** in which **BABA** occurs as a **substring precisely once.***

(See next slide)

Solution to example 7 on slide no. 9 is automaton *V* below



The automaton that generates ALL strings in which the substring 'BABA' occurs precisely once as a postfix upon arrival at state q

__ALL__ sequences of length __1__ or __2__ that differ from __BA__ reach from q to q_0 or q_1. To avoid __BABA__ through a second __BA__ sequence,a trap state __t__ is placed.

The automaton starting at initial state q_0 accepts ALL strings that do NOT have the substring __'BABA'__ in it

FACT** : If a DFA **X** accepts the language **L(X)** then the DFA that accepts the complement language **Σ*-L(X)** is same as **X** except **F** is replaced with **Q-F

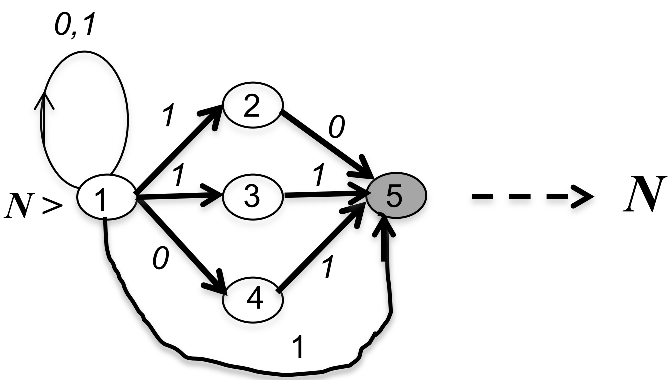
Construction of Equivalent DFA D from a given NFA N

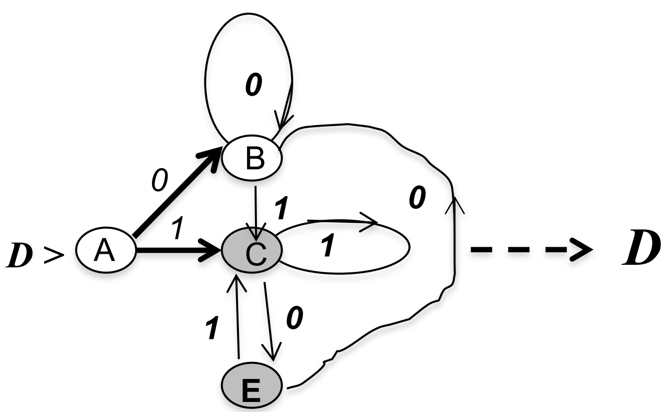
Problem *: Given an NFA* $\boldsymbol{N = (Q, \Sigma, \delta_N, Q_0, F_N)}$ *construct a DFA* $\boldsymbol{D = (2^Q, \Sigma, \delta_D, Q_0, F_D)}$ *such that* $\boldsymbol{L(N) = L(D)}$

Idea : *Construct DFA **D** where every state of **D** is a **subset** of states of **N ;** and let the **subset-to-subset transitions** of **N** be the simple **state-to-state transitions** in **D.** Whenever a final state in* $\boldsymbol{F_N}$ *is visited in **N** by a string **s** then by letting* $\boldsymbol{F_D}$ *be the **set of all subsets** of **Q** where at least one state of an **element of*** $\boldsymbol{F_D}$ *is in* $\boldsymbol{F_N}$ *it is also accepted by **D** by definition of **N** .*

*Example for DFA equivalent **D** for an NFA **N***

L=(s∈ {0,1}* | s = u.v ; |v| ≤ 2 ; v** has at least one **1 ; u,v∈ {0,1}* }



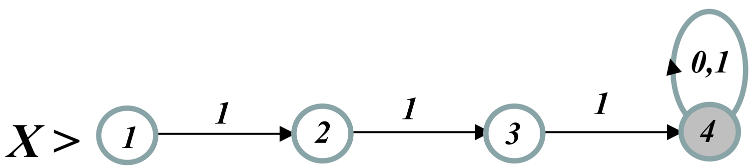


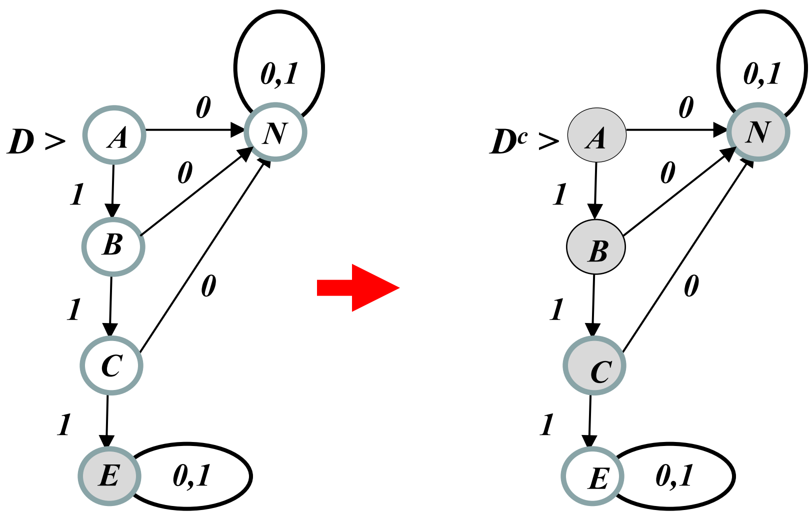
	state	input	next
> A →	*1*	*0*	*1,4* ***(B)***
	1	*1*	*1,2,3,5* ***(C)***
B →	*1,4*	*0*	*1,4* ***(B)***
	1,4	*1*	*1,2,3,5* ***(C)***
final C →	*1,2,3,5*	*0*	*1,4,5* ***(E)***
	1,2,3,5	*1*	*1,2,3,5* ***(C)***
final E →	*1,4,5*	*0*	*1,4* ***(B)***
	1,4,5	*1*	*1,2,3,5* ***(C)***

*Another example for DFA equivalent **D** for an NFA **X***

***L**=(s ∈ {0,1}* | s does **NOT** have a prefix **1.1.1**)*

***L^c**=(s ∈ {0,1}* | s = 1.1.1.v ; v ∈ {0,1}*)*





q	σ	q’
X > 1 =A	0	N=Null
1	1	2=B
2 =B	0	N
B	1	3=C
3=C	0	N
C	1	4=E*
4 = E*	0	E*
E*	1	E*
N	0	N
N	1	N

Theory

Problem : Given an NFA $N = (Q, \Sigma, \delta_N, Q_0, F_N)$ construct a DFA $D = (2^Q, \Sigma, \delta_D, Q_0, F_D)$ such that $L(N) = L(D)$

Solution :

(1) $\delta_D(X,\sigma) := \delta_N E(X,\sigma)$; $\delta_D(\varnothing, \sigma) := \varnothing, \ \forall \sigma \in \Sigma$

(2) $F_D := (Y \subseteq Q \mid Y \cap F_N \neq \varnothing)$

To prove that $L(D) = L(N)$ first show that $\delta_D E(Q_0,u) = \delta_N E(Q_0,u)$ using induction on the length of u .

$\delta_D E(Q_0,e) = \delta_N E(Q_0,e) = Q_0$ by definition **(basis ; s=e** case)

$\delta_D E(Q_0, s.a) = \delta_D(\delta_D E(Q_0, s), a) = \delta_D(X, a) = \delta_N E(X,a)$ where $X = \delta_D E(Q_0, s)$

But by induction hypothesis : $\delta_D E(Q_0,s) = \delta_N E(Q_0,s) = X$; hence

$\delta_D E(Q_0, s.a) = \delta_D(X, a) = \delta_N E(X,a) = \delta_N E(Q_0, s.a)$; by **def.** of $\delta_N E(Q_0,s.a)$

Finally **L(N)=L(D)** *is proved as follows :*

$s \in L(N) \Leftrightarrow \delta_N E(Q_0, s) \cap F_N \neq \varnothing$; *by def. of* **L(N)**

$\Leftrightarrow \delta_N E(Q_0, s) \in F_D$; *since* $F_D := (Y \subseteq Q \mid Y \cap F_N \neq \varnothing)$

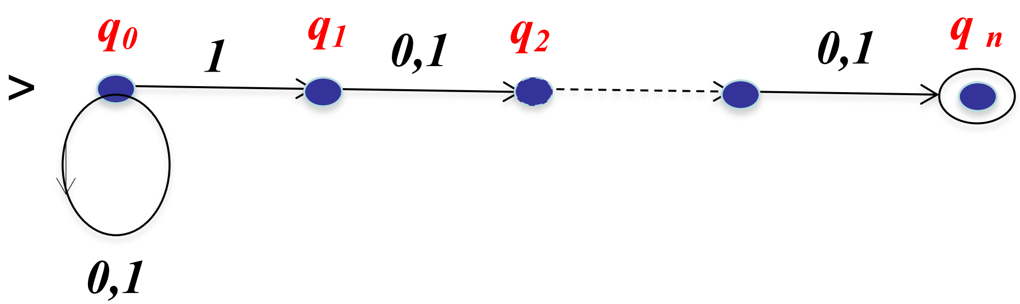
$\Leftrightarrow \delta_D E(Q_0, s) \in F_D$; *since* $\delta_D E(Q_0, s) = \delta_N E(Q_0, s)$

$\Leftrightarrow s \in L(D)$; *by def. of* **L(D)**

A ‘bad case’ example for NFA-to-DFA conversion

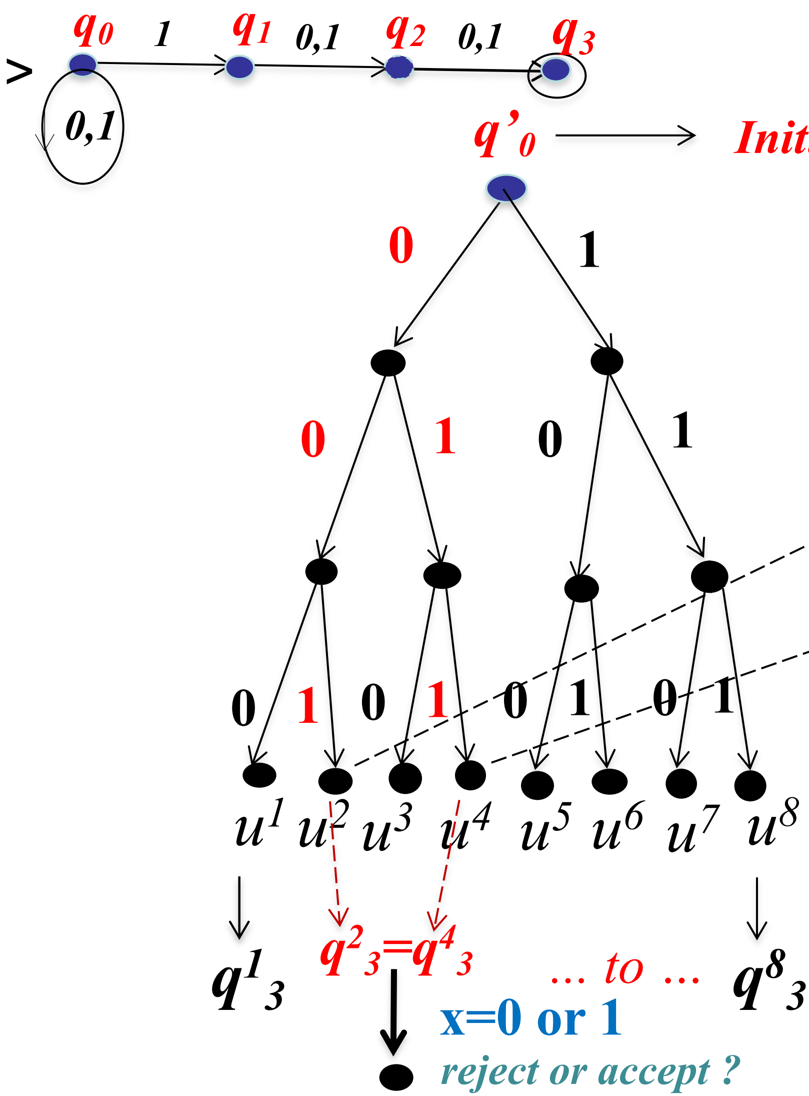
$L = (s \in \{0,1\}^* \mid s=u.1.v\ ;\ |v|=n-1\ ,\ n>1\ ,\ u,v \in \{0,1\}^*\ )$

An ***n+1*** *state NFA to accept* ***L***



Fact **:** *Any DFA* ***D*** *to accept* ***L*** *has at least* $\mathbf{2^n}$ *states*

A special case : $n=3$



Initial state of a DFA equivalent

Suppose not !!

say $q^2_3 = q^4_3$ *driven by*

u^2 = *001* *and* u^4 = *011* *respectively*

But that is impossible since inputs ***001x*** *(to be rejected) and* ***011x*** *(to be accepted)* ***are*** *either both accepted or both rejected violating the design !*

must all be 8 distinct states !

Proof of Fact

*(1) Consider all (2^n) sequences of **0** and **1**s of length **n ;** denote each by u^k for $k=1,\ldots,2^n$ and **j**th input of u^k by u_j^k for $j=1,\ldots,n$.*

*(2) Apply each sequence u^k starting from the initial state q'_0 of **D** and let q_n^k be the state of **D** arrived at the end of the application of u^k.*

Claim $k \neq p$ implies $q_n^k \neq q_n^p$!

*(3) Suppose the claim is false for some $k \neq p$ (i.e. $q_n^k = q_n^p$!) then let **j** be the first (smallest) index for which $u_j^k = 1$ and $u_j^p = 0$*

*(4) Then after **n-j** steps the corresponding states **merge** at the same value $q_n^k = q_n^p$*

*(5) But then it becomes impossible to differentiate inputs of length **n+j** starting with u^k and u^p although at **j**th stage one continues with **1** (to be accepted by **D**) and the other with **0** (to be rejected by **D**) ! **A contradiction !***

NFA with ε-transitions

$N\varepsilon = (Q, \Sigma, \delta_{N\varepsilon}, Q_0, F)$

Difference is in $\delta_{N\varepsilon} : Q \times (\Sigma \cup \varepsilon) \to 2^Q$

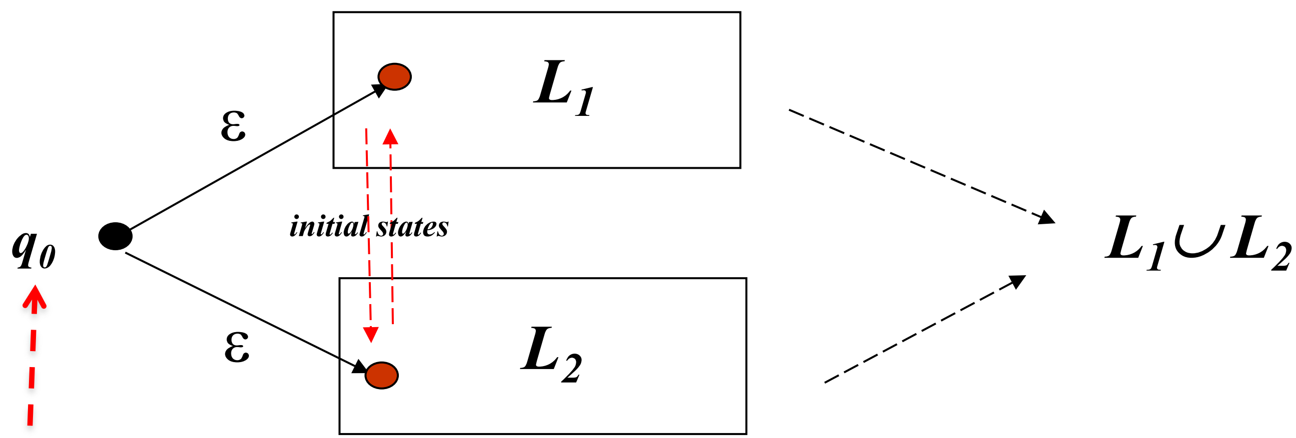
$\delta_{N\varepsilon}(q, \varepsilon) \in 2^Q$ is called *(a bundle of)* ***ε-transitions***

In computing the language accepted, ***L(Nε)****, ε-transitions*

do not count , i.e., they are defined as invisible and erased !

Typical Applications of ε - transitions

1- Implementation of the union language $L_1 \cup L_2$



A single initial state
connecting to all initial states

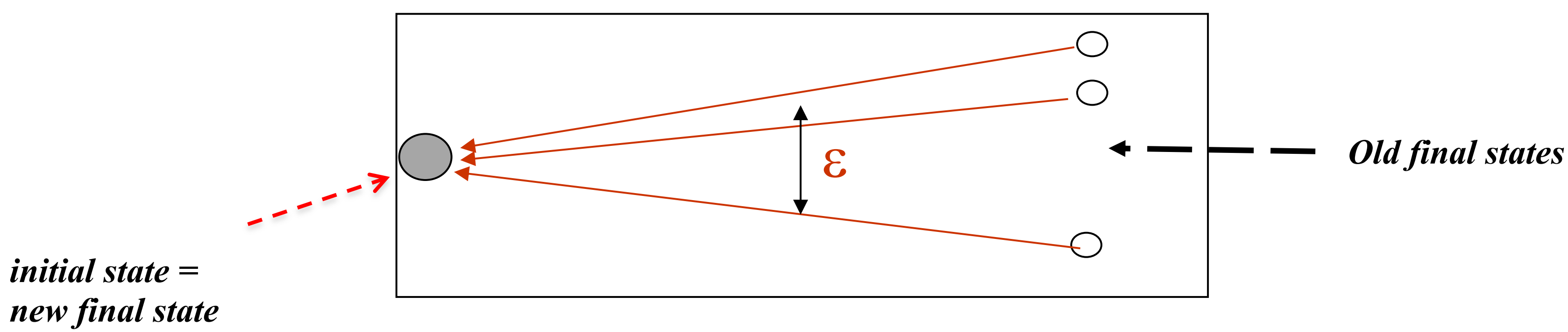
2- Implementation of the star or Kleene-closure language L* given L

What is L language ?*

concatenation of strings

$L^* := (s \in \Sigma^* \mid s = u_1 . u_2 . \ldots . u_k \;;\; u_j \in L , j = 1,\ldots,k \;;\; k \geq 0)$

Note that $s = e \in L^*$ *(k=0 case above)*



L to L*

Eliminating ε-transitions

Idea : define ε-closures inductively (recursively)

Let $\boldsymbol{X} \subseteq \boldsymbol{Q}$ *and compute* $\boldsymbol{ECLOSE(X)} \subseteq \boldsymbol{Q}$ *recursively as below :*

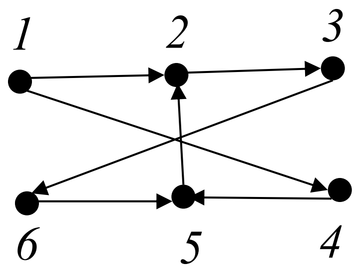
$\boldsymbol{ECLOSE\ (X) = X}$, ***basis***

If $\boldsymbol{y} \in \boldsymbol{ECLOSE(X)}$ *then set :*

$\boldsymbol{ECLOSE(X) := ECLOSE(X)} \cup \delta_{N\varepsilon}(\boldsymbol{y}, \varepsilon)$, ***recursion***

Example for computing E-closures

All transitions are epsilon-transitions



Progress in inductive steps →

E-CLOSURE (1) → (1) → (1,2,4) → (1,2,4,3,5) → (1,2,4,3,5,6)

E-CLOSURE (4) → (4) → (4,5) → (4,5,2) → (4,5,2,3) → (4,5,2,3,6)

The language $L(N\varepsilon)$ accepted by an automaton $N\varepsilon$ with ε-transitions

Extended Transition Function for $N\varepsilon$ **:**

$\delta_{N\varepsilon}E(X, e) := ECLOSE(X)$; **basis**

$\delta_{N\varepsilon}E(X, s.a) := \cup_{y \in Y} ECLOSE\,(\delta_{N\varepsilon}(y, a)),\ Y = \delta_{N\varepsilon}E(X, s)$: **induction**

$L(N\varepsilon)$ = *language accepted by* $N\varepsilon$

$$= \{ s \in \Sigma^* \mid \delta_{N\varepsilon}E(Q_0, s) \cap F \neq \varnothing \}$$

$\sim N :=$ **NFA-*equivalent*** *for* $N\varepsilon$ *with no* ε*-transitions*

$\sim N := (Q, \Sigma, \delta_{\sim N}, Q'_0, F)$

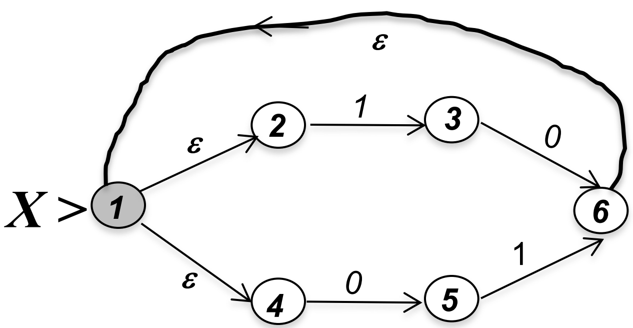
where : $\delta_{\sim N}(q, a) := \delta_{N\varepsilon}E(\{q\}, a)$; $Q'_0 := ECLOSE\,(Q_0)$

Fact : $L(\sim N) = L(N\varepsilon)$

Example for ε-NFA to NFA without ε-transitions transformation

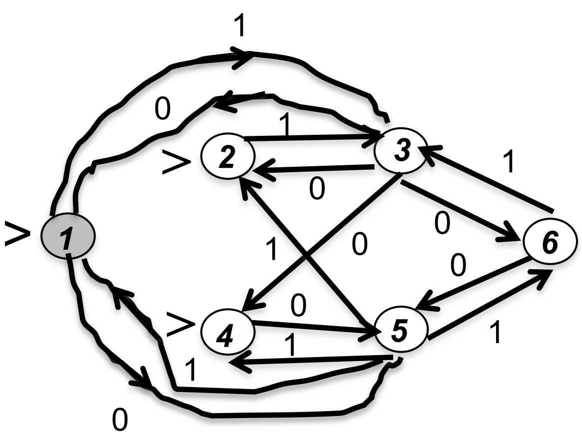
$L_1 = \{01\}$, $L_2 = \{10\}$ $\qquad$ $L := \{ L_1 \cup L_2 \}^*$

q	σ	q'
>1*	0	5
1	1	3
>2	0	∅
2	1	3
3	0	1,2,4,6
3	1	∅
>4	0	5
4	1	∅
5	0	∅
5	1	1,2,4,6
6	0	5
6	1	3

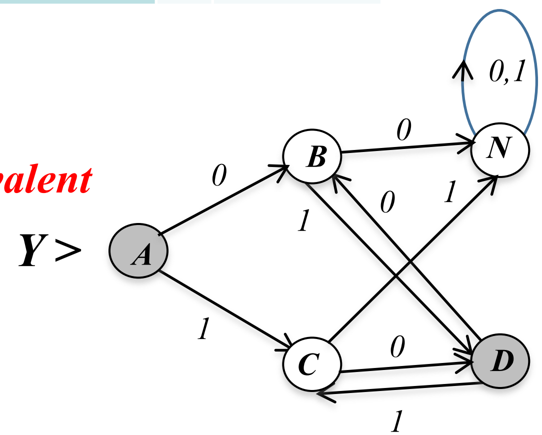


Q	σ	Q'
A*=1,2,4	0	B=5
A=1,2,4	1	C= 3
B=5	0	N
B=5	1	D*=1,2,4,6
C=3	0	D
C=3	1	N
D*	0	B
D	1	C
N	0	N
N	1	N

NFA Equivalent



DFA Equivalent



A Resume of equivalence formulas for DFA , NFA and ε-NFA

(1) $\delta_A : Q \times \Sigma \rightarrow Q$; $\delta_A E : Q \times \Sigma^* \rightarrow Q$; $s \in L(A) \Leftrightarrow \delta_A E(q_0, s) \in F$

(2) $\delta_N : Q \times \Sigma \rightarrow 2^Q$; $\delta_N E : 2^Q \times \Sigma^* \rightarrow 2^Q$; $s \in L(N) \Leftrightarrow \delta_N E(Q_0, s) \cap F \neq \varnothing$

(3) Deterministic Equivalent **D** of an NFA **N** such that **L(N) = L(D)**

$D = (2^Q, \Sigma, \delta_D, Q_0, F_D)$; $\delta_D(X, \sigma) := \cup_{\{v \in X\}} \delta_N(v, \sigma)$; $\delta_D(\varnothing, \sigma) := \varnothing$

$F_D := \{ Y \subseteq Q \mid Y \cap F_N \neq \varnothing \}$

(4) $\delta_{N\varepsilon} : Q \times \Sigma \cup \{\varepsilon\} \rightarrow 2^Q$; $\delta_{N\varepsilon} E : 2^Q \times \Sigma^* \rightarrow 2^Q$; $s \in L(N\varepsilon) \Leftrightarrow \delta_{N\varepsilon} E(Q_0, s) \cap F \neq \varnothing$

(5) Equivalent ~N without ε-transitions of an ε-NFA Nε such that **L(~N) = L(Nε)**

$\sim N := (Q, \Sigma, \delta_{\sim N}, Q'_0, F)$; $\delta_{\sim N}(q,a) := \delta_{N\varepsilon} E(\{q\}, a)$; $Q'_0 := ECLOSE(Q_0)$