Regular Expressions vs Finite State Automata

*- Regular Expressions **(RE**s**)** are an **algebraic** means for defining languages.*

*- Languages accepted by **DFA's** and **NFA's** vs **RE's***

Definition of a set RE of regular expressions

(over a finite set $\Sigma := \{\sigma_1, \sigma_2, \ldots \sigma_K\}$ *)*

Recursive Formal Definition

(A) *(Basis)* ***e*** , $\varnothing$ *and* $\sigma_1, \sigma_2, \ldots, \sigma_K$ *are all elements of* ***RE***

(B) *(Recursion)*

(1) If ***F*** *and* ***G*** *are in* ***RE*** *then so is* ***F+G***

(2) If ***F*** *and* ***G*** *are in* ***RE*** *then so is* ***F.G***

(3) If ***F*** *is in* ***RE*** *then so is* ***F****

(4) If ***F*** *is in* ***RE*** *then so is* ***(F)***

We call ***each element*** *of the set* ***RE*** *a* ***regular expression*** *(symbolized by* ***E****) !*

Example of a RE (over the set $\Sigma := \{0,1\}$)

*Each symbol **E** represents some regular expression*

Is ***1+(1.0*).(1*.0)+e*** *an element of* ***RE*** *?*

$$\mathbf{1+(1.0^*).(1^*.0)+e} \xrightarrow{0,1,e\ \in\ \mathbf{RE}} \mathbf{E+(E.E^*).(E^*.E)+E}$$

$$\xrightarrow{E^*\in\ \mathbf{RE}} \mathbf{E+(E.E).(E.E)+E} \xrightarrow{E.E\in\ \mathbf{RE}} \mathbf{E+(E).(E)+E}$$

$$\xrightarrow{(E)\in\ \mathbf{RE}} \mathbf{E+E.E+E} \xrightarrow{E.E\in\ \mathbf{RE}} \mathbf{E+E+E} \xrightarrow{E+E\in\ \mathbf{RE}} \mathbf{E+E}$$

$$\xrightarrow{E+E\in\ \mathbf{RE}} \mathbf{E} \in \mathbf{RE}$$

Language interpretation is a mapping $L : RE \rightarrow 2^{\Sigma^*}$ given by :

$L(e) := \{e\}$ *where* e : = *empty string*

$L(\varnothing) := \varnothing$ *where* $\varnothing$: = *null language (language with no strings)*

$L(\sigma_j) := \{\sigma_j\}$, $j=1, \ldots, K$

$L(F+G) := L(F) \cup L(G)$

$L(F.G) := L(F).L(G)$

$L(F^*) := L(F)^*$

$L((F)) := (L(F))$

Relation of Basic Operations on Languages to REs

(1) **Union** : $L = L_1 \cup L_2$ $\Rightarrow$ $E + E$

(2) **Concatenation** : $L = L_1 . L_2$ $\Rightarrow$ $E.E$

formal *logical notation for AND =* ***conjunction***

$L_1 . L_2 := (s \in \Sigma^* | \; s = u.v \; ; \; u \in L_1 \wedge v \in L_2)$

informal *logical notation for AND =* ***conjunction***

(3) **Closure** (star or Kleene closure) $L^* = \cup_{k=0,\infty} L^k$ $\Rightarrow$ E^*

$L^k := (s \in \Sigma^* | \; s = u_1 . u_2 \ldots u_k \; ; \; u_j \in L \text{ for } j=1,\ldots k)$

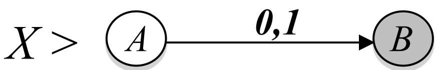
$L^0 := \{e\}$ (i.e. by definition)

Definition *: A language L is called a* ***regular language*** *if it is the language interpretation of a* ***regular expression***

Main Theorem

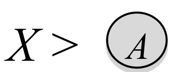
A language is ***regular*** *if and and only if it is accepted by some* ***finite state automaton*** *.*

*Examples of simple DFA **X** to **RE** conversions*

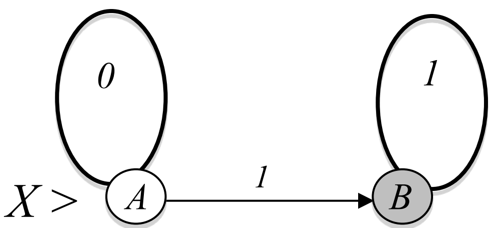


$L(X) = E = 0+1$

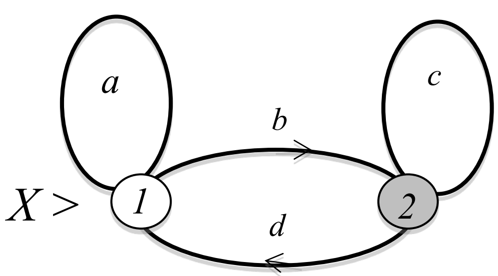
$L(X) = E = \varnothing$



$L(X) = E = e$



$L(X) = E = 0^*.1.1^*$



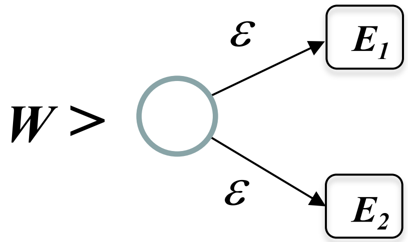
a.b.c not in *L'*

$\cancel{L'(X) = E = (a^*.b.c^*.d)^*.b.c^*}$ ***not equivalent***

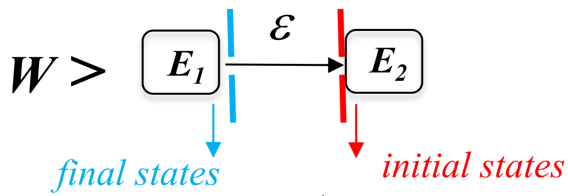
$L(X) = E = (a^*+b.c^*.d)^*.b.c^*$

*Examples of **RE** to **ε–NFA** conversions*

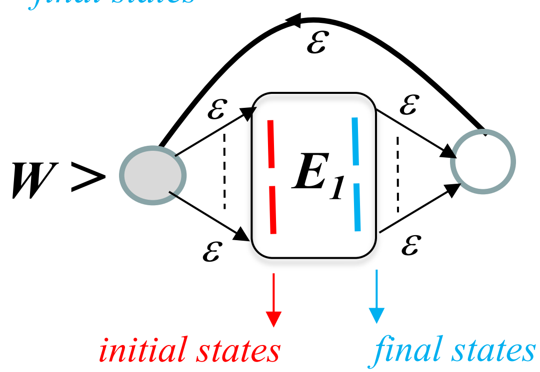
$E = E_1 + E_2$



$E = E_1 \,.\, E_2$



$E = (E_1)^*$



Proof of the Main Theorem

(if) **Idea :**

(1) Let a DFA $\mathbf{D = (Q, \Sigma, \delta, 1, F)}$ with $\mathbf{Q = \{1,2, \ldots, n\}}$

(2) Let R_{ij}^{k} denote the language corresponding to strings covering **all** paths of **D** that start at state **i** ; end at state **j** ; and is only allowed to visit intermediate states with labels $p \leq k$

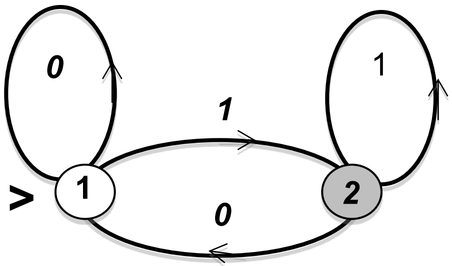
(3) Note that $\mathbf{L(D)} = \cup_{(m \in F)} R_{1m}^{n}$ where **1** is the initial state

(4) Prove by induction on **k** that R_{ij}^{k} is a **RE** for all $\mathbf{i,j = 1,\ldots,n}$ and $\mathbf{k = 0,\ldots, n}$.

(see the next slide first formula)

(5) Conclude that **L(D)** is a **RE.**

Illustration of the language R_{ij}^k



R_{11}^0 = *start at **1** and terminate at **1** (no intermediate visit is allowed)* = $0+e$

R_{12}^0 = *start at **1** and terminate at **2** (no intermediate visit is allowed)* = 1

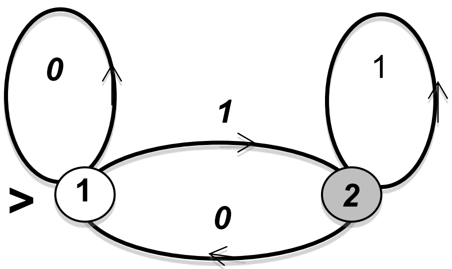
R_{11}^1 = *start at **1** , move to allowed intermediate state **1** as desired and terminate at **1*** = 0^*

R_{12}^1 = *start at **1** , move to intermediate state **1** as desired and finally terminate at **2*** = $0^*.1$

The Inductive Formula for DFA → RE

$R_{ij}^{k} = R_{ij}^{k-1} + R_{ik}^{k-1}. (R_{kk}^{k-1})^{*}. R_{kj}^{k-1}$; $i,j=1,\ldots,n$; $k=0,\ldots,n$

Example



$R_{11}^{0} = 0+e$; $R_{22}^{0} = 1+e$; $R_{21}^{0} = 0$; $R_{12}^{0} = 1$

$R_{11}^{1} = 0^{*}$; $R_{22}^{1} = 0.0^{*}.1+1+e$; $R_{21}^{1} = 0.0^{*}$; $R_{12}^{1} = 0^{*}.1$

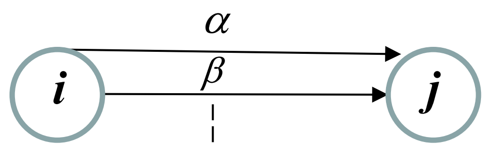
$R_{11}^{2} = \ldots$; $R_{22}^{2} = \ldots$; $R_{21}^{2} = \ldots$; $R_{12}^{2} = \ldots$

After Simplification : $L=R_{12}^{2} = (0^{*}+1.1^{*}.0)^{*}.1.1^{*}$

Continue with the Proof *(by induction on the superscript* ***k****)*

Basis *(k=0)*

$R_{ij}^{0} = \alpha + \beta + \ldots$ ***if***

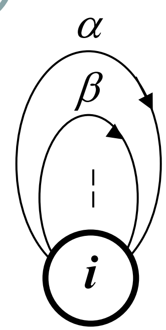


$E \rightarrow E+E$

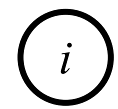
$\alpha, \beta, \ldots, e, \varnothing \in E$

$= \varnothing$ ***if***

$R_{ii}^{0} = \alpha + \beta + \ldots + e$ ***if***



$R_{ii}^{0} = e$ ***if***



Induction *(true for* ***k-1*** *, show for* ***k*** *)*

$R_{ij}^{k} = R_{ij}^{k-1} + R_{ik}^{k-1}. (R_{kk}^{k-1})^{*}. R_{kj}^{k-1}$; *i,j=1,…,n ; k=0,…,n*

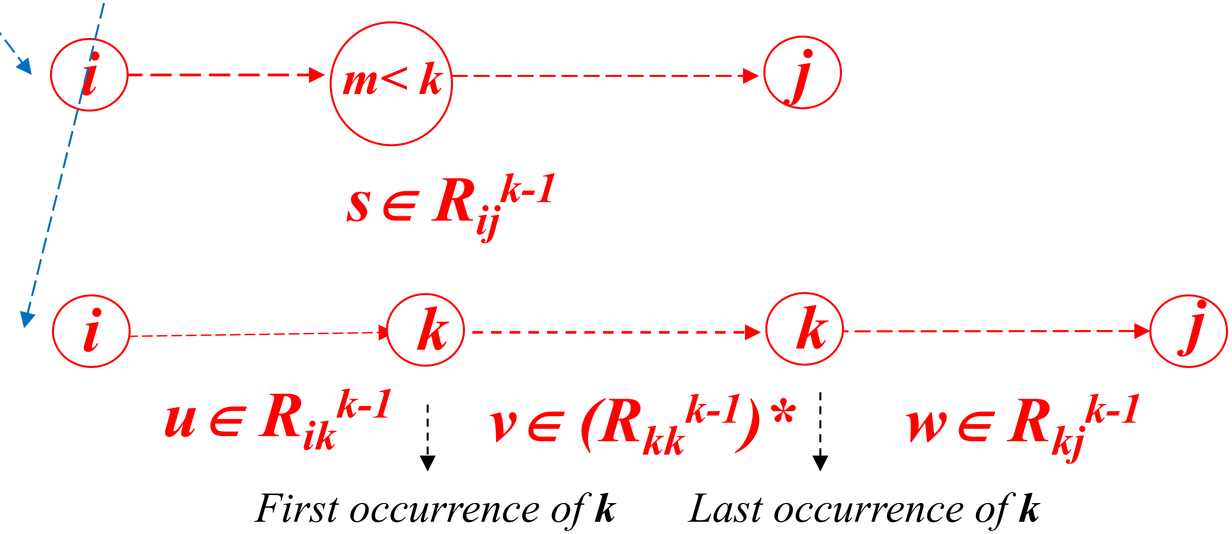
$(E) \rightarrow E \quad E^{*} \rightarrow E \quad E.E \rightarrow E$ ***(twice)*** $\quad E+E \rightarrow E$

$R_{kk}^{k-1} \in E \Rightarrow (R_{kk}^{k-1}) \in E \Rightarrow (R_{kk}^{k-1})^{*} \in E \Rightarrow R_{ik}^{k-1}. (R_{kk}^{k-1})^{*}. R_{kj}^{k-1} \in E$ ***etc.***

Interpreting the induction formula :

$$R_{ij}^{k} = \underline{R_{ij}^{k-1}} + \underline{R_{ik}^{k-1} . (R_{kk}^{k-1})^{*} . R_{kj}^{k-1}} \;\; ; \;\; i,j=1,\ldots,n \; ; \; k=0,\ldots,n$$

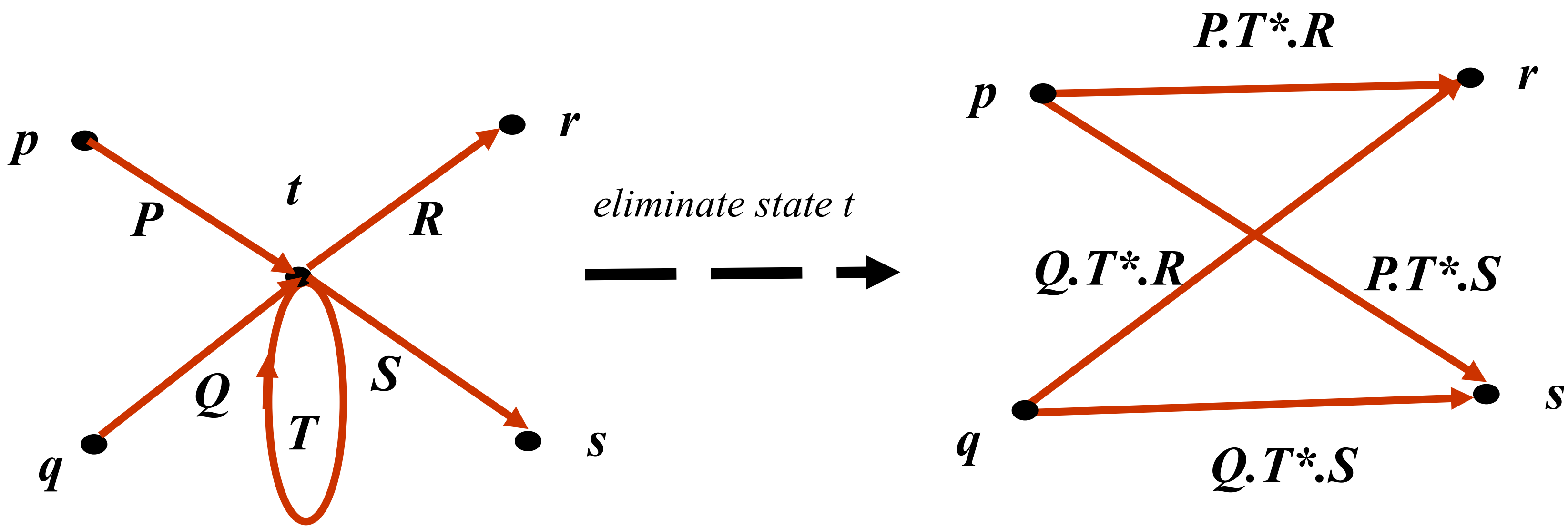
A path (string) s *in* R_{ij}^{k} *can be expressed in terms of a sequence of states as shown below :*



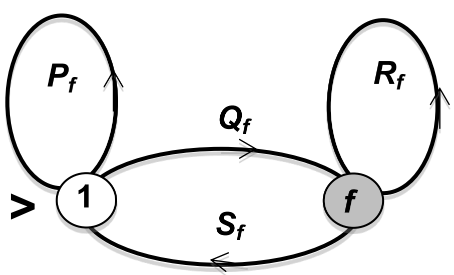
(OR) $s=u.v.w$

So : $L(D) = \sum_{(m \in F)} R_{1m}^{n} = E + E + \ldots + E =$ **a regular expression**

Alternative Proof of the Main Theorem (State Elimination)



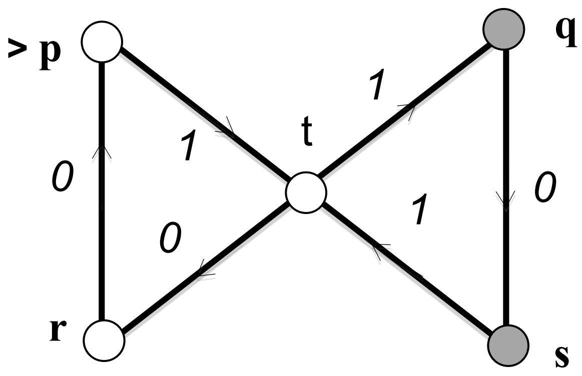
*After eliminating all non-initial and non-final states ; start eliminating all final states except one **f** in **F** and repeat this for each distinct **f** in **F**. Then the following picture(s) prevail*



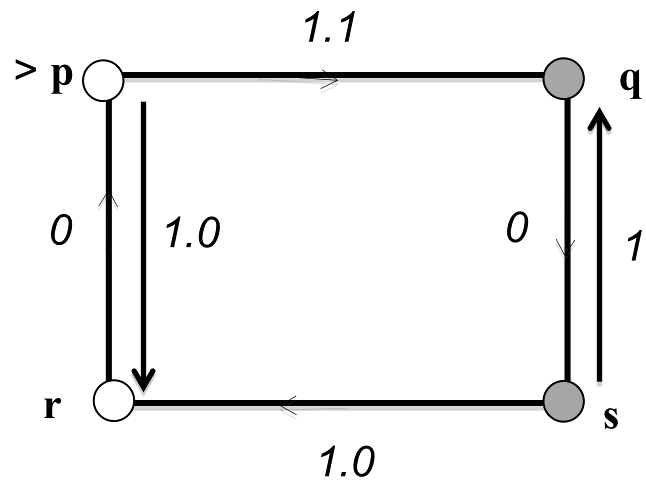
$$L_f = (P_f{}^* + Q_f . R_f{}^* . S_f)^* . Q_f . R_f{}^*$$

$$L = \Sigma_{(f \in F)} L_f$$

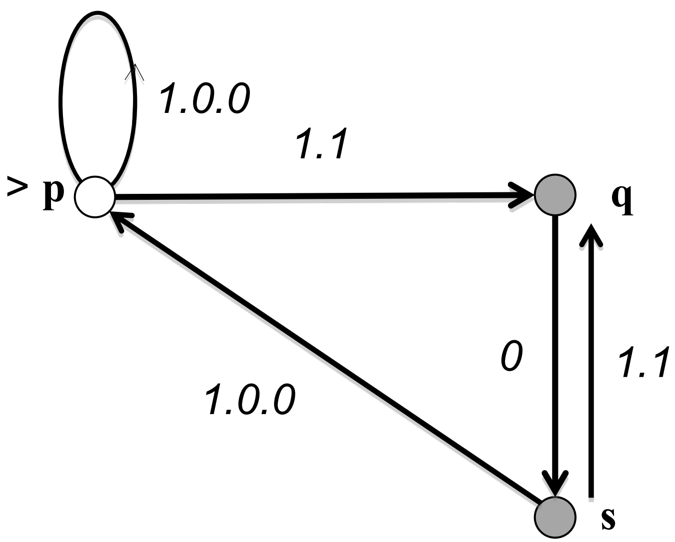
Example



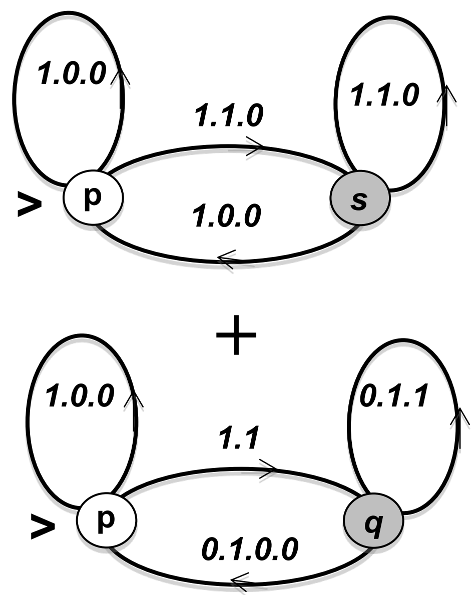
eliminate t



eliminate r



eliminate q and s



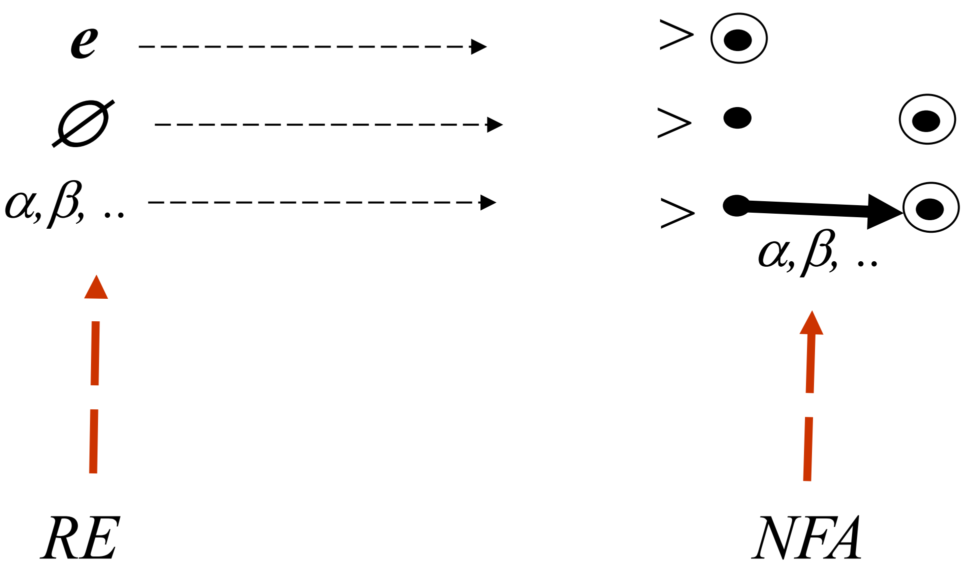
L_f =((1.0.0)* + (1.1.0).(1.1.0)*.(1.0.0))*.(1.1.0).(1.1.0)* + etc

Proof of the Main Theorem

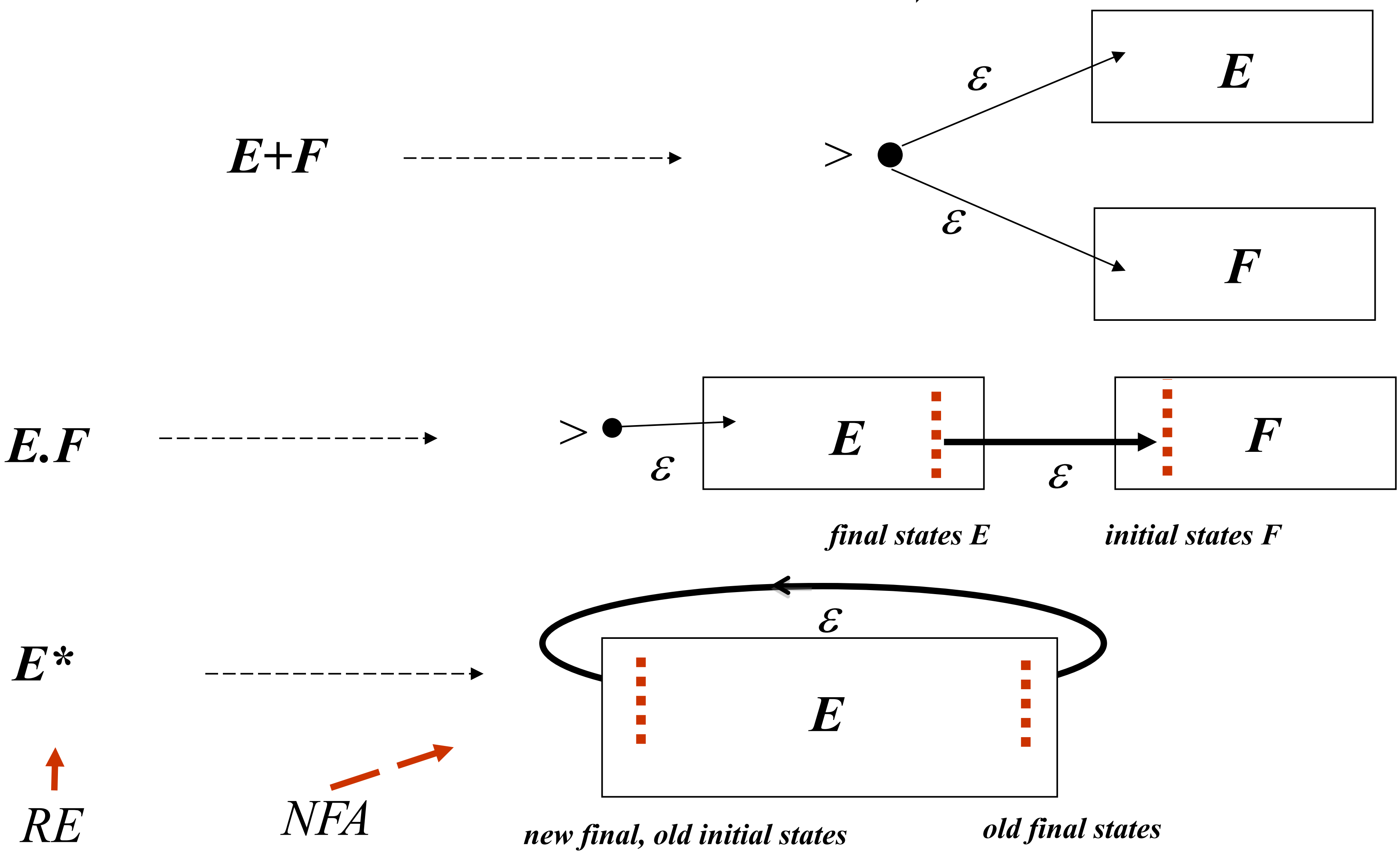
(only if) **Idea :**

given ***RE****s over the set* $\boldsymbol{\Sigma = (\alpha, \beta, \gamma, \ldots,)}$

Basis

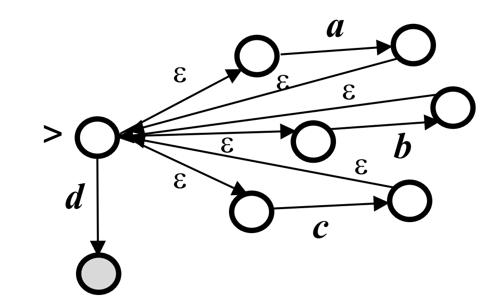


*Proof of the **Main Theorem** (continued)* (only if)

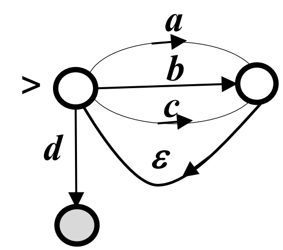


Some short cuts !

(a+b+c).d*

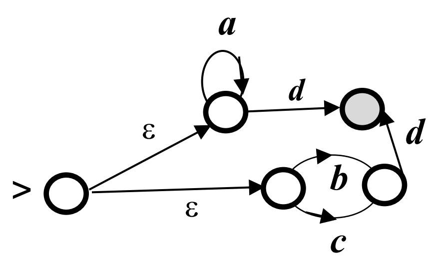


Simplified !

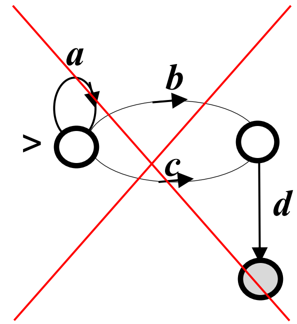


But !

(a+b+c).d*



NOT!



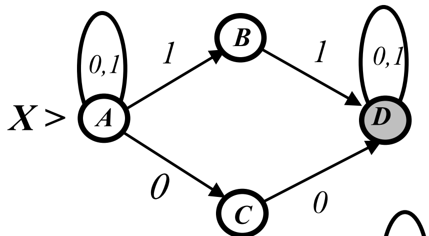
Examples

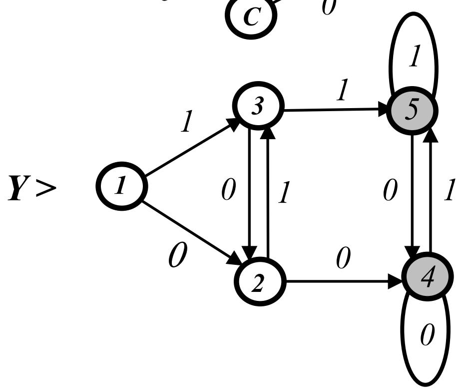
*Construct a regular expression **E** over the alphabet **Σ= {0,1}** to represent a language where there is no substring of **2** or more consecutive **1**'s AND there is no substring of **2** or more consecutive **0**'s.*

complement = a substring 1.1 or 0.0

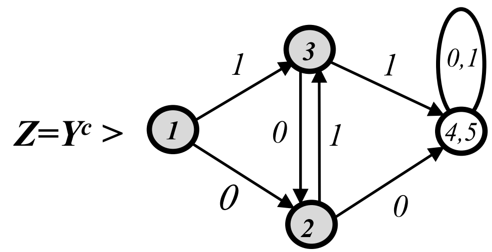
Solution

$$E^c = (0+1)^*.(1.1 + 0.0).(1+0)^*$$





Q	σ	Q'
A (1)	0	A,C
A	1	A,B
A,C (2)	0	A,C,D
A,C	1	A,B
A,B (3)	0	A,C
A,B	1	A,B,D
A,C,D (4*)	0	A,C,D
A,C,D	1	A,B,D
A,B,D (5*)	0	A,C,D
A,B,D	1	A,B,D



A digression on the previous Example

(i) *Construct a regular expression **E** over the alphabet **Σ= {0,1}** to represent a language where there is no substring of **2** or more consecutive **1's AND 2** or more consecutive **0's**.*

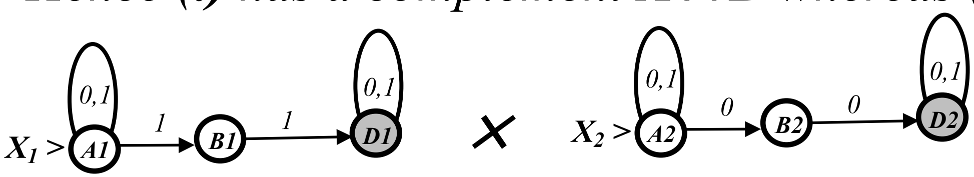
(ii) *Construct a regular expression **E** over the alphabet **Σ= {0,1}** to represent a language where ; there is no substring of **2** or more consecutive **1's AND** ; there is no substring with **2** or more consecutive **0's**.*

logical negation symbol

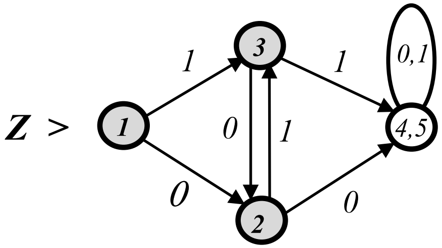
*Let **A** ::={ **11** is a substring of **L**} and **B** := { **00** is a substring of **L**}*

Then (i) above says $\neg(A \wedge B)$ *whereas **(ii)** says* $(\neg A) \wedge (\neg B)$

*Hence **(i)** has a complement* $A \wedge B$ *whereas **(ii)** has a complement* $A \vee B$

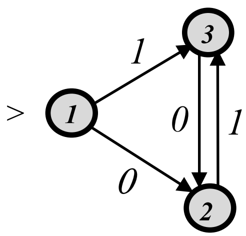


is the product NFA for $A \wedge B$ *case .*

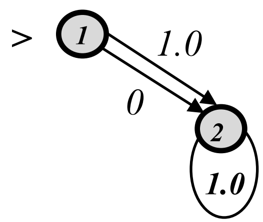


Solution *(by state elimination)*

1- Eliminate 4,5

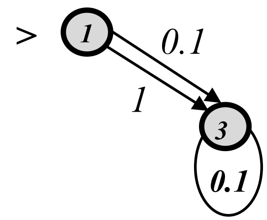


2- Eliminate all but 2



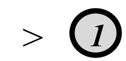
$(1.0+0).(1.0)^*+e$

3- Eliminate all but 3



$(0.1+1).(0.1)^*+e$

4- Eliminate all but 1



e

$$E = (1.0+0).(1.0)^* + (0.1+1).(0.1)^* + e$$

*0 OR sequences of consecutive **0**'s and **1**'s that **start with 1 OR 0 and end with 0***

*1 OR sequences of consecutive **0**'s and **1**'s that **start with 0 OR 1 and end with a 1***

Algebraic Laws For REs

Trivial Laws

(1) $L+M = M+L$; $(L+M) + N = L + (M+N)$; $(L.M).N = L.(M.N)$

(2) $\phi + L = L$; $e.L = L.e = L$; $\phi.L = \phi$

(3) $L.(M+N) = L.M + L.N$; $(L+M).N = L.N + M.N$; $L+L = L$

Non-trivial Laws

(4) $(L+M)^* = (L^*+M^*)^* = (L^*.M^*)^*$

(5) $(L.M)^* \subseteq (L^*.M^*)^*$ and $(L.M)^* = (L^*.M^*)^*$ iff $e \in L$ ***and*** $e \in M$

Proof of (4) $\rightarrow (L+M)^* = (L^*. M^*)^*$

Two steps : (1) $(L+M)^* \subseteq (L^*.M^*)^*$; *(2)* $(L^*.M^*)^* \subseteq (L+M)^*$

(1) Let $u \in (L+M)^*$ *then* $u = u_1 . u_2 . \dots . u_k$ for some integer $k \geq 0$ where for each j , $u_j \in L+M$;

but $L \subseteq L^* \subseteq L^*. e \subseteq L^* . M^*$ *and* $M \subseteq M^* \subseteq e . M^* \subseteq L^*. M^*$;

hence $u_j \in L^*. M^* + L^*. M^* = L^*. M^*$ and therefore $(L+M)^* \subseteq (L^* . M^*)^*$

(2) Conversely let $u \in (L^*.M^*)^*$ *then by definition* $u = u_1 . u_2 . \dots . u_k$ *where* $u_j \in L^*.M^*$;

hence $u_j = v_j^1 . v_j^2 \dots . v_j^{l(j)} . w_j^1 . w_j^2 \dots . w_j^{p(j)}$ *where* $v_j^m \in L \subseteq L+M$ and $w_j^m \in M \subseteq L+M$;

thus $u = z_1 . z_2 . \dots z_q$ *where* $q = \sum_{j=1,k} l(j)+p(j)$ *and each* $z_i \in L+M$. *Hence* $u \in (L+M)^*$;

this proves that $(L^*.M^*)^* \subseteq (L+M)^*$

Proof of $(L+M)^* = (L^*+M^*)^*$ given (4) $\rightarrow (L+M)^* = (L^*. M^*)^*$

Since $L \subseteq L^*$ *and* $M \subseteq M^*$ *it follows that* $(L+M)^* \subseteq (L^*+M^*)^*$

Conversely let $u \in (L^*+M^*)^*$ *then* $u = (v_1+w_1). \ \ldots \ . (v_k+w_k)$ *where for each* j $v_j \in L^*$ *and* $w_j \in M^*$.

We show that $u \in (L^*. M^*)^*$ *by using induction on* k.

For $k=1$ $\quad v_1 \in L^* \subseteq L^*. e \subseteq L^*.M^* \subseteq (L^*. M^*)^*$

similarly $w_1 \in M^* \subseteq e. M^* \subseteq L^*.M^* \subseteq (L^*. M^*)^*$ *hence* $v_1+w_1 \subseteq (L^*. M^*)^*$.

Now assume statement holds for $k\text{-}1$, *hence* $z := (v_1+w_1). \ \ldots \ . (v_{k-1} + w_{k-1}) \in (L^*. M^*)^*$

But using the above reasoning for v_1+w_1 *it follows that* $v_k+w_k \in (L^*. M^*)^*$

and therefore $u = z \,. (v_k+w_k) \in (L^*. M^*)^*. (L^*. M^*)^* = (L^*. M^*)^*$ *using the obvious identity* $K^* . K^* = K^*$ *for any language* K. *This proves that* $(L^*+M^*)^* \subseteq (L^*. M^*)^*$

but by (4) $(L+M)^* = (L^*. M^*)^*$ *hence* $(L^*+M^*)^* \subseteq (L + M)^*$ *and result follows*