Regular expressions

Are a formalism for describing a certain class of languages: declarative rather than computational view

Definition: given an alphabet Σ, regular expressions are strings over the alphabet $\Sigma \cup \{+, *, (,), \cdot, \varepsilon, \emptyset\}$ defined inductively as follows:

- basis: ε, $\emptyset$, and each $a \in \Sigma$ is a R.E.
- inductive step: if E and F are R.E., then so are:
  - $E+F$ (union)
  - $E \cdot F$ (concatenation)
  - E^* (closure)
  - (E) (parentheses)

Example: $a \cdot (a+b)^* \cdot b^* a$

Definition: language $\mathcal{L}(E)$ defined by a R.E. E is also defined inductively:

- $\mathcal{L}(\varepsilon) = \{\varepsilon\}$ empty word
- $\mathcal{L}(\emptyset) = \emptyset$ empty language
- for each $a \in \Sigma$, $\mathcal{L}(a) = \{a\}$
- $\mathcal{L}(E+F) = \mathcal{L}(E) \cup \mathcal{L}(f)$
- $\mathcal{L}((E)) = \mathcal{L}(E)$
- $\mathcal{L}(E \cdot F) = \mathcal{L}(E) \cdot \mathcal{L}(F)$ ... concatenation

<u>concatenation</u> of two languages L_1 and L_2:

$$L_1 \cdot L_2 = \{w \mid w = x \cdot y, x \in L_1, y \in L_2\}$$

Example: $E = \varepsilon + 1 \Rightarrow \mathcal{L}(E) = \{\varepsilon, 1\}$

$F = \varepsilon + 0 + 1 \Rightarrow \mathcal{L}(F) = \{\varepsilon, 0, 1\}$

$G = E \cdot F \Rightarrow \mathcal{L}(G) = \{\varepsilon, 0, 1, 10, 11\}$

- $\mathcal{L}(E^*) = (\mathcal{L}(E))^*$ ... closure

<u>closure</u> of a language L ?

we first define the <u>powers</u> of a language L :

- $L^0 = \{\varepsilon\}$.
- $L^k = L^{k-1} \cdot L$

Hence $L^k = \{w \mid w = x_1 \cdots x_k, \text{ with } \forall i, x_i \in L\}$

Closure of L : $L^* = L_0 \cup L_1 \cup L_2 \cup \cdots$

Example $\begin{cases} E = 0 + 1 & \Rightarrow \mathcal{L}(E) = \{0, 1\} \\ F = E^* & \Rightarrow \mathcal{L}(F) = \text{set of all binary strings} \end{cases}$

$\{ E = 0 \cdot 0 \quad \Rightarrow \mathcal{L}(E^*) = \{\varepsilon, 00, 0000, 000000, \dots\}$

= all even-length strings of 0's

20/10/2004

Positive closure of a language L

$L^+ = L^1 \cup L^2 \cup \cdots$

We can introduce a positive closure operator in R.E.

$\mathcal{L}(E^+) = (\mathcal{L}(E))^+$

Note : we have to distinguish between an expression E and the language $\mathcal{L}(E)$ defined by E

When we write $E = F$, we usually mean not syntactic equality but equality of the corresponding languages, i.e. $\mathcal{L}(E) = \mathcal{L}(F)$. In other words, equality is in the algebra of R.E.

<u>Precedence</u> of operators :

high $\downarrow$ low: $*$, $\cdot$, $+$

example: $E + F \cdot G^* = E + (F \cdot (G^*))$

Algebraic laws for R.E.

similar to the laws for arithmetic expressions, we can express laws for R.E.: treat $+$ as sum, $\cdot$ as product

(note! the laws are not the same as for arithmetic expressions)

- associativity of $\cdot$ and $+$

$$(E \cdot F) \cdot G = E \cdot (F \cdot G) = E \cdot F \cdot G$$
$$(E + F) + G = E + (F + G) = E + F + G$$

- commutativity of $+$

$$E + F = F + E$$

Note: $\cdot$ is not commutative: $E \cdot F \neq F \cdot E$

- distributivity:
  1) left distributive law of $\cdot$ over $+$: $E \cdot (F + G) = E \cdot F + E \cdot G$
  2) right —"— : $(F + G) \cdot E = F \cdot E + G \cdot E$

Proof of 1): the law actually holds for arbitrary languages, and does not require E, F, G to be R.E.

Hence, we prove: for arbitrary languages L, M, N:

$$L \cdot (M \cup N) = L \cdot M \cup L \cdot N$$

We show that for a string w we have $w \in L \cdot (M \cup N)$ iff $w \in L \cdot M \cup L \cdot N$

"Only-if": $w \in L \cdot (M \cup N) \Rightarrow w = x \cdot y$ with $x \in L$, $y \in M \cup N$
Since $y \in M \cup N$, either $y \in M$ or $y \in N$ (or both)
If $y \in M$, then $w = x \cdot y \in L \cdot M$, hence $w \in L \cdot M \cup L \cdot N$
(similarly for $y \in N$)

"If": $w \in L \cdot M \cup L \cdot N$, hence either $w \in L \cdot M$ or $w \in L \cdot N$.
If $w \in L \cdot M$, then $w = x \cdot y$, with $x \in L$, $y \in M$. (Similarly for $w \in L \cdot N$)
Hence $y \in M \cup N$, and $w = x \cdot y \in L \cdot (M \cup N)$.

Example: $0 \cdot 0 + 0 1^* = 0 \cdot (0 + 1^*)$

we can factor out a 0 from the union

What about $0 + 0 \cdot 1^*$?

if we factor out a 0, what remains for the summand on the left?

$$0 + 0 \cdot 1^* = 0 \cdot \varepsilon + 0 \cdot 1^* = 0 \cdot (\varepsilon + 1^*) = 0 \cdot 1^*$$

identity

since $\varepsilon \in \mathcal{L}(1^*)$

- identities and annihilators (hold for arbitrary languages)
  - $\emptyset + E = E + \emptyset = E$
  - $\varepsilon \cdot E = E \cdot \varepsilon = E$
  - $\emptyset \cdot E = E \cdot \emptyset = \emptyset$
- idempotency
  - $E + E = E$
  - $(E^*)^* = E^*$ Proof: $\boxed{\text{Exercise 3.4.1 f}}$
- other laws for closure (already seen)
  - $\emptyset^* = \varepsilon$
  - $\varepsilon^* = \varepsilon$
  - $E^+ = E \cdot E^* = E^* \cdot E$
  - $E^* = E^+ + \varepsilon$

    Note: if $\varepsilon \in \mathcal{L}(E)$, then $E^* = E^+$

$\boxed{\text{Exercise 3.4.4.}}$ Prove that: $(E^* F^*)^* = (E + F)^*$

Exercise 3.1.1 Write R.Es for the following languages

a) $\{ w \in \{a, b, c\}^* \mid w$ contains at least one a and at least one b $\}$

b) $\{ w \in \{0,1\}^* \mid w$'s tenth symbol from the right is a 1$\}$

c) $\{ w \in \{0,1\}^* \mid w$ contains at most one pair of consecutive 1's$\}$

Exercise 3.1.2 Write R.E.s for the following languages

a) The set of all strings over $\{0,1\}$ s.t. every pair of adjacent 0's appears before any pair of adjacent 1's

b) The set of strings of 0's and 1's whose number of 0's is divisible by 5

Solutions:

3.1.1 a) $c^* \cdot a \cdot (a+c)^* \cdot b \cdot (a+b+c)^* +$
$c^* \cdot b \cdot (b+c)^* \cdot a \cdot (a+b+c)^*$

b) $(0+1)^* 1 \cdot \underbrace{(0+1) \cdot \ldots \cdot (0+1)}_{9 \text{ times}}$

c) $0^* \cdot (1 \cdot 0^+)^* \cdot 1 \cdot 1 \cdot (0^+1) \cdot 0^* + 0^* \cdot (1 \cdot 0^+)^*$

3.1.2 a) $\underbrace{0^* (1 \cdot 0^+)^*}_{\text{no pair of adjacent 1's}} \cdot \underbrace{1^* \cdot (0 \cdot 1^+)^*}_{\text{no pair of adjacent 0's}}$

b) $(1^* \cdot 0 \cdot 1^* \cdot 0 \cdot 1^* \cdot 0 \cdot 1^* \cdot 0 \cdot 1^* \cdot 0 \cdot 1^*)^*$

Regular languages

What is the relationship between the classes of languages studied so far?

$$\begin{tikzcd}
\varepsilon\text{-NFA} \arrow[r, shift left, "\varepsilon\text{-elimination}"] & \text{NFA} \arrow[l, shift left] \arrow[r, shift left, "\text{subset-construction}"] & \text{DFA} \arrow[l, shift left] \arrow[dl, bend left, "?"] \\
& \text{R.E.} \arrow[ul, bend left, "?"] &
\end{tikzcd}$$

regular languages

Theorem: (R.E. $\to$ ε-NFA)

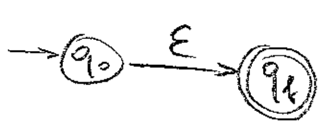
For every R.E. E there is an ε-NFA A_E s.t. $\mathcal{L}(A_E) = \mathcal{L}(E)$.

Proof: let us call an ε-NFA *simple* if
- it has only one final state
- the initial state has no incoming arcs
- the final — " — outgoing — " —

We show by structural induction that for each R.E. E there is a simple ε-NFA A_E s.t. $\mathcal{L}(E) = \mathcal{L}(A_E)$

Basis: $E = \varepsilon$, $E = \emptyset$, $E = a$ for some $a \in \Sigma$

A_ε



$A_\emptyset$

A_a

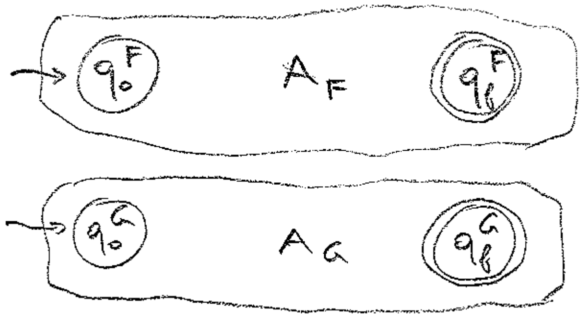
Inductive case: 1) $E = F + G$

2) $E = F \cdot G$

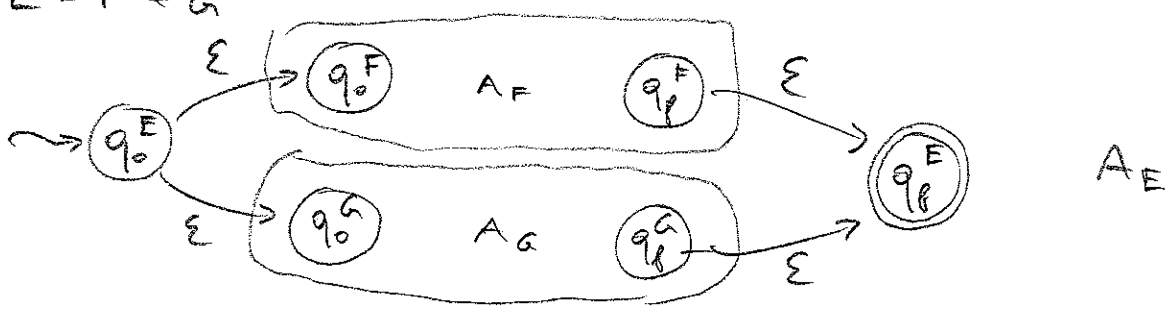
3) $E = F^*$

4) $E = (F)$

By I.H., there are simple ε-NFAs for F and G

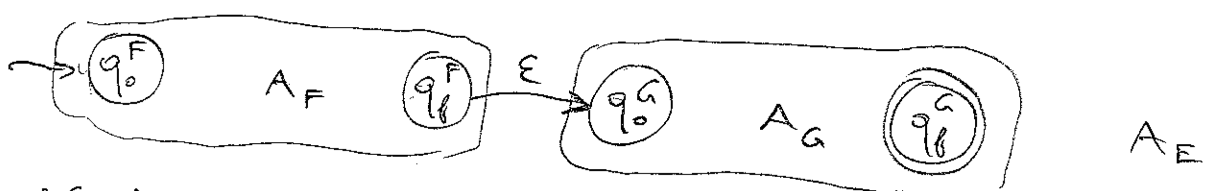


1) $E = F + G$



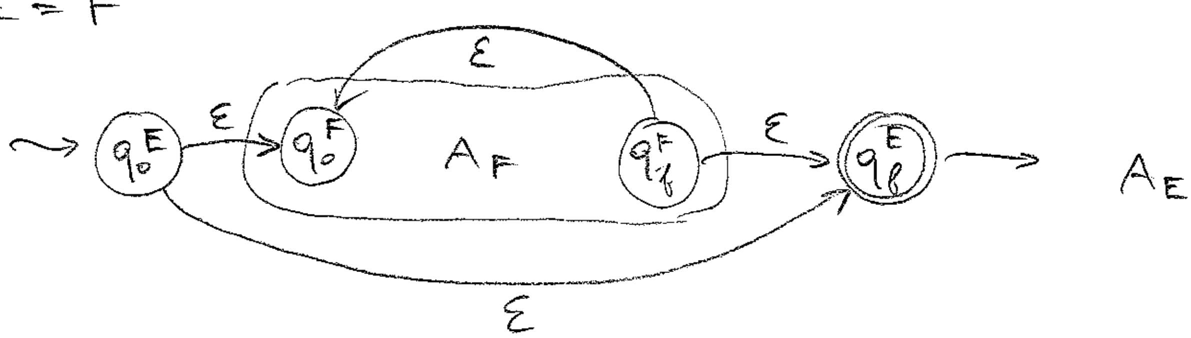
$$\mathcal{L}(A_E) = \mathcal{L}(A_F) \cup \mathcal{L}(A_G) \underset{\text{by I.H.}}{=} \mathcal{L}(F) \cup \mathcal{L}(G) = \mathcal{L}(F+G) = \mathcal{L}(E)$$

2) $E = F \cdot G$



$$\mathcal{L}(A_E) = \mathcal{L}(A_F) \cdot \mathcal{L}(A_G) \underset{\text{by I.H.}}{=} \mathcal{L}(F) \cdot \mathcal{L}(G) = \mathcal{L}(F \cdot G) = \mathcal{L}(E)$$

3) $E = F^*$

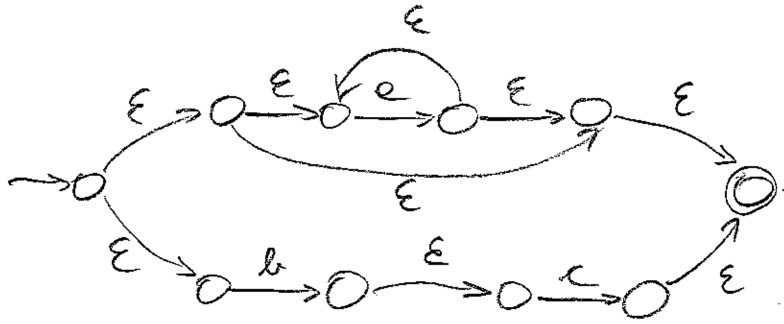


3) $E = (F)$

$$A_E = A_F$$

q.e.d.

<u>Example</u>: $E = a^* + b \cdot c$



25/11/2004

OPTIONAL

<u>Theorem</u> (DFA → R.E.)

For every DFA A there is a R.E. E_A s.t. $\mathcal{L}(E_A) = \mathcal{L}(A)$

<u>Proof</u>: Let $A = (Q, \Sigma, \delta, q_1, F)$

We assume without loss of generality (w.l.o.g) that
$Q = \{q_1, q_2, \ldots, q_n\}$

Let us define $L_{ij} = \{w \mid \hat{\delta}(q_i, w) = q_j\} =$
$\forall i,j \in \{1, \ldots, n\}$
$= \{w \mid w \text{ takes A from } q_i \text{ to } q_j\}$

note that $L_{ij} = \mathcal{L}(A_{ij})$ with $A_{ij} = (Q, \Sigma, \delta, q_i, \{q_j\})$

We aim at constructing R.E.s E_{ij} for L_{ij}.
Then we can take $E_A = \sum_{q_j \in F} E_{1j}$, since

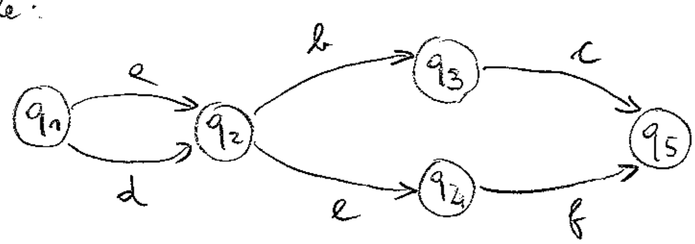
$$\mathcal{L}(E_A) = \bigcup_{q_j \in F} \mathcal{L}(E_{1j}) = \bigcup_{q_j \in F} L_{1j} = \{w \mid \hat{\delta}(q_1, w) \in F\} = L(A)$$

How can we compute E_{ij}?

Let us define $\forall i,j \in \{1, \ldots, n\}$, $\forall k \in \{0, \ldots, n\}$

$L_{ij}^k = \{w \mid$ A goes from q_i to q_j on input w,
passing only through $q_1, \ldots, q_k$ as intermediate states$\}$

Example:



$abc \in L^3_{15}$

$def \notin L^3_{15}$ but $def \in L^4_{15}$

$def \in L_{15}$

$L^0_{12} = \{a, d\}$

$L^3_{15} = \{abc, dbc\}$

$L^4_{15} = L^5_{15} = L_{15} = \{abc, dbc, aef, def\}$

Note: $L^n_{ij} = L_{ij}$

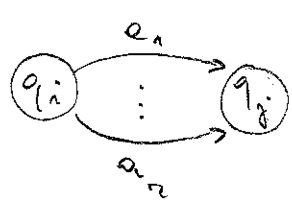
Hence, we are done if we can construct R.E.s E^k_{ij} for L^k_{ij}.
We can simply take $E_{ij} = E^n_{ij}$, and hence $E_A = \sum_{q_j \in F} E^n_{1j}$.

We construct E^k_{ij} by induction on k:

Basis: we construct E^0_{ij}, for all $i, j \in \{1, \dots, n\}$

Since $k=0$, we cannot go through any intermediate state.
2 cases, each with 2 subcases:

$i \neq j$

- 

$E^0_{ij} = a_1 + \dots + a_h$

- 

$E^0_{ij} = \emptyset$

$i = j$

- 

$E^0_{ii} = \varepsilon + a_1 + \dots + a_k$

- 

$E^0_{ii} = \varepsilon$

Induction: assume we have constructed E_{ij}^{k-1} $\forall i, j \in \{1, \dots, n\}$ (3.10)

we show how to construct E_{ij}^{k}

Observe.

- L_{ij}^{k} will include L_{ij}^{k-1}
- it additionally will contain those words that lead through q_k at least once, when going from q_i to q_j

$w = x_1 x_2 \cdots x_s$ where: $\rightsquigarrow$ represents transitions going at most through $\{x_1, \dots, x_{k-1}\}$

then $x_1 \in L_{ik}^{k-1}$

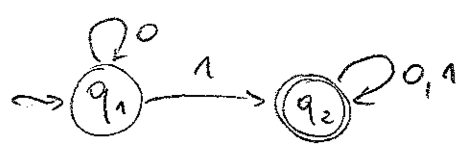
$x_2, \dots, x_{s-1} \in L_{kk}^{k-1}$

$x_s \in L_{kj}^{k-1}$

$\Rightarrow w \in L_{ik}^{k-1} \cdot (L_{kk}^{k-1})^* \cdot L_{kj}^{k-1}$

$$\Rightarrow E_{ij}^{k} = E_{ij}^{k-1} + E_{ik}^{k-1} \cdot (E_{kk}^{k-1})^* \cdot E_{kj}^{k-1}$$

Example:



A

k	E_{11}^{k}	E_{12}^{k}	E_{21}^{k}	E_{22}^{k}
0	$\varepsilon + 0$	1	$\emptyset$	$\varepsilon + 0 + 1$
1	$(\varepsilon+0) + (\varepsilon+0)(\varepsilon+0)^* \cdot (\varepsilon+0) = 0^*$	$1 + (\varepsilon+0)\cdot(\varepsilon+0)^* \cdot 1 = 0^* \cdot 1$	$\emptyset$	$\varepsilon + 0 + 1$
2	not needed	$0^* \cdot 1 + (0^* \cdot 1)\cdot(\varepsilon+0+1)^* \cdot (\varepsilon+0+1) = 0^* \cdot 1 \cdot (0+1)^*$	not needed	not needed

$\mathcal{L}(A) =$

$\mathcal{L}(E_{12}) =$

$\mathcal{L}(E_{12}^{2}) =$

$E_A = E_{12} = E_{12}^{2} = 0^* \cdot 1 \cdot (0+1)^*$

OPTIONAL

Theorem (DFA $\longrightarrow$ R.E.)

For every DFA A there is a R.E. E_A s.t. $\mathcal{L}(E_A) = \mathcal{L}(A)$

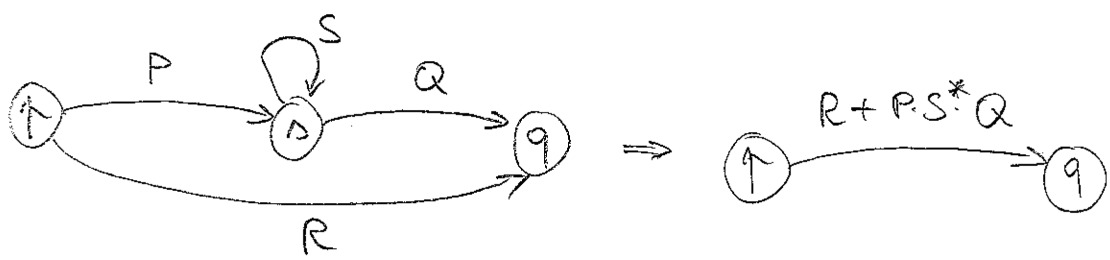
Proof sketch: We show how to construct E_A by eliminating states of A.

Consider the elimination of a state s:

If there was a path from state p_i to state q over s, after eliminating s the path does no longer exist

$\Rightarrow$ we have to compensate for that

We add a regular expression "connecting" p and q and capturing the missing path.

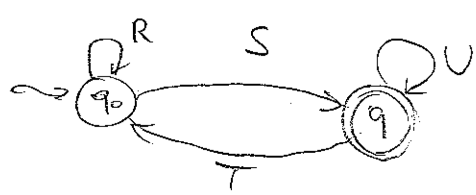


We can eliminate in this way all states except initial and final states:

Strategy:

a) For each final state q, eliminate all states except q, q_0

b) If $q \neq q_0$, we are left with:



The corresponding R.E. is $(R + S \cdot U^* \cdot T)^* \cdot S \cdot U^*$

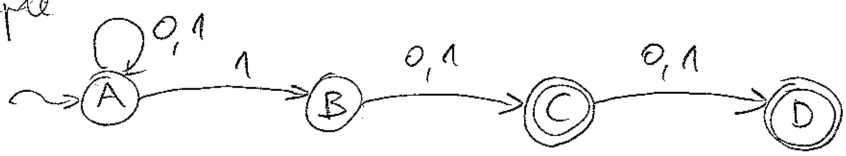
c) If $q_0 \in F$, we must eliminate all states except q_0.

We are left with

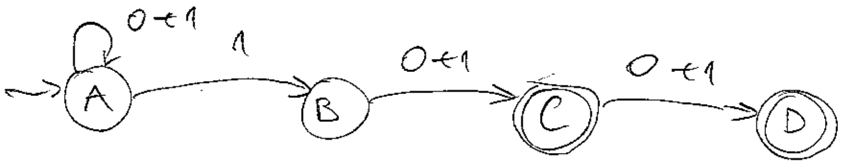
The corresponding R.E. is R^*

d) We take the union of all derived R.E.s.

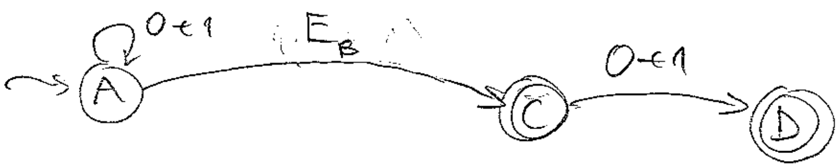
Example



We view all edge labels as R.E.s (missing labels mean $\emptyset$)



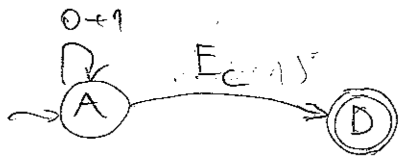
Eliminate B:



$$E_B = \emptyset + 1\cdot\emptyset^*\cdot(0+1) = 1\cdot\emptyset^*\cdot(0+1) = 1\cdot(0+1)$$

Eliminate

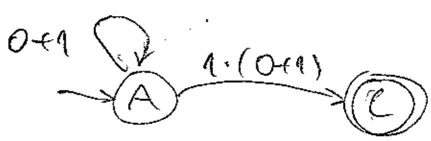
a) C:



$$E_C = \emptyset + 1\cdot(0+1)\cdot\emptyset^*\cdot(0+1) = 1\cdot(0+1)\cdot(0+1)$$

$$E_1 = (0+1)^*\cdot E_C = (0+1)^*\cdot 1\cdot(0+1)\cdot(0+1)$$

b) D:



$$E_2 = (0+1)^*\cdot 1\cdot(0+1)$$

$$E = E_1 + E_2 = (0+1)^*\cdot 1\cdot(0+1)\cdot(0+1) + (0+1)^*\cdot 1\cdot(0+1)$$

$$= (0+1)^*\cdot 1\,(0+1)\,(\varepsilon+0+1)$$