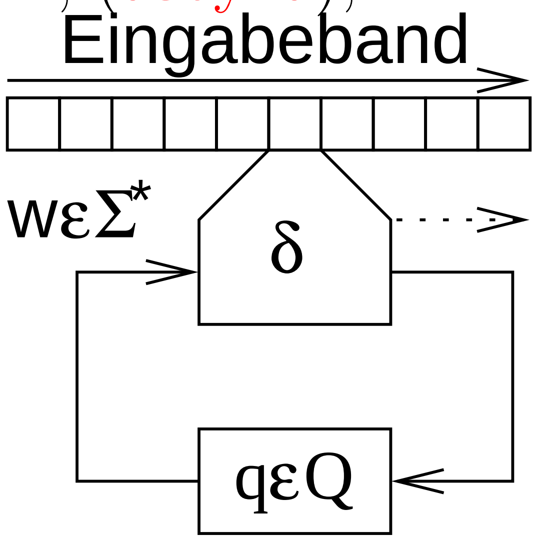
Регулярни езици

- ⇝ (Не)детерминистични крайни автомати
- ⇝ Регулярни изрази
- □ Нерегулярни езици
- □ Минимален автомат
- □ Разрешими проблеми

1.1.1 (Детерминистични) крайни автомати

Един детерминистичен краен автомат $A = (Q, \Sigma, \delta, s, F)$ се състои от:
(детерминистичен краен автомат=DFA)

- Q, крайно множество от състояния;
- Σ, крайно множество от (входни) символи, (азбука);
- $\delta : Q \times \Sigma \to Q$, функция на прехода;
- $s \in Q$, начално състояние;
- $F \subseteq Q$, множество от крайни състояния.



Как работи един краен автомат?

Разширяваме функцията δ върху думи:

$\hat{\delta}(q,\varepsilon) = q$

$\hat{\delta}(q,aw) = \hat{\delta}(\delta(q,a),w)$

$A = (Q,\Sigma,\delta,s,F)$ разпознава езика

$L(A) := \left\{ w \in \Sigma^* : \hat{\delta}(s,w) \in F \right\}$

Еквивалентна дефиниция:

$\hat{\delta}(q,\varepsilon) = q$

$\hat{\delta}(q,wa) = \delta(\hat{\delta}(q,w),a)$

Свойство: $\forall q,u,v : \hat{\delta}(q,uv) = \hat{\delta}(\hat{\delta}(q,u),v)$.

Д-во: индукция по u.

1. $u = \varepsilon$.

$\hat{\delta}(q,uv) = \hat{\delta}(q,v)$.

$\hat{\delta}(\hat{\delta}(q,u),v) = \hat{\delta}(\hat{\delta}(q,\varepsilon),v) = \hat{\delta}(q,v)$.

2. $u = au'$.

$\hat{\delta}(q,au'v) \overset{\text{деф}}{=} \hat{\delta}(\delta(q,a),u'v) \overset{\text{ип}}{=} \hat{\delta}(\hat{\delta}(\delta(q,a),u'),v) \overset{\text{деф}}{=}$
$\hat{\delta}(\hat{\delta}(q,au'),v)$.

Интерпретация с ориентиран граф

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

$G_A = (Q, E)$,

всяка дъга $e = (q, q') \in E$ има етикет $\ell(e) = a$ ако

$q' = \delta(q, a)$

Мулти-граф!

Лема:

$\forall w \in \Sigma^* : w \in L(A) \Leftrightarrow$

$\exists$път $P = sq_1q_2 \cdots f = s \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_k} f$

където $f \in F, w = a_1a_2 \cdots a_k$.

Терминология:

Под път в A разбираме път в G_A.

Означения за път

$P = sq_1q_2\cdots f$ редица от върхове(състояния)

$P = s \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_k} f$ редица от (директни) преходи

$P = s \stackrel{w}{\Rightarrow} f$, където $w = a_1\cdots a_k$, P е с етикет w

$q \stackrel{*}{\Rightarrow} r$ има път от q до r, т.е. r е достижим от q

По дефиниция $s \stackrel{*}{\Rightarrow} s$ (рефлексивност)

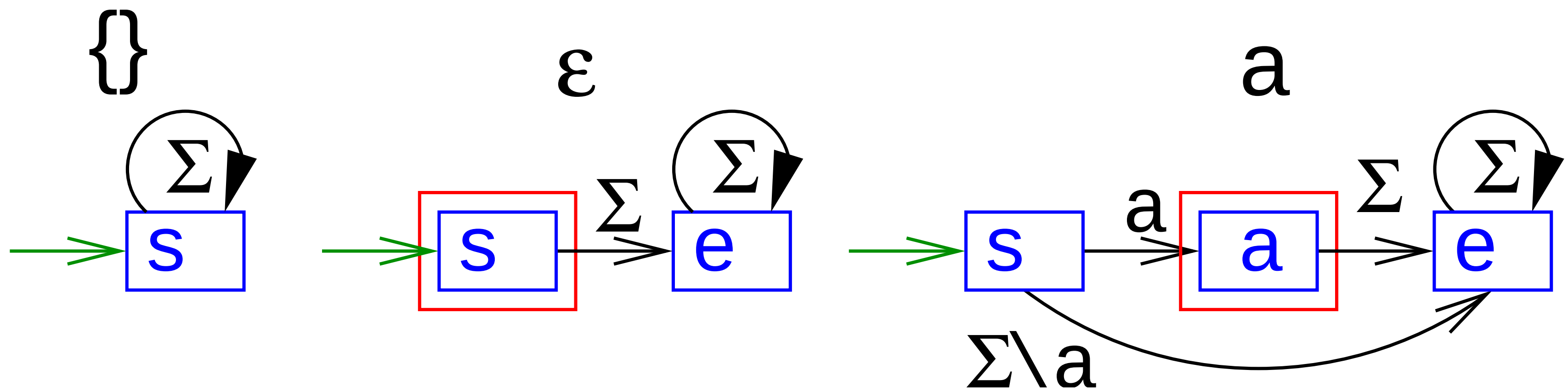
Свойство: $\hat{\delta}(q,w) = r \iff q \stackrel{w}{\Rightarrow} r$.

Дуалност: Граматики ↔ Машини

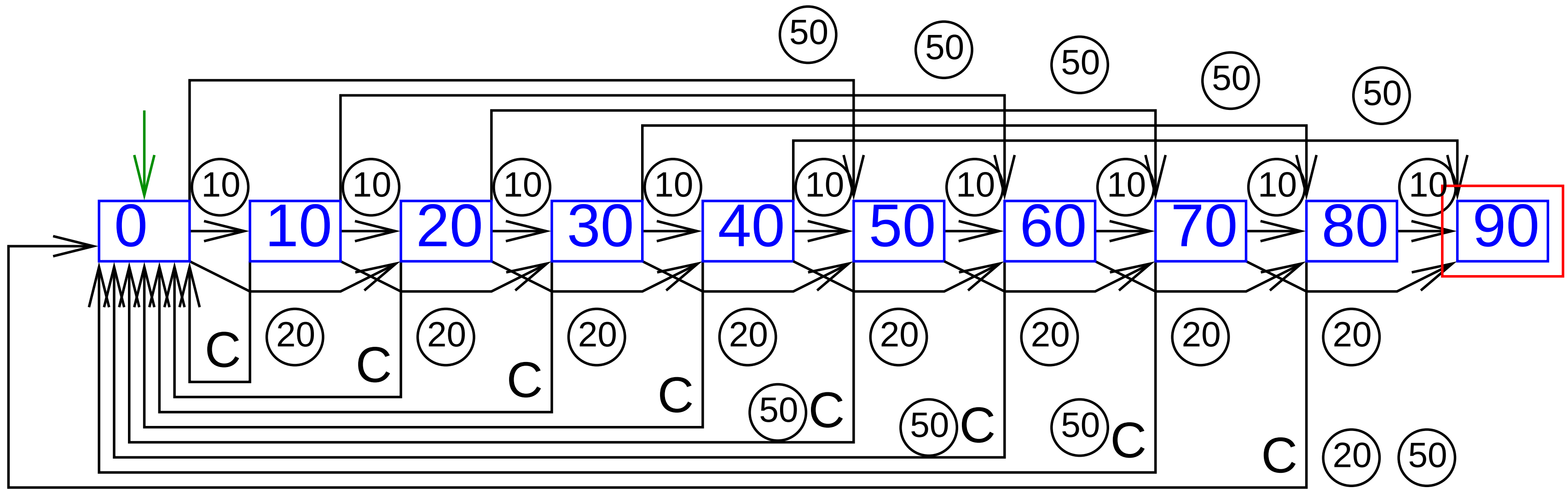
Граматиките генерират думи.

Машините приемат/разпознават думи.

Крайни автомати: Един прост пример



Пример: Билетен автомат



Конфигурация $(q,w) \in \Sigma^* \times Q$.

Дефиниция: $(q,w) \vdash_A (p,u) \iff w = au \ \&\ \delta(q,a) = p$.

Дефиниция:

(рефлексивно и транзитивно затваряне на $\vdash_A$)

$(q,\varepsilon) \vdash^* (q,\varepsilon)$.

$(q,aw) \vdash_A^* (p,u) \iff (q,aw) \vdash_A (r,w) \ \&\ (r,w) \vdash_A^* (p,u)$.

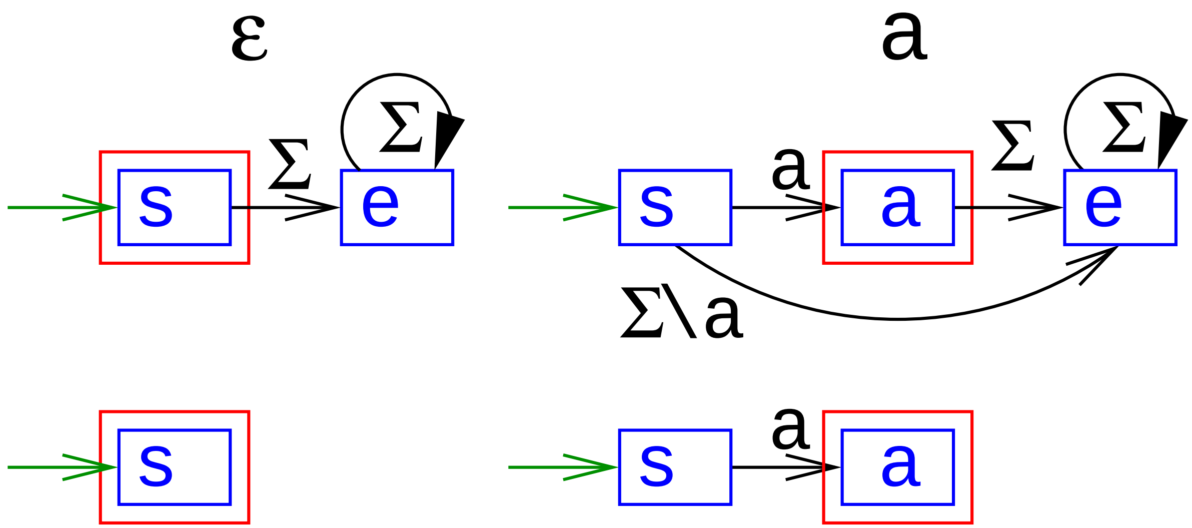
Твърдение: $w \in L(A) \iff \hat{\delta}(s,w) = f \in F \iff s \overset{w}{\Rightarrow} f \iff (s,w) \vdash_A^* (f,\varepsilon)$.

Тоталност

Един автомат е тотален, ако от всяко състояние има преход с всяка буква от Σ.

Често не даваме всички стойности на δ, т.е. автоматът може да не е тотален.

Конвенция: Има винаги error състояние e такова, че $\delta(q,c) = e$ когато не може да разпознаваме повече символи.
$\delta(e,c) = e$ $(\forall c \in \Sigma)$


ε
a
Σ
Σ
s
e
a
Σ
Σ
s
a
e
Σ\a
s
a
s
a

Твърдение:

Езиците, разпознавани от DFA са от Chomsky тип 3

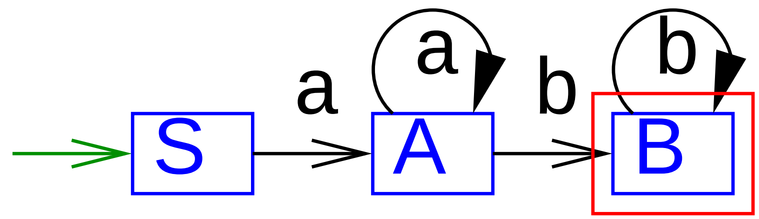
Нека $A = (Z, \Sigma, \delta, S, F)$ е DFA.

Да разгледаме граматиката $G = (Z, \Sigma, P, S)$, където

$$
\begin{aligned}
P = & \left\{ Q \to aQ' : \delta(Q,a) = Q' \right\} \cup \\
& \left\{ Q \to a : \delta(Q,a) = Q' \in F \right\} \cup \\
& \{ S \to \varepsilon : S \in F \} \quad .
\end{aligned}
$$

тогава $L(G) = L(A)$ (ε се елиминира. . .)

Пример: $\{a^n b^m : n \geq 1, m \geq 1\}$



$$G = (\{S, A, B\}, \{a, b\}, P, S)$$

q	c	$\delta(q,c)$	$\in P$
S	a	A	$S \longrightarrow aA$
A	a	A	$A \longrightarrow aA$
A	b	B	$A \longrightarrow bB, A \longrightarrow b$
B	b	B	$B \longrightarrow bB, B \longrightarrow b$

А $\delta(S, b)$?

Езиците, разпознавани от DFA са от Chomsky тип 3

Нека $A = (Z, \Sigma, \delta, S, F)$ е DFA.

Да разгледаме граматиката $G = (Z, \Sigma, P, S)$, където

$$
\begin{aligned}
P = & \left\{Q \to aQ' : \delta(Q,a) = Q'\right\} \cup \\
& \left\{Q \to a : \delta(Q,a) = Q' \in F\right\} \cup \\
& \{S \to \varepsilon : S \in F\} \quad .
\end{aligned}
$$

тогава $L(G) = L(A)$.

Идея: $\exists$ една 1-1 релация между

изводите $S \Rightarrow w_1 A_1 \Rightarrow w_1 w_2 A_2 \Rightarrow \cdots \Rightarrow w_1 w_2 \cdots w_{n-1} A_{n-1} \Rightarrow w$ и

DFA изчисляващите пътища $S \overset{w_1}{\Rightarrow} A_1 \overset{w_2}{\Rightarrow} A_2 \overset{w_3}{\Rightarrow} \cdots \overset{w_n}{\Rightarrow} f \in F$.

Д-во: $L(G) = L(A)$:

Ако $w = \varepsilon$: $\varepsilon \in L(G) \Leftrightarrow S \rightarrow \varepsilon \in P \Leftrightarrow S \in F \Leftrightarrow \varepsilon \in L(A)$

$A = (Z, \Sigma, \delta, S, F)$, $G = (Z, \Sigma, P, S)$ и $P = \{Q \rightarrow aQ' : \delta(Q, a) = Q'\} \cup \{Q \rightarrow a : \delta(Q, a) = Q' \in F\} \cup \{S \rightarrow \varepsilon : S \in F\}$

Д-во (скица) $L(G) = L(A)$ Ако $|w| = n$, $n > 0$:

$w_1 \cdots w_n \in L(G)$

$\Leftrightarrow S \Rightarrow w_1 A_1 \Rightarrow w_1 w_2 A_2 \overset{*}{\Rightarrow} w_1 \cdots w_{n-1} A_{n-1} \Rightarrow w_1 \cdots w_n$

$\Leftrightarrow \{S \to w_1 A_1, A_1 \to w_2 A_2, \ldots, A_{n-2} \to w_{n-1} A_{n-1}, A_{n-1} \to w_n\} \subseteq P$

$\Leftrightarrow \delta(S, w_1) = A_1, \delta(A_1, w_2) = A_2, \ldots, \delta(A_{n-1}, w_n) = A_n \in F$

$\Leftrightarrow \exists$изчислителен път $S \overset{w_1}{\Rightarrow} A_1 \overset{w_2}{\Rightarrow} A_2 \overset{w_3}{\Rightarrow} \cdots A_{n-1} \overset{w_n}{\Rightarrow} A_n \in F$

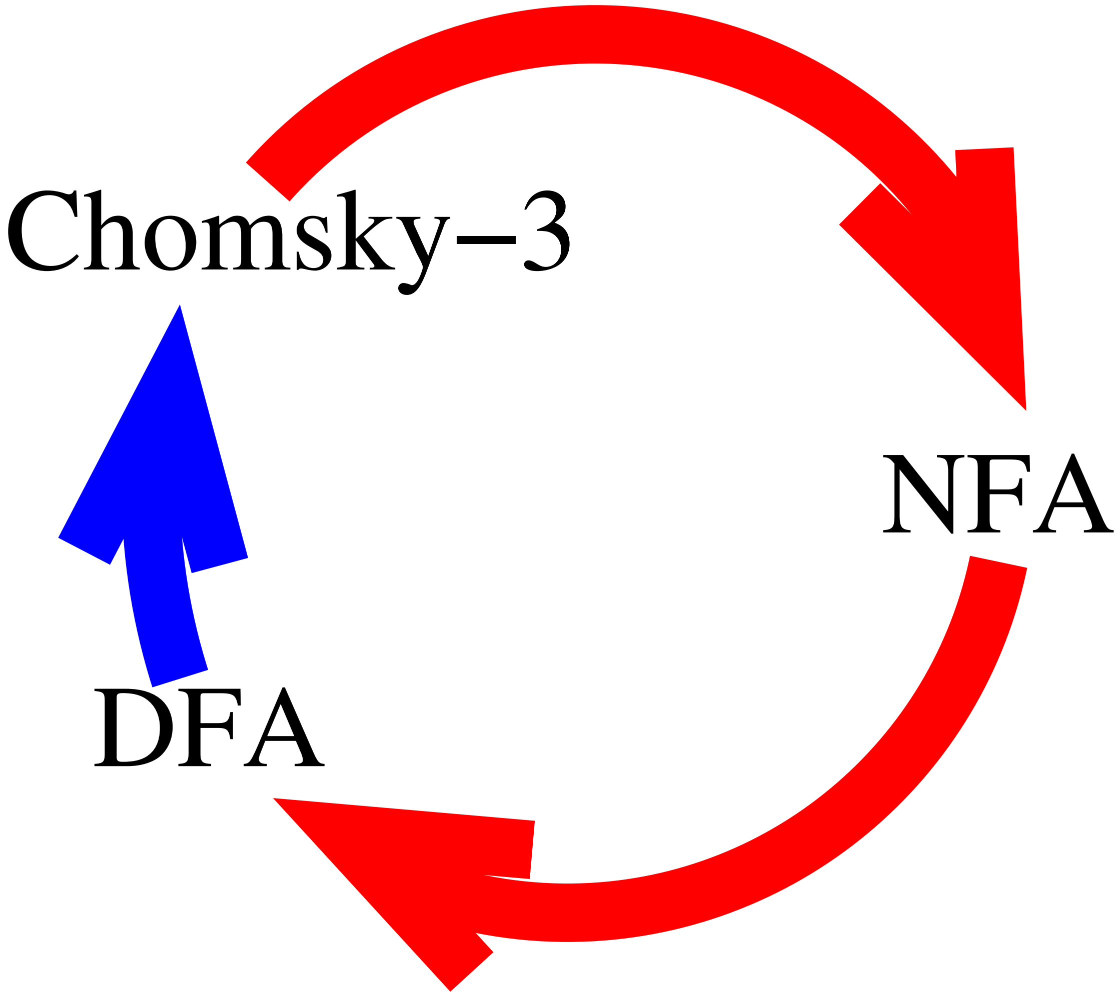
$\Leftrightarrow w_1 \cdots w_n \in L(A)$

(В 2 посоки '$\Leftrightarrow$') ■

винаги се доказва

$A = (Z, \Sigma, \delta, S, F)$, $G = (Z, \Sigma, P, S)$ и $P = \{Q \to aQ' : \delta(Q, a) = Q'\} \cup \{Q \to a : \delta(Q, a) = Q' \in F\} \cup \{S \to \varepsilon : S \in F\}$

Схемата



1.1.2 Недетерминистични крайни автомати NFA

- допускат се повече от един преход от дадено състояние с един символ

Недетерминистичен краен автомат $A = (Q, \Sigma, \delta, s, F)$

- Q, множество от състояния
- Σ, азбука
- $\delta : Q \times \Sigma \to 2^Q$, функция на прехода
- $s \in Q$, начално състояние
- $F \subseteq Q$, крайни състояния

Преходът от q до q' при вход a: $q' \in \delta(q, a)$

повече възможности!

Недетерминистичен краен автомат A

Вариант(еквивалент) — δ като релация.

- Q, множество от състояния
- Σ, азбука
- $\delta \subseteq Q \times \Sigma \times Q$ функция на прехода
- $s \in Q$, начално състояние
- $F \subseteq Q$, крайни състояния

A може да прави преход от q до q' когато $a \in \Sigma$ и $(q, a, q') \in \delta$.

Разширяване на δ

Подмножества от състояния: $\bar{\delta} : 2^Q \times \Sigma \to 2^Q$

$$\bar{\delta}(M,a) := \bigcup_{p \in M} \delta(p,a)$$

Подмножества от състояния и входна дума :
$\hat{\delta} : 2^Q \times \Sigma^* \to 2^Q$

$$\hat{\delta}(M,\varepsilon) := M$$
$$\hat{\delta}(M,aw) := \hat{\delta}(\bar{\delta}(M,a),w)$$

$$L(A) := \left\{ w \in \Sigma^* : \hat{\delta}(\{s\},w) \cap F \neq \emptyset \right\}$$

Интерпретация с (мулти) граф за $L(A)$

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

$G_A = (Q, E)$ δ - фунция

всяка $e = (q, q') \in E$ е с етикет $\ell(e) = a$ ако $q' \in \delta(q, a)$

$G_A = (Q, \delta)$ δ - релация

етикетираме дъгата e при $\ell(e) = a$ като (q, a, q').

"'Мулти"'= паралени дъги са разрешени(различни етикети)

$w \in L(A) \Leftrightarrow \exists$ път $P = s \overset{a_1}{\Rightarrow} q_1 \overset{a_2}{\Rightarrow} \cdots \overset{a_k}{\Rightarrow} f$ in A (в G(A)) :

$$f \in F \wedge w = a_1 a_2 \cdots a_k$$

Пътят $P = s \overset{w}{\Rightarrow} f$ от s до крайно състояние f, с етикет w наричаме приемащ за w.

Лема: $\hat{\delta}(M,w) = \left\{q \in Q : \exists p \in M : p \overset{w}{\Rightarrow} q\right\}.$

Д-во с индукция по $|w|$:

$\hat{\delta}(M,\varepsilon) = M$

$n \rightsquigarrow n+1$:

$\hat{\delta}(M,aw) = \hat{\delta}(\bar{\delta}(M,a),w)$

$= \left\{r \in Q : (\exists q \in \bar{\delta}(M,a)) : q \overset{w}{\Rightarrow} r\right\}$ (ИП)

$= \left\{r \in Q : (\exists p \in M)(\exists q \in \delta(p,a)) : q \overset{w}{\Rightarrow} r\right\}$ (Деф. $\bar{\delta}$)

$= \left\{r \in Q : (\exists p \in M) : p \overset{aw}{\Rightarrow} r\right\}$ (Интерпретация с граф.)

$\square$

Следствие: $L(A) = \left\{w \in \Sigma^* : \exists f \in F : s \overset{w}{\Rightarrow} f\right\}$

Дефиниция: $(q,w) \vdash_A (p,u) \iff w = au \;\&\; p \in \delta(q,a)$.

Означение:

($\vdash^*$ е рефлексивното и транзитивно затваряне на $\vdash_A$)

$(q,\varepsilon) \vdash^* (q,\varepsilon)$.

$(q,aw) \vdash_A^* (p,u) \iff (q,aw) \vdash_A (r,w) \;\&\; (r,w) \vdash_A^* (p,u)$.

Свойства

1. Ако $r \in \delta(q,a)$ и $(r,w) \vdash_A *(p,\varepsilon)$, то $(q,aw) \vdash_A^* (p,\varepsilon)$.

2. $\hat{\delta}(\{q\},w) = \left\{p \in Q : q \overset{w}{\Rightarrow} p\right\} = \{p \in Q : (q,w) \vdash_A^* (p,\varepsilon)\}$.

3. $\hat{\delta}(\{q\},uv) = \bigcup_{r \in \hat{\delta}(\{q\},u)} \hat{\delta}(\{r\},v) = \hat{\delta}(\hat{\delta}(\{q\},u),v)$.

4. $w \in L(A) \iff \hat{\delta}(\{s\},w) \cap F \neq \emptyset \iff (\exists f \in F : s \overset{w}{\Rightarrow} f) \iff (\exists f \in F)(q,w) \vdash_A^* (f,\varepsilon)$.

NFA$\rightarrow$DFA

Даден: NFA $A=(Q,\Sigma,\delta,s,F)$

Теорема: (Детерминизация на NFA)[Рабин, Скот 1959]
DFA $A' := (2^Q,\Sigma,\bar{\delta},\{s\},\{M\subseteq Q : M\cap F\neq\emptyset\})$ разпознава $L(A)$.

Упражнение: Дайте алгоритъм, който по даден NFA A и дума w да изчислява $\hat{\delta}(\{s\},w)$ за време $\mathcal{O}(|w|\cdot|\delta|)$. Тук $|\delta|$ е броят на преходите от вида $p\in\delta(q,a)$, достатъчни да дефинираме δ.

Детерминизация на NFA

$A = (Q, \Sigma, \delta, \underline{s}, F)$

$\underline{A'} := (2^Q, \Sigma, \bar{\delta}, \{s\}, F')$, $F' := \{M \subseteq Q : M \cap F \neq \emptyset\}$, където

$\bar{\delta}(M, a) := \bigcup_{p \in M} \delta(p, a)$

Твърдим: $L(A') = L(A)$

Д-во: Първо с индукция по w проверяваме, че

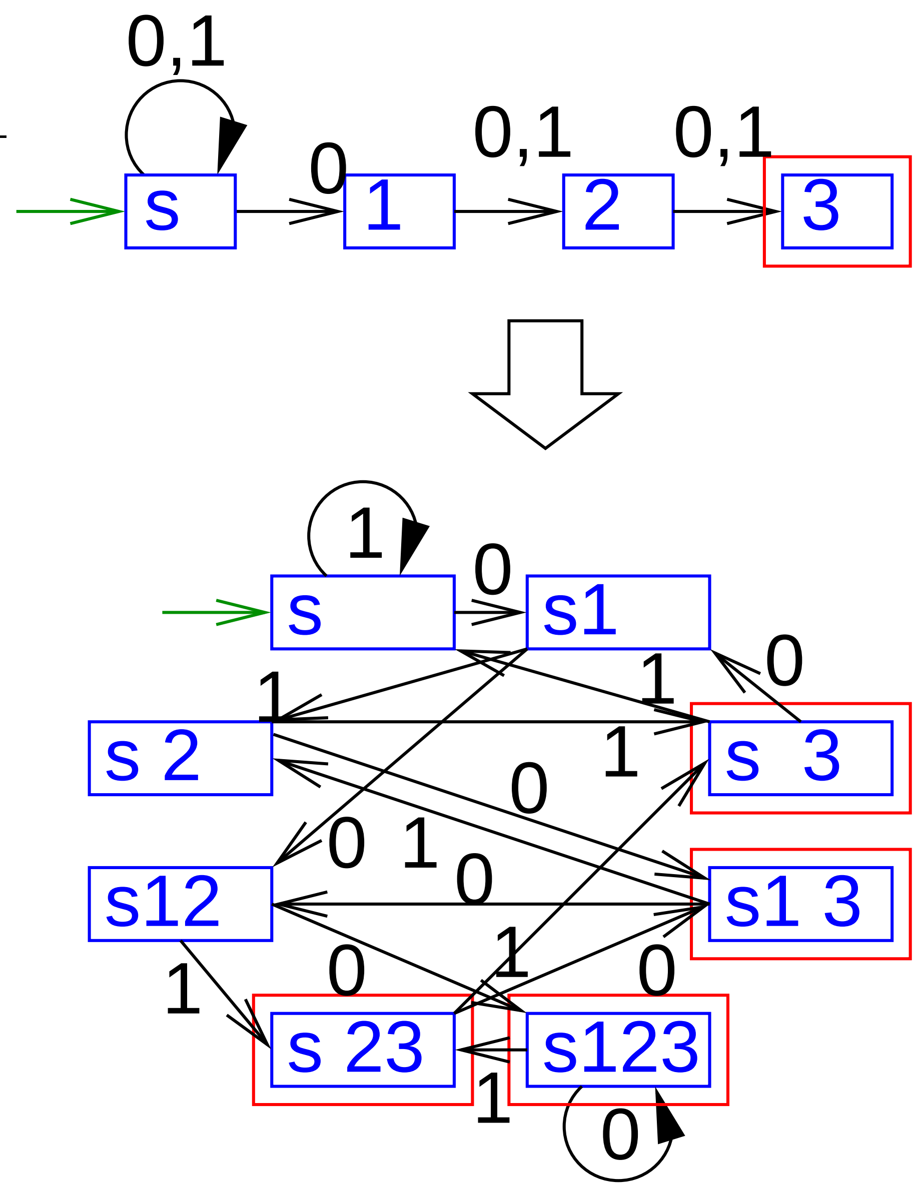
$\hat{\bar{\delta}}(\{s\}, w) = \hat{\delta}(\{s\}, w)$. Тогава

$$
\begin{aligned}
L(A) &= \left\{ w \in \Sigma^* : \hat{\delta}(\{s\}, w) \cap F \neq \emptyset \right\} && \text{Деф. L(A)} \\
&= \left\{ w \in \Sigma^* : \hat{\delta}(\{s\}, w) \in F' \right\} && \text{Деф. F'} \\
&= L(A') && \text{Деф. L(A')}
\end{aligned}
$$

($\hat{\delta}$ играе двойна роля!) $\square$

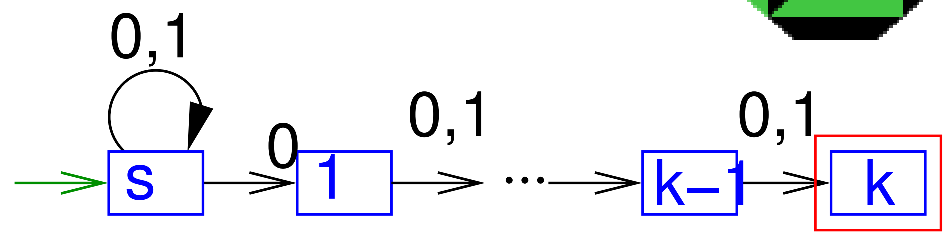
Пример

q	$\bar{\delta}(q,0)$	$\bar{\delta}(q,1)$
s	$s,1$	s
$s,1$	$s,1,2$	$s,2$
$s,2$	$s,1,3$	$s,3$
$s,3$	$s,1$	s
$s,1,2$	$s,1,2,3$	$s,2,3$
$s,1,3$	$s,1,2$	$s,2$
$s,2,3$	$s,1,3$	$s,3$
$s,1,2,3$	$s,1,2,3$	$s,2,3$



По-общ пример

A:



Твърдение:

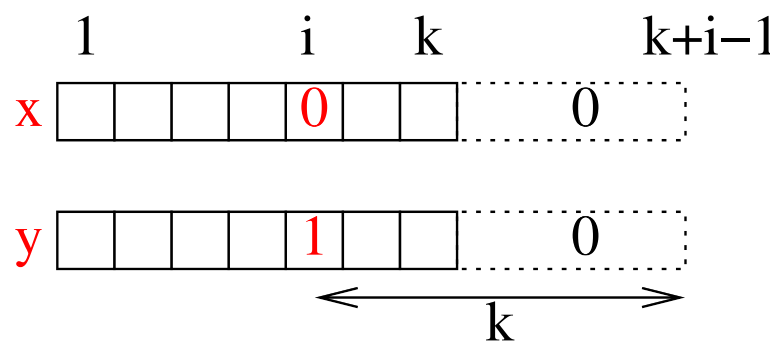
$\nexists$DFA $A' = (Q, \Sigma, \delta, s, F) : L(A') = L(A) \wedge (|Q| < 2^k)$

Д-во: Да предположим, че: $\exists A'$ и $|Q| < 2^k$

$\longrightarrow \exists x \neq y \in \{0,1\}^k : \hat{\delta}(s,x) = \hat{\delta}(s,y)$(Принцип на Дирихле)

където i: $x[i] \neq y[i]$,

Нека $x[i] = 0$, $y[i] = 1$.



Тогава $x0^{i-1} \in L(A)$

и $y0^{i-1} \notin L(A)$.

Но, $\hat{\delta}(s, x0^{i-1}) = \hat{\delta}(\hat{\delta}(s,x), 0^{i-1})$

$= \hat{\delta}(\hat{\delta}(s,y), 0^{i-1}) = \hat{\delta}(s, y0^{i-1})$.

Така или и двете думи $x0^{i-1}$ и $y0^{i-1}$ се приемат, или и двете не се приемат. Противоречие. $\square$

Прилагане на алгоритъма за детерминизация

Разглеждаме само подмножествата достижими от $\{s\}$:

Q':= $\{\{s\}\}$ // състояния на A'

Queue todo:= Q'

while $\exists M \in$todo do

todo:= todo$\setminus M$

foreach $a \in \Sigma$ do

if $M' = \bar{\delta}(M, a) \notin Q'$ then

insert M' into Q'

insert M' into todo

Често $|Q'| \ll 2^{|Q|}$!

Тип-3→NFA

Нека $G = (V, \Sigma, P, S)$ е гарматика от тип 3.

Да разгледаме NFA

$A = (V \cup \{f\}, \Sigma, \delta, S, \{f\} \cup \{S : S \to \varepsilon \in P\})$, където

$$\delta = \{(q, a, q') : q \to aq' \in P\} \cup$$
$$\{(q, a, f) : q \to a \in P\}$$

(Релационно означение за δ).

Има 1-1 релация между изводите от вида

$S \Rightarrow w_1 A_1 \Rightarrow w_1 w_2 A_2 \Rightarrow \cdots \Rightarrow w_1 w_2 \cdots w_{n-1} A_{n-1} \Rightarrow w$ in G и

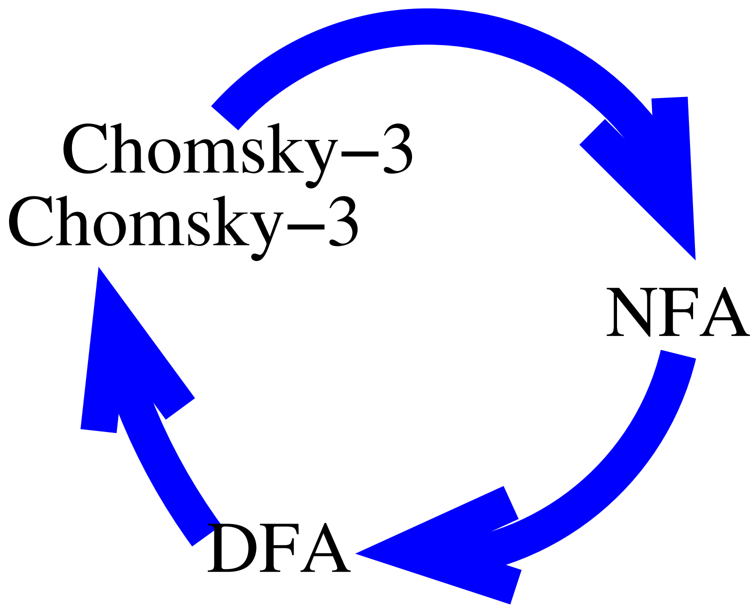
приемащите пътища от вида

$S \xrightarrow{w_1} A_1 \xrightarrow{w_2} A_2 \xrightarrow{w_3} \cdots \xrightarrow{w_n} f$ с A.

Следователно $L(A) = L(G)$.

Еднозначни гарматики от тип-3

Твърдение: $\forall L \in$ type-3 $: \exists$ type-3 граматика с еднозначни изводи.



Д-во : Нека A е DFA и $L(A) = L$.

Съответната граматика от тип-3 за A има еднозначни изводи.

Регулярни езици

- $\emptyset$, $\{\varepsilon\}$ и $\{a\}$ за всяко $a \in \Sigma$ са основни регулярни езици;
- Ако L_1 и L_2 са регулярни, то и $L_1 \cup L_2$ е регулярен;
- Ако L_1 и L_2 са регулярни , то и $L_1.L_2$ е регулярен;
- Ако L е регулярен, то и L^* е регулярен.

Един език е регулярен, ако се получава от основните с помощта на операциите обединение, конкатенация и звезда, приложени краен брой пъти.

Регулярни изрази

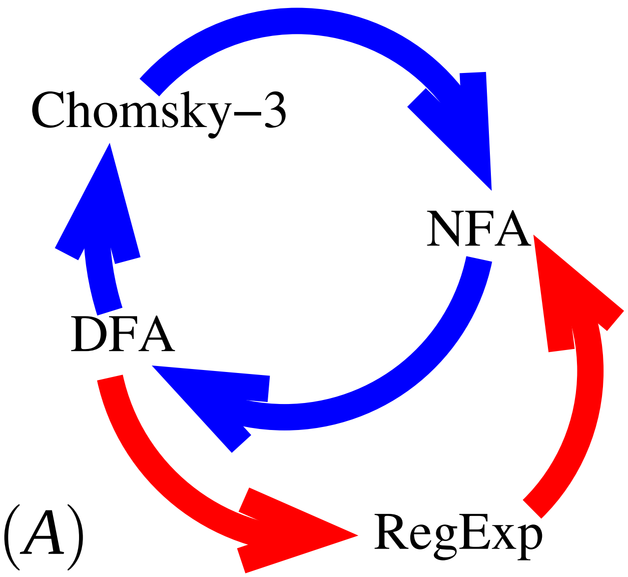
да се докаже:

$\longrightarrow$ регулярен израз α

$\rightsquigarrow$ NFA A и $L(\alpha) = L(A)$

$\longleftarrow$ DFA A

$\rightsquigarrow$ регулярен израз α и $L(\alpha) = L(A)$

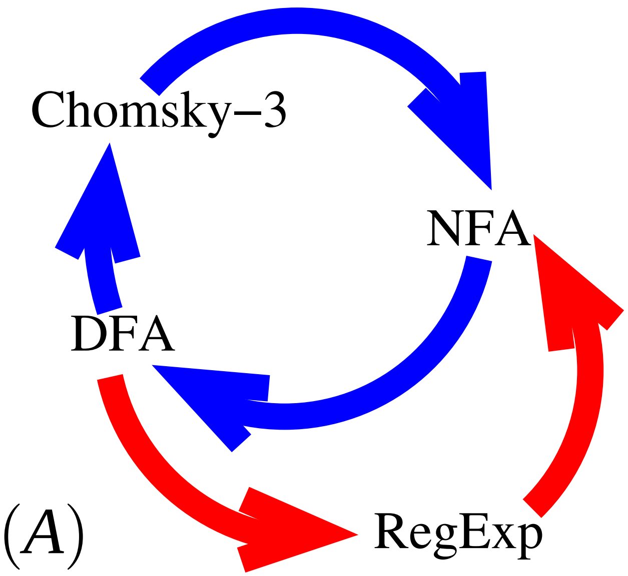


Теорема на Клини:

Регулярни изрази $\Leftrightarrow$ type-3 езици

[Kleene]

Схема



да се докаже:

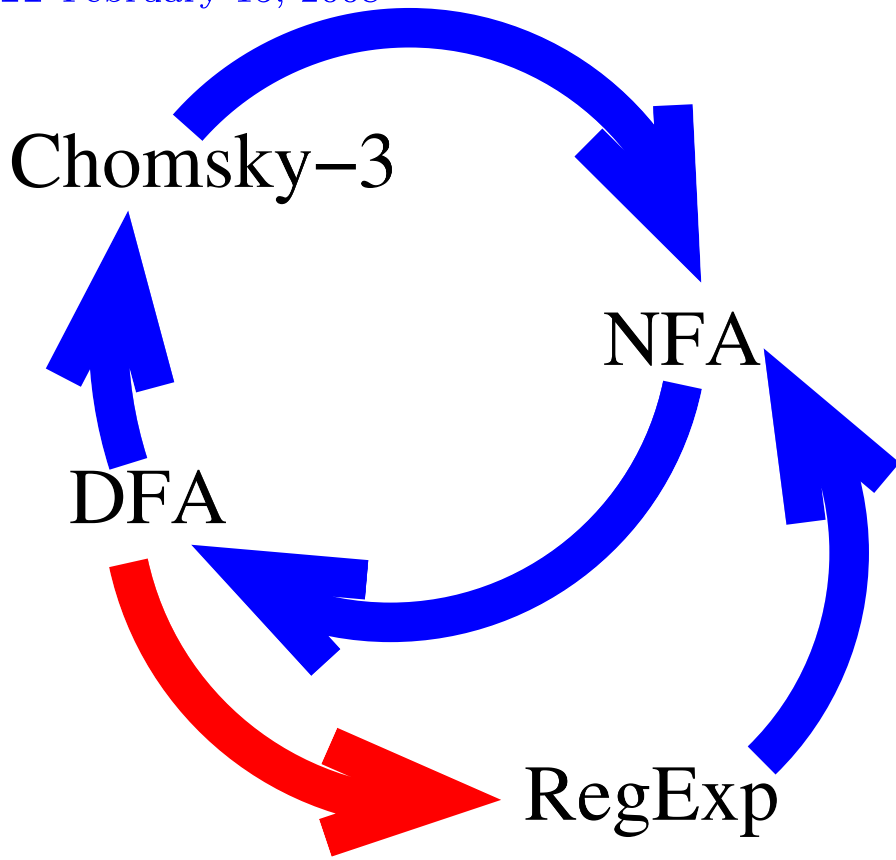
$\rightarrow$ регулярен израз α

$\rightsquigarrow$ NFA Aи $L(\alpha) = L(A)$

$\leftarrow$ DFA A

$\rightsquigarrow$ регулярен израз αи $L(\alpha) = L(A)$

DFA→RegExp



Цел:

- регулярните изрази са универсална връзка

1.1.3 Регулярни изрази

Всеки регулярен израз описва един регулярен език.

израз	описва	забележка
$\emptyset$	$\emptyset$	
ε	$\{\varepsilon\}$	
a	$\{a\}$	$a \in \Sigma$
$\alpha \cup \beta$	$L(\alpha) \cup L(\beta)$	α описва $L(\alpha)$ (синоним: $\alpha \vert \beta$)
$\alpha \cdot \beta$	$L(\alpha) \cdot L(\beta)$	β описва $L(\beta)$
(α)	$L(\alpha)$	
α^*	$L(\alpha)^*$	
α^+	$L(\alpha)^+$	

Конвенции: пропускаме '$\cdot$', пропускаме $L(\cdot)$

Синтаксис (рег. израз) versus

Семантика (рег. езици)

Компютърните програми боравят със синтактични обекти.

Програмна верификация

Ние доказваме от семантична гледна точка, че обектът ще се обаработи коректно.

Пример

- предпоследната цифра е 0: $(0 \cup 1)^* 0 (0 \cup 1)$
- съдържа 10: $(0 \cup 1)^* 10 (0 \cup 1)^*$
- не съдържа 10: $0^* 1^*$
- съдържа 101: $(0 \cup 1)^* 101 (0 \cup 1)^*$
- не съдържа 101: $0^* 1^* \cup (0^* 1^* 100)^* 0^* 1^* 10 (\varepsilon \cup 00^* 1^*)$
- всички цели числа:
  $(\varepsilon \cup + \cup -)(1 \cup \cdots \cup 9)(0 \cup \cdots \cup 9)^* \cup 0$

Затвореност относно регулярните операции

Един език L се нарича автоматен, ако има краен автомат A такъв, че $L(A) = L$.

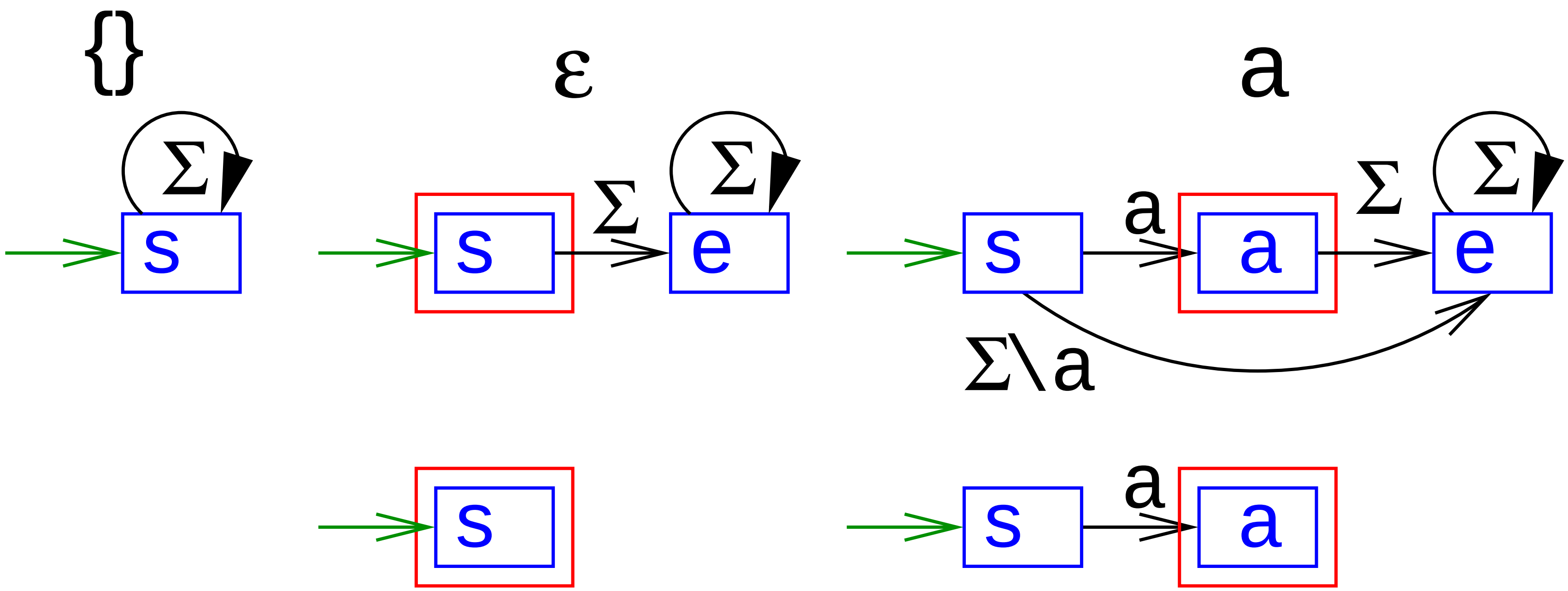
Теорема Всеки регулярен език е автоматен.

Д-во идея:

ще построим автомати, разпознаващи основните езици (основните езици са автоматни)

ще покажем, че регулярните операции запазват автоматността

Базов случай

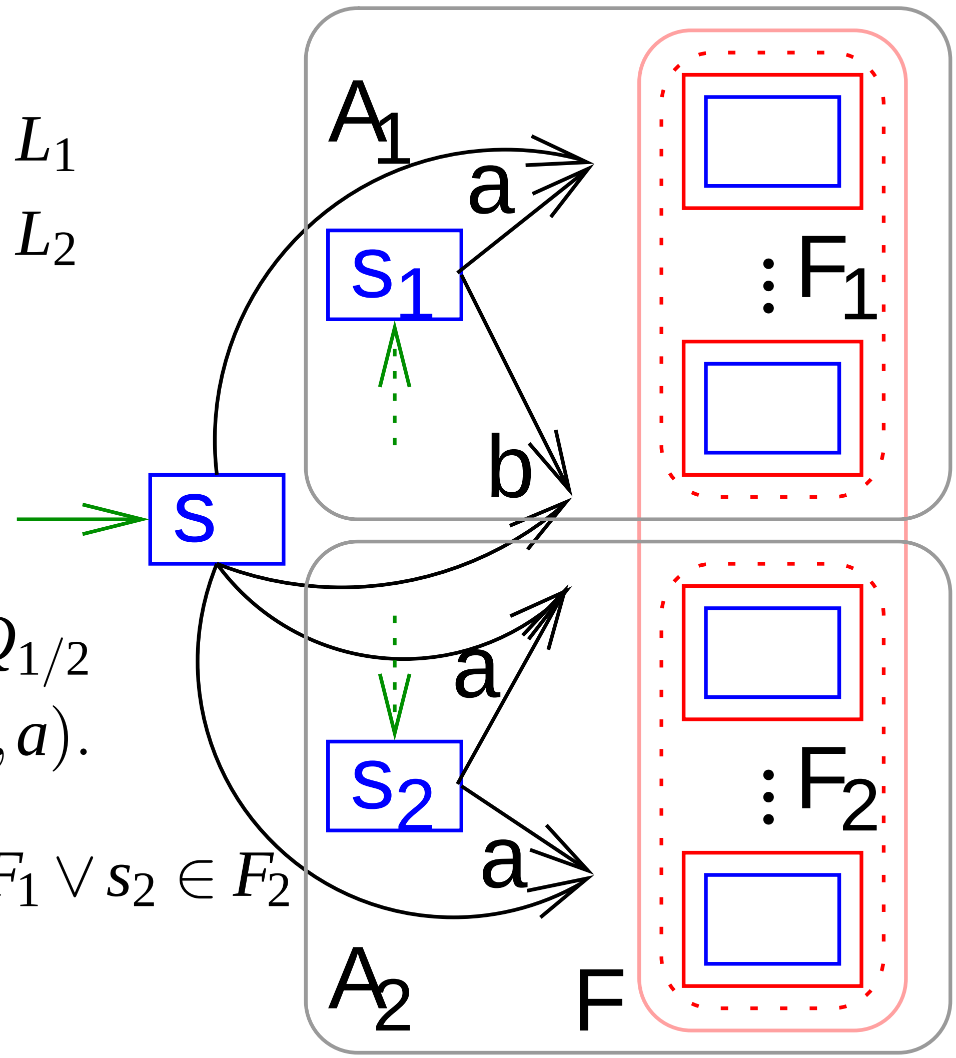


$L_1 \cup L_2$

$A_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ и $L(A_1) = L_1$

$A_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ и $L(A_2) = L_2$

и БОО $Q_1 \cap Q_2 = \emptyset$



$A := (\{s\} \cup Q_1 \cup Q_2, \Sigma, \delta, s, F)$

δ е дефинирана като $\delta_{1/2}$ за $Q_{1/2}$

$\forall a \in \Sigma : \delta(s, a) := \delta(s_1, a) \cup \delta(s_2, a)$.

$$F := \begin{cases} F_1 \cup F_2 \cup \{s\} & \text{ако } s_1 \in F_1 \vee s_2 \in F_2 \\ F_1 \cup F_2 & \text{иначе} \end{cases}$$

Д-во на $L_1 \cup L_2 \subseteq L(A)$

Нека $w \in L_1 = L(A_1)$ (произволна).

Ако $w = \varepsilon$

$\longrightarrow s_1 \in F_1 \longrightarrow s \in F \longrightarrow w \in L(A)$.

Ако $w = ax$:

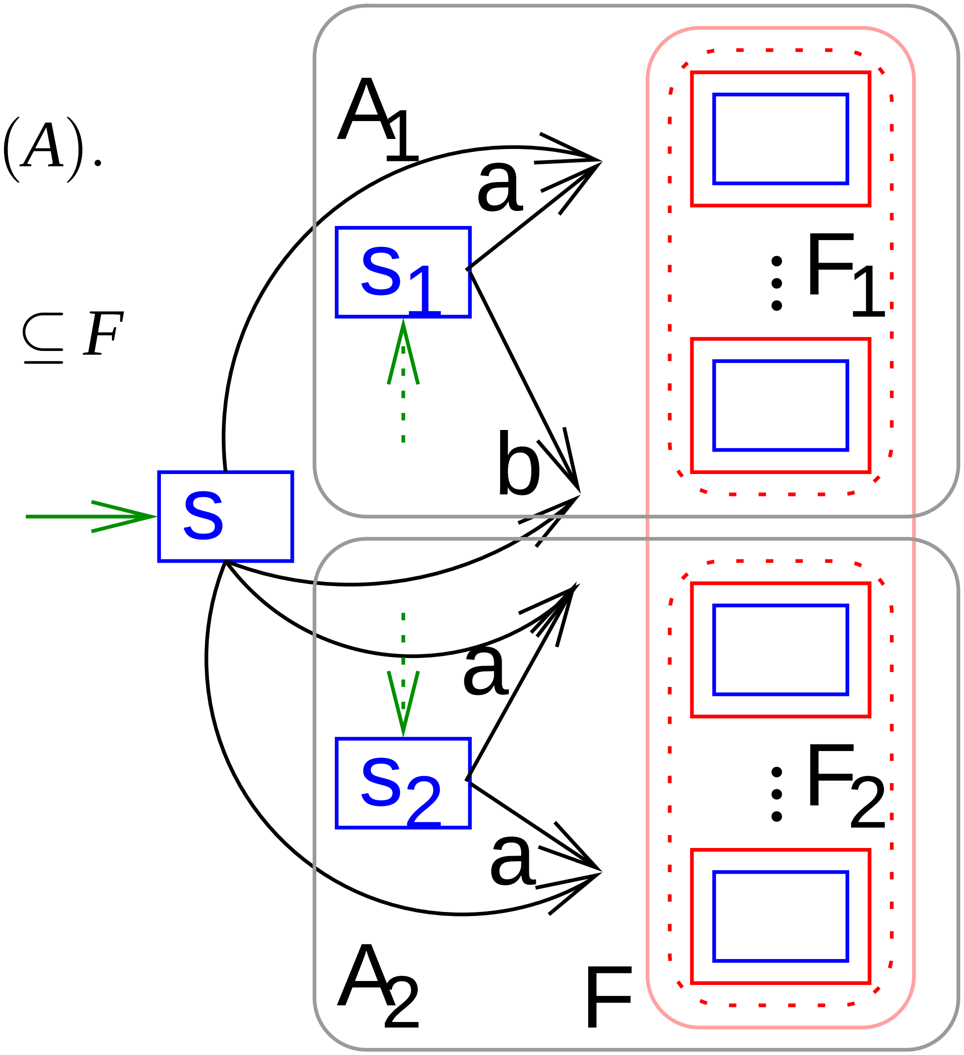
$\longrightarrow \exists$ път $P_1 = s_1 \overset{a}{\Rightarrow} q_1 \overset{x}{\Rightarrow} f_1 \in F_1 \subseteq F$

$\longrightarrow \exists$ път $P = s \overset{a}{\Rightarrow} q_1 \overset{x}{\Rightarrow} f_1 \in F$

$\longrightarrow w \in L(A)$.

$w \in L_2 = L(A_2)$

$\longrightarrow \cdots \rightarrow w \in L(A)$.



Д-во на $L(A) \subseteq L_1 \cup L_2$

Нека w е произволна дума $w \in L(A)$.

Ако $w = \varepsilon \longrightarrow s \in F \longrightarrow s_1 \in F_1 \vee s_2 \in F_2$

$\longrightarrow \varepsilon \in L_1 \vee \varepsilon \in L_2 \longrightarrow \varepsilon \in L_1 \cup L_2$

Ако $w = ax$:

$\longrightarrow \exists$ път $P = s \stackrel{a}{\Rightarrow} q \stackrel{x}{\Rightarrow} f \in F$.

Ако $q = q_1 \in Q_1$:

$\longrightarrow \exists$ път $P_1 = s_1 \stackrel{a}{\Rightarrow} q_1 \stackrel{x}{\Rightarrow} f \in F_1$.

(само състояния,

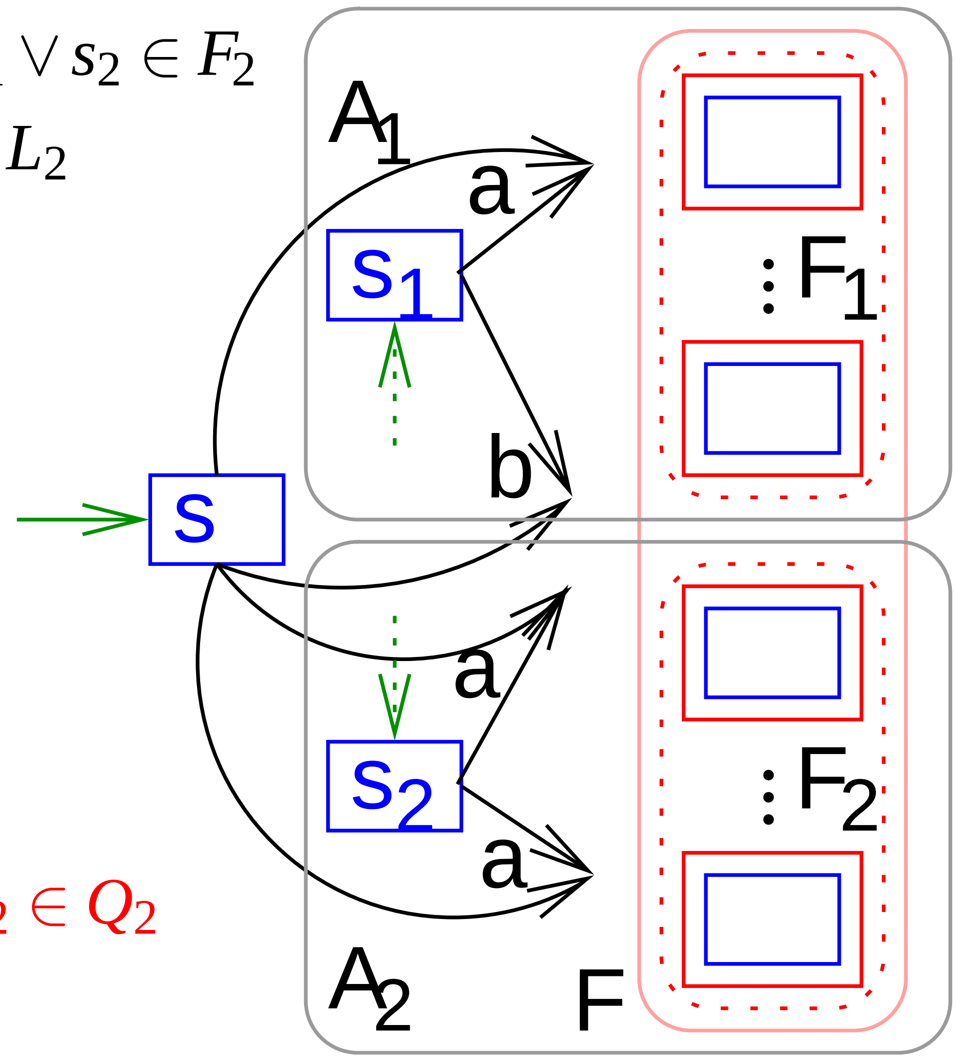
достижими от q_1 са в Q_1.)

$\longrightarrow ax = w \in L_1 \subseteq L_1 \cup L_2$

В противен случай: $\longrightarrow q = q_2 \in Q_2$

$\longrightarrow \exists$ път $P_2 = s_2 \stackrel{a}{\Rightarrow} q_2 \stackrel{x}{\Rightarrow} f \in F_2$.

$\longrightarrow ax = w \in L_2 \subseteq L_1 \cup L_2$



$\square$

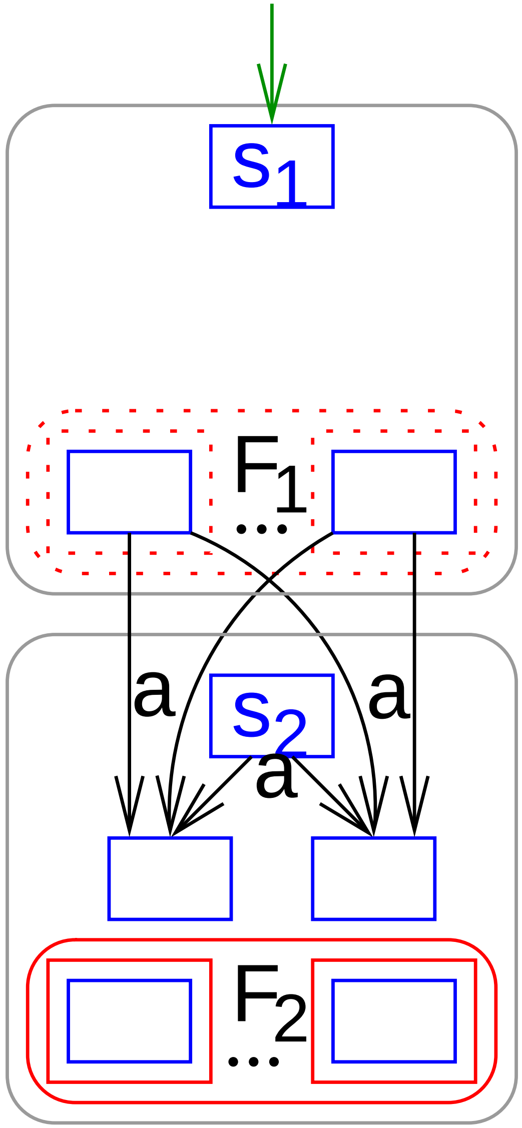
$L_1 \cdot L_2$

$A_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ и $L(A_1) = L_1$
$A_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ и $L(A_2) = L_2$
и $Q_1 \cap Q_2 = \emptyset$

$A := (Q_1 \cup Q_2, \Sigma, \delta, s_1, F), \forall a \in \Sigma :$

$$\delta(q,a) := \begin{cases} \delta_1(q,a) & \text{ако } q \in Q_1 \setminus F_1 \\ \delta_1(q,a) \cup \delta_2(s_2,a) & \text{ако } q \in F_1 \\ \delta_2(q,a) & \text{иначе} \end{cases}$$

$$F := \begin{cases} F_1 \cup F_2 & \text{ако } s_2 \in F_2 \\ F_2 & \text{иначе} \end{cases}$$

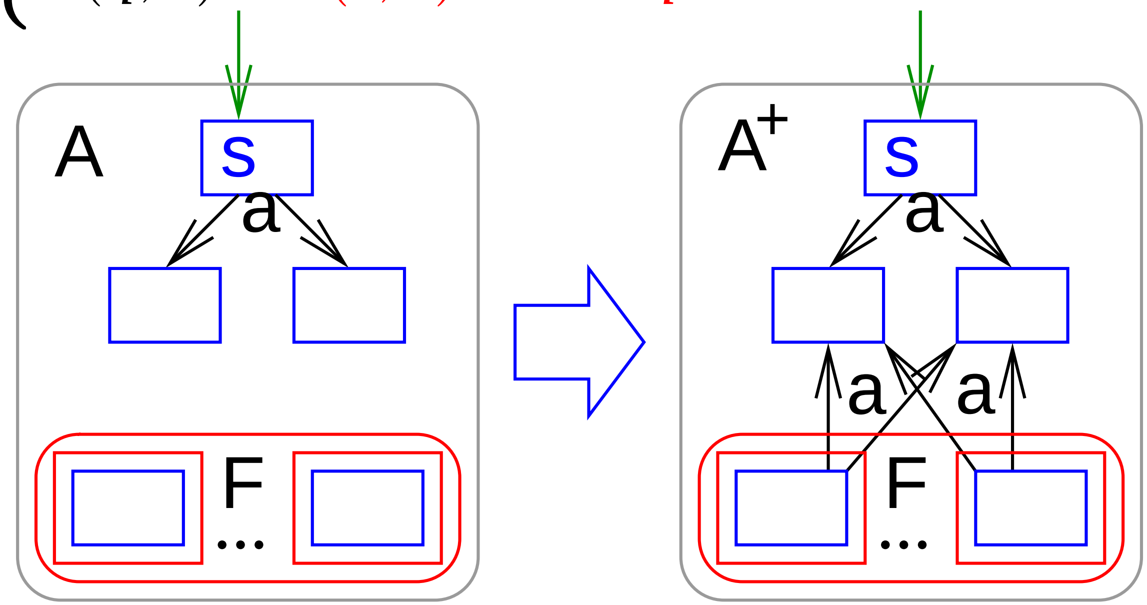


Позитивна обвивка $L^+ = \bigcup_{i \geq 1} L^i$

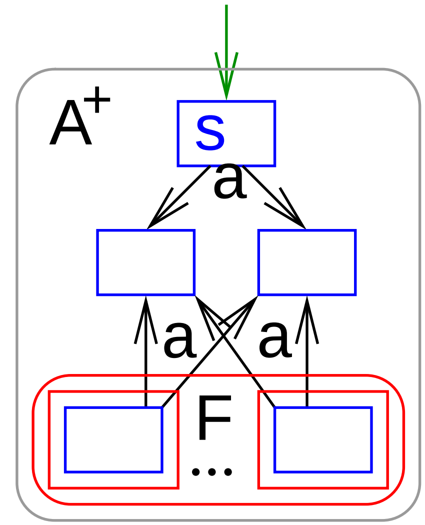
$A = (Q, \Sigma, \delta, s, F)$ и $L(A) = L$

$A^+ := (Q, \Sigma, \delta^+, s, F), \forall a \in \Sigma :$

$$\delta^+(q,a) := \begin{cases} \delta(q,a) & \text{ако } q \in Q \setminus F \\ \delta(q,a) \cup \delta(s,a) & \text{ако } q \in F \end{cases}$$



Д-во на $L(A^+) \subseteq L^+$



Нека $w \in L(A^+)$ е произволна и $w \neq \varepsilon$

Нека $P = s \overset{a_0}{\Rightarrow} q_0 \overset{*}{\Rightarrow} f$ е приемащ път за w.

Декомпозираме P на преходи от вида $f_j \overset{a_j}{\Rightarrow} q_j$

by $q_j \notin \delta(f_j, a_j)$, $j \in 1..i$, $i \geq 0$.

$\longrightarrow f_j \in F$, $q_j \in \delta(s, a_j)$.

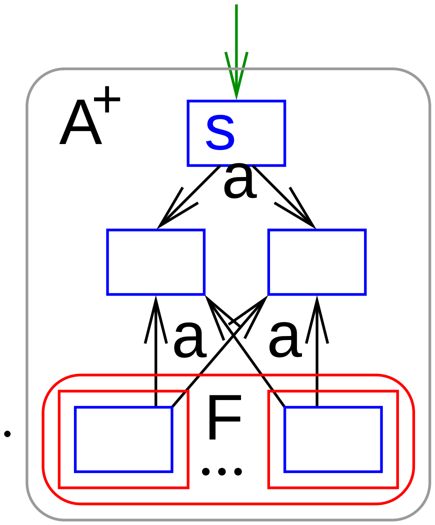
$$P = s \overbrace{\overset{a_0}{\Rightarrow} q_0 \overset{x_0}{\Rightarrow} f_1 \overset{a_1}{\Rightarrow} q_1 \overset{x_1}{\Rightarrow} f_2 \overset{*}{\Rightarrow} f_i \overset{a_i}{\Rightarrow} q_i \overset{x_i}{\Rightarrow}}^{a_0 x_0 a_1 x_1 \cdots a_i x_i = w} f$$

Дефинираме $P_j := s \overset{a_j}{\Rightarrow} q_j \overset{x_j}{\Rightarrow} f_{j+1}$ (с $f_{i+1} := f$).

$\longrightarrow \forall j \in 0..i : P_j$ е един приемащ път A.

$\longrightarrow w \in L^+$ $\square$

Д-во на $L^i \subseteq L(A^+)$ за $i \geq 1$



Нека $w = w_1 \cdots w_i \in L^i$ $(\varepsilon \neq w_i \in L)$.

Да разгледаме $P_j = s \overset{a_j}{\Rightarrow} q_j \overset{x_j}{\Rightarrow} f_j,\ j \in 1..i,\ f_j \in F$,

които свидетелстват за $w_1 \in L, \ldots, w_i \in L$.

$$\longrightarrow P = \overbrace{s \overset{a_1}{\Rightarrow} q_1 \overset{x_1}{\Rightarrow} f_1 \overset{a_2}{\Rightarrow} q_2 \overset{x_2}{\Rightarrow} f_2 \overset{*}{\Rightarrow} f_{i-1} \overset{a_i}{\Rightarrow} q_i \overset{x_i}{\Rightarrow}}^{w} f_i$$

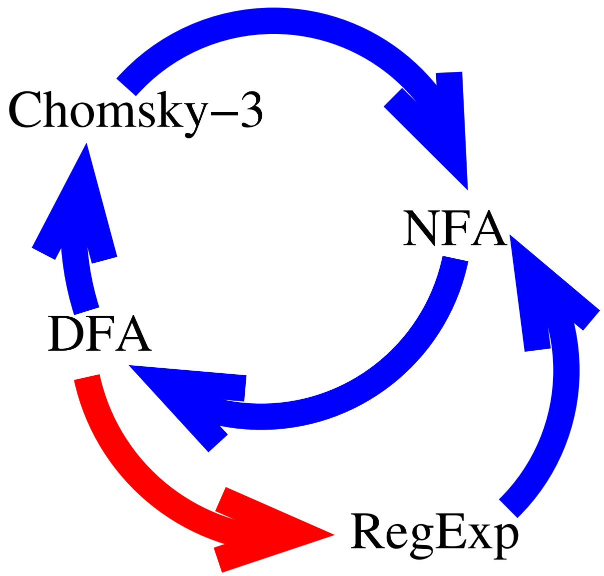
е път в A^+, свидетелстващ за $w \in L(A^+)$.

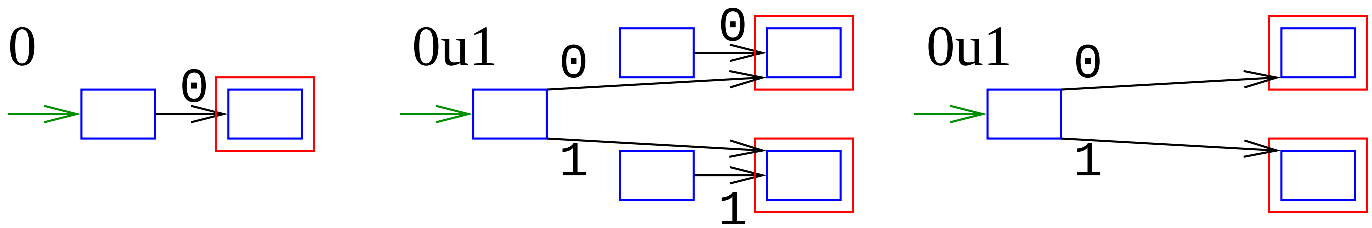
$\longrightarrow w \in L(A^+)$ $\square$

Обвивка на Клини(звезда) L^*

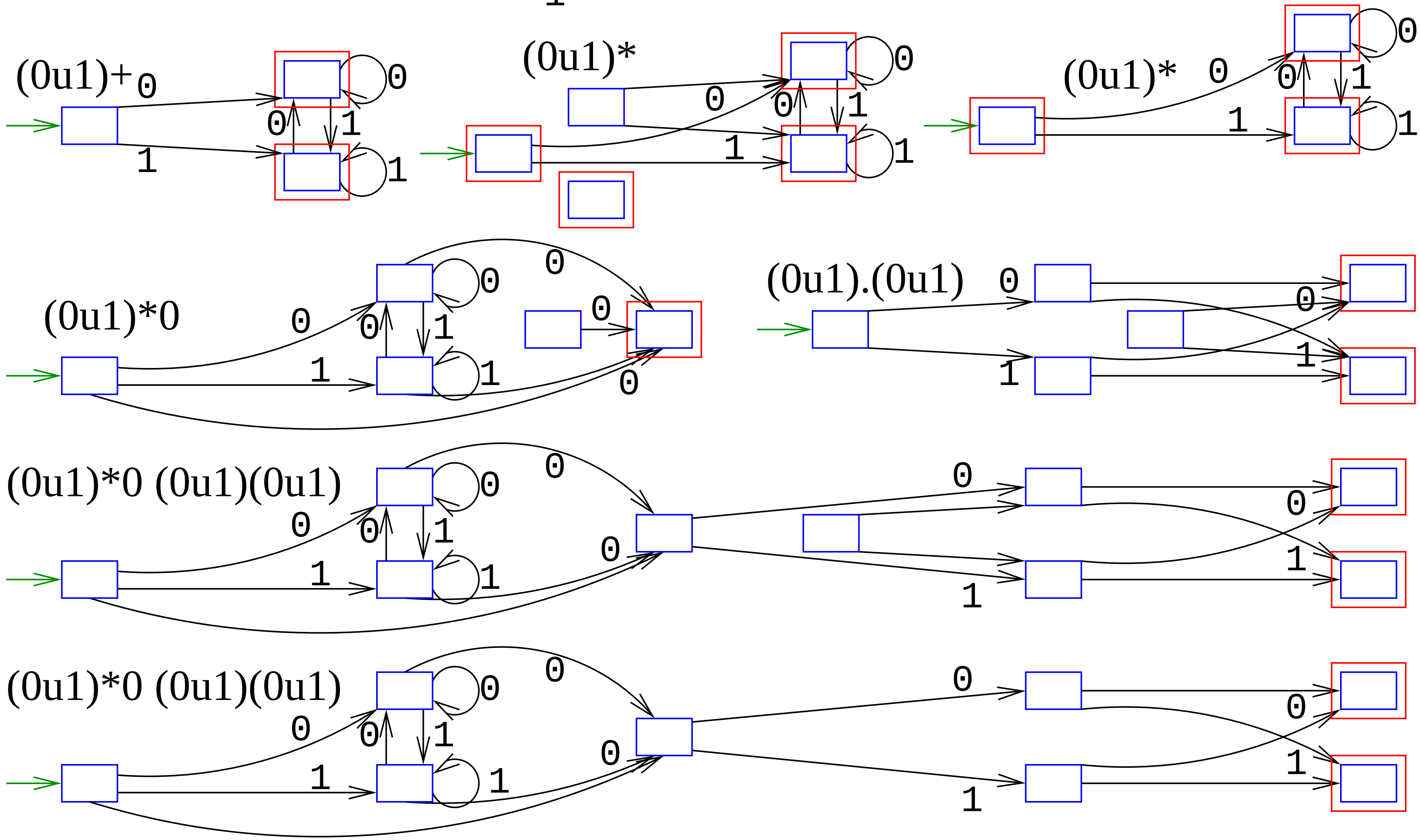
Построяваме автомат за $\varepsilon \cup L^+ = L^*$.

■



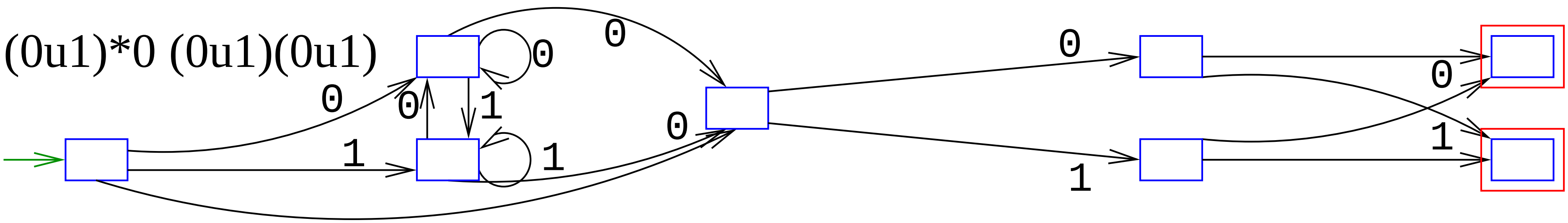


Beispiele

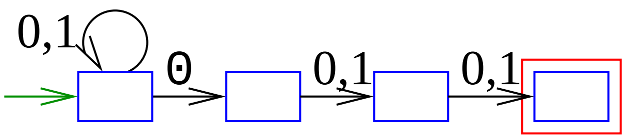


Пример

(0u1)*0 (0u1)(0u1)



(0u1)*0 (0u1)(0u1)



Приложение за търсене в текст

Unix-Tool grep:grep REGULAR-EXPRESSION FILE

- Търсим във всички стрингове във FILE, които са в L(REGULAR-EXPRESSION)
- Много синтаксис: a-g, :alnum:,...
- По-лесно е, ако го транслираме в регулярен израз.
- Бързо приложение - превръщаме в детерминистичен автомат.

Приложение при scanner-(generator), lex, flex

Input: регулярен израз

Output: краен автомат (C code),

Runtime-input: програмата като стрингове

Runtime-output: Програма за token (Пакети) като числа, идентификатори, ключови думи.

- time-critical всеки символ ще се сканира, коментарите се изпускат, десетичните числа се превръщат в двоични ,. . .
- прави представянето стандартно- премахва шпациите ,. . .
- опростява по-нататък синтактичния анализ

Други приложения

- Редактори, например emacs
- Script-езици като Perl
- java.util.regex Library
- C++ Boost.Regex Library
- .net framework
- Parsing за xml документи

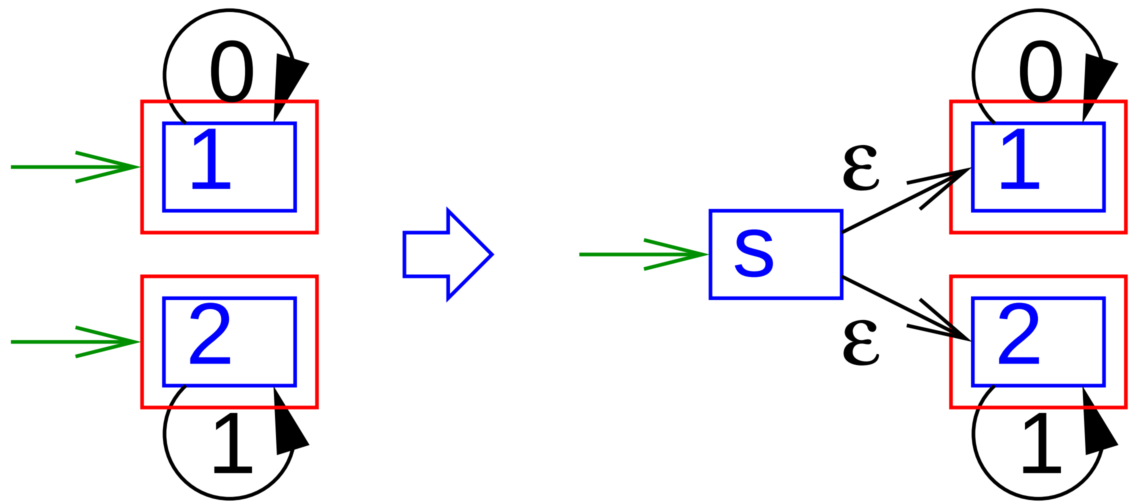
ε-преходи

Разрешени са директни (ε преходи) без да се чете символ.

εNFA

- Q, множество от състояния
- Σ, азбука
- $\delta : Q \times (\Sigma \cup \varepsilon) \to 2^Q$, функция на прехода
- $s \in Q$, начално състояние
- $F \subseteq Q$, крайни състояния

Примери: $0^* \cup 1^*$



RegExp→ εNEA: $L_1 \cup L_2$

$A_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ и $L(A_1) = L_1$

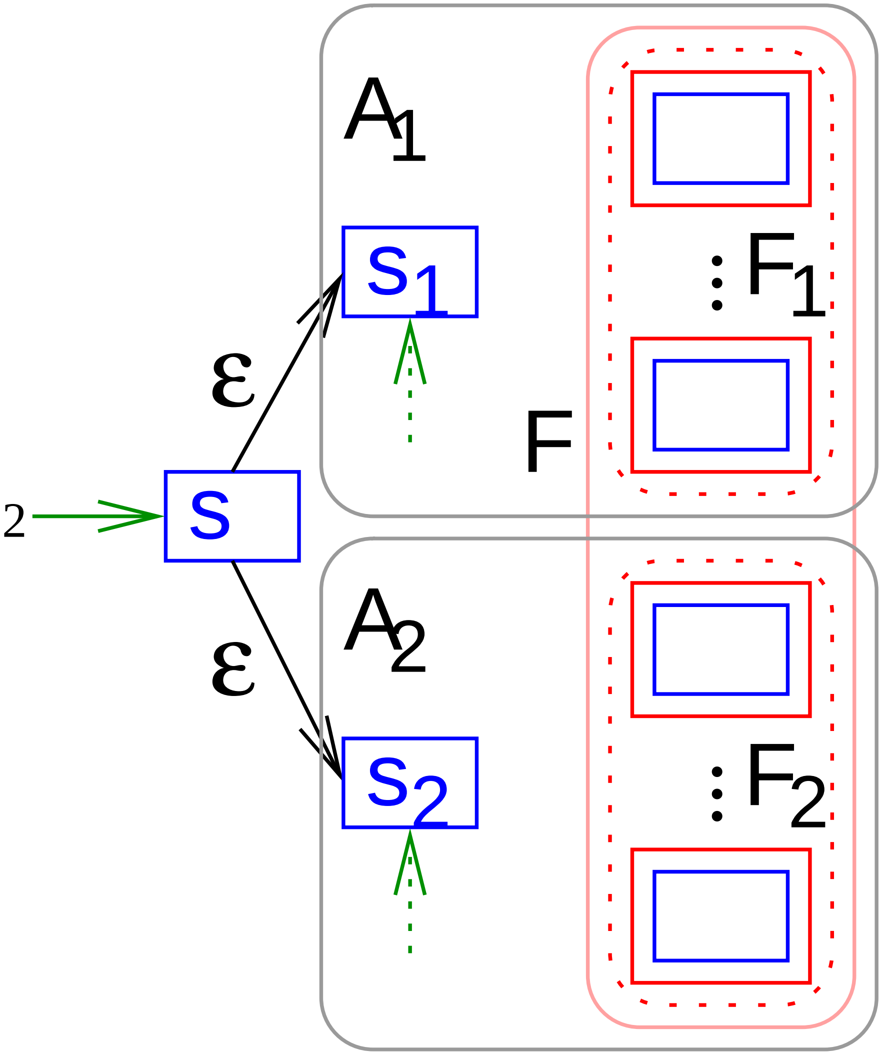
$A_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ и $L(A_2) = L_2$

и $Q_1 \cap Q_2 = \emptyset$

$A := (\{s\} \cup Q_1 \cup Q_2, \Sigma, \delta, s, F_1 \cup F_2)$

δ е дефинирана като $\delta_{1/2}$ on $Q_{1/2}$

ε преходи от s до s_1 и s_2.



$L_1 \cdot L_2$

$A_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$, където $L(A_1) = L_1$

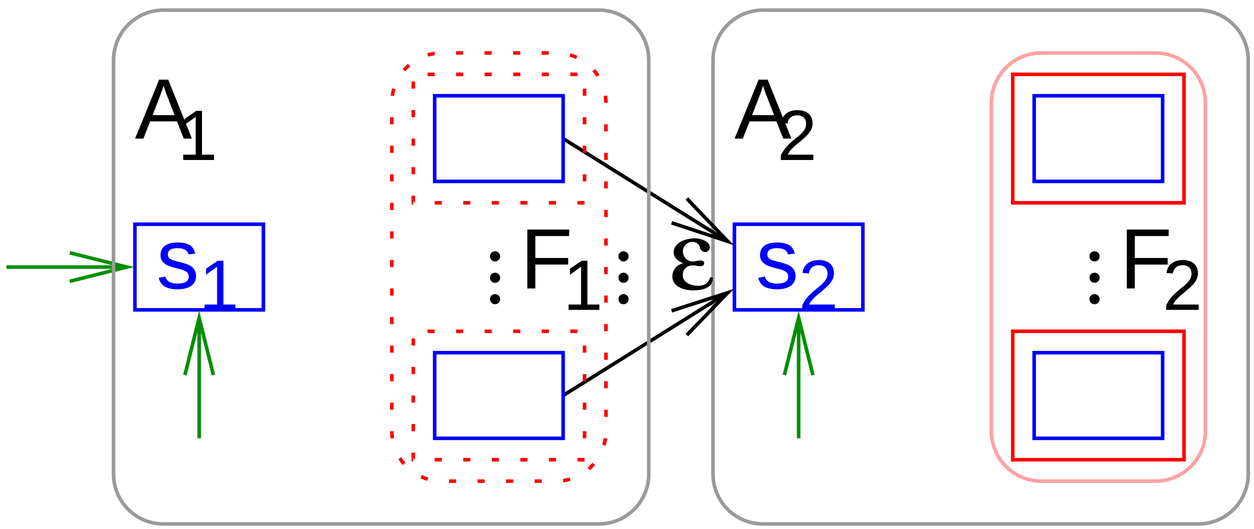
$A_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$, където $L(A_2) = L_2$

и $Q_1 \cap Q_2 = \emptyset$

$A := (Q_1 \cup Q_2, \Sigma, \delta, s_1, F_2)$

δ е дефинирена като $\delta_{1/2}$ on $Q_{1/2}$

ε преходи от F_1 до s_2.



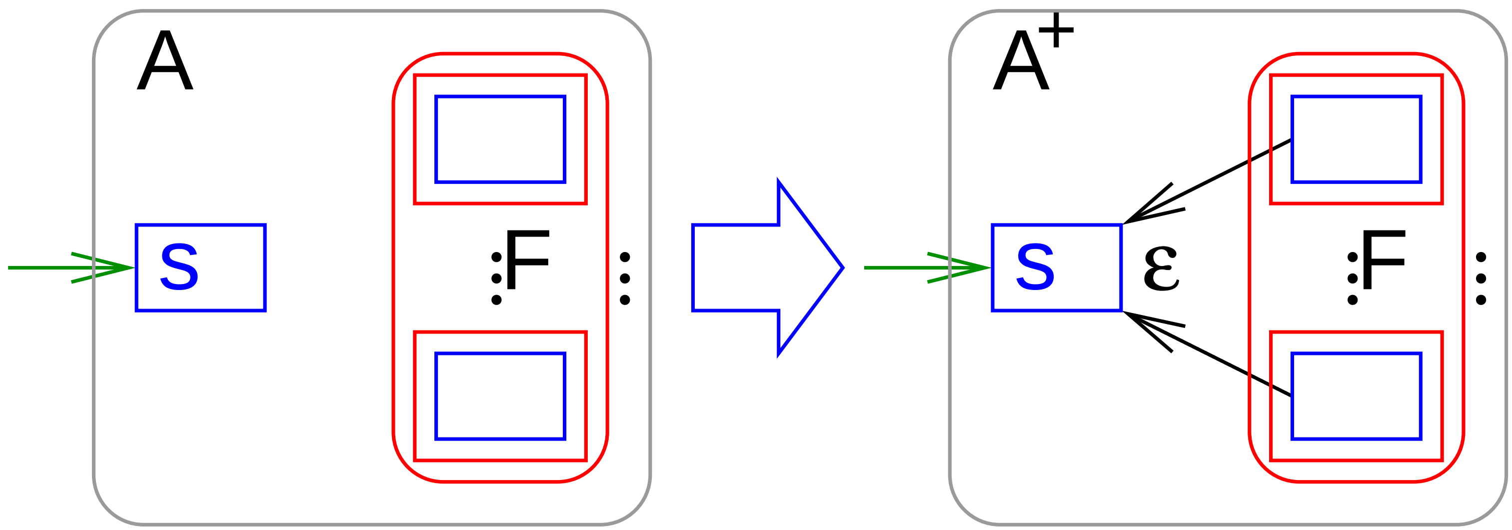
$$L^+ = \bigcup_{i \geq 1} L^i$$

$A = (Q, \Sigma, \delta, s, F)$ и $L(A) = L$

$A^+ := (Q, \Sigma, \delta^+, s, F)$

δ^+ е дефинирана като δ

ε преходи $f \to s \forall f \in F$.



εNFA $A \rightsquigarrow$ NFA $\bar{A}$

Д-во (идея):

Заместваме всеки ε преход и преход към следващ символ в A с директен преход към следващия символ в $\bar{A}$.

Трябва да внимаваме с крайните състояния.

Теорема на Клини

Теорема Всеки автоматен език е регулярен.

Д-во: Даден: DFA $A = (\{1,\ldots,n\}, \Sigma, \delta, s, F)$

Резултат: регулярен израз α такъв, че $L(A) = L(\alpha)$.

За всяко $f \in F$ нека $L_f = \left\{ w \in \Sigma^* : \hat{\delta}(s,w) = f \right\}$.

Ще намерим RegExp за L_f. Тъй като $L(A) = \bigcup_{f \in F} L_f$,

теоремата ще е доказана, защото F е крайно.

Даден: DFA $A_f = (\{1,\ldots,n\}, \Sigma, \delta, s, \{f\})$

Резултат: регулярен израз α и $L_f = L(A_f) = L(\alpha)$.

Нека $L_{ij} := L((\{1,\ldots,n\}, \Sigma, \delta, i, \{j\}))$

В частност $L_{sf} = L_f$.

Ако $i \neq j$: $L^0_{ij} := \{a \in \Sigma : j \in \delta(i,a)\}$

Ако $i = j$: $L^0_{ij} := \{a \in \Sigma : j \in \delta(i,a)\} \cup \{\varepsilon\}$

$L^m_{ij} := \left\{ w \in \Sigma^* : \exists \text{работен път } i \stackrel{w}{\Rightarrow} j = iPj \text{ и } P \in \{1,\ldots,m\}^* \right\}$

Тук преход iPj озаначава преход от i до j, с междинни състояния с номера $\leq m$.

Забележете, че $L_{ij} = L^n_{ij}$.

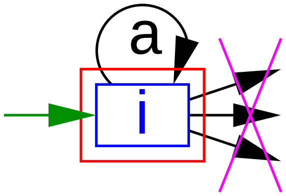
Ще построим реулярен израз за L^m_{ij} индуктивно, използвайки регулярните изрази за по-малките m.

L^m_{ij}:= $\left\{ w \in \Sigma^* : \exists\text{работен път } i \overset{w}{\Rightarrow} j = iPj \text{и } P \in \{1,\ldots,m\}^* \right\}$

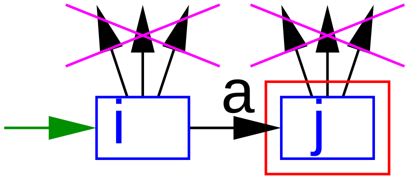
Даден: регулярен израз α^k_{ij}, $k < m$ и $L(\alpha^k_{ij}) = L^k_{ij}$

Резултат: α^m_{ij} и $L(\alpha^m_{ij}) = L^m_{ij}$

Ако $m = 0, i = j$: $\alpha^0_{ii} = \bigcup_{a \in \Sigma : \delta(i,a) = i} a \cup \varepsilon$

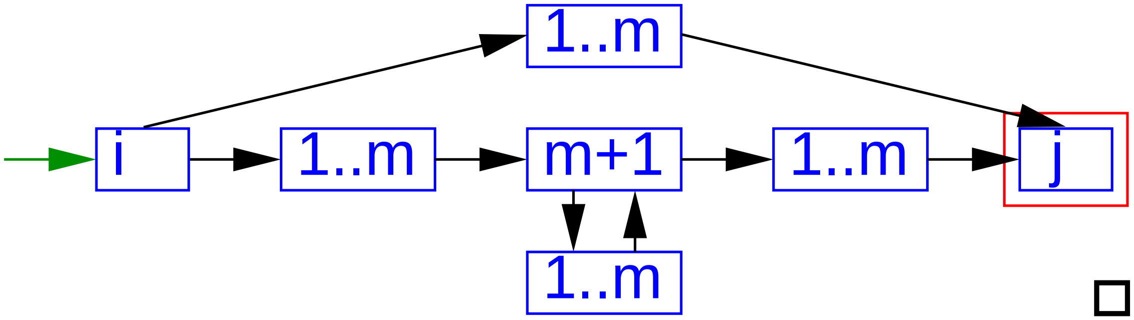


Ако $m = 0, i \neq j$: $\alpha^0_{ij} = \bigcup_{a \in \Sigma : \delta(i,a) = j} a$



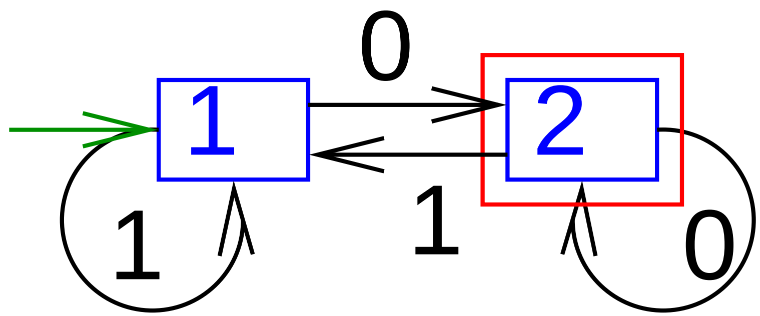
Ако $m \rightsquigarrow m+1$:

$$\alpha^{m+1}_{ij} = \alpha^m_{ij} \cup \alpha^m_{i,m+1} \cdot (\alpha^m_{m+1,m+1})^* \cdot \alpha^m_{m+1,j}$$



□

Пример



$$\alpha_{11}^{0} = 1 \cup \varepsilon \qquad \alpha_{22}^{0} = 0 \cup \varepsilon \qquad \alpha_{12}^{0} = 0 \qquad \alpha_{21}^{0} =$$

$$\begin{aligned} \alpha_{12}^{1} &= \alpha_{12}^{0} \cup \alpha_{11}^{0} \cdot (\alpha_{11}^{0})^{*} \cdot \alpha_{12}^{0} \\ &= 0 \cup (1 \cup \varepsilon) \cdot (1 \cup \varepsilon)^{*} \cdot 0 \\ &= 1^{*}0 \end{aligned} \qquad \begin{aligned} \alpha_{22}^{1} &= \alpha_{22}^{0} \cup \alpha_{21}^{0} \cdot (\alpha_{11}^{0})^{*} \\ &= 0 \cup \varepsilon \cup 1 \cdot (1 \cup \varepsilon)^{*} \\ &= 1^{*}0 \cup \varepsilon \end{aligned}$$

$$\begin{aligned} \alpha_{12}^{2} &= \alpha_{12}^{1} \cup \alpha_{12}^{1} \cdot (\alpha_{22}^{1})^{*} \cdot \alpha_{22}^{1} \\ &= 1^{*}0 \cup 1^{*}0 \cdot (1^{*}0 \cup \varepsilon)^{*} \cdot (1^{*}0 \cup \varepsilon) \\ &= 1^{*}0(1^{*}0)^{*} \end{aligned}$$

$L(\alpha_{12}^{2}) = L_{12}^{2} = L_{12} = L_{2}$, където $F = \{2\}$.

1.1.4 Pumping лема (лема за покачването)

Ако L регулярен език

$$\longrightarrow \exists n \in \mathbb{N} : \forall w \in L : |w| > n$$

$$\longrightarrow \exists u, v, x : w = uvx \wedge$$

1. $|v| \geq 1 \wedge$
2. $|uv| \leq n \wedge$
3. $\forall i \in \mathbb{N}_0 : uv^i x \in L$

С думи:

Достатъчно дългите думи на един регулярен език имат непразна поддума която можем да "pump"ваме (итерираме) без да напускаме езика.

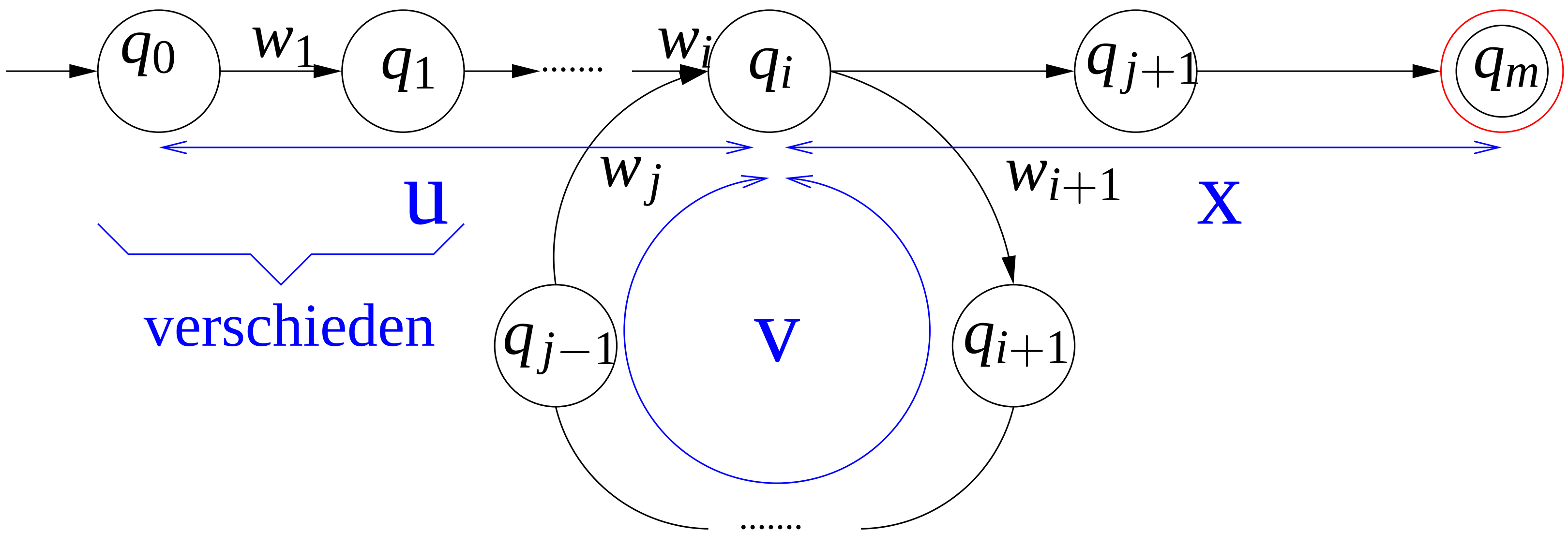
Д-во на Pumping лемата

L регулярен $\longrightarrow \exists n \in \mathbb{N} : \forall w \in L : |w| > n \longrightarrow \exists u, v, x :$
$w = uvx \wedge |v| \geq 1 \wedge |uv| \leq n \wedge \forall i \in \mathbb{N}_0 : uv^i x \in L$

Д-во: Нека $A = (Q, \Sigma, \delta, q_0, F)$ DFAи $L(A) = L$.

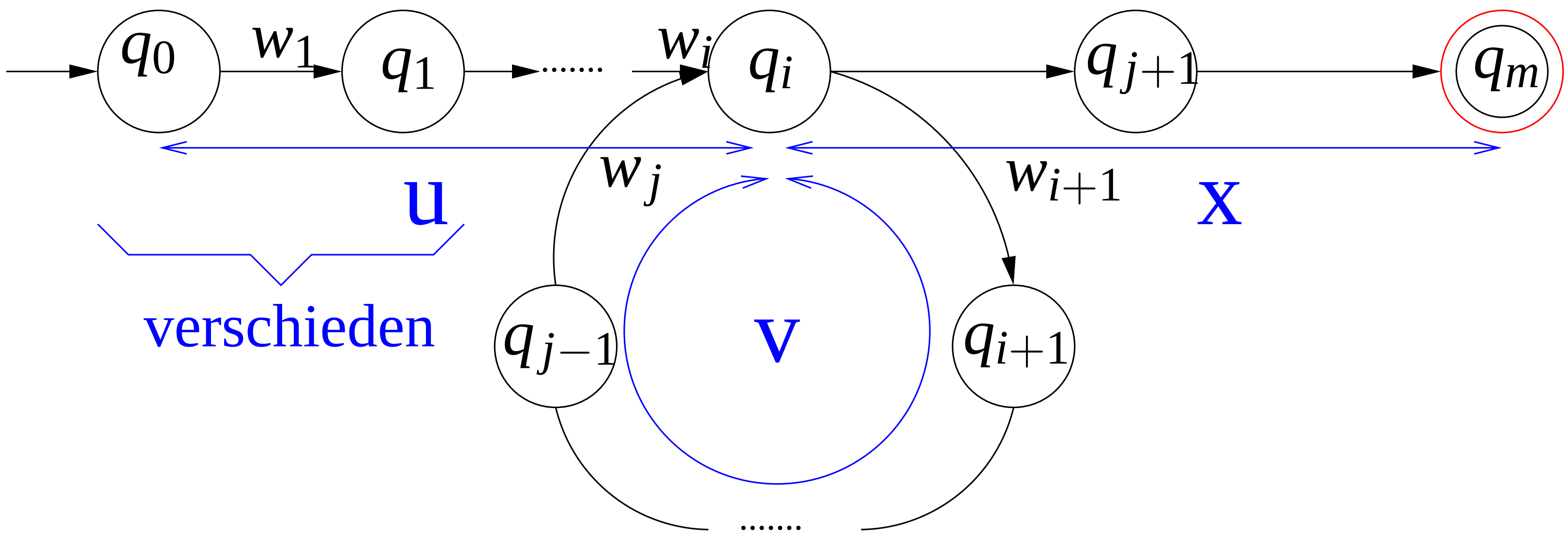
Нека $n = |Q|$ и $w \in L$ с $|w| = m > n$ (произволна).

Нека $q_0, \ldots, q_m$ състояния.



$(\exists i < j \leq n : q_i = q_j) \longrightarrow |v| \geq 1,\ |uv| \leq n,\ uv^i x$ са също в езика

Д-во на Pumping лемата



$(\exists i < j \leq n : q_i = q_j) \longrightarrow |v| \geq 1,\ |uv| \leq n,\ uv^i x$ са също в езика

$w = w_1 \ldots w_m;\ u = w_1 \ldots w_i;\ v = w_{i+1} \ldots w_j;\ x = w_{j+1} \ldots w_m$

$(q_0, w) \vdash^* (q_i, w_{i+1} \ldots w_j \ldots w_m) \vdash^* (q_j, w_{j+1} \ldots w_m) \Rightarrow$

$(q_0, w_1 \ldots w_i) \vdash^* (q_i, \varepsilon)\ \&\ (q_i, w_{i+1} \ldots w_j) \vdash^* (q_j, \varepsilon)\ \&\ q_i = q_j$

$\Rightarrow (q_0, w_1 \ldots w_i w_{j+1} \ldots w_m) \vdash^* (q_m, \varepsilon)$

$\Rightarrow (q_0, ux) \vdash^* (q_m, \varepsilon)\ \&\ (q_0, uv^i x) \vdash^* (q_m, \varepsilon).$

Пример: $L = \left\{a^k b^k : k \in \mathbb{N}\right\}$

Да допуснем, че L е регулярен.

Нека n е числото от Pumping лемата и нека

$w = a^n b^n = uvx$ в съответствие с Pumping лемата, тогава

$ux \in L$.

$|uv| \leq n, |v| \geq 1 \longrightarrow v = a^\ell$ за $\ell \geq 1$.

$ux = a^{n-\ell} b^n \in L$.

Противоречие. $\blacksquare$

Пример: Балансирани скоби $L_{()}$

Да допуснем, че L е регулярен.

Нека n е числото от Pumping лемата за L и да разгледаме $w = (^n)^n = uvx$ съгласно Pumping лемата $ux \in L_{()}$ и $|v| > 1$ и $|uv| \leq n$.

Тогава $v = (^i$, $i \neq 0$

и $ux = (^{n-i})^n \notin L_{()}$ Противоречие.

$L = \{0^p : p$ is a prime number$\}$

Да допуснем, че L е регулярен.

Нека n е числото от Pumping лемата за L.

Нека $p \geq n+2$ е просто число.($\exists$ безкрайно много прости числа) $\longrightarrow 0^p \in L = uvw$, $|v| \geq 1$, $|uw| \geq 2$.

Pumping-лема: $uv^{|uw|}w \in L$.

$\longrightarrow |uw| + |uw| \cdot |v| = |uw|(1+|v|)$ е просто число.

Два нетривиални делителя $|uw| \geq 2$ и $(1+|v|) \geq 2$.

Противоречие. $\blacksquare$

Pumping-лемата

не е достатъчно условие за регулярност

Пример: $L = \left\{c^m a^\ell b^\ell : m, \ell \geq 0\right\} \cup \{a, b\}^*$ не е регулярен,

но

ако $n \geq 1$ е произволно и $x \in L$ с $|x| \geq n$.

1. $x \in a^* b^*$:

$$x = \underbrace{\varepsilon}_{u} \underbrace{a}_{v} \underbrace{a^m b^{n-m-1}}_{w}$$

1. $|v| = 1 \geq 1$
2. $|uv| = 1 \leq n$
3. $uv^i w = a^i a^m b^{n-m-1} \in a^* b^* \subseteq L$

Pumping-лемата

не е достатъчно условие за регулярност

Пример: $L = \left\{c^m a^\ell b^\ell : m, \ell \geq 0\right\} \cup \{a,b\}^*$ не е регулярен,

Нека n е произволно, $w \in L$ произволно с $|w| \geq n$.

1. Ако $w \in a^* b^*$: го видяхме.
2. Ако $w = c^m a^\ell b^\ell$, $m \geq 1$:

Разгледайте $w = \underbrace{\varepsilon}_{u} \underbrace{c}_{v} \underbrace{c^{m-1} a^\ell b^\ell}_{x}$

1. $|v| = 1 \geq 1$
2. $|uv| = 1 \leq n$
3. $uv^i x = c^{m-1+i} a^\ell b^\ell \in L$

1.1.5 Релации на еквивалентност и минимален автомат

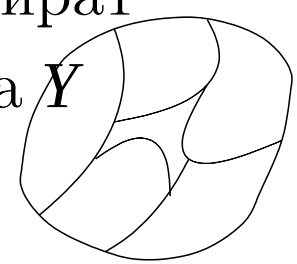
Идея: работим директно с L, без да разглеждаме конкретен автомат.

Припомняне: Релация на еквивалентност

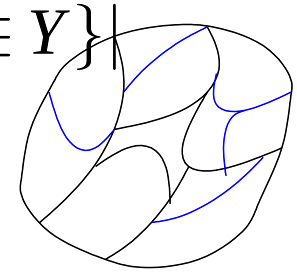
Една релация $R \subseteq Y \times Y$ се нарича релация на еквивалентност, ако R е:

- рефлексивна $\forall x : xRx$
- транзитивна $\forall xyz : xRy \wedge yRz \longrightarrow xRz$
- симетрична. $\forall xy : xRy \longrightarrow yRx$

Клас на еквивалентност: $[x] = \{y : xRy\}$. Класовете на еквивалентност са непресичащи се и индуцират частична наредба в Y, т.е. всеки елемент на Y принадлежи точно на един клас на еквив.



Индекс: индекс$|R| :=$ $|$Клас на еквив.$| = |\{[x] : x \in Y\}|$



Прецизиране: R прецизира R' ($R \subseteq R'$)

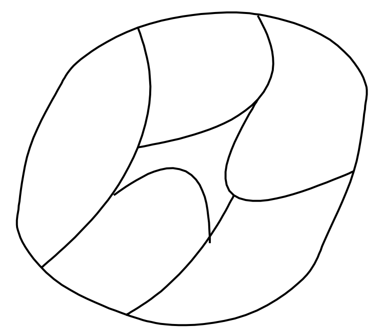
Лема: R прецизира $R' \longrightarrow \forall$ класове на еквивалентност $[x]_R : [x]_R \subseteq [x]_{R'}$

Д-во:

$$y \in [x]_R \Leftrightarrow (y,x) \in R$$

$$\xrightarrow{R \subseteq R'} (y,x) \in R'$$

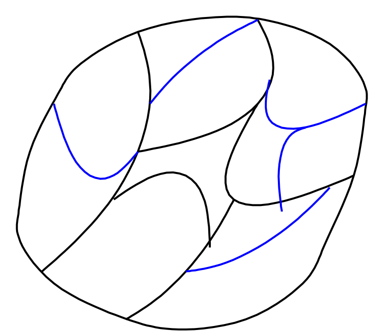
$$\Leftrightarrow y \in [x]_{R'}$$



Следствие: R прецизира $R' \longrightarrow |R| \geq |R'|$

Д-во: Разгледайте $\rho([x]_R) = [x]_{R'}$.

Проверете, че е добре дефинирана функция, която върху.



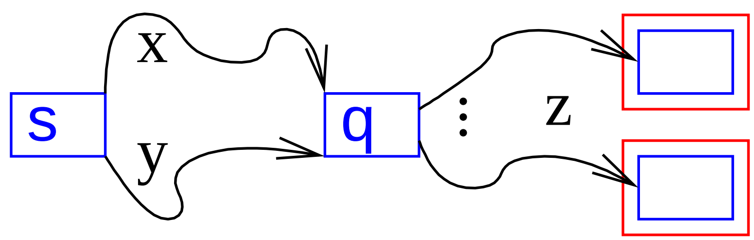
Релация на Нероуд

За езика L релацията на Нероуд е дефинирана като

$R_L := \{(x,y) \in \Sigma^* \times \Sigma^* : \forall z \in \Sigma^* : xz \in L \Leftrightarrow yz \in L\}$

Идея: класовете на еквивалентност съответстват на състоянията.

Защо?



DFA пораждат релация на еквивалентност

Нека $M = (Q, \Sigma, \delta, s, F)$ е DFA и $L(M) = L$.

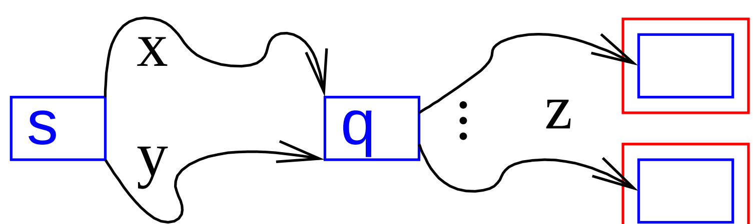
$R_M := \left\{(x,y) \in \Sigma^* \times \Sigma^* : \hat{\delta}(s,x) = \hat{\delta}(s,y)\right\}$.

релация на еквивалентност! по един клас на еквивалентност (за достижимо от s) състояние.

Лема 1: R_M прецизира релацията на Нероуд $R_L =$

$$\{(x,y) \in \Sigma^* \times \Sigma^* : \forall z \in \Sigma^* : xz \in L \Leftrightarrow yz \in L\}$$

Д-во : $\forall (x,y) \in \Sigma^* \times \Sigma^* : \hat{\delta}(s,x) = \hat{\delta}(s,y) \longrightarrow$
$\forall z : \hat{\delta}(s,xz) = \hat{\delta}(s,yz) \longrightarrow \forall z : xz \in L \Leftrightarrow yz \in L$



Безкраен индекс на релацията на Нероуд

Наблюдение: индексът $|R_L| = \infty \longrightarrow L$ не е регулярен.

Д-во: Да допуснем, че L е регулярен.
$\longrightarrow \exists$ DFA $M = (Q, \Sigma, \delta, s, F) : L(M) = L$.
$\longrightarrow R_M$ прецизира R_L.
$\longrightarrow |Q| \geq |R_M| \geq |R_L| = \infty$.
Противоречие.

Следователно: Ако L е регулярен, то индексът $|R_L| < \infty$.

Автомат от класовете на еквивалентност

$R_L := \{(x,y) \in \Sigma^* \times \Sigma^* : \forall z \in \Sigma^* : xz \in L \Leftrightarrow yz \in L\}$

Идея: когато класовете на еквивалентност $[w_1], \ldots, [w_k]$ на R_L съответстват на състоянията на един DFA $M_\equiv$, тогава по лемата по-долу минималният автомат за L е:

$M_\equiv := (\{[w_1], \ldots, [w_k]\}, \Sigma, \delta_\equiv, [\varepsilon], F_\equiv)$ с
$F_\equiv := \{[w] : w \in L\}$ и
$\delta_\equiv([w], a) := [wa]$.

Лема: $\delta_\equiv$ е добре дефинирана
Лема: $\hat{\delta}_\equiv([\varepsilon], w) = [w]$
Лема: $L(M_\equiv) = L$

Минимален автомат

$R_L := \{(x,y) \in \Sigma^* \times \Sigma^* : \forall z \in \Sigma^* : xz \in L \Leftrightarrow yz \in L\}$

$M_\equiv := (\{[w_1], \ldots, [w_k]\}, \Sigma, \delta_\equiv, [\varepsilon], F_\equiv)$, където

$F_\equiv := \{[w] : w \in L\}$ и

$\delta_\equiv([w], a) := [wa]$.

Лема: $\delta_\equiv$ е добре дефинирана

$xR_Ly \longrightarrow \forall a \in \Sigma : xaR_Lya$ дясно инвариантна

$xR_Ly \longrightarrow \forall z \in \Sigma^* : xz \in L \Leftrightarrow yz \in L$

$\longrightarrow \forall az \in \Sigma^* : x(az) \in L \Leftrightarrow y(az) \in L$

$\Leftrightarrow \forall a \in \Sigma : \forall z \in \Sigma^* : (xa)z \in L \Leftrightarrow (ya)z \in L$

$\longrightarrow \forall a \in \Sigma : xaR_Lya$ $\square$

Минимален автомат

$R_L := \{(x,y) \in \Sigma^* \times \Sigma^* : \forall z \in \Sigma^* : xz \in L \Leftrightarrow yz \in L\}$

$M_\equiv := (\{[w_1], \ldots, [w_k]\}, \Sigma, \delta_\equiv, [\varepsilon], F_\equiv)$, където

$F_\equiv := \{[w] : w \in L\}$ и

$\delta_\equiv([w], a) := [wa]$.

Лема: $\hat{\delta}_\equiv([x], y) = [xy]$

Индукция по $|y|$:

$\hat{\delta}_\equiv([x], \varepsilon) = [x]$.

$\hat{\delta}_\equiv([x], aw) \overset{\text{деф.}\hat{\delta}_\equiv}{=} \hat{\delta}_\equiv(\delta_\equiv([x], a), w) \overset{\text{деф.}\delta_\equiv}{=} \hat{\delta}_\equiv([xa], w) =$
$[xaw]$. $\square$

Минималният автомат: разпознава L

$R_L := \{(x,y) \in \Sigma^* \times \Sigma^* : \forall z \in \Sigma^* : xz \in L \Leftrightarrow yz \in L\}$

$M_{\equiv} := (\{[w_1], \ldots, [w_k]\}, \Sigma, \delta_{\equiv}, [\varepsilon], F_{\equiv})$ с

$F_{\equiv} := \{[w] : w \in L\}$ и

$\delta_{\equiv}([w], a) := [wa]$.

Лема: $L(M_{\equiv}) = L$.

$w \in L(M_{\equiv})$

$\Leftrightarrow \hat{\delta}_{\equiv}([\varepsilon], w) \in \{[w] : w \in L\}$ Деф. $M_{\equiv}$

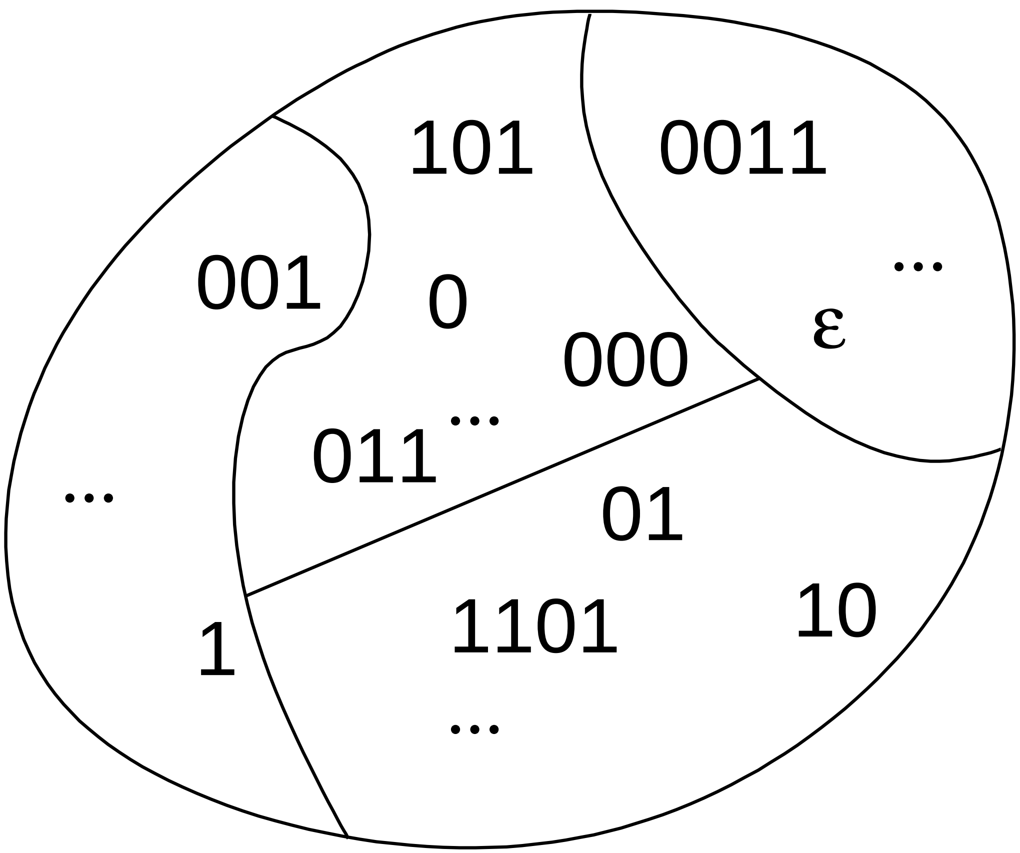
$\Leftrightarrow [w] \in \{[w] : w \in L\}$ предишната лема

$\Leftrightarrow w \in L$ класовете на еквив. са или изцяло в, или изцяло извън L

$([w] \in \{[w] : w \in L\} \longrightarrow \exists x \in L : [x] = [w] \longrightarrow xR_Ly \longrightarrow$
$\forall z : xz \in L \Leftrightarrow wz \in L \longrightarrow x\varepsilon \in L \Leftrightarrow w\varepsilon \in L)$

Пример

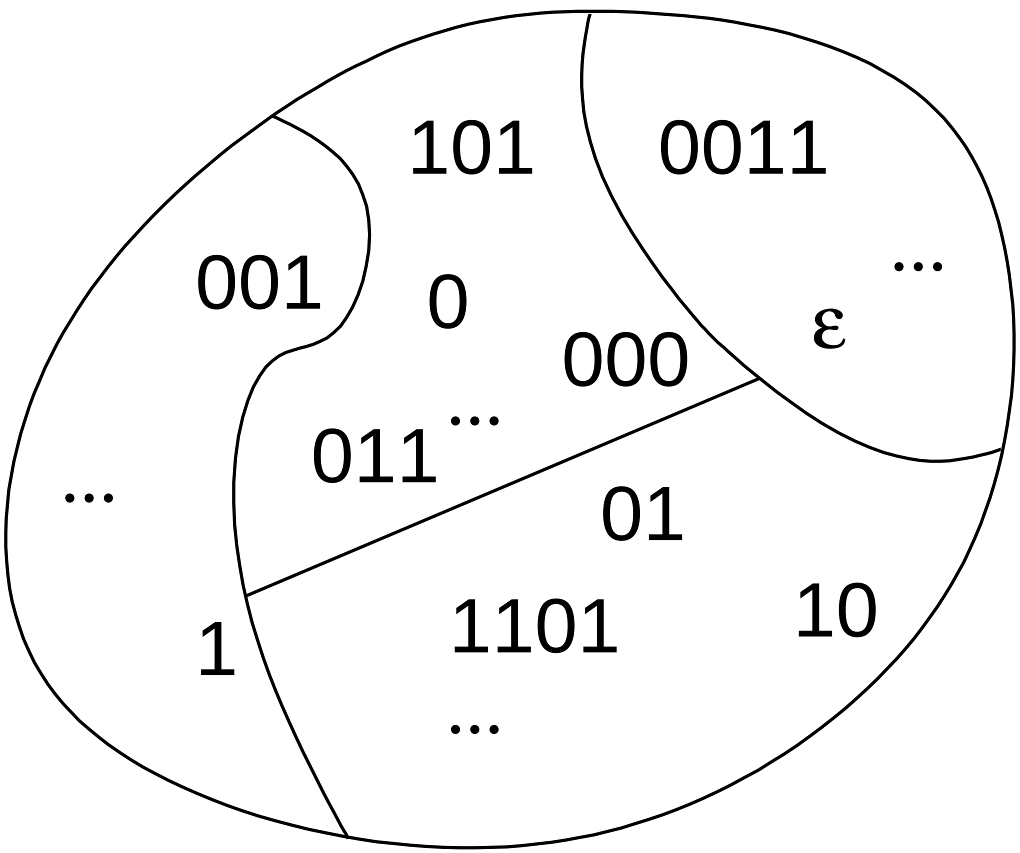
$L \subseteq \{0,1\}^*$ език, всички думи с четен брой единици и четен брой нули



Класовете на еквивалентност?

Пример

$L \subseteq \{0,1\}^*$ език, всички думи с четен брой единици и четен брой нули

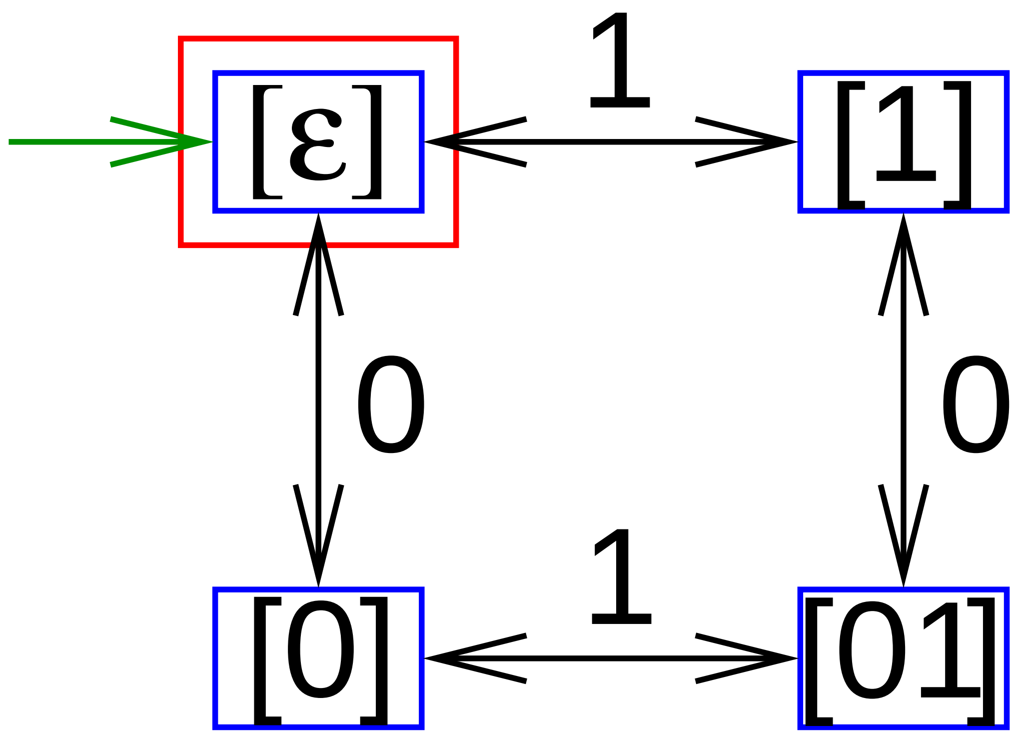


Класовете на еквивалентност:

$[\varepsilon], [0], [1], [01]$

Пример

$L \subseteq \{0,1\}^*$ език, всички думи с четен брой единици и четен брой нули



Теорема на Майхил-Нероуд

Нека

$R_L := \{(x,y) \in \Sigma^* \times \Sigma^* : \forall z \in \Sigma^* : xz \in L \Leftrightarrow yz \in L\}$.

L не е регулярен $\longrightarrow |R_L| = \infty$

Теорема на Майхил-Нероуд L регулярен $\iff |R_L| < \infty$.

Нека

$M_\equiv := (\{[w_1], \ldots, [w_k]\}, \Sigma, \delta_\equiv, [\varepsilon], F_\equiv)$, и

Тогава $L(M_\equiv) = L$

Ако L е реулярен и $M = (Q, \Sigma, \delta, s, F)$ произволен DFA с $L(M) = L$, то R_M е прецизиране за R_L. Следователно $|R_L| \leq |Q|$, т.е. $M_\equiv$ е минимален автомат (с най-малък брой състояния), разпознаващ L.

Един автомат се нарича свързан, ако всяко състояние едостижимо от началното.

Следствие: Всички минимални автомати за L са изоморфни на $M_\equiv$.

Д-во: Нека $M = (Q, \Sigma, \delta, s, F)$ е свързан DFA, $L(M) = L$ и $|Q| = |R_L|$. Ще покажем, че $M \cong M_\equiv$, т.е. M е изоморфен на $M_\equiv$.

За всяко $q \in Q$ има дума w, такава че $\hat{\delta}(s, w) = q$.

Дефинираме $\kappa(q) = [w]$.

- деф на κ е коректна

  $\hat{\delta}(s, w_1) = \hat{\delta}(s, w) \longrightarrow w_1 R_L w \longrightarrow [w_1] = [w]$.

  $w_1 z \in L \iff \hat{\delta}(s, w_1 z) \in F \iff \hat{\delta}(\hat{\delta}(s, w_1), z) \in F \iff \hat{\delta}(\hat{\delta}(s, w), z) \in F \iff \hat{\delta}(s, wz) \in F \iff wz \in L$

- κ е биекция
  (еднозначна) Нека $q \neq q_1$ и $\hat{\delta}(s, w_1) = q_1$.
  Допускаме, че
  $\kappa(q) = \kappa(q_1) \longrightarrow [w] = [w_1] \longrightarrow |R_M| > |R_L|$.
  Противоречие.
  (върху) $\forall w(q = \hat{\delta}(s, w) \longrightarrow \kappa(q) = [w])$.

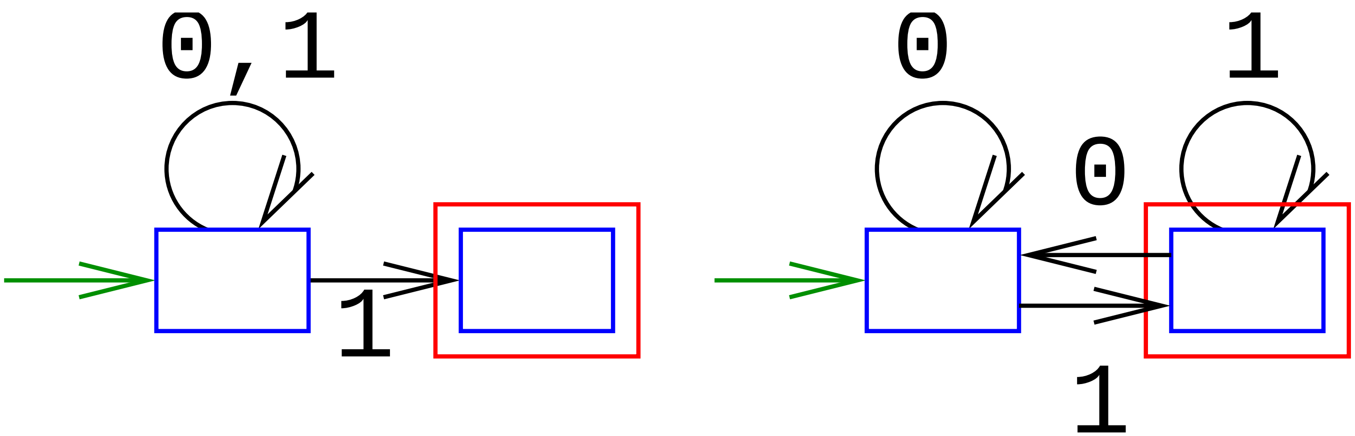
- $\kappa(s) = [\varepsilon]$ $(\hat{\delta}(s, \varepsilon) = s)$

- $\kappa(\delta(q, a)) = \delta_{\equiv}(\kappa(q), a)$
  $q = \hat{\delta}(s, w) \longrightarrow \delta(q, a) = \hat{\delta}(s, wa) \longrightarrow \kappa(\delta(q, a)) =$
  $[wa] = \delta_{\equiv}([w], a) = \delta_{\equiv}(\kappa(q), a)$

- $f \in F \iff \kappa(f) \in F_{\equiv}$.

Един контрапример NFA

Има структурно различни минимални NFAs за $(0 \cup 1)^* 1$.



Упражнение: Напишете функцията на прехода.

Конструкция

на минималния автомат

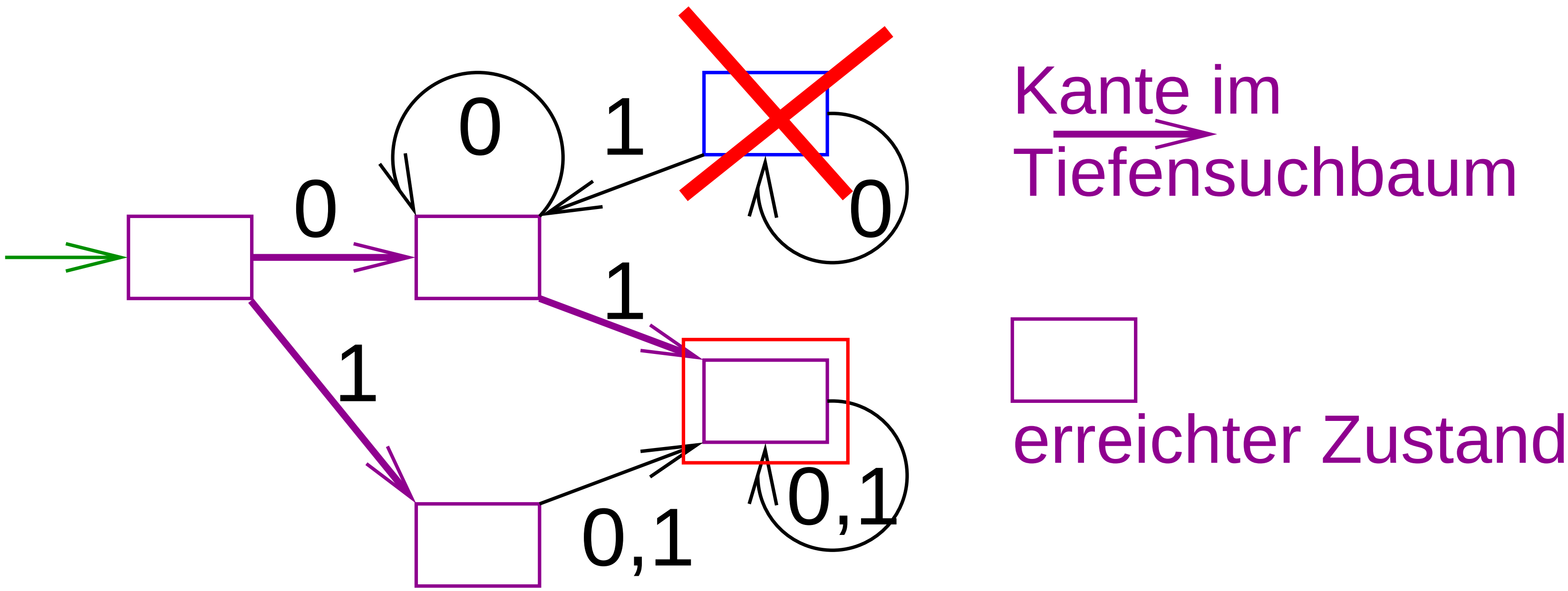
Махаме състоянията, недостижими от s.

Алгоритъм: Търсене в дълбочина в графа G_A за s.

Маркираме всички достижими състояния.

Махаме недостижимите състояния.

Изпълнение — Примери



Еквивалентни състояния

Идея: разгледайте DFA $M = (Q, \Sigma, \delta, s, F)$

(без недостижими състояния)

M не е минимален $\longrightarrow$

R_M прецизира R_L $\longrightarrow$

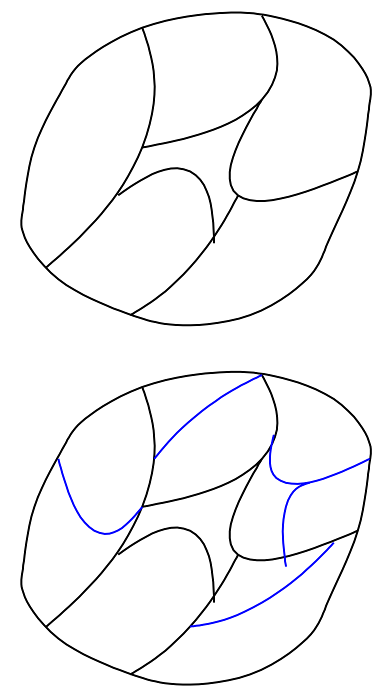
$\exists q \neq r \in Q : [q]_M \dot{\cup} [r]_M \subseteq K$

за някой клас на екв. K за R_L

q, r се наричат еквивалентни, $q \equiv r$.

В частност:

$\forall w \in \Sigma^* : \hat{\delta}(q, w) \in F \Leftrightarrow \hat{\delta}(r, w) \in F$



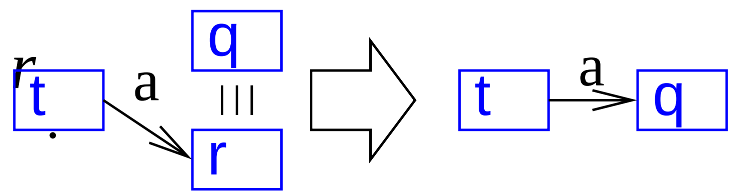
Махане на еквивалентните състояния

Да разгледаме $q \neq r \in Q : q \equiv r$ и $r \neq s$

Махаме r:

$M' := (Q \setminus \{r\}, \Sigma, \delta', s, F \setminus \{r\})$ където

$$\delta'(t,a) := \begin{cases} q & \text{ако } \delta(t,a) = r \\ \delta(t,a) & \text{иначе} \end{cases}$$



Лема: $L(M') = L$

Д-во:Упражнение

Минимизация на състоянията

Първа стъпка:

Function minDFA(M)

  махаме състоянията недостижими от s

  while $\exists q, r \in Q : q \equiv r \wedge q \neq r \wedge r \neq s$ do

    махаме r от M

  return M

Проблем: Как да намерим еквиваленните състояния?

$q \equiv r$ iff $\forall z \in \Sigma^* : \hat{\delta}(q, z) \in F \Leftrightarrow \hat{\delta}(r, z) \in F$

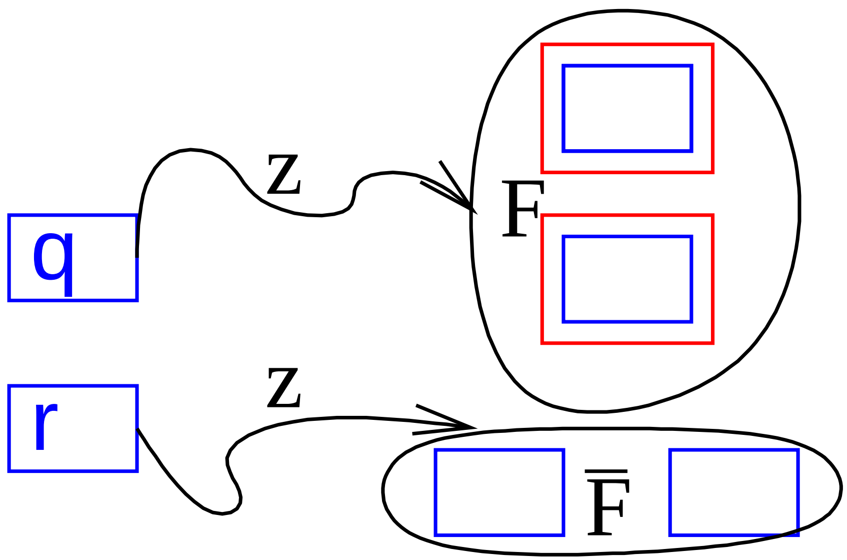
Всеки квантор по не крайно множество!

Нееквивалентни състояния

$q \equiv r$ iff $\forall z \in \Sigma^* : \hat{\delta}(q,z) \in F \Leftrightarrow \hat{\delta}(r,z) \in F$

$q \not\equiv r$ iff $\exists z \in \Sigma^* : \hat{\delta}(q,z) \in F \not\Leftrightarrow \hat{\delta}(r,z) \in F$

z свидетелства за нееквивалентност.



Проблем: да се намерят свидетели за нееквивалентност

Най-къси свидетели за нееквивалентност

$\forall q \in F, r \notin F : \varepsilon$ testifies $q \not\equiv r$.

Нека $w = aw'$ е най-къс свидетел за $q \not\equiv r$.

Наблюдение: w' е свидетел за $q' := \delta(q,a) \not\equiv \delta(r,a) =: r'$

Лема: w' е най-къс свидетел за $q' \not\equiv r'$

Доказателство с допускане на противното: Да допуснем:

w'' е по-къс свидетел за $q' \not\equiv r'$

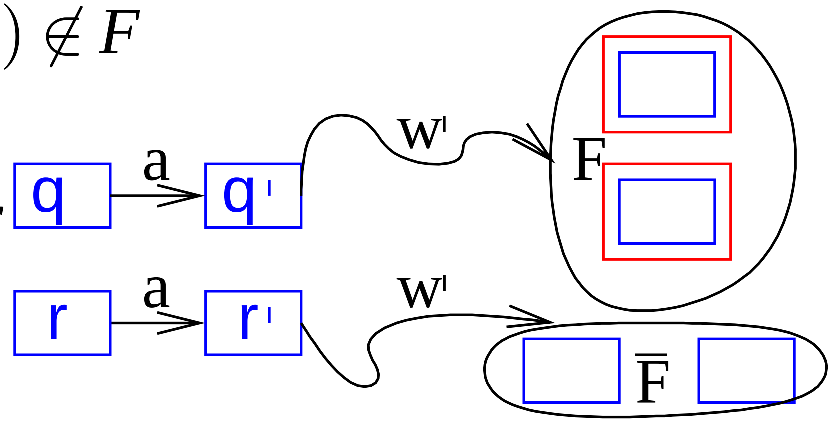
$\longrightarrow \hat{\delta}(q', w'') \in F \wedge \hat{\delta}(r', w'') \notin F$ БОО

$\longrightarrow \hat{\delta}(\delta(q,a), w'') \in F \wedge \hat{\delta}(\delta(r,a), w'') \notin F$

$\longrightarrow \hat{\delta}(q, aw'') \in F \wedge \hat{\delta}(r, aw'') \notin F$

$\longrightarrow aw''$ е по-къс свидетел за $q \not\equiv r$

Противоречие.



Най-къси свидетели за не-еквивалентност

ε свидетелства $q \not\equiv r$, ако $q \in F, r \notin F$.

Нека $w = aw'$ е най-къс свидетел за $q \not\equiv r$.

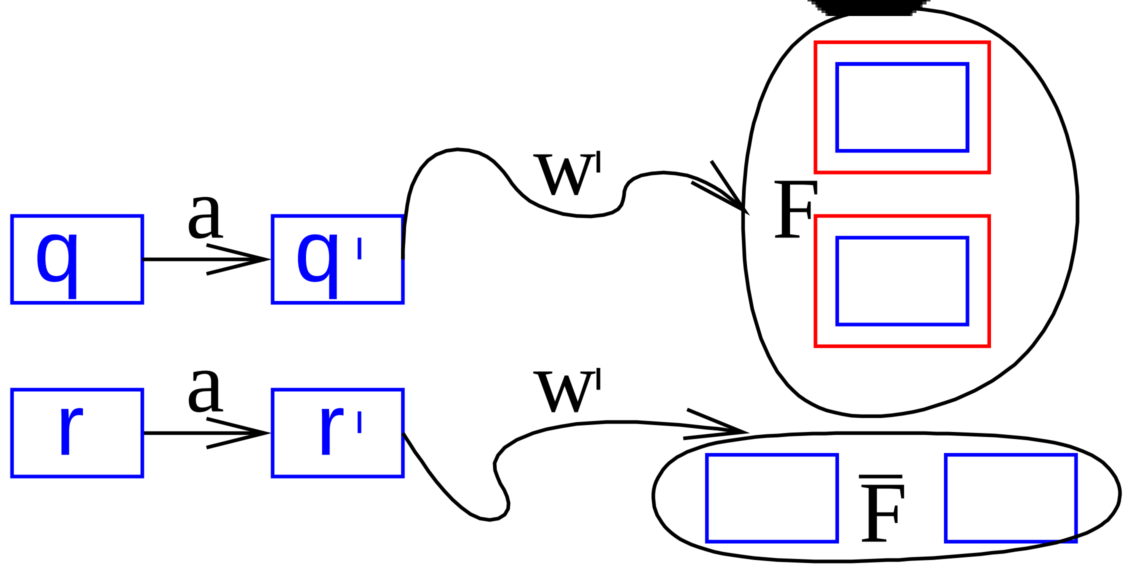
Лема: w' е най-къс свидетел за $q' := \delta(q,a) \not\equiv \delta(r,a) =: r'$

Заключение:

най-късите свидетелели са намерени систематично и ефективно.

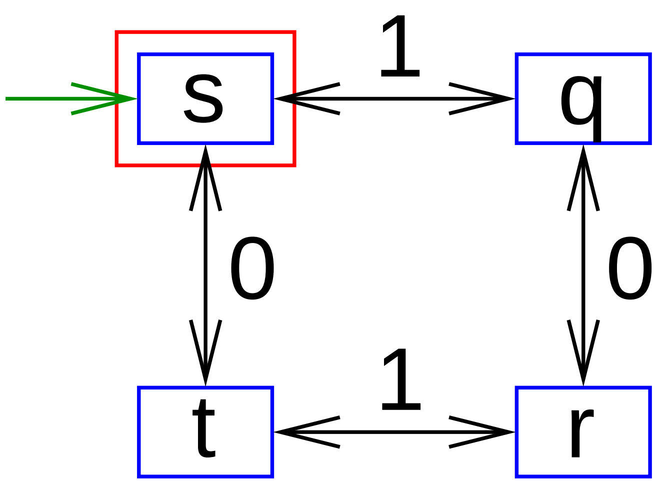
$\rightsquigarrow$ всички нееквиваленстности

$\rightsquigarrow$ класовете на еквивалентност за състояния.


q
a
q'
w'
F
r
a
r'
w'
$\overline{F}$

Пример

$L \subseteq \{0,1\}^*$ език, всички думи с четен брой нули и четен брой нули



$0 = 0\varepsilon$ е най-къс свидетел за $t \neq r$.

$\rightsquigarrow \varepsilon$ е най-късият свидетел за $s = \delta(t,0) \neq \delta(r,0) = q$.

Нека $w = aw'$ е най-късият свидетел за $q \not\equiv r$.

Лема: w' е най-късият свидетел за

$q' := \delta(q,a) \not\equiv \delta(r,a) =: r'$

Нека N е множеството от всички не-еквивалентни двойки от състояния за свидетелите с дължина $\leq k$.

За кои двойки от състояния $\{q,r\}$ има свидетели с дължина $k+1$?

Лема: $\{\{q,r\} \subseteq Q : \exists a \in \Sigma : \{\delta(q,a),\delta(r,a)\} \in N\}$

Един лесен алгоритъм

$N := \emptyset$ // маркирани двойки

$N' := \{\{q,r\} \subseteq Q : q \in F \not\Leftrightarrow r \in F\}$ // следващите маркирани двойк

while $N' \neq \emptyset$ do

$\quad N := N \cup N'$

$\quad N' := \{\{q,r\} \subseteq Q : \exists a \in \Sigma : \{\delta(q,a), \delta(r,a)\} \in N\} \setminus N$

Общо време: $\mathcal{O}\left(|\Sigma| \cdot |Q|^3\right)$

Инициализация: $\mathcal{O}\left(|Q|^2\right)$

Време за цикъла: $\mathcal{O}\left(|\Sigma| \cdot |Q|^2\right)$

Колко цикъла? Сигурно $\leq |Q|^2$.

По-точно наблюдение: $\leq |Q|$ цикли

Минимизация на състоянията за време $\mathcal{O}(|\Sigma| \cdot |Q| \log |Q|)$

[Hopcroft 1971]. Data structures.

Леко опростяване :
[Blum, Minimization of finite automata in $O(n \log n)$ time, Inf. Proc. Некаters, 1996.]

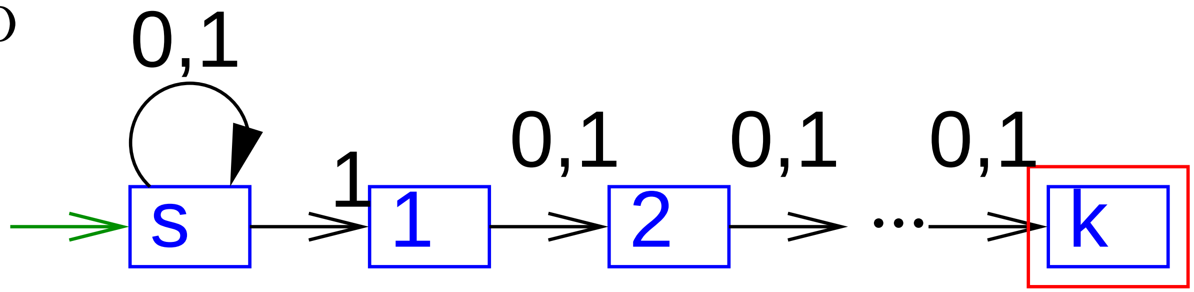
Защо минимизация на състоянията

- минимално място за таблицата на преходите между състояния
- минимален автомат очевидно $\leadsto$ ние научаваме нещо за самия език.

Но, когато δ е представена като списък от преходи или програма, искаме да оптимизираме дължината ѝ и времето за изпълнение.

Изобщо $\leadsto$ активна научна област.

Пример

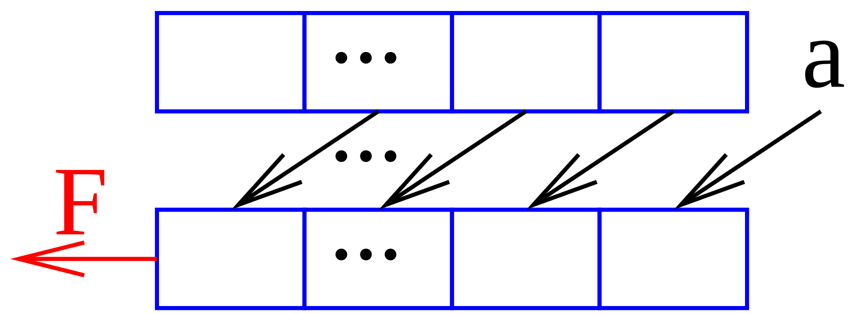


$L=\{0,1\}^*1\{0,1\}^{k-1}$

Минималният автомат има 2^k състояния.

$(\{0,\ldots,2^k-1\},\{0,1\},\delta,0,F)$

$\delta(q,a)=2q+a$, $q\in F\Leftrightarrow q[k-1]=1$



Верификация за нерегулярност

- Pumping Лема:

  +: Лесно се прилага

  −: Само необходимо условие

- Релацията на Нероуд

  +: Необходимо и достатъчно условие $(R_L) = \infty$

  −: Малко трудно се проверява

Релация на Нероуд

Пример: $L = \{a^n b^n : n \geq 1\}$

Твърдение: $\forall k > 1, j \neq k > 1 : [a^k b] \neq [a^j b]$

$[a^k b] = \left\{a^k b, a^{k+1} bb, \ldots\right\} = \left\{a^{k+i} b^{i+1}\right\}$

така винаги $k-1$ повече a-та от b-та.

Следователно $[a^k b]$ и $[a^j b]$ са непресичащи. $\square$

Релация на Нероуд

Пример: $L = \left\{c^m a^\ell b^\ell : m, \ell \geq 0\right\} \cup \{a, b\}^*$

Твърдение: $\forall k > 1, j \neq k > 1 : [ca^k b] \neq [ca^j b]$

$[ca^k b] = \left\{c^m a^{k+i} b^{1+i} : m \geq 0, i \geq 1\right\}$

така винаги $k-1$ повече a-та от b-та.

Следователно $[ca^k b]$ и $[ca^j b]$ са непресичащи се. $\square$

1.1.6 Свойства на затвореност

Нека L, L' са регулярни езици.

Тогава и следните езици са регулярни:

$L \cup L'$, L^*, $L \cdot L'$: по дефиниция на рег. израз.

$\bar{L}:= \Sigma^* \setminus L$: Да разгледаме DFA $A = (Q, \Sigma, \delta, s, F)$ с $L(A) = L$.
Нека $\bar{A}:= (Q, \Sigma, \delta, s, Q \setminus F)$. Тогава $L(\bar{A}) = \bar{L}$.

$L \cap L' = \overline{\bar{L} \cup \bar{L'}}$ (Де Морган)

$L \setminus L' = L \cap \bar{L'}$

L^R: Упражнение. Упътване: Индукция по регулярен израз.

(Product автомат)

Конструкции на DFA за теоретико-мнжествените операции

L и L' са регулярни езици, дефинирани с DFAs

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

$A' = (Q', \Sigma, \delta', s', F')$.

Идея: Автоматът $A_\times$ емулира поведението на A и A'.

Product автомат: $A_\times := (Q \times Q', \Sigma, \delta_\times, (s, s'), F_\times)$ с

$\delta_\times((q, q'), a) = (\delta(q, a), \delta(q', a))$

Дефинираме F в съответствие с операциите:

$L \cup L'$: $F_\times := Q \times F' \cup F \times Q'$

$L \cap L'$ $F_\times := F \times F'$

1.1.7 Разрешимост

на прости свойства на един краен автомат

Word problem

$w \in L$?

Изброждаме DFA A.

Симулираме A с вход w.

Дали има крайно състояние, което е достижимо?

Линейно време, ако DFA ако е даден автомата!

Проблемът за празнотата на езика

$L = \emptyset$?

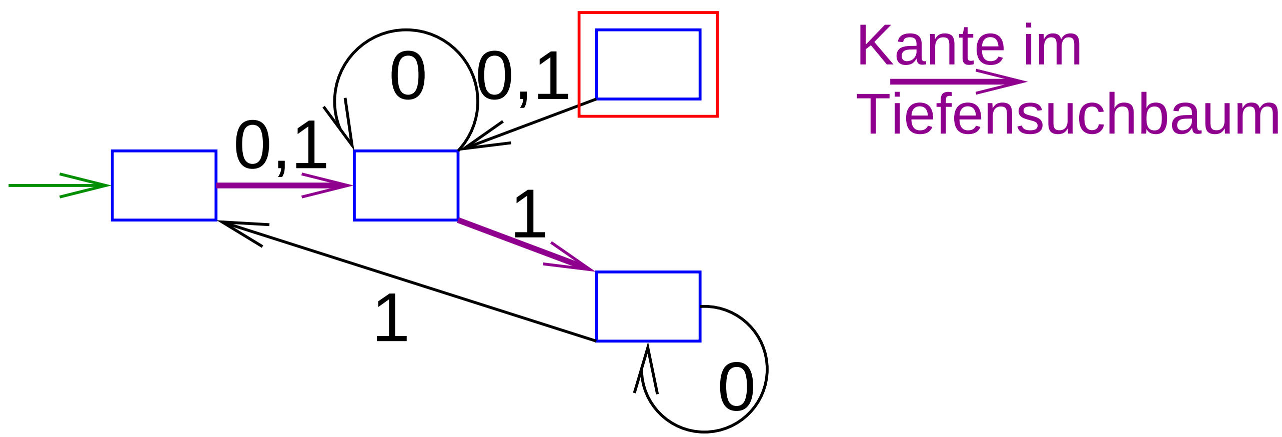
Представяне на DFA или NFA A:

$L = \emptyset \Leftrightarrow \neg\exists f \in F : f$ е от s достижимо

$\rightsquigarrow$ търсене в дълбочина, линейно време, както и за NFA.

Пример

$L(A) = \emptyset$



Проблемът за крайност на езика I — с Pumping Лемата

Нека n е числото от Pumping-Лемата за L - регулярен

Твърдение: $|L(G)| = \infty \Leftrightarrow \exists z \in L(G) : n \leq |z| < 2n$

Д-во:

$z \in L(G), n \leq |z| < 2n \longrightarrow$ Pumping лемата осигурява

$|L| = \infty$.

Ако $|L(G)| = \infty$ да разгледаме $z \in L(G)$ с минимална дължина $|z| \geq n$.

Да допуснем, че $|z| \geq 2n$.

$\xrightarrow{\text{Pumping Лема}} z = uvw$,

$1 \leq |v| \leq |uv| \leq n, uw \in L(G) \longrightarrow |uw| \geq n$.

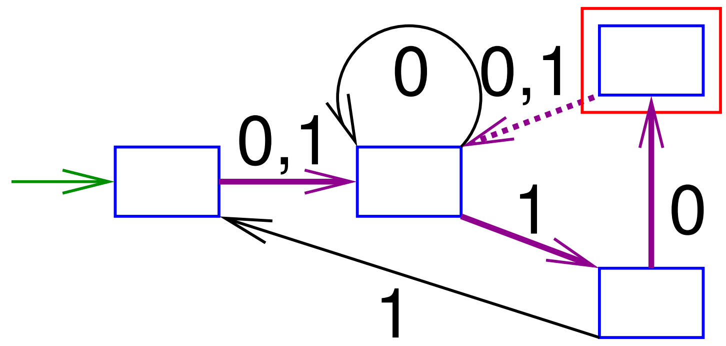
Противоречие с минималността на $|z|$.

Проблемът за крайност на езика II — намиране на цикли

$|L(A)| = \infty$? $\Leftrightarrow \exists$ приемащ път, съдържащ цикъл.
Нека NFA има $F = \{f\}$. Нека $G_A = (Q, E)$,
$E = \{(q,r) : \exists a \in \Sigma \cup \{\varepsilon\} : r \in \delta(q,a)\}$

1. Махаме състоянията, от които f не е достижимо. Търсене в дълбочина в $\bar{G_A} = (Q, \{(q,r) : (r,q) \in E\})$ за f.

2. Можем ли да достигнем от s цикъл? $\Leftrightarrow$ Дали търсенето в дълбочина среща s в G_A от вече посетен възел?



Kante im Tiefensuchbaum

Rückwärtskante im Tiefensuchbaum

Проблемът за пълнота

$L(A) = \Sigma^*$?

$\Leftrightarrow \neg\exists q \in Q \setminus F : q$ е от s достижимо?

$\rightsquigarrow$ търсене в дълбочина, линейно време, само за DFA!

(Еквивалентно: празнота на $\bar{L}$)

Пълнота на NFA:

Трансформираме в DFA. Не е известен по-добър алгоритъм.

Проблемът за еквивалентност

L и L' са регулярни езици разпознавани от DFAs A, A'.

Въпрос $L = L'$?

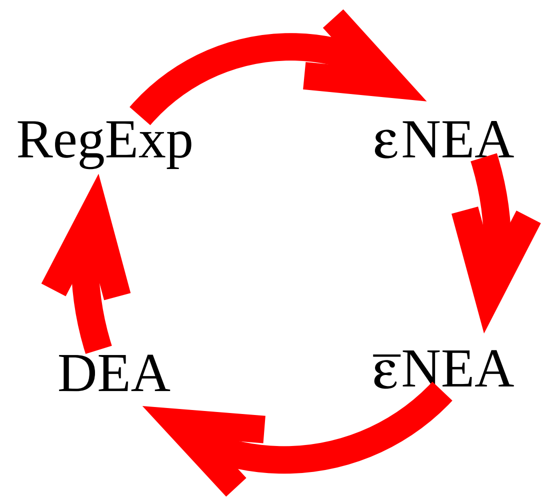
$\Leftrightarrow \neg\exists w : (w \in L \wedge w \notin L') \vee (w \notin L \wedge w \in L')$

$\Leftrightarrow \neg\exists w : (w \in L \wedge w \in \overline{L'}) \vee (w \in \overline{L} \wedge w \in L')$

$\Leftrightarrow (L \cap \overline{L'}) \cup (\overline{L} \cap L') = \emptyset$

за пример с product автомат

Проблем: бавно



Еквивалентност на DFA

L и L' са регулярни езици дефинирани от DFAs
$A = (Q, \Sigma, \delta, s, F)$, $A' = (Q', \Sigma, \delta', s', F')$.

Идея: Минималният автомат е "‘единствен"’.
$\rightsquigarrow$ минимизирайте двата автомата и дикажете, че са "‘равни"’.

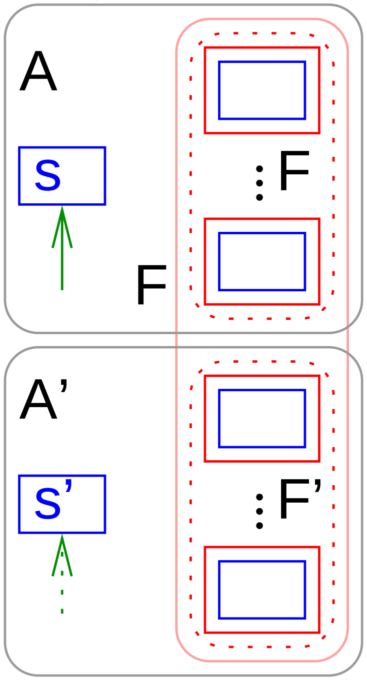
Проблем: Възможно е да са преименувани състоянията.
Сложността на изоморфизъм между по-общи графи е отрит въпрос.

Еквиванетност на DFA

L и L' са регулярни езици дефинирани с DFA

$A=(Q,\Sigma,\delta,s,F)$, $A'=(Q',\Sigma,\delta',s',F')$. Нека $Q\cap Q'=\emptyset$.

Въпрос: $L=L'$?



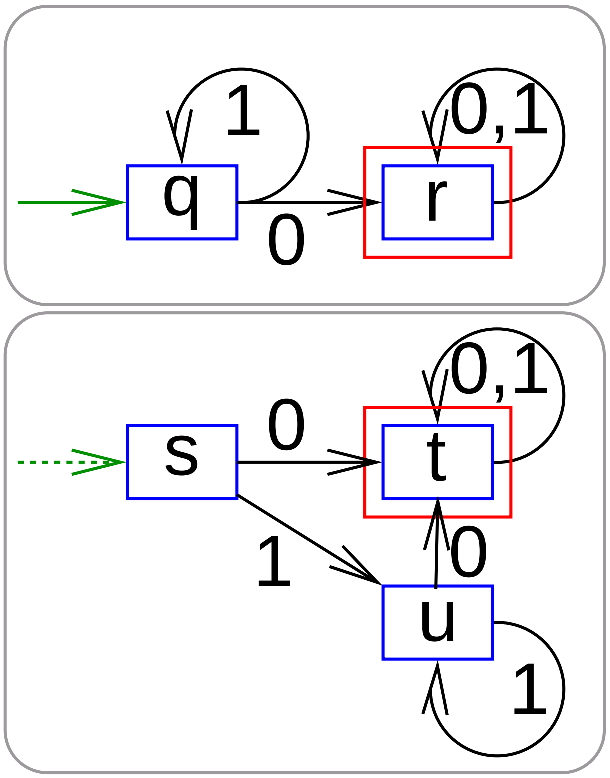
Да разгледаме $A_\cup := (Q\cup Q',\Sigma,\delta_\cup,s,F\cup F')$,

$$\delta_\cup(q,a)=\begin{cases}\delta(q,a) & \text{ако } q\in Q\\ \delta'(q,a) & \text{ако } q\in Q'\end{cases}$$

Намерете класовете на еквивалентност от състояния за $A_\cup$. $L=L' \Leftrightarrow s\equiv s'$.

Пример

$L \subseteq \{0,1\}^*$ език, всички думи с поне една нула



Алгоритъмът за маркиране на нееквивалентните двойки състояния ни дава:
$\{q,r\},\{q,t\},\{s,r\},\{s,t\},\{u,r\},\{u,t\}$

$\rightsquigarrow q \equiv s$

$\rightsquigarrow$ Двата автомата са еквивалентни.

Обобщение на крайни автомати и регулярни езици

- Най-простият машинен модел
- Алгоритмични правила (минимизация на състоянията, трансформиране в регулярен израз,. . .)
- Разпознаваният език е напълно разбираем
- Полезни приложения: обработка на текст, компилатори, . . .
- Концепцията за недетерминизъм