

Kapitola II.

Lexikální analýza:

Modely

Regulární výrazy (RV): Definice

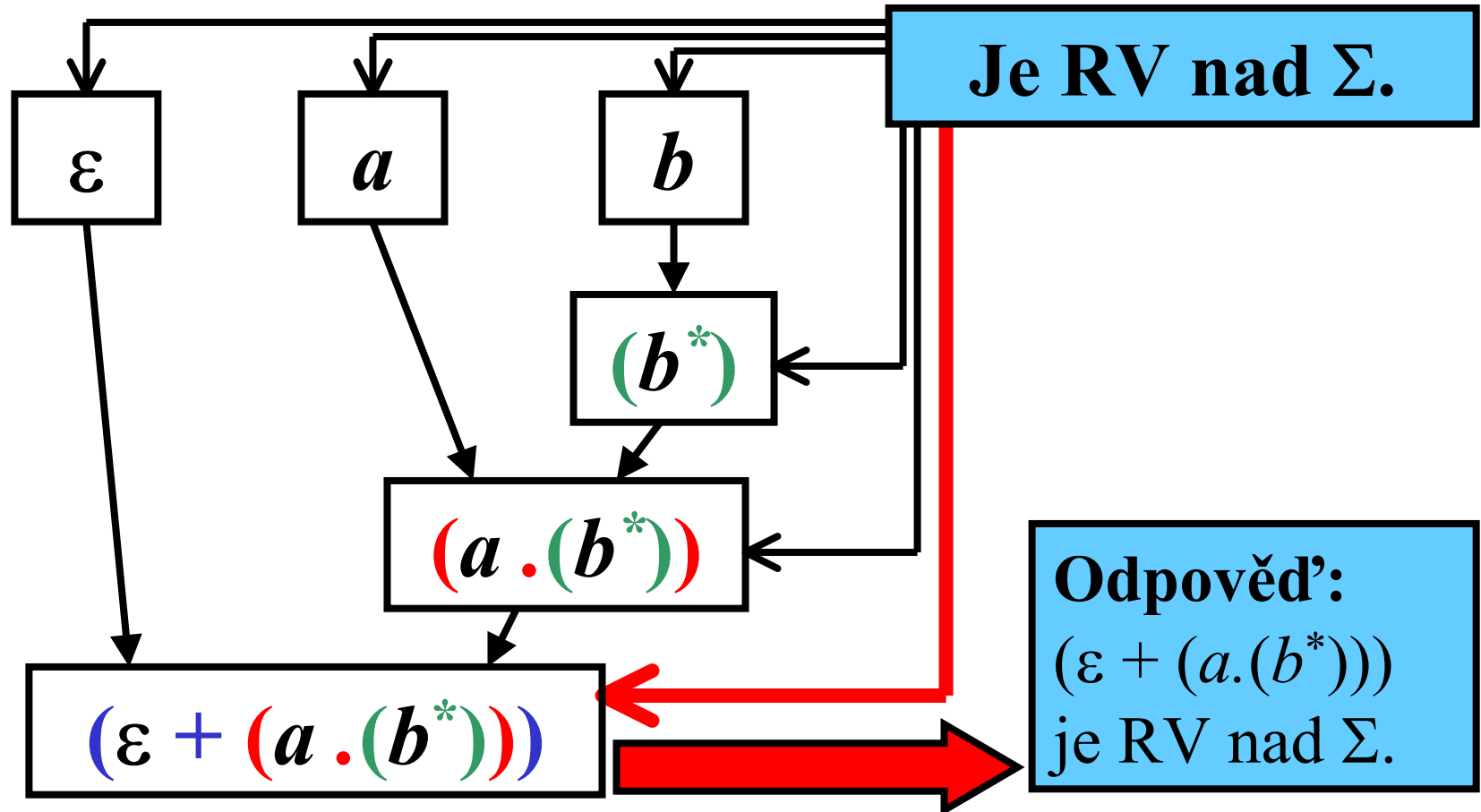
Myšlenka: Jsou to výrazy s operátory $.$, $+$ a $*$, které značí v tomto pořadí konkatenaci, sjednocení a iteraci

Definice: Necht' Σ je abeceda. *Regulární výrazy* nad abecedou Σ a *jazyky*, které značí, jsou definovány následovně:

- \emptyset je RV značící prázdnou množinu (prázdný jazyk)
- ε je RV značící jazyk $\{\varepsilon\}$
- a , kde $a \in \Sigma$, je RV značící jazyk $\{a\}$
- Necht' r a s jsou regulární výrazy značící po řadě jazyky L_r a L_s , potom:
 - $(r.s)$ je RV značící jazyk $L = L_r L_s$
 - $(r + s)$ je RV značící jazyk $L = L_r \cup L_s$
 - (r^*) je RV značící jazyk $L = L_r^*$

Regulární výrazy: Příklad

Otázka: Je $(\varepsilon + (a.(b^*)))$ regulární výraz nad abecedou $\Sigma = \{a, b\}$?



Zjednodušení RV

1) Redukce závorek zavedením priorit operátorů:

Priority: $*$ $>$ $.$ $>$ $+$

2) RV $r.s$ může být zapsán jako rs

3) RV rr^* nebo r^*r může být zapsán jako r^+

Příklad:

$((a.(a^*)) + ((b^*).b))$ může být zapsán $a.a^* + b^*.b$,

a $a.a^* + b^*.b$ může být zapsán $a^+ + b^+$

Regulární jazyk (RJ)

Myšlenka: Každý RV značí regulární jazyk

Definice: Necht' L je jazyk. L je *regulární jazyk* (RJ), pokud existuje regulární výraz r , který tento jazyk značí.

Konvence: $L(r)$ označuje jazyk, který značí RV r .

Příklady:

$$r_1 = ab + ba$$

$$\text{značí } L_1 = \{ab, ba\}$$

$$r_2 = a^+b^*$$

$$\text{značí } L_2 = \{a^n b^m : n \geq 1, m \geq 0\}$$

$$r_3 = ab(a + b)^*$$

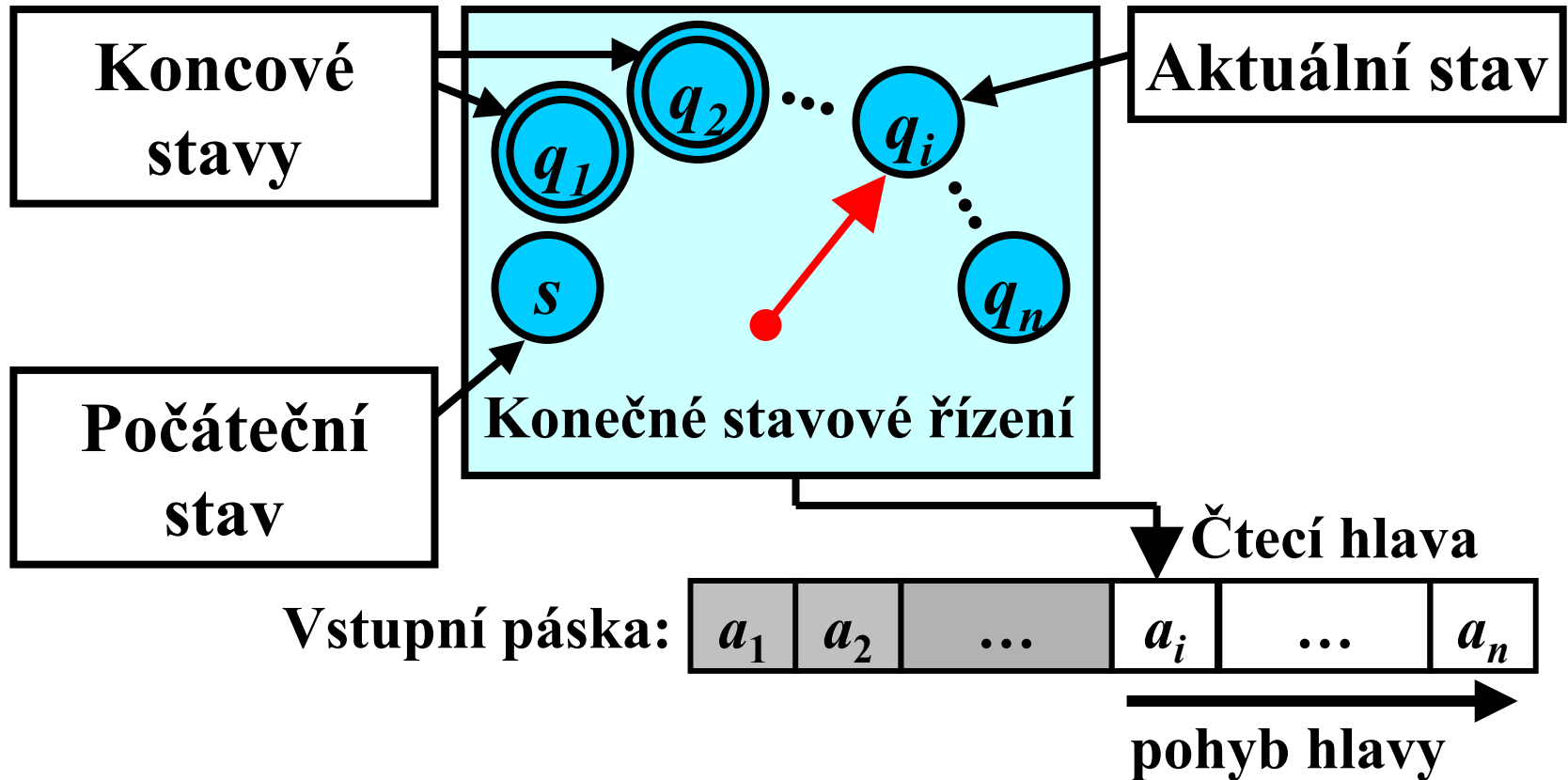
$$\text{značí } L_3 = \{x : ab \text{ je prefix } x\}$$

$$r_4 = (a + b)^* ab(a + b)^* \text{ značí } L_4 = \{x : ab \text{ je podřetězec } x\}$$

L_1, L_2, L_3, L_4 jsou regulární jazyky nad Σ

Konečné automaty (KA)

Myšlenka: Nejjednodušší model založený na konečné množině stavů a výpočetních pravidel.



Konečné automaty: Definice

Definice: *Konečný automat* (KA) je pětice:

$$M = (Q, \Sigma, R, s, F), \text{ kde}$$

- Q je *konečná množina stavů*
- Σ je *vstupní abeceda*
- R je *konečná množina pravidel* tvaru: $pa \rightarrow q$,
kde $p, q \in Q, a \in \Sigma \cup \{\varepsilon\}$
- $s \in Q$ je *počáteční stav*
- $F \subseteq Q$ je *množina koncových stavů*

Matematická poznámka k pravidlům:

- Čistě matematicky, R je relace z $Q \times (\Sigma \cup \{\varepsilon\})$ do Q
 - Místo relačního zápisu $(pa, q) \in R$, zapisujeme: $pa \rightarrow q \in R$
-
- $pa \rightarrow q$ znamená, že při přečtení a M udělá přechod z p do q
 - pokud $a = \varepsilon$, není ze vstupní pásky přečten symbol

Grafická reprezentace

 označuje stav $q \in Q$

 označuje počáteční stav $s \in Q$

 označuje koncový stav $f \in F$

 \xrightarrow{a}  označuje pravidlo $pa \rightarrow q \in R$

Grafická reprezentace: Příklad

$M = (Q, \Sigma, R, s, F)$,

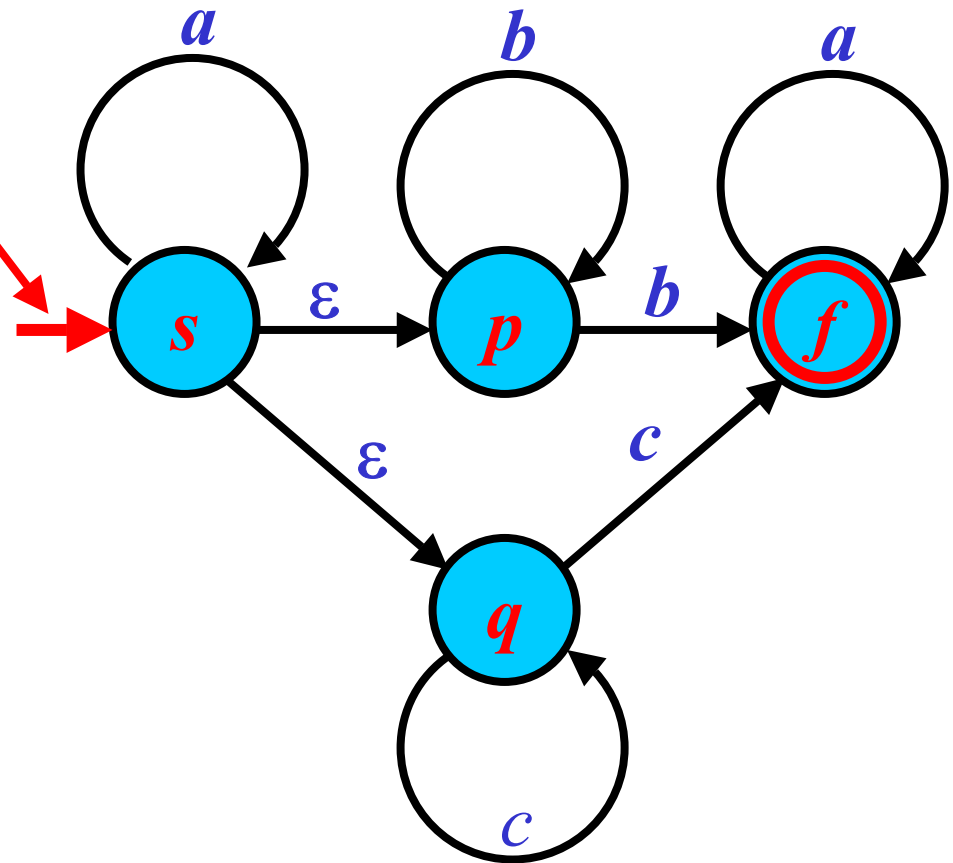
kde:

- $Q = \{s, p, q, f\}$;

- $\Sigma = \{a, b, c\}$;

- $R = \{sa \rightarrow s,$
 $s \rightarrow p,$
 $pb \rightarrow p,$
 $pb \rightarrow f,$
 $s \rightarrow q,$
 $qc \rightarrow q,$
 $qc \rightarrow f,$
 $fa \rightarrow f\}$;

- $F = \{f\}$



Tabulková reprezentace

- **Sloupce:** Prvky z $\Sigma \cup \{\varepsilon\}$
- **Řádky:** Stavy z Q
- **První řádek:** Počáteční stav
- **Podtržené:** Koncové stavy

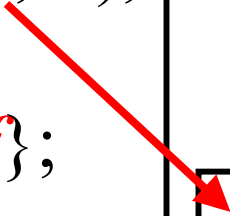
	...	<i>a</i>	...	ε
<i>s</i>				
...				
<i>p</i>		<i>t(p, a)</i>		
...				
<u><i>f</i></u>				

$$t(p, a) = \{q: pa \rightarrow q \in R\}$$

Tabulková reprezentace: Příklad

$M = (Q, \Sigma, R, s, F)$,
kde:

- $Q = \{s, p, q, f\}$;
- $\Sigma = \{a, b, c\}$;
- $R = \{sa \rightarrow s,$
 $s \rightarrow p,$
 $pb \rightarrow p,$
 $pb \rightarrow f,$
 $s \rightarrow q,$
 $qc \rightarrow q,$
 $qc \rightarrow f,$
 $fa \rightarrow f\}$;
- $F = \{f\}$

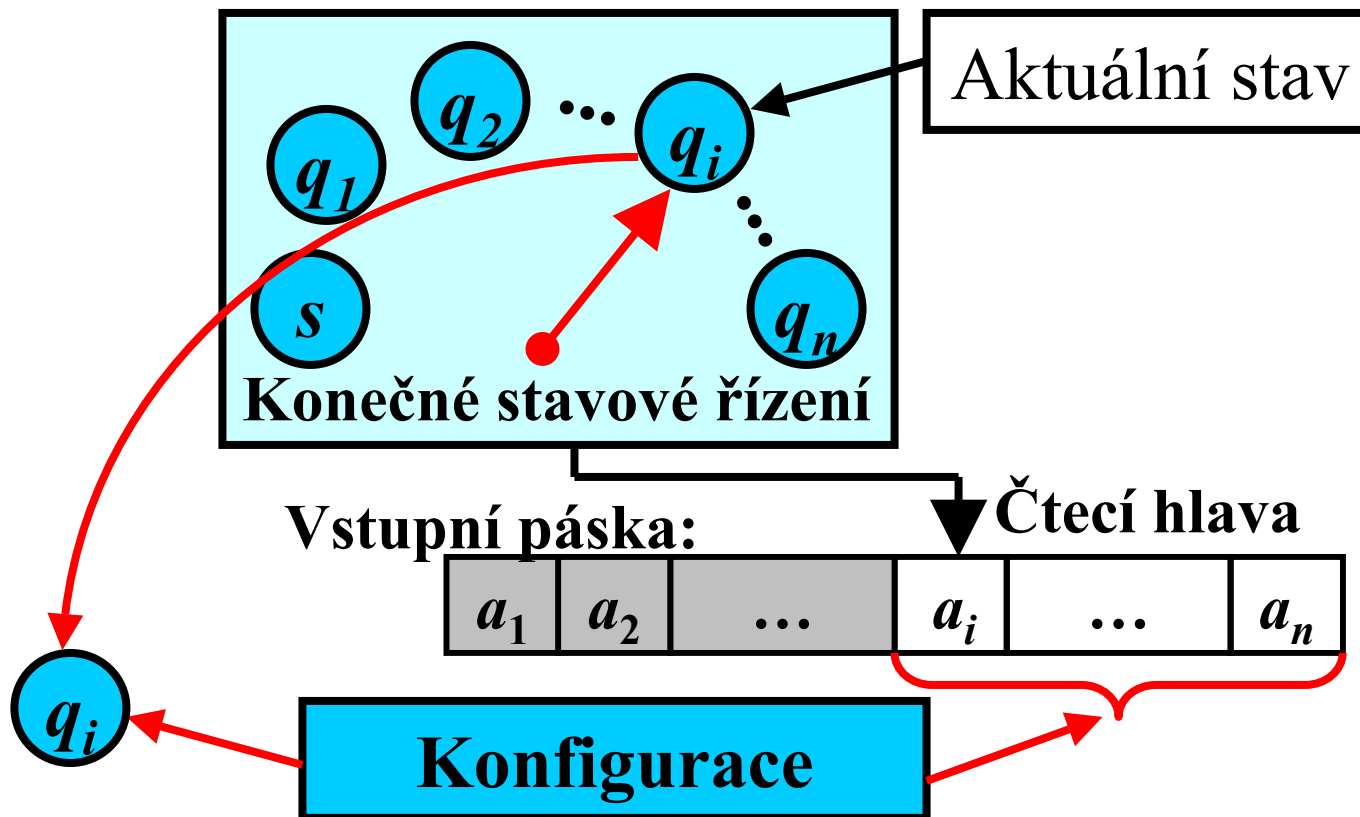


	<i>a</i>	<i>b</i>	<i>c</i>	ϵ
<i>s</i>	{ <i>s</i> }	\emptyset	\emptyset	{ <i>p, q</i> }
<i>p</i>	\emptyset	{ <i>p, f</i> }	\emptyset	\emptyset
<i>q</i>	\emptyset	\emptyset	{ <i>q, f</i> }	\emptyset
<i>f</i>	{ <i>f</i> }	\emptyset	\emptyset	\emptyset

Konfigurace

Myšlenka: Instance popisu KA

Definice: Necht' $M = (Q, \Sigma, R, s, F)$ je KA.
Konfigurace KA M je řetězec $\chi \in Q\Sigma^*$



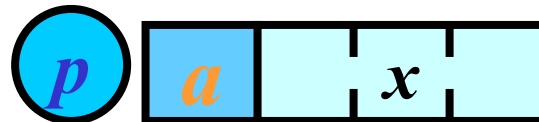
Přechod

Myšlenka: Jeden výpočetní krok KA

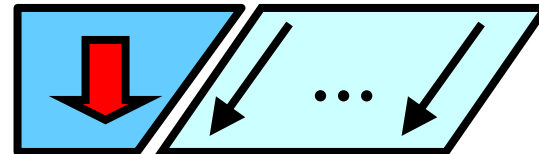
Definice: Necht' pa a qx jsou dvě konfigurace KA M , kde $p, q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$ a $x \in \Sigma^*$. Necht' $r = pa \rightarrow q \in R$ je pravidlo. Potom M může provést *přechod* z pa do qx za použití r , zapsáno $pa \vdash qx [r]$ nebo zjednodušeně $pa \vdash qx$

Pozn.: pokud $a = \varepsilon$, není ze vstupní pásky přečten symbol

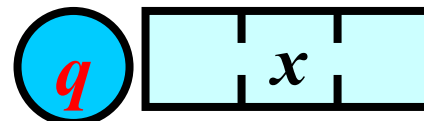
Konfigurace:



Pravidlo: $pa \rightarrow q$



Nová konfigurace:



Sekvence přechodů 1/2

Myšlenka: několik výpočetních kroků po sobě

Definice: Necht' χ je konfigurace. M provede *nula přechodů* z χ do χ ; zapisujeme:

$$\chi \vdash^0 \chi [\varepsilon] \text{ nebo zjednodušeně } \chi \vdash^0 \chi$$

Definice: Necht' $\chi_0, \chi_1, \dots, \chi_n$ je sekvence přechodů konfigurací pro $n \geq 1$ a $\chi_{i-1} \vdash \chi_i [r_i]$, $r_i \in R$ pro všechna $i = 1, \dots, n$, což znamená:

$$\chi_0 \vdash \chi_1 [r_1] \vdash \chi_2 [r_2] \dots \vdash \chi_n [r_n]$$

Pak M provede *n-přechodů* z χ_0 do χ_n ; zapisujeme:

$$\chi_0 \vdash^n \chi_n [r_1 \dots r_n] \text{ nebo zjednodušeně } \chi_0 \vdash^n \chi_n$$

Sekvence přechodů 2/2

Pokud $\chi_0 \vdash^{-n} \chi_n [\rho]$ pro nějaké $n \geq 1$, pak
 $\chi_0 \vdash^+ \chi_n [\rho]$.

Pokud $\chi_0 \vdash^{-n} \chi_n [\rho]$ pro nějaké $n \geq 0$, pak
 $\chi_0 \vdash^* \chi_n [\rho]$.

Příklad: Uvažujme

$abc \vdash qbc$ [1: $pa \rightarrow q$] a $qbc \vdash rc$ [2: $qb \rightarrow r$].

Potom: $abc \vdash^{-2} rc$ [1 2],

$abc \vdash^+ rc$ [1 2],

$abc \vdash^* rc$ [1 2]

Přijímaný jazyk

Myšlenka: M přijímá řetězec w , pokud je celý přečten pomocí sekvencí přechodů a skončí v nějakém koncovém stavu

Definice: Necht' $M = (Q, \Sigma, R, s, F)$ je KA.
Jazyk přijímaný konečným automatem M , $L(M)$, je definován:

$$L(M) = \{w: w \in \Sigma^*, sw \stackrel{*}{\vdash} f, f \in F\}$$

$M = (Q, \Sigma, R, s, F)$:

pokud $q_n \in F$ pak $w \in L(M)$;
jinak $w \notin L(M)$

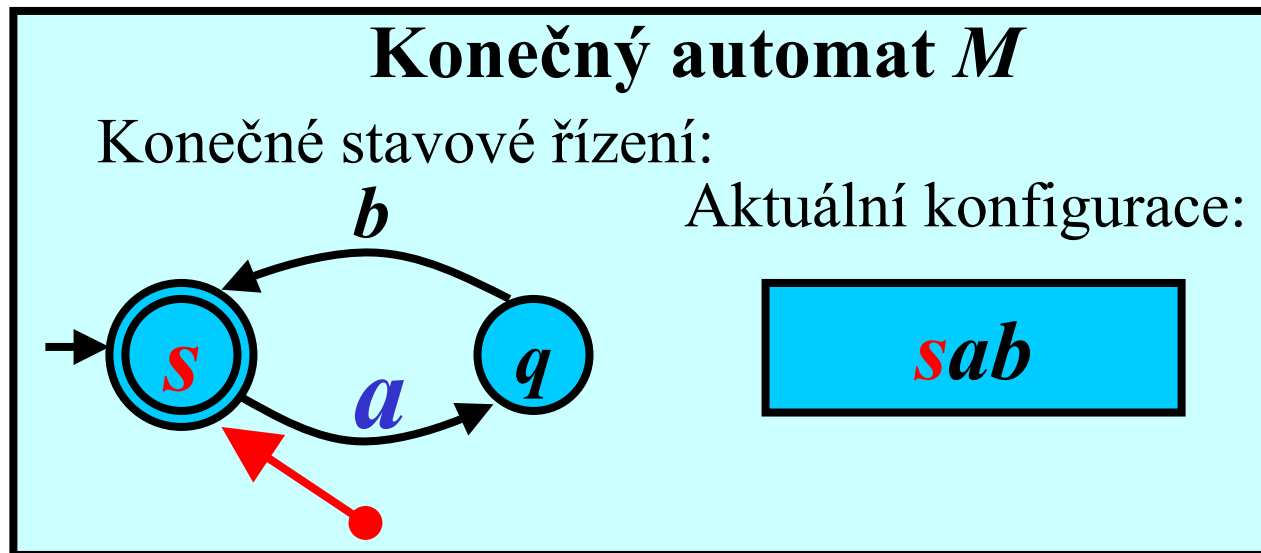
$$s \underbrace{a_1 a_2 \dots a_n}_w \vdash q_1 a_2 \dots a_n \vdash \dots \vdash q_{n-1} a_n \vdash q_n$$

Konečný automat: Příklad 1/3

$M = (Q, \Sigma, R, s, F)$, kde:

$Q = \{s, q\}$, $\Sigma = \{a, b\}$, $R = \{sa \rightarrow q, qb \rightarrow s\}$, $F = \{s\}$

Otázka: $ab \in L(M)$?



Vstupní páska:

a	b
-----	-----

 sab

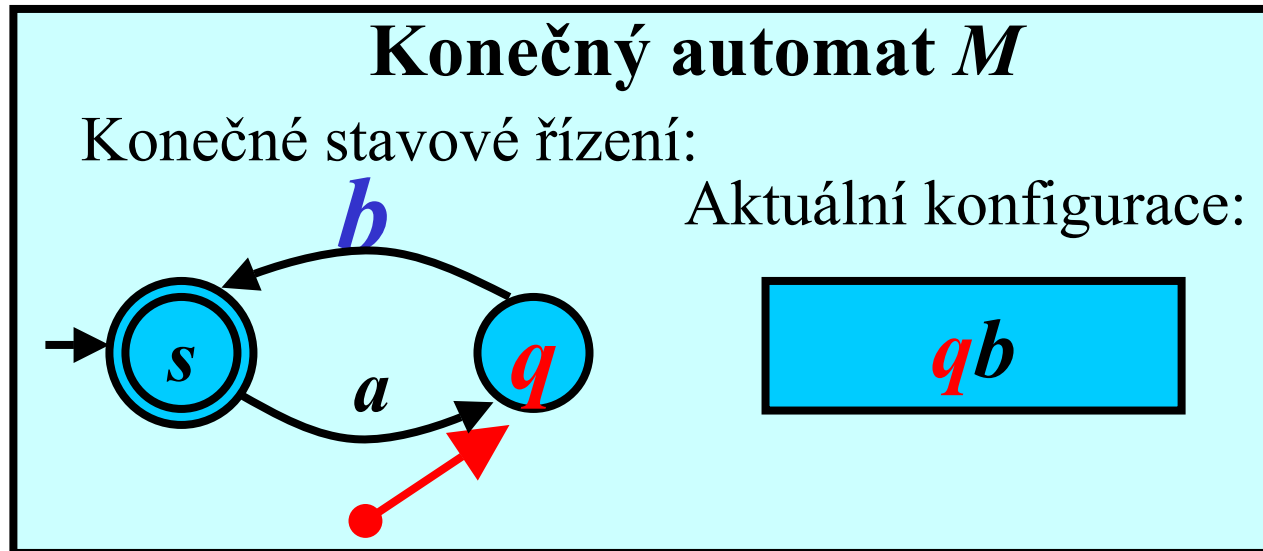
Čtecí hlava

Konečný automat: Příklad 2/3

$M = (Q, \Sigma, R, s, F)$, where:

$Q = \{s, q\}$, $\Sigma = \{a, b\}$, $R = \{sa \rightarrow q, qb \rightarrow s\}$, $F = \{s\}$

Question: $ab \in L(M)$?



Vstupní páska:

a	b
-----	-----

 Čtecí hlava

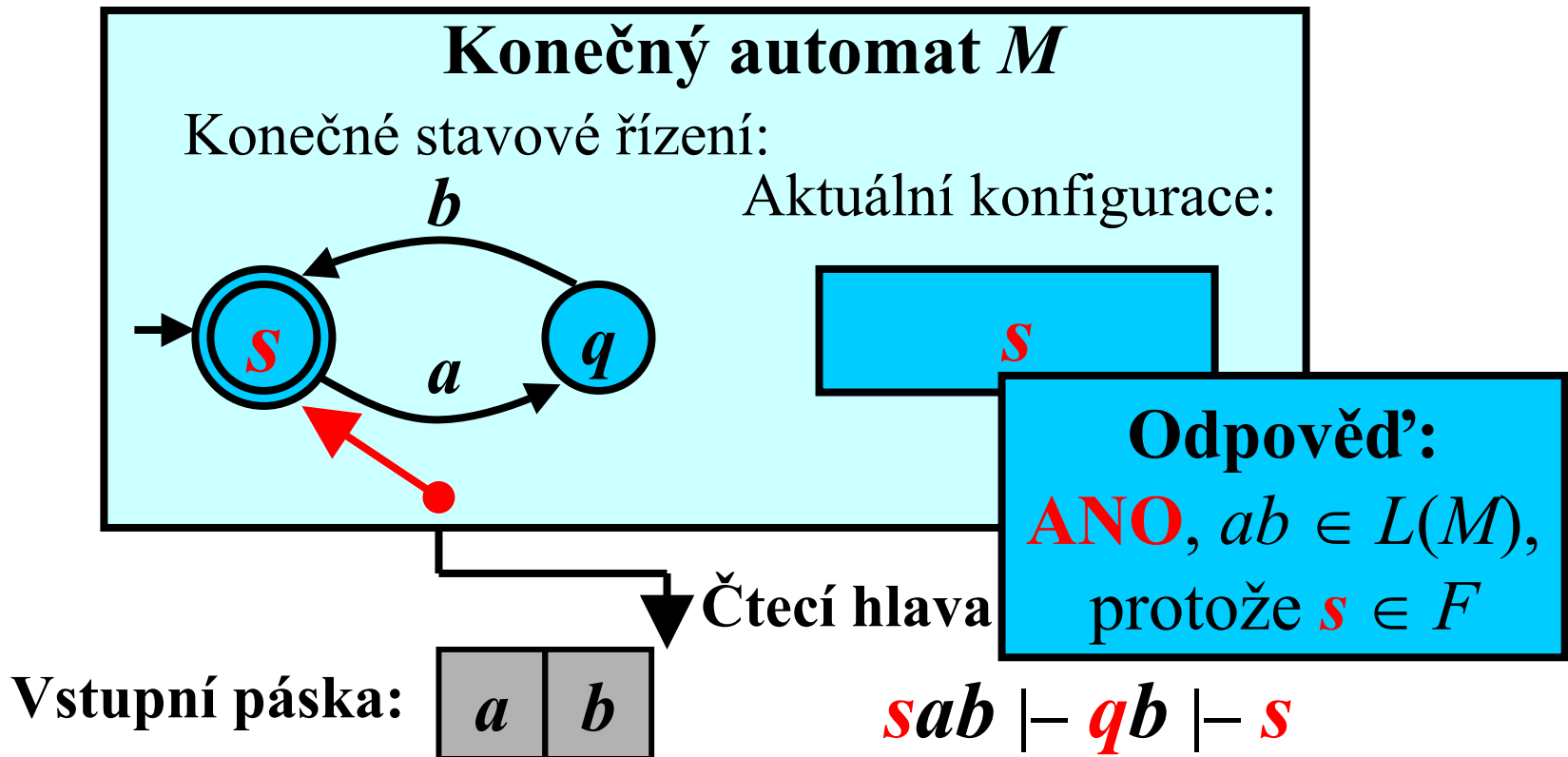
$sab \mid - qb$

Konečný automat: Příklad 3/3

$M = (Q, \Sigma, R, s, F)$, where:

$Q = \{s, q\}$, $\Sigma = \{a, b\}$, $R = \{sa \rightarrow q, qb \rightarrow s\}$, $F = \{s\}$

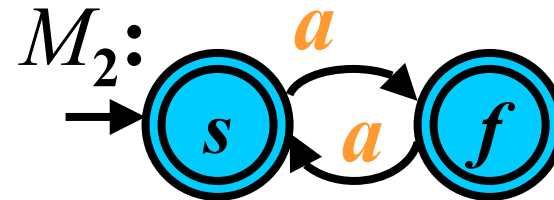
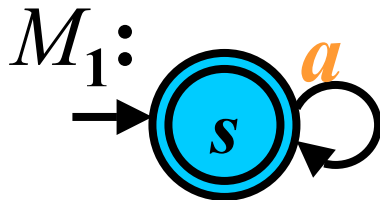
Question: $ab \in L(M)$?



Ekvivalentní modely

Definice: Dva modely pro popis formálních jazyků (např. končené automaty) jsou ekvivalentní, pokud specifikují tentýž jazyk.

Příklad:



Otázka: Je M_1 ekvivalentní s M_2 ?

Odpověď: M_1 a M_2 jsou ekvivalentní, protože

$$L(M_1) = L(M_2) = \{a^n : n \geq 0\}$$

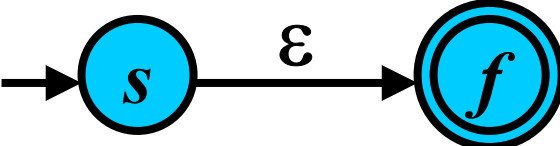
Převod z RV na KA: Základy 1/5

Myšlenka: Algoritmus, který převede libovolný RV na ekvivalentní KA

- Pro RV $r = \emptyset$ existuje ekvivalentní KA M_{\emptyset} .

Důkaz: $M_{\emptyset} :$ 

- Pro RV $r = \varepsilon$ existuje ekvivalentní KA M_{ε} .

Důkaz: $M_{\varepsilon} :$ 

- Pro RV $r = a$, $a \in \Sigma$ existuje ekvivalentní KA M_a .

Důkaz: $M_a :$ 

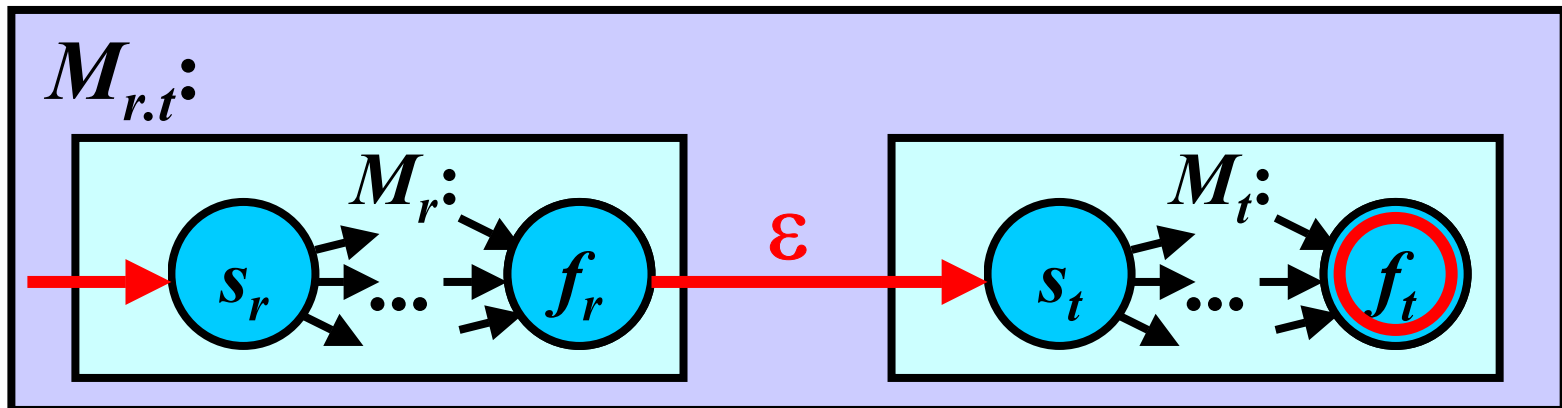
Převod z RV na KA: Konkatence 2/5

- Necht' r je RV nad Σ a $M_r = (Q_r, \Sigma, R_r, s_r, \{f_r\})$ je KA pro který platí $L(M_r) = L(r)$.
- Necht' t je RV nad Σ a $M_t = (Q_t, \Sigma, R_t, s_t, \{f_t\})$ je KA pro který platí $L(M_t) = L(t)$.
- Potom pro RV $r.t$ existuje ekvivalentní KA $M_{r.t}$

Důkaz: Necht' $Q_r \cap Q_t = \emptyset$.

Popis konstrukce:

$$M_{r.t} = (Q_r \cup Q_t, \Sigma, R_r \cup R_t \cup \{f_r \rightarrow s_t\}, s_r, \{f_t\})$$



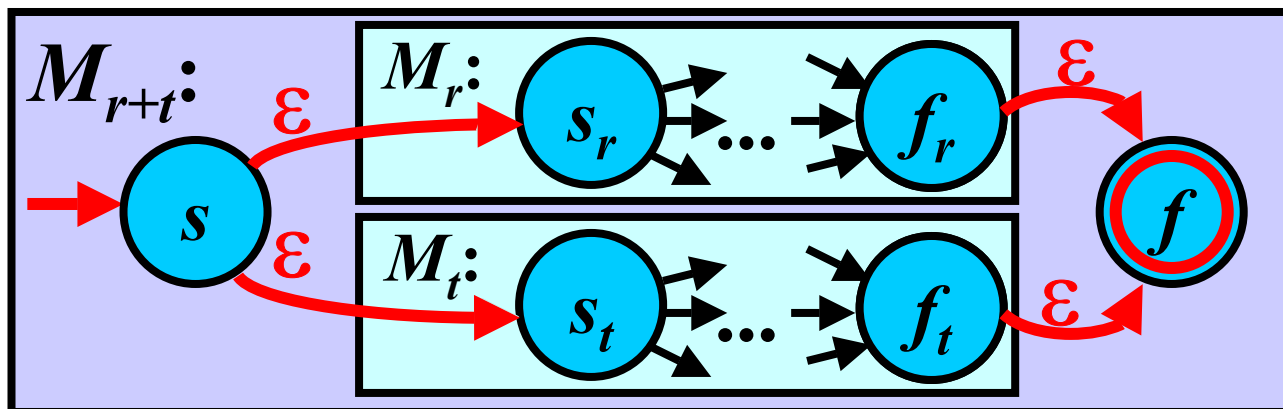
Převod z RV na KA: Sjednocení 3/5

- Necht' r je RV nad Σ a $M_r = (Q_r, \Sigma, R_r, s_r, \{f_r\})$ je KA pro který platí $L(M_r) = L(r)$.
- Necht' t je RV nad Σ a $M_t = (Q_t, \Sigma, R_t, s_t, \{f_t\})$ je KA pro který platí $L(M_t) = L(t)$.
- Potom pro RV $r + t$ existuje ekvivalentní KA M_{r+t}

Důkaz: Necht' $Q_r \cap Q_t = \emptyset$; $s, f \notin Q_r \cup Q_t$.

Popis konstrukce:

$$M_{r+t} = (Q_r \cup Q_t \cup \{s, f\}, \Sigma, R_r \cup R_t \cup \{s \rightarrow s_r, s \rightarrow s_t, f_r \rightarrow f, f_t \rightarrow f\}, s, \{f\})$$



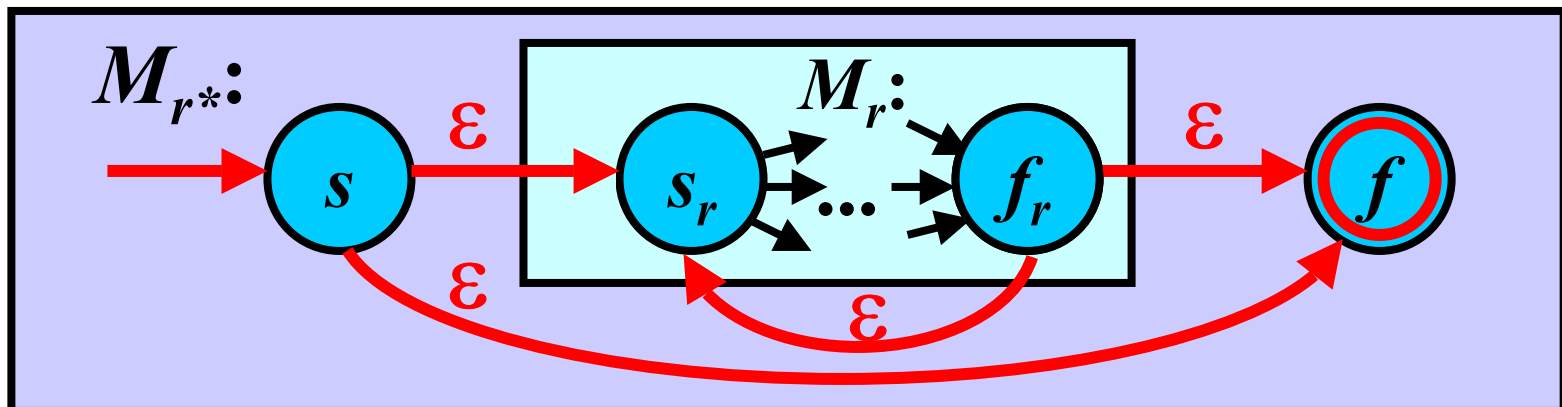
Převod z RV na KA: Iterace 4/5

- Necht' r je RV nad Σ a $M_r = (Q_r, \Sigma, R_r, s_r, \{f_r\})$ je KA pro který platí $L(M_r) = L(r)$.
- Potom pro RV r^* existuje ekvivalentní KA M_{r^*}

Důkaz: Necht' $s, f \notin Q_r$.

Popis konstrukce:

$$M_{r^*} = (Q_r \cup \{s, f\}, \Sigma, R_r \cup \{s \rightarrow s_r, f_r \rightarrow f, f_r \rightarrow s_r, s \rightarrow f\}, s, \{f\})$$

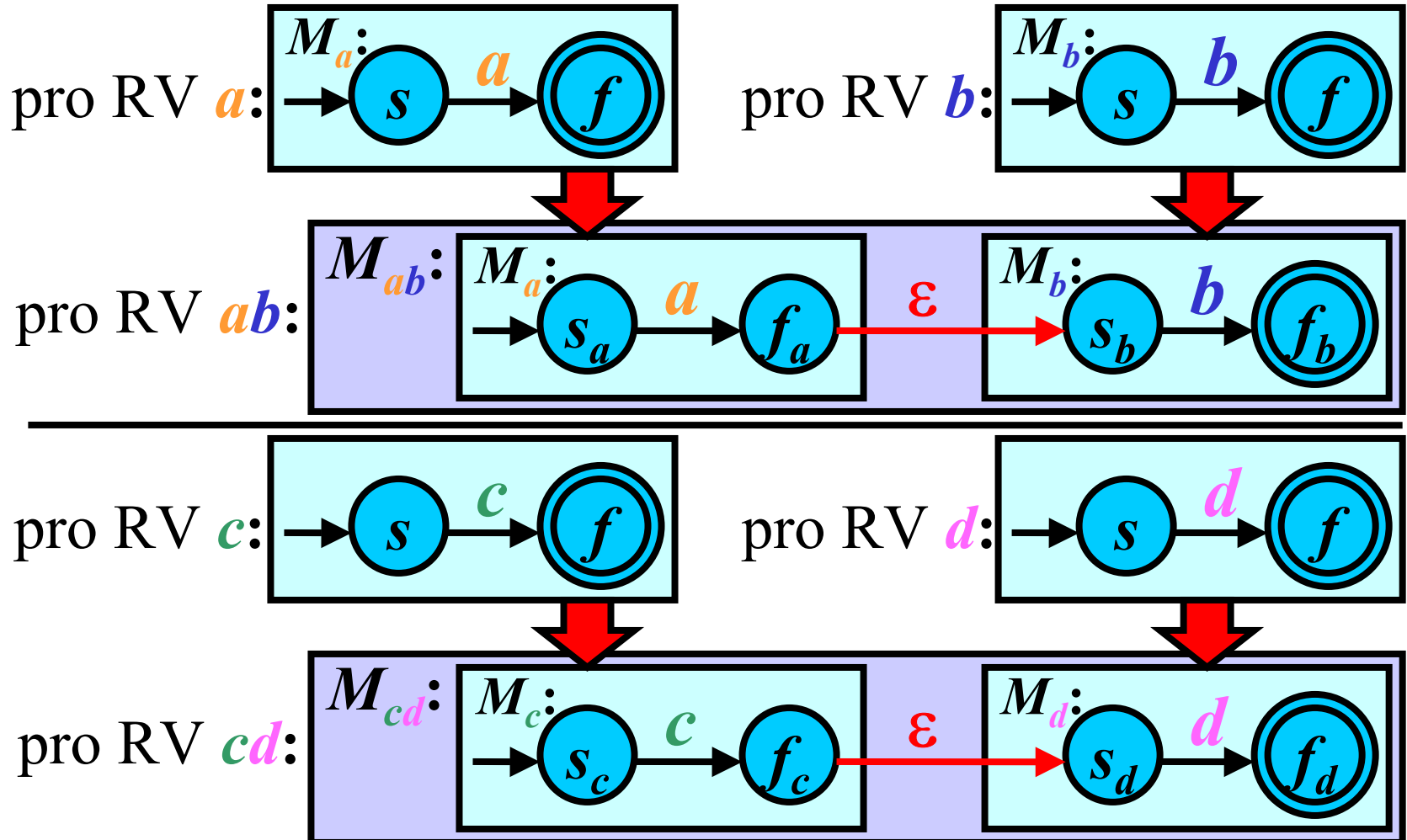


Převod z RV na KA: Souhrn 5/5

- **Vstup:** RV r nad Σ
 - **Výstup:** KA M , pro který platí: $L(r) = L(M)$
-
- **Metoda:**
 - “Zevnitř” RV r opakovaně použij následující pravidla ke konstrukci konečného automatu M :
 - pro RV \emptyset vytvoř KA M_{\emptyset}
 - pro RV ε vytvoř KA M_{ε}
 - pro RV $a \in \Sigma$ vytvoř KA M_a
 - Necht’ pro RV r a t již existují po řadě KA M_r a M_t
- } \longrightarrow (viz. 1/5)
- Potom:**
- pro RV $r.t$ vytvoř KA $M_{r.t}$ (viz. 2/5)
 - pro RV $r + t$ vytvoř KA M_{r+t} (viz. 3/5)
 - pro RV r^* vytvoř KA M_{r^*} (viz. 4/5)

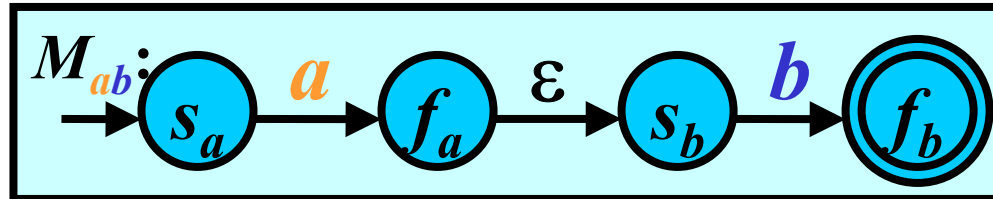
Převod z RV na KA: Příklad 1/3

Převodme RV $r = ((ab) + (cd))^*$ na ekvivalentní KA M

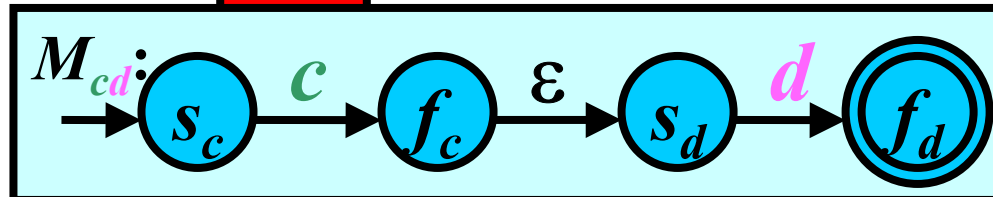


Převod z RV na KA: Příklad 2/3

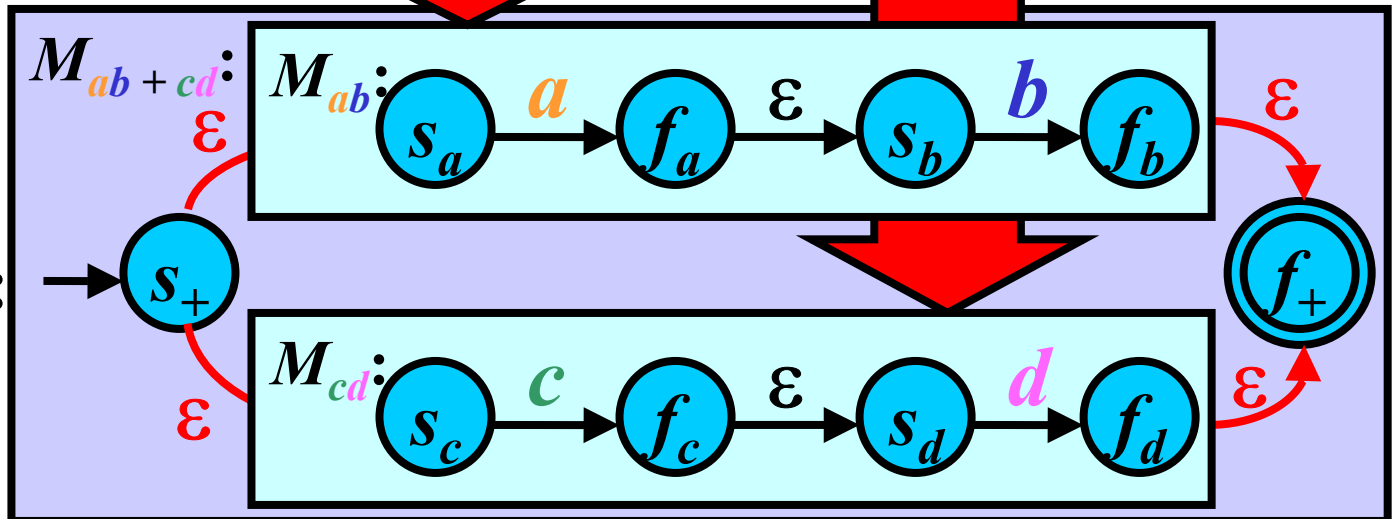
pro RV ab :



pro RV cd :

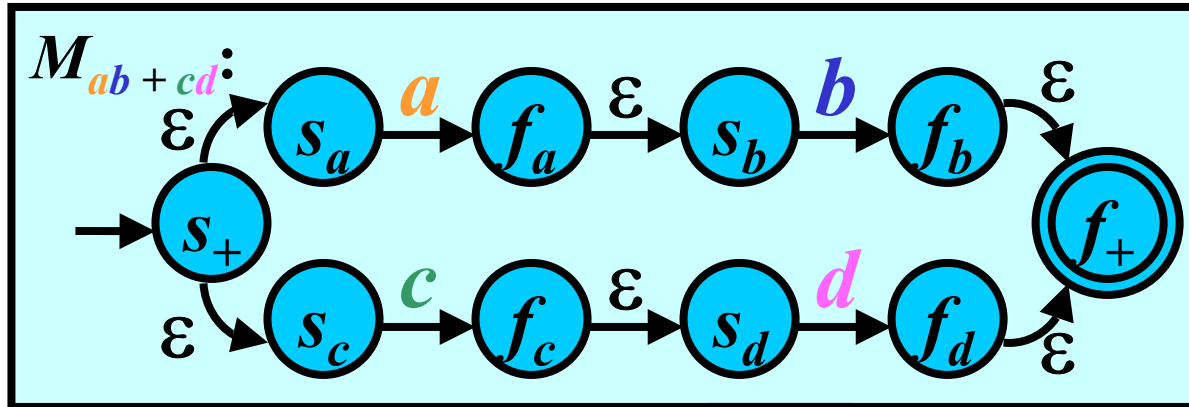


pro RV
 $ab + cd$:

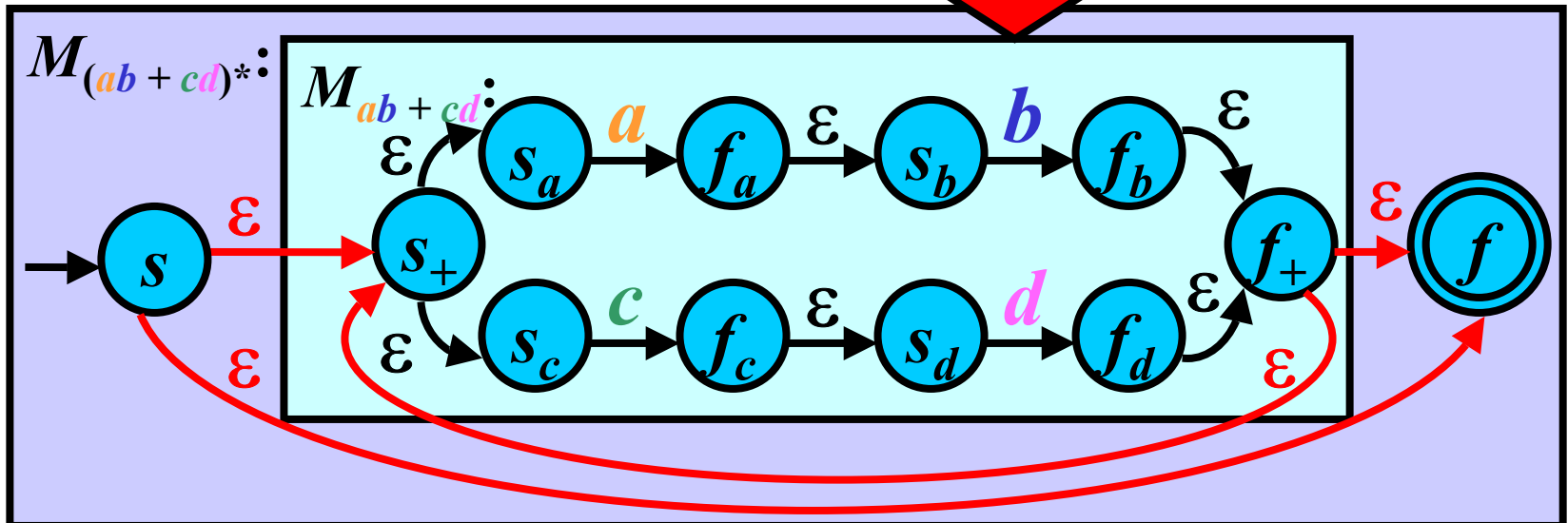


Převod z RV na KA: Příklad 3/3

pro RV
 $ab + cd$:



pro výsledný RV $(ab + cd)^*$:



Modely pro regulární jazyky

Tvrzení: Pro každý RV r existuje KA M , pro který platí: $L(r) = L(M)$.

Důkaz je založen na předchozím algoritmu.

Tvrzení: Pro každý KA M existuje RV r , pro který platí $L(M) = L(r)$.

Důkaz: viz. str. 210 v knize [Meduna: Automata a Languages]

Závěr: Fundamentální modely pro regulární jazyky jsou:

- 1) **Regulární výrazy**
- 2) **Konečné automaty**