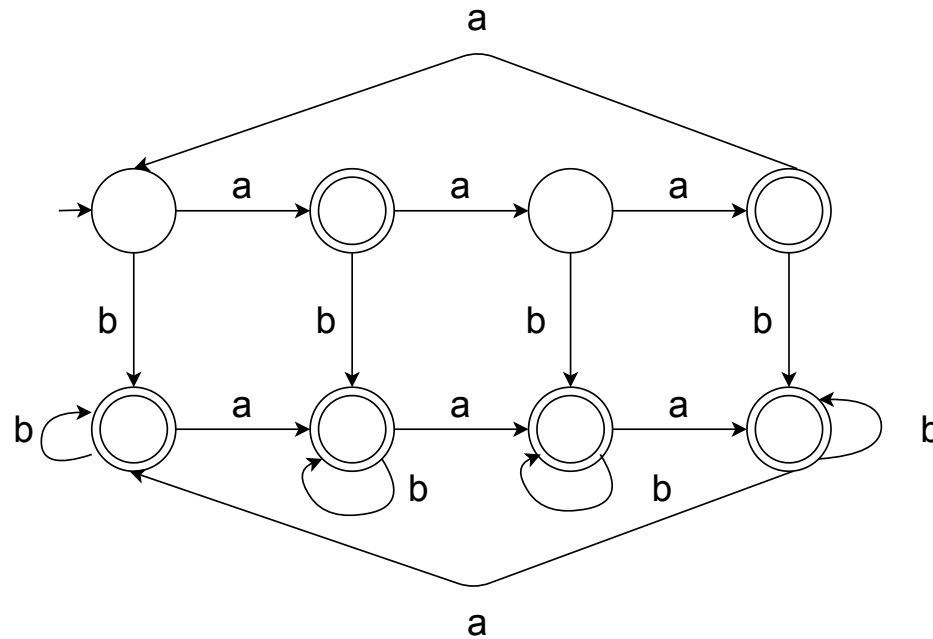


Minimalizace Konečných Automatů

Motivační příklad

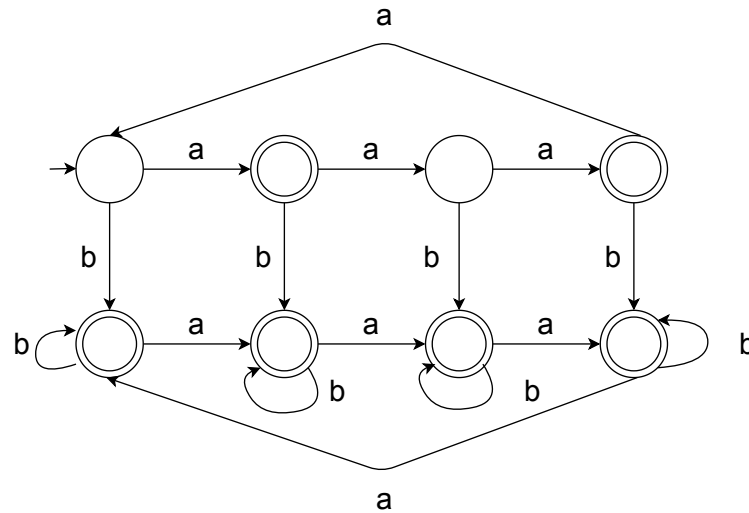
❖ Uvažme jazyk $L = \{w \in \{a, b\}^* \mid \#_a(w) \bmod 2 = 1 \vee \#_b(w) > 0\}$



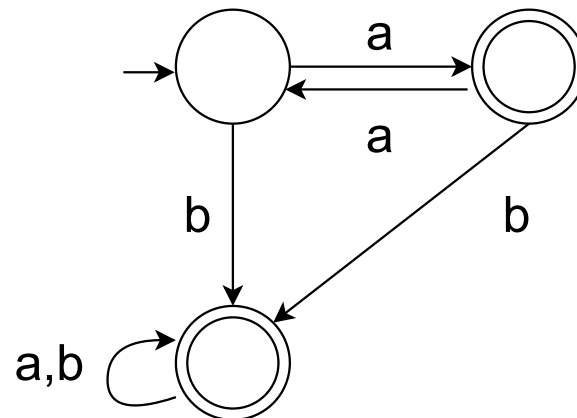
❖ Existuje menší automat akceptující jazyk L ?

Motivační příklad

- ❖ Uvažme jazyk $L = \{w \in \{a, b\}^* \mid \#_a(w) \bmod 2 = 1 \vee \#_b(w) > 0\}$



- ❖ Existuje menší automat akceptující jazyk L ?



Eliminace nedosažitelných stavů

Definice 2.1 Necht' $M = (Q, \Sigma, \delta, q_0, F)$ je konečný automat. Stav $q \in Q$ nazveme **dosažitelný**, pokud existuje $w \in \Sigma^*$ takové, že $(q_0, w) \stackrel{*}{\vdash}_M (q, \varepsilon)$. Stav je **nedosažitelný**, pokud není dosažitelný.

Algoritmus 2.1 Eliminace nedosažitelných stavů

Vstup: DKA $M = (Q, \Sigma, \delta, q_0, F)$.

Výstup: DKA M' bez nedosažitelných stavů, $L(M) = L(M')$.

Metoda:

1. $i := 0$
2. $S_i := \{q_0\}$
3. repeat
4. $S_{i+1} := S_i \cup \{q \mid \exists p \in S_i \exists a \in \Sigma : \delta(p, a) = q\}$
5. $i := i + 1$
6. until $S_i = S_{i-1}$
7. $M' := (S_i, \Sigma, \delta|_{S_i}, q_0, F \cap S_i)$

Jazykově nerozlišitelné stavy

Definice 2.2

- Necht' $M = (Q, \Sigma, \delta, q_0, F)$ je úplně definovaný DKA. Říkáme, že řetězec $w \in \Sigma^*$ rozlišuje q_1, q_2 , jestliže $(q_1, w) \stackrel{*}{\vdash}_M (q_3, \varepsilon) \wedge (q_2, w) \stackrel{*}{\vdash}_M (q_4, \varepsilon)$ pro nějaké q_3, q_4 a právě jeden ze stavů q_3, q_4 je v F .
- Říkáme, že stavy $q_1, q_2 \in Q$ jsou k -nerozlišitelné a píšeme $q_1 \stackrel{k}{\equiv} q_2$, právě když neexistuje $w \in \Sigma^*$, $|w| \leq k$, který rozlišuje q_1 a q_2 .
- Stavy q_1, q_2 jsou nerozlišitelné, značíme $q_1 \equiv q_2$, jsou-li pro každé $k \geq 0$ k -nerozlišitelné.

❖ **Poznámka:** Dá se snadno dokázat, že \equiv je relací ekvivalence na Q , tj. relací, která je reflexivní, symetrickou a tranzitivní.

Definice 2.3 Úplně definovaný DKA M nazýváme **redukovaný**, jestliže žádný stav z Q není nedostupný a žádné dva stavy nerozlišitelné.

Věta 2.1 Necht' $M = (Q, \Sigma, \delta, q_0, F)$ je úplně definovaný DKA a $|Q| = n, n \geq 2$. Platí

$$\forall q_1, q_2 \in Q : q_1 \equiv q_2 \Leftrightarrow q_1 \stackrel{n-2}{\equiv} q_2.$$

Důkaz. „ \Rightarrow “ triviální, ukážeme „ \Leftarrow “:

1. Jestliže $|F| = 0$ nebo $|F| = n$, pak platí $q_1 \stackrel{n-2}{\equiv} q_2 \Rightarrow q_1 \equiv q_2$.

2. Necht' $|F| > 0 \wedge |F| < n$. Ukážeme, že platí $\equiv = \stackrel{n-2}{\equiv} \subseteq \stackrel{n-3}{\equiv} \subseteq \dots \subseteq \stackrel{1}{\equiv} \subseteq \stackrel{0}{\equiv}$:

- Zřejmě platí:

(a) $\forall q_1, q_2 \in Q : q_1 \stackrel{0}{\equiv} q_2 \Leftrightarrow (q_1 \in F \wedge q_2 \in F) \vee (q_1 \notin F \wedge q_2 \notin F)$, tj.

$$q_1 \stackrel{0}{\equiv} q_2 \Leftrightarrow (q_1 \in F \Leftrightarrow q_2 \in F).$$

(b) $\forall q_1, q_2 \in Q \forall k \geq 1 : q_1 \stackrel{k}{\equiv} q_2 \Leftrightarrow (q_1 \stackrel{k-1}{\equiv} q_2 \wedge \forall a \in \Sigma : \delta(q_1, a) \stackrel{k-1}{\equiv} \delta(q_2, a))$.

- Relace $\stackrel{0}{\equiv}$ je ekvivalencí určující rozklad $\{F, Q \setminus F\}$.

- Je-li $\stackrel{k+1}{\equiv} \neq \stackrel{k}{\equiv}$, pak $\stackrel{k+1}{\equiv}$ je vlastním zjemněním $\stackrel{k}{\equiv}$, tj. obsahuje alespoň o jednu třídu více než rozklad $\stackrel{k}{\equiv}$.

- Jestliže pro nějaké k platí $\stackrel{k+1}{\equiv} = \stackrel{k}{\equiv}$, pak také $\stackrel{k+1}{\equiv} = \stackrel{k+2}{\equiv} = \stackrel{k+3}{\equiv} = \dots$ podle (b) a tedy $\stackrel{k}{\equiv}$ je hledaná ekvivalence.

- Protože F nebo $Q \setminus F$ obsahuje nejvýše $n - 1$ prvků, získáme relaci \equiv po nejvýše $n - 2$ zjemněních $\stackrel{0}{\equiv}$.

□

Převod na redukovaný DKA

Algoritmus 2.2 Převod na redukovaný DKA

Vstup: Úplně definovaný DKA $M = (Q, \Sigma, \delta, q_0, F)$.

Výstup: Redukovaný DKA $M' = (Q', \Sigma, \delta', q'_0, F')$, $L(M) = L(M')$.

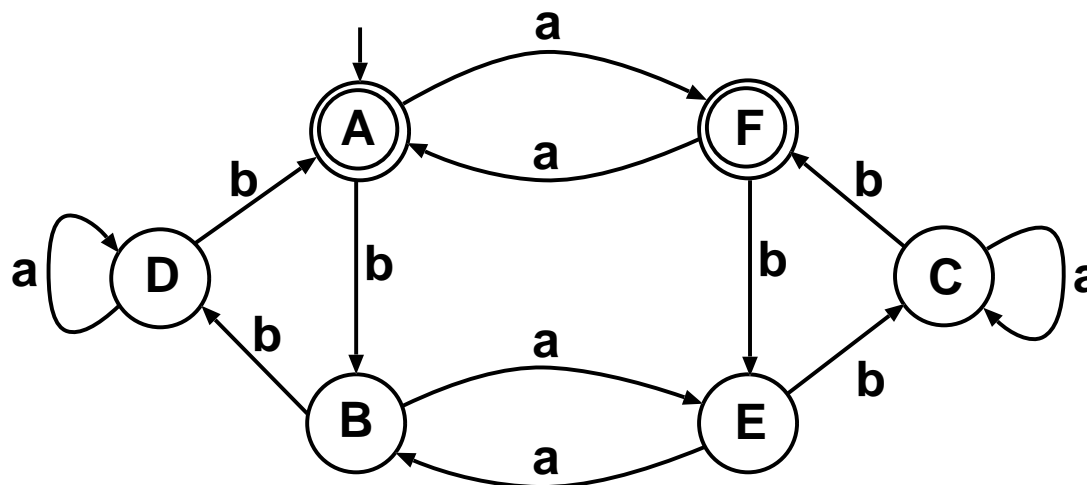
Metoda:

1. Odstraň nedostupné stavy s využitím alg. 2.1.
2. $i := 0$
3. $\equiv^0 := \{(p, q) \mid p \in F \iff q \in F\}$
4. repeat
5. $\equiv^{i+1} := \{(p, q) \mid p \equiv^i q \wedge \forall a \in \Sigma : \delta(p, a) \equiv^i \delta(q, a)\}$
6. $i := i + 1$
7. until $\equiv^i = \equiv^{i-1}$
8. $Q' := Q / \equiv^i$
9. $\forall p, q \in Q \forall a \in \Sigma : \delta'([p], a) = [q] \iff \delta(p, a) = q$
10. $q'_0 = [q_0]$
11. $F' = \{[q] \mid q \in F\}$

❖ *Poznámka:* Výraz $[x]$ značí ekvivalenční třídu určenou prvkem x .

Příklad minimalizace DKA

Příklad 2.1 Převeďte níže uvedený DKA (zadaný diagram přechodů) na odpovídající redukovaný DKA.



1. Neobsahuje nedostupné stavy.

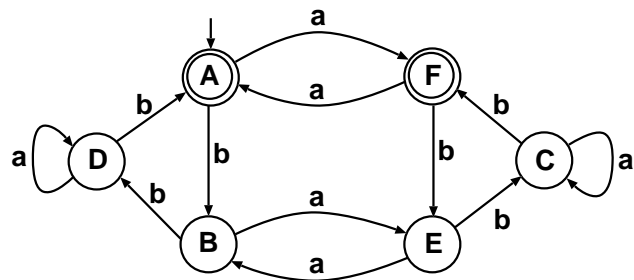
3. $\overset{0}{\equiv} = \{\{A, F\}, \{B, C, D, E\}\}$

5.1. $\overset{1}{\equiv} = \{\{A, F\}, \{B, E\}, \{C, D\}\}$

$\overset{0}{\equiv}$	δ	a	b
$I:$	A	F_I	B_{II}
	F	A_I	E_{II}
$II:$	B	E_{II}	D_{II}
	C	C_{II}	F_I
	D	D_{II}	A_I
	E	B_{II}	C_{II}

Pokračuje na druhé straně...

Pro zopakování automat z předchozího slajdu, v jehož minimalizaci níže pokračujeme:

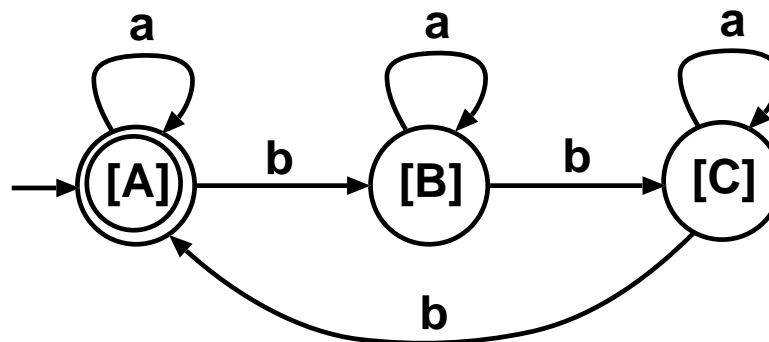


$$5.2. \quad \overset{2}{\equiv} = \{\{A, F\}, \{B, E\}, \{C, D\}\} = \overset{1}{\equiv} = \equiv$$

$\overset{1}{\equiv}$	δ	a	b
$I:$	A	F_I	B_{II}
	F	A_I	E_{II}
$II:$	B	E_{II}	D_{III}
	E	B_{II}	C_{III}
$III:$	C	C_{III}	F_I
	D	D_{III}	A_I

8. $Q' = \{[A], [B], [C]\}$, kde $[A] = \{A, F\}$, $[B] = \{B, E\}$, $[C] = \{C, D\}$

Výsledný automat:



Strukturální vlastnosti regulárních jazyků

Konečné jazyky

Věta 2.2 Každý konečný jazyk je regulární.

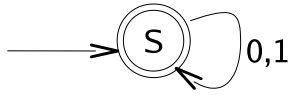
Důkaz. Necht' $L = \{w_1, w_2, \dots, w_n\}$, $w_i \in \Sigma^*$.

Pak $L = L(G)$, kde $G = (\{S\}, \Sigma, \{S \rightarrow w_1, S \rightarrow w_2, \dots, S \rightarrow w_n\}, S)$. G je zřejmě gramatika typu 3.

□

Opak věty 3.1 zjevně **neplatí**:

Příklad 2.2 Sestrojte gramatiku typu 3 generující jazyk $\{0, 1\}^*$.

Řešení:  $\Rightarrow G = (\{S\}, \{0, 1\}, \{S \rightarrow \varepsilon, S \rightarrow 0S, S \rightarrow 1S\}, S)$

Pumping lemma

Věta 2.3 Necht' L je regulární jazyk. Pak existuje celočíselná konstanta $p > 0$ taková, že platí:

$$\begin{aligned}w \in L \wedge |w| \geq p &\Rightarrow w = xyz \wedge \\ &y \neq \varepsilon \wedge |xy| \leq p \wedge \\ &xy^iz \in L \text{ pro } i \geq 0.\end{aligned}$$

❖ Ekvivalentní formulace Pumping lemmatu (použití explicitní alternace kvantifikátorů) :

$$\begin{aligned}L \in \mathcal{L}_3 &\Rightarrow \exists p > 0 : \\ &\forall w \in \Sigma^* : w \in L \wedge |w| \geq p \Rightarrow \\ &(\exists x, y, z \in \Sigma^* : w = xyz \wedge y \neq \varepsilon \wedge |xy| \leq p \wedge \forall i \geq 0 : xy^iz \in L)\end{aligned}$$

❖ **Poznámka:** Neformálně řečeno Pumping lemma tvrdí, že v každé dostatečně dlouhé větě každého regulárního jazyka jsme schopni poblíž jejího začátku najít poměrně krátkou sekvenci, kterou je možné vypustit, resp. zopakovat libovolný počet krát, přičemž zůstáváme stále v rámci daného jazyka.

Důkaz. Pumping lemmatu

Nechť $L = L(M)$, $M = (Q, \Sigma, \delta, q_0, F)$ je konečný automat, kde $|Q| = n > 0$. Položme $p = n$. Je-li $w \in L$ a $|w| \geq n$, pak M přijme větu w „průchodem“ alespoň $n + 1$ konfiguracemi a tudíž alespoň dvě z nich obsahují stejný stav, tedy:

$$(q_0, w) = (q_0, xyz) \overset{*}{\vdash} (r, yz) \overset{k}{\vdash} (r, z) \overset{*}{\vdash} (q_F, \varepsilon), \quad q_F \in F$$

pro nějaký stav $r \in Q$ a k takové, že $0 < k \leq n$. Dále je zřejmé, že k „zopakování“ stavu r dojde nejpozději po přečtení prvních n znaků vstupního řetězce a tudíž $|xy| \leq p$.

Pak ale existuje posloupnost konfigurací:

$$\begin{aligned} (q_0, xy^i z) & \overset{*}{\vdash} (r, y^i z) \\ & \overset{+}{\vdash} (r, y^{i-1} z) \\ & \vdots \\ & \overset{+}{\vdash} (r, z) \\ & \overset{*}{\vdash} (q_F, \varepsilon) \end{aligned}$$

z které plyne $xy^i z \in L(M)$, a to nejen pro $i > 0$, ale i pro případ $i = 0$:

$$(q_0, xz) \overset{*}{\vdash} (r, z) \overset{*}{\vdash} (q_F, \varepsilon), \quad q_F \in F$$

□

Význam Pumping lemmatu

❖ Jak můžeme dokázat, že daný problém je/není řešitelný pomocí uvažovaných výpočetních prostředků (např. jestli pro daný jazyk existuje KA)?

- ukázat existenci řešení je jednoduché: poskytneme řešení (např. KA)
- ukázat neexistenci je principiálně náročnější: nemůžeme vyzkoušet všechny možná řešení (všech KA je nekonečně mnoho)

❖ Pumping lemma nám dovoluje dokazovat neexistenci řešení (tj. neexistenci KA pro daný jazyk)

❖ V rámci TIN si ukážeme i další techniky (diagonalizace, redukce), které lze použít i pro jiné výpočetní třídy

Použití Pumping lemmatu

$(L \in \mathcal{L}_3 \Rightarrow A) \Leftrightarrow (\neg A \Rightarrow L \notin \mathcal{L}_3)$ Obměna implikace

$A \equiv \exists p > 0 :$
 $\forall w \in \Sigma^* : w \in L \wedge |w| \geq p \Rightarrow$
 $(\exists x, y, z \in \Sigma^* : w = xyz \wedge y \neq \epsilon \wedge |xy| \leq p \wedge \forall i \geq 0 : xy^i z \in L)$

$\neg A \equiv \forall p > 0 :$
 $\exists w \in \Sigma^* : w \in L \wedge |w| \geq p \wedge$
 $(\forall x, y, z \in \Sigma^* : w = xyz \wedge y \neq \epsilon \wedge |xy| \leq p \Rightarrow \exists i \geq 0 : xy^i z \notin L)$

❖ K důkazu, že jazyk L není regulární stačí dokázat tvrzení $\neg A$.

Příklad 2.3 Dokažte, že jazyk $L = \{0^n 1^n \mid n \geq 1\}$ není regulární.

Důkaz:

❖ Pro libovolné $p > 0$ zvolíme slovo $w = 0^p 1^p$ ($w \in L \wedge |w| \geq p$).

❖ Dále uvažme všechny rozdělení $w = xyz$, kde $y \neq \varepsilon \wedge |xy| \leq p$. Je zřejmé, že $y \in \{0\}^+$.

$$\underbrace{000 \dots 0}_{y} 111 \dots 1$$

❖ Pak ale pro libovolné $y \in \{0\}^+$ (libovolné rozdělení), $\exists i \geq 0$, pro které $xy^i z \notin L$ — nesouhlasí počet 0 a 1 (zde to platí pro všechna $i \neq 1$).

❖ Ukázali jsme, že pro L platí tvrzení $\neg A$ (viz. předchozí slajd) a tudíž $L \notin \mathcal{L}_3$.

□

Příklad 2.4 Dokažte, že jazyk $L = \{a^q \mid q \text{ je prvočíslo}\}$ není regulární.

Důkaz:

❖ Pro libovolné $p > 0$ zvolíme slovo $w = a^r$, kde r prvočíslo větší než p .

❖ Dále uvažme všechny rozdělení $w = xyz$, kde $y \neq \varepsilon \wedge |xy| \leq p$. Je zřejmé, že $y = a^k$, kde $0 < k \leq p$.

❖ Pak ale pro libovolné k (libovolné rozdělení), zvolme $i = r + 1$. Dostáváme, že $|xy^i z| = |xy^{r+1} z| = |xyz| + |y^r| = r + r \cdot k = r \cdot (k + 1)$, což však není prvočíslo (pro žádné k), a tedy $xy^{r+1} z \notin L$

❖ Ukázali jsme, že pro L platí tvrzení A a tudíž $L \notin \mathcal{L}_3$.

□

Myhill-Nerodova věta

Motivace

❖ Myhill-Nerodova věta

- charakterizuje některé zásadní vztahy mezi konečnými automaty nad abecedou Σ a jistými ekvivalenčními relacemi nad řetězci ze Σ^* ,
- popisuje některé z **nutných a postačujících podmínek pro to, aby daný jazyk byl jazykem regulárním** (používá se často k důkazu neregularity jazyka),
- poskytuje formální bázi pro elegantní důkaz **existence unikátního** (až na isomorfismus) **minimálního DKA** k danému regulárnímu jazyku.

Pravá kongruence a prefixová ekvivalence

❖ Zopakování: **ekvivalence** \sim je binární relace, která je *reflexivní, symetrická a tranzitivní*. **Index ekvivalence** \sim je počet tříd rozkladu Σ^* / \sim . Je-li těchto tříd nekonečně mnoho, definujeme index jako ∞ .

Definice 2.4 Necht' Σ je abeceda a \sim je ekvivalence na Σ^* . Ekvivalence \sim je **pravou kongruencí** (je zprava invariantní), pokud pro každé $u, v, w \in \Sigma^*$ platí

$$u \sim v \implies uw \sim vw$$

Věta 2.4 Ekvivalence \sim na Σ^* je pravá kongruence právě tehdy, když pro každé $u, v \in \Sigma^*$, $a \in \Sigma$ platí $u \sim v \implies ua \sim va$.

Důkaz. „ \implies “ je triviální, „ \impliedby “ lze snadno ukázat indukcí nad délkou w . □

Definice 2.5 Necht' L je libovolný (ne nutně regulární) jazyk nad abecedou Σ . Na množině Σ^* definujeme relaci \sim_L zvanou **prefixová ekvivalence** pro L takto:

$$u \sim_L v \stackrel{def}{\iff} \forall w \in \Sigma^* : uw \in L \iff vw \in L$$

Myhill-Nerodova věta

Věta 2.5 Necht' L je jazyk nad Σ . Pak následující tvrzení jsou ekvivalentní:

1. L je jazyk přijímaný deterministickým konečným automatem.
2. L je sjednocením některých tříd rozkladu určeného pravou kongruencí na Σ^* s konečným indexem.
3. Relace \sim_L má konečný index.

Důkaz. Dokážeme následující implikace:

- **1 \Rightarrow 2**
- **2 \Rightarrow 3**
- **3 \Rightarrow 1**

Z definice ekvivalence ($a \Leftrightarrow b \stackrel{def}{\iff} a \Rightarrow b \wedge b \Rightarrow a$) a ze základní tautologie výrokové logiky ($a \Rightarrow b \wedge b \Rightarrow c \Rightarrow a \Rightarrow c$) plyne tvrzení věty.

□

Důkaz implikace 1 \Rightarrow 2

❖ Je-li L přijímán DKA, pak L je sjednocením některých tříd rozkladu určeného pravou kongruencí na Σ^* s konečným indexem.

❖ Pro DKA $M = (Q, \Sigma, \delta, q_0, F)$ zavedme zobecněnou přechodovou funkci

$\hat{\delta} : Q \times \Sigma^* \rightarrow Q$ tak, že $\forall q_1, q_2 \in Q, w \in \Sigma^* : \hat{\delta}(q_1, w) = q_2 \Leftrightarrow (q_1, w) \stackrel{*}{\vdash}_M (q_2, \varepsilon)$.

Důkaz. Pro daný L přijímaný konečným automatem M zkonstruujeme \sim s potřebnými vlastnostmi:

- Necht' $M = (Q, \Sigma, \delta, q_0, F)$ a δ je totální.
- Zvolíme \sim jako binární relaci na Σ^* takovou, že $u \sim v \iff \hat{\delta}(q_0, u) = \hat{\delta}(q_0, v)$.
- Ukážeme, že \sim má potřebné vlastnosti:
 - \sim je *ekvivalence*: je reflexivní, symetrická a tranzitivní.
 - \sim má *konečný index*: třídy rozkladu odpovídají stavům automatu.
 - \sim je *pravá kongruence*: Necht' $u \sim v$ a $a \in \Sigma$. Pak $\hat{\delta}(q_0, ua) = \delta(\hat{\delta}(q_0, u), a) = \delta(\hat{\delta}(q_0, v), a) = \hat{\delta}(q_0, va)$ a tedy $ua \sim va$.
 - L je sjednocením některých tříd $\Sigma^* \setminus \sim$: těch, které odpovídají F .

□

Důkaz implikace 2 \Rightarrow 3

❖ Existuje-li relace \sim splňující podmínku 2, pak \sim_L má konečný index.

Důkaz.

- Pro všechny $u, v \in \Sigma^*$ takové, že $u \sim v$, platí $u \sim_L v$:
 - Necht' $u \sim v$. Ukážeme, že také $u \sim_L v$, tj. $\forall w \in \Sigma^* : uw \in L \Leftrightarrow vw \in L$.
 - Víme, že $uw \sim vw$ a protože L je sjednocením některých tříd rozkladu $\Sigma^* \setminus \sim$, platí též $uw \in L \Leftrightarrow vw \in L$.
- Víme tedy, že $\sim \subseteq \sim_L$ (tj. \sim_L je největší pravá kongruence s danými vlastnostmi).
- Každá třída \sim je obsažena v nějaké třídě \sim_L .
- Index \sim_L nemůže být větší než index \sim .
- \sim má konečný index a tedy i \sim_L má konečný index.

□

Důkaz implikace 3 \Rightarrow 1

❖ Má-li \sim_L konečný index, pak L je přijímán nějakým konečným automatem.

Důkaz. Zkonstruuujeme $M = (Q, \Sigma, \delta, q_0, F)$ přijímající L :

- $Q = \Sigma^* / \sim_L$ (stavy jsou třídy rozkladu Σ^* relací \sim_L),
- $\forall u \in \Sigma^*, a \in \Sigma : \delta([u], a) = [ua]$,
- $q_0 = [\varepsilon]$,
- $F = \{[x] \mid x \in L\}$.

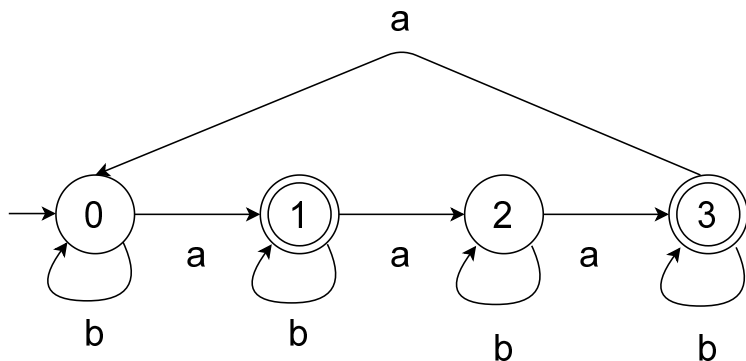
Uvedená konstrukce je korektní, tj. $L = L(M)$:

- Indukcí nad délkou slova v ukážeme, že $\forall v \in \Sigma^* : \hat{\delta}([\varepsilon], v) = [v]$.
- $v \in L \iff [v] \in F \iff \hat{\delta}([\varepsilon], v) \in F$.

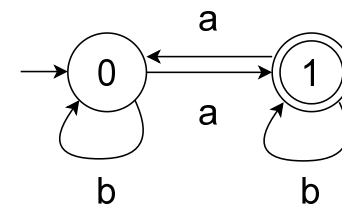
□

Příklad: Interpretace M.-N. věty

❖ Uvažme jazyk $L = \{w \in \{a, b\}^* \mid \#_a(w) \bmod 2 = 1\}$.



DKA A_1



DKA A_2

Pravá kongruence \sim_1 odpovídající DKA A_1 :

$$u \sim_1 v \iff \#_a(u) \equiv \#_a(v) \pmod{4}$$

$$\{a, b\}^* \setminus \sim_1 = \{[0]_4, [1]_4, [2]_4, [3]_4\}$$

$$\text{kde } [i]_4 = \{w \in \{a, b\}^* \mid \#_a(w) \bmod 4 = i\}$$

$$L = [1]_4 \cup [3]_4$$

\sim_2 odpovídající DKA A_2 :

$$u \sim_2 v \iff \#_a(u) \equiv \#_a(v) \pmod{2}$$

$$\{a, b\}^* \setminus \sim_2 = \{[0]_2, [1]_2\}$$

$$[i]_2 = \{w \in \{a, b\}^* \mid \#_a(w) \bmod 2 = i\}$$

$$L = [1]_2$$

$\sim_1 \subseteq \sim_2 = \sim_L$ (A_2 je minimální automat pro L)

Důkaz neregularity pomocí M.-N. věty

Příklad 2.5 Dokažte, že jazyk $L = \{a^n b^n \mid n \geq 0\}$ není regulární.

Důkaz.

- Žádné řetězce $\varepsilon, a, a^2, a^3, \dots$ nejsou \sim_L -ekvivalentní, protože $a^i b^i \in L$, ale $a^j b^i \notin L$ pro $i \neq j$.
- \sim_L má tedy nekonečně mnoho tříd (neboli nekonečný index).
- Dle Myhill-Nerodovy věty tudíž nemůže být L přijímán žádným konečným automatem.

□

Důkaz regularity pomocí M.-N. věty

Příklad 2.6 Dokažte, že jazyk $L = \{w \in \{a, b\}^* \mid 2010 \leq \#_a(w) \leq 2020\}$ je regulární.

Důkaz.

- Uvažme relaci pravé kongruence \sim definovanou následovně:

$$u \sim v \Leftrightarrow (\#_a(u) = \#_a(v)) \vee (\#_a(u) > 2020 \wedge \#_a(v) > 2020)$$

- \sim je *ekvivalence*: je reflexivní, symetrická a tranzitivní.
- \sim je *pravá kongruence*: Necht' $u \sim v$, pak $ua \sim va$, jelikož $\#_a(ua) = \#_a(va) = \#_a(u) + 1$ nebo $\#_a(ua) > 2020 \wedge \#_a(va) > 2020$
Rovněž $ub \sim vb$, jelikož $\#_a(ub) = \#_a(u) \wedge \#_a(v) = \#_a(vb)$.
- \sim má konečný index (rozklad Σ^* má 2022 tříd).
- L je sjednocením tříd rozkladu $[x_i]$ pro $2010 \leq i \leq 2020$, kde

$$[x_i] = \{w \in \{a, b\}^* \mid \#_a(w) = i\}$$

- Dle Myhill-Nerodovy věty je L přijímán konečným automatem.

□

M.-N. věta a minimalita DKA

Věta 2.6 (2. varianta Myhill-Nerodovy věty) Počet stavů libovolného minimálního DKA přijímajícího L je roven indexu \sim_L . (Takový DKA existuje právě tehdy, když je index \sim_L konečný.)

Důkaz.

- Každý DKA (můžeme uvažovat DKA bez nedosažitelných stavů) určuje jistou pravou kongruenci s konečným indexem a naopak.
- Je-li L regulární, je \sim_L největší pravou kongruencí s konečným indexem takovou, že L je sjednocením některých tříd příslušného rozkladu.
- Konečný automat, který odpovídá \sim_L (viz důkaz $3 \Rightarrow 1$ Myhill-Nerodovy věty), je tedy minimální konečný automat přijímající L .

□

Uzávěrové vlastnosti regulárních jazyků

Uzavěrové vlastnosti regulárních jazyků

Věta 2.7 Třída regulárních jazyků **je uzavřena** (mimo jiné) vzhledem k operacím:

\cup (sjednocení),

\cdot (konkatenace) a

$*$ (iterace).

\cap (průnik)

$co-$ (doplněk/komplement)

Důkaz. Uzavřenost na operace \cup , \cdot a $*$ plyne z definice regulárních množin a ekvivalence regulárních množin a regulárních jazyků.

Důkaz pokračuje dále.

Důkaz.

1. Dokážeme uzavřenost vzhledem ke komplementu nad abecedou Σ . K jazyku L sestrojíme *úplně definovaný deterministický* KA M .

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

takový, že $L = L(M)$. Pak KA M'

$$M' = (Q, \Sigma, \delta, q_0, Q \setminus F)$$

zřejmě přijímá jazyk $co-L = \Sigma^* \setminus L$ (tj. komplement jazyka L).

2. Uzavřenost vzhledem k průniku plyne z de Morganových zákonů:

$$L_1 \cap L_2 = co-(co-(L_1 \cap L_2)) = co-(co-L_1 \cup co-L_2)$$

a tedy $L_1, L_2 \in \mathcal{L}_3 \Rightarrow L_1 \cap L_2 \in \mathcal{L}_3$.

□

❖ Alternativní důkazy uzávěrových vlastností (konstrukce příslušných gramatik a automatů) ukážeme na cvičení

Příklady na uzávěrové vlastnosti

Rozhodněte a dokažte, zda platí následující tvrzení:

1. $\forall L_1, L_2 \subseteq \Sigma^* : L_1 \in \mathcal{L}_3 \wedge L_2 \in \mathcal{L}_3 \iff (L_1 \cup L_2) \in \mathcal{L}_3$
2. $\forall L_1, L_2 \subseteq \Sigma^* : L_1 \in \mathcal{L}_3 \wedge L_2 \in \mathcal{L}_3 \Rightarrow (L_1 \diamond L_2) \in \mathcal{L}_3$, kde
 $L_1 \diamond L_2 = \{uv \mid u, v \in L_1 \cup L_2\}$

Příklady na uzávěrové vlastnosti

Rozhodněte a dokažte, zda platí následující tvrzení:

1. $\forall L_1, L_2 \subseteq \Sigma^* : L_1 \in \mathcal{L}_3 \wedge L_2 \in \mathcal{L}_3 \iff (L_1 \cup L_2) \in \mathcal{L}_3$
2. $\forall L_1, L_2 \subseteq \Sigma^* : L_1 \in \mathcal{L}_3 \wedge L_2 \in \mathcal{L}_3 \Rightarrow (L_1 \diamond L_2) \in \mathcal{L}_3$, kde
 $L_1 \diamond L_2 = \{uv \mid u, v \in L_1 \cup L_2\}$

Řešení 1: Tvrzení neplatí. Ukážeme, že neplatí implikace \Leftarrow . Zvolme $L_1 = \{a^n b^n \mid n > 0\}$ a $L_2 = \Sigma^*$. Pak $L_1 \cup L_2 = \Sigma^* \in \mathcal{L}_3$, ale $L_1 \notin \mathcal{L}_3$. Opačná implikace platí přímo z uzávěrových vlastností regulárních jazyků.

Příklady na uzávěrové vlastnosti

Rozhodněte a dokažte, zda platí následující tvrzení:

1. $\forall L_1, L_2 \subseteq \Sigma^* : L_1 \in \mathcal{L}_3 \wedge L_2 \in \mathcal{L}_3 \iff (L_1 \cup L_2) \in \mathcal{L}_3$
2. $\forall L_1, L_2 \subseteq \Sigma^* : L_1 \in \mathcal{L}_3 \wedge L_2 \in \mathcal{L}_3 \Rightarrow (L_1 \diamond L_2) \in \mathcal{L}_3$, kde
 $L_1 \diamond L_2 = \{uv \mid u, v \in L_1 \cup L_2\}$

Řešení 1: Tvrzení neplatí. Ukážeme, že neplatí implikace \Leftarrow . Zvolme $L_1 = \{a^n b^n \mid n > 0\}$ a $L_2 = \Sigma^*$. Pak $L_1 \cup L_2 = \Sigma^* \in \mathcal{L}_3$, ale $L_1 \notin \mathcal{L}_3$. Opačná implikace platí přímo z uzávěrových vlastností regulárních jazyků.

Řešení 2: Tvrzení platí. Uvědomme si, že $L_1 \diamond L_2 = (L_1 \cup L_2) \cdot (L_1 \cup L_2)$. Z uzavřenosti regulárních jazyků vzhledem k operacím \cup a \cdot tudíž dostáváme požadované tvrzení.

Rozhodnutelné problémy regulárních jazyků

Rozhodnutelné problémy v \mathcal{L}_3

Základní problémy:

- problém **neprázdnoti**: $L \neq \emptyset$?
- problém **universality**: $L = \Sigma^*$?
- problém **náležitosti**: $w \in L$?
- problém **ekvivalence**: $L(G_1) = L(G_2)$?

Věta 2.8 Ve třídě \mathcal{L}_3 je rozhodnutelný problémy **neprázdnoti** a **universality** jazyka i problém **náležitosti** řetězce (do jazyka).

Důkaz.

K jazyku $L \in \mathcal{L}_3$ sestrojíme úplně definovaný DKA M , $L = L(M)$:

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

neprázdnot: $L(M) \neq \emptyset \iff \exists q \in Q : (q \in F \wedge q \text{ je dostupný z } q_0)$

universalita: $L(M) = \Sigma^* \iff \forall q \in Q : (q \in F \vee q \text{ není dostupný z } q_0)$

náležitost: $w \in L \iff (q_0, w) \stackrel{*}{\vdash} (q, \varepsilon) \wedge q \in F$

□

Věta 2.9 Nechť $L_1 = L(G_1)$ a $L_2 = L(G_2)$ jsou dva jazyky generované regulárními gramatikami G_1 a G_2 . Pak je rozhodnutelný problém **ekvivalence**, tj. $L(G_1) = L(G_2)$.

Důkaz.

Nechť $M_1 = (Q_1, \Sigma_1, \delta_1, q_0^1, F_1)$, resp. $M_2 = (Q_2, \Sigma_2, \delta_2, q_0^2, F_2)$ jsou KA přijímající jazyky L_1 , resp. L_2 takové, že $Q_1 \cap Q_2 = \emptyset$.

Vytvoříme konečný automat M takto:

$$M = (Q_1 \cup Q_2, \Sigma_1 \cup \Sigma_2, \delta_1 \cup \delta_2, q_0^1, F_1 \cup F_2)$$

a vypočítáme relaci \equiv nerozlišitelnosti stavů z $Q_1 \cup Q_2$ pro automat M .

Pak

$$L(G_1) = L(G_2) \iff q_0^1 \equiv q_0^2$$

□