

63. Konečné automaty (jazyky přijímané KA, varianty KA, minimalizace KA, Myhill-Nerodova věta).

Konečný automat

Nedeterministický konečný automat (NKA) je pětice $A = (Q, \Sigma, \delta, s_0, F)$, kde:

- Q je konečná, neprázdná množina **stavů**
- Σ je **abeceda** (konečná, neprázdná množina symbolů)
- $\delta : Q \times \Sigma \rightarrow 2^Q$ je **přechodová funkce**
- $s_0 \in Q$ je **počáteční stav**
- $F \subseteq Q$ je množina **koncových stavů**

Zápisem $Q \times \Sigma \rightarrow 2^Q$ rozumíme zobrazení z množiny $Q \times \Sigma$ do množiny 2^Q . Množina $Q \times \Sigma$ obsahuje dvojice (q, a) , kde q je nějaký stav a a je symbol abecedy. Pokud zobrazení δ dostane na vstup takovou dvojici (q, a) , pro kterou je δ definováno, zobrazí jej na prvek množiny 2^Q , tedy na nějakou podmnožinu množiny Q . Jinak řečeno, zobrazení δ nám říká, do kterých stavů je možné přejít ze stavu q při přečtení symbolu a .

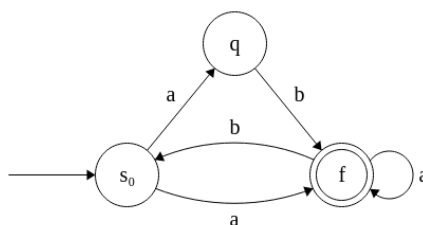
Pozn.: Přechodovou funkci je možné chápat i tak, že je vždy totální (je definovaná pro každý možný stav a symbol abecedy), ovšem v některých případech může vstup zobrazovat na prázdnou množinu $\emptyset$, což znamená, že z daného stavu s daným symbolem nevede přechod do žádného stavu.

Pozn.: Mnohdy se definuje i celá (neprázdná) množina počátečních stavů, přičemž při běhu automatu je počáteční stav nedeterministicky zvolen.

Konečný automat znázorňujeme jako orientovaný graf s ohodnocením hran symboly z množiny Σ . Zdvojený obvod uzlu označuje koncový stav, šipka vedoucí do stavu odnikud označuje počáteční stav.

V níže uvedeném případě $A = (\{s_0, q, f\}, \{a, b\}, \{((s_0, a), \{q, f\}), ((q, b), \{f\}), ((f, a), \{f\}), ((f, b), \{s_0\})\}, s_0, \{f\})$, neboli

- $Q = \{s_0, q, f\}$
- $\Sigma = \{a, b\}$
- $\delta(s_0, a) = \{q, f\}, \delta(q, b) = \{f\}, \delta(f, a) = \{f\}, \delta(f, b) = \{s_0\}$
- $F = \{f\}$



At' $A = (Q, \Sigma, \delta, s_0, F)$ je konečný automat. **Konfigurace** konečného automatu je dvojice $(q, w) \in Q \times \Sigma^*$. Konfigurací rozumíme aktuální stav výpočtu konečného automatu, kde q označuje stav, v němž se momentálně v běhu automatu nacházíme, a w je dosud nepřečtená část vstupu.

At' $A = (Q, \Sigma, \delta, s_0, F)$ je konečný automat. **Přechodová relace** $\vdash \subseteq (Q \times \Sigma^*) \times (Q \times \Sigma^*)$ ($\vdash$ se nazývá turniket) je binární relací na množině konfigurací definovaná následovně:

$$(q, aw) \vdash (r, w) \Leftrightarrow q, r \in Q \wedge a \in \Sigma \wedge w \in \Sigma^* \wedge r \in \delta(q, a),$$

tedy ze stavu q přejdeme do stavu r , přičemž ze vstupu přečteme symbol a , což ovšem můžeme udělat jen za předpokladu, že takový přechod automat umožňuje. Ve výše uvedeném automatu například platí $(s_0, abbaa) \vdash (q, bbaa)$, neboť pokud se nacházíme ve stavu s_0 a přečteme symbol a , lze přejít do stavu q .

K přechodové relaci lze vytvořit její **reflexivní a tranzitivní uzávěr** $\vdash^*$, což je $\subseteq$ -minimální nadmnožina relace $\vdash$ taková, že je navíc reflexivní a tranzitivní. Jinak řečeno, do relace $\vdash$ přidáme minimální nutný počet dvojic (minimální vzhledem k relaci $\subseteq$) tak, aby po této změně byla reflexivní i tranzitivní.

At' $A = (Q, \Sigma, \delta, s_0, F)$ je konečný automat. **Jazyk přijímaný automatem** je množina

$$L(A) = \{w \in \Sigma^* \mid \exists f \in F : (s_0, w) \vdash^* (f, \varepsilon)\},$$

tedy slovo je automatem přijato tehdy, pokud je jeho čtení započato v počátečním stavu a dokončeno v některém z koncových stavů automatu. Ilustrujme přijetí slova *abbaa* výše uvedeným automatem:

$$(s_0, abbaa) \vdash (q, bbaa) \vdash (f, baa) \vdash (s_0, aa) \vdash (f, a) \vdash (f, \varepsilon),$$

tedy

$$(s_0, abbaa) \vdash^* (f, \varepsilon).$$

At' Σ je abeceda. Množina všech jazyků nad abecedou Σ je nespočetná množina 2^{Σ^*} . Množina všech jazyků, které mohou být přijaty konečným automatem pracujícím s abecedou Σ , se nazývá množina **regulárních jazyků** $\mathcal{L}_3 \subset 2^{\Sigma^*}$.

Příklady regulárních jazyků nad abecedou $\Sigma = \{a, b\}$:

- $\emptyset, \Sigma^* = \{a, b\}^*$
- $\{\varepsilon\}, \{a\}, \{b\}, \{a, b\}$, případně **libovolný konečný jazyk**
- $\{a^n \mid n \geq 0\}$
- $\{w \in \{a, b\}^* \mid \#_a(w) \bmod 2 = \#_b(w) \bmod 3\}$
- $\{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w) \wedge |w| < 10\}$

Příklady jazyků nad abecedou $\Sigma = \{a, b\}$, které nejsou regulární:

- $\{a^n b^n \mid n \geq 0\}$
- $\{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w) \wedge |w| > 10\}$
- $\{ww \mid w \in \{a, b\}^*\}$
- $\{<S> \mid <S> \text{ je kód Postova systému nad kódovou abecedou } \{a, b\}, \text{ který nemá řešení}\}$

Varianty konečných automatů

Deterministický konečný automat (DKA) je pětice $A = (Q, \Sigma, \delta, s_0, F)$, kde

- Q je konečná, neprázdná množina **stavů**
- Σ je **abeceda** (konečná, neprázdná množina symbolů)
- $\delta : Q \times \Sigma \rightarrow 2^Q$ je **přechodová funkce**, kde $\forall q \in Q : \forall a \in \Sigma : |\delta(q, a)| \leq 1$
- $s_0 \in Q$ je **počáteční stav**
- $F \subseteq Q$ je množina **koncových stavů**

V případě deterministického konečného automatu je z daného stavu s daným symbolem možné přejít maximálně do jednoho dalšího stavu. Každý deterministický konečný automat je triviálně nedeterministický.

Pozn.: Přechodovou funkci zde pro jednoduchost chápeme jako totální (je definovaná pro každý možný stav a symbol abecedy), přičemž v některých případech zobrazuje do jednoprvkové množiny, jindy do prázdné množiny.

Rozšířený konečný automat (RKA) je pětice $A = (Q, \Sigma, \delta, s_0, F)$, kde

- Q je konečná, neprázdná množina **stavů**
- Σ je **abeceda** (konečná, neprázdná množina symbolů)
- $\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow 2^Q$ je **přechodová funkce**
- $s_0 \in Q$ je **počáteční stav**
- $F \subseteq Q$ je množina **koncových stavů**

V případě rozšířeného konečného je možné vytvářet takzvané **ϵ -přechody**, při jejichž použití není přečten žádný symbol ze vstupu (přechod potom značí pouhou změnu stavu v aktuální konfiguraci). Každý nedeterministický konečný automat je triviálně rozšířený.

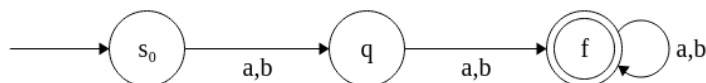
Úplně definovaný (úplný) deterministický konečný automat (ÚDKA) je pětice $A = (Q, \Sigma, \delta, s_0, F)$, kde A je deterministický automat s **totální** přechodovou funkcí, přičemž $\forall q \in Q : \forall a \in \Sigma : |\delta(q, a)| = 1$, tedy každá dvojice (q, a) , kde q je stav a a je symbol abecedy, je zobrazena na množinu obsahující právě jeden stav. Nemůže se tedy stát, že by v některém stavu nebylo možné s některým symbolem provést přechod.

Ať $A = (Q, \Sigma, \delta, s_0, F)$ je úplný deterministický konečný automat a $q, r \in Q$ jsou dva stavy.

Řekneme, že řetězec $w \in \Sigma^*$ **rozlišuje** stavy $q, r \Leftrightarrow (q, w) \vdash^* (f_1, \epsilon) \wedge (r, w) \vdash^* (f_2, \epsilon) \wedge (f_1 \in F \oplus f_2 \in F)$

$f_2 \in F$ ($\oplus$ značí exkluzivní OR, tedy XOR), tedy po dočtení řetězce w ze stavů q, r se právě v jednom případě ocitneme v koncovém stavu.

V níže uvedeném automatu stavy s_0, q rozlišuje řetězec b , neboť $(s_0, b) \vdash^* (q, \varepsilon)$, kde q je nekonečný stav, a $(q, b) \vdash^* (f, \varepsilon)$, kde f je koncový stav.

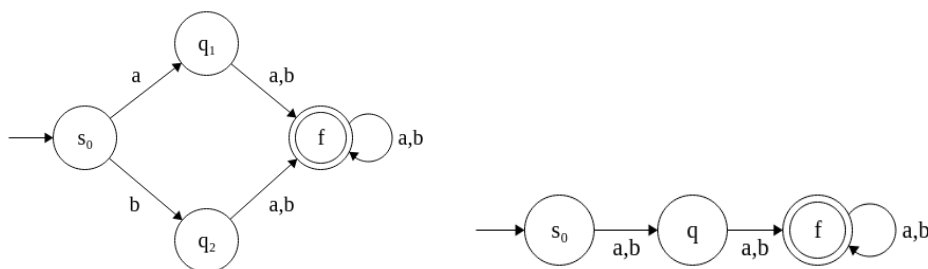


Ať $A = (Q, \Sigma, \delta, s_0, F)$ je úplný deterministický konečný automat a $q, r \in Q$ jsou dva stavy. Stavy q, r jsou **k-nerozlišitelné** (psáno $q \equiv_k r$), pokud $\forall w \in \Sigma^* : |w| \leq k \Rightarrow w$ nerozlišuje stavy q, r . Neexistuje tedy žádný řetězec délky k nebo kratší, který by rozlišoval stavy q, r . Stavy q, r jsou **nerozlišitelné** tehdy, pokud platí, že $\forall k \in \mathbb{N} : q \equiv_k r$, tedy stavy jsou k -nerozlišitelné pro libovolné přirozené k . Jinak řečeno, stavy q, r jsou nerozlišitelné tehdy, pokud by se shodovaly jazyky automatu A v případě, že by stav q byl počáteční a v případě, že by stav r byl počáteční.

Ať $A = (Q, \Sigma, \delta, s_0, F)$ je konečný automat. Stav $q \in Q$ je **nedosažitelný (nedostupný)**, pokud $\forall w \in \Sigma^* : \neg((s_0, w) \vdash^* (q, \varepsilon))$, tedy do stavu q není možné dospět žádnou konečnou sekvencí přechodů z počátečního stavu.

Ať $A = (Q, \Sigma, \delta, s_0, F)$ je úplný deterministický konečný automat. Automat A je **redukovaný (minimální)**, pokud žádný jeho stav není nedosažitelný a pro žádné dva různé stavy neplatí, že by byly nerozlišitelné. Ke každému regulárnímu jazyku existuje právě jeden redukovaný ÚDKA (až na pojmenování stavů).

Na následujícím obrázku vlevo je znázorněn automat A , který není redukovaný, neboť stavy q_1, q_2 jsou nerozlišitelné. Vpravo vidíme ekvivalentní automat (přijímající stejný jazyk), který je redukovaný



Transformace variant konečných automatů

Představili jsme si modely NKA, DKA, RKA, ÚDKA, redukovaný ÚDKA. Všechny tyto modely mají **stejnou vyjadřovací sílu**, tedy všechny přijímají právě **třidu regulárních jazyků** $\mathcal{L}_3$. Jednotlivé modely je možné na sebe vzájemně převádět. Představíme si několik převodů. Všechny ostatní vyplynou z tranzitivity.

Převod NKA na RKA. Každý nedeterministický konečný automat je triviálně rozšířený, je jeho speciálním případem. Nepoužívá v sobě epsilon přechody, ale plně vyhovuje definici rozšířeného konečného automatu. Není tedy třeba provádět žádný převod.

Převod DKA na NKA. Každý deterministický konečný automat je triviálně nedeterministický, je jeho speciálním případem. Nepoužívá v sobě nedeterminismus, ale plně vyhovuje definici nedeterministického konečného automatu. Není tedy třeba provádět žádný převod.

Převod ÚDKA na DKA. Každý úplný deterministický konečný automat je triviálně deterministický, je jeho speciálním případem. Není tedy třeba provádět žádný převod.

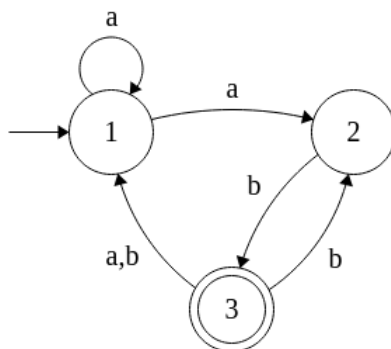
Převod redukovaného ÚDKA na ÚDKA. Každý redukovaný úplný deterministický konečný automat je triviálně úplný deterministický, je jeho speciálním případem. Není tedy třeba provádět žádný převod.

Převod NKA na DKA: Determinizace. Ať $A = (Q, \Sigma, \delta, s_0, F)$ je NKA. Jelikož z jednoho stavu může být možné při čtení symbolu přejít do několika různých stavů, lze změnit stavový prostor automatu tím způsobem, že tyto cílové stavy jako celek budou představovat jediný stav nového automatu, aby bylo dosaženo determinismu. Původní stavový prostor Q lze tedy změnit na jeho potenční množinu 2^Q .

Převod proběhne následovně. Z NKA $A = (Q, \Sigma, \delta, s_0, F)$ konstruujeme DKA $A' = (Q', \Sigma, \delta', s_0', F')$. Postupně budujeme stavový prostor Q' . Každý stav automatu A' bude nějakou podmnožinou množiny Q .

- $s_0' = \{s_0\}$
- pokud q' je již vytvořený stav automatu A' , pak definujeme $\forall a \in \Sigma : \delta'(q, a) = \cup \{r \in q \mid \delta(r, a) \in q'\}$
- $q \in F' \Leftrightarrow q \in Q' \wedge q \cap F \neq \emptyset$, tedy stav označíme jako koncový tehdy, pokud je jeho průnik s množinou koncových stavů původního automatu neprázdný

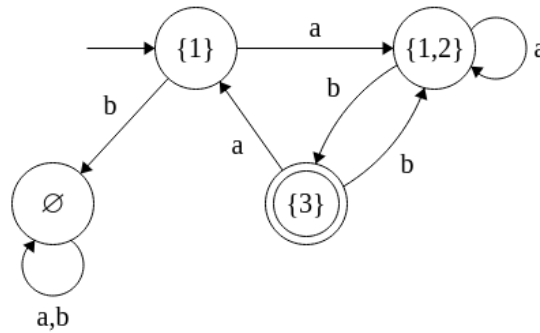
Mějme nedeterministický konečný automat uvedený na obrázku. Provedme jeho determinizaci.



Použijeme tabulkovou metodu, kdy postupně budujeme prostor dosažitelných stavů. Šipka doleva znázorňuje počáteční stav, šipka doprava koncový stav.

	a	b
--	---	---

$\rightarrow\{1\}$	$\{1, 2\}$	$\emptyset$
$\{1, 2\}$	$\{1, 2\}$	$\{3\}$
$\leftarrow\{3\}$	$\{1\}$	$\{1, 2\}$
$\emptyset$	$\emptyset$	$\emptyset$



Převod RKA na ÚDKA. Ať $A = (Q, \Sigma, \delta, s_0, F)$ je RKA, $q \in Q$. Definujeme zobrazení ε -uzávěr : $Q \rightarrow 2^Q$:

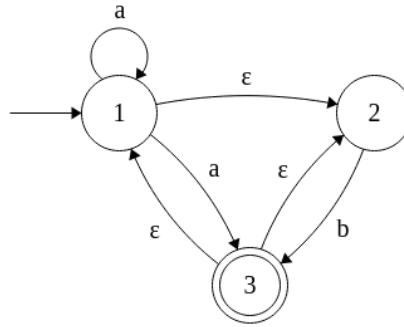
$$\varepsilon\text{-uzávěr}(q) = \{r \in Q \mid \exists w \in \Sigma^* : (q, w) \vdash^* (r, w)\}$$

ε -uzávěr pro stav q je množina stavů, které jsou z q dosažitelné bez přečtení symbolu. Do ε -uzávěru stavu q tedy patří samotný stav q a všechny stavy, které jsou z q dosažitelné pomocí ε -přechodů. Zobecníme ε -uzávěry i pro množiny jako sjednocení ε -uzávěrů pro jednotlivé prvky vstupní množiny.

Převod proběhne následovně. Z RKA $A = (Q, \Sigma, \delta, s_0, F)$ konstruujeme ÚDKA $A' = (Q', \Sigma, \delta', s_0', F')$. Postupně budujeme stavový prostor Q' . Každý stav automatu A' bude nějakou podmnožinou množiny Q .

- $s_0' = \varepsilon\text{-uzávěr}(s_0)$
- pokud q' je již vytvořený stav automatu A' , pak definujeme $\forall a \in \Sigma : \delta'(q, a) = \varepsilon\text{-uzávěr}(\delta(q, a))$, přičemž $\varepsilon\text{-uzávěr}(\delta(q, a))$ je nový stav, pokud dosud neexistuje. Každý stav zde odpovídá nějaké množině stavů původního automatu A
- $q \in F' \Leftrightarrow q \in Q' \wedge q \cap F \neq \emptyset$, tedy stav označíme jako koncový tehdy, pokud je jeho průnik s množinou koncových stavů původního automatu neprázdný

Mějme automat uvedený na obrázku.



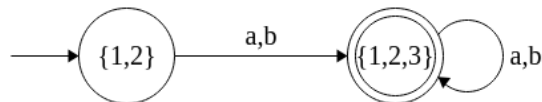
Vypočtěme ϵ -uzávěry stavů:

- $\epsilon\text{-uzávěr}(1) = \{1, 2\}$
- $\epsilon\text{-uzávěr}(2) = \{2\}$
- $\epsilon\text{-uzávěr}(3) = \{1, 2, 3\}$

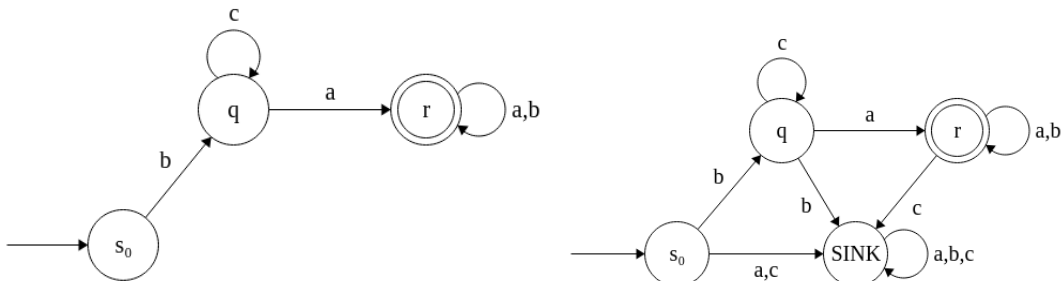
Jelikož stav 1 byl počáteční, bude v novém automatu počáteční stav $\{1, 2\}$.

- $\delta'(\{1, 2\}, a) = \epsilon\text{-uzávěr}(\{1, 3\}) = \{1, 2, 3\}$
- $\delta'(\{1, 2\}, b) = \epsilon\text{-uzávěr}(\{3\}) = \{1, 2, 3\}$
- $\delta'(\{1, 2, 3\}, a) = \epsilon\text{-uzávěr}(\{1, 2, 3\}) = \{1, 2, 3\}$
- $\delta'(\{1, 2, 3\}, b) = \epsilon\text{-uzávěr}(\{1, 2, 3\}) = \{1, 2, 3\}$

Nový stav $\{1, 2, 3\}$ je koncový, neboť stav 3 byl koncový v původním automatu.

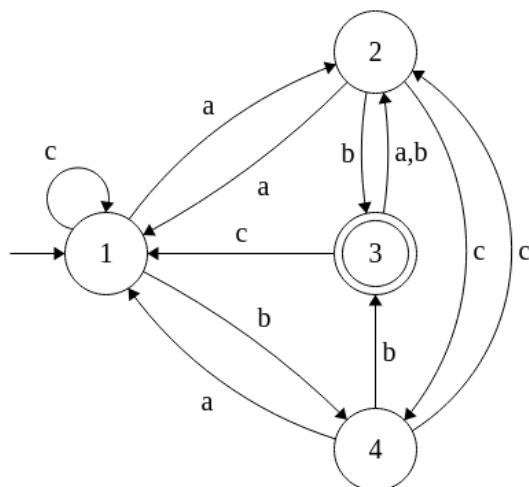


Převod DKA na ÚDKA. Při převodu deterministického konečného automatu A na ekvivalentní (přijímající stejný jazyk) úplně definovaný konečný automat A' stačí vytvořit nový nekonečný stav (takzvaný SINK stav, odtok). Pokud pro některý stav q a některý symbol a je v automatu A hodnota $\delta(q, a) = \emptyset$, pak v automatu A' zavedeme $\delta(q, a) = \{\text{SINK}\}$. Podobně $\delta(\text{SINK}, a) = \{\text{SINK}\}$ pro každý symbol a . Na obrázku vlevo je uveden deterministický konečný automat A s abecedou $\{a, b, c\}$ a k němu ekvivalentní úplný deterministický konečný automat A'.



Převod ÚDKA na redukovaný ÚDKA: Minimalizace. Ať $A = (Q, \Sigma, \delta, s_0, F)$ je ÚDKA. Po odstranění nedostupných stavů postupně počítáme relace k -nerozlišitelnosti pro zvyšující se k od 0 výše. Ekvivalenční třídy relace představují kandidáty na stavy nového automatu. Jakmile se relace v dalším kroku nezmění, byly nalezeny stavy redukovaného automatu.

Uvažujme o uvedeném automatu.



Relaci 0-nerozlišitelnosti lze vypočítat jednoduše, neboť dva stavy jsou 0-nerozlišitelné tehdy, pokud jsou oba koncové, nebo oba nekoncové. Vytvoříme ekvivalenční třídy $I = \{1, 2, 4\}$ a $II = \{3\}$ relace Ξ_0 . Následně pro všechny stavy a pro všechny symboly vypíšeme, do které ekvivalenční třídy bychom dospěli, pokud bychom z jednotlivých stavů provedli krok při čtení daného symbolu. To je uvedeno v tabulce níže.

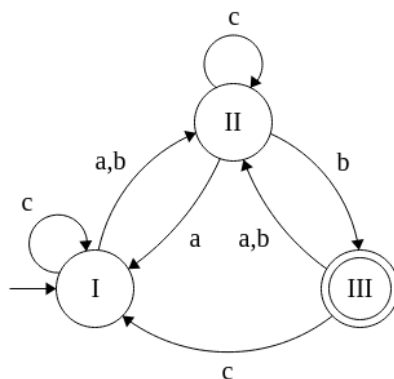
Ξ_0	stav	a	b	c
$\rightarrow I$	1	I	I	I
	2	I	II	I
	4	I	II	I
$\leftarrow II$	3	I	I	I

Vidíme, že řádky pro stavy 1 a 2 se liší. To znamená, že ačkoliv jsou tyto stavy 0-nerozlišitelné, nejsou 1-nerozlišitelné, neboť je rozlišuje řetězec b . Při výpočtu relace 1-nerozlišitelnosti tedy rozštěpíme existující třídy tak, aby se ve stejné třídě nacházely stavy, které byly 0-nerozlišitelné a současně měly v uvedené tabulce stejně ohodnocené řádky. Vznikne tak nová relace Ξ_1 s ekvivalenčními třídami $I = \{1\}$, $II = \{2, 4\}$, $III = \{3\}$.

Ξ_1	stav	a	b	c
$\rightarrow I$	1	II	II	I
II	2	I	III	II

	4	I	III	II
←III	3	II	II	I

Jelikož se v rámci každé ekvivalenční třídy shodují ohodnocení řádků, našli jsme stavy redukovaného automatu dané těmito ekvivalenčními třídami. Následující obrázek odpovídá redukovanému automatu. Lze snadno nahlédnout, že jsme sloučili stavy 2 a 4, které v původním automatu byly nerozlišitelné.

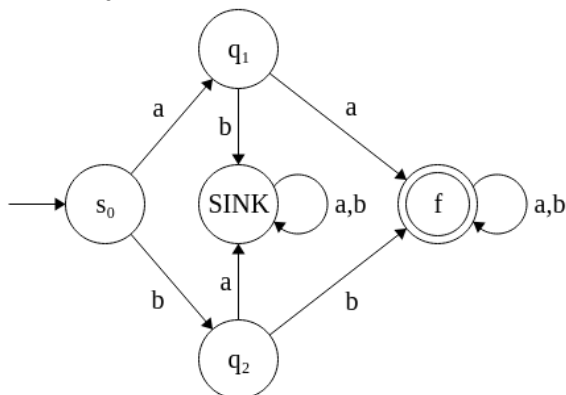


Myhill-Nerodova věta

At' $A = (Q, \Sigma, \delta, s_0, F)$ je ÚDKA a $q \in Q$ je stav. **Jazyk přístupových řetězců stavu q** je množina

$$L^{-1}(q) = \{w \in \Sigma^* \mid (s_0, w) \vdash^* (q, \varepsilon)\},$$

tedy množina všech řetězců, které by byly dočteny ve stavu q , pokud by byly čteny z počátečního stavu s_0 . Jinak řečeno, $L^{-1}(q)$ by byl jazyk automatu A , pokud by q byl jediný koncový stav. Uvažujme o následujícím automatu.



Jazyky přístupových řetězců jednotlivých stavů jsou následující:

- $L^{-1}(s_0) = \{\varepsilon\}$
- $L^{-1}(q_1) = \{a\}$
- $L^{-1}(q_2) = \{b\}$
- $L^{-1}(f) = \{aa, bb\} \cdot \{a, b\}^*$
- $L^{-1}(\text{SINK}) = \{ab, ba\} \cdot \{a, b\}^*$

Jelikož je automat úplný, leží každý řetězec z množiny Σ^* v některé z výše uvedených množin. Jelikož je automat deterministický, jsou tyto množiny po různých dvojicích disjunktní. To ale znamená, že systém množin

$$\{L^{-1}(s_0), L^{-1}(q_1), L^{-1}(q_2), L^{-1}(f), L^{-1}(\text{SINK})\} = \{\{\epsilon\}, \{a\}, \{b\}, \{aa, bb\} \cdot \{a, b\}^*, \{ab, ba\} \cdot \{a, b\}^*\}$$

je rozkladem množiny Σ^* . Připomeňme, že ke každému rozkladu množiny přísluší nějaké relace ekvivalence na této množině.

Ať $A = (Q, \Sigma, \delta, s_0, F)$ je ÚDKA. Relaci $\sim \subseteq \Sigma^* \times \Sigma^*$ nazveme **pravou kongruencí**, pokud současně

- $\sim$ je relace ekvivalence
- $\forall u, v, w \in \Sigma^* : u \sim v \Rightarrow uw \sim vw$
- $(\forall u, v \in \Sigma^* : \forall a \in \Sigma : u \sim v \Rightarrow ua \sim va)$, což je ekvivalentní s předchozím bodem)

Lze říci, že dva řetězce jsou v relaci pravé kongruence, pokud po jejich dočtení skončíme ve stejném stavu automatu. Pokud jsou dva řetězce v relaci $\sim$, pak budou v relaci i tehdy, pokud za ně připojíme libovolnou příponu. To odpovídá intuitivnímu chápání konečného automatu. Pokud dva řetězce dočteme ve stejném stavu, bude jakýkoliv další běh automatu probíhat v obou případech identicky, budeme pokračovat stejnou cestou neohledě na to, jakou další příponu přečteme. Relace pravé kongruence tedy vždy popisuje nějaký **úplný deterministický konečný automat**, pokud má příslušný rozklad konečný počet tříd.

Jelikož je relace $\sim$ ekvivalencí na množině Σ^* , je zřejmé, že existuje rozklad $\Sigma^* / \sim$, který je roven systému množin přístupových řetězců jednotlivých stavů automatu představenému výše.

Ať $A = (Q, \Sigma, \delta, s_0, F)$ je ÚDKA. Relaci $\sim_L \subseteq \Sigma^* \times \Sigma^*$ nazveme **prefixovou ekvivalencí**, pokud současně

- $\sim_L$ je relace ekvivalence
- $\forall u, v \in \Sigma^* : u \sim_L v \Leftrightarrow (\forall w \in \Sigma^* : uw \in L \Leftrightarrow vw \in L)$

Aby byly dva řetězce v relaci prefixové ekvivalence, musí být dočteny ve stejném stavu automatu, tedy prefixová ekvivalence je speciálním případem pravé kongruence. Definice si však vynucuje, aby žádné dva stavy tohoto automatu nebyly nerozlišitelné. Pokud by to tomu totiž došlo, platila by podmínka $(\forall w \in \Sigma^* : uw \in L \Leftrightarrow vw \in L)$, ovšem řetězce u, v by nebyly v relaci pravé kongruence. Prefixová ekvivalence je tedy speciálním případem pravé kongruence popisující **redukovaný automat** pro jazyk L , pokud má příslušný rozklad konečný počet tříd. To znamená, že relace prefixové ekvivalence je největší pravou kongruencí vzhledem k relaci $\subseteq$.

Myhill-Nerodova věta je tvrzení o regulárních jazycích, které dává do souvislosti konečné automaty, relace pravé kongruence a relace prefixové ekvivalence. Její znění je následující:

- **Následující tři tvrzení jsou ekvivalentní:**
 - Pro jazyk L existuje DKA, který ho přijímá
 - L je sjednocením některých tříd rozkladu Σ^* dle relace pravé kongruence $\sim$ s konečným indexem (konečným počtem tříd rozkladu $\Sigma^* / \sim$)

- Relace prefixové ekvivalence $\sim_L$ má konečný index (konečný počet tříd rozkladu $\Sigma^*/\sim_L$)

Myhill-Nerodovu větu je možné použít pro důkaz regularity, nebo neregularity jazyka L .

Chceme-li dokázat, že jazyk L je regulární, najdeme relaci pravé kongruence $\sim$ s konečným indexem takovou, že L je sjednocením některých tříd rozkladu $\Sigma^*/\sim$. Chceme-li dokázat, že jazyk L není regulární, ukážeme, že relace $\sim_L$ má nekonečný index, tedy automat, který by L popisoval, by musel mít nekonečný počet stavů, což odporuje definici konečného automatu.

Jelikož je prefixová ekvivalence největší pravou kongruencí, i všechny pravé kongruence $\sim$, kde L je sjednocením některých tříd rozkladu $\Sigma^*/\sim$, by měly nekonečný index. Z toho plyne, že neexistuje konečný automat, který L přijímá, tedy L není regulární.

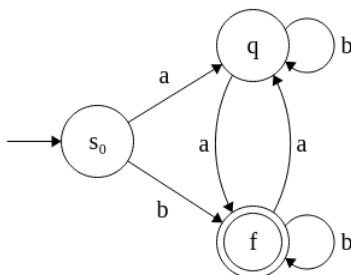
Ukázka důkazu neregularity pomocí Myhill-Nerodovy věty.

- Ukažme, že jazyk $L = \{a^n b^n \mid n \geq 0\}$ není regulární
 - Pro spor předpokládejme, že L je regulární
 - Z Myhill-Nerodovy věty plyne, že relace prefixové ekvivalence $\sim_L$ má konečný index, tedy můžeme si být jisti, že je roven nějakému přirozenému číslu n , neboli $\exists n \in \mathbb{N} : |\Sigma^*/\sim_L| = n$
 - Vyberme $n+1$ slov z množiny $\Sigma^* : \varepsilon, a, \dots, a^n$
 - Zavedme parametry $i, j \in \mathbb{N}$, kde $0 \leq i \leq n \wedge 0 \leq j \leq n \wedge i \neq j$, abychom se obecně mohli odkazovat ke dvěma různým slovům a^i, a^j z našeho výběru slov
 - Myhill-Nerodova věta říká, že $\forall u, v \in \Sigma^* : u \sim_L v \Leftrightarrow (\forall w \in \Sigma^* : uw \in L \Leftrightarrow vw \in L)$, tedy aby řetězce a^i, a^j byly v relaci $\sim_L$, muselo by platit $\forall w \in \Sigma^* : a^i w \in L \Leftrightarrow a^j w \in L$. Zkusme najít řetězec w , který by tuto podmínku porušil
 - Zvolme $w = b^i$, potom $a^i b^i \in L$, ale $a^j b^i \notin L$, neboť jsme zavedli $i \neq j$, slova a^i, a^j tedy nejsou v relaci $\sim_L$
 - To ale znamená, že žádná dvě různá slova z našeho výběru $\varepsilon, a, \dots, a^n$ nemohou být v relaci $\sim_L$, neboť jsme se pomocí parametrů i, j odkazovali k libovolným dvěma různým slovům z tohoto výběru
 - Pokud nejsou žádná dvě různá slova $\varepsilon, a, \dots, a^n$ v relaci $\sim_L$, nemohou být ve stejné třídě rozkladu $\Sigma^*/\sim_L$, tedy těchto tříd musí být určitě nejméně $n+1$, neboť v našem výběru je $n+1$ slov
 - My jsme ale v úvodu řekli, že tříd rozkladu $\Sigma^*/\sim_L$ je přesně n , což je spor. Ukázalo se, že pro každé přirozené číslo n platí, že počet tříd tohoto rozkladu je vyšší než n , tedy je nekonečný
 - Spor vychází z předpokladu, že L je regulární jazyk, tudíž L není regulární

Ukázka důkazu regularity pomocí Myhill-Nerodovy věty.

- Ukažme, že jazyk $L = \{w \in \{a, b\}^* \mid \#_a(w) \bmod 2 = 0\}$ je regulární
 - Sestrojíme relaci pravé kongruence $\sim$ takovou, že L bude sjednocením některých tříd rozkladu $\Sigma^*/\sim$ a relace $\sim$ bude mít konečný index
 - $\forall u, v \in \Sigma^* : u \sim v \Leftrightarrow (\#_a(u) \bmod 2 = \#_a(v) \bmod 2 = 0 \wedge u \neq \varepsilon \wedge v \neq \varepsilon) \vee (\#_a(u) \bmod 2 = \#_a(v) \bmod 2 = 1) \vee (u = v = \varepsilon)$

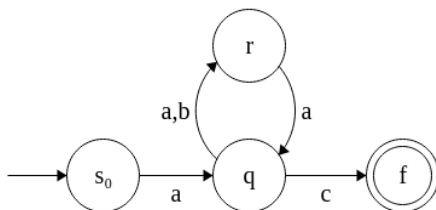
- Rozklad $\Sigma^* \setminus L = \{\{\epsilon\}, \{w \in \{a, b\}^+ \mid \#_a(w) \bmod 2 = 0\}, \{w \in \{a, b\}^+ \mid \#_a(w) \bmod 2 = 1\}\}$, tedy sjednocením některých tříd tohoto rozkladu s konečným indexem získáme jazyk L
- L je tedy regulární



Pumping lemma pro regulární jazyky

Pumping lemma pro regulární jazyky je tvrzení, které popisuje vlastnost, která musí platit pro každý regulární jazyk. Pokud tedy pro nějaký jazyk tato vlastnost neplatí, určitě není regulární. Tímto způsobem je možné **dokazovat neregularitu** jazyka.

Intuice. Mějme DKA na obrázku.



Tento automat obsahuje cyklus, díky čemuž dokáže i pomocí konečného stavového prostoru přijímat nekonečný jazyk. Pokud automat přijímá nějaký řetězec, jehož délka je vyšší než počet stavů automatu, pak je jasné, že při tomto akceptujícím běhu bylo nutné některý stav navštívit vícekrát (projít cyklem).

Uvažujme o přijetí slova $w = \mathbf{abaaac}$. Jelikož $|w| = 6$, ale automat má pouze 4 stavy, jistě musí při akceptujícím běhu slova w být navštíven cyklus:

$$(s_0, \mathbf{abaaac}) \vdash (\mathbf{q}, \mathbf{baaac}) \vdash (\mathbf{r}, \mathbf{aaac}) \vdash (q, \mathbf{aac}) \vdash (r, \mathbf{ac}) \vdash (q, \mathbf{c}) \vdash (f, \epsilon)$$

Pokud bychom z tohoto akceptujícího běhu daný cyklus vypustili, pak by takto přečtené slovo určitě bylo také přijato:

$$(s_0, \mathbf{aaac}) \vdash (q, \mathbf{aac}) \vdash (r, \mathbf{ac}) \vdash (q, \mathbf{c}) \vdash (f, \epsilon)$$

Podobně, pokud bychom daný cyklus v libovolném kladném konečném počtu zopakovali, takto přečtené slovo by rovněž bylo automatem přijato:

$$(s_0, \mathbf{aaabaaac}) \vdash (\mathbf{q}, \mathbf{aabaac}) \vdash (\mathbf{r}, \mathbf{abaaac}) \vdash (\mathbf{q}, \mathbf{baaac}) \vdash (\mathbf{r}, \mathbf{aaac}) \vdash (q, \mathbf{aac}) \vdash (r, \mathbf{ac}) \vdash (q, \mathbf{c}) \vdash (f, \epsilon)$$

Navíc, pokud čteme nějaké velmi dlouhé slovo (které je výrazně delší než počet stavů automatu), musí být navštíven cyklus už někdy ze začátku čtení slova (ve stejném nebo nižším počtu kroků než je počet stavů automatu), neboť s každým krokem před prvním navštívením cyklu snižujeme množinu stavů, které ještě nebyly navštíveny. Jakmile tuto množinu vyprázdníme, další krok nás jistě povede do stavu, který jsme už dříve navštívili (uzavřeme cyklus).

$$\begin{aligned}
 (s_0, \text{aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaac}) &\vdash \\
 (q, \text{aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaac}) &\vdash \\
 (r, \text{aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaac}) &\vdash \\
 (q, \text{aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaac}) &\vdash \\
 (q, \text{aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaac}) &\vdash^* \\
 \dots &\vdash^* \\
 (f, \varepsilon) &
 \end{aligned}$$

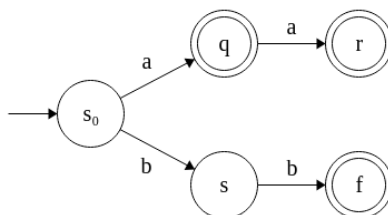
Tyto tři vlastnosti využijeme při formulaci Pumping lemmatu pro regulární jazyky

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

Jinak řečeno, pokud je jazyk L regulární, pak existuje dostatečně vysoké číslo k (můžeme chápat například jako počet stavů automatu přijímajícího daný jazyk), že každé slovo, které patří do jazyka L a je delší než k , lze rozdělit na tři části x, y, z (lze chápat tak, že část x je část slova čtená před navštívením prvního cyklu, část y je část slova čtená v rámci prvního běhu cyklem a část z je zbytek slova), přičemž část y je neprázdná (neboť v rámci běhu cyklem je nutné něco přečíst), části x, y jsou poměrně krátké (neboť první běh cyklem je dokončen už velice brzy ze začátku čtení slova) a část y lze v libovolném konečném počtu zopakovat, přičemž takto přečtené slovo stejně bude automatem přijato (neboť počet provedení cyklu nijak neovlivní dosažení koncového stavu, pokud původní slovo automatem přijato bylo).

Jde o vlastnost, která přímo plyne z konstrukce konečných automatů přijímajících nekonečné jazyky. Pokud tedy nějaký jazyk tuto vlastnost nemá, nelze pro něj zkonstruovat konečný automat, tedy není regulární.

Pumping lemma pro regulární jazyky platí i pro konečné jazyky, ačkoliv automaty přijímající takové jazyky nemají žádné dosažitelné a současně ukončující cykly, které by bylo možné v běhu opakovat, nebo vypouštět. Uvažujme o následujícím automatu.



Tento automat přijímá konečný jazyk $\{a, aa, bb\}$. Pokud bychom v Pumping lemmatu zvolili $k = 5$, pak by bylo lemma splněno i pro tento konečný jazyk, neboť žádné slovo z něj není delší než

k , tedy levá strana implikace na pravé straně Pumping lemmatu by nebyla nikdy splněna, tudíž pravá strana Pumping lemmatu by platila vždy.

Toto ovšem platí pouze pro konečné jazyky, neboť v nich jsou délky řetězců shora ohraničeny.

Důkaz neregularity pomocí Pumping lemmatu. Pro důkaz neregularity pomocí Pumping lemmatu převedeme Pumping lemma do tvaru obměněné věty.

$$(\forall k > 0 : \exists w \in \Sigma^* : w \in L \wedge |w| \geq k \wedge \\ \forall x, y, z \in \Sigma^* : xyz = w \wedge |y| > 0 \wedge |xy| \leq k \Rightarrow \exists i \in \mathbb{N} : xy^iz \notin L) \Rightarrow L \notin \mathcal{L}_3$$

Pokud pro jazyk L ukážeme, že levá strana Pumping lemmatu ve tvaru obměněné věty platí, pak z toho plyne, že jazyk L není regulární. Ukažme důkaz neregularity jazyka $L = \{a^n b^n \mid n \geq 0\}$.

Uvažujme o libovolném $k > 0$. Pro každé takové k zvolme řetězec $w \in L$, který je delší nebo stejně dlouhý jako k . Zvolme například $w = a^k b^k$, toto slovo vždy patří do jazyka L a je delší než k . Nyní uvažujme o všech možných rozděleních $w = xyz$, která splňují ty podmínky, že $|y| > 0$ a $|xy| \leq k$. Tato rozdělení můžeme vyjádřit následovně:

$$\begin{aligned} x &= a^\alpha \\ y &= a^\beta \\ z &= a^{k-\alpha-\beta} b^k, \end{aligned}$$

kde $\alpha + \beta \in \mathbb{N}$, $\beta > 0$ a $\alpha + \beta \leq k$. Lze snadno nahlédnout, že $xyz = a^\alpha a^\beta a^{k-\alpha-\beta} b^k = a^k b^k = w$. Takto jsme parametricky vyjádřili veškerá možná rozdělení splňující dané požadavky. Nyní zvolme $i = 2$ a studujme slovo $w' = xy^iz = xy^2z = a^\alpha a^{2\beta} a^{k-\alpha-\beta} b^k = a^{k+\beta} b^k \notin L$, neboť $\beta > 0$, což odporuje definici jazyka L , neboť počty symbolů a , b se nerovnají. Splnili jsme levou stranu obměněné věty, tedy jazyk L není regulární.

Pokud v textu najdete chybu, nebudete něčemu rozumět nebo budete mít dojem, že by bylo vhodné něco doplnit, kontaktujte na discordu uživatele kocotom.