V minulé přednášce jsme zavedli regulární jazyky jako ty jazyky L, pro které existuje konečný automat (DFA) M, který přijímá jazyk L. Nejdříve si ukážeme jednu vlastnost, kterou každý regulární jazyk má (pumping lemma), potom vlastnost, která regulární jazyky plně charakterizuje (Nerodova věta).

Začneme **nutnou** podmínku pro to, aby daný jazyk byl regulární. Tvrzení se také nazývá lemma o vkládání. Poznamenejme, že tato podmínka není postačující.

2.1.16 Pumping lemma pro regulární jazyky. Pro každý regulární jazyk L nad abecedou Σ existuje přirozené číslo n s touto vlastností:

Každé slovo $u \in L$, které je delší než n, lze rozdělit na tři slova $u = xwy$ tak, že

1. $|xw| \leq n$,
2. $w \neq \varepsilon$
3. a pro každé přirozené číslo $i = 0, 1, \ldots$ platí $xw^iy \in L$.

□

Dříve než pumping lemma dokážeme, ukažme, jak se dá toto tvrzení využít. Z toho, že se jedná pouze o nutnou podmínku, vyplývá, že je možno jej využít pouze pro důkaz, že nějaký jazyk **není** regulární.

2.1.17 Fakt. Jazyk $L = \{0^m\, 1^m\, ;\, m \geq 0\}$ není regulární jazyk. □

Zdůvodnění. Kdyby L byl regulární jazyk, muselo by existovat přirozené číslo n s vlastnostmi 1 až 3 z pumping lemmatu 2.1.16. Ukážeme, že to vede ke sporu.

Předpokládejme, že takové číslo n existuje. Vezmeme slovo $u = 0^n\, 1^n$, které je délky $2n$, což je víc než n. Podle pumping lemmatu lze u rozložit na tři slova $u = xwy$ s vlastnostmi 1 až 3. Ukážeme, že to není možné.

Víme, že xw je prefix slova u a $|xw| \leq n$, proto se slovo xw skládá ze samých 0. Navíc $w \neq \varepsilon$, musí proto $w = 0^k$ po nějaké $k \geq 1$. Pak ale slovo $xw^2y = 0^{n+k}1^n$ nemá stejný počet 0 i 1, tj. neleží v jazyce L, ale podle pumping lemmatu by v L ležet mělo. Odvodili jsme spor; chyba byla v tom, že jsme předpokládali, že L je regulární. jazyk.

2.1.18 Myšlenka důkazu pumping lemmatu. Uvažujme libovolný regulární jazyk L. Protože L je regulární, existuje DFA $M = (Q, \Sigma, \delta, q_0, F)$, který tento jazyk přijímá (tj. $L = L(M)$).

Označme n počet jeho stavů. Vezměme libovolné slovo $u \in L$ délky větší než n. Sled ve stavovém diagramu, který odpovídá práci automatu nad slovem u, musí obsahovat cyklus, protože má více hran než je jeho počet stavů, tudíž to nemůže být cesta. Označme

- x slovo, které odpovídá té části sledu, než poprvé vstoupíme do prvního cyklu,
- w slovo, které odpovídá jednomu průchodu tímto cyklem, a
- y slovo, které odpovídá zbylé části sledu.

Není těžké se přesvědčit, že slova x, w, y splňují všechny vlastnosti z pumping lemmatu.

Poznámka. Jak už jsme uvedli, podmínka z pumping lemmatu je pouze nutná, není postačující. Následující věta plně charakterizuje regulární jazyky, tj. jazyky, které jsou přijímány konečným automatem.

2.1.19 Nerodova věta. Je dán jazyk L nad abecedou Σ. Pak L je regulární jazyk právě tehdy, když existuje ekvivalence T na množině všech slov $\Sigma^\star$ taková, že

1. L je sjednocení některých tříd ekvivalence T.
2. Jestliže pro nějaká slova $u, v \in \Sigma^\star$ platí $u\ T\ v$, pak pro každé slovo $w \in \Sigma^\star$ platí také $uw\ T\ vw$.
3. T má pouze konečně mnoho tříd ekvivalence.

□

Poznamenejme, že druhá podmínka vlastně říká, že ekvivalence T je pravá kongruence monoidu $(\Sigma^\star, \cdot, \varepsilon)$.

Myšlenka důkazu. Jestliže je jazyk L regulární, pak existuje DFA $M = (Q, \Sigma, \delta, q_0, F)$, takový, že $L = L(M)$. Definujme relaci T na $\Sigma^\star$ takto:

$$u\ T\ v \quad \text{právě tehdy, když} \quad \delta^\star(q_0, u) = \delta^\star(q_0, v).$$

Není těžké ukázat, že takto definovaná relace splňuje všechny podmínky Nerodovy věty.

Předpokládejme, že pro jazyk L existuje ekvivalence T splňující všechny podmínky z Nerodovy věty. Označme $[u]_T$ třídu ekvivalence T obsahující slovo u. Definujme DFA $M = (Q, \Sigma, \delta, q_0, F)$ takto:

$$Q = \{[u]_T\,;\, u \in \Sigma^\star\},\;\; q_0 = [\varepsilon]_T,\; F = \{[u]_T\,;\, [u]_T \subseteq L\};$$

$$\delta([u]_T, a) = [ua]_T \quad \text{pro každé } a \in \Sigma.$$

Není těžké ukázat, že DFA M přijímá jazyk L. □

2.1.20 Poznámka. Nerodova věta se také dá použít k tomu, abychom ukázali, že některý jazyk není regulární. Ukážeme si to na příkladu.

Fakt. Jazyk $L = \{0^m\, 1^m\,;\, m \geq 0\}$ není regulární jazyk.

Zdůvodnění. Předpokládejme, že jazyk L je regulární. Pak existuje ekvivalence T s vlastnostmi z Nerodovy věty 2.1.19. Uvažujme slova

$$0, 0^2, 0^3, 0^4, \ldots, 0^n, \ldots$$

Těchto slov je nekonečně mnoho a ekvivalence T má pouze konečně mnoho tříd, proto musejí existovat přirozená čísla i, k, $i \neq k$, taková, že $0^i\ T\ 0^k$.

Protože ekvivalence T splňuje druhou podmínku z Nerodovy věty, musí platit $0^i w\ T\ 0^k w$ pro každé slovo w. Zvolme $w = 1^i$. pak dostáváme

$$0^i 1^i\ T\ 0^k 1^i.$$

Ovšem slovo $0^i 1^i$ patří do jazyka L, kdežto slovo $0^k 1^i$ do jazyka L nepatří. To je ale v rozporu s první podmínkou z Nerodovy věty; totiž, že jazyk L je sjednocením některých tříd ekvivalence T. Proto jazyk L není regulární. □

2.1.21 Jak dokázat, že daný automat přijímá daný jazyk. Obvykle není problém vymyslet si pro daný regulární jazyk L *nějaký* automat, problém je *dokázat*, že funguje, tj. že přijímá správný jazyk.

Vyzkoušením činnosti automatu na několika příkladech lze prokázat, že automat funguje špatně (přijímá, co nemá, nebo nepřijímá, co má), ale nelze prokázat, že na všech nekonečně mnoha slovech funguje dobře.

Jednou z možností, jak dokázat, že automat přijímá daný jazyk, je využití Nerodovy věty — metoda invariantů.

Každému stavu q přiřadíme přesný popis všech slov, které převedou počáteční stav q_0 do stavu q. Tím popíšeme jednotlivou třídu ekvivalence T z Nerodovy věty. To se většinou dělá pomocí formule $I_q(u)$ (kde slovo u je proměnná); a tuto formuli nazveme *invariantem* pro stav q. Abychom opravdu dokázali, že náš automat pracuje správně, je třeba, aby:

1. Prázdné slovo splňovalo invariant I_{q_0} počátečního stavu q_0.
2. Každé slovo u nad vstupní abecedou splňovalo právě jeden invariant (tj. každé slovo u leží v právě jedné třídě ekvivalence T).
3. Pro každý stav q a každý vstupní symbol x platila implikace *Jestliže slovo u odpovídá invariantu $I_q(u)$, pak slovo ux odpovídá invariantu $I_{\delta(q,x)}(ux)$.*

   Tj.

   $$I_q(u) \Rightarrow I_{\delta(q,x)}(ux) \quad .$$

4. Pro každé slovo nad vstupní abecedou platilo, že $u \in L$ právě tehdy, když u splňuje invariant pro některý koncový stav $q \in F$ (tj. platilo $I_q(u)$ pro nějaký koncový stav q).

Vymyslet invarianty nemusí být úplně snadné, ale jsou-li již vymyšleny, pak ověření výše uvedených podmínek je již rutinní záležitostí.

2.1.22 Ekvivalentní automaty. Pro jeden regulární jazyk může existovat více DFA, které tento jazyk přijímají. Nejlepší by bylo, kdybychom byli schopni najít ten „nejjednodušší“ automat mezi nimi. Pro deterministické automaty to je možné – ukážeme postup, jak najít redukovaný automat k danému DFA a ukážeme, že jestliže dva automaty přijímají stejný jazyk, pak k nim redukované automaty se liší pouze přejmenováním stavů (jsou isomorfní). Dokonce i přejmenování stavů (isomorfismus) se dá najít jednoduchým algoritmem. Dříve než redukované automaty zavedeme, uvedeme pojem ekvivalentních automatů.

Definice. Řekneme, že dva automaty M_1 a M_2 jsou *ekvivalentní*, jestliže přijímají stejný jazyk, tj. jestliže $L(M_1) = L(M_2)$. □

V dalším textu ukážeme, jak k danému DFA najít automat, který je „co nejjednodušší“ a který přijímá stejný jazyk. To bude právě redukovaný automat.

Postup rozdělíme na dva kroky: 1) Nejprve z automatu odstraníme nedosažitelné stavy; 2) pak stavy, které „nerozlišíme“ nějakým vstupním slovem, prohlásíme za ekvivalentní — tj. „za stejné“.

2.1.23 Dosažitelné stavy. Je dán DFA $M = (Q, \Sigma, \delta, q_0, F)$. Řekneme, že stav $q \in Q$ je *dosažitelný*, jestliže existuje slovo $u \in \Sigma^\star$ takové, že $\delta^\star(q_0, u) = q$. □

Jinými slovy, stav q je dosažitelný, jestliže je dosažitelný z počátečního stavu q_0 ve stavovém diagramu automatu M (tj. z q_0 vede do q orientovaný sled).

Je zřejmé, že stavy, které nejsou dosažitelné, nemají vliv na jazyk, který daný automat přijímá.

2.1.24 Postup nalezení dosažitelných stavů. Množinu dosažitelných stavů najdeme např. následujícím postupem, který je vlastně prohledávání do šířky stavového diagramu.

$Q_0 := \{q_0\}$
`repeat`
    $Q_{i+1} := Q_i \cup \{\delta(q, a)\,;\, q \in Q_i, a \in \Sigma\}$
`until` $Q_{i+1} = Q_i$.
`return` $Q' = Q_i$

2.1.25 Ekvivalence stavů $\sim$. Máme dán DFA $M = (Q, \Sigma, \delta, q_0, F)$. Řekneme, že dva stavy $p, q \in Q$ jsou *ekvivalentní*, jestliže pro každé slovo $u \in \Sigma^\star$ platí

$$\delta^\star(p, u) \in F \quad \text{právě tehdy, když} \quad \delta^\star(q, u) \in F.$$

Fakt, že dva stavy p a q jsou ekvivalentní, zapisujeme $p \sim q$. □

2.1.26 Redukovaný automat. Nyní přistoupíme k definici redukovaného automatu.

Definice. Je dán DFA $M = (Q, \Sigma, \delta, q_0, F)$. Řekneme, že M je *redukovaný*, jestliže nemá nedosažitelné stavy a žádné jeho dva různé stavy nejsou ekvivalentní. □

Jinými slovy, jestliže ekvivalence $\sim$ je identická ekvivalence na množině stavů Q.