21 Deterministický konečný automat, jazyk přijímaný konečným automatem. (A4B01JAG)

21.1 Jazyky - úvod

21.1.1 Abeceda

Konečnou neprázdnou množinu Σ budeme nazývat *abecedou*. Prvky množiny Σ nazýváme symboly, písmeny apod.

21.1.2 Slovo nad abecedou

Pro danou abecedu Σ je *slovo nad* Σ libovolná konečná posloupnost prvků abecedy Σ. Tedy např. pro $\Sigma = \{\text{a, b}\}$ jsou *aab*, *b*, *bbaba* slova nad Σ.

Prázdné slovo, značíme je ϵ, je posloupnost, která neobsahuje ani jeden symbol.

21.1.3 Délka slova

Je dáno slovo nad abecedou Σ. *Délka slova* je rovna délce posloupnosti, tj. počtu symbolů, které se ve slově nacházejí. Délku slova u značíme $|u|$.

Tedy, délka slova *aab* je rovna 3, délka *b* je 1, délka prázdného slova ϵ je 0.

21.1.4 Zřetězení slov

Je dána abeceda Σ. Pro dvě slova u, v nad abecedou Σ definujeme operaci *zřetězení* takto: Je-li $u = a_1a_2 \dots a_n$ a $v = b_1b_2 \dots b_k$, pak

$$u \cdot v = a_1a_2 \dots a_nb_1b_2 \dots b_k.$$

Často znak pro operaci zřetězení vynecháváme; píšeme tedy uv místo přesnějšího $u \cdot v$.

Zřetězení slov je asociativní operace na množině všech slov nad danou abecedou, prázdné slovo ϵ je neutrální prvek této operace.

$\Sigma^\star$, Σ^+. Označíme $\Sigma^\star$ množinu všech slov nad abecedou Σ. (Tj. prázdné slovo patří do $\Sigma^\star$) Pak $\Sigma^\star$ spolu s operací zřetězení tvoří monoid, jehož neutrálním prvkem je prázdné slovo ϵ.

Označíme Σ^+množinu všch neprázdných slov nad abecedou Σ. (Tj. $\Sigma^+ = \Sigma^\star \setminus \{\epsilon\}$.) Pak $\Sigma^\star$ spolu s operací zřetězení tvoří pologrupu.

Zřetězení slov není komutativní. Např. pro $u = aab$ a $v = b$ je $uv = aabb$, ale $vu = baab$.

Pro libovolná slova u a v nad stejnou abecedou platí:

$$|uv| = |u| + |v| .$$

Je-li u slovo nad abecedou Σ, pak můžeme opakování slova n-krát zapsat zjednodušeně jako $uuuu...u = u^n$. Přesněji:

$$u^0 = \epsilon,$$

$$u^{i+1} = uu^i \text{ pro každé } i.$$

21.1.5 Podslovo

Je dáno slovo u. Řekneme, že slovo w je podslovem slova u, jestliže existují slova x, y taková, že

$$u = xwy.$$

21.1.6 Prefix slova

Je dáno slovo u. Řekneme, že slovo w je prefix slova u, jestliže existuje slovo y takové, že

$$u = wy.$$

21.1.7 Jazyk nad abecedou

Je dána abeceda Σ. *Jazyk* L nad abecedou Σ je libovolná množina slov, tj. $L \subseteq \Sigma^\star$.

Je-li Σ abeceda, pak množina všech slov $\Sigma^\star$ je spočetná. Jazyků, jako podmnožin spočetné množiny, je víc - nespočetně mnoho.

21.2 Deterministické konečné automaty

21.2.1 Použití

Konečné automaty se používají v různých oborech. Jako příklady můžeme uvést překladače, dále se používají při zpracování přirozeného jazyka, pří návrzích hardwaru.

Zhruba řečeno, konečný automat obsahuje konečnou množinu stavů Q, konečnou množinu vstupů Σ, přechodovou funkci δ a počáteční stav q_0. V některých případech ještě i množinu výstupních symbolů Y a výstupních funkcí.

21.2.2 Příklad 1

Uvažujme zjednodušený příklad automatu na kávu. Automat přijímá mince 1 Kč, 2 Kč a 5 Kč. Automat vydává jediný druh kávy, káva stojí 7 Kč. Automat na tlačítko s vrátí nevyužité peníze. Tento příklad uvedeme podrobněji.

Zde $Q = \{0,1,2,3,4,5,6\}$, $\Sigma = \{1,2,5,s\}$, $Y = \{K,0,1,2,3,4,5,6\}$, přechodová a výstupní funkce jsou dány následující tabulkou:

V prvním sloupci jsou stavy, ve kterých se automat může nacházet, v prvním řádku jsou vstupní symboly. V řádku odpovídajícím stavu q a sloupci se vstupen a je dvojice (nový stav, výstup). (K znamená kávu, číslo udává vrácené peníze).

	1	2	5	s
0	1/0	2/0	5/0	0/0
1	2/0	3/0	6/0	0/1
2	3/0	4/0	0/K	0/2
3	4/0	5/0	1/K	0/3
4	5/0	6/0	2/K	0/4
5	6/0	0/K	3/K	0/5
6	0/K	1/K	4/K	0/6

21.2.3 Stavový diagram

Kromě tabulky, můžeme konečný automat zadat též stavovým diagramem.

Je dán konečný automat s množinou stavů Q, množinou vstupních symbolů Σ, přechodovou funkcí δ. *Stavovým diagramem* nazýváme orientovaný ohodnocený graf, jehož vrcholy jsou stavy (tj. $V = Q$) a orientovaná hrana vede z vrcholu q do vrcholu p právě tehdy, když $\delta(q,\ a) = p$; v tomto případě je hrana ohodnocena vstupním symbolem a pro Moorův automat a DFA, nebo dvojicí $a/\lambda(q,\ a)$ v případě, že se jedná o Mealyho automat.

Jesliže se jedná o Mooreův automat, vrcholy stavového diagramu jsou navíc ohodnoceny značkovací funkcí β. Pro akceptor, tj DFA, označujeme pouze množinu koncových stavů, a to buď šipkou mířící ze stavu ven nebo jiným označením stavů, které patří do množiny F. Počáteční stav q_0 je označován šipkou mířící do něj.

21.2.4 Obecně rozlišujeme čtuři typy automatů

Mealyho automat, Mooreův automat, akceptor a automat bez výstupu. Dále se budeme zabývat hlavně tzv. akceptory.

21.2.5 Mealyho automat

Mealyho automat je šestice $(Q,\ \Sigma,\ Y,\ \delta,\ q_0,\ \lambda)$. kde Q, Σ, Y a q_0 mají stejný význam jako v 21.2.1, přechodová funkce je zobrazení δ: $Q \times \Sigma \to Q$ a výstupní funkce je zobrazení λ: $Q \times \Sigma \to Y$.

Mealyho automat vypisuje během své činnosti výstupní funkci. Výstup závisí na aktuálním stavu a na přijatém znaku. Ve stavovém diagramu se výstupní hodnota píše nad každou hranu.

21.2.6 Moorův automat

Moorův automat je šestice $(Q, \Sigma, Y, \delta, q_0, \beta)$. kde Q, Σ, Y, δ a q_0 mají stejný význam jako v 21.2.5, β je zobrazení β: $Q \to Y$ (říká se mu značkovací funkce).

Moorův automat taky vypisuje během své činnosti značkovací funkci. Výstup závisí pouze na aktuálním stavu. Ve stavovém diagramu se výstupní hodnota píše do vrcholů.

21.2.7 Akceptor, též DFA

DFA (Deterministic finite automaton) je pětice $(Q, \Sigma, \delta, q_0, F)$, kde Q, Σ, δ a q_0 mají stejný význam jako v 21.2.6 a $F \subseteq Q$ je množina koncových (též přijímajících) stavů.

Jedná se vlastně o Mooreův automat, kde množina výstupních symbolů má dva prvky, totiž $Y = \{0, 1\}$, a proto značkovací funkci β nahrazujeme množinou těch stavů, kterým značkovací funkce přiřazuje 1.

21.2.8 Automat bez výstupu

Automat bez výstupu je „společnou částí" všech výše uvedených automatů; tj. jedná se o (Q, Σ, δ, q_0).

21.2.9 Rozšířená přechodová funkce

Je dán automat (Q, Σ, δ). Rozšířená přechodová funkce $\delta^\star$: $Q \times \Sigma^\star \to Q$ je definovaná induktivně takto:

1. $\delta^\star(q, \epsilon) = q$, pro všechna $q \in Q$,
2. $\delta^\star(q, ua) = \delta(\delta^\star(q, u), a)$, pro všechna $q \in Q$, $a \in \Sigma$, $u \in \Sigma^\star$.

21.2.10 Jazyk přijímaný konečným automatem

Je dán DFA $M = (Q, \Sigma, \delta, q_0, F)$. Řekneme, že slovo $u \in \Sigma^\star$ je *přijímáno* automatem M právě tehdy, když

$$\delta^\star(q_0, u) \in F.$$

Množina všech slov, které automat přijímá, se nazývá *jazyk přijímaný M*, značíme ji *L(M)*. Tedy,

$$L(M) = \{\omega | \delta^\star(q_0, \omega) \in F\}.$$

21.2.11 Regulární jazyky

Každý jazyk, který je přijímán některým DFA, nazveme *regulární jazyk*. Třídu všech regulárních jazyků označujeme **Reg**.

21.2.12 Pumping lemma pro regulární jazyky

Pro každý regulární jazyk L nad abecedou Σ (tj. jazyk, který je přijímán nějakým DFA) existuje přirozené číslo n s touto vlastností:

Jestliže nějaké slovo $u \in L$ je delší než n (tj. $|u| > n$), pak u lze rozdělit na tři slova $u = xwy$ tak, že

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

Důkaz: Předpokládejme, že jazyk L je regulární. Tedy existuje DFA M, který tento jazyk přijímá. 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 (má větší délku než je počet vrcholů – stavů). Označme x slovo, které odpovídá té části sledu než poprvé vstoupíme do cyklu, w slovo, které odpovídá jednomu průchodu tímto cyklem, a y slovo odpovídající zbylé části sledu.

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

Využití pumping lemmatu: Jazyk $L = \{0^m1^m | m \geq 0\}$ není regulární jazyk.

Kdyby L byl regulární jazyk, muselo by existovat přirozené číslo n s vlastností z 21.2.12. Položme $u = 0^n1^n$. Zřejmě $u \in L$ a $|u| = 2n > n$. Tedy $u = xwy$, $w \neq \epsilon$, $|xw| \leq n$ a $xw^2y \in L$. To ale není možné; slovo w by muselo obsahovat jen 0, protože délka slova xw je menší nebo rovna n a prefix slova u délky n je 0^n. Navíc slovo w není prázdné, a tedy $w = 0^k$ pro $0 < k \leq n$. Pak ale slovo xw^2y je rovno $0^{n+k}1^n$ a nemá stejný počet 0 i 1, tj. neleží v jazyce L. Spor.

21.2.13 Ekvivalentní automaty

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

21.2.14 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 M (tj. z q_0 vede do q orientovaný sled).

Je zřejmé, že stavy, které jsou nedosažitelné, nemají vliv na jazyk, který daný automat přijímá.

21.2.15 Ekvivalence stavů $\sim$

Máme 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

$$\delta^\star(p,u) \in F \text{ iff } \delta^\star(q,u) \in F.$$

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

21.2.16 Redukovaný automat

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. (To znamená, že ekvivalence $\sim$ je identická ekvivalence.)

21.2.17 Konstrukce relace $\sim$

Konstruujeme relace $\sim_i$, $i = 0, 1, \ldots$, na množině všech stavů Q takto:

- $p \sim_0 q$ právě tehdy, když buď $p,q \in F$ nebo $p,q \notin F$;
- je-li $i \geq 0$, pak $p \sim_{i+1} q$ právě tehdy, když $p \sim_i q$ a pro každé $a \in \Sigma$ máme $\delta(p,\ a) \sim_i \delta(q,\ a)$.

Věta: Platí

$$\sim_0 \supseteq \sim_1 \supseteq \ldots \supseteq \sim_i \supseteq \ldots$$

Existuje k takové, že $\sim_k$ je rovna $\sim_{k+1}$. Pak pro každé $j \geq 1$ platí $\sim_k = \sim_{k+j}$ a tedy $\sim_k = \sim$.

21.2.18 Algoritmus redukce

Je dán DFA $M = (Q, \Sigma, \delta, q_0, F)$.

1. Zkonstruujeme množinu Q' všech dosažitelných stavů automatu M. Postupujeme např. hledáním do šířky ze stavu q_0 ve stavovém diagramu.
2. Podle předchozího odstavce zkonstruujeme ekvivalenci $\sim$ pro DFA $M' = (Q, \Sigma, \delta, q_0, F \cap Q')$.
3. Vytvoříme DFA $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$, kde $Q_1 = Q'/\sim = \{[q]_\sim | q \in Q'\}$, $q_1 = [q_0]_\sim$, $\delta_1([q]_\sim, a) = [\delta(q,a)]_\sim$ a $F_1 = \{[q]_\sim | q \in F \cap Q'\}$.

Automat M_1 vznikl takto: za stavy jsme vzali třídy ekvivalence $\sim$, počáteční stav je třída, ve které leží původní počáteční stav q_0, přechodová funkce „pracuje" na třídách (což je možné vzhledem k vlastnosti $\sim$) a množina koncových stavů je množina těch tříd, ve kterých leží koncové stavy automatu M'.

21.2.19 Příklad

Je dán DFA M následující tabulkou:

	a	b
1	2	3
2	2	4
3	3	5
4	2	7
5	6	3
6	6	6
7	7	4
8	2	3
9	9	4

Nalezněte redukovaný automat k DFA M.

Řešení: Nejprve najdeme všechny dosažitelné stavy automatu M. Jsou to stavy {1,2,3,4,5,6,7}. Tedy $Q' = \{1,2,3,4,5,6,7\}$, $F' = F = \{3,5,6\}$.

Automat M' je dán tabulkou:

	a	b
1	2	3
2	2	4
3	3	5
4	2	7
5	6	3
6	6	6
7	7	4

Vytvoříme rozklad R_0 ekvivalence $\sim_0$:

$$O = \{1, 2, 4, 7\} \quad F = \{3, 5, 6\} .$$

Platí

$\delta(1, a) = 2 \in O$, $\delta(2, a) = 2 \in O$, $\delta(4, a) = 2 \in O$, $\delta(7, a) = 7 \in O$, $\delta(1, b) = 3 \in F$, $\delta(2, b) = 4 \in O$

Tedy musíme množinu O rozdělit na dvě podmnožiny, a to $\{1\}$ a $\{2, 4, 7\}$. Dále

$\delta(3, a) = 3 \in F$, $\delta(5, a) = 6 \in F$, $\delta(6, a) = 6 \in F$, $\delta(3, b) = 5 \in F$, $\delta(5, b) = 3 \in F$, $\delta(6, b) = 6 \in F$.

Proto množinu F nedělíme.

Rozklad odpovídající ekvivalenci $\sim_1$ je

$$A = \{1\} , \ O = \{2, 4, 7\} , \ F = \{3, 5, 6\} .$$

Výpočet zahrneme do tabulky

	a	b	$\sim_0$	a	b	$\sim_1$
1	2	3	O	O	F	A
2	2	4	O	O	O	O
3	3	5	F	F	F	F
4	2	7	O	O	O	O
5	6	3	F	F	F	F
6	6	6	F	F	F	F
7	7	4	O	O	O	O

Analogicky postupujeme k vytvoření ekvivalence $\sim_1$. Výpočet již zkrátíme jen do tabulky.

	a	b	$\sim_0$	a	b	$\sim_1$	a	b	$\sim_2$
1	2	3	O	O	F	A	O	F	A
2	2	4	O	O	O	O	O	O	O
3	3	5	F	F	F	F	F	F	F
4	2	7	O	O	O	O	O	O	O
5	6	3	F	F	F	F	F	F	F
6	6	6	F	F	F	F	F	F	F
7	7	4	O	O	O	O	O	O	O

Z tabulky vyplývá , že $\sim_1 = \sim_2$. Proto $\sim_1 = \sim$ je hledaná ekvivalence.

Máme tedy tři třídy ekvivalence, a to A, O a F. Redukovaný automat M_1 je dán tabulkou

	a	b
A	O	F
O	O	O
F	F	F

Není těžké nahlédnout, že automat M_1 přijímá jazyk $L = \{bu | u \in \{a, b\}^\star\}$.

Věta: Automat M i k němu redukovaný automat M_1 přijímají stejný jazyk, tj. jsou ekvivalentní.

Věta: Dva DFA M_1 a M_2 přijímají stejný jazyk (tj. jsou ekvivalentní) právě tehdy, když jejich odpovídající redukované automaty se liší pouze pojmenováním stavů.

21.2.20 Nerodova věta

Je dán jazyk L nad abecedou Σ. Pak L je regulární jazyk (tj. je přijímán nějakým DFA) právě tehdy, když existuje ekvivalence R na množině všech slov $\Sigma^\star$ taková, že

1. L je sjednocení některých tříd ekvivalence R.
2. R splňuje následující podmínku: Je-li uRv pro u, $v \in \Sigma^\star$, pak pro každé slovo $w \in \Sigma^\star$ platí $uwRvw$.
3. R má pouze konečně mnoho tříd ekvivalence.

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

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

$$uRv \quad \text{iff} \quad \delta^\star(q_0, u) = \delta^\star(q_0, v).$$

Takto definovaná relace splňuje všechny podmínky Nerodovy věty.

Předpokládejme, že pro jazyk L existuje ekvivalence R splňující všechny podmínky z Nerodovy věty. Definujme DFA $M = (Q,\ \Sigma,\ \delta,\ q_0,\ F\ )$ takto:

$$Q = \{[u]_R \,|u \in \Sigma^\star\}, \quad q_0 = [\epsilon]_R, \quad F = \{[u]_R \,|\, [u]_R \subseteq L\};$$

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

Pak DFA M přijímá jazyk L.

Poznámka: Podobně jako pumping lemma i Nerodova věta se dá použít k tomu, abychom ukázali, že některý jazyk není regulární. Navíc je ale možné Nerodovu větu použít i pro konstrukci automatu, který daný jazyk přijímá.