

Automaty a gramatiky

Ladislav Láška

13. června 2010

Obsah

1	Konečné automaty	2
1.1	Konečný automat, přechodová funkce	2
1.2	Přijímané slovo, jazyk	2
1.3	Regulární jazyky	2
1.4	Kongruence	2
1.5	Nerodova věta	3
1.6	Pumping lemma	3
1.7	Ekvivalence automatů a homomorfismus	4
1.8	Věta o ekvivalenci automatů	4
1.9	Dosažitelné stavy	4
1.10	Hledání dosažitelných stavů	5
1.11	Ekvivalentní stavy	5
1.12	Ekvivalence stavů po i krocích	5
1.13	Vlastnosti ekvivalence po i krocích	6
1.14	Algoritmus hledání ekvivalence stavů	6
1.15	Automatová kongruence	6
1.16	Podílový automat	7
1.17	Podílový automat a ekvivalence stavů	7
1.18	Redukce automatu	7
1.19	Věta o isomorfismu reduktů	8
1.20	Normalizace automatu	8
1.21	Hledání nejkratšího slova	8
2	Nedeterministické konečné automaty	9
2.1	Převod NKA na KA	9
2.2	λ -přechod	9
3	Dvousměrné konečné automaty	9
3.1	Věta o síle dvousměrných automatů	9
4	Množinové operace nad jazyky	10
4.1	Množinové operace nad regulárními jazyky	10
4.2	Uzavřenost množinových operací nad regulárními jazyky	11

1 Konečné automaty

1.1 Konečný automat, přechodová funkce

Definice Konečným automatem nazveme pětici

$$A = (Q, X, \delta, q_0, F) \quad (1)$$

kde po řadě (Q, X, δ, q_0, F) jsou stavový prostor, vstupní abeceda, přechodová funkce, počáteční stav a množina přijímajících stavů.

Definice Rozšířená přechodová funkce $\delta^*: Q \times X^* \rightarrow Q$ definujeme induktivně:

$$\delta^*(q, \lambda) = q \quad (2)$$

$$\delta^*(q, wx) = \delta(\delta^*(q, w), x), \quad x \in X, w \in X^* \quad (3)$$

1.2 Přijímané slovo, jazyk

Definice Slovo w je přijímáno automatem A právě když

$$\delta^*(q_0, w) \in F \quad (1)$$

Definice Jazyk přijímaný automatem A definujeme jako množinu přijímaných slov:

$$L(A) = \{w | w \in X^* \wedge \delta^*(q_0, w) \in F\} \quad (2)$$

1.3 Regulární jazyky

Definice Regulární jazyky jsou právě ty jazyky, které jsou rozpoznatelné konečným automatem.

1.4 Kongruence

Definice Necht' X je konečná abeceda a \sim je relace ekvivalence na X^* . Potom \sim :

1. je **pravá kongruence**, jestliže

$$\forall u, v, w \in X^* \quad u \sim v \Rightarrow uw \sim vw \quad (1)$$

2. je **konečného indexu**, jestliže rozklad X^*/\sim má konečný počet tříd.

1.5 Nerodova věta

Věta Nechť L je jazyk nad konečnou abecedou X . Potom následující tvrzení jsou ekvivalentní:

1. L je rozpoznatelný konečným automatem.
2. Existuje pravá kongruence \sim konečného indexu na X^* tak, že L je sjednocením jistých tříd rozkladu X^*/\sim .

Důkaz

1. \rightarrow 2. (máme automat, konstruuje pravou kongruenci)
Definujeme relaci:

$$u \sim v \iff \delta^*(q_0, u) = \delta^*(q_0, v) \quad (1)$$

Což je přímo z definice ekvivalence a je to pravá kongruence (díky vlastnostem rozšířené předchodové funkce). Navíc protože má automat konečně mnoho stavů, má také konečně mnoho tříd ekvivalence.

2. \rightarrow 1. (máme pravou kongruenci konečného indexu, konstruuje automat) Definujeme automat jako:
 - (a) abeceda X je již dána
 - (b) stavy Q nechť jsou třídy rozkladu
 - (c) stav $q_0 = [\lambda]$
 - (d) koncové stavy jsou třídy rozkladu definující jazyk: $L = \bigcup_{i=1..n} c_i$, $F = \{c_1, \dots, c_n\}$
 - (e) přechodovou funkci $\delta([u], x) = [ux]$

1.6 Pumping lemma

Věta Nechť L je regulární jazyk. Potom existuje přirozené číslo n takové, že libovolné slovo $z \in L$, kde $|z| \geq n$ lze psát ve tvaru uvw a

$$|uv| \leq n \quad \wedge \quad |v| \geq 1 \quad \wedge \quad \forall i \geq 0 \quad uv^i w \in L$$

Důkaz Jako n vezmeme počet stavů příslušného automatu. Pokud vezmeme slovo délky alespoň n , podle krabičkového principu se nutně musí nějaký stav opakovat. Vezmeme tedy stav p pravý takový. Nahlédneme:

$$\delta(q_0, u) = p \quad (1)$$

$$\delta(q_0, uv) = p \quad (2)$$

kde u a uv jsou slova, která nás dovedla do stavu p . Zřejmě tedy $|uv| \leq n$, protože se opakoval pouze stav p a $|v| \geq 1$, protože se p opakoval. Smyčku v tedy můžeme projít libovolně krát.

1.7 Ekvivalence automatů a homomorfismus

Definice Automaty A a B jsou **ekvivalentní**, právě když rozpoznávají stejný jazyk: $L(A) = L(B)$.

Definice Nechtě A_1 a A_2 jsou konečné automaty a $h : Q_1 \rightarrow Q_2$ zobrazení. Řekneme, že h je **automatový homomorfismus**, právě když:

1. $h(q_1) = q_2$
2. $h(\delta_1(q, x)) = \delta_2(h(q), x)$
3. $q \in F_1 \Leftrightarrow h(q) \in F_2$

Definice Homomorfismus nazveme **isomorfismus** právě když je prostý.

1.8 Věta o ekvivalenci automatů

Věta Nechtě A_1 a A_2 jsou konečné automaty a existuje homomorfismus $h : A_1 \rightarrow A_2$. Pak jsou A_1 a A_2 ekvivalentní.

Důkaz Konečnou iterací. Podle definice homomorfismu platí:

$$h(\delta_1(q, w)) = \delta_2(h(q), w) \quad w \in X^* \quad (1)$$

Tedy

$$w \in L(A_1) \Leftrightarrow \delta_1(q_1, w) \in F_1 \quad (2)$$

$$\Leftrightarrow h(\delta_1(q_1, w)) \in F_2 \quad (3)$$

$$\Leftrightarrow \delta_2(h(q_1), w) \in F_2 \quad (4)$$

$$\Leftrightarrow \delta_2(q_2, w) \in F_2 \Leftrightarrow w \in L(A_2) \quad (5)$$

1.9 Dosažitelné stavy

Definice Nechtě A je konečný automat. Řekneme, že stav $q \in Q$ je **dosažitelný**, právě když $\exists w \in X^* \quad \delta^*(q_0, w) = q$.

Věta Nechtě A je konečný automat a $P \subseteq Q$ jsou právě všechny dosažitelné stavy A .

Důkaz (triviální)

1.10 Hledání dosažitelných stavů

Algoritmus Hledejme dosažitelné stavy po i krocích:

1. $M_0 = \{q_0\}$
2. Spočítej dosažitelné stavy pro $i + 1$:

$$M_{i+1} = M_i \cup \{q \mid q \in Q \wedge \exists p \in M_i, x \in X : \delta(p, x) = q\} \quad (1)$$

3. Pokud $M_{i+1} == M_i$, konči. Jinak opakuj 2. pro další i .

Důkaz (triviální)

1.11 Ekvivalentní stavy

Definice Nechť A je konečný automat a $p, q \in Q$ jsou jeho stavy. Řekneme, že stavy p a q jsou ekvivalentní, právě když:

$$\forall w \in X^* \quad \delta^*(p, w) \in F \Leftrightarrow \delta^*(q, w) \in F \quad (1)$$

1.12 Ekvivalence stavů po i krocích

Definice Nechť A je konečný automat a $p, q \in Q$ jsou jeho stavy. Řekneme, že stavy p a q jsou ekvivalentní po i krocích ($p \sim^i q$) právě když:

$$\forall w \in X^*, |w| \leq i : \quad \delta^*(p, w) \in F \Leftrightarrow \delta^*(q, w) \in F \quad (1)$$

Tvrzení Nechť A je konečný automat a $p, q \in Q$ jsou jeho stavy. Potom:

$$p \sim q \Leftrightarrow \forall i \quad p \sim^i q \quad (2)$$

Důkaz (triviální)

Algoritmus Iterativní konstrukce ekvivalence po i -krocích:

$$p \sim^0 q \quad p \in F \Leftrightarrow q \in F \quad (3)$$

$$p \sim^{i+1} q \quad p \sim^i q \quad \wedge \quad \forall x \in X : \delta(p, x) \sim^i \delta(q, x) \quad (4)$$

Důkaz (triviální)

1.13 Vlastnosti ekvivalence po i krocích

Věta Necht' A je konečný automat a \sim^i je relace ekvivalence po i krocích. Potom:

1. $\forall i \geq 0$ je \sim^i ekvivalence na Q . Označme $\mathcal{R} = Q / \sim^i$.
2. \mathcal{R}_{i+1} zjemňuje \mathcal{R}_i
3. $\mathcal{R}_{i+1} = \mathcal{R} \Rightarrow \forall t > i \mathcal{R}_t = \mathcal{R}_i$
4. Necht' $|Q| = n$, potom $\exists k \leq n - 1 : \mathcal{R}_{k+1} = \mathcal{R}_k$
5. $\mathcal{R}_{k+1} = \mathcal{R}_k \Rightarrow (p \sim q \Leftrightarrow p \sim^k q)$

Důkaz

1. Triviálně z definice
2. Triviálně z definice
3. Triviálně z iteračního algoritmu (podmínky se proti minulé iteraci nezměnily)
4. Triviálně z maximálního počtu rozkladových tříd (konečného)
5. Triviálně podle definice a 3.

1.14 Algoritmus hledání ekvivalence stavů

Algoritmus Postupujeme iteračním algoritmem, dokud nenarazíme na dvě po sobě jdoucí stejné třídy ekvivalence. Stavů nacházející se ve stejné třídě ekvivalence na konci algoritmu jsou si ekvivalentní.

Důkaz (triviální)

1.15 Automatová kongruence

Definice Necht' \equiv je relace ekvivalence na Q . Řekneme, že \equiv je **automatovou kongruencí**, právě když:

$$\forall p \equiv q \in Q \Rightarrow (p \in F \Leftrightarrow q \in F) \wedge \forall x \in X : (\delta(p, x) \equiv \delta(q, x))$$

Tvrzení Ekvivalence stavů je automatovou kongruencí.

Důkaz (triviální z definice)

1.16 Podílový automat

Věta Necht' $A = (Q, X, \delta, q_0, F)$ je konečný automat a \equiv je automatová kongruence. Potom $A/\equiv = (Q/\equiv, X, \delta_\equiv, [q_0]_\equiv, \{[q]_\equiv | q \in F\})$, kde $\delta_\equiv([q], x) = [\delta(q, x)]_\equiv$, je konečný automat ekvivalentní s A .

Definice Automat A/\equiv nazýváme **podílový automat**.

Důkaz

1. Podílový automat je konečný automat: množiny stavů a abeceda jsou z definice konečné, přechodová funkce je definována korektně z vlastností automatové kongruence.
2. Ekvivalence automatů: stačí nalézt homomorfismus, ten jsme však dostali v podobě kongruence.

1.17 Podílový automat a ekvivalence stavů

Věta Necht' A je konečný automat a \sim je ekvivalence stavů. Potom :

1. A/\sim je konečný automat ekvivalentní s A
2. žádné stavy v A/\sim si nejsou ekvivalentní

Důkaz

1. Relace ekvivalence stavů je automatovou kongruencí, tedy automaty si jsou ekvivalentní.
2. Triviálně z definice ekvivalence. (lze dokázat sporem)

1.18 Redukce automatu

Definice Konečný automat je **redukovaný** právě když nemá nedosažitelné stavy a žádné dva stavy nejsou ekvivalentní. Automat B je **reduktem** automatu A právě když je B redukovaný a A a B jsou ekvivalentní.

Věta Ke každému konečnému automatu existuje nějaký jeho redukt.

Důkaz Konstruktivně vyřazením stavů a nalezením podílového automatu ekvivalencí stavů.

1.19 Věta o isomorfismu reduktů

Věta Pro libovolné dva redukované konečné automaty jsou následující tvrzení ekvivalentní:

1. automaty jsou ekvivalentní
2. automaty jsou isomorfní

Důkaz

1. \rightarrow 2. (hledáme isomorfismu z ekvivalentních reduktů)
Zkonstruujeme funkci $h : Q_1 \rightarrow Q_2$. Pro stav $q \in Q_1$ (který je dosažitelný) existuje slovo $u \in X^*$ vedoucí do q a v $h(q) = \delta_2(q_2, u)$. Což je funkce (protože jsou automaty redukované) a je prostá a na.
2. \rightarrow 1. (víme z definice)

Poznámka Pomocí redukce umíme řešit problémy:

1. Zjištění ekvivalence automatů (zredukujeme, znormalizujeme a porovnáme)
2. Zjištění $L(A) = \emptyset$
3. Zjištění $L(A) = X^*$

1.20 Normalizace automatu

Definice Normalizovaný tvar automatu vytvoříme tak, že:

1. Zafixujeme pořadí písmen v abecedě
2. Počáteční stav označíme 1
3. Tabulku popisující automat vyplňujeme po řádcích zleva doprava a pokud narazíme na nový stav, přiřadíme mu následující volné číslo.

1.21 Hledání nejkratšího slova

Pozorování Nejkratší slovo nalezenem při redukci automatu - přes rozkladové třídy. První stavy, kde se liší, udávají začátek slova, pak přečteme všechny stavy až do počátku.

2 Nedeterministické konečné automaty

Definice Nedeterministickým konečným automatem nazveme pěticí

$$A = (Q, X, \delta, S, F) \quad (1)$$

kde Q je množina stavů, X je vstupní abeceda, $\delta : Q \times X \rightarrow P(Q)$ přechodová funkce a S, F jsou množiny počátečních a koncových stavů.

2.1 Převod NKA na KA

Věta Je-li A nedeterministický konečný automat, potom lze sestavit konečný automat B takový, že $L(A) = L(B)$.

Důkaz Podmnožinová konstrukce: budeme simulovat všechny možné cesty NKA a zapisovat je do KA. Protože těch může být pouze konečně mnoho, bude i výsledný automat konečný. Navíc bude rozpoznávat stejný jazyk.

2.2 λ -přechod

Definice λ -přechod je hrana automatu, při které nedochází ke čtení vstupního symbolu.

3 Dvousměrné konečné automaty

Dvousměrným konečným automatem nazveme pěticí

$$A = (Q, X, \delta, q_0, F) \quad (1)$$

kde Q je množina stavů, X je vstupní abeceda, $\delta : Q \times X \rightarrow Q \times \{-1, 0, +1\}$ přechodová funkce, q_0 počáteční stav a F je množina koncových stavů.

3.1 Věta o síle dvousměrných automatů

Věta Jazyky přijímané dvousměrnými konečnými automaty jsou právě jazyky přijímané konečnými automaty.

Důkaz Je zřejmé, že KA je 2KA posouvající hlavu stále doprava. Důkaz obrácené implikace můžeme provést například pomocí Nerodovy věty. K tomu si ale musíme připravit podklad:

Zavedeme funkci f_u , která bude sledovat výpočet automatu nad slovem u . Mějme tedy slovo uv a definujeme funkci f následovně (a zavedeme stav q'_0 jako vstupní stav a definujeme $\delta(q'_0) = \delta(q_0)$ - to nezmění výpočet a zajistí nám to potřebné rozlišení prvního vstupu do výpočtu):

1. $f(q'_0) = p$ kde stav p je stav, kdy poprvé přejdeme hlavou ze slova u do v (nebo 0 pokud se tak nikdy nestane)
2. $f(p) = q$ kde stav q je stav, kdy přejdeme hlavou ze slova u do v pokud jsme vstoupili do slova u z v ve stavu p .

Všimneme si, že takto definovaná funkce pro každé slovo přesně udává postup výpočtu automatu. Zavedeme tedy ekvivalenci slov takto:

$$u \sim w \Leftrightarrow f_u = f_w \quad (1)$$

Tedy slova jsou ekvivalentní, pokud se nad nimi automat chová stejně. Nyní se podíváme na vlastnosti relace \sim :

1. Podle definice jde o ekvivalenci.
2. Má konečný index (počet takových funkcí je omezený, protože je omezení počet stavů).
3. Jde o pravou kongruenci: $u \sim w \Rightarrow uv \sim uw$ platí, protože rozhraní slov $u|v$ a $u|w$ je stejné (definované přímo funkcí f) a nad v je výpočet automatu stejný.
4. $L(A)$ je sjednocením jistých tříd rozkladu, protože $w \in L(A) \Leftrightarrow f_w(q'_0) \in F$

Tedy podle Nerodovy věty je $L(A)$ regulární jazyk a může být přijímán konečným automatem.

4 Množinové operace nad jazyky

4.1 Množinové operace nad regulárními jazyky

Definice Definujeme následující množinové operace nad jazyky:

1. **Sjednocení** $L_1 \cup L_2 = \{w | w \in L_1 \vee w \in L_2\}$
2. **Průnik** $L_1 \cap L_2 = \{w | w \in L_1 \wedge w \in L_2\}$
3. **Rozdíl** $L_1 - L_2 = \{w | w \in L_1 \wedge w \notin L_2\}$
4. **Doplňěk** $-L = \{w | w \notin L\} = X^* - L$

4.2 Uzavřenost množinových operací nad regulárními jazyky

Věta Sjednocení, průnik, rozdíl i doplněk jsou uzavřeny nad regulárními jazyky.

Důkaz Zkonstruujeme automat přijímající daný jazyk (v případě průniku a rozdílu simulujeme běh dvou automatů, tedy musíme rozšířit množinu stavů na $Q = Q_1 \times Q_2$ a patřičně rozšířit přechodovou funkci):

1. Sjednocení - použijeme nedeterminismus
2. Průnik - koncové stavy budou stavy $F = F_1 \times F_2$
3. Rozdíl - koncové stavy budou stavy $F = F_1 \times (Q_2 - F_2)$
4. Doplněk - prohodíme přijímající a nepřijímající stavy

4.3 Řetězcové operace nad regulárními jazyky

Definujeme následující řetězcové operace nad regulárními jazyky:

1. **Zřetězení** $L_1 \cdot L_2 = \{uv | u \in L_1 \wedge v \in L_2\}$
2. **Mocniny jazyka** $L^{i+1} = L^i \cdot L$
3. **Pozitivní iterace** $L^+ = L^1 \cup L^2 \cup \dots$
4. **Obecná iterace** $L^* = L^0 \cup L^+$
5. **Otočení jazyka** $L^R = \{u^R | u \in L\}$
6. **Levý kvocient** L_1 podle L_2 $L_2 \setminus L_1 = \{v | uv \in L_1 \wedge u \in L_2\}$
7. **Pravý kvocient** L_1 podle L_2 $L_1 / L_2 = \{u | uv \in L_1 \wedge v \in L_2\}$