Kapitola 1

Úvod

V přednášce se zaměříme hlavně na konečný popis obecně nekonečných množin řetězců symbolů dané množiny A. Prvkům množiny A budeme říkat písmena, řetězcům (konečným posloupnostem) písmen budeme říkat slova, množinám slov pak jazyky. Zaměříme se na dva základní typy popisu: rozpoznávání a generování. Pro rozpoznávání využijeme pojem automatu, pro generování pak pojem gramatiky.

Zhruba řečeno, automat postupně zpracovává vstupní slovo a na základě stavu, do kterého se dostane po přečtení celého slova, rozhodne, zda slovo bylo přijato (do jazyka patří) nebo nebylo přijato (do jazyka nepatří). S formální definicí automatu se setkáme v první části.

Naproti tomu gramatika je vlastně soupis pravidel, jak „vygenerovat“ všechna slova jazyka. Jako příklad jazyka, pro který máme gramatiku, která jej generuje, uveďme např. správně utvořené algebraické výrazy, nebo správně napsané programy v daném programovacím jazyce. Gramatikám (jako silnějšímu nástroji než je jen konečný automat) se věnujeme v druhé polovině přednášky.

Dříve než přistoupíme ke studiu konečných automatů, připomeneme základní pojmy týkající se jazyků.

1.1 Jazyky

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

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

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

1.1.3 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 slova b je 1, délka prázdného slova ε je 0.

V dalším textu používáme ještě následující značení: pro slovo u nad abecedou Σ a $c \in \Sigma$ symbol $|u|_c$ označuje počet výskytů symbolu c ve slově u. Tedy např. pro $u = aab$ je $|u|_a = 2$, $|u|_b = 1$.

1.1.4 Zřetězení slov. Je dána abeceda Σ. Pro dvě slova u, v nad abecedou Σ definujeme operaci *zřetězení* takto: Je-li $u = a_1 a_2 \dots a_n$ a $v = b_1 b_2 \dots b_k$, pak

$$u \cdot v = a_1 a_2 \dots a_n b_1 b_2 \dots b_k.$$

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

1.1.5 Tvrzení. Zřetězení slov je asociativní operace na množině všech slov nad danou abecedou. □

1.1.6 Poznámka. Platí, že každé slovo $u = a_1a_2 \dots a_n$ nad abecedou Σ, vzniklo (postupným) zřetězením slov délky 1, tj.

$$u = a_1 \cdot a_2 \cdot \ldots \cdot a_n.$$

Proto se někdy místo slovo nad abecedou používá termín *řetězec.*

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

Označíme Σ^+ množinu všech neprázdných slov nad abecedou Σ. (Tj. $\Sigma^+ = \Sigma^\star \setminus \{\varepsilon\}$.) Množina Σ^+ spolu s operací zřetězení tvoří pologrupu. □

1.1.8 Vlastnosti operace zřetězení na $\Sigma^\star$ a Σ^+.

1. Zřetězení slov není komutativní. Např. pro $u = aab$ a $v = b$ je $uv = aabb$, ale $vu = baab$.
2. Pro libovolná slova u a v nad stejnou abecedou platí:

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

1.1.9 Mocniny. Je-li u slovo nad abecedou Σ, pak definujeme mocniny slova u takto:

$$u^0 = \varepsilon, \qquad u^{i+1} = uu^i \text{ pro každé } i.$$

□

1.1.10 Podslovo, prefix, sufix. Je dáno slovo u. Řekneme, že slovo w je *podslovem* slova u, jestliže existují slova x, y taková, že

$$u = xwy.$$

Slova x, y mohou být i prázdná. Speciálně, slovo u je podslovem sebe sama; ano $u = \varepsilon u \varepsilon$.

Jestliže

$$u = wy,$$

řekneme, že slovo w je *prefix slova u.*

Jestliže

$$u = xw,$$

řekneme, že slovo w je *sufix slova u.*

Jestliže neprázdné slovo w, $w \neq u$, je podslovo (resp. prefix, resp. sufix) slova u, pak w se nazývá *vlastní podslovo, (resp. vlastní prefix, resp. vlastní sufix*) slova u. □

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

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

Kapitola 2

Konečné automaty

2.1 Deterministické konečné automaty

Konečné automaty se používají v různých oborech. Jako příklady můžeme uvést překladače, dále se používají při zpracování přirozeného jazyka, při návrzích hardwaru, i dalších. M zde nejprve vysvětlíme konečný automat neformálně, poté ukážeme několik příkladů a teprve poté zavedeme formální definici konečného automatu.

2.1.1 Konečný automat je abstraktní zařízení, které si pamatuje svůj stav a reaguje na vnější podněty změnami svého stavu, přičemž všech možných stavů i všech možných podnětů je konečně mnoho. Některé druhy automatů navíc produkují výstup.

Ze všech zařízení, která jsou schopna si něco pamatovat (tj. u nichž budoucí činnost může být ovlivněna historií), jsou konečné automaty nejjednodušší možné.

Jedním z cílů teorie automatů je zjistit, jaké jsou hranice toho, co konečné automaty dovedou a co už naopak je mimo jejich možnosti.

2.1.2 Množina stavů. Konečný automat se v každém okamžiku nachází v přesně jednom *stavu*, který je prvkem konečné množiny stavů (obvykle značené písmenem Q). Podstatné je, že kromě svého stavu si konečný automat vůbec nic nepamatuje (tj. nepamatuje si, jak se do tohoto stavu dostal).

Můžeme to formulovat i takto: Budoucí činnost automatu, tedy to, jak bude automat reagovat na příští vstupní podněty, obecně závisí na dosavadní historii, tedy na tom, co se s automatem dělo v minulosti, ale informace o celé předchozí historii je v konečném automatu zkoncentrována do jediného stavu, ve kterém se teď automat nachází.

2.1.3 Vstupní abeceda. Podněty, na které konečný automat reaguje, formalizujeme jako tzv. vstupní symboly, vstupní písmena, která jsou prvky tzv. vstupní abecedy obvykle značené Σ.

Při praktickém použití může být vstupním symbolem např. znak přečtený ze vstupního souboru, příchod paketu v počítačové síti, nebo stisk tlačítka nějakého fyzického zařízení.

2.1.4 Změny stavů. V deterministickém konečném automatu je změna stavu jednoznačně určena dosavadním stavem a vstupním symbolem, což lze formalizovat tzv. *přechodovou funkcí* $\delta : Q \times \Sigma \to Q$, která každému stavu $q \in Q$ a každému vstupnímu symbolu $x \in \Sigma$ přiřazuje nový stav.

Změna stavu se děje v okamžiku příchodu vstupního symbolu. Mezi příchody vstupních symbolů a jimi vyvolanými změnami stavů se v automatu nic neděje.

2.1.5 Výstup. Výstup konečného automatu může záviset na dvojici stav, ve kterém se automat nachází, a vstup podnět, na který automat reaguje. V takovém případě mluvíme o *Mealyho automatu*. Nebo výstup závisí pouze na stavu, ve kterém se automat nachází — pak mluvíme o *Mooreově automatu*. Jestliže výstupem Mooreova automatu je „stav je koncový“ nebo „stav není koncový“, můžeme rozdělit stavy na „koncové“, tj. „přijímající“, a „nekoncové“, tj. „nepřijímající“, a hledat takové posloupnosti podnětů, po kterých automat skončí v koncovém stavu. Poslednímu typu konečného automatu budeme říkat deterministický konečný automat a zkracovat to na DFA (Deterministic Finite Automaton). Ve starší literatuře najdete i pojem „akceptor“.

2.1.6 Příklad 1 – automat na kávu. 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.

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

	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

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 vstupem a je dvojice (nový stav, výstup). (K znamená kávu, číslo udává vrácené peníze.)

2.1.7 Příklad 2 – posuvný registr. Stavem je uspořádaná k-tice naposled přečtených symbolů. Přečtením dalšího symbolu přejdeme do nové k-tice, tj. do nového stavu.

Na přednášce ukážeme posuvný registr, který realizuje celočíselné dělení čtyřmi daného čísla v binárním zápise čteném počínaje nejvyšším řádem.

2.1.8 Příklad 3. Zjistit, zda se v daném slovu u (textu) vyskytuje podslovo aab (šablona), lze konečným automatem.

V tomto příkladě máme 4 stavy

- q_0 buď jsme ještě nečetli žádný symbol nebo poslední čtený symbol nebyl a a přečtená část neobsahovala slovo aab jako podslovo;
- q_1 přečtená část končí jedním a, ale ne dvěma a a přečtená část neobsahuje slovo aab jako podslovo;
- q_2 přečtená část končí aa a neobsahuje slovo aab jako podslovo;
- q_3 přečtená část slova obsahuje podslovo aab.

Stav q_3 je koncovým (přijímajícím) stavem.

2.1.9 Základní typy automatů. Obecně rozlišujeme čtyři typy automatů: Mealyho automat, Mooreův automat, deterministický konečný automat DFA (akceptor) a automat bez výstupu. Pro popis jazyků se budeme zabývat hlavně deterministickými konečnými automaty – DFA.

Mealyho automat je šestice $(Q, \Sigma, Y, \delta, q_0, \lambda)$. kde

- Q je konečná množina stavů,
- Σ je konečná neprázdná množina vstupních symbolů,
- Y je konečná neprázdná množina výstupních symbolů,
- q_0 je počáteční stav,
- δ je přechodová funkce, tj. zobrazení $\delta\colon Q \times \Sigma \to Q$,
- λ je výstupní funkce, tj. zobrazení $\lambda\colon Q \times \Sigma \to Y$. □

Příkladem Mealyho automatu je např. automat na kávu 2.1.6.

Mooreův automat je šestice $(Q, \Sigma, Y, \delta, q_0, \beta)$. kde Q, Σ, Y, δ a q_0 mají stejný význam v případě Mealyho automatu a β je značkovací funkce, tj. zobrazení $\beta\colon Q \to Y$. □

Příkladem Mooreova automatu je např. posuvný registr 2.1.7.

DFA, též akceptor, je je pětice $(Q, \Sigma, \delta, q_0, F)$, kde Q, Σ, δ a q_0 mají stejný význam jako v případě Mooreova automatu a $F \subseteq Q$ je množina koncových (též přijímajících) stavů. □
Příkladem DFA je např. automat z 2.1.8.

Poznámka. DFA je vlastně 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.

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

2.1.10 Stavový diagram. Kromě tabulky můžeme konečný automat zadat též stavovým diagramem.

Je dán konečný automat s množinou stavů Q, množinou vstupních symbolů Σ a přechodovou funkcí δ. *Stavovým diagramem* nazýváme orientovaný ohodnocený graf, jehož vrcholy jsou stavy automatu (tj. $V = Q$) a orientovaná hrana vede z vrcholu q do vrcholu p právě tehdy, když $\delta(q, a) = p$. Taková hrana je ohodnocena a, je-li automat DFA nebo Mooreův; a je ohodnocená dvojicí $a/\lambda(q, a)$, je-li automat Mealyho.

Jestliže se jedná o Mooreův automat, vrcholy stavového diagramu jsou navíc ohodnoceny hodnotou 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.

2.1.11 Rozšířená přechodová funkce popisuje chování automatu nejen nad jedním vstupním symbolem, ale nad celým vstupním slovem. Definujeme ji (jako řadu pojmů teorie automatů) induktivně.

Definice. Je dán automat s množinou stavů Q, vstupní abecedou Σ a přechodovou funkcí δ. *Rozšířená přechodová funkce* $\delta^\star \colon Q \times \Sigma^\star \to Q$ je definovaná induktivně takto:

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

□

2.1.12 Tvrzení. Je dán automat (Q, Σ, δ). Potom pro každý stav $q \in Q$ a každá slova $u, v \in \Sigma^\star$ platí

$$\delta^\star(q, uv) = \delta^\star(\delta^\star(q, u), v).$$

□

Toto tvrzení se dokáže např. indukcí podle $n = |u| + |v|$.

Neformální zdůvodnění. Představte si práci konečného automatu ve stavu q nad slovem $u = a_1 a_2 \ldots a_k$ jako sled ve stavovém diagramu, který začíná ve stavu (vrcholu) q a je ohodnocen postupně a_1, a_2, …, a_k. Tento sled končí ve stavu $\delta^\star(q, a_1 \ldots a_k)$.

Přesněji: Označme $q = p_0$ (což je $\delta^\star(q, \varepsilon)$), $p_1 = \delta(p_0, a_1)$ (což je $\delta^\star(q, a_1)$), $p_2 = \delta(p_1, a_2)$ (což je $\delta^\star(q, a_1 a_2)$), …, $p_k = \delta(p_{k-1}, a_k)$ (což je $\delta^\star(q, a_1 \ldots a_k)$). Pak $p_0, p_1, \ldots, p_k$ je sled, který začíná ve stavu q, je označen slovem a_1, a_2, …, a_k a končí ve stavu $\delta^\star(q, a_1 \ldots, a_k)$.

Nyní je tvrzení 2.1.12 zřejmé, jedná se totiž o „navázání" dvou sledů – sledu z q do $\delta^\star(q, u)$ a sledu z $\delta^\star(q, u)$ do $\delta^\star(q, uv)$.

2.1.13 Poznámka. Na konečný automat se můžeme dívat ještě následujícím způsobem: Je dána vstupní páska, která na začátku obsahuje vstupní slovo. Dále je dána řídící jednotka, která pomocí čtecí hlavy čte (postupně odleva doprava) vstupní symboly. Automat na základě stavu, ve kterém se nachází řídící jednotka, a na základě vstupního symbolu, který čte hlava, změní stav a posune čtecí hlavu po vstupní pásce o jedno pole doprava. Jestliže v okamžiku, kdy čtecí hlava „přečte" celé vstupní slovo (hlava se dostane za poslední symbol vstupního slova), je řídící jednotka v koncovém stavu, automat slovo přijme, v opačném případě automat slovo nepřijme.

2.1.14 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, jestliže

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

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

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

□

2.1.15 Regulární jazyky jsou jazyky, které jsou přijímány některým DFA.

Definice. Jazyk L nazveme *regulární jazyk*, jestliže existuje DFA M takový, že $L = L(M)$. Třídu všech regulárních jazyků označujeme **Reg**. □