

# I. Matematické základy informatiky

Update: 7. května 2018

## 1 Konečné automaty, regulární výrazy, uzávěrové vlastnosti třídy regulárních jazyků.

### 1.1 Konečné automaty

**Konečný automat** (KA) tvoří množina stavů, vstupní abeceda, přechodová funkce, počáteční a koncové stavy. Můžeme jej znázornit jako **tabulku**, **graf** či **strom**.

Konečné automaty se dělí na **deterministické** a **nedeterministické**. Deterministický konečný automat má pouze jeden počáteční stav a přechodová funkce vrací jeden stav. Zatímco nedeterministický KA může mít více počátečních stavů a přechodová funkce vrací množinu stavů.

- **Slovo** přijaté automatem je taková sekvence symbolů (ze vstupní abecedy), pro kterou automat skončí v koncovém stavu.
- **Regulární jazyk** je takový jazyk (množina slov) který lze popsat konečným automatem.

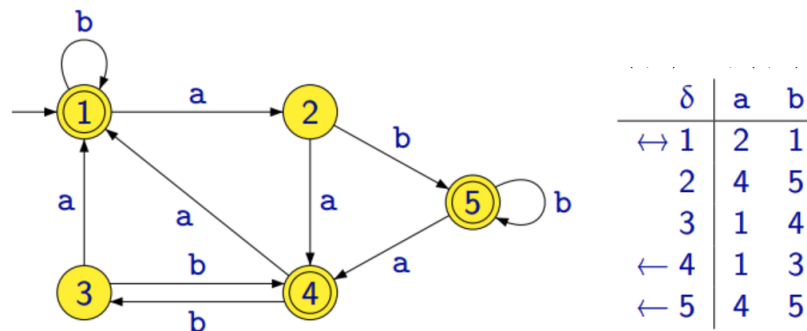
#### 1.1.1 Deterministický konečný automat (DKA)

Skládá se ze **stavů** a **přechodů**. Jeden ze stavů je označen jako **počáteční stav** a některé jsou označeny jako **přijímací**. Je definován jako **uspořádaná pětice**  $(Q, \Sigma, \delta, q_0, F)$ , kde:

- $Q$  je konečná neprázdná množina **stavů**.
- $\Sigma$  (*sigma*) je konečná neprázdná množina vstupních symbolů, tzv. **vstupní abeceda**.
- $\delta$  (*delta*) je **přechodová funkce**,  $\delta : Q \times \Sigma \rightarrow Q$ .
- $q_0$  je **počáteční stav**,  $q_0 \in Q$ .
- $F$  je neprázdná množina **koncových** neboli **přijímajících stavů**,  $F \subseteq Q$ .

#### Příklad

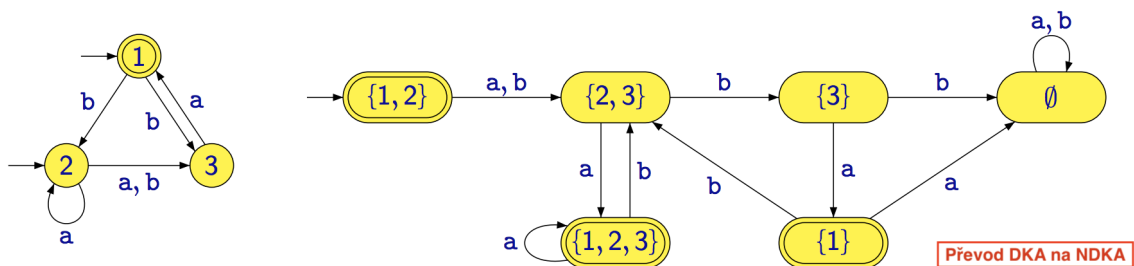
- $Q = \{1, 2, 3, 4, 5\}$ ,  $\Sigma = \{a, b\}$ ,  $F = \{1, 4, 5\}$
- $\delta(1, a) = 2$ ;  $\delta(1, b) = 1$ ;  $\delta(3, a) = 1$ ;  $\delta(3, b) = 4$ ;  $\delta(2, a) = 4$ ;  $\delta(2, b) = 5$ ;  $\delta(4, a) = 1$ ;  $\delta(4, b) = 3$ ;  $\delta(5, a) = 4$ ;  $\delta(5, b) = 5$



### 1.1.2 Nedeterministický konečný automat (NDKA)

Formálně je NDKA definován jako pětice  $A = (Q, \Sigma, \delta, I, F)$ , s tím rozdílem, že oproti deterministickému KA má **více počátečních stavů** a **přechodová funkce vrací množinu stavů**:

- $\delta$  je přechodová funkce, vrací množinu stavů,  $\delta : Q \times \Sigma \rightarrow P(Q)$ .
- $I$  je konečná množina počátečních stavů,  $I \in Q$ .



Na rozdíl od deterministického automatu:

- Může z jednoho stavu vést **libovolný počet přechodů** označených stejným symbolem (i nulové  $\epsilon$ ).
- Není zde nutné, aby z každého stavu vystupovaly všechny symboly, které do něj vstoupily  $\rightarrow$  **nemusí ošetřovat všechny varianty**, pouze odhadne, kterou cestou půjde.
- Nedeterministický automat přijímá dané slovo, jestliže **existuje alespoň jeden jeho výpočet**, který vede k přijetí tohoto slova.
- V automatu může být **víc než jeden počáteční stav**.
- Lze ho **převést na deterministický**. Při převodu automatu, který má  $n$  stavů může mít výsledný nedeterministický až  $2^n$  stavů.

### 1.1.3 Normovaný tvar

Začnu v počátečním stavu a procházím navštívené stavy a vytvářím tabulku. Každý KA má **právě 1** normovaný tvar. Také lze tímto způsobem zjistit, zda jsou automaty **ekvivalentní**.

## 1.2 Regulární výrazy

Regulární výraz je **řetězec popisující celou množinu řetězců**, konkrétně **regulární jazyk**. Regulární výrazy také můžeme chápat jako jednoduchý způsob, jak **popsat konečný automat** umožňující generovat všechna možná slova patřící do daného jazyka.

V regulárních výrazech využíváme znaky **abecedy** a symboly pro **sjednocení**, **zřetězení** a **iterace** regulárních výrazů. Za regulární výraz se považuje i samotný znak abecedy (např.  $a$ ) stejně jako **prázdné slovo**  $\epsilon$  a **prázdný jazyk**  $\emptyset$ .

### 1.2.1 Definice regulárních výrazů

Regulární výrazy popisují jazyky nad abecedou  $A = \Sigma : \emptyset, \epsilon, a$  (kde  $a \in \Sigma$ ) jsou regulární výrazy:

- $\emptyset$  označuje **prázdný jazyk**,
- $\epsilon$  označuje jazyk  $\{\epsilon\}$ ,
- $a$  označuje jazyk  $\{a\}$ .

Dále, jestliže  $\alpha, \beta$  jsou regulární výrazy, pak i  $(\alpha + \beta)$ ,  $(\alpha \cdot \beta)$ ,  $(\alpha^*)$  jsou regulární výrazy, kde:

- $(\alpha + \beta)$  označuje **sjednocení** jazyků označených  $\alpha$  a  $\beta$ ,
- $(\alpha \cdot \beta)$  označuje **zřetězení** jazyků označených  $\alpha$  a  $\beta$ ,
- $(\alpha^*)$  označuje **iteraci** jazyka označeného  $\alpha$ .

**Neexistují žádné další regulární výrazy** než ty definované podle předchozích dvou bodů.

### Příklady

Ve všech případech je  $\Sigma = \{0, 1\}$ :

- **01** (0 a 1) ... jazyk tvořený jedním slovem 01,
- **0+1** (0 nebo 1) ... jazyk tvořený dvěma slovy 0 a 1,
- **(01)\*** ... jazyk tvořený slovy  $\epsilon, 01, 0101, 010101, \dots$ ,
- **(0+1)\*** ... jazyk tvořený všemi slovy nad abecedou  $\{0, 1\}$ ,
- **(01)\*111(01)\*** ... jazyk tvořený všemi slovy obsahující podslovo 111, předcházení i následované libovolným počtem slov 01,
- **(0+1)\*00+(01)\*111(01)\*** ... jazyk tvořený všemi slovy, která buď končí 00 nebo obsahují podslovo 111 předcházené i následované libovolným počtem slov 01,
- **(0+1)\*1(0+1)\*** ... jazyk tvořený všemi slovy obsahujícími alespoň jeden symbol 1,
- **0\*(10\*10\*)\*** ... jazyk tvořený všemi slovy obsahujícími sudý počet symbolů 1.

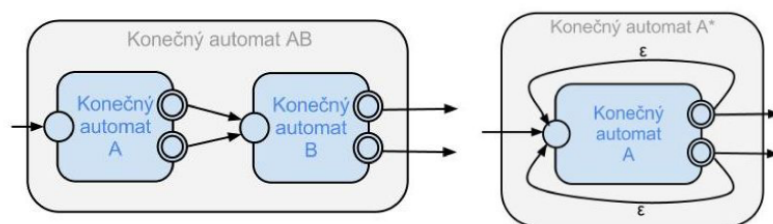
### 1.3 Uzávěrové vlastnosti třídy regulárních jazyků

**Uzavřenost množiny nad operací** znamená, že výsledek operace s libovolnými prvky z množiny bude opět spadat do dané množiny. Třidu regulárních jazyků značíme **REG**. Regulární výrazy (tedy i KA) jsou uzavřené vůči operacím:

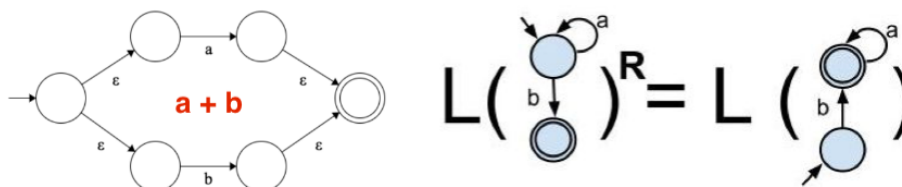
- **Sjednocení, průnik, doplněk** – je-li  $L_1, L_2 \in \text{REG}$ , pak také  $L_1 \cup L_2, L_1 \cap L_2, L_1'$  jsou v REG.
- **Zřetězení, iterace** – je-li  $L_1, L_2 \in \text{REG}$ , pak také  $L_1 \cdot L_2, L_1^*$  jsou v REG.
- **Zrcadlový obraz** – je-li  $L \in \text{REG}$ , pak také  $L^R$  jsou v REG.

#### 1.3.1 Operace sjednocení, zřetězení, iterace a zrcadlový obraz u KA

- **Iterace** – spojíme koncové stavy jednoho KA s počátečními druhého KA  $\epsilon$  přechodem. Na obrázku generuje automat  $A^*$  jazyk  $L(A^*) = L(A)^*$ , který je iterací jazyku generovaného modrého automatu  $A$ .
- **Zřetězení** – spojíme koncové stavy jednoho s počátečními stavy druhého. Na obrázku generuje konečný automat  $AB$  jazyk  $L(AB) = L(A) \cdot L(B)$ .



- **Sjednocení** –  $L(A + B) = L(A) + L(B)$  získáme tak, že vytvoříme **nový počáteční stav**, ze kterého vedeme  $\epsilon$  přechody do počátečních stavů obou automatů. Poté obdobě z koncových stavů obou automatů vedeme  $\epsilon$  přechody do **nového koncového**.
- **Zrcadlový obraz** – pustíme automat pozpátku, celý jej převrátíme. **Přehodíme orientaci všech přechodů**, z počátečních stavů uděláme koncové a naopak.



- **Doplněk** – u DKA provedeme prohození označení přijímajících a ostatních stavů, u NDKA je nejprve nutné provést převod na DKA.

- 2 Bezkontextové gramatiky a jazyky. Zásobníkové automaty, jejich vztah k bezkontextovým gramatikám.

- 3 Matematické modely algoritmů -Turingovy stroje a stroje RAM. Složitost algoritmu, asymptotické odhady. Algoritmicky nerozhodnutelné problémy.

#### 4 Třídy složitosti problémů. Třída PTIME a NPTIME, NP-úplné problémy.

- 5 Jazyk predikátové logiky prvního řádu. Práce s kvantifikátory a ekvivalentní transformace formulí.



- 6 Pojem relace, operace s relacemi, vlastnosti relací. Typy binárních relací. Relace ekvivalence a relace uspořádání.

- 7 Pojem operace a obecný pojem algebra. Algebry s jednou a dvěma binárními operacemi.

- 8 FCA – formální kontext, formální koncept, konceptuální svazy. Asociační pravidla, hledání často se opakujících množin položek.

## 9 Metrické a topologické prostory – metriky a podobnosti.

## 10 Shlukování.

- 11 Náhodná veličina. Základní typy náhodných veličin. Funkce určující rozdělení náhodných veličin.

- 12 Vybraná rozdělení diskrétní a spojité náhodné veličiny - binomické, hypergeometrické, negativně binomické, Poissonovo, exponenciální, Weibullovo, normální rozdělení.

### 13 Popisná statistika. Číselné charakteristiky a vizualizace kategoriálních a kvantitativních proměnných.



- 14 Metody statistické indukce. Intervalové odhady. Princip testování hypotéz. Okruhy pokývají předměty Teoretická informatika, Pravděpodobnost a statistika, Matematika pro zpracování znalostí