Regulárne jazyky

Definice 1.6 Gramatika G je čtveřice $G = (N,\ \Sigma,\ P,\ S)$, kde

1. N je konečná množina nonterminálních symbolů.
2. Σ je abeceda (tj. konečná množina terminálních symbolů), kde $N \cap \Sigma = \emptyset$.
3. P je konečná podmnožina kartézského součinu

$$(N \cup \Sigma)^* N (N \cup \Sigma)^* \times (N \cup \Sigma)^*$$

   nazývaná množina přepisovacích pravidel
4. $S \in N$ je výchozí (startovací) symbol gramatiky G.

❖ Prvek $(\alpha, \beta) \in P$ je přepisovací pravidlo a zapisuje se ve tvaru

$$\alpha \rightarrow \beta,$$

kde α je levá strana, β je pravá strana pravidla $\alpha \rightarrow \beta$.

Definice 1.7 Nechť $G = (N, \Sigma, P, S)$ je gramatika a nechť λ, μ jsou řetězce z $(N \cup \Sigma)^*$. Mezi λ a μ platí binární relace $\underset{G}{\Rightarrow}$, zvaná přímá derivace, můžeme-li řetězce λ a μ vyjádřit ve tvaru

$$\lambda = \gamma\alpha\delta$$
$$\mu = \gamma\beta\delta$$

$\gamma, \delta \in (N \cup \Sigma)^*$ a $\alpha \rightarrow \beta$ je nějaké přepisovací pravidlo z P. Pak píšeme

$$\lambda \underset{G}{\Rightarrow} \mu \text{ nebo}$$
$$\lambda \Rightarrow \mu.$$

Definice 1.8 Nechť $G = (N, \Sigma, P, S)$ je gramatika a $\Rightarrow$ relace přímé derivace na $(N \cup \Sigma)^*$. Relace $\overset{+}{\Rightarrow}$ označuje tranzitivní uzávěr relace $\Rightarrow$ a nazývá se relací derivace. Platí-li $\lambda \overset{+}{\Rightarrow} \mu$, pak existuje posloupnost

$$\lambda = \nu_0 \Rightarrow \nu_1 \Rightarrow \ldots \Rightarrow \nu_n = \mu,\ n \geq 1,$$

která se nazývá derivací délky n.

Definice 1.9 Nechť $G = (N, \Sigma, P, S)$ je gramatika. Řetězec $\alpha \in (N \cup \Sigma)^*$ nazýváme větnou formou, platí-li $S \stackrel{*}{\Rightarrow} \alpha$. Větná forma, která obsahuje pouze terminální symboly se nazývá věta.

Jazyk $L(G)$ generovaný gramatikou G je množina:

$$L(G) = \{w \in \Sigma^* \mid S \stackrel{*}{\Rightarrow} w\}$$

Definice 1.10 Jazyk generovaný gram. typu $i \in \{0, 1, 2, 3\}$, se nazývá jazykem typu i.

Existuje synonymní označení jazyků:

- Jazyk typu 0 — rekurzivně vyčíslitelný jazyk.
- Jazyk typu 1 — kontextový jazyk.
- Jazyk typu 2 — bezkontextový jazyk.
- Jazyk typu 3 — regulární jazyk.

Je definována na základě tvaru přepisovacích pravidel:

- Typ 0 – obecné (neomezené) gramatiky:

$$\alpha \to \beta \qquad \alpha \in (N \cup \Sigma)^* N (N \cup \Sigma)^*, \beta \in (N \cup \Sigma)^*$$

- Typ 1 – kontextové gramatiky:

$$\alpha A \beta \to \alpha \gamma \beta \qquad A \in N,\ \alpha, \beta \in (N \cup \Sigma)^*, \gamma \in (N \cup \Sigma)^+$$

nebo $S \to \varepsilon$, pakliže se S neobjevuje na pravé straně žádného pravidla

(Alternativní definice definující stejnou třídu jazyků: $\alpha \to \beta,\ |\alpha| \leq |\beta|$ nebo $S \to \varepsilon$ omezené jako výše.)

- Typ 2 – bezkontextové gramatiky:

$$A \to \alpha \qquad A \in N,\ \alpha \in (N \cup \Sigma)^*$$

- Typ 3 – regulární gramatiky:

$$A \to xB \qquad \text{nebo}$$
$$A \to x \mid \varepsilon \qquad A, B \in N,\ x \in \Sigma$$

(Alternativní tvary gramatik, které mají stejnou vyjadřovací sílu, jsou uvedeny v opoře.)

Definice 1.11 Konečným nedeterministický automatem (NKA) rozumíme automat M specifikovaný 5-ticí

$$M = (Q, \Sigma, \delta, q_0, F), \qquad \text{kde:}$$

1. Q je konečná množina stavů,
2. Σ je konečná vstupní abeceda,
3. δ je funkce přechodů (přechodová funkce) tvaru $\delta : Q \times \Sigma \to 2^Q$,
4. $q_0 \in Q$ je počáteční stav,
5. $F \subseteq Q$ je množina koncových stavů.

Je-li $\forall q \in Q \ \forall a \in \Sigma : |\delta(q, a)| \leq 1$, pak M nazýváme deterministickým konečným automatem (zkráceně DKA).

Deterministický konečným automat je také často specifikován 5-ticí

$$M = (Q, \Sigma, \delta, q_0, F), \qquad \text{kde:}$$

- δ je parciální funkce tvaru $\delta : Q \times \Sigma \to Q$,
- a význam ostatních složek zůstává zachován.

Je-li přechodová funkce δ totální, pak M nazýváme úplně definovaným deterministickým konečným automatem.

Definice 1.12 Nechť $M = (Q, \Sigma, \delta, q_0, F)$ je konečný automat (tj. NKA).

❖ Konfigurace C konečného automatu M je uspořádaná dvojice

$$C = (q, w), \ \ (q, w) \in Q \times \Sigma^*$$

kde q je aktuální stav, w je dosud nezpracovaná část vstupního řetězce.

❖ Počáteční konfigurace je konfigurace $(q_0, a_1 a_2 \ldots a_n)$.

❖ Koncová konfigurace je konfigurace $(q_F, \varepsilon), \ \ q_F \in F$.

❖ Přechodem automatu M rozumíme binární relaci $\underset{M}{\vdash}$ v množině konfigurací C

$$\underset{M}{\vdash} \subseteq (Q \times \Sigma^*) \times (Q \times \Sigma^*)$$

která je definována takto:

$$(q, w) \underset{M}{\vdash} (q', w') \overset{def.}{\Longleftrightarrow} w = aw' \ \wedge \ q' \in \delta(q, a) \ \ pro \ q, q' \in Q, a \in \Sigma, w, w' \in \Sigma^*$$

Definice 1.13 Rozšířený konečný automat (RKA) je pětice $M = (Q, \Sigma, \delta, q_0, F)$, kde

- Q je konečná množina stavů,
- Σ je konečná vstupní abeceda,
- δ je zobrazení $Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$,
- $q_0 \in Q$ je počáteční stav,
- $F \subseteq Q$ je množina koncových stavů.

Definice 1.14 Nechť Σ je konečná abeceda. Regulární množinu nad Σ definujeme rekurzívně takto:

1. $\emptyset$ (tj. prázdná množina) je regulární množina nad Σ,
2. $\{\varepsilon\}$ je regulární množina nad Σ,
3. $\{a\}$ je regulární množina nad Σ pro všechny $a \in \Sigma$,
4. jsou-li P a Q regulární množiny nad Σ, pak také
   (a) $P \cup Q$,
   (b) $P.Q$,
   (c) P^*
   jsou regulární množiny nad Σ.
5. Žádné jiné množiny, než ty, které lze získat pomocí výše uvedených pravidel, nejsou regulárními množinami.

Definice 1.15 Regulární výrazy nad Σ a regulární množiny, které označují, jsou rekurzívně definovány takto:

1. $\emptyset$ je regulární výraz označující regulární množinu $\emptyset$,
2. ε je regulární výraz označující regulární množinu $\{\varepsilon\}$,
3. a je regulární výraz označující regulární množinu $\{a\}$ pro všechny $a \in \Sigma$,
4. jsou-li p, q regulární výrazy označující regulární množiny P a Q, pak
   (a) $(p + q)$ je regulární výraz označující regulární množinu $P \cup Q$,
   (b) (pq) je regulární výraz označující regulární množinu $P.Q$,
   (c) (p^*) je regulární výraz označující regulární množinu P^*.
5. Žádné jiné regulární výrazy nad Σ neexistují.

Definice 1.18 Soustava rovnic nad reg. výrazy je ve standardním tvaru vzhledem k neznámým $\Delta = \{X_1, X_2, ..., X_n\}$, má-li soustava tvar

$$\bigwedge_{i \in \{1,...,n\}} X_i = \alpha_{i0} + \alpha_{i1} X_1 + \alpha_{i2} X_2 + ... + \alpha_{in} X_n$$

Definice 2.1 Nechť $M = (Q, \Sigma, \delta, q_0, F)$ je konečný automat. Stav $q \in Q$ nazveme dosažitelný, pokud existuje $w \in \Sigma^*$ takové, že $(q_0, w) \overset{*}{\underset{M}{\vdash}} (q, \varepsilon)$. Stav je nedosažitelný, pokud není dosažitelný.

Definice 2.2

- Nechť $M = (Q, \Sigma, \delta, q_0, F)$ je úplně definovaný DKA. Říkáme, že řetězec $w \in \Sigma^*$ rozlišuje q_1, q_2, jestliže $(q_1, w) \overset{*}{\underset{M}{\vdash}} (q_3, \varepsilon) \wedge (q_2, w) \overset{*}{\underset{M}{\vdash}} (q_4, \varepsilon)$ pro nějaké q_3, q_4 a *právě jeden* ze stavů q_3, q_4 je v F.
- Říkáme, že stavy $q_1, q_2 \in Q$ jsou k-nerozlišitelné a píšeme $q_1 \overset{k}{\equiv} q_2$, právě když neexistuje $w \in \Sigma^*$, $|w| \leq k$, který rozlišuje q_1 a q_2.
- Stavy q_1, q_2 jsou nerozlišitelné, značíme $q_1 \equiv q_2$, jsou-li pro každé $k \geq 0$ k-nerozlišitelné.

Definice 2.3 Úplně definovaný DKA M nazýváme redukovaný, jestliže žádný stav z Q není nedostupný a žádné dva stavy nerozlišitelné.

Definice 2.4 Nechť Σ je abeceda a $\sim$ je ekvivalence na Σ^*. Ekvivalence $\sim$ je pravou kongruencí (je zprava invariantní), pokud pro každé $u, v, w \in \Sigma^*$ platí

$$u \sim v \Longrightarrow uw \sim vw$$

Definice 2.5 Nechť L je libovolný (ne nutně regulární) jazyk nad abecedou Σ. Na množině Σ^* definujeme relaci $\sim_L$ zvanou prefixová ekvivalence pro L takto:

$$u \sim_L v \overset{def}{\Longleftrightarrow} \forall w \in \Sigma^* : uw \in L \Longleftrightarrow vw \in L$$

Bezkontextové jazyky

Definice 4.1 Gramatika $G = (N, \Sigma, P, S)$ si nazývá bezkontextovou gramatikou, jestliže všechna pravidla z P mají tvar

$$A \to \alpha, \quad A \in N, \quad \alpha \in (N \cup \Sigma)^*$$

Definice 4.2 Nechť δ je věta nebo větná forma generovaná v gramatice $G = (N, \Sigma, P, S)$ a nechť $S = v_0 \Rightarrow v_1 \Rightarrow \ldots \Rightarrow v_k = \delta$ její derivace v G. Derivační strom příslušející této derivaci je vrcholově ohodnocený strom s těmito vlastnostmi:

1. Vrcholy derivačního stromu jsou ohodnoceny symboly z množiny $N \cup \Sigma \cup \{\varepsilon\}$; kořen stromu je označen výchozím symbolem S.
2. Přímé derivaci $v_{i-1} \Rightarrow v_i, i = 0, 1, \ldots, k$ kde
   - $v_{i-1} = \mu A \lambda, \ \mu, \lambda \in (N \cup \Sigma)^*, \ A \in N$
   - $v_i = \mu \alpha \lambda$
   - $A \to \alpha, \ \alpha = X_1 \ldots X_n$ je pravidlo z P,

   odpovídá právě n hran $(A, X_j), j = 1, \ldots, n$ vycházejících z uzlu A, jež jsou uspořádány zleva doprava v pořadí $(A, X_1), (A, X_2), \ldots, (A, X_n)$.
3. Ohodnocení koncových uzlů derivačního stromu vytváří zleva doprava větnou formu nebo větu δ (plyne z 1. a 2.).

Definice 4.3 Nechť $S \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \alpha_n = \alpha$ je derivace větné formy α. Jestliže byl v každém řetězci $\alpha_i, i = 1, \ldots, n-1$ přepsán nejlevější (nejpravější) nonterminál, pak tuto derivaci nazýváme levou (pravou) derivací větné formy α.

Definice 4.4 Nechť G je gramatika. Říkáme, že věta w generovaná gramatikou G je víceznačná, existují-li alespoň dva různé derivační stromy s koncovými uzly tvořícími větu w. Gramatika G je víceznačná, pokud generuje alespoň jednu víceznačnou větu. V opačném případě mluvíme o jednoznačné gramatice.
Jazyky, které lze generovat víceznačnou gramatikou, ale které nelze generovat jednoznačnou gramatikou, se nazývají jazyky s inherentní víceznačností.

Definice 4.5 Zásobníkový automat P je n-tice $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

1. Q je konečná množina vnitřních stavů
2. Σ je konečná vstupní abeceda
3. Γ je konečná zásobníková abeceda
4. δ je přechodová funkce ve tvaru $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to 2^{Q \times \Gamma^*}$
5. $q_0 \in Q$ je počáteční stav
6. $Z_0 \in \Gamma$ je startovací symbol zásobníku
7. $F \subseteq Q$ je množina koncových stavů

Definice 4.6 Nechť $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je zásobníkový automat. Konfigurací automatu P nazveme trojici $(q, w, \alpha) \in Q \times \Sigma^* \times \Gamma^*$, kde

1. q je přítomný stav vnitřního řízení
2. w je dosud nezpracovaná část vstupního řetězce
3. α je obsah zásobníku ($\alpha = Z_{i_1} Z_{i_2} \dots Z_{i_k}$, Z_{i_1} je vrchol)

Přechod ZA P je binární relace $\vdash_P$ definovaná na množině konfigurací:

$$(q, w, \beta) \vdash_P (q', w', \beta') \stackrel{def}{\Longleftrightarrow} w = aw' \wedge \beta = Z\alpha \wedge \beta' = \gamma\alpha \wedge (q', \gamma) \in \delta(q, a, Z),$$

kde $q, q' \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, $w, w' \in \Sigma^*$, $Z \in \Gamma$ a $\alpha, \beta, \beta', \gamma \in \Gamma^*$.

- Je-li $a = \varepsilon$, pak odpovídající přechod nazýváme ε-přechodem.
- Relace $\vdash_P^i, \vdash_P^*, \vdash_P^+$ jsou definovány obvyklým způsobem.
- Platí-li pro řetězec $w \in \Sigma^*$ relace $(q_0, w, Z_0) \vdash_P^* (q, \varepsilon, \gamma)$, kde $q \in F$ a $\gamma \in \Gamma^*$, pak říkáme, že w je přijímán zásobníkovým automatem P (q_0, w, Z_0), resp. (q, ε, γ) je počáteční, resp. koncová konfigurace.
- Definujeme jazyk přijímaný zásobníkovým automatem P: $L(P) = \{w | (q_0, w, Z_0) \vdash_P^* (q, \varepsilon, \gamma) \wedge q \in F\}$.

Definice 4.7 Rozšířený zásobníkový automat P je sedmice $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde δ je přechodová funkce definovaná takto:

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^* \rightarrow 2^{Q \times \Gamma^*}$$

Definice 4.8 Zásobníkový automat nebo rozšířený zásobníkový automat $P = (Q, \Sigma, \Gamma, \delta, q_0, Z, \emptyset)$ přijímá s vyprázdněním zásobníku, pokud

$$L(P) = \{w \mid (q_0, w, Z_0) \vdash^* (q, \varepsilon, \varepsilon), q \in Q\}$$

Definice 4.9 Nechť G_1 a G_2 jsou gramatiky libovolného typu Chomského klasifikace. G_1 a G_2 jsou ekvivalentní, pokud $L(G_1) = L(G_2)$.

Definice 4.10 Nechť $G = (N, \Sigma, P, S)$ je gramatika a $X \in N \cup \Sigma$ symbol. Říkáme, že symbol X je nedostupný v G, jestliže v G neexistuje derivace $S \Rightarrow^* \alpha X \beta$ pro nějaké $\alpha, \beta \in (N \cup \Sigma)^*$. Symbol X nazýváme zbytečný v G, jestliže v G neexistuje derivace tvaru $S \Rightarrow^* \alpha X \beta \Rightarrow^* zxy$ pro nějaké $\alpha, \beta \in (N \cup \Sigma)^*$ a $zxy \in \Sigma^*$.

Definice 4.11 G je gramatikou bez ε-pravidel, jestliže neobsahuje žádné ε-pravidlo (pravidlo tvaru $A \rightarrow \varepsilon$), nebo, pokud $\varepsilon \in L(G)$, potom obsahuje jediné ε-pravidlo $S \rightarrow \varepsilon$ a S se pak nevyskytuje na pravé straně žádného přepisovacího pravidla.

Definice 4.12 Pravidlo tvaru $A \rightarrow B$, kde $A, B \in N$ nazýváme jednoduché pravidlo.

Definice 4.13 Nechť $G = (N, \Sigma, P, S)$ je gramatika, $A \in N$. Gramatika G obsahuje cyklus, jestliže $A \Rightarrow^+ A$.

Definice 4.14 Gramatika bez zbytečných symbolů, ε-pravidel a bez cyklů se nazývá vlastní gramatikou.

Definice 4.15 Bezkontextová gramatika $G = (N, \Sigma, P, S)$ je v Chomského normální formě, má-li každé pravidlo z P jeden z těchto tvarů:

1. $A \to BC$, kde $A, B, C \in N$
2. $A \to a$, kde $a \in \Sigma$
3. je-li $\varepsilon \in L(G)$, pak $S \to \varepsilon$ je jediné ε-pravidlo a S se nevyskytuje na pravé straně žádného přepisovacího pravidla.

Definice 4.16 Bezkontextová gramatika $G = (N, \Sigma, P, S)$ je v Greibachové normální formě, je-li G gramatikou bez ε-pravidel a každé pravidlo z P (vyjma případného pravidla $S \to \varepsilon$) má tvar:
$A \to a\alpha$, kde $a \in \Sigma, \alpha \in N^*$

Definice 6.1 Nechť $\mathcal{L}$ je třída jazyků a nechť $L \subseteq \Sigma^*$ je jazykem třídy $\mathcal{L}$. Dále nechť $\Sigma = \{a_1, a_2, ..., a_n\}$ pro nějaké $n \in \mathbb{N}$ a nechť jazyky označené $L_{a_1}, L_{a_2}, ..., L_{a_n}$ jsou rovněž jazyky třídy $\mathcal{L}$. Říkáme, že třída $\mathcal{L}$ je uzavřena vzhledem k substituci, jestliže pro každý výběr jazyků $L, L_{a_1}, L_{a_2}, ..., L_{a_n}$ je také jazyk $\sigma_{L_{a_1}, L_{a_2}, ..., L_{a_n}}(L)$

$$\sigma_{L_{a_1}, L_{a_2}, ..., L_{a_n}}(L) = \{x_1 x_2 ... x_m \mid b_1 b_2 ... b_m \in L \land \forall i \in \{1, ..., m\} : x_i \in L_{b_i}\}$$

ve třídě $\mathcal{L}$.

Definice 6.2 Nechť Σ a Δ jsou abecedy a $L \subseteq \Sigma^*$ je jazyk nad abecedou Σ.

- Zobrazení $h : \Sigma^* \to \Delta^*$ nazveme morfismem nad slovy, platí-li $\forall w = a_1 a_2 ... a_n \in \Sigma^* : h(w) = h(a_1)h(a_2)...h(a_n)$.
- Morfismus jazyka $h(L)$ pak definujeme jako $h(L) = \{h(w) \mid w \in L\}$.

Definice 6.3 Bezkontextová gramatika $G = (N, \Sigma, P, S)$ má vlastnost sebevložení, jestliže existují $A \in N$ a $u, v \in \Sigma^+$ takové, že $A \Rightarrow^+ uAv$ a A není zbytečný nonterminál. Bezkontextový jazyk má vlastnost sebevložení, jestliže každá gramatika, která jej generuje, má vlastnost sebevložení.

Definice 6.4 Zásobníkový automat $P = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ nazýváme deterministický zásobníkový automat (DZA), jestliže pro každé $q \in Q$ a $z \in \Gamma$ platí buď

- $\forall a \in \Sigma : |\delta(q, a, z)| \leq 1 \land \delta(q, \varepsilon, z) = \emptyset$, nebo
- $\forall a \in \Sigma : \delta(q, a, z) = \emptyset \land |\delta(q, \varepsilon, z)| \leq 1$.

Definice 6.5 Nechť $L = L(P)$, kde P je deterministický zásobníkový automat. Jazyk L se pak nazývá deterministickým bezkontextovým jazykem.

Turingove stroje

Definice 7.1 Turingův stroj (TS) je šestice tvaru $M = (Q, \Sigma, \Gamma, \delta, q_0, q_F)$, kde:

- Q je konečná množina vnitřních (řídicích) stavů,
- Σ je konečná množina symbolů nazývaná vstupní abeceda, $\Delta \notin \Sigma$,
- Γ je konečná množina symbolů, $\Sigma \subset \Gamma$, $\Delta \in \Gamma$, nazývaná pásková abeceda,
- parciální funkce $\delta : (Q \setminus \{q_F\}) \times \Gamma \to Q \times (\Gamma \cup \{L, R\})$, kde $L, R \notin \Gamma$, je přechodová funkce,
- q_0 je počáteční stav, $q_0 \in Q$ a
- q_F je koncový stav, $q_F \in Q$.

Definice 7.2

1. Řetězec $w \in \Sigma^*$ je přijat TS $M = (Q, \Sigma, \Gamma, \delta, q_0, q_F)$, jestliže M při aktivaci z počáteční konfigurace pásky $\underline{\Delta} w \Delta \ldots$ a počátečního stavu q_0 zastaví přechodem do koncového stavu q_F, tj. $(q_0, \Delta w \Delta^\omega, 0) \overset{*}{\underset{M}{\vdash}} (q_F, \gamma, n)$ pro nějaké $\gamma \in \Gamma^*$ a $n \in \mathbb{N}$.
2. Množinu $L(M) = \{w \mid w \text{ je přijat TS } M\} \subseteq \Sigma^*$ nazýváme jazyk přijímaný TS M.

Definice 7.3 Nedeterministický TS je šestice $M = (Q, \Sigma, \Gamma, \delta, q_0, q_F)$, kde význam jednotlivých složek je shodný s deterministickým TS až na δ, jež má tvar:

$$\delta : (Q \setminus \{q_F\}) \times \Gamma \longrightarrow 2^{Q \times (\Gamma \cup \{L, R\})}$$

Definice 7.4 Jazyk $L(M)$ NTS $M = (Q, \Sigma, \Gamma, \delta, q_0, q_F)$ je množina řetězců $w \in \Sigma^*$ takových, že M při aktivaci z q_0 při počátečním obsahu pásky $\underline{\Delta} w \Delta \ldots$ **může** zastavit přechodem do q_F.

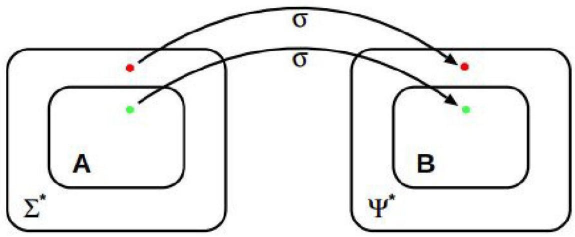
Definice 8.1 Jazyk $L \subseteq \Sigma^*$ se nazývá

- rekurzívně vyčíslitelný, jestliže $L = L(M)$ pro nějaký TS M,
- rekurzívní, jestliže $L = L(M)$ pro nějaký *úplný* TS M.

Definice 8.2 Nechť P je problém specifikovaný jazykem L_P nad Σ. Problém P nazveme:

- rozhodnutelný, pokud L_P je rekurzívní jazyk, tj. existuje TS, který L_P rozhoduje (přijme každý řetězec $w \in L_P$, a zamítne každý řetězec $w \in \Sigma^* \setminus L_P$),
- nerozhodnutelný, když není rozhodnutelný, a
- částečně rozhodnutelný, jestliže L_P je rekurzívně vyčíslitelný jazyk.

Definice 9.1 Nechť A, B jsou jazyky, $A \subseteq \Sigma^*$, $B \subseteq \Psi^*$. Redukce jazyka A na jazyk B je totální, rekurzívně vyčíslitelná funkce $\sigma : \Sigma^* \to \Psi^*$ taková, že
$\forall w \in \Sigma^*. w \in A \Leftrightarrow \sigma(w) \in B$.



❖ Existuje-li redukce jazyka A na B, říkáme, že A je redukovatelný na B, což značíme $A \leq B$.

Definice 9.2

- Postův systém nad abecedou Σ je dán neprázdným seznamem S dvojic neprázdných řetězců nad Σ, $S = \langle(\alpha_1, \beta_1), ..., (\alpha_k, \beta_k)\rangle$, $\alpha_i, \beta_i \in \Sigma^+$, $k \geq 1$.
- Řešením Postova systému je každá neprázdná posloupnost přirozených čísel $I = \langle i_1, i_2, ..., i_m\rangle$, $1 \leq i_j \leq k$, $m \geq 1$, taková, že:

  $$\alpha_{i_1}\alpha_{i_2}...\alpha_{i_m} = \beta_{i_1}\beta_{i_2}...\beta_{i_m}$$

  (Pozor: m není omezené a indexy se mohou opakovat!)
- Postův problém (PCP) zní: Existuje pro daný Postův systém řešení?

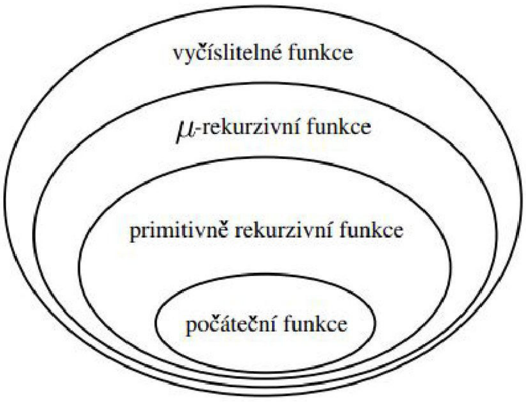
Definice 11.1 *Třída primitivních rekurzivních funkcí* obsahuje všechny funkce, které mohou být z počátečních funkcí vytvořeny:

(a) kombinací

(b) kompozicí

(c) primitivní rekurzí

Definice 11.3 Třída totálních vyčíslitelných funkcí se nazývá *μ-rekurzivní funkce*.



Definice 11.4 *Třída parciálně rekurzivních funkcí* je třída parciálních funkcí, které mohou být vytvořeny z počátečních funkcí aplikací:

(a) kombinace

(b) kompozice

(c) primitivní rekurze

(d) minimalizace

Definice 11.5 Turingův stroj $M = (Q, \Sigma, \Gamma, \delta, q_0, q_F)$ *vyčísluje (počítá)* parciální funkci $f : \Sigma^{*m} \to \Sigma_1^{*n}$, $\Sigma_1 \subseteq \Gamma, \Delta \notin \Sigma_1$, jestliže pro každé $(w_1, w_2, \ldots, w_m) \in \Sigma^{*m}$ a odpovídající počáteční konfiguraci $\underline{\Delta} w_1 \Delta w_2 \Delta \ldots \Delta w_m \Delta^\omega$ stroj M:

1. v případě, že $f(w_1, \ldots, w_m)$ je definována, pak M zastaví a páska obsahuje $\underline{\Delta} v_1 \Delta v_2 \Delta \ldots \Delta v_n \Delta\Delta\Delta$, kde $(v_1, v_2, \ldots, v_n) = f(w_1, \ldots, w_m)$
2. v případě, že $f(w_1, \ldots, w_m)$ není definována, M cykluje (nikdy nezastaví) nebo zastaví abnormálně.

Parciální funkce, kterou může počítat nějaký Turingův stroj se nazývá funkcí *Turingovsky vyčíslitelnou*.

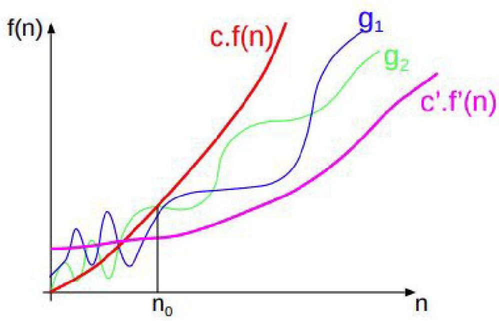
Zložitosti

Definice 12.1 Řekneme, že k-páskový DTS (resp. NTS) M přijímá jazyk L nad abecedou Σ v čase $T_M : \mathbb{N} \to \mathbb{N}$, jestliže $L = L(M)$ a M přijme (resp. *může* přijmout) každé $w \in L$ v nanejvýš $T_M(|w|)$ krocích.

Definice 12.2 Řekneme, že k-páskový DTS (resp. NTS) M přijímá jazyk L nad abecedou Σ v prostoru $S_M : \mathbb{N} \to \mathbb{N}$, jestliže $L = L(M)$ a M přijme (resp. *může* přijmout) každé $w \in L$ při použití nanejvýš $S_M(|w|)$ buněk pásky – *nepočítáme zde buňky pásky, na nichž je zapsán vstup, ale nikdy na nich nespočine hlava stroje.*

Definice 12.3 Nechť $\mathcal{F}$ je množina funkcí $f : \mathbb{N} \to \mathbb{N}$. Pro danou funkci $f \in \mathcal{F}$ definujeme množiny funkcí $O(f(n))$, $\Omega(f(n))$ a $\Theta(f(n))$ takto:

- Asymptotické horní omezení funkce $f(n)$ je množina $O(f(n)) = \{g(n) \in \mathcal{F} \mid \exists c \in \mathbb{R}^+ \, \exists n_0 \in \mathbb{N} \, \forall n \in \mathbb{N} : n \geq n_0 \Rightarrow 0 \leq g(n) \leq c.f(n)\}$.
- Asymptotické dolní omezení funkce $f(n)$ je množina $\Omega(f(n)) = \{g(n) \in \mathcal{F} \mid \exists c \in \mathbb{R}^+ \, \exists n_0 \in \mathbb{N} \, \forall n \in \mathbb{N} : n \geq n_0 \Rightarrow 0 \leq c.f(n) \leq g(n)\}$.
- Asymptotické oboustranné omezení funkce $f(n)$ je množina $\Theta(f(n)) = \{g(n) \in \mathcal{F} \mid \exists c_1, c_2 \in \mathbb{R}^+ \, \exists n_0 \in \mathbb{N} \, \forall n \in \mathbb{N} : n \geq n_0 \Rightarrow 0 \leq c_1.f(n) \leq g(n) \leq c_2.f(n)\}$.



Definice 12.4 Mějme dány funkce $t, s : \mathbb{N} \to \mathbb{N}$ a nechť T_M, resp. S_M, značí časovou, resp. prostorovou, složitost TS M. Definujeme následující časové a prostorové třídy složitosti deterministických a nedeterministických TS:

- $DTime[t(n)] = \{L \mid \exists$ k-páskový DTS $M : L = L(M)$ a $T_M \in O(t(n))\}$.
- $NTime[t(n)] = \{L \mid \exists$ k-páskový NTS $M : L = L(M)$ a $T_M \in O(t(n))\}$.
- $DSpace[s(n)] = \{L \mid \exists$ k-páskový DTS $M : L = L(M)$ a $S_M \in O(s(n))\}$.
- $NSpace[s(n)] = \{L \mid \exists$ k-páskový NTS $M : L = L(M)$ a $S_M \in O(s(n))\}$.

Definice 12.5 Funkci $t : \mathbb{N} \to \mathbb{N}$ nazveme časově zkonstruovatelnou (time constructible), jestliže existuje vícepáskový TS, jenž pro libovolný vstup w zastaví po přesně $t(|w|)$ krocích.

Definice 12.6 Funkci $s : \mathbb{N} \to \mathbb{N}$ nazveme prostorově zkonstruovatelnou (space constructible), jestliže existuje vícepáskový TS, jenž pro libovolný vstup w zastaví s využitím přesně $s(|w|)$ buněk pásky.

Definice 12.7 Nechť $\mathcal{R}$ je třída funkcí. Jazyk $L_1 \subseteq \Sigma_1^*$ je $\mathcal{R}$ redukovatelný (přesněji $\mathcal{R}$ many-to-one reducible) na jazyk $L_2 \subseteq \Sigma_2^*$, což zapisujeme $L_1 \leq_{\mathcal{R}}^m L_2$, jestliže existuje funkce f z $\mathcal{R}$ taková, že $\forall w \in \Sigma_1^* : w \in L_1 \Leftrightarrow f(w) \in L_2$.

Definice 12.8 Nechť $\mathcal{R}$ je třída funkcí a $\mathcal{C}$ třída jazyků. Jazyk L_0 je $\mathcal{C}$-těžký ($\mathcal{C}$-hard) vzhledem k $\mathcal{R}$ redukovatelnosti, jestliže $\forall L \in \mathcal{C} : L \leq_{\mathcal{R}}^m L_0$.

Definice 12.9 Nechť $\mathcal{R}$ je třída funkcí a $\mathcal{C}$ třída jazyků. Jazyk L_0 nazveme $\mathcal{C}$-úplný ($\mathcal{C}$-complete) vzhledem k $\mathcal{R}$ redukovatelnosti, jestliže $L_0 \in \mathcal{C}$ a L_0 je $\mathcal{C}$-těžký ($\mathcal{C}$-hard) vzhledem k $\mathcal{R}$ redukovatelnosti.

Definice 12.10 *Polynomiální redukce* jazyka L_1 nad abecedou Σ_1 na jazyk L_2 nad abecedou Σ_2 je funkce $f : \Sigma_1^* \to \Sigma_2^*$, pro kterou platí:

1. $\forall w \in \Sigma_1^* : w \in L_1 \Leftrightarrow f(w) \in L_2$
2. f je vyčíslitelná DTS v polynomiálním čase.

Existuje-li polynomiální redukce jazyka L_1 na L_2, říkáme, že L_1 se *(polynomiálně) redukuje na* L_2 a píšeme $L_1 \leq_P^m L_2$.