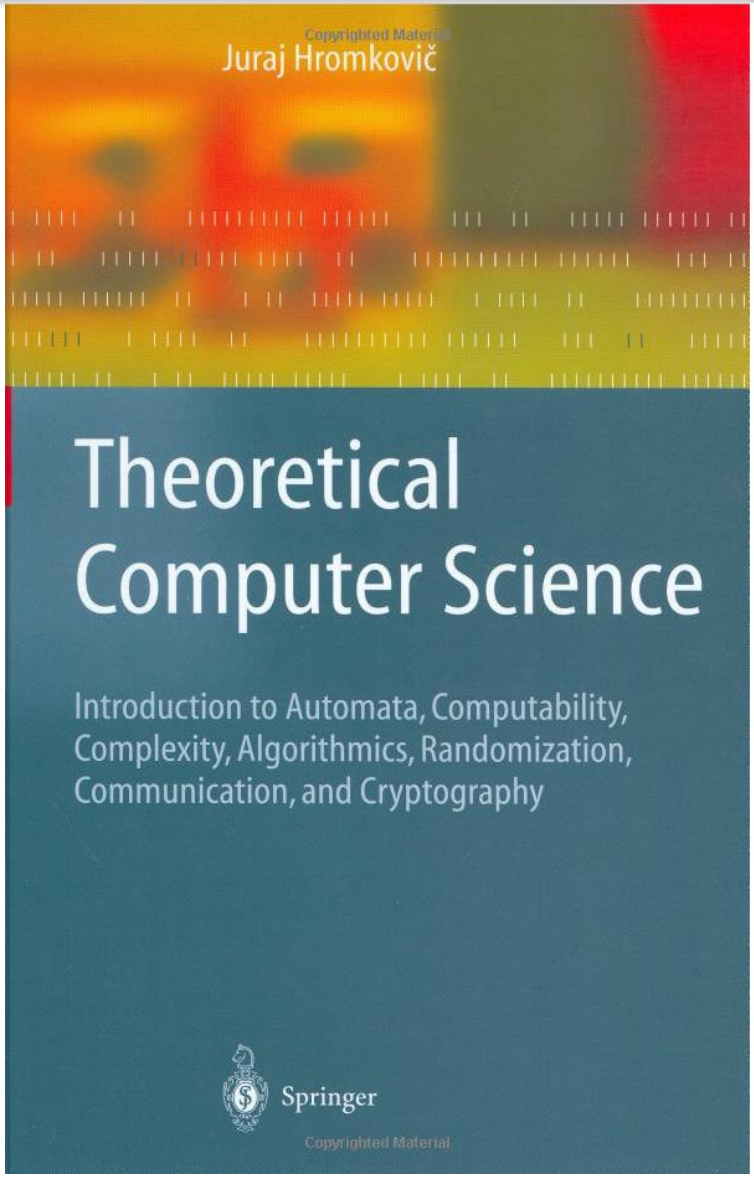
Úvod do teoretickej informatiky

(definície)

M. Winczer

Katedra základov a vyučovania informatiky
Fakulta matematiky, fyziky a informatiky
Univerzita Komenského

16. február 2026


Copyrighted Material
Juraj Hromkovič
Theoretical Computer Science
Introduction to Automata, Computability, Complexity, Algorithmics, Randomization, Communication, and Cryptography
Springer
Copyrighted Material

Pravidlá

- Priebežné hodnotenie tvoria dve písomky (31.3. a 19.5. 2026 o 18:00, posl. A,B. Nahradné písomky nebudú!). Každý študent musí v priebežnom hodnotení získať aspoň 27% bodov.
- Priemer známok z písomiek je dolná hranica známky.

Stránka predmetu:
`http://edu.fmph.uniba.sk/~winczer/uti.html`

Abecedy, Slová, Jazyky

Definícia 2.1 Neprázdna konečná množina sa nazýva **abeceda**. Prvky abecedy Σ sa nazývajú **symboly** abecedy Σ.

Definícia 2.2 Nech je Σ abeceda. Konečnú postupnosť symbolov zo Σ nazývame **slovo** nad abecedou Σ. **Prázdne slovo** λ je jediné slovo, ktoré má nula symbolov (niekde sa označuje aj ϵ).
Počet všetkých symbolov slova w nad abecedou Σ voláme **dĺžka slova** w a označujeme $|w|$. (Teda dĺžka postupnosti w.)

Abecedy, Slová, Jazyky

Množinu všetkých slov nad Σ označujeme Σ^*
Množinu všetkých slov nad Σ bez prázdneho slova označujeme $\Sigma^+ = \Sigma^* - \{\lambda\}$.
Dohoda: V reprezentácii slov medzi symbolmi nepíšeme čiarky. Zapisujeme

$$x_1x_2 \ldots x_n \text{ namiesto } x_1, x_2, \ldots, x_n.$$

Abecedy, Slová, Jazyky

Definícia 2.9 Nech je Σ abeceda. Zobrazenie $K : \Sigma^* \times \Sigma^* \to \Sigma^*$, dané ako

$$K(x, y) = x \cdot y = xy,$$

pre všetky $x, y \in \Sigma^*$, voláme **zreťazenie** vzhľadom na Σ.

Definícia 2.12 Nech je Σ abeceda. Pre všetky $x \in \Sigma^*$ a ľubovoľné kladné celé číslo i definujeme i-tu mocninu x ako

$$x^i = xx^{i-1}, \text{ kde } x^0 = \lambda.$$

Abecedy, Slová, Jazyky

Definícia 2.13 Nech je Σ abeceda. Pre $v, w \in \Sigma^*$

- v je **podslovo** $w \Leftrightarrow \exists x, y \in \Sigma^* : w = xvy$.
- v je **sufix** $w \Leftrightarrow \exists x \in \Sigma^* : w = xv$.
- v je **prefix** $w \Leftrightarrow \exists y \in \Sigma^* : w = vy$.
- keď $v \neq \lambda$ je **vlastné** podslovo slova w resp. [sufix, prefix] vtedy a len vtedy, ak $v \neq w$ a v je podslovo [sufix, prefix] slova w.

Definícia 2.15 Nech je Σ abeceda, $x \in \Sigma^*$ a $a \in \Sigma$. Počet výskytov a v x označujeme $|x|_a$.
Pre každú množinu A, $|A|$ označuje kardinalitu A.
$\mathcal{P}(A) = \{S | S \subseteq A\}$.

Abecedy, Slová, Jazyky

Definícia 2.16 Nech $\Sigma = \{s_1, s_2, \ldots s_m\}, m \geq 1$ je abeceda a nech $s_1 < s_2 < \cdots < s_m$ je usporiadanie prvkov z Σ. **Kanonické usporiadanie** nad Σ^* je pre všetky $u, v \in \Sigma^*$ definované takto:

$$\begin{aligned} u < v \Leftrightarrow & (|u| < |v|) \\ & \vee \big(|u| = |v| \wedge u = xs_iu' \wedge v = xs_jv' \\ & \text{pre nejaké } x, u', v' \in \Sigma^* \text{ a } i < j\big). \end{aligned}$$

Abecedy, Slová, Jazyky

Definícia 2.17 ľubovoľnú podmnožinu Σ^* nazývame **jazyk** nad abecedou Σ. Doplnok L^C jazyka L vzhľadom na Σ je jazyk $\Sigma^* - L$.
$L_0 = \emptyset$ je **prázdny jazyk.**
$L_\lambda = \{\lambda\}$ je jazyk obsahujúci len prázdne slovo.
Nech sú L_1 a L_2 jazyky nad Σ. Potom

$$L_1 \cdot L_2 = L_1 L_2 = \{vw | v \in L_1 \text{ a } w \in L_2\}$$

je **zreťazenie jazykov** L_1 a L_2.

Abecedy, Slová, Jazyky

Nech je L jazyk nad Σ. Definujeme

$$L^0 = L_\lambda, \qquad \text{a} \qquad L^{i+1} = L^i \cdot L \text{ pre všetky } i \in \mathbb{N},$$

$$L^* = \bigcup_{i \in N} L^i, \qquad \text{a} \qquad L^+ = \bigcup_{i \in N - \{0\}} L^i = L \cdot L^*.$$

L^* sa volá **Kleeneho uzáver** jazyka L.

Abecedy, Slová, Jazyky

Definícia 2.27 Nech sú Σ_1 a Σ_2 dve ľubovoľné abecedy. Funkciu $h : \Sigma_1^* \to \Sigma_2^*$, ktorá spĺňa podmienky:

(i) $h(\lambda) = \lambda$ a

(ii) $h(uv) = h(u)h(v)$ pre všetky $u, v \in \Sigma_1^*$

nazývame **homomorfizmus**.
$\boldsymbol{h^{-1}}$ definujeme takto: $h^{-1}(y) = \{x \mid h(x) = y\}, x \in \Sigma_1^*, y \in \Sigma_2^*$.

Pre ľubovoľný jazyk $L \subseteq \Sigma_1^*$ definujeme

$$\boldsymbol{h(L)} = \{h(w) \mid w \in L\}.$$

Pre ľubovoľný jazyk $L \subseteq \Sigma_2^*$ definujeme

$$\boldsymbol{h^{-1}(L)} = \cup_{w \in L} h^{-1}(w).$$

Algoritmické problémy

Algoritmus je program, ktorý zastane na každom vstupe.

Program A je zobrazenie

$$A : \Sigma_1^* \to \Sigma_2^*$$

Definícia 2.32 Rozhodovací problém (Σ, L) je pre danú abecedu Σ a daný jazyk L rozhodnúť pre každé $x \in \Sigma^*$, či

$$x \in L \text{ alebo } x \notin L.$$

Algoritmus A **rieši** rozhodovací problém (Σ, L), ak pre všetky $x \in \Sigma^*$:

$$A(x) = \begin{cases} 1, & \text{ak } x \in L \\ 0, & \text{ak } x \notin L \end{cases}$$

Hovoríme, že A **rozpoznáva** L.

Algoritmické problémy

Ak existuje algoritmus, ktorý rozpoznáva jazyk L, hovoríme, že jazyk L je **rekurzívny**.

Definícia 2.37 Nech Σ a Γ sú abecedy. Hovoríme, že algoritmus A **počíta funkciu** $f : \Sigma^* \to \Gamma^*$, ak pre všetky $x \in \Sigma^*$

$$A(x) = f(x).$$

Definícia 2.38 Nech Σ a Γ sú abecedy a $R \subseteq \Sigma^* \times \Gamma^*$ je relácia na Σ^* a Γ^*. Algoritmus A **počíta** R (alebo A **rieši problém relácie** R), ak pre všetky $x \in \Sigma^*$

$$(x, A(x)) \in R.$$

Algoritmické problémy

Definícia 2.39 Optimalizačný problém je 6-tica $\mathcal{U} = (\Sigma_I, \Sigma_O, L, \mathcal{M}, cost, goal)$, kde

(i) Σ_I je abeceda, volá sa **vstupná abeceda**

(ii) Σ_O je abeceda, volá sa **výstupná abeceda**

(iii) $L \subseteq \Sigma_I^*$ je jazyk **prípustných vstupov** (slová zo zmysluplnou interpretáciou)
$x \in L$ sa nazýva **inštanciou problému $\mathcal{U}$**

(iv) $\mathcal{M}$ je funkcia z L do $\mathcal{P}(\Sigma_O^*)$, pre každé $x \in L$ voláme množinu $\mathcal{M}(x)$ **množina prípustných riešení pre x**

(v) $cost$ je funkcia, $cost : \bigcup_{x \in L}(\mathcal{M}(x) \times \{x\}) \to \mathbb{R}^+$, voláme ju **cenová funkcia**

Algoritmické problémy

(vi) $goal \in \{minimum, maximum\}$ je cieľ.

Prípustné riešenie $\alpha \in \mathcal{M}(x)$ sa nazýva **optimálne** pre inštanciu x problému U, keď

$$cost(\alpha, x) = \mathbf{Opt}_{\mathcal{U}}(x) = goal\{cost(\beta, x) \mid \beta \in \mathcal{M}(x)\}.$$

Hovoríme, že algoritmus A **rieši** problém $\mathcal{U}$, ak pre všetky $x \in L$

(i) $A(x) \in \mathcal{M}(x)$ ($A(x)$ je prípustné riešenie pre inštanciu x problému $\mathcal{U}$) a

(ii) $cost(A(x), x) = goal\{cost(\beta, x) \mid \beta \in \mathcal{M}(x)\}.$

Ak $goal = minimum$, $\mathcal{U}$ voláme **minimalizačný problém**.
Ak $goal = maximum$, $\mathcal{U}$ voláme **maximalizačný problém**.

Algoritmické problémy

Hovoríme, že optimalizačný problém $\mathcal{U}_1 = (\Sigma_I, \Sigma_O, L', \mathcal{M}, cost, goal)$ je **podproblém** optimalizačného problému $\mathcal{U}_2 = (\Sigma_I, \Sigma_O, L, \mathcal{M}, cost, goal)$, ak $L' \subseteq L$.

Definícia 2.47 Nech je Σ abeceda a $x \in \Sigma^*$. Hovoríme, že algoritmus **generuje** slovo x, ak je výstupom A slovo x pre vstup λ.

Definícia 2.48 Nech je Σ abeceda a $L \subseteq \Sigma^*$. Algoritmus A **počíta** L, ak pre každé kladné celé číslo n je výstupom A $x_1, x_2, \ldots, x_n$, pričom je to prvých n slov L usporiadaných kanonickým usporiadaním.

Konečné automaty

Definícia 3.1 (Deterministický) **konečný automat (KA)** je 5-tica $M = (Q, \Sigma, \delta, q_0, F)$, kde

(i) Q je konečná neprázdna množina **stavov**,

(ii) Σ je abeceda, **vstupná abeceda** M. (Prípustné vstupy sú všetky slová nad Σ.)

(iii) q_0 je počiatočný stav,

(iv) $F \subseteq Q$ je množina **akceptujúcich stavov** a

(v) δ je funkcia. $\delta : Q \times \Sigma \to Q$, voláme ju **prechodová funkcia** M. ($\delta(q, a) = p$ znamená, že ak M v stave q prečíta symbol a, zmení stav na p.)

Konfigurácia M je ľubovoľný prvok z $Q \times \Sigma^*$.
Konfiguráciu $(q_0, x) \in \{q_0\} \times \Sigma^*$ voláme **počiatočná konfigurácia M na x**.

Konečné automaty

Každá konfigurácia patriaca do $Q \times \{\lambda\}$ sa nazýva **koncová konfigurácia**

Krok M je relácia (na konfiguráciách):

$$\vdash_M \subseteq (Q \times \Sigma^*) \times (Q \times \Sigma^*),$$

definovaná takto:

$$(q, w) \vdash_M (p, x) \Leftrightarrow w = ax, a \in \Sigma \text{ a } \delta(q, a) = p.$$

Konečné automaty

Výpočet C stroja M je konečná postupnosť $C = C_0, C_1, \ldots, C_n$ konfigurácií takých, že $C_i \underset{M}{\vdash} C_{i+1}$ pre všetky $0 \leq i \leq n-1$.

C je **výpočet M na vstupe $x \in \Sigma^*$**, ak $C_0 = (q_0, x)$ a $C_n \in Q \times \{\lambda\}$.

Ak $C_n \in F \times \{\lambda\}$ hovoríme, že C je **akceptujúci výpočet** M na vstupe x alebo ekvivalentne **M akceptuje x**.

Ak $C_n \in (Q - F) \times \{\lambda\}$ hovoríme, že C je **zamietajúci výpočet** M na vstupe x alebo ekvivalentne **M zamieta x**.

Konečné automaty

Jazyk $L(M)$ akceptovaný M definujeme ako

$$\begin{aligned} L(M) &= \{w \in \Sigma^* \mid \text{výpočet } M \text{ na } w \text{ skončí} \\ &\qquad\qquad \text{v koncovej konfigurácii } (q, \lambda), q \in F\} \\ &= \{w \in \Sigma^* \mid M \text{ akceptuje } w\}. \end{aligned}$$

Trieda **regulárnych jazykov**

$$\mathcal{L}(KA) = \{L(M) \mid M \text{ je } KA\}$$

je trieda všetkých jazykov akceptovaných konečnými automatmi. Jazyk L z $\mathcal{L}(KA)$ je **regulárny**.

Konečné automaty

Definícia 3.2 Nech $M = (Q, \Sigma, \delta, q_0, F)$ je konečný automat.

Definujeme $\overset{*}{\underset{M}{\vdash}}$ ako tranzitívny a reflexívny uzáver relácie krok $\underset{M}{\vdash}$ stroja M t.j.

$(q, w) \overset{*}{\underset{M}{\vdash}} (p, u) \Leftrightarrow (q = p$ a $w = u)$ alebo $\exists k \in \mathbb{N} - \{0\}$ také, že

(i) $w = a_1 a_2 \ldots a_k u, a_i \in \Sigma$ pre $i = 1, 2, \ldots, k$ a

(ii) $\exists r_1, r_2, \ldots, r_{k-1} \in Q$ takých, že

$$(q, w) \underset{M}{\vdash} (r_1, a_2 \ldots a_k u) \underset{M}{\vdash} (r_2, a_3 \ldots a_k u) \underset{M}{\vdash} \cdots (r_{k-1}, a_k u) \underset{M}{\vdash} (p, u).$$

Definujeme $\hat{\delta} : Q \times \Sigma^* \to Q$ takto:

(i) $\hat{\delta}(q, \lambda) = q$ pre všetky $q \in Q$ a

(ii) $\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$ pre všetky $a \in \Sigma, w \in \Sigma^*, q \in Q$.

Konečné automaty

Každý automat rozdelí Σ^* do $|Q|$ tried

$$\boldsymbol{KL[p]} = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) = p\} = \{w \in \Sigma^* \mid (q_0, w) \overset{*}{\underset{M}{\vdash}} (p, \lambda)\}.$$

Je zrejmé, že pre všetky $p, q \in Q, p \neq q$ platí

$$\bigcup_{p \in Q} KL[p] = \Sigma^* \qquad \text{a} \qquad KL[p] \cap KL[q] = \emptyset.$$

Konečné automaty

Simulácia

Lema 3.10 (Simulácia) Nech Σ je abeceda a $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ a $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$ sú konečné automaty. Pre operáciu $\odot \in \{\cup, \cap, -\}$ existuje KA M taký, že

$$L(M) = L(M_1) \odot L(M_2).$$

$M = (\Sigma, Q, \delta, q_0, F_\odot)$, kde

(i) $Q = Q_1 \times Q_2$,

(ii) $q_0 = (q_{01}, q_{02})$,

(iii) pre všetky $q \in Q_1$, $p \in Q_2$ a $a \in \Sigma$ je
$\delta((q,p),a) = (\delta_1(q,a), \delta_2(p,a))$,

(iv) ak $\odot = \cup$, $F = F_1 \times Q_2 \cup Q_1 \times F_2$,
ak $\odot = \cap$, $F = F_1 \times F_2$,
ak $\odot = -$, $F = F_1 \times (Q_2 - F_2)$.

Konečné automaty

Dôkazy neexistencie

Lema 3.12 Majme konečný automat $A = (Q, \Sigma, \delta, q_0, F)$. Nech pre $x, y \in \Sigma^*, x \neq y$:

$$(q_0, x) \overset{*}{\underset{A}{\vdash}} (p, \lambda) \quad \text{a} \quad (q_0, y) \overset{*}{\underset{A}{\vdash}} (p, \lambda)$$

pre nejaké $p \in Q$. Potom pre ľubovoľné $z \in \Sigma^*$ existuje $r \in Q$ také, že $xz \in KL[r]$ aj $yz \in KL[r]$ a platí, že

$$xz \in L(A) \Leftrightarrow yz \in L(A).$$

Konečné automaty

Dôkazy neexistencie, pumpovacia lema pre regulárne jazyky

Lema 3.14 Pre každý regulárny jazyk L existuje konštanta $n_0 \in \mathbb{N}$ taká, že každé slovo $w \in \Sigma^*$, pre ktoré $|w| \geq n_0$, sa dá vyjadriť ako

$$w = yxz,$$

kde

(i) $|yx| \leq n_0$,

(ii) $|x| \geq 1$ a

(iii) buď $\{yx^kz \mid k \in \mathbb{N}\} \subseteq L$ alebo $\{yx^kz \mid k \in \mathbb{N}\} \cap L = \emptyset$.

Konečné automaty

Definícia 3.1 Nedeterministický konečný automat (NKA) je 5-tica $M = (Q, \Sigma, \delta, q_0, F)$, kde

(i) Q je konečná neprázdna množina **stavov**,

(ii) Σ je abeceda, **vstupná abeceda** M. (Prípustné vstupy sú všetky slová nad Σ.)

(iii) q_0 je počiatočný stav,

(iv) $F \subseteq Q$ je množina **akceptujúcich stavov** a

(v) δ je funkcia. $\delta : Q \times \Sigma \to \mathcal{P}(Q)$, voláme ju **prechodová funkcia** M.

Konfigurácia M je ľubovoľný prvok z $Q \times \Sigma^*$.
Konfiguráciu $(q_0, x) \in \{q_0\} \times \Sigma^*$ voláme **počiatočná konfigurácia M na x**.

Konečné automaty

Každá konfigurácia patriaca do $Q \times \{\lambda\}$ sa nazýva **koncová konfigurácia**

Krok M je relácia (na konfiguráciách):

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

definovaná takto:

$$(q, w) \underset{M}{\vdash} (p, x) \Leftrightarrow w = ax, a \in \Sigma \text{ a } p \in \delta(q, a).$$

Konečné automaty

Výpočet C stroja M je konečná postupnosť $C = C_0, C_1, \ldots, C_n$ konfigurácií takých, že $C_i \underset{M}{\vdash} C_{i+1}$ pre všetky $0 \leq i \leq n-1$.

Výpočet M na vstupe $x \in \Sigma^*$ je výpočet $C_0, C_1, \ldots, C_m$ stroja M, kde

(i) $C_0 = (q_0, x)$ a

(ii) buď $C_m \in Q \times \{\lambda\}$, alebo
$C_m = (q, ax)$, pre nejaké $a \in \Sigma$ a $q \in Q$ také, že $\delta(q, a) = \emptyset$.

Ak $C_m \in F \times \{\lambda\}$ hovoríme, že C je **akceptujúci výpočet** M na vstupe x. Ak existuje akceptujúci výpočet M na x hovoríme, že **M akceptuje slovo x**.

Konečné automaty

Jazyk $L(M)$ akceptovaný M definujeme ako

$$\begin{aligned} L(M) &= \{w \in \Sigma^* \mid \text{výpočet } M \text{ na } w \text{ skončí} \\ &\qquad\qquad \text{v koncovej konfigurácii } (q, \lambda), q \in F\} \\ &= \{w \in \Sigma^* \mid M \text{ akceptuje } w\}. \end{aligned}$$

Konečné automaty

$\overset{*}{\underset{M}{\vdash}}$ je tranzitívny a reflexívny uzáver relácie krok $\underset{M}{\vdash}$ stroja M.

Pre prechodovú funkciu δ definujeme $\hat{\delta} : Q \times \Sigma^* \to \mathcal{P}(Q)$ pre všetky $q \in Q$, $a \in \Sigma$, a $w \in \Sigma^*$ takto:

(i) $\hat{\delta}(q, \lambda) = \{q\}$,

(ii) $\hat{\delta}(q, wa) = \bigcup_{r \in \hat{\delta}(q,w)} \delta(r, a)$

$= \{p \mid \text{existuje } r \in \hat{\delta}(q, w) \text{ také, že } p \in \delta(r, a)\}$.

Konečné automaty

Simulácia NKA na DKA, veta 3.26

Pre NKA $M = \{Q, \Sigma, \delta_M, q_0, F\}$, vytvoríme DKA $A = (Q_A, \Sigma, \delta_A, q_{0A}, F_A)$

(i) $Q_A = \{\langle P \rangle \mid P \subseteq Q\}$,

(iii) $q_{0A} = \langle \{q_0\} \rangle$,

(iv) $F_A = \{\langle P \rangle \mid P \subseteq Q \text{ a } P \cap F \neq \emptyset\}$,

(v) $\delta_A : Q_A \times \Sigma \to Q_A$, pre všetky $\langle P \rangle \in Q_A$ a $a \in \Sigma$ definovaná ako

$$\delta_A(\langle P \rangle, a) = \left\langle \bigcup_{p \in P} \delta_M(p, a) \right\rangle .$$

Konečné automaty

Definícia Regulárny výraz E nad abecedou Σ je výraz vytvorený podľa nasledujúcich pravidiel a reprezentuje jazyk $L(E)$ definovaný takto

(i) $\emptyset$ je regulárny výraz a $L(\emptyset) = \emptyset$.

(ii) Pre ľubovoľné $a \in \Sigma \cup \{\lambda\}$ je a regulárny výraz a $L(a) = \{a\}$.

(iii) Ak sú E_1, E_2 regulárne výrazy, potom sú regulárne výrazy aj $(E_1 + E_2), (E_1 \cdot E_2), (E_1^*)$
a
$L((E_1 + E_2)) = L(E_1) \cup L(E_2)$,
$L((E_1 \cdot E_2)) = L(E_1) \cdot L(E_2)$, $L((E_1^*)) = (L(E_1))^*$.

(iv) iné regulárne výrazy nad Σ nie sú.

Konvencie pri zápise regulárnych výrazov:

(i) Priorita operátorov (od najvyššej) $*$, $\cdot$, $+$., t.j. netreba písať všade zátvorky.

(ii) Operátor zreťazenia sa vynecháva.

(iii) $\{w_1, \ldots, w_n\}$ označuje konečný jazyk obsahujúci slová $w_1, \ldots, w_n$.

Konečné automaty

Reg. výraz pre daný DKA

Nech DKA $A = (Q, \Sigma, \delta, q_0, F)$ a $Q = \{0, 1, \ldots, n\}$, $q_0 = 0$.
Pre $0 \leq i, j \leq n$, $-1 \leq k \leq n$ definujeme množinu $R^k_{i,j}$ takto

$$R^k_{i,j} = \{w \in \Sigma^* \mid \hat{\delta}(i, w) = j \text{ a } \hat{\delta}(i, u) \leq k$$
$$\text{pre všetky vlastné prefixy } u \text{ slova } w\}.$$

$R^k_{i,j}$ obsahuje len také slová, ktoré spôsobia prechod automatu A zo stavu i do stavu j a prechádzajú pri tom len cez stavy z množiny $\{0, 1, \ldots, k\}$.

$$L(A) = \bigcup_{j \in F} R^n_{0,j}$$

Konečné automaty

Reg. výraz pre daný DKA

{ dynamické programovanie }

$$R_{i,j}^{-1} = \{a \mid \delta(i,a) = j\} \cup \{\lambda \mid \text{ak } i = j\}$$

$$R_{i,j}^{k} = \begin{cases} \emptyset & \text{ak } k = -1, i \neq j, \delta(i,a) \neq j \\ \{a\} & \text{ak } k = -1, i \neq j, \delta(i,a) = j \\ \{\lambda\} & \text{ak } k = -1, i = j, \delta(i,a) \neq j \\ \{\lambda, a\} & \text{ak } k = -1, i = j, \delta(i,a) = j \\ R_{i,j}^{k-1} \cup R_{i,k}^{k-1} (R_{k,k}^{k-1})^* R_{k,j}^{k-1} & \text{ak } k > -1. \end{cases}$$

Turingov stroj

Definícia 4.1 Turingov stroj (**TS**) je sedmica $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$, kde

(i) Q je konečná množina, nazýva sa **množina stavov** M,

(ii) Σ je **vstupná abeceda**, $¢, \sqcup \notin \Sigma$,

(iii) Γ je abeceda, nazýva sa **pracovná abeceda**. $\Sigma \subseteq \Gamma$ a $¢, \sqcup \in \Gamma$,

(iv) $\delta : (Q - q_{\text{accept}}, q_{\text{reject}}) \times \Gamma \to Q \times \Gamma \times \{\text{L,R,N}\}$ je **prechodová funkcia** stroja **M**, pre ktorú platí pre všetky $q \in Q - q_{\text{accept}}, q_{\text{reject}}$, že $\delta(q, ¢) \in Q \times \{¢\} \times \{\text{R,N}\}$,

(v) q_0 je **počiatočný stav**,

(vi) $q_{\text{accept}} \in Q - \{q_{\text{reject}}\}$ je **akceptujúci stav**,

(vii) $q_{\text{reject}} \in Q - \{q_{\text{accept}}\}$ je **zamietajúci stav**.

Turingov stroj

Konfigurácia C stroja M je prvok
z **conf**$(M) = \{¢\} \cdot \Gamma^* \cdot Q \cdot \Gamma^+ \cup Q \cdot \{¢\} \cdot \Gamma^*$.

Počiatočná konfigurácia M na vstupnom slove x je $q_0¢x$.

Krok stroja M je relácia $\vdash_M$ na konfiguráciách

($\vdash_M \subseteq \text{conf}(M) \times \text{conf}(M)$)

(i) $x_1x_2 \ldots x_{i-1}qx_ix_{i+1} \ldots x_n \vdash_M x_1x_2 \ldots x_{i-1}pyx_{i+1} \ldots x_n$, ak
$\delta(q, x_i) = (p, y, \mathrm{N})$

(ii) $x_1x_2 \ldots x_{i-1}qx_ix_{i+1} \ldots x_n \vdash_M x_1x_2 \ldots x_{i-2}px_{i-1}yx_{i+1} \ldots x_n$, ak
$\delta(q, x_i) = (p, y, \mathrm{L})$

(iii) $x_1x_2 \ldots x_{i-1}qx_ix_{i+1} \ldots x_n \vdash_M x_1x_2 \ldots x_{i-1}ypx_{i+1} \ldots x_n$, ak
$\delta(q, x_i) = (p, y, \mathrm{R})$, $i < n$ a $x_1x_2 \ldots x_{n-1}qx_n \vdash_M x_1x_2 \ldots x_{n-1}yp\sqcup$,
ak $\delta(q, x_n) = (p, y, \mathrm{R})$.

Turingov stroj

Výpočet M je potenciálne nekonečná postupnosť konfigurácií $C_0, C_1, \ldots$ taká, že $C_i \underset{M}{\vdash} C_{i+1}$, pre $i = 0, 1, 2, \ldots$.

Ak $C_0 \underset{M}{\vdash} C_1 \underset{M}{\vdash} \cdots C_i$, $i \in \mathbb{N}$, tak píšeme $C_0 \overset{*}{\underset{M}{\vdash}} C_i$.

Výpočet M na vstupe x je výpočet, ktorý začína v počiatočnej konfigurácii $C_0 = q_0 ¢ x$ a je buď nekonečný, alebo končí v konfigurácii $w_1 q w_2$, kde $w_1, w_2 \in \Gamma^*$ a $q \in \{q_{\text{accept}}, q_{\text{reject}}\}$.

Výpočet M na x je **akceptujúci** ak končí v akceptujúcej konfigurácii $w_1 q_{\text{accept}} w_2$.
Výpočet M na x je **zamietajúci** ak končí v zamietajúcej konfigurácii $w_1 q_{\text{reject}} w_2$.

Turingov stroj

Ak je výpočet M na x akceptujúci(zamietajúci), hovoríme, že M **akceptuje(zamieta)** x.

Ak je výpočet M na x zamietajúci alebo nekonečný, hovoríme, že M **neakceptuje**.

Turingov stroj

Jazyk $L(M)$ akceptovaný M definujeme ako

$$\boldsymbol{L(M)} = \{w \in \Sigma^* \mid M \text{ akceptuje } w\}.$$

Hovoríme, že M počíta funkciu $F : \Sigma^* \to \Gamma^*$, keď pre všetky $x \in \Sigma^* : q_0 ¢ x \overset{*}{\underset{M}{\vdash}} q_{\text{accept}} ¢ F(x)$.

Turingov stroj

Jazyk L nazývame **rekurzívne vyčísliteľný** (recursively enumerable) ak existuje TS M taký, že $L = L(M)$.

Množina

$$\mathcal{L}_{\mathrm{RE}} = \{L(M) \mid M \text{ je TS}\}$$

sa nazýva **trieda rekurzívne vyčísliteľných jazykov**.

Jazyk $L \subseteq \Sigma^*$ sa nazýva **rekurzívny** (recursive), alebo rozhodovací problém (Σ, L) sa nazýva **rozhodnuteľný** (decidable), ak $L = T(M)$ pre TS M, taký, že pre všetky $x \in \Sigma^*$

(i) $q_0 ¢ x \overset{*}{\underset{M}{\vdash}} y q_{\mathrm{accept}} z, y, z \in \Gamma^*,$ keď $x \in L$ a

(ii) $q_0 ¢ x \overset{*}{\underset{M}{\vdash}} u q_{\mathrm{reject}} v, u, v \in \Gamma^*,$ keď $x \notin L$.

Ak platí (i) aj (ii) hovoríme, že M **zastane na každom vstupe**, alebo, že M **vždy zastane**.

Turingov stroj

{Turingov stroj, ktorý vždy zastane je formálny model "algoritmu".}

Množina

$$\mathcal{L}_{\mathrm{R}} = \{L(M) \mid M \text{ je TS, ktorý vždy zastane}\}$$

je trieda **rekurzívnych (algoritmicky rozpoznateľných) jazykov**

Funkcia $F : \Sigma_1^* \to \Sigma_2^*$, kde Σ_1, Σ_2 sú abecedy, sa nazýva **vypočítateľná** (computable), ak existuje TS M, ktorý počíta F.

Turingov stroj

k-páskový Turingov stroj (**TS**) dostaneme z obyčajného TS zmenou niektorých jeho častí:

(i) vstupná páska je len **na čítanie**, ľavý okraj vstupnej pásky označuje symbol ¢ a pravý okraj špeciálny symbol $\$$. Na vstupnej páske sa hlava môže pohybovať oboma smermi, ale nemôže sa posunúť pred ¢ ani za $\$$.

(ii) má k **pracovných pások**. Každá pracovná páska má svoju hlavu na čítanie a zápis, na ľavom okraji pásky je symbol ¢, pred ktorý sa hlava nemôže dostať. Pásky obsahujú na začiatku výpočtu len symboly $\sqcup$. Hlavy na páskach sa môžu hýbať vpred aj vzad. Políčko pracovnej pásky je symbol z Γ. Pracovné pásky sú očíslované od 1 po k.

Turingov stroj

Konfigurácia $(q, w, i, u_1, i_1, u_2, i_2, \ldots, u_k, i_k)$ je prvok z $Q \times \Sigma^* \times N \times (\Gamma^* \times N)^k$.
(je tam stav q, vstupné slovo w, pozícia i čítacej hlavy na vstupnom slove, a k dvojíc – pracovná páska u_j a pozícia i_j hlavy na nej) Keď $w = a_1 a_2 \ldots a_n, a_i \in \Sigma$, čítacia hlava číta symbol a_i.

Pre $1 \leq j \leq k$ je obsah pracovnej pásky ¢u_j a platí, že $i_j < |u_j|$.

V **počiatočnej konfigurácii** na vstupnom slove w je na vstupnej páske ¢w\$, číta sa 1. symbol w, pracovné pásky obsahujú ¢␣ . . . , hlavy čítajú symbol ¢.

Turingov stroj

Krok stroja opisuje prechodová funkcia
$\delta : Q \times (\Sigma \cup \{¢, \$\}) \times \Gamma^k \to Q \times \{L, R, N\} \times (\Gamma \times \{L, R, N\})^k$
Argument $(q, a, b_1, \ldots, b_k)$

(i) q je momentálny stav.
(ii) $a \in \Sigma \cup \{¢, \$\}$ je symbol, ktorý číta hlava na vstupnej páske.
(iii) $b_1, \ldots, b_k$ sú symboly, ktoré čítajú hlavy na pracovných páskach.

Vykonanie kroku zahŕňa:

(i) prepísanie každého z k symbolov $b_1, \ldots, b_k$ na pracovných páskach,
(ii) zmenu stavu stroja
(iii) eventuálne posunutie všetkých $k + 1$ hláv tak, že žiadna nevyjde mimo pásky

Turingov stroj

Ak výpočet stroja M na w príde do stavu q_{accept}, potom M akceptuje w. Ak príde do q_{reject}, alebo sa nezastaví, potom M slovo w neakceptuje.

Poznámka: Uvedomte si, že v našom označení sú obyčajný TS a jednopáskový TS dva rôzne stroje. Obyčajný ma jedinú pásku, kde je na začiatku vstup a stroj ju používa na čítanie aj zápis. Jednopáskový (k-páskový) TS má jednu pásku len vstupnú (je na nej vstup a dá sa len čítať) a jednu (k) pracovnú(ých) pások, na ktoré sa dá písať a dajú sa aj čítať.

Turingov stroj

Ekvivalentnosť strojov

Nech sú A a B dva stroje pracujúce nad rovnakou abecedou Σ. A **je ekvivalentné** B, keď pre každý vstup $x \in \Sigma^*$ platí

(i) A akceptuje x práve vtedy keď B akceptuje x,

(ii) A zamieta x práve vtedy keď B zamieta x,

(iii) výpočet A na x je nekonečný práve vtedy keď výpočet B na x je nekonečný.

Poznámka: Ekvivalencia A a B implikuje, že $L(A) = L(B)$. Ale platnosť $L(A) = L(B)$ neimplikuje ekvivalenciu A a B.

Turingov stroj

k k-páskovému TS existuje ekvivalentný obyčajný TS

Označme si k-páskový Turingov stroj TS A a obyčajný TS B. B bude simulovať prácu A. Jeden spôsob ako realizovať simuláciu je, že konfigurácia B bude reprezentovať konfiguráciu A (jedna ku jednej)
k-páskový TS A má pracovnú abecedu Γ_A. Pracovnú abecedu Γ_B TS B definujeme takto:

$$\Gamma_B = (\Sigma \cup \{¢, \$, \sqcup\}) \times \{\sqcup, \uparrow\} \times (\Gamma_A \times \{\sqcup, \uparrow\})^k \cup \Sigma \cup \{¢\}$$

$$a = (a_0, a_1, \ldots, a_{2k+1}) \in \Sigma_B$$

Turingov stroj

Definícia 4.15 Dva modely strojov (triedy strojov) na riešenie rozhodovacích problémov $\mathcal{A}, \mathcal{B}$ sú **ekvivalentné** keď

(i) pre každý stroj $A \in \mathcal{A}$ existuje stroj $B \in \mathcal{B}$, ktorý je ekvivalentný s A a

(ii) pre každý stroj $B \in \mathcal{B}$ existuje stroj $A \in \mathcal{A}$, ktorý je ekvivalentný s B.

Turingov stroj

Church-Turingova téza

Turingove stroje sú formalizáciou označenia „algoritmus“, t.j. trieda rekurzívnych jazykov (rozhodnuteľných rozhodovacích problémov) zodpovedá triede algoritmicky (automaticky) rozpoznateľných jazykov

Turingov stroj

Definícia 4.21 Nedeterministický Turingov stroj (NTS) je 7-mica $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$, kde

(i) $Q, \Sigma, \Gamma, q_0, q_{\text{accept}}, q_{\text{reject}}$ majú rovnaký význam ako pri deterministickom TS,

(ii) $\delta : (Q - q_{\text{accept}}, q_{\text{reject}}) \times \Sigma \to \mathcal{P}(Q \times \Gamma \times \{\text{L,R,N}\})$ je **prechodová funkcia** stroja **M**, pre ktorú platí pre všetky $q \in Q - q_{\text{accept}}, q_{\text{reject}}$, že $\delta(q, ¢) \subseteq Q \times \{¢\} \times \{\text{R,N}\}$,

Turingov stroj

Konfigurácia je rovnaká ako pri deterministickom TS.

Krok stroja M je relácia $\vdash_M$ na konfiguráciách
($\vdash_M \subseteq \text{conf}(M) \times \text{conf}(M)$)

Jazyk $L(M)$ akceptovaný NTS M definujeme rovnako ako pre deterministický TS

$$L(M) = \{w \in \Sigma^* \mid M \text{ akceptuje } x\}.$$

Turingov stroj

Definícia 4.25 Nech $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ je NTS a x je slovo nad vstupnou abecedou Σ. **Strom výpočtu $T_{M,x}$ stroja M na x** je orientovaný strom definovaný takto:

(i) Každý vrchol $T_{M,x}$ je označený konfiguráciou.

(ii) Koreň stromu je jediný vrchol, do ktorého nevchádza hrana a je označený počiatočnou konfiguráciou $q_0 ¢ x$ stroja M na x.

(iii) Každý vrchol $T_{M,x}$ označený konfiguráciou C má práve toľko synov, do koľkých konfigurácií sa z C vieme dostať na jeden krok výpočtu. Synovia sú označení príslušnými nasledovnými konfiguráciami konfigurácie C.

Turingov stroj

"kompilácia" programu pre TS

Majme TS $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$, kde
$Q = \{q_0, q_1, \ldots, q_m, q_{\text{accept}}, q_{\text{reject}}\}$ a $\Gamma = \{A_1, A_2, \ldots, A_r\}$.
$\text{Code}(q_i) = 10^{i+1}1$, pre $i = 0, \ldots, m$,
$\text{Code}(q_{\text{accept}}) = 10^{m+2}1$,
$\text{Code}(q_{\text{reject}}) = 10^{m+3}1$,
$\text{Code}(A_j) = 110^j11$, pre $j = 1, \ldots, r$,
$\text{Code}(N) = 1110111$,
$\text{Code}(R) = 1110^2111$,
$\text{Code}(L) = 1110^3111$,
$\text{Code}(\delta(p, A_l) = (q, A_m, \alpha)) =$
$\#\text{Code}(p)\text{Code}(A_l)\text{Code}(q)\text{Code}(A_m)\text{Code}(\alpha)$
pre prechod $\delta(p, A_l) = (q, A_m, \alpha)$, kde $p \in \{q_0, \ldots, q_m\}$, $q \in Q$,
$l, m \in \{1, \ldots, r\}$ a $\alpha \in \{L, R, N\}$.

Turingov stroj

$\text{Code}(M) = \#0^{m+3}\#0^{r}\#\text{Code}(Prechod_1)\text{Code}(Prechod_2)\ldots$

Ešte sa musíme zbaviť znaku $\#$. Jedno z mnohých riešení je homomorfizmus h, $h(0) = 00$, $h(1) = 11$, $h(\#) = 01$.

Definícia 4.30 Pre Turingov stroj M označíme **kód TM M**

$$\textbf{Kod}(M) = h(\text{Code}(M)) \text{ a}$$

$$\textbf{KodTM} = \{\text{Kod}(M) \mid M \text{ je TS}\}.$$

$A_{\textbf{ver}}$ označíme algoritmus (TS), ktorý rozhodne rozhodovací problém $(\Sigma_{\text{bool}}, \text{KodTM})$, t.j. či dané $x \in (\Sigma_{\text{bool}})^*$ je kód TS.

Turingov stroj

Definícia 4.33 $x \in (\Sigma_{\text{bool}})^*$. Pre ľubovoľné kladné celé i hovoríme, že x je **kód i-teho TS** ak

(i) $x = \text{Kod}(M)$ pre TS M a

(ii) množina $\{y \in (\Sigma_{\text{bool}})^* \mid$
y je pred x vzhľadom na kanonické usporiadanie$\}$
obsahuje presne $i - 1$ slov, ktoré sú kódmi TS.

Ak $x = Kod(M)$ je kód i-teho TS, potom M je **i-ty TS M_i**. i je **poradové číslo TS M_i**.

Vypočítateľnosť

Existuje jazyk, ktorý nie je rekurzívne vyčísliteľný (neexistuje TS, ktorý by ho rozpoznával).

Množina KodTM má menšiu kardinalitu ako množina všetkých jazykov nad Σ_{Bool}.

Definícia 5.1 A a B sú množiny.

Hovoríme, že $|A| \leq |B|$, ak existuje injektívne zobrazenie z A do B.
Hovoríme, že $|A| = |B|$, ak $|A| \leq |B|$ a súčasne $|B| \leq |A|$.
Hovoríme, že $|A| < |B|$, ak $|A| \leq |B|$ a neexistuje injektívne zobrazenie z B do A.

Vypočítateľnosť

Definícia 5.4 Množina A sa nazýva spočítateľná (countable) ak je konečná, alebo $|A| = |\mathbb{N}|$.

Príklady spočítateľných množín:
abeceda Σ, Σ^*, KodTM, $\mathbb{Z}$, $\mathbb{N}^+ \times \mathbb{N}^+$, $\mathbb{Q}^+$, $\mathbb{N}^+ \times \mathbb{N}^+ \times \mathbb{N}^+$

ale

Množina [0,1] nie je spočítateľná.
$\mathcal{P}((\Sigma_{\text{Bool}})^*)$ nie je spočítateľná.

Vypočítateľnosť

diagonalizačný jazyk L_{diag}

$w_1, w_2, \ldots$ sú všetky slová nad Σ_{Bool} v kanonickom usporiadaní.
$M_1, M_2, \ldots$ je postupnosť všetkých TS.
$d_{ij} = 1 \Leftrightarrow M_i$ akceptuje j-te slovo w_j.

Definujme $L(M_i) = \{w_j \mid d_{ij} = 1 \text{ pre všetky } j \in \mathbb{N}^+\}$.

Diagonalizačný jazyk L_{diag} je rôzny od každého jazyka $L(M_i)$.

$$\begin{aligned} \boldsymbol{L_{\text{diag}}} &= \{w \in (\Sigma_{\text{Bool}})^* \mid w = w_i, \text{ pre nejaké } i \in \mathbb{N}^+ \\ &\qquad \text{a súčasne } M_i \text{ neakceptuje } w_i\}. \\ &= \{w \in (\Sigma_{\text{Bool}})^* \mid w = w_i, \text{ pre nejaké } i \in \mathbb{N}^+ \\ &\qquad \text{a súčasne } d_{ii} = 0\}. \end{aligned}$$

Vypočítateľnosť

Definícia 5.23 Σ_1 a Σ_2 sú abecedy. $L_1 \subseteq \Sigma_1^*$ a $L_2 \subseteq \Sigma_2^*$ sú jazyky. Hovoríme, že L_1 **je (rekurzívne) redukovateľný na** L_2 a označujeme $L_1 \leq_R L_2$ ak

$$L_2 \in \mathcal{L}_R \Rightarrow L_1 \in \mathcal{L}_R.$$

Vzhľadom na algoritmickú riešiteľnosť je L_2 aspoň tak ťažký ako L_1.

Definícia 5.24 Nech sú $L_1 \subseteq \Sigma_1^*$ a $L_2 \subseteq \Sigma_2^*$ jazyky nad abecedami Σ_1 a Σ_2. Hovoríme, že M **redukuje jazyk** L_1 **na jazyk** L_2, $L_1 \leq_m L_2$, keď existuje TS M, ktorý počíta zobrazenie $f_m : \Sigma_1^* \to \Sigma_2^*$ s vlastnosťou $x \in L_1 \Leftrightarrow f_m(x) \in L_2$, pre všetky $x \in \Sigma_1^*$, Funkcia f_m sa nazýva **redukcia** L_1 na L_2

Vypočítateľnosť

Lema 5.25 Ak $L_1 \leq_m L_2$, potom $L_1 \leq_R L_2$.

Lema 5.27 $L \leq_R L^C$ a súčasne $L^C \leq_R L$.

Dôsledok 5.28 $(L_{\text{diag}})^C \notin \mathcal{L}_{\text{R}}$.

Lema 5.29 $(L_{\text{diag}})^C \in \mathcal{L}_{\text{RE}}$.

Dôsledok 5.30 $(L_{\text{diag}})^C \in \mathcal{L}_{\text{RE}} - \mathcal{L}_{\text{R}}$, teda $\mathcal{L}_{\text{R}} \subsetneq \mathcal{L}_{\text{RE}}$.

Vypočítateľnosť

Definícia 5.31 Univerzálny jazyk ja jazyk

$$L_{\mathsf{U}} = \{\mathrm{Kod}(M)\#w \mid w \in (\Sigma_{\mathrm{Bool}})^* \text{ a } M \text{ akceptuje } w\}.$$

Veta 5.32 Existuje **univerzálny** TS U taký, že

$$L(U) = L_{\mathsf{U}}.$$

Teda platí, že $L_{\mathsf{U}} \in \mathcal{L}_{\mathsf{RE}}$.

Veta 5.33 $L_{\mathsf{U}} \notin \mathcal{L}_{\mathsf{R}}$.

Vypočítateľnosť

Definícia 5.35 Problém zastavenia (halting problem) je rozhodovací problém $(\{0,1,\#\}^*, L_{\mathsf{H}})$, kde

$$L_{\mathsf{H}} = \{\mathsf{Kod}(M)\#x \mid x \in (\Sigma_{\mathsf{Bool}})^* \text{ a } M \text{ zastaví na } x\}.$$

Veta 5.37 $L_{\mathsf{H}} \notin \mathcal{L}_{\mathsf{R}}$.

Vypočítateľnosť

$$L_{\text{empty}} = \{\text{Kod}(M) \mid L(M) = \emptyset\}.$$

$$(L_{\text{empty}})^C = \{x \in \{0,1\}^* \mid (x \neq \text{Kod}(M') \text{ pre všetky TS } M') \text{ alebo } (x = \text{Kod}(M) \text{ a } L(M) \neq \emptyset)\}.$$

Lema 5.39 $(L_{\text{empty}})^C \in \mathcal{L}_{\text{RE}}$.

Lema 5.40 $(L_{\text{empty}})^C \notin \mathcal{L}_{\text{R}}$.

Dôsledok 5.41 $L_{\text{empty}} \notin \mathcal{L}_{\text{R}}$.

Dôsledok 5.43 Jazyk
$L_{\text{eq}} = \{\text{Kod(M)\#Kod}(M') \mid L(M) = L(M')\} \notin \mathcal{L}_{\text{RE}}$.

Vypočítateľnosť

Riceova veta

Definícia Vlastnosť jazykov z $\mathcal{L}_{\mathsf{RE}}$ je nejaká ich podmnožina.

Vlastnosť je **triviálna** keď je buď $\emptyset$ alebo celé $\mathcal{L}_{\mathsf{RE}}$

Rieceova veta Každá netriviálna vlastnosť jazykov z $\mathcal{L}_{\mathsf{RE}}$ je nerozhodnuteľná ($\notin \mathcal{L}_{\mathsf{R}}$).