Turingův stroj – syntaxe

Definice. (Deterministický) Turingův stroj (Turing Machine, TM) je devítice $\mathcal{M} = (Q, \Sigma, \Gamma, \triangleright, \sqcup, \delta, q_0, q_{acc}, q_{rej})$, kde

- Q je konečná množina, jejíž prvky nazýváme **stavy**,
- Σ je konečná množina, tzv. **vstupní abeceda**,
- Γ je konečná množina, tzv. **pracovní abeceda**, $\Sigma \subseteq \Gamma$,
- $\triangleright \in \Gamma \smallsetminus \Sigma$ je **levá koncová značka**,
- $\sqcup \in \Gamma \smallsetminus \Sigma$ je symbol označující **prázdné políčko**,
- $\delta : \ (Q \smallsetminus \{q_{acc}, q_{rej}\}) \times \Gamma \ \rightarrow \ Q \times \Gamma \times \{L, R\}$ je **totální přechodová funkce**,
- $q_0 \in Q$ je **počáteční stav**,
- $q_{acc} \in Q$ je **akceptující stav**,
- $q_{rej} \in Q$ je **zamítající stav.**

Dále požadujeme, aby pro každé $q \in Q$ existoval $p \in Q$ takový, že $\delta(q, \triangleright) = (p, \triangleright, R)$ (tj. $\triangleright$ nelze přepsat ani posunout hlavu za okraj pásky).

Označení. $$\sqcup^\omega = \sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\ldots$$

Definice. **Konfigurace** Turingova stroje je trojice $(q, z, n) \in Q \times \{y\sqcup^\omega \mid y \in \Gamma^*\} \times \mathbb{N}_0$, kde

- q je stav,
- $y\sqcup^\omega$ je obsah pásky,
- n značí pozici hlavy na pásce.

Počáteční konfigurace pro vstup $w \in \Sigma^*$ je trojice $(q_0, \triangleright w\sqcup^\omega, 0)$.

Akceptující konfigurace je každá trojice tvaru (q_{acc}, z, n).

Zamítající konfigurace je každá trojice tvaru (q_{rej}, z, n).

Výpočet Turingova stroje

Označení. Pro libovolný nekonečný řetěz z nad Γ, z_n označuje n-tý symbol řetězu z (z_0 je nejlevější symbol řetězu z). Dále $s_b^n(z)$ označuje řetěz vzniklý ze z nahrazením z_n symbolem b.

Definice. Na množině všech konfigurací stroje $\mathcal{M}$ definujeme binární relaci **krok výpočtu** $\vdash_{\mathcal{M}}$ takto:

$$(p, z, n) \quad \vdash_{\mathcal{M}} \quad \begin{cases} (q, s_b^n(z), n+1) & \text{pro } \delta(p, z_n) = (q, b, R) \\ (q, s_b^n(z), n-1) & \text{pro } \delta(p, z_n) = (q, b, L) \end{cases}$$

Výpočet TM $\mathcal{M}$ na vstupu w je maximální (konečná nebo nekonečná) posloupnost konfigurací $K_0, K_1, K_2, \ldots$, kde K_0 je počáteční konfigurace pro w a $K_i \vdash_{\mathcal{M}} K_{i+1}$ pro všechna $i \geq 0$.

Stroj $\mathcal{M}$ **akceptuje** slovo w právě když výpočet $\mathcal{M}$ na w je konečný a jeho poslední konfigurace je akceptující.

Stroj $\mathcal{M}$ **zamítá** slovo w právě když výpočet $\mathcal{M}$ na w je konečný a jeho poslední konfigurace je zamítající.

Stroj $\mathcal{M}$ pro vstup w **cyklí** právě když výpočet $\mathcal{M}$ na w je nekonečný.

Jazyk akceptovaný TM $\mathcal{M}$ definujeme jako

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

Ukázky Turingových strojů

Simulátor TM:

`http://www.fi.muni.cz/~xbarnat/tafj/turing/`

Různé úrovně popisu TM

- formální
- neformální implementační
- vysokoúrovňový

Vícepáskový Turingův stroj

Definice. ***k*-páskový Turingův stroj** je definován stejně jako TM s vyjímkou přechodové funkce δ, která je definována jako totální funkce

$$\delta : \ (Q \smallsetminus \{q_{acc}, q_{rej}\}) \times \Gamma^k \ \rightarrow \ Q \times \Gamma^k \times \{L, R\}^k .$$

Konfigurace mají tvar $(q, z_1, \ldots, z_k, n_1, \ldots, n_k) \in Q \times (\Gamma^*.\{\sqcup^\omega\})^k \times \mathbb{N}_0^k$.
Počáteční konfigurace pro $w \in \Sigma^*$ je $(q_0, \triangleright w \sqcup^\omega, \triangleright \sqcup^\omega, \ldots, \triangleright \sqcup^\omega, 0, \ldots, 0)$.

Definice **akceptující/zamítající konfigurace** a $\vdash_{\mathcal{M}}$ se změní podobně.

Ekvivalence vícepáskového a jednopáskového TM

Věta. Pro každý vícepáskový Turingův stroj existuje ekvivalentní (jednopáskový) TM.

Důkaz.

1. Neprázdný obsah k pásek a polohy hlav vložíme za sebe na 1 pásku.
2. Simulace jednoho kroku = zjistit informace pod hlavami, zapsat nové a posunout hlavy.
3. Je-li třeba další políčko nějaké pásky, posuneme zbývající obsah.

$\square$

Nedeterministický Turingův stroj

Definice. Nedeterministický Turingův stroj $\mathcal{M}$ je definován stejně jako TM s vyjímkou přechodové funkce δ, která je definována jako totální funkce $\delta: \ (Q \smallsetminus \{q_{acc}, q_{rej}\}) \times \Gamma \ \rightarrow \ 2^{Q\times\Gamma\times\{L,R\}}$.

Všechny pojmy se definují stejně jako u deterministického TM. Drobné změny jsou jen u definice kroku výpočtu $\vdash_{\mathcal{M}}$ a akceptace slova.

$$(p, z, n) \ \vdash_{\mathcal{M}} \ (q, s_b^n(z), n+1) \quad \text{jestliže} \quad (q, b, R) \in \delta(p, z_n)$$
$$(p, z, n) \ \vdash_{\mathcal{M}} \ (q, s_b^n(z), n-1) \quad \text{jestliže} \quad (q, b, L) \in \delta(p, z_n)$$

$\mathcal{M}$ **akceptuje** slovo w, právě když existuje výpočet z počáteční konfigurace pro w do nějaké akceptující konfigurace.

Ekvivalence nedeterministického a deterministického TM

Věta. Pro každý nedeterministický Turingův stroj $\mathcal{N}$ existuje ekvivalentní deterministický TM.

Intuice:

Důkaz. Sestrojíme 3-páskový deterministický TM prozkoumávající výpočtový strom stroje $\mathcal{N}$. Tento 3-páskový stroj lze převést na jednopáskový deterministický TM.

Nechť $k = \max\{|\delta(q,z)| \mid q \in Q \smallsetminus \{q_{acc}, q_{rej}\}, z \in \Gamma\}$.

1. páska obsahuje vždy pouze vstup, nemění se.
2. páska slouží k simulaci nedeterministického stroje.
3. páska obsahuje řetězec $\{1, \ldots, k\}^*$ určující, který uzel stromu je právě prohledáván. Prohledávání začne uzlem 1.

Hledáme akceptující konfiguraci ve výpočtovém stromě prohledáním do šířky. Kontrola jednoho uzlu výpočtového stromu:

1. Zkopíruj první pásku na druhou.
2. Na 2. pásce simuluj $\mathcal{N}$, nedeterministické volby řeš podle čísel na 3. pásce. Narazíš-li na akceptující stav, akceptuj. V ostatních případech (příslušná volba neexistuje nebo $\mathcal{N}$ dojde do zamítajícího stavu nebo došly čísla na 3. pásce) pokračuj dalším krokem.
3. Nahraď řetězec na 3. pásce chápaný jako číslo v $(k+1)$-ární soustavě nejbližším vyšším číslem neobsahujícím nuly a začni znovu. $\square$

Další varianty Turingova stroje

- Turingův stroj s oddělenou vstupní páskou
- Turingův stroj s oboustranně nekonečnou páskou
- stroj se dvěma zásobníky
- stroj se vstupní páskou a dvěma čítači
- …

Všechny tyto varianty mají tutéž vyjadřovací sílu.

Churchova teze

Churchova (Church-Turingova) teze: Každý proces, který lze intuitivně nazvat algoritmem, se dá realizovat na Turingově stroji.

Další ekvivalentní formalismy:

- Minského stroje
- λ-kalkul
- while-programy
- …

Turingovy stroje a třídy jazyků

Věta. Jazyk L je rekursivně spočetný (tj. generovaný gramatikou typu 0) $\iff$ L je akceptovaný nějakým Turingovým strojem.

Důkaz. Neuveden. $\square$

Definice. Turingův stroj se vstupní abecedou Σ se nazývá **úplný**, je-li každý jeho výpočet konečný (akceptující nebo zamítající). Jazyk se nazývá **rekursivní**, pokud je akceptovaný nějakým úplným Turingovým strojem.

Terminologie

- (Obecný) TM $\mathcal{M}$ **akceptuje**/**rozpoznává**/**přijímá** jazyk $L(\mathcal{M})$.
- Je-li TM $\mathcal{M}$ úplný, říkáme, že $\mathcal{M}$ **rozhoduje** jazyk $L(\mathcal{M})$.

Přehled jazykových tříd

Jazyky	Gramatiky (typ)	Automaty
rekursivně spočetné	frázové (0)	Turingovy stroje
rekursivní	-	úplné Turingovy stroje
kontextové	kontextové (1)	lineárně ohraničené TM
bezkontextové	bezkontextové (2)	zásobníkové automaty
deterministické CFL	-	deterministické PDA
regulární	regulární (3)	konečné automaty

Třída na nižším řádku je vždy vlastní podtřídou třídy na vyšším řádku.

Uzávěrové vlastnosti rekursivních a rekursivně spočetných jazyků

Věta. Rekursivně spočetné i rekursivní jazyky jsou uzavřené na **sjednocení**.

Důkaz.

Nechť L_1, L_2 jsou jazyky akceptované TM $\mathcal{M}_1, \mathcal{M}_2$ s disjunktními množinami stavů. Stroj $\mathcal{M}$ akceptující $L_1 \cup L_2$ se nedeterministicky rozhodne, zda na svém vstupu spustí $\mathcal{M}_1$ nebo $\mathcal{M}_2$. Pokud spuštěný stroj akceptuje nebo zamítne, stroj $\mathcal{M}$ udělá totéž. Pokud $\mathcal{M}_1, \mathcal{M}_2$ byly úplné, je i $\mathcal{M}$ úplný. $\square$

Uzávěrové vlastnosti rekursivních a rekursivně spočetných jazyků

Věta. Rekursivně spočetné i rekursivní jazyky jsou uzavřené na **průnik**.

Důkaz.

Nechť L_1, L_2 jsou jazyky akceptované TM $\mathcal{M}_1, \mathcal{M}_2$ s disjunktními množinami stavů. Stroj $\mathcal{M}$ akceptující $L_1 \cap L_2$ spustí na svém vstupu w nejprve $\mathcal{M}_1$ a pokud akceptuje, pak na w spustí i $\mathcal{M}_2$. $\mathcal{M}$ akceptuje jen pokud oba stroje akceptují. Pokud $\mathcal{M}_1, \mathcal{M}_2$ byly úplné, je i $\mathcal{M}$ úplný. $\square$

Uzávěrové vlastnosti rekursivních a rekursivně spočetných jazyků

Věta. Rekursivně spočetné i rekursivní jazyky jsou uzavřené na **zřetězení**.

Důkaz.

Nechť L_1, L_2 jsou jazyky akceptované TM $\mathcal{M}_1, \mathcal{M}_2$ s disjunktními množinami stavů. Dvojpáskový stroj $\mathcal{M}$ akceptující $L_1.L_2$ vstupní slovo nedeterministicky rozdělí na dvě části, každou část dá na jednu pásku. Na první části slova spustí $\mathcal{M}_1$. Pokud akceptuje, spustí na druhé části $\mathcal{M}_2$. $\mathcal{M}$ akceptuje jen pokud oba stroje akceptují. Pokud $\mathcal{M}_1, \mathcal{M}_2$ byly úplné, je i $\mathcal{M}$ úplný. □

Uzávěrové vlastnosti rekursivních a rekursivně spočetných jazyků

Věta. Rekursivně spočetné i rekursivní jazyky jsou uzavřené na **iteraci**.

Důkaz.

Nechť L_1 je jazyky akceptovaný TM $\mathcal{M}_1$. Zkonstruujeme dvojpáskový TM $\mathcal{M}$ rozpoznávající L_1^*. $\mathcal{M}$ akceptuje, pokud je na první pásce prázdné slovo. V opačném případě nedeterministicky přesune nějaký neprázdný prefix slova z první pásky na druhou pásku, kde nad ním spustí $\mathcal{M}_1$. Pokud $\mathcal{M}_1$ zamítne, tak i $\mathcal{M}$ zamítne. Jinak opakuje celou proceduru. Je-li $\mathcal{M}_1$ úplný, je i $\mathcal{M}$ úplný. $\square$

Uzávěrové vlastnosti rekursivních a rekursivně spočetných jazyků

Věta. Rekursivní jazyky jsou uzavřené na **doplněk**.

Důkaz.

Nechť L_1 je jazyky akceptovaný úplným (deterministickým) TM $\mathcal{M}_1$. Úplný stroj $\mathcal{M}$ rozhodující *co*$-L_1$ získáme záměnou akceptujícího a zamítajícího stavu. $\square$

Uzávěrové vlastnosti rekursivních a rekursivně spočetných jazyků

Věta. Třída rekursivně spočetných jazyků **není** uzavřená na **doplněk**.

Důkaz. Provedeme později. □

Třída rekursivně spočetných jazyků **je uzavřená** na sjednocení, průnik, zřetězení, iteraci, pozitivní iteraci, mocniny.

Třída rekursivních jazyků **je uzavřená** na sjednocení, průnik, zřetězení, iteraci, pozitivní iteraci, mocniny, doplněk.

Vztah rekursivních a rekursivně spočetných jazyků

Věta. Jazyk L je rekursivní, právě když jsou jazyky L a $co{-}L$ rekursivně spočetné.

Důkaz. "$\Longrightarrow$" plyne z uzavřenosti rekursivních jazyků na komplement a z toho, že každý rekursivní jazyk je i rekursivně spočetný.

"$\Longleftarrow$":

Nechť $\mathcal{M}_1, \mathcal{M}_2$ jsou TM rozpoznávající L a $co{-}L$. Sestrojíme dvojpáskový stroj $\mathcal{M}$ rozhodující L. $\mathcal{M}$ nejprve zkopíruje vstupní slovo w na druhou pásku. Pak paralelně vykonává běh $\mathcal{M}_1$ na první pásce a běh stroje $\mathcal{M}_2$ na druhé pásce. Pokud stroj $\mathcal{M}_1$ akceptuje, pak $w \in L$ a stroj $\mathcal{M}$ také akceptuje. Pokud stroj $\mathcal{M}_2$ akceptuje, pak $w \notin L$ a stroj $\mathcal{M}$ zamítá. $\square$

Problémy jako jazyky

Problém rozhodnout, zda daný řetězec w má vlastnost P lze ztotožnit s množinou $\{w \mid w \text{ má vlastnost } P\}$.

Objekty O lze kódovat jako slova $\langle O \rangle$. Problém, zda O má vlastnost P ztotožníme s jazykem $\{\langle O \rangle \mid O \text{ má vlastnost } P\}$.

Příklad. Problém rozhodnout, zda daný konečný graf je souvislý, ztotožníme s jazykem $\{\langle G \rangle \mid G \text{ je konečný souvislý graf}\}$.

Kódování TM

Každý TM lze zakódovat do binárního řetězce. Předpokládáme, že $\mathcal{M}$

- má stavy $Q = \{q_1, q_2, q_3, \dots, q_n\}$
- q_1 je iniciální stav, q_2 akceptující stav, q_3 zamítající stav
- má páskovou abecedu $\Gamma = \{X_1, X_2, \dots, X_z\}$
- $X_1 = \triangleright$ je levá koncová značka, $X_2 = \sqcup$ je symbol pro prázdné pole

Přechod $\delta(q_i, X_j) = (q_k, X_l, L)$ kódujeme řetězcem $0^i10^j10^k10^l10$.
Přechod $\delta(q_i, X_j) = (q_k, X_l, R)$ kódujeme řetězcem $0^i10^j10^k10^l100$.

Z kódů jednotlivých přechodů (v libovolném pořadí) sestavíme kód $\mathcal{M}$:

$$\langle \mathcal{M} \rangle = 111\, \textit{kód}_1\, 11\, \textit{kód}_2\, 11 \dots 11\, \textit{kód}_r\, 111$$

Řetězce nekódující žádný TM považujeme za kód TM akceptujícího $\emptyset$.

Slova kódujeme podobně. Celkem $\langle \mathcal{M}, w \rangle = \langle \mathcal{M} \rangle \# \langle w \rangle$.

Univerzální Turingův stroj

Věta. Existuje **univerzální Turingův stroj** $\mathcal{U}$, který dokáže simulovat libovolný zadaný TM na zadaném vstupu:

$$\mathcal{U} \text{ akceptuje } \langle\mathcal{M}\rangle\#\langle w\rangle \iff \mathcal{M} \text{ akceptuje } w$$

Důkaz.

Stroj $\mathcal{U}$ je třípáskový. Nejprve ověří, že vstup je tvaru $\{0,1\}^*\#\{0,1\}^*$ (pokud ne, zamítá). Na první pásce je kód simulovaného stroje $\mathcal{M}$, na druhé pásce probíhá jeho výpočet, na třetí je uložen jeho stav. V každém kroku se na základě simulovaného stavu a obsahu pásky najde na první pásce příslušný přechod, který se pak provede. $\square$

Rozhodnutelnost problémů

Definice. Problém P odpovídající jazyku $L = \{\langle O \rangle \mid O \text{ má vlastnost } P\}$ je

- **rozhodnutelný**, právě když L je rekursivní
- **nerozhodnutelný**, právě když L není rekursivní
- **částečně rozhodnutelný (semirozhodnutelný)**, právě když L je rekursivně spočetný

Problém akceptování

Problém akceptování (problém příslušnosti pro Turingovy stroje) je problém rozhodnout, zda daný TM $\mathcal{M}$ akceptuje dané slovo w nad jeho vstupní abecedou. Problém ztotožníme s jazykem

$$ACC = \{\langle \mathcal{M}, w \rangle \mid \mathcal{M} \text{ je TM a } \mathcal{M} \text{ akceptuje } w\}.$$

Věta. Problém akceptování je částečně rozhodnutelný.

Důkaz. Plyne z existence univerzálního Turingova stroje. $\square$

Věta. Problém akceptování je nerozhodnutelný.

Důkaz. (Sporem:) Předpokládejme, že existuje TM $\mathcal{A}$ rozhodující problém akceptování. Tedy $\mathcal{A}$ akceptuje $\langle \mathcal{M}, w \rangle$, právě když $\mathcal{M}$ akceptuje w.

S využitím $\mathcal{A}$ zkonstruujeme TM $\mathcal{D}$: dostane-li $\mathcal{D}$ na vstupu zakódovaný stroj $\langle \mathcal{M} \rangle$, zeptá se stroje $\mathcal{A}$, zda $\mathcal{M}$ akceptuje svůj vlastní kód $\langle \mathcal{M} \rangle$ a následně odpověď otočí. Tedy

$$\begin{array}{lll} \mathcal{D} \text{ akceptuje } \langle \mathcal{M} \rangle, & \text{pokud} & \mathcal{M} \text{ neakceptuje } \langle \mathcal{M} \rangle \text{ a} \\ \mathcal{D} \text{ neakceptuje } \langle \mathcal{M} \rangle, & \text{pokud} & \mathcal{M} \text{ akceptuje } \langle \mathcal{M} \rangle. \end{array}$$

Nyní spustíme $\mathcal{D}$ na vstupu $\langle \mathcal{D} \rangle$:

$$\begin{array}{lll} \mathcal{D} \text{ akceptuje } \langle \mathcal{D} \rangle, & \text{pokud} & \mathcal{D} \text{ neakceptuje } \langle \mathcal{D} \rangle \text{ a} \\ \mathcal{D} \text{ neakceptuje } \langle \mathcal{D} \rangle, & \text{pokud} & \mathcal{D} \text{ akceptuje } \langle \mathcal{D} \rangle. \end{array}$$

To je spor. Stroj $\mathcal{A}$ tedy neexistuje a problém akceptování je nerozhodnutelný. $\square$

Ne-semirozhodnutelné problémy

Věta. Doplněk problému akceptování není ani částečně rozhodnutelný, tedy $co\!-\!ACC$ není rekursivně spočetný.

Důkaz.

$\square$

Důsledek. Třída rekursivně spočetných jazyků není uzavřená na doplněk.

Problém zastavení

Problém zastavení (halting problem) je problém rozhodnout, zda daný TM $\mathcal{M}$ má na daném slově *w* nad jeho vstupní abecedou konečný výpočet (tedy zda $\mathcal{M}$ na vstupu *w* zastaví). Problém ztotožníme s jazykem

$$HALT = \{\langle \mathcal{M}, w \rangle \mid \mathcal{M} \text{ je TM a výpočet } \mathcal{M} \text{ na } w \text{ je konečný}\}.$$

Věta. Problém zastavení je částečně rozhodnutelný.

Důkaz. Pomocí univerzálního Turingova stroje simulujeme $\mathcal{M}$ na *w*. Pokud simulovaný výpočet skončí, akceptujeme. $\square$

Věta. Problém zastavení je nerozhodnutelný.

Důkaz. (Sporem:) Předpokládejme, že existuje úplný TM $\mathcal{H}$ rozhodující problém zastavení. Pak ovšem umíme sestrojit TM $\mathcal{A}$ rozhodující problém akceptování. Stroj $\mathcal{A}$ dekóduje dvojici $\langle \mathcal{M}, w \rangle$ ze vstupu a změní $\mathcal{M}$ tak, že místo přechodů do zamítajícího stavu začne cyklit. Modifikovaný stroj $\mathcal{M}'$ zastaví, právě když $\mathcal{M}$ akceptuje. Nyní stačí spustit $\mathcal{H}$ na vstupu $\langle \mathcal{M}', w \rangle$. Dostáváme tedy úplný TM $\mathcal{A}$ rozhodující problém akceptování. To je spor. Úplný stroj $\mathcal{H}$ tedy neexistuje. $\square$