Kódovanie Turingových strojov a univerzálny Turingov stroj

Peter Kostolányi

30. novembra 2022

1 Normálny tvar deterministických Turingových strojov

Deterministický Turingov stroj A akceptuje vstup w práve vtedy, keď sa jeho výpočet na slove w aspoň raz dostane do konfigurácie s akceptačným stavom – a to aj v prípade, že sa stroj neskôr zastaví v neakceptačnej konfigurácii alebo sa nezastaví nikdy. Časť výpočtu nasledujúca za prvou akceptačnou konfiguráciou je však z hľadiska akceptácie zjavne nepodstatná – keďže už stroj raz bol v akceptačnej konfigurácii, akceptuje bez ohľadu na neskorší priebeh výpočtu.

Odstránením všetkých prechodov stroja A vedúcich z akceptačných stavov preto dostaneme stroj, ktorý je s pôvodným strojom A ekvivalentný. Ľubovoľný akceptačný výpočet tohto stroja sa navyše evidentne zastaví hneď nato, ako sa po prvý – a jediný – raz ocitne v akceptačnej konfigurácii.

Ku každému deterministickému Turingovmu stroju A teda existuje ekvivalentný deterministický Turingov stroj A' v *normálnom tvare bez prechodov z akceptačných stavov*. Výpočet takéhoto stroja na vstupe w môže prebiehať troma odlišnými spôsobmi:

1. Výpočet sa zastaví v akceptačnej konfigurácii – potom $w \in L(A')$.
2. Výpočet nikdy nepríde do akceptačnej konfigurácie a zastaví sa v neakceptačnej konfigurácii – potom $w \notin L(A')$.
3. Výpočet nikdy nepríde do akceptačnej konfigurácie a nikdy sa nezastaví – potom $w \notin L(A')$.

V nasledujúcich partiách týchto poznámok, ako aj neskôr v súvislosti s teóriou rozhodnuteľnosti *budeme vždy predpokladať, že deterministické Turingove stroje sú v uvedenom normálnom tvare.*

2 Kódovanie Turingových strojov do binárnych reťazcov

Opíšeme teraz jeden zo spôsobov kódovania deterministických Turingových strojov do binárnych reťazcov. Budeme pritom uvažovať iba stroje nad vstupnou abecedou $\Sigma = \{0, 1\}$; analogické metódy by sa však dali použiť aj pre stroje nad ľubovoľnou vstupnou abecedou.

Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je deterministický Turingov stroj so vstupnou abecedou $\Sigma = \{0, 1\}$. Bez ujmy na všeobecnosti môžeme predpokladať, že pre nejaké $k \in \mathbb{N}$ je $K = \{q_0, q_1, \ldots, q_k\}$ a pre nejaké $m \in \mathbb{N}$ je $\Gamma = \{0, 1, c_1, \ldots, c_m\}$. Stavy, symboly abecedy $\Gamma \cup \{\mathbf{B}\}$ a posuny hlavy potom môžeme zakódovať napríklad nasledujúcim spôsobom:

$$\begin{array}{lll}
\langle q_0 \rangle = 01, & \langle 0 \rangle = 01, & \langle -1 \rangle = 01, \\
\langle q_1 \rangle = 001, & \langle 1 \rangle = 001, & \langle 0 \rangle = 001, \\
\langle q_2 \rangle = 0001, & \langle \mathbf{B} \rangle = 0001, & \langle 1 \rangle = 0001, \\
\vdots & \langle c_1 \rangle = 00001, & \\
\langle q_k \rangle = 0^{k+1}1, & \langle c_2 \rangle = 000001, & \\
 & \vdots & \\
 & \langle c_m \rangle = 0^{m+3}1. &
\end{array}$$

Stroj $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ následne môžeme zakódovať do podoby slova nad abecedou $\{0, 1, \#\}$, napríklad ako

$$\langle\langle A \rangle\rangle = \#\#\langle K \rangle\#\#\langle \Gamma \cup \{\mathbf{B}\} \rangle\#\#\langle F \rangle\#\#\langle \delta \rangle\#\#,$$

kde kódy $\langle K\rangle$, $\langle \Gamma \cup \{\mathbf{B}\}\rangle$ sú dané ako

$$\begin{aligned}\langle K\rangle &= \langle q_0\rangle\#\langle q_1\rangle\#\ldots\#\langle q_k\rangle,\\ \langle \Gamma \cup \{\mathbf{B}\}\rangle &= \langle 0\rangle\#\langle 1\rangle\#\langle \mathbf{B}\rangle\#\langle c_1\rangle\#\ldots\#\langle c_m\rangle;\end{aligned}$$

ak navyše $F = \{q_{i_1}, \ldots, q_{i_s}\}$ pre nejaké $0 \leq i_1 < \ldots < i_s \leq k$, môžeme položiť

$$\langle F\rangle = \langle q_{i_1}\rangle\#\langle q_{i_2}\rangle\#\ldots\#\langle q_{i_s}\rangle$$

a kód $\langle\delta\rangle$ prechodovej funkcie δ, danej ako

$$\begin{aligned}\delta(p_1, a_1) &= (p'_1, b_1, d_1),\\ \delta(p_2, a_2) &= (p'_2, b_2, d_2),\\ &\vdots\\ \delta(p_t, a_t) &= (p'_t, b_t, d_t),\end{aligned}$$

môžeme definovať nasledujúcim spôsobom:

$$\langle\delta\rangle = \langle p_1\rangle\langle a_1\rangle\langle p'_1\rangle\langle b_1\rangle\langle d_1\rangle\#\langle p_2\rangle\langle a_2\rangle\langle p'_2\rangle\langle b_2\rangle\langle d_2\rangle\#\ldots\#\langle p_t\rangle\langle a_t\rangle\langle p'_t\rangle\langle b_t\rangle\langle d_t\rangle.$$

Niektoré zo súčastí takto definovaného kódu $\langle\langle A\rangle\rangle$ sú očividne zbytočné. Čitateľ by iste sám dokázal definovať kód Turingovho stroja množstvom rôznych spôsobov.

Od kódu $\langle\langle A\rangle\rangle$ nad abecedou $\{0, 1, \#\}$ je už len krok ku kódu $\langle A\rangle$ nad binárnou abecedou $\{0, 1\}$. Ten možno získať z kódu $\langle\langle A\rangle\rangle$ použitím homomorfizmu $0 \mapsto 01, 1 \mapsto 001, \# \mapsto 0001$.

Je zrejmé, že nie každý binárny reťazec zodpovedá zmysluplnému kódu deterministického Turingovho stroja. Aby sme predišli problémom s tým spojeným, budeme všetky „nezmyselné“ kódy považovať za kódy niektorého konkrétneho Turingovho stroja $A_\emptyset$ akceptujúceho prázdny jazyk.

Poznámka 1. Deterministickému Turingovmu stroju vo všeobecnosti zodpovedá viacero rôznych kódov. Napríklad stroju $A_\emptyset$ zodpovedajú okrem jeho zmysluplných kódov aj všetky nezmyselné kódy. Ani ostatné stroje ale nemajú kód určený jednoznačne – napríklad kód prechodovej funkcie $\langle\delta\rangle$ závisí od poradia vymenovania prechodov, ktoré je vo všeobecnosti ľubovoľné. Keby sme tiež očíslovania stavov a pracovných symbolov nechápali ako dané *a priori*, ale ako *ľubovoľne zvolené*, boli by aj stavy a pracovné symboly zdrojmi ďalšej nejednoznačnosti (takýto pohľad je nutný, ak chceme naozaj vedieť kódovať *všetky* stroje).

Notácia $\langle A\rangle$ je teda trochu nepresná, pretože kód stroja A ňou nie je jednoznačne určený. Vo všeobecnosti platí, že pod symbolom $\langle A\rangle$ máme na mysli *ľubovoľný* kód stroja A a napríklad pod zápisom

$$L = \{\langle A\rangle \mid A \text{ je det. TS nad vstupnou abecedou } \{0, 1\}\}$$

chápeme jazyk *všetkých* kódov *všetkých* deterministických Turingových strojov nad binárnou vstupnou abecedou. Pri takejto interpretácii nám spomínaná nejednoznačnosť nebude vadiť.

3 Univerzálny Turingov stroj

Na prednáške bolo dokázané, že existuje *univerzálny Turingov stroj* U, ktorý na vstupe tvaru $\langle A\rangle\#w$, kde A je deterministický Turingov stroj so vstupnou abecedou $\{0, 1\}$ a $w \in \{0, 1\}^*$ je binárny reťazec, dokáže odsimulovať výpočet stroja A na slove w. Stroj U pritom vstup $\langle A\rangle\#w$ akceptuje práve vtedy, keď $w \in L(A)$.

Idea univerzálneho Turingovho stroja je tak veľmi podobná idei programovateľných počítačov: dodá sa „program“ A spoločne s „dátami“ w a stroj následne vykoná daný program na daných dátach.

Univerzálny Turingov stroj môže začať svoj výpočet na vstupe $\langle A\rangle\#w$ napríklad úpravou slova w na jeho kód (podľa predpisu $0 \mapsto 01, 1 \mapsto 001$ z predchádzajúceho oddielu) a jeho následnou úpravou na vhodnú reprezentáciu počiatočnej konfigurácie stroja A. Stroj U bude následne simulovať výpočet stroja A – v každom kroku zistí (z konfigurácie udržiavanej na páske) kód stavu stroja A a kód písmena čítaného jeho hlavou. Následne túto informáciu konfrontuje s kódom stroja A uloženým na páske, aby zistil prípadný výstup prechodovej funkcie stroja A pre daný stav a písmeno. Ak je tento výstup definovaný, stroj U vhodne upraví konfiguráciu udržiavanú na páske a pokračuje simuláciou ďalšieho kroku výpočtu stroja A – v prípade, že sa stav simulovaného stroja A zmenil na akceptačný, akceptuje aj stroj U. Ak výstup prechodovej funkcie nie je definovaný, výpočet stroja U končí.

Jazyk $L(U)$ akceptovaný takto skonštruovaným strojom U – takzvaný *univerzálny jazyk* – budeme označovať L_U:

$$L_U = \{\langle A\rangle\#w \mid A \text{ je det. TS nad vstupnou abecedou } \{0,1\};\ w \in \{0,1\}^*;\ w \in L(A)\}.$$

Z uvedeného vyplýva, že $L_U \in \mathscr{L}_{RE}$.