

16 Rozšíření Turingových strojů

16.1 Stroje s více páskami

Definice. Definujeme k -páskový Turingův stroj jako devítici $(Q, \Sigma, \Gamma, \triangleright, \sqcup, \delta, q_0, q_a, q_r)$, kde všechny symboly mají stejný význam jako v definici Turingova stroja vyjma přechodové funkce δ , která je definována jako zobrazení:

$$\delta: (Q \setminus \{q_a, q_r\}) \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R, S\}^k$$

Činnost vícepáskového stroje si představme tak, že na každé pásce je umístěna jedna hlava a každá z hlav provádí akci v rámci jednoho kroku výpočtu. Každá hlava může na aktuální stav reagovat jinou akcí. Všimněte si, že jsme do množiny $\{L, R\}$ přidali symbol S . Ten interpretujeme tak, že příslušná hlava stojí na místě. Tím umožníme některým hlavám se pohybovat, zatímco jiné stojí. U jednopáskového Turingova stroje jsme tuto možnost nepotřebovali, protože pokud by měla hlava provést na stejném políčku více přepisů, mohla je udělat již v jednom kroce a posunout se.

Definice. Konfigurace k -páskového stroje je libovolná $(2k + 1)$ -tice:

$$(q, w_1, \dots, w_k, n_1, \dots, n_k) \in Q \times (\{\triangleright\} \cdot \Gamma^* \cdot \{\sqcup^\omega\})^k \times \mathbb{N}_0^k$$

Počáteční konfigurace je potom konfigurace $(q_0, \triangleright w \sqcup^\omega, \triangleright \sqcup^\omega, \dots, \triangleright \sqcup^\omega, 0, \dots, 0)$.

Na začátku výpočtu vícepáskového stroje je vstup zapsán na první (nazýváme ji též *vstupní*) pásce, zatímco ostatní pásy jsou prázdné. Analogické pojmy jako akceptující a zamítající konfigurace a relaci krok výpočtu \vdash pro vícepáskové stroje bychom definovali intuitivně.

Příklad 1. Popíšeme vícepáskový stroj rozpoznávající jazyk:

$$L = \{a^n b^n c^n \mid n \geq 0\}$$

Stroj bude mít dvě pásy a bude fungovat následovně:

1. Hlava na vstupní pásce se bude pohybovat pouze zleva doprava a nebude znaky na vstupní pásce měnit.
2. Během čtení áček první hlavou bude druhá hlava tyto znaky opisovat na druhou pásku.
3. Během čtení b první hlavou se bude druhá hlava vracet. Jestliže druhá hlava narazí na začátek pásky dříve nebo později, než první hlava přečte první c , stroj zamítá.
4. Během čtení c první hlavou se bude druhá hlava posunovat opět doprava. Stroj akceptuje pouze v případě, že se obě hlavy ocitnou ve stejném kroce nad prázdným políčkem.

Činnost stroje popíšeme přímo definicí přechodové funkce (tabulka by v tomto případě byla zbytečně rozsáhlá a obsahovala by spoustu prázdných políček):

$$\begin{aligned}\delta(q_0, \triangleright, \triangleright) &= (q_0, \triangleright, \triangleright, R, R) \\ \delta(q_0, a, \sqcup) &= (q_0, a, a, R, R) \\ \delta(q_0, b, \sqcup) &= (q_1, b, \sqcup, S, L) \\ \delta(q_1, b, a) &= (q_1, b, a, R, L) \\ \delta(q_1, c, \triangleright) &= (q_2, c, \triangleright, S, R) \\ \delta(q_2, c, a) &= (q_2, c, a, R, R) \\ \delta(q_2, \sqcup, \sqcup) &= (q_a, -, -, -, -)\end{aligned}$$

Ve všech ostatních případech funkce δ vrací $(q_r, -, -, -, -)$.

Všimněme si, že oproti jednopáskovému stroji se problém akceptování jazyka L dvoupáskovým skutečně zjednodušil. Tento stroj má vedle mandatorních stavů $(q_0, q_a$ a $q_r)$ pouze dva pomocné stavy a slovo mu stačí přečíst pouze jednou. Popis vícepáskového stroje již ale není kompaktní záležitostí. Proto se často uchýlíme pouze ke slovnímu popisu stroje, jako je tomu u následujícího jazyka.

Příklad 2. Sestrojíme dvoupáskový stroj rozpoznávající jazyk:

$$L = \{a^p \mid p \text{ je prvočíslo}\}$$

Stroj bude fungovat následovně:

1. Stroj prochází vstupní pásku. Pakliže nejsou na pásce alespoň dva znaky a nebo je zde zapsán jiný znak než a , stroj zamítá.
2. Na druhou pásku si stroj запиše dva znaky a .
3. Stroj zkontroluje zda délka slova na vstupní pásce je dělitelná počtem znaků na druhé pásce. To na příklad tak, že opakovaně prochází druhou pásku od začátku do konce a souběžně s tím posunuje hlavu na vstupní pásce doprava. Mohou nastat následující případy:
 - (a) Obě hlavy současně ocitnou na obou páskách na konci hned při prvním průchodu druhé pásky. V tom případě jsou na obou páskách stejně dlouhé řetězce a automat akceptuje.
 - (b) Obě hlavy se současně ocitnou na obou páskách na konci při jiném než prvním průchodu druhé pásky. V tom případě automat zamítá, neboť našel dělitele délky slova na vstupní pásce.
 - (c) Hlava na první pásce narazí na konec slova, zatímco hlava na druhé pásce na konci slova není. V tom případě se hlava na druhé pásce přesune na konec, připíše jeden symbol a navíc a opakuje třetí krok.

Všimněte si, že schéma tohoto popisu již více připomíná běžné imperativní programování, než tomu bylo u programování Turingova stroje.

Věta 16.1.1. Ke každému vícepáskovému stroji existuje ekvivalentní jednopáskový stroj.

Důkaz. Uveďme pouze myšlenku, přesná konstrukce by byla technicky zbytečně náročná. Jednopáskový stroj bude simulovat činnost vícepáskového tak, že bude mít obsah všech pásek, oddělen speciálním znakem, zapsaný na jedné za sebou (dále tyto obsahy označujeme jako úseky). Každému kroku vícepáskového stroje bude odpovídat průchod hlavy celým obsahem pásky, která provede na každém úseku jeden krok výpočtu. Precizní konstrukce by přitom musela řešit následující technické obtíže:

1. Pro každou pásku se musí zavést zvláštní množina stavů, neboť hlavy jednotlivých pásek mohly reagovat ve stejném stavu různým způsobem.
2. Hlava si musí po každém kroku poznačit (třeba speciálním znakem), na kterém políčku příslušného úseku skončila výpočet.
3. Jestliže některá páska využívá prázdné políčka, musí se obsah následujících úseků posunovat doprava.

□

16.2 Nedeterministické stroje

Definice. *Nedeterministický Turingův stroj* je devítice $(Q, \Sigma, \Gamma, \triangleright, \sqcup, \delta, q_0, q_a, q_r)$, kde všechny symboly mají stejný význam jako v definici Turingova stroje vyjma přechodové funkce δ , která je definována jako zobrazení:

$$\delta: (Q \setminus \{q_a, q_r\}) \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L, R\})$$

Pojmy konfigurace, počáteční, akceptující a zamítající konfigurace jsou shodné jako u Turingova stroje, definice kroku výpočtu se změní pouze technicky. Výpočet nedeterministického stroje na nějakém slově už si nemůžeme představovat jako prostou posloupnost konfigurací, ale jako výpočetní strom, jehož uzly tvoří jednotlivé konfigurace, s počáteční konfigurací v kořeni a akceptujícími a zamítajícími konfiguracemi v listech. Přitom stroj akceptuje právě tehdy, když tento strom obsahuje alespoň jeden list s akceptující konfigurací.

Příklad 3. Napíšeme nedeterministický Turingův stroj rozpoznávající jazyk:

$$L = \{ww \mid w \in \{a, b\}^*\}$$

Jen poznamenejme, že tento jazyk není bezkontextový. Při návrhu Turingova stroje využijeme nedeterminizmu pro rozpoznání poloviny slova:

	\triangleright	a	b	X	\sqcup
q_0	(q_0, \triangleright, R)	(q_1, a, R)	(q_1, b, R)		q_a
q_1		$(q_1, a, R), (q_z^a, X, L)$	$(q_1, b, R), (q_b^z, X, L)$		
q_z^a	(q^a, \triangleright, R)	(q_z^a, a, L)	(q_z^a, b, L)	(q_z^a, X, L)	
a_z^b	(q^b, \triangleright, R)	(q_z^b, a, L)	(q_z^b, b, L)	(q_z^b, X, L)	
q^a		(q_X, X, R)		(q^a, X, R)	
q^b			(q_X, X, R)	(q^b, X, R)	
q_X		(q_X, a, R)	(q_X, b, R)	(q_2, X, R)	
q_2		(q_z^a, X, L)	(q_z^b, X, L)	(q_2, X, R)	(q_c, \sqcup, L)
q_c	q_a			(q_c, X, L)	

Věta 16.2.1. Pro každý nedeterministický Turingův stroj existuje ekvivalentní deterministický Turingův stroj.

Důkaz. Uvedeme jen myšlenku, přesná konstrukce by byla zbytečně technická. Simulaci nedeterministického stroje budeme provádět prohledáváním výpočetního stromu původního stroje. Tento průchod musí být do šířky, neboť výpočetní strom může obsahovat nekonečné větve (v případě, že jeho výpočet cyklí). Stroj bude mít tři pásy:

1. První páska obsahuje vždy pouze vstup, nemění se.
2. Druhá páska slouží k simulaci výpočtu nedeterministického stroje.
3. Třetí páska obsahuje řetězec $\{1, \dots, k\}^*$, kde:

$$k = \max\{|\delta(q, x)| \mid q \in Q \setminus \{q_a, q_r\}, x \in \Gamma\}$$

Tento řetězec popisuje cestu ve výpočetním stromě z kořene do nějakého uzlu jako sérii rozhodnutí, které pravidlo z přechodové funkce v daném kroce stroj použije.

Stroj pracuje následujícím způsobem:

1. Stroj opíše vstup na druhou pásku.
2. Simuluje činnost původního stroje na druhé pásce podle obsahuje třetí pásky.
3. Pokud došel do akceptujícího nebo zamítajícího stavu, stroj akceptuje nebo zamítá. V opačném případě inkrementuje konfiguraci na třetí pásce a vrací se do bodu 1.

□

16.3 Shrnutí

Obě předchozí rozšíření základního modelu Turingova stroje jsou tedy konzervativní, tj. nepřidáváme žádnou novou výpočetní sílu. Můžeme přitom oba modely

kombinovat a práci si ještě zjednodušit. Uveďme (pouze schematicky) návrh nedeterministického třípáskového stroje, který rozpoznává jazyk:

$$L = \{a^n \mid n \text{ je složené číslo}\}$$

Automat na druhou a třetí pásku zapíše náhodně řetězec a délky větší než 1 (neboli automat uhodne dělitele čísla n). Ve zbytku výpočtu deterministicky zkontroluje, zda součin délek řetězců na pomocných páskách je roven délce vstupu. To může provést třeba tak, že za každý znak na třetí pásce projde celou druhou pásku a na vstupní pásce provede stejný počet posunů doprava. Pokud se na konci tohoto procesu nachází na posledním znaku vstupu, akceptuje, v opačném případě zamítá.

Mohli bychom zavádět mnoho dalších formalizmů, které rozšiřují model Turingova stroje nebo jsou od něj odvozené:

- Stroj s oboustranně nekonečnou páskou.
- Stroj s vícerozměrnou páskou, tj. rozšiřujeme množinu $\{L, S, R\}$ o možnost pohybovat se do dalších směrů.
- Stroj se dvěma zásobníky.
- Stroj se vstupní páskou a dvěma počítadly.
- Pravděpodobnostní Turingovy stroje.

Všechny tyto varianty mají stejnou vyjadřovací sílu jako základní model. Historicky se objevilo mnoho dalších výpočetních modelů, ale každý pokus nalézt silnější formalismus než Turingův stroj selhal. To vedlo k formulování následující teze, která se po svých autorech jmenuje Churchova-Turingova:

Každý proces, který lze intuitivně nazvat algoritmem, se dá realizovat pomocí modelu Turingova stroje.

Tato teze není matematickým teorémem (protože se odvolává na lidskou intuici), nelze dokázat ani vyvrátit. Shrnuje pouze základní historickou zkušenost teoretické informatiky.

16.4 Příklady k procvičení

Cvičení 1. Popište činnost stroje, který rozpoznává jazyk:

$$L = \{a^{n^2} \mid n \in \mathbb{N}\}$$

Cvičení 2. Popište činnost stroje, který rozpoznává jazyk:

$$L = \{a^{n!} \mid n \in \mathbb{N}\}$$

Cvičení 3. Popište činnost stroje, který řeší problém kliky. Problém kliky můžeme popsat jako jazyk:

$$\text{CLIQUE} = \{\langle G \rangle \# k \mid k \in \mathbb{N} \wedge \text{graf } G \text{ obsahuje kliku velikosti } k\}$$

Značkou $\langle G \rangle$ máme na mysli nějaké zakódování struktury grafu na pásku, nicméně touto technikou se příliš nezabývejte. Klika grafu je definována jako úplný podgraf, tj. podgraf, v němž jsou všechny vrcholy spolu spojeny hranou.