Kapitola 1

Státnice - Algoritmicky nerozhodnutelné problémy

1.1 Turingovy stroje

Definice (Turingův stroj)

Deterministický Turingův stroj (DTS) M s k-páskami, kde k je konstanta, je pětice

$$M = (Q, \Sigma, \delta, q_0, F)$$

- $Q =$ konečná množina stavů řídící jednotky
- $\Sigma =$ konečná pásková abeceda
- $\delta : (Q \backslash F) \times \Sigma^k \mapsto Q \times \Sigma^k \times \{L, N, R\}^k$ je přechodová funkce (částečná)
- $q_0 \in Q =$ počáteční stav
- $F \subseteq Q =$ množina přijímajících stavů

Definice (Konfigurace TS/Postovo slovo)

Konfigurace (jednopáskového) TS, neboli Postovo slovo, je obsah nejmenší souvislé části pásky, která obsahuje všechna neprázdná a čtené políčko; poloha hlavy a stav. Zapisujeme:

$$UqsV$$

kde U, V jsou části pásky nalevo a napravo od hlavy, q je stav a s čtené políčko.

Poznámka (Kombinování TS)

- Spojení dvou TS: napřed počítá $M1$, na výsledek pustím $M2$, tj. $M1 \wedge M2$
- Větvení (if-then-else): ve stroji $M1$ ze stavu q_0 přechod do (poč. stavu) $M2$, z q_1 do $M3$
- While-cyklus: složené spojení a větvení

Nutná opatrnost – stejná vnější abeceda, disjunktní vnitřní stavy atd.

Poznámka (Modifikace a omezení (stručně))

- Omezení pohybu na jeden směr – síla stroje klesne na úroveň konečných automatů
- Omezení na povinný pohyb (L/P) – OK
- Jen jedna činnost v taktu (buď zápis, nebo posuv) – OK
- Jednostranně omezená páska, více pásek, více hlav – stále stejná síla
- Okraje pásky z obou stran – páska není nekonečná, mám jen konfiguraci stroje – můžu mít potřebu pásku zvětšovat a zmenšovat (je možné)
- Omezení na 2 aktivní stavy – OK (jeden ale nestačí)

- Omezení na 2 symboly abecedy – OK (z toho jedno je "blank")
- K simulaci TS stačí 2 zásobníkové automaty – z jednoho zásobníku uděláme obsah pásky nalevo, z druhého napravo (vč. čteného znaku) a přehazujeme znaky.

1.2 Univerzální Turingův stroj

Věta (Univ. Turingův stroj)

Máme dánu abecedu A. Existuje univerzální TS $\mathcal{U}$ nad A, který pro každý TS nad A simuluje výpočet.

$$\mathcal{U}(\text{kód}(T) + \text{kód}(S)) \simeq T(S)$$

Důkaz

- Vezmeme $A = \{\lambda, |\}$, což stačí. BÚNO má každý TS jediný koncový stav q_f, počáteční stav buď q_s. Počet stavů – m – může být velký. Kód stavu q_i budiž blok znaků délky $m + 2$ ($|$ + i-krát $|$ + $m - i$-krát λ + λ).
- Pro $i \geq 1$ máme vždy dvě instrukce (jedna pro λ, druhá pro $|$). Ty se dají zakódovat do bloku $\times O_1 \times O_2 \times O_3 \cdots \times O_m \times$, kde $\times$ je pomocný symbol (v abecedě $\mathcal{U}$ být může) a O_i jsou kódy obou instrukcí pro stav r_i – kód zapisovaného písmene, pohybu a cílového stavu.
- Páska $\mathcal{U}$ pak vypadá následovně:

$$Y[\text{blok1}]Y[\text{blok2}]\Delta[\text{blok3}] \times O_1 \times O_2 \cdots \times O_m \times$$

  - "blok1" je konfigurace pův. stroje, jen obsah právě čteného pole je nahrazen pomocným symbolem M.
  - "blok2" kóduje aktuální stav pův. stroje.
  - "blok3" je jedna buňka, v níž je uložen obsah právě čteného pole.
- Univerzální stroj potom sestává z testu bloku2, zda obsahuje koncový stav, procedury vyčištění pásky a vydání výsledku a odsimulování jednoho kroku práce původního stroje
- Simulace:

1. najít relevantní blok O_i – stav i si nelze pamatovat přímo, proto musím z bloku 2 postupně umazávat $|$ a posouvat nějaký spec. symbol "zarážku" doprava
2. posunout zarážku na konkrétní instrukci podle bloku3 (čteného znaku)
3. provést instrukci (po kouskách přenést nový stav do bloku 2, pak 6 možností zapisování písmene a pohybu, při pohybu a mazání pozor na okraje pásky)

Důsledek

Díky tomu lze všechny TS očíslovat.

Věta (Halting problém)

Halting problém není algoritmicky rozhodnutelný.

Důkaz

Sporem nechť máme TS $H(T, K)$ rozhodující, zda se TS T zastaví nad daty K (a H se zastaví vždy a vydá buď 0 nebo 1). Potom lze vyrobit $Alg(K)$ takový, že $Alg(K)$ se zastaví, právě když $\mathcal{U}(K + K)$ se nezastaví (pomocí H). Pak $Alg(K)$ má nějaký kód, nazveme jej Q. Pak ale

$$Alg(Q) \text{ zastavi} \Leftrightarrow \mathcal{U}(Q + Q) \text{ nezastavi} \Leftrightarrow Alg(Q) \text{ nezastavi}$$

a to je spor.

Poznámka (Silně a slabě omezené mazání)

Omezíme mazání v TS:

- **slabě** – máme spec. symbol "kaňka" ($*$) a pravidla:
  - $\lambda \to$ cokoliv
  - cokoliv $\neq \lambda \to *$
- **silně** – máme abecedu jen $\{\lambda, *\}$ a povolený jen přepis $\lambda \to *$.

Oba dva případy mají stejnou sílu jako běžný TS (silné se slabým dá simulovat: kaňku kódovat jako blok samých kaněk, převést abecedu; normální TS se dá simulovat slabým postupným překreslováním konfigurací vedle sebe na pásku se současným měněním stavu).

Lze algoritmicky rozhodnout, zda TS T s konfigurací K někdy přepíše λ na něco jiného (existuje horní odhad počtu kroků v popsané části pásky). Nelze ale rozhodnout, zda TS T s konfigurací K někdy přepíše ne-λ na λ – to je ekvivalentní Halting problému (T simulujme silně omezeným T_1 a přidejme T_2, který smaže 1 kaňku. Pokud se T_1T_2 zastaví, musel se zastavit i T_1 a tím bychom rozhodli zastavení T).