Modeli računanja

15 10.2013

Če želimo dokazovati spodnje meje časovne zahtevnosti računskih problemov, moramo poleg kodirne sheme natančno definirati tudi pojem algoritma. Tu je možnih več pristopov, med najpogosteje uporabljenimi in za analizo časovne zahtevnosti najprimernejšimi je formalizacija pojma računskega stroja.

2.1. Turingov stroj

Turingov stroj (TS) sestavljajo nadzorna enota, bralno-pisalna glava in v desno neomejen trak, razdeljen na celice. TS deluje v korakih. Na vsakem koraku je nadzorna enota v enem od končno mnogo možnih stanj, glava se nahaja nad eno od celic traku, v vsaki celici pa je zapisan eden od končno mnogo možnih tračnih simbolov. TS ima vgrajeno prehodno funkcijo (program), ki glede na trenutno stanje nadzorne enote in trenutni simbol pod glavo določa novo stanje nadzorne enote, novi simbol pod glavo ter pomik glave (za 1 v levo ali v desno).

Formalno smo definirali ***nedeterministični Turingov stroj (NTS)*** kot urejeno sedmerko $S=(Q,q_0,q_s,\Sigma,\Gamma,\_,\delta)$, kjer je

1. Q končna množica *stanj* stroja S,
2. $q_0\in Q$ *začetno stanje* stroja S,
3. $q_s\in Q$ *sprejemno stanje* stroja S,
4. Σ *vhodna abeceda* stroja S,
5. Γ *tračna abeceda* stroja S ($\Sigma\subseteq\Gamma$, $\Gamma\cap Q=\emptyset$),
6. $\_\in\Gamma\setminus\Sigma$ *prazni simbol* stroja S, in
7. $\delta:(Q\setminus\{q_s\}\times\Gamma)\rightarrow P(Q\times\Gamma\times\{L,D\})$ *prehodna funkcija* stroja S.

NTS $S=(Q,q_0,q_s,\Sigma,\Gamma,\_,\delta)$ je ***determinističen Turingov stroj (DTS)***, če za vse $q\in Q\setminus\{q_s\}$ in $a\in\Gamma$ velja: $|\delta(q,a)|\leq 1$.

Definirali smo pojem *konfiguracije* NTS S in si ogledali dva načina uporabe Turingovih strojev:

a) za računanje vrednosti funkcij iz $\Sigma*\rightarrow\Gamma*$,

b) za reševanje problema pripadnosti dane besede $w\in\Sigma*$ izbranemu formalnemu jeziku iz $P(\Sigma*)$.

Uvedli smo grafično predstavitev NTS in kot zgled sestavili dva DTS: za računanje funkcije naslednika (eniško zapisanega) naravnega števila in za razpoznavanje dvojiških palindromov. Formalno smo definirali relacijo prehajanja konfiguracij in z njeno pomočjo jezik, ki ga razpoznava Turingov stroj *S*.

23.10.2013

Časovno zahtevnost $T_S:N\rightarrow N$ Turingovega stroja S smo definirali kot funkcijo, ki naravnemu številu n priredi največje izmed števil $n+1$ in $t_S(x)$ za vse $x\in\Sigma_n$, kjer je $t_S(x)$ največje število korakov pri kateremkoli izračunu stroja S z vhodno besedo x.

Ogledali smo si dve različici Turingovega stroja, in sicer *stroj z v obe smeri neomejenim trakom* in k*-tračni stroj*, kjer je $k \geq 2$. Pokazali smo, da lahko osnovni model Turingovega stroja S simuliramo z eno ali drugo različico v času $O(T_S(n))$. Stroj S z v obe smeri neomejenim trakom lahko simuliramo z osnovnim modelom v času $O(T_S(n))$, k-tračni stroj S pa v času $O(T_S^2(n))$.

30.10.2013

2.2. Stroj z neposrednim dostopom

Definirali smo še *stroj z neposrednim dostopom* (*SND*, angl. *random access machine* oz. *RAM*), ki je idealiziran model računalnika s programom na ravni zbirnega jezika. Časovno zahtevnost SND smo definirali z uporabo logaritemske cene za posamezne ukaze. Pokazali smo, da lahko s SND simuliramo DTS S v času $O(T_S(n) \log T_S(n))$.

6.11.2013

Pokazali smo, da lahko z večtračnim DTS simuliramo SND S v času $O(T_S^3(n))$.

Zaključek: Različne variante DTS in SND so torej *polinomsko povezane*: probleme, ki jih lahko rešimo v polinomskem času na enem od njih, lahko rešimo v polinomskem času tudi na vseh drugih.