

NP-polni problemi

3.1. Nedeterminizem

Opozorili smo, da nedeterministični Turingov stroj (NTS) ni model nobenega realističnega računskega stroja, ampak je le miselni pripomoček za preprostejši opis algoritmov, ki vključujejo iskanje in pregledovanje.

Definirali smo *problem izpolnljivosti izjavnih izrazov v konjunktivni normalni obliki (KNO)* (angl. SAT) in *problem izpolnljivosti izjavnih izrazov v k-KNO* (angl. k-SAT).

Skicirali smo determinističen algoritem za problem SAT in pokazali, da ima časovno zahtevnost $\Omega(2^{|I|/\sqrt{|I|}})$, kjer je $|I|$ dolžina zapisa danega izjavnega izraza I . Ta algoritem torej ni polinomski.

Nato smo skicirali še nedeterminističen algoritem za problem SAT in pokazali, da ima časovno zahtevnost $\mathcal{O}(|I|)$. Ta algoritem ima torej polinomsko časovno zahtevnost.

Pokazali smo, da lahko NTS S simuliramo z DTS v času $\mathcal{O}(c^{TS(n)})$, kjer je $c > 1$ neka konstanta (odvisna od stroja S).

3.2. Razreda P in NP

Fiksirajmo abecedo Σ , kjer je $|\Sigma| \geq 2$.

Definicija. \mathbf{P} je razred vseh jezikov nad Σ , ki jih lahko razpoznavamo z DTS v polinomskem času.

Definicija. \mathbf{NP} je razred vseh jezikov nad Σ , ki jih lahko razpoznavamo z NTS v polinomskem času.

13.11.2013

Navedli smo še nekaj zgledov problemov iz \mathbf{NP} : *Delitev, Natančno pokritje, Kromatično število, Klika, Hamiltonov cikel (HC)*.

Pokazali smo, da je razred \mathbf{P} zaprt za unijo, presek in komplement jezikov.

Pokazali smo, da je razred \mathbf{NP} zaprt za unijo in presek jezikov. Za komplemente problemov SAT, Delitev, Natančno pokritje, Kromatično število, Klika, HC ne vemo, ali so v \mathbf{NP} .

3.3. Polinomska prevedljivost problemov

Definicija. Jezik $J_1 \subseteq \Sigma^*$ je *prevedljiv* na jezik $J_2 \subseteq \Sigma^*$ v *polinomskem času*, če obstaja preslikava $f: \Sigma^* \rightarrow \Sigma^*$, za katero velja:

1. $\forall w \in \Sigma^*: (w \in J_1 \iff f(w) \in J_2)$,
2. f je izračunljiva z DTS v polinomskem času.

V tem primeru pišemo: $J_1 \rightarrow_P J_2$.

Trditev. Naj bo $J_1 \rightarrow_P J_2$. Potem velja:

1. $J_2 \in \mathbf{P} \implies J_1 \in \mathbf{P}$
2. $J_2 \in \mathbf{NP} \implies J_1 \in \mathbf{NP}$

3. $J_1 \notin P \implies J_2 \notin P$

4. $J_1 \notin NP \implies J_2 \in NP$

20.11.2013

Pokazali smo, da je relacija prevedljivosti jezikov v polinomskem času tranzitivna.

Jezik $J \subseteq \Sigma^*$ je *netrivialen*, če $J \neq \emptyset, \Sigma^*$. Pokazali smo, da lahko vsak jezik iz P prevedemo na katerikoli netrivialen jezik v polinomskem času.

3.4. Cook-Levinov izrek

Definicija. Jezik $J \subseteq \Sigma^*$ je *NP-poln*, če $J \in NP$ in je vsak jezik iz NP prevedljiv na J v polinomskem času.

Oznaka: $NPC = \{ J \in \Sigma^*; J \text{ NP-poln} \}$

Trditev. Naj bo $J \in NPC$. Potem velja:

$J \in P \iff P = NP$.

Cook-Levinov izrek. $SAT \in NPC$.

Lema. Naj bo $J_1 \in NPC$, $J_2 \in NP$ in J_1 prevedljiv na J_2 v polinomskem času. Potem je tudi $J_2 \in NPC$.

27.11.2013

Trditev. $3-SAT \in NPC$.

Definirali smo še naslednje probleme: *Vozliščno pokritje (VP)*, *Antiklika* in *Trirazsežno prirejanje (3DM)*.

Trditev. $VP \in NPC$.

4.12.2013

Izrek. $HC \in NPC$.

Izrek. $3DM \in NPC$.

11.12.2013

Trditev. $Delitev \in NPC$.

Omenili smo tri tehnike dokazovanja NP-polnosti: *Načrtovanje komponent*, *lokalno zamenjavo* in *zožitev na NP-poln podproblem*. Kot zgled zadnje smo dokazali:

Trditev. $Nahrbtnik \in NPC$.

18.12.2013

Definirali smo **pseudopolinomske algoritme** in s pomočjo dinamičnega programiranja konstruirali pseudopolinomski algoritem za problem Nahrbtnika. Definirali smo še **krepko NP-polne probleme** in pokazali, da iz obstoja pseudopolinomskega algoritma za katerega od krepko NP-polnih problemov sledi $P = NP$. Kot primer NP-polnega problema, ki ni krepko NP-poln (razen v primeru $P = NP$), smo navedli problem Nahrbtnika, kot primer krepko NP-polnega problema pa problem Potujočega trgovca (PT) in tudi vse NP-polne probleme, pri

katerih je po absolutni vrednosti največje nastopajoče celo število omejeno s polinomske funkcije dolžine zapisa podatkov.

Definirali smo še ***NP-težke probleme*** v najširšem smislu.