ALGORITMUSOK ÉS BONYOLULTSÁGELMÉLET

Matematika MSc hallgatók számára

8. Előadás

Előadó: Hajnal Péter

Jegyzetelő: Körmendi Kristóf

2010. március 29.

Az Immerman-Szelepcsényi tétel

A bonyolultságelmélet fontos kérdése, hogy a különböző nyelvosztályok hogyan viszonyulnak egymáshoz a tartalmazási relációra nézve. Az Immerman-Szelepcsényi tétel eszközt ad arra, hogy az $\mathcal{NL}$ és a co-$\mathcal{NL}$ osztályokat összehasonlítsuk.

Emlékeztető. Legyen $\vec{G}$ irányított gráf, legyen továbbá s, t a gráf két csúcsa. Ekkor az ELÉRHETŐSÉG nyelv pontosan azokat $\left(\vec{G}, s, t\right)$ hármasokat tartalmazza, amelyekre létezik $\vec{st}$ út $\vec{G}$-ben.

Az ELÉRHETŐSÉG nyelv $\mathcal{NL}$-teljes nyelv.

1. Tétel (Immerman-Szelepcsényi, 1987). *ELÉRHETŐSÉG* $\in$ *co-*$\mathcal{NL}$*.*

Megjegyzés. A tétel állítása kifejtve: Létezik olyan T Turing-gép, hogy tetszőleges $\left(\vec{G}, s, t\right)$ hármasra, ha $\vec{G}$-ben nincs $\vec{st}$ út, tehát $\left(\vec{G}, s, t\right) \notin$ ELÉRHETŐSÉG, akkor van T-nek olyan futása, ami felismeri ezt, ha pedig van $\vec{G}$-ben $\vec{st}$ út, tehát $\left(\vec{G}, s, t\right) \in$ ELÉRHETŐSÉG, akkor T-nek minden futása a „NEM-STIMMEL” állapotban végződik, továbbá T logaritmikus tárat használ.

Bizonyítás. Jelölje N_i azon csúcsok számát, melyek az s csúcsból legfeljebb i lépésben elérhetőek (az ilyen csúcsokre mint i-elérhetőekre hivatkozunk). Ha N_i értékét ismerjük, akkor fel tudjuk sorolni az i-elérhető csúcsokat, amely felsorolásból megkaphatjuk az $(i+1)$-elérhető csúcsok egy felsorolását, és így N_{i+1} értékét. Nyilvánvalóan s-ből 0 lépésben csak s-t érhetjük el, így $N_0 = 1$.

Ha egy v csúcs elérhető az s csúcsból, akkor létezik a $\vec{G}$ gráfban egy legfeljebb $|V| - 1$ hosszú $\vec{sv}$ út. Tehát az s-ből elérhető csúcsok halmaza megegyezik az s-ből legfeljebb $|V| - 1$ lépésben elérhető csúcsokkal. Következésképpen $N_{|V|-1} = N_{|V|}$. Világos, hogy, ha $N_i = N_{i+1}$ valamely i-re, akkor az i-elérhető csúcsok halmaza megegyezik az összes elérhető csúcs halmazával.

Az algoritmus pontosabb leírása: A munkaszalagra vigyük fel az i értéket melyre ki akarjuk számolni N_i-t és a már ismert N_{i-1} értéket, majd egy számlálót, amely az N_i értékét fogja felvenni.

$\triangleright$	i	$\#$	N_{i-1}	$\#$	N_i	$\#$	$\ldots$

Célunk ezen helyzet elérése, amennyiben az N_{i-1} érték „megbízható”. Az N_i-nek fenntartott hely egy számláló, ami kezdetben 0. Egy jelölt-területen minden $u \in V$ csúcsra ellenőrizzük, hogy az $(i-1)$-elérhető csúcsok valamelyikének szomszédja-e.

Ha igen, akkor növeljük meg a számláló értékét 1-el. A jelölt terület csak $\mathcal{O}(\log n)$ nagyságú. Egyszerre egy csúcsot tartalmaz, ha a jelöletet elvetettük vagy pedig a jelölt elfogadtuk (és a számlálót megnöveltük), akkor a helyére írjuk a következő u-t. A jelölt tesztelésére kódjaik nagyság szerinti sorrendjében felsoroljuk egy listázás-területen az $(i-1)$-elérhető csúcsokat (NEM-DETERMINIZMUS használata és $\mathcal{O}(\log n)$ tárigény, az előző filozófiával). Felsorolásunk során bizonyítjuk, hogy listázásunk aktuális csúcsa $(i-1)$-elérhető (bizonyító-területre van szükségünk, ahol NEM-DETERMINIZMUS-t használunk és az elérési úthoz $\mathcal{O}(\log n)$ a tárigényünk az előző filozófiával, továbbá egy számlálót tartunk fel a legfeljebb $i-1$ lépés betartásához). Amennyiben listánk v_j aktuális csúcsáról kiderült, hogy valóban $(i-1)$-elérhető, akkor egy lista-számláló-területen lévő számlálót növeljük, TOVÁBBÁ megtörténhet az u jelölt egy tesztelése: „v_j-ből vezet-e él u-hoz”. Ha igen, akkor u az i-elérhető csúcsok listázásának következő eleme. Az ezeket számláló számlálót növeljük és a következő csúcsot vesszük sorra, amíg az összest át nem néztük. Ha nem, akkor a listázási területen elővesszük v_{j+1}-et, amiről bizonyítjuk korrektségét és haladunk tovább. A tesztek sora akkor ér véget, amikor az $(i-1)$-elérhető csúcsok számlálója elérte N_{i-1}-et. Ha az utolsó teszten se halad át u, akkor a következő csúcs következik (az i-elérhető csúcsok listázása során nem történt semmi, u-t átugortuk).

A munkaszalag egy rövid szakasza, a jelölt-terület és listázás-terület határa:

...	u	#	v_j	...

Fent egy „jog-követő” futást írtunk le. Azaz a tippelt $(i-1)$-elérhető csúcsok valóban azok voltak. A hozzájuk tartozó bizonyító-területen valóban sorban egy rövid elérési út pontjait tipppelte meg gépünk. Ha ez nem így történik, akkor gépünk ezt észreveszi és „NEM-STIMMEL” állapotba kerül.

Ha eljutunk az összes jelölt csúcs tesztelése végéig (a NEM-STIMMEL állapot elkerülésével), akkor amennyiben N_{i-1} értéke korrekt volt N_i értéke is az. Ha $N_i = N_{i-1}$, akkor csak annyit kell ellenőriznünk, hogy t szerepel-e pozitívan tesztelt jelölt-csúcsok között. Ha nem, akkor ezzel a futással igazoltuk, hogy $\left(\vec{G}, s, t\right) \notin$ ELÉRHETŐSÉG. A $\left(\vec{G}, s, t\right) \in$ ELÉRHETŐSÉG esetén ilyen futás nincs.

A teljes tárigény természetesen $\mathcal{O}(\log n)$. ■

Az ELÉRHETŐSÉG nyelv teljessége miatt a következő állítás nyilvánvaló.

2. Következmény. $\mathcal{NL} = co\text{-}\mathcal{NL}$.

Természetesen a fenti algoritmus/nem-determinisztikus eljárás bármilyen nemdetermiszitikus gép konfiguráció gráfjára alkalmazható (feltesszük, hogy a tárigényt egy szép/kezelhető függvény írja le). Így a következő kovetkezményhez jutunk:

3. Következmény. *Ha $s(n)$ szép tárfüggvény és $s(n) \geq \log(n)$, akkor*

$$\mathcal{NSPACE}(s(n)) = co\text{-}\mathcal{NSPACE}(s(n)).$$

Problémák $\mathcal{NP}$ és $\mathcal{PSPACE}$ között

A következőkben olyan problémákra nézünk példákat melyekről tudjuk, hogy $\mathcal{PSPACE}$-beliek, azonban nyitott kérdés, hogy $\mathcal{NP}$-be esnek-e.

Definíció. Legyen V egy változóhalmaz. Legyen $\overline{V}$ a negált változók halmaza és $L = V\dot\cup\overline{V}$ a literálok halmaza. Ekkor a $C \subset V \cup \overline{V}$ alakú halmazokat klózoknak nevezzük. Minden diszjunktív normálformát felfoghatunk klózok összességeként (azaz egy halmazrendszerként L felett). Ha $\varphi = \vee_{i\in I} C_i$, akkor a φ DNF hossza alatt a $|\varphi| = \sum_{i\in I} |C_i|$ értéket értjük. Minden φ DNF-hez hozzárendelhetünk egy $f_\varphi : \{0,1\}^{|V|} \to \{0,1\}$ függvényt, amely a V-beli változók minden kiértékeléséhez φ értékét rendeli.

Példa. Egy adott diszjunktív normálformáról azt akarjuk ellenőrizni, hogy a vele minden kiértékelésben megegyező diszjunktív normálformák közül az adott-e a legrövidebb.

$$\text{OPT-DNF} = \{\varphi \text{ DNF } \mid \forall\psi \text{ DNF}, |\psi| < |\varphi| \text{ esetén } f_\psi \neq f_\varphi\}$$

Definíció. Legyen V tetszőleges halmaz, $\mathcal{H} \subset P(V)$. Azt mondjuk, hogy $A \subset V$ telített, ha bármely $R \subset A$ esetén létezik $E \in \mathcal{H}$, hogy $R = A \cap E$.

$$\dim_{\text{V-Cs}}(\mathcal{H}) = \max\{A \subset V \mid A \text{ telített}\}.$$

Megjegyzés. A $\mathcal{H}$ halamzrendszer tömör kódolása alatt a következőt értjük. Legyenek ν, μ olyanok, hogy $2^{\nu-1} \leq |V| < 2^\nu$, és $2^{\mu-1} \leq |\mathcal{H}| < 2^\mu$. Ekkor a csúcsokat, azaz V elemeit kódolhatjuk legfeljebb ν hosszú 0-1 sorozatokkal, míg ugyanezt megtehetjük az élekre legfeljebb μ hosszú 0-1 sorozatokkal. Legyen $f_\mathcal{H} : \{0,1\}^\nu \times \{0,1\}^\mu \to \{0,1\}$ olyan függvény, mely megmondja hogy egy adott él illeszkedik-e egy adott csúcsra. Ez az $f_\mathcal{H}$ függvény egy $\mathcal{C}$ hálózat segítségével kiszámítható. A $(|V|, |E|, \mathcal{C})$ hátmast nevezzük $\mathcal{H}$ tömör kódolásának. Jelölése $\ulcorner\mathcal{H}\urcorner$. Egyszerű gráfok tömör kódolásához mindössze a csúcsokat kell kódolnunk és egy olyan függvényt kiszámítanunk mely két adott csúcsról megmondja, hogy azok szomszédosak-e.

Példa. Egy halmazrendszerről szeretnénk eldönteni, hogy egy adott értékket meghalad-e a V-Cs dimenziója.

$$\text{V-CS DIM} = \{\,(\ulcorner\mathcal{H}\urcorner, k)\big|\dim_{\text{V-Cs}}(\mathcal{H}) \geq k\}.$$

Példa. Egyszerű gráfokban szeretnénk meghatározni a maximális független ponthalmaz gráfparaméter értékét. Tudjuk, hogy

$$\{(G,\kappa) \mid \alpha(G) \geq \kappa\} \in \mathcal{NP}, \text{ és } \{(G,\lambda) \mid \alpha(G) \leq \lambda\} \in \text{co-}\mathcal{NP}.$$

A PONTOS-FÜGGETLEN-PONTHALMAZ nyelvről, melynél az az eldöntendő kérdés, hogy az $\alpha(G)$ gráfparaméter értéke megyegyezik-e egy adott értékkel csak annyit tudunk hogy $\mathcal{NP}$ és egy co$\mathcal{NP}$-beli nyelv metszete, speciálisan $\mathcal{PSPACE}$-beli.

Vegyük észre hogy a példákat felfoghatjuk úgy mint $\mathcal{P}$-beli problémák, melyeket univerzális és egzisztenciális kvantorokkal toldottunk meg. Hiszen

$$\omega \in \text{ OPT-DNF, ha } \forall\psi\ \exists x\ \ [\psi \text{ DNF } |\psi| < |\varphi| \to f_\psi(x) \neq f_\varphi(x)]\,,$$

$$\omega \in \text{ V-CS DIM, ha } \exists A\ \forall R\ \exists E\ \ [E \in \mathcal{H}, R \subset A, \text{ és } R = A \cap E]\,,$$

$$\omega \in \text{ PFH, ha } \exists I\ \forall J\ \ [|I| = k, I \text{ független}, |J| > k \to J \text{ nem független}]\,.$$

Emlékeztető. Talán több hallgató számára hasonlóság mutatkozik a nem-determinsiztikus számítás tanúszalagos változatára. Ez alapján L akkor és csak akkor $\mathcal{NP}$-beli, ha alkalmas polinomiális Turing-gépre

$$\omega \in L \text{ akkor és csak akkor, ha } \exists \text{ polinomiális hosszú } \tau : T(\omega, \tau),$$

továbbá L akkor és csak akkor co$\mathcal{NP}$-beli, ha alkalmas polinomiális Turing-gépre

$$\omega \in L \text{ akkor és csak akkor, ha } \forall \text{ polinomiális hosszú } \tau : T(\omega, \tau),$$

Polinomiális hierarchia

Az előző észrevételeink megalapozzák a következő két definíciót.

Definíció. $\Sigma_i\mathcal{P} = \{\, L \mid \omega \in L \iff \exists^p x_1\ \forall^p x_2\ \exists^p x_3\ \ldots\ Q^p x_i\ T(\omega, x_1, x_2, \ldots, x_i)\}$, ahol a kvantorok p felső indexe egy polinom, ami azt jelenti hogy a kvantor után álló x a $\sum^{p(|\omega|)}$ halmazból kerül ki.

Definíció. $\Pi_i\mathcal{P} = \{\, L \mid \omega \in L \iff \forall^p x_1\ \exists^p x_2\ \forall^p x_3\ \ldots\ Q^p x_i\ T(\omega, x_1, x_2, \ldots, x_i)\}$

Megjegyzés. $\Sigma_0\mathcal{P} = \Pi_0\mathcal{P} = \mathcal{P}$, és $\Sigma_1\mathcal{P} = \mathcal{NP}$, $\Pi_1\mathcal{P} = \text{co}\mathcal{NP}$.

Észrevétel. A fenti definíciókhoz tartozó példákban szereplő kvantorsorozathoz új kvantorokat adva, amelyek „fal"-változókra hivatkoznak ugyanazon nyelv leírásához jutunk. Formálisan $\Pi_i\mathcal{P} \subset \Sigma_{i+1}\mathcal{P}$, $\Sigma_i\mathcal{P} \subset \Pi_{i+1}\mathcal{P}$.

Ezen észrevétel alapján kétféleképpen is leírhatjuk a polinomiális hierarchiának nevezett nyelvosztályt:

Definíció. Polinomiális hierarchia alatt a

$$\mathcal{PH} = \bigcup_{i\in\mathbb{N}} \Sigma_i\mathcal{P} = \bigcup_{i\in\mathbb{N}} \Pi_i\mathcal{P}.$$

nyelvosztályt értjük