ALGORITMUSOK ÉS BONYOLULTSÁGELMÉLET

Matematika MSc hallgatók számára

5. Előadás

Előadó: Hajnal Péter
Jegyzetelő: Domanovszki Bettina 2010 március 1.

További példák, redukciók

Példa: 3-SZÍNEZHETŐSÉG. Inputként egy G gráfot kapunk, melyről el szeretnénk dönteni, hogy 3 színnel jól színezhető e, vagyis $\chi(G) \le 3$?

$$\text{3-SZÍNEZHETŐSÉG} \in \mathcal{NP},$$

azaz alkalmas, nemdeterminisztikus, polinomiális idejű (INPUT méretben) T Turing-gép esetén

- $G \in$ 3-SZÍNEZHETŐSÉG $\Rightarrow$ létezik τ tanúszalag tartalom, hogy (G,τ) -t elfogadja T
- $G \notin$ 3-SZÍNEZHETŐSÉG $\Rightarrow$ nem létezik ilyen τ.

Legyen τ a csúcshalmaz egy 3-színezése.

Kérdés, hogy T Turing-gép τ szalagtartalma $V(G) \to \{1,2,3\}$ függvényt kódol-e. Ha igen, akkor az is kérdéses, hogy ez jó színezés e. Ha jó színezés, akkor ELFOGAD, különben NEM STIMMEL állapotba lép.

Mind a mai napig megoldatlan kérdés, hogy 3-SZÍNEZHETŐSÉG $co\mathcal{NP}$-beli-e.

Emlékeztető: Hajós tétel: G nem 3-SZÍNEZHETŐ, akkor és csakis akkor, ha K_4-ből felépíthető Hajós operációkkal, melyek a következők:
(H1): Él vagy csúcs hozzáadása egy gráfhoz.
(H2): Legyen x és x' két csúcsa $V(G)$-nek, amelyek nem szomszédosak. Az x-et es x'-t azonosítjuk egy $\hat{x}$ csúccsal. $\hat{x}$ szomszédsága x és x' G-beli szomszédságának uniója.
(H3): Legyen G és G' két gráf mindegyikben egy-egy éllel kitüntetve: $xy \in E(G)$, $x'y' \in E(G')$.
Legyen $\hat{G}$ a következő gráf: $V(\hat{G}) = (V(G) - \{x\}) \cup (V(G') - \{x'\}) \cup \{\hat{x}\}$, továbbá $E(\hat{G}) = E(G) - \{xy, x'y'\} \cup \{yy'\}$, ahol egy xz él $\hat{x}$ és z csúcsokra illeszkedik.

Vegyünk egy nemdeterminisztikus Turing-gépet, mely a Hajós tételt felhasználva dönti el a 3-SZÍNEZHETŐSÉG problémát. Adott τ tanúszalag tartalma egy $G_1, G_2, \dots, G_i, \dots, G_s$ gráfsorozat, ahol $G_s = G$ (input gráf), és ahol G_i gráfot úgy kódoljuk, hogy felsoroljuk, mely korábbi gráfokból, és mely operációkkal nyerjük. A Turing-gép ellenőrzi, hogy G_i egy K_4 gráf vagy az előzőekből Hajós-operációk valamelyikével nyerhető-e, ha igen, akkor ELFOGAD, ha nem, akkor NEM STIMMEL állapotba kerül. Csak τ hosszában polinomiális a futás, a közbülső gráfok esetében nagyon megnőhet.

Nagy kérdés, hogy az s, vagy a $\sum |G_i|$ becsülhető-e felülről jól, ahol $|G_i|$ a G_i gráf kódjának mérete. Ha ez G méretének polinomjával becsülhető, akkor a fenti gép $co\mathcal{NP}$ –beliségét bizonyítana.

Példa: KLIKK, ahol az input egy (G,t) pár, ahol G gráf, $t \in N$. El kell döntenünk, hogy az adott G gráfban létezik e t elemű klikk, azaz KLIKK={ (G,t) $\omega(G) \geq t$ }.

KLIKK$\in \mathcal{NP}$.

Legyen a τ tanúszalagon a csúcshalmaz egy K részhalmaza. Az algoritmus kiszámolja K elemszámát, és összehasonlítja t-vel. Ha nem egyenlők egymással, akkor NEM STIMMEL állapotba lép, ha pedig egyenlők, akkor megvizsgálja, hogy K összes csúcspárja összekötött e. Ha igen, akkor ELFOGAD, különben NEM STIMMEL.

Példa: FÜGGETLENPONTHALMAZ, mely a fenti példához hasonlóan szintén egy (G,t) párt kap inputként, és a probléma megoldásaként el kell döntenünk, hogy G-ben létezik e t elemű független ponthalmaz, azaz FÜGGETLENPONTHALMAZ ={ (G,t) $\alpha(G) \geq t$ }.

A fenti feladathoz hasonlóan veszünk egy T Turing-gépet. A τ tanúszalagon a csúcshalmaz egy K részhalmaza található, melynek elemszáma ha nem egyezik meg t-vel, akkor NEM STIMMEL állapotba lép, ha megegyezik, akkor T megvizsgálja, hogy K összes csúcspárja független e. Ha függetlenek, akkor ELFOGAD, különben NEM STIMMEL. Így kapjuk, hogy

FÜGGETLENPONTHALMAZ$\in \mathcal{NP}$.

Példa: LEFOGÁS szintén (G,t) párt kap, és a kérdés, hogy G - ben van e t elemű lefogó ponthalmaz, vagyis LEFOGÁS={ (G,t) $\tau(G) \leq t$ }.

Emlélkeztető: K klikk G-ben$\Leftrightarrow$ K független ponthalmaz $\overline{G}$-ben, ahol $E(\overline{G}) = \binom{V(G)}{2} - E(G)$

$\Leftrightarrow$ $\overline{K} = V(G) - K$ lefogó ponthalmaz $\overline{G}$-ben.

Az utolsó három probléma közötti kapcsolatok:

Adott (G,k), melyről el szeretnénk dönteni, hogy FÜGGETLENPONTHALMAZ-beli-e. Ekkor létezik egy T Turing-gép, mely a (G,k) párt $(\overline{G},k)$ párrá alakítja. Ez a Turing-gép lehet $\mathcal{P}$-beli, de kódolástól függően lehet $\mathcal{L}$-beli is, hiszen, ha szomszédsági mátrix segítségével kódoljuk G-t, akkor logaritmikus időben megkaphatjuk $\overline{G}$-t. Észrevehető, hogy $(G,k) \in$ FÜGGETLENPONTHALMAZ akkor és csakis akkor, ha $(\overline{G},k) \in$ KLIKK.
Azt is megfigyelhetjük, hogy $(G,k) \in$ FÜGGETLENPONTHALMAZ akkor és csakis akkor, ha $(G,|V(G)| - k) \in$ LEFOGÁS. Ezek alapján

LEFOGÁS $\in \mathcal{NP}$.

A következő definíció és példák összegzik ötletünket:

Definíció: $L, L' \subseteq \Sigma^*$, C bonyolultsági osztály. L redukálható L'-re, jelben $L \prec_C L'$, ha $\exists$ T kiszámítható Turing-gép, hogy:

- T C komplexitású
- $\forall \omega \in L \Leftrightarrow T(\omega) \in L'$

/Értelmezés: L' legalább olyan nehéz, mint L 'modulo C'/

Példa: $L_1 \prec_P L_2$ és $L_2 \in \mathcal{P} \Rightarrow L_1 \in \mathcal{P}$

Ennek igazolásához tekintsük a T Turing-gépet, mely az L_1-ről L_2-re történő redukciót végzi, valamint $\tilde{T}$-ot, mely L_2 $\mathcal{P}$-beliségét dönti el. Legyen adott az ω input. Ekkor

$$\omega \in \Sigma^n \xrightarrow{p} T(\omega) \in \Sigma^{p(n)} \xrightarrow{q(p(n))} \tilde{T}(T(\omega)) \in \Sigma^{q(p(n))}.$$

Mivel tudjuk, hogy ha p és q valós együtthatós polinomok, akkor $p + q \circ p$ is valós együtthatós polinom, vagyis L_1 is eldönthető polinom időben.

Példa: $L_1 \prec_{PSPACE} L_2$ és $L_2 \in \mathcal{P} \subseteq \mathcal{PSPACE} \Rightarrow$ nem szükségszerű, hogy $L_1 \in \mathcal{P}$.

Legyen T a redukcióért felelős Turing-gép, $\tilde{T}$ pedig ugyanaz, mint fent. Ha tetszőleges ω input tárigénye n, akkor $T(\omega)$ exponenciális időigényű is lehet, vagyis L_1 exponenciális időigényű is lehet. Azt viszont észrevehetjük, hogy $L_1 \in \mathcal{PSPACE}$.

Definíció: (Boole-)Hálózat a $\vec{G}$ irányított körmentes gráf, ahol a következők teljesülnek. Minden csúcs befoka legfeljebb 2. Az $F = \{v : 0 = befok(v)\}$ halmazt forrásnak nevezzük, így $V(\vec{G}) - F = \{v : 0 < befok(v)\} = \{v : befok(v)\, 1\, vagy\ 2\}$.

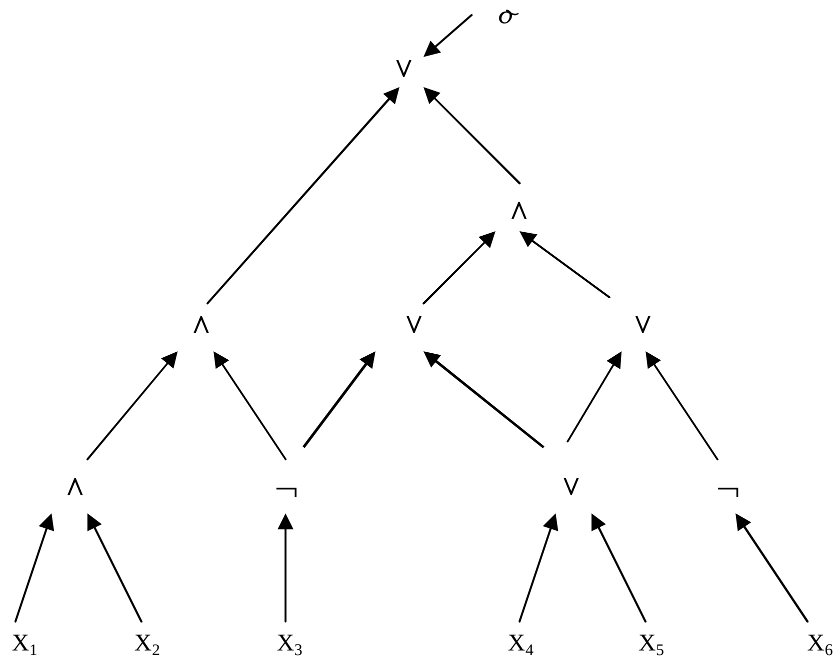
A változók halmaza a $\mathcal{V} = \{x_1, \ldots, x_n\}$ halmaz.

A gyökér, azaz az output csúcs $\sigma \in V(\vec{G})$, az a csúcs, melyre igaz, hogy $kifok(\sigma) = 0$.

Adott egy leképezés: $l : V(\vec{G}) \to C$, mely a csúcshalmazról a C címkehalmazra képez.

- $x \in F$, akkor $l(x) \in \{0, 1, x_1, \ldots, x_n\}$
- $x \notin F$, akkor $l(x) \in \{\wedge, \vee, \neg\}$, ahol $\{\wedge, \vee, \neg\}$ a De Morgan bázis. Ha $l(x) \in \{\wedge, \vee\}$, akkor $befok(x) = 2$, ha pedig $l(x) = \neg$, akkor $befok(x) = 1$.

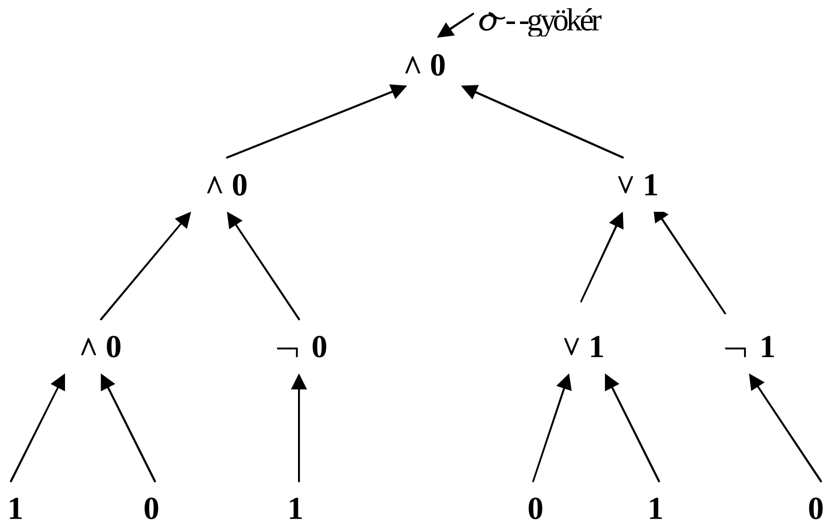
A következő ábrán látható módon le lehet rajzolni egy hálózatot.



Megjegyzés: Ha az alapgráf egy fagráf a gyökérfelé irányítva, akkor a hálózatot (Boole-) formulának hívjuk.

Definíció: Legyen C hálózat $\mathcal{V}=\{x_1,\ldots,x_n\}$ változókkal. Ekkor C–hez rendelhető egy $f_c:\{0,1\}^V\to\{0,1\}$ Boole-függvény (a $\mathcal{V}$–ből $\{0,1\}$-be képező függvényeket a változók kiértékelésének nevezzük), ahol $\{0,1\}^V=\{(\varepsilon_1,\ldots,\varepsilon_n):\varepsilon_i\in\{0,1\}\}$.

Ezeket a függvényeket a következőképpen értelmezzük: Rekurzió segítségével, vagyis „lentről-felfelé” a C hálózatnak megfelelő $\vec{G}$ gráf miden csúcsához $\{0,1\}$-beli értéket rendelünk: F elemeihez a kiértékelés alapján, a többi csúcshoz a hozzárendelt logikaijel és szomszédai értéke alapján. A kiértékeléshez rendelt érték az output csúcshoz rendelt érték.
A következő rajz a hálózat számításának szemléltetésére szolgál:



Példa: HÁLÓZAT-KIÉRTÉKELÉS adott (C,v) pár esetén ELFOGAD állapotba lép, ha v értéke a C hálózatban 1, azaz HÁLÓZAT-KIÉRTÉKELÉS $=\{(C,v): f_c(v)=1\}$. A hálózat definíciójából látszik, hogy a HÁLÓZAT-KIÉRTÉKELÉS polinom időben lefut. Azaz

HÁLÓZAT-KIÉRTÉKELÉS$\in\mathcal{P}$.

Példa: HÁLÓZAT-SAT adott C hálózatról eldönti, hogy létezik e olyan v, amelyhez C 1-et rendel, azaz HÁLÓZAT-SAT$=\{C:\exists v\ f_c(v)=1\}$.

Alkalmas, nemdeterminisztikus T Turing-gép τ tanúszalagján v található. T kiértékeli $f_c(v)$-t, amely ha 1-et ad, akkor ELFOGAD, különben NEM STIMMEL állapotba lép. Azaz

HÁLÓZAT-SAT $\in\mathcal{NP}$.

Teljes problémák

Emlékeztető: Legyen T nemdeterminisztikus Turing-gép, amely elfogadja L-et $s(n)$ tárban. Ekkor létezik $f(T)$ determinisztikus kiszámoló Turing-gép, ami ω inputon kiszámolja a G_ω

konfigurációgráfot és két speciális csúcsát (START, ELFOGAD) $\mathrm{O}(s(n)+\log n)$ tárigénnyel. Az ω L-beli akkor és csakis akkor, ha (G_ω, START, ELFOGAD)$\in$ ELÉRHETŐSÉG.

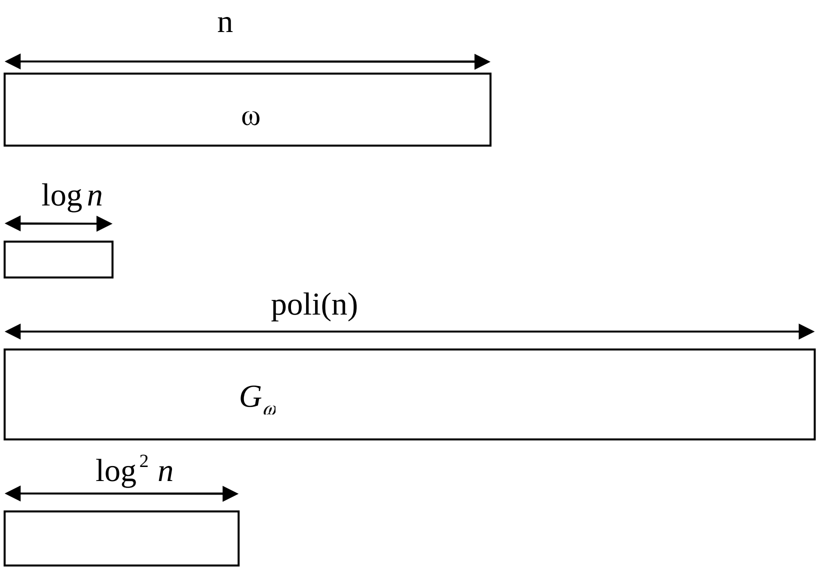
A fentiek átfogalmazása speciális esetben, amikor $s(n)=\log n$. Az L tetszőleges nyelv $\mathcal{NL}$-beli, hiszen az ELÉRHETŐSÉG is $\mathcal{NL}$-beli, vagyis L logaritmikus tárban redukálható az ELÉRHETŐSÉG-re, azaz $L \prec_L$ ELÉRHETŐSÉG.

Az általános esetben, ha $s(n) \geq \log n$ szép tárfüggvény, a determinisztikus kiszámoló Turing-gép $\mathrm{O}(s(n))$ tárigénnyel számol. Tehát az L tetszőleges $\mathcal{NSPACE}(s(n))$-beli nyelvre $L \prec_{\mathcal{SPACE}\, s(n)}$ ELÉRHETŐSÉG.

Azt is tudjuk, hogy ELÉRHETŐSÉG$\in \mathcal{SPACE}(\log^2 n)$. Összegezve:

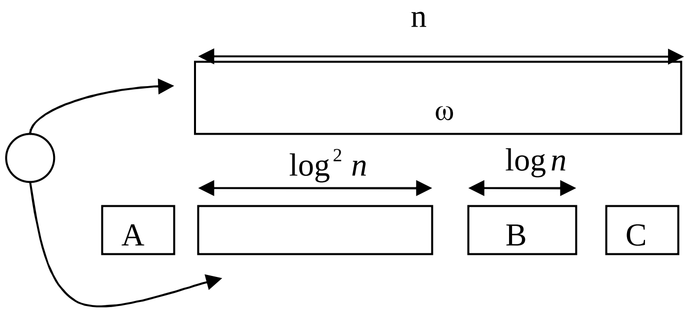
<u>1. Következmény</u>: $\mathcal{NL} \subseteq \mathcal{SPACE}(\log^2 n)$.

<u>Bizonyítás</u>: Legyen $L \in \mathcal{NL}$ tetszőleges nyelv, és azt szeretnénk belátni, hogy ekkor $L \in \mathcal{SPACE}(\log^2 n)$. Mivel $L \prec_{\mathcal{L}}$ ELÉRHETŐSÉG, ezért létezik T Turing-gép, mely a redukciót végzi. Az L eldöntése tetszőleges ω inputra a következőképpen zajlik:
$\omega \in L \xrightarrow{T} \vec{G}_\omega \xrightarrow{\tilde{T}} ELFOGAD\ vagy\ ELVET$, ahol $\tilde{T}$ az a Turing-gép, mely ELÉRHETŐSÉG-et $\log^2 n$ tárban eldönti. A lenti ábrából jól látható, hogy $\vec{G}_\omega$ nem fér bele a tárunkba, hiszen ennek leírásához polinom tárigény szükséges, de ha egy módosított, hatékonyabb algoritmust alkalmazunk, nincs is szükség $\vec{G}_\omega$ leírására.



Az IGAZI algoritmus csak a második fázist futtatja le, vagyis csak $\tilde{T}$-ot. Felveszünk egy B pótszalagot, és egy A és C számlálót. Ahhoz, hogy $\tilde{T}$-ot végre tudjuk hajtani, olvasnunk kell a $\vec{G}_\omega$-t, de ez nekünk nincs meg, ezért a munkaszem, és munkakéz a B pótszalag fölé megy, és elkezdi futtatni T-t (outputszalag nélkül). Nincs outputszalagunk, de T átmeneti függvénye alapján minden pillanatban tudjuk, mit írnánk le. Az A azt nézi, hogy $\vec{G}_\omega$ felett hol van a szem, míg C azt számolja, hogy hány output bitet írtunk ki. Ha ez a két érték megegyezik, akkor

megjegyzi az éppen leírandó output bitet, és visszatér $\tilde{T}$ futásához, tudva hogy $\tilde{T}$ mit olvasna inputjáról.
Az IGAZI algoritmus helyigénye $\mathrm{O}(\log^2 n)$.



2. Következmény: Az $s(n)$ szép tárfüggvény esetén $\mathcal{NSPACE}(s(n)) \subseteq \mathcal{SPACE}(s^2(n))$.
Bizonyítás: Hasonlóan megy, mint az előző következmény bizonyítása.

3. Következmény: $\mathcal{NPSPACE}=\mathcal{PSPACE}$.
(Gyakorlaton szerepelt részletesen.)

Definíció: C és C' bonyolultsági osztályok. Ekkor az L C'-teljes a C osztályra, ha $L \in C$, és ha minden $L' \in C$ nyelvre $L' \prec_{C'} L$.
/Értelmezés: L az egyik legnehezebb (modulo C') C-beli nyelv/

Példa: A definícióból észrevehető, hogy az ELÉRHETŐSÉG $\mathcal{L}$-teljes $\mathcal{NL}$-re

Definíció: L $\mathcal{NP}$-teljes, ha $\mathcal{P}$ teljes $\mathcal{NP}$-re, vagyis ha $L \in \mathcal{NP}$, és minden $L' \in \mathcal{NP}$ nyelvre $L' \prec_P L$.

Megjegyzés: Ha L $\mathcal{NP}$-teljes, és ha $L \in \mathcal{P}$, akkor $\mathcal{NP}=\mathcal{P}$. Ennek igazolásához a kölcsönös tartalmazást kell belátni. Az $\mathcal{NP} \supseteq \mathcal{P}$ irányú tartalmazás triviális, a másik irány egy korábbi észrevételünk.

Definíció: $L \equiv_C L'$ akkor és csakis akkor, ha $L \prec_C L'$ és ha $L' \prec_C L$.

Lemma:
i. $\prec_P$ tranzitív.
ii. $\prec_{\mathcal{L}}$ tranzitív.
iii. $\equiv_P$ és $\equiv_L$ ekvivalenciareláció.

Bizonyítás:
i. Nyilvánvaló.
ii. Lásd 1. Következmény bizonyítása.
iii. Nyilvánvaló i. és ii.-ből.