Kiegészítő anyag az Algoritmuselmélet tárgyhoz III.

(a Rónyai–Ivanyos–Szabó: Algoritmusok könyv mellé)

Friedl Katalin
BME SZIT
friedl@cs.bme.hu

2008. június 9.

A bonyolultságelmélet alapjai: a P és az NP osztály

Egy algoritmust (elméleti szempontból) hatékonynak mondunk, ha lépésszáma felülről becsülhető a bemenet hosszának egy polinomjával, azaz ha $f(n)$ jelöli az algoritmus maximális képésszámát az n hosszú bemeneteken, akkor $f(n) = O(n^k)$ teljesül valamilyen k pozitív konstansra. Természetesen, ha a kimenet mondjuk exponenciális hosszú, akkor az algoritmus nem lehet polinom lépésszámú. Ez a helyzet például ha az x pozitív egész bementből a 2^x számot akarjuk kiszámolni, hiszen a bemenet hossza, azaz, az x leírásához szükséges bitek száma $\lceil \log(x+1) \rceil$, míg a kimenet hossza $x+1$ bit. Azonban előfordulhat, hogy bár a kimenet rövid (1 bit), mégsem ismert a feladatra hatékony algoritmus. (Sőt olyan is előfordul, hogy egyáltalán nincs is algoritmus, de most ezzel az utóbbi esettel nem foglalkozunk – majd az MSc képzésben.)

Mostantól csak olyan típusú feladatokkal fogunk foglalkozni, amelyekre a válasz 1 bit, ezek az úgynevezett *eldöntési problémák.* Ilyenek például, hogy

- egy pozitív egész szám prímszám-e,
- egy gráfban van-e Hamilton-kör,
- egy gráf két adott s és t pontja között vezet-e út.

1. Definíció. *Egy eldöntési problémához tartozó* nyelv *azoknak a bemeneteknek a halmaza, amelyekre a válasz* igen. *A lehetséges bemeneteket (amik tehát vagy beletartoznak L-be vagy nem),* szavaknak *hívjuk.*

Az előző példákból az alábbi nyelveket kapjuk

- PRÍM nyelv, ami a prímszámokból áll,
  PRÍM= $\{m > 0 : m \text{ prímszám}\}$
  (Itt a szavak a bitsorozatok, amikre, mint kettes számrendszerben felírt számokra tekintünk.)

- H a Hamilton-kört tartalmazó gráfokból álló nyelv,
  H$= \{G : G$ irányítatlan gráf, van benne Hamilton-kör$\}$
  (Itt a szavak bitsorozatok, ha a szó hossza n^2, akkor ezt mint egy $n \times n$ mátrixot tekintjük, a gráf pedig az a gráf, aminek ez a szomszédossági mátrixa.)

- ÚT az olyan gráfok nyelve, amiben van s és t között út,
  ÚT $= \{(G, s, t) : G$ irányítatlan gráf , s, t a G csúcsai, G-ben vezet s-ből t-be út$\}$

2. Definíció. *Jelölje* P *azoknak a nyelveknek a halmazát, amelyekhez van olyan algoritmus, ami minden x bemenetre eldönti, hogy x a nyelvbe tartozik vagy sem, és az algoritmus lépésszáma $O(|x|^k)$ valamely k pozitív számra. (Itt $|x|$ az x bemenet hosszát jelöli, k független x-től.)*

Ezt úgy is mondjuk, hogy van a nyelvet polinom időben felismerő (eldöntő) algoritmus.

Az előző példákat vizsgálva könnyen látszik, hogy ÚT $\in$ P, mert egy szélességi bejárással a nyelvbe tartozás eldönthető és ez egy polinom idejű algoritmus. Az is igaz, hogy PRÍM $\in$P, de ez nem túl egyszerű, sokáig megoldatlan kérdés volt. A H nyelvről nem ismert, hogy P-ben van-e, azaz, hogy van-e rá polinom időben működő algoritmus (bizonyos speciális gráfosztályok esetén van ilyen).

3. Definíció. *Jelölje* NP *azoknak az L nyelveknek a halmazát, amelyekre teljesül a következő: van olyan $k > 0$ konstans és L_p nyelv úgy, hogy $L_p \in$ P valamint*

$$x \in L \quad \Leftrightarrow \quad \exists y,\ |y| = O(|x|^k) \text{ és } (x, y) \in L_p.$$

Ez azt jelenti, hogy ha $x \in L$, akkor van egy polinom hosszú bizonyíték (ez az y), amire az (x, y) pár együtt polinom időben igazolja, hogy $x \in L$. Az y-t az x tanújának vagy súgásnak is szokták hívni. Ha viszont $x \notin L$, akkor nem lehet ilyen ,,polinom időben meggyőző" bizonyíték.

1. Megjegyzés. *Az* NP *a* nemdeterminisztikus polinom idő *rövidítése, és azt fejezi ki, hogy a definícióban szereplő y-t nem feltétlenül tudjuk hatékonyan megtalálni. Az* NP*-be tartozáshoz azonban elegendő, hogy a tanú létezik amennyiben $x \in L$ és ha megkapjuk y-t, akkor gyorsan tudjuk ellenőrizni, hogy ez valóban jó tanú. Ezért is szokás súgásnak hívni, mert elég, ha valaki súg nekünk egy y szót (utána a helyességét már magunk is tudjuk polinom időben ellenőrizni).*

Példák

- H $\in$ NP, mert egy $G \in$ H gráfhoz jó tanú a gráf csúcsainak egy, a Hamilton-kör mentén való felsorolása. Mivel ez a tanú a gráf csúcsait sorolja fel, ezért valóban polinom hosszú lesz. Az L_p nyelv az olyan $(G, v_1, \ldots, v_n)$ bemenetekből áll, ahol G egy irányítatlan gráf, $v_1, \ldots, v_n$ pedig a G csúcsainak egy olyan permutációja, amire fennáll, hogy $\{v_1, v_2\}$, $\{v_2, v_3\}$, …, $\{v_{n-1}, v_n\}$, $\{v_n, v_1\}$ mindegyike éle a gráfnak. Világos, hogy ha adott a G gráf és a $v_1, \ldots, v_n$ sorrend, akkor valóban polinom időben ellenőrizhető, hogy a $(G, v_1, \ldots, v_n)$ bemenet benne van L_p nyelvben.

- Legyen ÖSSZETETT$= \{m > 0 : m$ egész, de nem prímszám$\}$.
  Könnyű látni, hogy ÖSSZETETT $\in$ NP, mert legyen

  $$L_p = \{(m, d) : 1 < d < m,\ d \text{ osztója } m\text{-nek}\}.$$

  Erre teljesül, hogy $L_p \in$ P, egy m összetett számhoz jó tanú az m-nek egy valódi d osztója. Ennek hossza legfeljebb annyi, mint m hossza, azaz $k = 1$.

- Legyen 3SZÍN $= \{G : G$ egy irányítatlan egyszerű gráf, aminek csúcsai kiszínezhetők 3 színnel $\}$.
  Itt jó tanú lesz a gráf csúcsainak egy 3 színnel színezése, ami, ha n csúcsa van a gráfnak, akkor leírható n darab 1 és 3 közötti számmal,

  $$L_p = \{(G, c_1, \ldots, c_n) : 1 \leq c_i \leq 3 \text{ egy jó színezése } G \text{ csúcsainak}\}.$$

  Ekkor $L_p \in$ P.

4. Definíció. *Egy L nyelv $\overline{L}$* komplementere *azokból a szavakból áll, amik nincsenek L-ben, azaz $\overline{L} = \{x : x \notin L\}$.*

Általában azt tételezzük fel, hogy a szavak a véges hosszú bitsorozatok, ilyenkor $\overline{L}$ pontosan azokból a bitsorozatokból áll, amelyek nincsenek L-ben.

5. Definíció. *Jelölje* co NP *az* NP*-beli nyelvek* komplementereiből *álló halmazt, azaz* co NP $= \{L : \overline{L} \in \text{NP}\}$.

A definíció miatt, míg az NP nyelvek esetében a definíció szerint a nyelvbe tartozásra van polinom hosszú, polinom időben ellenőrizhető bizonyíték, a co NP nyelveknél arra van rövid bizonyíték, ha valami nem tartozik a nyelvbe.

1. Állítás. P $\subseteq$ NP *és* P $\subseteq$ co NP.

Bizonyítás: Ha $L \in$ P, akkor választhatjuk az L_p nyelvet L-nek és y legyen az üres szó (azaz ilyenkor nincs szükségünk arra, hogy megsejtsünk egy bizonyítékot, polinom időben el fel tudjuk ismerni magát az L nyelvet is). $\square$

A számítástudomány talán legfontosabb (és legnépszerűbb) nyitott kérdése, hogy vajon P és NP megegyezik vagy nem. Ha P $\subset$ NP valódi tartalmazás, akkor ez azt jelentené, hogy vannak olyan kérdések, amikre van rövid (polinom hosszú) bizonyíték, de ezt a bizonyítékot nem lehet polinom időben megtalálni, azaz egy jó válasz ellenőrzése algoritmikusan lehet lényegesen egyszerűbb, mint egy jó válasz megtalálása (ami köznapi logikával elég hihetőnek tűnik). Mint látni fogjuk, sok nevezetes probléma bonyolultsága múlik ezen a kérdésen. Ennek is köszönhető, hogy az ezredfordulón a Clay Mathematics Institute – 6 másik matematikai problémával együtt – a P $\stackrel{?}{=}$ co NP problémára is 1 millió dolláros díjat tűzött ki (lásd www.claymath.org/millennium/).

Ahhoz, hogy jobban megérthessük ezt a problémát, további fogalmakra lesz szükség. Nyelvek bonyolultságának összehasonlítására szolgál a polinomiális visszavezetés, vagy más néven Karp-redukció.

6. Definíció. *Legyen L_1 és L_2 két nyelv. Az L_1 nyelv* Karp-redukciója (polinomiális visszavezetése) *az L_2 nyelvre egy olyan polinom időben számolható f függvény, ami minden szóhoz egy szót rendel, és teljesül rá, hogy*

$$x \in L_1 \Leftrightarrow f(x) \in L_2.$$

A jelölése: $L_1 \prec L_2$.

A jelölés arra utal, hogy, mint majd mindjárt látni fogjuk, az L_2 nyelv algoritmikusan legalább olyan nehéz, mint az L_1.

2. Állítás. *Ha $L_2 \in \mathrm{P}$ és $L_1 \prec L_2$, akkor $L_1 \in \mathrm{P}$.*

Bizonyítás: A feltétel szerint van egy polinomiális $\mathcal{A}$ algoritmus, ami tetszőleges z szóról eldönti, hogy $z \in L_2$ vagy $z \notin L_2$. Legyen x egy szó, amiről azt akarjuk tudni, hogy L_1-be tartozik-e. Erre egy jó algoritmus, ha előbb kiszámoljuk $f(x)$-et és utána erre alkalmazzuk $\mathcal{A}$-t. Mivel $x \in L_1 \Leftrightarrow f(x) \in L_2$, ezért az $\mathcal{A}$ által az $f(x) \overset{?}{\in} L_2$ kérdésre adott válasz megegyezik az általunk keresett válasszal az $x \overset{?}{\in} L_1$ kérdésre. Az eljárás lépésszáma: $f(x)$ kiszámolása $O(|x|^k)$ lépés valamely k-ra, és ezért $f(x)$ hosszára igaz, hogy $|f(x)| = O(|x|^k)$. Az $f(x)$ bemeneten $\mathcal{A}$ lépésszáma $O(|f(x)|^\ell)$ valamely ℓ-re mert $\mathcal{A}$ polinom idejű algoritmus, így tehát a lépésszám összesen $O(|x|^k) + O(|f(x)|^\ell) = O(|x|^{k\ell})$, ami valóban polinomja a bemenet $|x|$ hosszának. $\square$

3. Állítás. *Ha $L_2 \in \mathrm{NP}$ és $L_1 \prec L_2$, akkor $L_1 \in \mathrm{NP}$.*

Bizonyítás: Jelölje $L_{2p} \in \mathrm{P}$ azt a polinom időben felismerhető nyelvet, amit az NP definíciójából L_2-re kapunk és legyen f a Karp-redukció. Ekkor $L_{1p} = \{(x, y) : (f(x), y) \in L_{2p}\}$ jó lesz L_1-hez, hiszen egyrészt

$$x \in L_1 \quad \Leftrightarrow \quad f(x) \in L_2 \quad \Leftrightarrow \quad \exists y \quad (f(x), y) \in L_{2p} \quad \Leftrightarrow \quad \exists y \quad (x, y) \in L_{1p},$$

másrészt pedig y polinom hosszú x hosszához képest, amit az előzőhöz hasonlóan láthatunk: y polinom hosszú $f(x)$ hosszához képest, mert y az $f(x)$-nek tanúja. Továbbá $f(x)$ polinom hosszú az $|x|$-hez képest, mert f polinom időben számolható, tehát $|y|$ polinomiális $|x|$-ben.

Az kell még, hogy $L_{1p} \in \mathrm{P}$, de ez következik abból, hogy (x, y)-ból $(f(x), y)$ polinom időben számolható és a Karp-redukció definíciója szerint $L_{2p} \in \mathrm{P}$. $\square$

4. Állítás. *Ha $L_1 \prec L_2$, akkor $\overline{L_1} \prec \overline{L_2}$.*

Bizonyítás: Vegyük észre, hogy ha f egy $L_1 \prec L_2$ Karp-redukció, akkor ugyanez a függvény egyben egy $\overline{L_1} \prec \overline{L_2}$ Karp-redukció is, hiszen $x \in L_1 \quad \Leftrightarrow \quad f(x) \in L_2$ ugyanazt jelenti, mint $x \notin L_1 \quad \Leftrightarrow \quad f(x) \notin L_2$. $\square$

5. Állítás. *Ha $L_2 \in \mathrm{co\,NP}$ és $L_1 \prec L_2$, akkor $L_1 \in \mathrm{co\,NP}$.*

Bizonyítás: Mivel $L_1 \prec L_2$, ezért az előző állítás szerint $\overline{L_1} \prec \overline{L_2}$ is teljesül. A definíció szerint $L_i \in \mathrm{co\,NP}$ pontosan akkor, ha $\overline{L_i} \in \mathrm{NP}$. A feltétel szerint $\overline{L_2} \in \mathrm{NP}$, tehát a 3. állítás miatt $\overline{L_1} \in \mathrm{NP}$, azaz $L_1 \in \mathrm{co\,NP}$ $\square$

6. Állítás. *Ha $L_1 \prec L_2$ és $L_2 \prec L_3$, akkor $L_1 \prec L_3$, azaz a $\prec$ reláció tranzitív.*

Bizonyítás: Legyen f az első és g a második Karp-redukció függvénye. Ekkor $g(f(x))$ egy $L_1 \prec L_3$ Karp-redukció. Ehhez azt kell csak meggondolni, hogy ha $|f(x)| = O(|x|^k)$ és $|g(z)| = O(|z|^\ell)$, akkor $|g(f(x))| = O(|x|^{k\ell})$, ami $|x|$-nek polinomja és $x \in L_1 \quad \Leftrightarrow \quad f(x) \in L_2 \quad \Leftrightarrow \quad g(f(x)) \in L_3$. $\square$

7. Definíció. *Egy L nyelv* NP-nehéz, *ha minden $L' \in \mathrm{NP}$ nyelvre teljesül, hogy $L' \prec L$.*

8. Definíció. *Egy L nyelv* NP-teljes, *ha az L nyelv NP-nehéz és $L \in \mathrm{NP}$.*

Szemléletesen egy NP-nehéz nyelv legalább olyan nehéz mint bármelyik NP-beli nyelv (de lehet sokkal nehezebb is), míg az NP-teljes nyelvek az NP osztály legnehezebb nyelvei, hisz maguk is NP-ben vannak, és legalább olyan nehezek, mint bármely NP-beli nyelv.

A $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$ kérdés eldöntéséhez elegendő lenne egy NP-teljes nyelvről megmutatni, hogy P-ben van, hiszen ha az L nyelv NP-teljes, akkor minden $L' \in \mathrm{NP}$ nyelvre teljesül, hogy $L' \prec L$, és ha ugyanekkor $L \in \mathrm{P}$ is igaz, akkor a 2. állítás értelmében $L' \in \mathrm{P}$ minden $L' \in \mathrm{NP}$ nyelvre.

Ezért a hátralevő részben NP-teljes nyelvekkel foglalkozunk. Történetileg az első NP-teljességi bizonyítást Stephen Cook és Leonid Levin egymástól függetlenül, nagyjából egy időben, a 70-es évek elején adta. Az ebben szereplő nyelv bizonyos fajta logikai fomulákról szól. A későbbiekben azután az NP-teljes nyelvek száma gyorsan nőtt, már a 70-es évek végén megtöltöttek egy egész könyvet.

Ha már van legalább egy NP-teljes nyelvünk, akkor a további NP-teljességi bizonyítások a következő séma szerint készíthetőek:

1. Belátjuk, hogy $L \in \mathrm{NP}$
2. Egy tetszőleges NP-teljes L' nyelvre megmutatjuk, hogy $L' \prec L$.

Ez a séma a Karp-redukció tranzitivitása (6. állítás) miatt működik, hiszen az NP-teljesség definíciója szerint minden $L'' \in \mathrm{NP}$ nyelvre $L'' \prec L'$ és így $L' \prec L$-ből következik, hogy $L'' \prec L$ is teljesül minden $L'' \in \mathrm{NP}$ nyelvre.

Kiindulási nyelvként bizonyítás nélkül fogadjuk el, hogy a már korábban látott 3SZÍN NP-teljes. (Mint már említettük, nem ez volt az első bizonyítottan NP-teljes nyelv, de nekünk jó kezdőpont lesz.)

1. Tétel. *A* 3SZÍN *nyelv* NP*-teljes.*

2. Megjegyzés. *A hasonlóan definiált* 2SZÍN *nyelvre* 2SZÍN $\in \mathrm{P}$*, hiszen az, hogy egy gráf színezhető-e 2 színnel egyszerűen ellenőrizhető (pl. szélességi bejárással kiszínezhetők a csúcsok két színnel, ha egyáltalán van ilyen színezés).*

A 3SZÍN nyelv nehézségének felhasználásal már be tudjuk bizonyjtani más nyelvekről, hogy NP-teljesek. Először azt szeretnénk megmutatni, hogy gráfban a legnagyobb független ponthalmaz méretének meghatározása is algoritmikusan nehéz feladat. Ehhez előbb ezt át kell fordítani egy eldöntési problémává. Technikai okok miatt ezt úgy célszerű csinálni, hogy az eldöntési problémában nem a maximum értékére kérdezünk rá, hanem arra, hogy van-e elég nagy független ponthalmaz. Legyen

$$\textsc{maxftl} = \{(G, k) : G \text{ egy gráf, amiben van } k \text{ darab független pont}\}.$$

2. Tétel. *Ha lenne egy $\mathcal{A}$ polinom idejű algoritmus, ami felismeri a* MAXFTL *nyelvet, akkor polinom időben meg is lehet határozni egy adott gráfban a maximális független ponthalmaz méretét.*

Bizonyítás: Ha az n pontú G gráfban keresünk maximális független ponthalmazt, akkor alkalmazzuk $\mathcal{A}$-t először a $(G, \lceil n/2 \rceil)$ párra. Ha a válasz „nem”, azaz nincs $\lceil n/2 \rceil$ független pont, akkor próbálkozzunk a $(G, \lceil n/4 \rceil)$ párral. Ellenkező esetben pedig, a $(G, \lceil 3n/4 \rceil)$ párral. Hasonlóan folytatva bináris kereséssel meghatározhatjuk azt a legnagyob k értéket, amire $(G, k) \in$ MAXFTL. Az eljárás során az $\mathcal{A}$ algoritmus alkalmazásainak száma $\Theta(\log n)$ és mindegyik polinom lépésig tart, tehát összesen is polinom idejű algoritmust kapunk. □

3. Megjegyzés. *Ha nem csak a méretét akarjuk meghatározni, hanem meg is akarunk találni egy legnagyobb független ponthalmazt G-ben, az is megoldható lenne polinom időben az $\mathcal{A}$ algoritmus felhasználásával.*

Most megmutatjuk, hogy a MAXFTL nyelv felismerése valószínűleg nehéz.

3. Tétel. *A* MAXFTL *nyelv* NP*-teljes.*

Bizonyítás: MAXFTL $\in$ NP, hiszen k olyan pontja a gráfnak, ami között nem megy él megfelelő tanú arra, hogy $(G, k) \in$ MAXFTL.

Az NP-teljességhez mutatunk egy 3SZÍN $\prec$ MAXFTL Karp-redukciót. Ehhez egy olyan polinom időben számolható függvényt kell megadni, ami minden G gráfhoz hozzárendel egy (H, k) párt úgy, hogy $G \in$ 3SZÍN $\Leftrightarrow$ $(H, k) \in$ MAXFTL. Jelölje a G gráf pontjait $v_1, \ldots, v_n$. A H-nak $3n$ darab pontja lesz, legyenek ezek $a_1, \ldots, a_n$, $b_1, \ldots, b_n$, $c_1, \ldots, c_n$. Ha G-ben van él v_i és v_j között, akkor menjen él az a_i és a_j, a b_i és b_j, valamint a c_i és c_j pontok között is (azaz vettünk 3 példányt a G gráfból). Ezeken felül még minden $1 \le i \le n$ indexre húzzuk be mindhárom élet az a_i, b_i és c_i pontok között (azaz egy G-beli pont mindhárom példánya között), k pedig legyen n. Világos, hogy ez a konstrukció adott G-re polinom időben megvalósítható, hiszen a H-nak háromszor annyi pontja lesz mint G-nek, éleinek száma $3e + 3n$, ahol e jelöli G éleinek számát.

Előbb belátjuk, hogy ha $G \in$ 3SZÍN, akkor a H gráfban van n független pont: tekintsük a G gráf egy 3 színnel való színezését. A H-ban vegyük azt a ponthalmazt, ami az a_i-k közül azokat tartalmazza, amik a színezés első színosztályába tartozó pontoknak felelnek meg, a b_i-k közül a második színosztályba tartozó pontok megfelelőit, a c_i-k közül pedig a harmadik színosztály megfelelőit. Ekkor minden i-re az a_i, b_i, c_i csúcsok közül pontosan egyet választottunk, tehát összesen n darabot. Ezek a kiválasztott csúcsok függetlenek H-ban, mert a kiválasztott a_i-k G-ben függetlenek (hiszen azonos színűre voltak színezve G-ben) és ugyanez igaz a b_i-kre illetve a c_i-kre is. Továbbá, mivel egy G-beli pontnak csak egy példánya van benne, ezért a kiválasztott a, b és c pontok között sincsenek élek. Így tehát $(H, n) \in$ MAXFTL.

Még azt kell megmutatni, hogy ha $(H, n) \in$ MAXFTL, akkor $G \in$ 3SZÍN. A H gráf egy F független ponthalmaza minden i-re az a_i, b_i, c_i pontok közül csak egyet tartalmazhat. Mivel n pont van F-ben, ezért minden i-re az a_i, b_i, c_i pontok egyikét tartalmaznia is kell. Az F pontjaiból az a típusúak az eredeti G-nek egy független ponthalmazát adják, és ugyanez igaz a b, illetve a c típusú pontokra is. Egy H-beli n méretű független ponthalmazból az alábbi módon lehet a G gráf egy 3 színnel való színezését megkapni: A független ponthalmazba eső a típusú pontok G-beli megfelelőit színezzük az első színnel, a b típusú pontok megfelelőit a második színnel, a c típusú pontok megfelelőit a harmadik színnel. Ezzel G pontjainak megadtuk egy 3 színnel való színezését, tehát $G \in$ 3SZÍN. □

4. Megjegyzés. *Ha a nyelvet úgy definiáltuk volna, hogy olyan (G, k) párokból áll, melyekre a G-beli maximális független ponthalmaz mérete pontosan k, akkor azzal a nehézséggel néztünk volna szembe, hogy nem világos mi lenne egy jó tanú a nyelvbe tartozásra. Ebben az esetben k pont megadása nem elég, mert bár azt tudjuk polinom időben ellenőrizni, hogy az adott k pont valóban egy független halmazt alkot, de mi garantálja azt, hogy ennél több pontból álló független ponthalmaz nincs a gráfban?*

A MAXFTL nyelvhez hasonlóan definiálhatjuk a legnagyobb teljes részgráfhoz tartozó nyelvet,

$$\text{MAXKLIKK} = \{(G, k) : G \text{ egy gráf, amiben van } k \text{ pontú teljes részgráf}\}.$$

Erre is hasonlóak igazak, mint a MAXFTL nyelvre.

4. Tétel. *Ha lenne egy $\mathcal{A}$ polinom idejű algoritmus, ami felismeri a* MAXKLIKK *nyelvet, akkor polinom időben meg is lehetne határozni egy adott gráfban egy maximális méretű teljes részgráf pontjainak számát.*

Bizonyítás: A bizonyítás itt is hasonlóan megy, a legnagyobb klikk mérete az $\mathcal{A}$ algoritmus segítségével bináris kereséssel meghatározható. □

5. Tétel. *A* MAXKLIKK *nyelv* NP*-teljes.*

Bizonyítás: Az világos, hogy MAXKLIKK $\in$ NP, mert az a k pont, amik által feszített részgráf egy teljes k pontú gráf, polinom méretű és polinom időben ellenőrizhető tanú.

Az NP-teljesség bizonyításához mutatunk egy MAXFTL $\prec$ MAXKLIKK Karp-redukciót. Legyen $f(G, k) = (\overline{G}, k)$, ahol $\overline{G}$ a G gráf komplementerét jelöli. Ez egy, a gráf méretében polinom időben számolható függvény és nyilván

$$(G, k) \in \text{MAXFTL} \quad \Leftrightarrow \quad (\overline{G}, k) \in \text{MAXKLIKK}.$$

$\square$

További NP-teljes nyelvek (bizonyítás nélkül)

- H $= \{G : G$ irányítatlan gráfban van Hamilton-kör$\}$
- HÚT $= \{G : G$ irányítatlan gráfban van Hamilton-út$\}$
- s-tHÚT $= \{(G, s, t) : G$ irányítatlan gráfban van olyan Hamilton-út, aminek végpontjai s és $t\}$

6. Tétel. *A* RÉSZGRÁFIZO *nyelv* NP*-teljes, ahol*

$$\text{RÉSZGRÁFIZO} = \{(G_1, G_2) : \exists G' \leq G_2,\ G' \simeq G_1\},$$

(azaz a G_2 gráfnak van a G_1 gráffal izomorf részgráfja).

Bizonyítás: Az NP-beliséget mutatja, ha tanúként megadjuk, hogy a G_1 melyik csúcsának a G_2 melyik csúcsa kell, hogy megfeleljen. Ez a tanú a bemenet hosszában (azaz a két gráf megadásának hosszában) polinomiális, és polinom időben lehet ellenőrizni, hogy ez valóban a G_1 és G_2 egy részgráfja közötti izomorfimus. (Nem elegendő csak a G_1-nek megfelelő csúcsok halmazát megadni, mert izomorfizmus találása a tudomány mai állása szerint nehéznek látszik, ld. a tétel utáni megjegyzést. Viszont azt ellenőrizni, hogy a csúcsok között adott megfeleltetés jó-e nem áll másból, mint hogy minden pontpárra ellenőrizzük, hogy ha G_1-ben van közöttük él, akkor G_2-ben is.)

Az NP-teljesség bizonyításához mutassuk meg hogy van H $\prec$ RÉSZGRÁFIZO Karp-redukció. Ehhez definiáljuk azt a függvényt, ami minden n pontú G gráfhoz a (C, G) párt rendeli, ahol C egy n pontú kör. Ez a függvény kiszámolható polinom időben. Mivel G-ben pontosan akkor van Hamilton-kör, ha van benne C-vel izomorf részgráf, ezért $G \in$H $\Leftrightarrow (C, G) \in$ RÉSZGRÁFIZO. $\square$

A H nyelv helyett használhattuk volna például a MAXKLIKK nyelvet is.

5. Megjegyzés. *A hasonlónak tűnő*

$$\text{GRÁFIZO} = \{(G_1, G_2) : G_1 \simeq G_2\}$$

nyelvre igaz, hogy NP*-ben van, de nem ismert, hogy* NP*-teljes-e, vagy hogy esetleg van-e rá polinom idejű algoritmus.*

Néhány fontos, nem gráfokról szóló NP-teljes eldöntési feldat (bizonyítás nélkül)

- 3DH: Vegyünk három azonos méretű véges halmazt, $|A| = |B| = |C|$, és tekintsünk néhány olyan három elemű halmazt, melyek mindhárom halmazból egy-egy elemet tartalmaznak. Kérdés, hogy kiválasztható-e ezekből a 3 elemű halmazokból néhány úgy, hogy az $A \cup B \cup C$ alaphalmaz minden eleme pontosan egy kiválasztott halmazban legyen benne.

A nyelv formális leírása:

$$\begin{aligned} \text{3DH} &= \{(A,B,C;F_1,F_2,\cdots,F_n) : |A| = |B| = |C|,\ F_i \subseteq A \times B \times C, \\ &\quad \exists I \subseteq \{1,2,\ldots,n\},\quad \text{az } F_j \text{ halmazok } (j \in I) \text{ az } A \cup B \cup C \\ &\quad \text{minden elemét pontosan egyszer fedik le}\}. \end{aligned}$$

- X3C: Itt egyetlen A véges alaphalmaz van, és ennek adottak bizonyos 3 elemű részhalmazai. Az eldöntési feladat itt is az, hogy ezekből néhányat ki tudunk-e választani úgy, hogy A minden eleme pontosan egy kiválasztott halmazban legyen benne.

$$\begin{aligned} \text{X3C} &= \{(A;F_1,F_2,\ldots,F_n) : \exists I \subseteq \{1,2,\ldots,n\},\ \cup_{j\in I} F_j = A \\ &\quad \text{és } j \in I\text{-re az } F_j \text{ halmazok diszjunktak}\}. \end{aligned}$$

- RH (részhalmazösszeg): A döntési feladat: adottak $s_1, s_2, \ldots, s_n$ pozitív egész számok, és még egy b pozitív egész szám is. Kérdés, hogy az s_i-k közül kiválasztható-e néhány úgy, hogy összegük épp b-vel legyen egyenlő.

  A megfelelő nyelv:

$$\begin{aligned} \text{RH} &= \{(s_1,s_2,\ldots,s_n;b) : s_i, b > 0 \text{ egészek }, \\ &\quad \exists I \subseteq \{1,2,\ldots,n\}, \sum_{j\in I} s_j = b\}. \end{aligned}$$

- PARTÍCIÓ: A döntési feladatban adottak $s_1, s_2, \ldots, s_n$ pozitív egész számok. A kérdés, hogy két részre oszthatóak-e úgy, hogy a két rész összege azonos legyen, azaz van-e olyan $I \subseteq \{1,2,\ldots,n\}$, melyre

$$\sum_{j\in I} s_j = \sum_{j\notin I} s_j,$$

  tehát

$$\begin{aligned} \text{PARTÍCIÓ} &= \{(s_1,s_2,\ldots,s_n) : s_i > 0 \text{ egészek }, \\ &\quad \exists I \subseteq \{1,2,\ldots,n\}, \sum_{j\in I} s_j = \sum_{j\notin I} s_j\}. \end{aligned}$$

- HÁTIZSÁK: Adottak az $s_1, s_2, \ldots, s_n$ és $v_1, v_2, \ldots, v_n$ pozitív egész számok, valamint b, k pozitív egészek. Kérdés, hogy van-e olyan $I \subseteq \{1,2,\ldots,n\}$, melyre $\sum_{j\in I} s_j \leq b$ és $\sum_{j\in I} v_j \geq k$. Kevésbé formálisan: van n tárgy, az i-ediknek s_i a súlya és v_i az értéke. Továbbá van egy hátizsákunk, aminek teherbírása b. Kérdés, hogy ki tudunk-e választani úgy néhány tárgyat, hogy összességében ne lépjük túl a hátizsák teherbírását, de legalább k értéket elpakoljunk. (Az eredeti változatban ez egy maximalizálási feladat, adott b súlykorláthoz határozzuk meg a lehető legnagyobb összértéket, ami elpakolható – a fenti ennek az eldöntési változata.)

- LÁDAPAKOLÁS: Adott n tárgy, az i-edik mérete $0 < s_i \leq 1$ racionális szám. Kérdés, hogy belepakolható-e az összes tárgy k darab ládába akkor, ha minden ládába legfeljebb 1 összméretnyi tárgyat tehetünk.
- EP (azaz egész értékű programozás): Tekintsük a $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$, $x_j \geq 0$ egyenlőtlenségrendszert, ahol a_{ij}, b_i adott egész számok. Kérdés, hogy ha adottak még a $c_j \quad (1 \leq j \leq n)$ egész számok is, akkor mennyi lesz a $\sum_{j=1}^{n} c_j x_j$ maximális értéke, ha minden x_j egész szám kell legyen. Az EP ennek a maximalizálási feladatnak az eldöntési változata.

6. Megjegyzés. *A hasonlóan definiálható* 2DH *nyelv esetében 2 elemű halmazok szerepelnek. Ezek felfoghatók egy páros gráf éleinek, és ilyen szemlélettel a nyelv pontosan azokat a páros gráfokat tartalmazza, melyekben van teljes párosítás, tehát* 2DH $\in$ P *(magyar módszer).*

7. Megjegyzés. *Az előzőhöz hasonló gondolatmenettel látható, hogy* X2C *azokból a gráfokból áll, amikben van teljes párosítás. Ismert, hogy a teljes párosítás létezése eldönthető polinom időben, így tehát* X2C $\in$ P.

8. Megjegyzés. *Ha az utolsó (EP) problémánál nem kötjük ki, hogy minden x_j egész szám kell legyen, akkor kapjuk a* lineáris programozás *(röviden* LP*) feladatot. Az ehhez tartozó eldöntési feladat a* P *osztályban van (és van rá több, a gyakorlatban is hatékonyan működő programcsomag). Viszont, ha megköveteljük, hogy a megoldás egész legyen, akkor már a feladat* NP*-teljes lesz. Ez utóbbi tény nem meglepő, mivel a korábban felsorolt nyelvek mindegyike megfogalmazható egész programozás feladatként is.*

Feladatok

1. Jelölje L_1 az irányítatlan összefüggő gráfokból álló nyelvet és L_2 a Hamilton-kört tartalmazó gráfokból álló nyelvet. Lehetséges-e, hogy $L_1 \prec L_2$, illetve hogy $L_2 \prec L_1$?

   Megoldás: Az, hogy egy gráf összefüggő-e, polinom időben eldönthető (pl. egy szélességi vagy mélységi bejárással), tehát $L_1 \in$ P. Tudjuk, hogy az L_2 nyelv NP-teljes (L_2 =H). Mivel P $\subseteq$ NP, ezért az NP-teljesség definíciója miatt $L_1 \prec L_2$.

   Másrészt viszont, ha létezik az $L_2 \prec L_1$ Karp-redukció, akkor a 2. állítás miatt $L_2 \in$ P, és így P = NP. Nem kizárt, hogy ez igaz legyen, de a tudomány mai állása szerint nem ismert, hogy P $\stackrel{?}{=}$ NP.

2. Igazolja, hogy ha a 3SZÍN nyelv benne van co NP-ben, akkor NP = co NP.

   Megoldás: Tudjuk, hogy a 3SZÍN nyelv NP-teljes, tehát minden $L \in$ NP nyelv visszavezethető rá, azaz $L \prec$ 3SZÍN. A feltétel szerint 3SZÍN $\in$ co NP, és ezért az 5. állítás miatt $L \in$ co NP, amiből NP $\subseteq$ co NP következik. Másrészt, ha $L \in$ co NP, akkor $\overline{L} \in$ NP, tehát $\overline{L} \prec$ 3SZÍN. A feltevés szerint 3SZÍN $\in$ co NP, ezért az 5. állítás miatt $\overline{L} \in$ co NP, azaz $L \in$ NP.

3. Ismert, hogy a síkgráfokból álló nyelv P-ben van. Legyen

$$\text{SÍKMAXKLIKK} = \{(G, k) \mid G \text{ egy síkgráf, amiben van } k \text{ pontú klikk } \}$$

Mutassa meg, hogy ez a nyelv NP-teljes, vagy mutassa meg, hogy a nyelv P-ben van.

Megoldás: Tudjuk, hogy egy síkgráfban nem lehet 4-nél több pontú teljes részgráf, hiszen K_5 nem rajzolható síkba. Ekkor viszont a nyelvbe tartozás az alábbi módon eldönthető:

Ellenőrizzük, hogy az adott gráf síkgráf-e, ha nem, akkor készen vagyunk (ezt a feladat szövege szerint meg tudjuk tenni polinom időben);

Ha $k > 4$, akkor a válasz ,,nem", és készen vagyunk;

Ha $k \leq 4$, akkor ellenőrizzük a gráf pontjainak az összes k elemű részhalmazát. Amennyiben találunk közöttük olyat, amin a gráf teljes, akkor a válasz ,,igen", egyébként pedig ,,nem". Mivel egy n pontú gráfból kevesebb mint $n^k \leq n^4$ féleképpen tudunk $k \leq 4$ pontot kiválasztani, és egy választás ellenőrzése $O(k^2) = O(4^2) = O(1)$ lépés, az algoritmusnak ez a része is polinomiális.

4. P-beli vagy NP-teljes az alábbi nyelv?

$$L = \{(G, k) : G \text{ gráf, amiben van legalább } k \text{ élű kör}\}$$

Megoldás: Belátjuk, hogy NP-teljes. Az NP-beliségre tanú egy ilyen kör, melyről ellenőriznünk kell, hogy tényleg kör-e G-ben és hogy tényleg legalább k éle van. Mindkettő megtehető polinom időben.

Ezután megadunk egy H $\prec L$ visszavezetést. Egy tetszőleges G gráfhoz rendeljük hozzá az $f(G) = (G, n)$ párt, ahol n a G pontjainak száma. Ekkor $G \in \text{H} \Leftrightarrow f(G) = (G, n) \in L$. Mivel f polinom időben számolható, ezért ez egy jó Karp-redukció.

5. P-beli vagy NP-teljes az alábbi nyelv?

$$L = \{G : G \text{ gráf csúcsai kiszínezhetőek 4 színnel}\}$$

Megoldás: Megmutatjuk, hogy az L nyelv NP-teljes. Az NP-beliségre tanú egy 4 színnel való színezés, aminek leírása polinom hosszú, az ellenőrzés során pedig csak azt kell megvizsgálni, hogy az élek végpontjai különböző színűek-e. Az NP-teljességhez megadunk egy 3SZÍN $\prec$ 4SZÍN visszavezetést. Egy tetszőleges G gráfból készítsük el azt a G' gráfot, amit úgy kapunk, hogy egy új x pontot hozzáveszünk a gráfhoz, és ezt az x pontot minden régivel összekötjük. Ez a konstrukció polinom időben végrehajtható. Világos, hogy ha $G \in$ 3SZÍN, akkor $G' \in$ 4SZÍN. A másik irányhoz tegyük fel, hogy G' kiszínezhető 4 színnel. Ekkor az x pont színe egyetlen másik pont színezésére sem használható, hiszen x minden más ponttal össze van kötve. Ez azt jelenti, hogy a G' gráf többi része (ami az eredeti G-vel izomorf) 3 színnel van kiszínezve, tehát $G \in 3\text{SZÍN}$.

6. Az L nyelv álljon azokból a G gráfokból, melyeknek csúcsai kiszínezhetőek 3 színnel úgy, hogy az egyik színt csak egyszer használjuk. P-beli vagy NP-teljes ez a nyelv?

   Megoldás: Megmutatjuk, hogy ez a nyelv P-ben van. Egy gráf akkor és csak akkor rendelkezik a kívánt tulajdonsággal, ha van egy olyan pontja, amit elhagyva páros gráfot kapunk. Ezt viszont tudjuk polinom időben ellenőrizni, mert annak eldöntésére, hogy egy gráf páros gráf-e, van polinom idejű algoritmus (egy szélességi bejárás mentén osztjuk a pontokat két osztálya, ha ellentmondásra jutunk, akkor a gráf nem páros), és ezt kell minden egyes pont elhagyása után a maradék gráfra végrehajtani.

7. Egy hivatal új épületbe fog költözni. Az épület minden emeletén ugyanakkora terület használható fel irodák kialakítására. Minden részleg megmondta, hogy összesen mekkora irodaterületre tart igényt. Azt akarjuk eldönteni, hogy meg lehet-e oldani a költözést úgy, hogy egyetlen részleg se legyen kettévágva, azaz egy részleg teljes egészében egy emeleten legyen (de egy emeletre kerülhet több részleg is). Igazolja, hogy a problémához kapcsolódó nyelv P-ben van, vagy azt, hogy a nyelv NP-teljes.

   Megoldás: Jelölje a kérdéses nyelvet L.

   $$\begin{aligned} L \quad = \quad & \{(a_1, a_2, \ldots, a_n; b; k) : a_i, b, k > 0 \text{ egészek, az} \\ & a_i \text{ számok beoszthatók } k \text{ diszjunk csoportba úgy, hogy a} \\ & \text{minde csoportban az elemek összege } \leq b\} \end{aligned}$$

   Itt a_i jelöli az i-edik részleg igényét, k az épület szintjeinek számát b pedig az egy szinten használható terület nagyságát.

   Ez a nyelv nyilván NP-ben van, hiszen egy beosztás lehet a tanú, aminek helyességét polinom időben ellenőrizni tudjuk (és persze a hossza is polinomiális).

   Megmutatjuk, hogy van $\text{PARTÍCIÓ} \prec L$ Karp-redukció.

   Legyen $(s_1, s_2, \ldots, s_n)$ a PARTÍCIÓ feladat egy lehetséges bemenete. Ennek feleltessük meg azt a feladatot, amikor a hivatal épületében 2 szint van, $b = (\sum_{s_i})/2$ és az i-edik részleg helyigénye $a_i = s_i$. Ezzel a megfeleltetéssel $(s_1, s_2, \ldots, s_n) \in \text{PARTÍCIÓ} \Leftrightarrow$ a költözés megoldható. Tehát az L nyelvre visszavezethető egy NP-teljes nyelv, és $L \in \text{NP}$, ezért L maga is NP-teljes.