ALGORITMUSOK ÉS BONYOLULTSÁGELMÉLET

Matematika MSc hallgatók számára

7. Előadás

Előadó: Hajnal Péter
Jegyzetelő: Karsai Dávid 2011. Március 22.

Emlékeztető. Egy $L \in_T \mathcal{NP}$ nyelvből logaritmikus tárral (így polinom időben) hálózatot konstruálhatunk, a hálózat „tetején" a bemenetet és a tanúszalagot kódoló polinom sok bit található, majd polinom kapun (amelyek szerepe a számítás polinom sok konfigurációja kódjainak kiszámítása) keresztül eljutunk egy kapuhoz, ami által kiszámított bit kódolja a gép utolsó ELFOGAD/ELVET állapotát.

1. További teljes nyelvek

1. Következmény.

- *(i) Tetszőleges $L \in \mathcal{NP}$ nyelvre $L \prec_{\mathcal{L}}$ HÁLÓZAT-SAT*
- *(ii) HÁLÓZAT-SAT $\mathcal{NP}$-teljes*
- *(iii) $\mathcal{P} = \mathcal{NP}$ akkor és csak akkor, ha HÁLÓZAT-SAT $\in \mathcal{P}$*

Bizonyítás. (i) $L \in_T \mathcal{NP}$, így egy tetszőleges $\omega \in L$ esetén létezik hozzá egy tanú: $\tau = (t_1, t_2, \ldots, t_{p(n)})$, amelyre $T(\omega, \tau)$ ELFOGAD állapotba jut. Azaz a múlt órán megkonstruált C hálózatra, $C(\lceil\omega\rceil, y_1, y_2, \ldots, y_{q(n)})$ az 1 értéket számolja ki, ha az y változók helyére a τ kódjának bitjeit írjuk. Megfordítva is igaz. Ha $C(\lceil\omega\rceil, y_1, y_2, \ldots, y_{q(n)})$-nek találunk egy kielégítését, akkor egy tanú kódját találjuk. Azaz $C(\lceil\omega\rceil, y_1, y_2, \ldots, y_{q(n)})$ kódjának legyártása (ami az emlélekeztető alapján $\mathcal{L}$-ben megoldható) egy jó redukció.

(ii) Az (i) részből és abból, hogy HÁLÓZAT-SAT$\in \mathcal{NP}$ (tanú egy kielégítő bemenet), következik, hogy HÁLÓZAT-SAT NP-teljes.

(iii) Mivel HÁLÓZAT-SAT $\in \mathcal{NP}$, így ha $\mathcal{P} = \mathcal{NP}$, akkor $\mathcal{P}$-beli is. Viszafelé, ha HÁLÓZAT-SAT$\in \mathcal{P}$, akkor tetszőleges $L \in \mathcal{NP}$ nyelvet redukáljunk HÁLÓZAT-SAT-ra, a redukált problémát $\mathcal{P}$-ben el tudjuk dönteni. A két lépés együtt is polinomiális és az L nyelv eldöntési problémáját oldja meg. Ebből $L \in \mathcal{P}$ következik, így $\mathcal{NP} \subseteq \mathcal{P}$, tehát $\mathcal{P} = \mathcal{NP}$ adódik. ■

2. Tétel (Cook—Levin-tétel). *SAT (CNF formula kielégíthetősége) $\mathcal{NP}$-teljes.*

Bizonyítás. Korábban már szerepelt, hogy SAT $\in \mathcal{NP}$, így elég adnunk egy visszavezetést HÁLÓZAT-SAT-ról SAT-ra, hiszen ekkor az előző következmény (i) pontja és a polinomiális redukció tranzitívitása alapján tetszőleges $L \in \mathcal{NP}$ nyelvet vissza tudunk vezetni SAT-ra. Ezt is két lépésben tesszük, a HÁLÓZAT-SAT-ot visszavezetjük BOOLE-EGYENLETRENDSZER-SAT-ra, majd azt SAT-ra.

Definíció. $\varphi_i(x_1, x_2, \ldots, x_n) = \psi_i(x_1, x_2, \ldots, x_n)$, $i = 1, 2, \ldots, \ell$ egyenletrendszert Boole-egyenletrendszernek nevezzük, ha φ_i és ψ_i Boole-formulák. Az $\{x_i\}_{i=1}^n$ változók egy 0-1/igaz-hamis értékadás az egyeneletrendszer megoldása, ha minden $i = 1, 2, \ldots, \ell$ esetén φ_i és ψ_i értéke ugyanaz.

BOOLE-EGYENLETRENDSZER-SAT az a nyelv, ami a megoldható/kielégíthető Boole-egyenletrendszerek kódját tartalmazza.

Legyen H egy hálózat. Minden csúcsával azonosítunk egy változót. Ez input-csúcsok esetén a csúcs címkéje. A többi csúcsra (kapukra) mind különváltozókat feleltetünk meg. Minden kapuhoz tartozik egy egyenlet

- $x_g = \neg x_h$, ha a g kapu negáció kapu és a h kapuból kapja inputját ($\overrightarrow{hg}$ él a hálózatban).
- $x_g = x_h \wedge x_{h'}$, ha a g kapu hálózatbeli címkéje konjunkció és inputjait h és h' kapukból kapja.
- $x_g = x_h \vee x_{h'}$, ha a g kapu hálózatbeli címkéje diszjunkció és inputjait h és h' kapukból kapja.
- $x_g = 1$, ha a g kapu a hálózat output kapuja.

Ezzel megkaptuk egy Boole-egyenletrendszert a hálózatból. Ha a hálózat 1-et számol ki egy értékadáson (az értékadás kielégíthetőséget bizonyít), akkor a kapuk által kiszámolt bitekkel együtt egy megoldását kapjuk az egyeneletrendszernek. Fordítva is igaz, ha az egyenletrendszer megoldásából kiemeljük az eredeti input változók értékadásait, akkor ezen a hálózat 1-et számol ki (sőt minden kapu a hozzárendelt változó megoldásbeli értékét számolja ki). Azaz a hálózatból legyártott egyenletrendszer megoldhatósága ekvivalens azzal, hogy a hálózat kielégíthető.

Az '=' jeleket '$\leftrightarrow$' logikai jelekre cserélve az egyes egyenleteknek megfelelő formulákat kapunk. Egy értékadás akkor és csak akkor teljesíti az egyenletet, ha igazzá teszi a hozzárendelt formulát. A kapott logikai kifejezések mindegyike legfeljebb három változót tartalmaznak. Könnyű őket CNF formára hozni. Ha az összes egyenletnek megfeleltetett CNF formulát 'és' logikai jellel összekapcsoljuk, szintén CNF formát kapunk. Így az egyenletrendszerhez hozzárendeltünk egy φ CNF formulát. A hozzárendelés $\mathcal{P}$-ben kiszámolható. Az egyenletrendszer megoldhatósága ekvivalens a formula kielégíthetőségével.

Vagyis „programunk" második redukcióját is megadtuk, a tételt bebizonyítottuk. ■

Definíció (kvantifikált Boole-formula probléma, QBF). Legyen a φ Boole-formula a következő alakú: $\varphi(x_1, y_1, x_2, y_2, \ldots, x_n, y_n)$. Ekkor a

$$\forall x_1 \exists y_1 \forall x_2 \exists y_2 \ldots \forall x_n \exists y_n \varphi(x_1, y_1, x_2, y_2, \ldots, x_n, y_n)$$

vagy igaz, vagy hamis. A probléma az, hogy döntsük el, hogy melyik.

3. Tétel. *QBF $\mathcal{PSPACE}$-teljes, vagyis*

- *(i) QBF $\in \mathcal{PSPACE}$,*
- *(ii)minden $L \in \mathcal{PSPACE}$ esetén $L \prec_{\mathcal{P}} QBF$.*

Bizonyítás. Azt látjuk be, hogy QBF $\mathcal{NPSPACE}$-teljes. Ez a tétel állítása, hiszen tudjuk, hogy $\mathcal{NPSPACE} = \mathcal{PSCPACE}$.

(i) Könnyen belátható.

(ii) Ahogy általában a tár korlátozott döntéseknél tettük, egy tetszőleges $L \in_T \mathcal{NPSPACE}$ nyelvre az ω L-hez tartozásának döntési feladatát visszavezethetjük az ELÉRHETŐSÉG eldöntésére a ($G_{T,\omega}$, START, ELFOGAD) input esetén, ahol $G_{T,\omega}$ a redukált konfigurációk gráfja:

$$\omega \in L \Leftrightarrow \text{ létezik START-ELFOGAD (irányított) út } G_{T,\omega}\text{-ban.}$$

$G_{T,\omega}$ egy 2^{n^α} pontszámú gráf. Egy csúcs kódja logaritmikus a csúcsok számában. Vagyis mindegyik csúcs n^α hosszban kódolható.

Persze nem az ELÉRHETŐSÉG inputját gyártjuk le. Ennek „költsége" $\mathcal{PSPACE}$, ami megegyezik L bonyolultságával, így nincs értelme. (A gondolatmenet $\mathcal{NL}$ és $\mathcal{L}$ viszonyában működött.)

Az „igazi" visszavezetés egy formulát gyárt le. Ehhez szükségünk lesz néhány jelölésre.

Jelölés. Legyen $N := n^\alpha$.

$u \overrightarrow{\sim} v$ jelölje azt, hogy alapgráfunkban u-ból elérhető (irányított értelemben) a v csúcs.

$u \overrightarrow{\sim}_{\leq i} v$ jelölje azt, hogy alapgráfunkban u-ból legfeljebb i lépéssel elérhetünk v-be.

Esetünkben az alapgráf $G_{T,\omega}$, pontszáma 2^N. Így $u \overrightarrow{\sim} v$ akkor és csak akkor, ha $u \overrightarrow{\sim}_{\leq 2^N} v$.

Az alapötletünk az, hogy felosztjuk az utunkat két részre egy középső k csúccsal, előbb csak egy k csúcsig megyünk el START-ból maximum 2^{N-1} lépésben, majd onnan szintén maximum 2^{N-1} lépésben ELFOGAD-ig.

Eddigi ötleteink formalizálva:

$$\begin{array}{lcl} \omega \in L & \Leftrightarrow & \text{START} \overrightarrow{\sim}_{\leq 2^N} \text{ELFOGAD}, \\ \text{START} \overrightarrow{\sim}_{\leq 2^N} \text{ELFOGAD} & \Leftrightarrow & \exists_V k (\text{START} \overrightarrow{\sim}_{\leq 2^{N-1}} k) \wedge (k \overrightarrow{\sim}_{\leq 2^{N-1}} \text{ELFOGAD}). \end{array}$$

Az összes csúcs (START,k,ELFOGAD) N (n-ben polinomiális) hosszú bitsorozattal kódolt. A $\exists_V k$ azt jelenti, hogy az összes csúcsot kódoló bitsorozatra vonakozólag létezik.

Gondolhatunk arra, hogy nincs mit tenni csak iterálni kell ezeket az ötleteket. Formulánkban két helyen is szerepel az $\overrightarrow{\sim}_{\leq 2^{N-1}}$ reláció. Iterálásnál (N mélységre van szükségünk) a formula nagysága exponenciálisan nagy lenne. Még egy ötletre van szükségünk. A fenti formulánk ekvivalens azzal, hogy

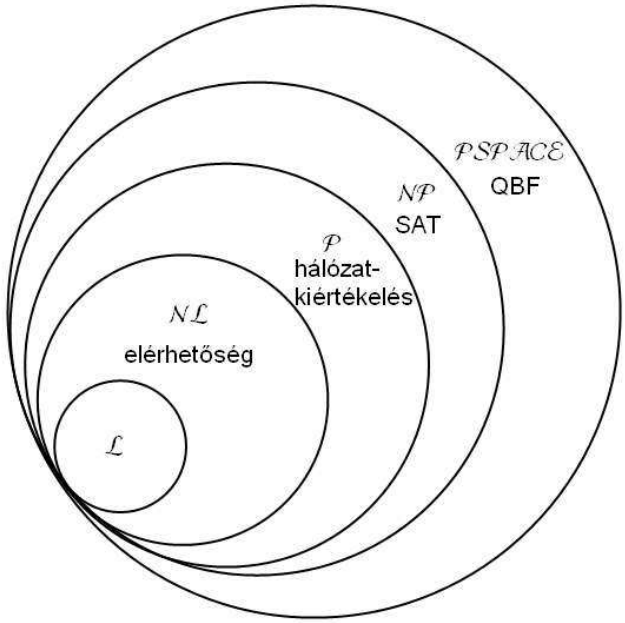
$$\exists_V k \forall_V c \forall_V c' : ((c = \text{START} \wedge c' = k) \vee (c = k \wedge c' = \text{ELFOGAD})) \rightarrow c \overrightarrow{\sim}_{\leq 2^{N-1}} c'.$$

Ezt a gondolatmenetet iterálva már a kapott formula mérete polinomiális lesz. $N = n^\alpha$ iterálás kell és mindegyik csak hozzáad a fomrulához egy polinomiális hosszú új részt. A bizonyítás befejezéseként csak ellenőrizzük, hogy az iteráció elindul: Az iteráció legmélyén lévő formulák: $u \overrightarrow{\sim}_{\leq 1} v$ azt jelentik, hogy '$u = v$ vagy u-ból vezet él v-hez'. Ez egy egyszerű (kvantor nélküli) Boole-formulával megfogalmazható reláció.

Az $\exists_V v$ jelölés csalóka. v N darab bittel kódolt, ezek mindegyike létezik kvantorral kötött. Úgy tűnik, hogy az alternáló kvantorok feltétele nem teljesül. Nincs probléma. Új változók bevezetésével a kvantorok alternálása megoldható. Ha az új változókat csak kvantifikált formában használjuk, akkor nem befolyásolják a formula logikai értékét. ■

Megjegyzés. A QBF feladat inputjában nem az volt a fontos, hogy létezik kvantor után minden kvantor következik és fordítva, hanem az, hogy kvantorok váltakozásának száma tetszőlegesen nagy lehet.

Ezzel a legfontosabb nyelvosztályokra találtunk teljes problémákat. Azaz olyan problémákat, amelyeke tükrözik a nyelvosztály teljes nehézségét. Ha ezekről a teljes problámákról tudunk valami „okosat" mondani, akkor az egész osztályról kapunk fontos információt.



Ábra osztályokról, és bennük teljes problémákról

1. ábra.

2. $\mathcal{NP}$-teljes nyelvek

$\mathcal{NP}$ különösen fontos osztály. A Millieneumi problémák egyike a $\mathcal{P}$ és $\mathcal{NP}$ osztályok viszonyának tisztázása. Erre az osztályra vonatkozó teljes problémák nagyon változatosak. Teljesen különböző matematikai kutatások során ilyen kérdésekbe ütközünk. Az alábbiakban a HÁLÓZAT-SAT és SAT problémákon túl is mutatunk $\mathcal{NP}$-teljes kérdéseket.

Az újabb $\mathcal{NP}$-teljes nyelvek felmutatásához nem kell a definícióig visszamenni. „Karp-elv": Ha T $\mathcal{NP}$-teljes probléma, $L \in \mathcal{NP}$, továbbá $T \prec_P L$, akkor L is $\mathcal{NP}$-teljes.

Definíció. Adott φ CNF formulában, ha minden klóz legfeljebb három literált tartalmaz, akkor $3^{\leq}$-SAT-nak, nevezzük a kielégíthetőségi problémáját.

Ha minden klóz pontosan három literált tartalmaz, akkor $3^{=}$-SAT-nak nevezzük a kielégíthetőségi problémáját.

4. Lemma. *$3^{=}$-SAT$\preceq$ $3^{\leq}$-SAT és $3^{\leq}$-SAT$\preceq$ $3^{=}$-SAT.*

Bizonyítás. Az első állítás triviális az előbbi definícióból. Nézzük, mi a helyzet, ha adott egy $3^{\leq}$-SAT probléma. Ez előáll olyan klózok 'és'-el való összekapcsolásaként, amikben maximum 3 literál van. Ha egy klózban pont három van, akkor azzal nem csinálunk semmit. Ha kettő van, például x és y benne a két literál, akkor ehelyett a klóz helyett két új klózt vezetünk be: $(x \vee y \vee z)$ és $(x \vee y \vee \neg z)$, ahol z egy új változó, ami nem szerepelt még sehol a CNF formulában. Mivel egy CNF formula pontosan akkor kielégíthető, ha minden klóz egyszerre kielégíthető, ezért ha a változtatás előtt kielégíthető volt, akkor ugyanazzal az értékadással az új két klóz is igaz lesz. Ha nem volt kielégíthető, akkor ezután sem lesz, mivel az új két klóz közül az egyik mindig igaz lesz, a másik pedig pontosan akkor lesz igaz, ha $(x \vee y)$ igaz, így $(x \vee y \vee z) \wedge (x \vee y \vee \neg z) = (x \vee y)$. Egy literált tartalmazó klózok esetén ugyanez az ötlet működik. ■

Az oda-vissza redukció léte (ha polinomiális számolás elhanyagolható) a két feladat ekvivalenciáját jelenti.

Feladat. $L \equiv_{\mathcal{P}} L'$, ha $L \preceq_{\mathcal{P}} L'$ és $L' \preceq_{\mathcal{P}} L$. Ekkor $\equiv_{\mathcal{P}}$ ekvivalenciareláció.

A továbbiakban a $3^{=}$-SAT és $3^{\leq}$-SAT problémákat 3-SAT-ként hivatkozzuk. Ha ezt látjuk, választhatunk melyik változata kényelmesebb számunkra.

5. Tétel. *SAT*$\preceq$ 3-*SAT.*

Bizonyítás. SAT$\preceq 3^{\leq}$-SAT redukciót írunk le.

Minden egyes klóz helyett új klózokat vezetünk be. Legyen C egy klóz: $(l_1 \vee l_2 \vee \cdots \vee l_n)$. Legyenek $y_0, y_1, \ldots, y_n$ új változók, és az új klózok pedig a következők:

$$\neg y_0, y_0 \vee l_1 \vee \neg y_1, \ldots, y_{i-1} \vee l_i \vee \neg y_i, \ldots, y_{n-1} \vee l_n \vee \neg y_n, y_n$$

Ekkor ha az eredeti klózt az ℓ_i-k értékadása nem tette igazzá (vagyis mindegyik l_i hamis), akkor az új klózt sem lehet az új változók alkalmas értékadásával igazzá tenni: elindulva az elejéről mindig csak egy lehetőségünk van a következő változó értékének megválasztására hogy igaz maradjon a kifejezés, és a legvégén y_n-nek is hamis értéket kell adnunk. Fordítva, ha létezik i, amire l_i igaz, akkor megvalósítható az új klózokban is olyan értékadás, hogy igaz legyen a klózok 'és'-sel való összekapcsolása is.

Minden klózra párhuzamosan elvégezve ezt az átalakítást (mindegyikhez diszjunkt y-változók halmazát véve), jó visszavezetést kapunk. ■

Mivel az előző visszavezetés polinom időben megtehető és 3-SAT $\in \mathcal{NP}$ (mivel a SAT speciális esete), ezért 3-SAT $\mathcal{NP}$-teljes probléma.