ALGORITMUSOK ÉS BONYOLULTSÁGELMÉLET

Matematika MSc hallgatók számára

10. Előadás

Előadó: Hajnal Péter
Jegyzetelő: Kovácsházi Anna 2011. Április 19.

Emlékeztető. $L \in_T \mathcal{P}$ akkor, ha létezik $\{C_n\}$ hálózat-sorozat, amely

(i) n inputbitből számol ki egy bitet,

(ii) csúcsainak száma/mérete kisebb, mint $p(n)$, valamely p polinomra,

(iii) a hálózat input bitjei Σ^ν elemeit kódolják ($n = (\log_2 |\Sigma|) \cdot \nu$, $\omega \in \Sigma^n$ input esetén ω kódja legyen $\lceil \omega \rceil$), továbbá $\omega \in L$ akkor és csakis akkor, ha $C_n(\lceil \omega \rceil) = 1$, azaz C_n az ω kódján az elfogadást/elvetést kódoló bitet számolja ki,

(iv) C_n $\mathcal{L}$-ben megkonstruálható 1^n-ből.

Megjegyzés. A visszafele irány is igaz. Azaz, ha az L nyelvhez van fenti típusú hálózat-sorozat, akkor ω-ból felírhatók az ω-t kódoló bitek, ha ezek száma ν, akkor C_ν a hálózatot is ki tudjuk számolni, végül kiértékelni. Az eredeő eljárás mutatja, hogy $L \in \mathcal{P}$.

1. SAT és hálózatok

Definíció. $L \in \mathcal{P}^{\text{nem-uniform}}$ akkor és csakis akkor, ha (i)-(iii) igaz. Azaz $\mathcal{P}$ fenti leírásából csak a hálózat-sorozat „uniformitását” nem követeljük meg.

Megjegyzés. A $\mathcal{P}^{\text{nem-uniform}}$ nyelvosztály „furcsa” az eddigiekhez képest. Nincs benne az eddig ismert legbővebb kiszámíthatósággal kapcsolatos osztályban, $\mathcal{S}$-ben, azaz a felsorolható nyelvek osztályában.

$$\begin{array}{ll} \mathcal{S} & \supseteq \mathcal{D} \supseteq \mathcal{EXPSPACE} \supseteq \ldots \supseteq \mathcal{NP} \supseteq \mathcal{P} \ldots \\ \not\subseteq & \\ \mathcal{P}^{\text{nem-uniform}} & \end{array}$$

Vegyük a természetes számok egy $\mathcal{B}$ bonyolult (nem felsorolható) részhalmazát. Legyen $L := \{1^\ell : \ell \in \mathcal{B}\}$.

L nyilván nincs benne $\mathcal{S}$-ben, hiszen ha ezt a halmazt fel tudnánk sorolni, akkor $\mathcal{B}$ elemeit is felsorolnánk.

$L \subset \{0,1\}^*$ benne van $\mathcal{P}^{\text{nem-uniform}}$-ban: Kétféle inputméret van. Egyik inputméretben semmit sem kell elfogadni, mert olyan hosszban egyik szó sem tartozik a nyelvhez. A másik inputméret pedig olyan, hogy csak csupa egyes inputot kell elfogadni. Az első esetben hálózatunk az első változót és negáltját ÉS-sel köti össze. A második esetben a hálózat az összes változót ÉS-sel köti össze. Észrevehetjük, hogy a kétféle hálózat önmagában nagyon egyszerű (mérete polinomiális, sőt lineáris), de nagyon bonyolultan, rapszódikusan változik. Mivel nem követeljük meg, hogy megkonstruálható legyen, így beleillik a modellünkbe.

Emlékeztető. A $SAT \in \mathcal{P}$-ből következik, hogy $\mathcal{P} = \mathcal{NP}$. Sejtés: $SAT \notin \mathcal{P}$. Azaz SAT nem számolható ki $\mathcal{L}$-ben megkonstruálható polinomiális hálózat-sorozattal.

Kérdés: $SAT \in \mathcal{P}^{\text{nem-uniform}}$? Sejtés: $SAT \notin \mathcal{P}^{\text{nem-uniform}}$. Azaz SAT nem-uniform polinomiális méretű hálózat-sorozattal sem számolható ki.

Az alábbiakban ezt a sejtést támasztjuk alá.

1. Lemma. *A következők ekvivalensek:*

(i) $L \in \mathcal{P}^{nem\text{-}uniform}$.

(ii) Létezik az ω input hosszában polinomiális T nemdeterminisztikus (tanúszalagos) Turing-gép, és létezik $\{t_n\}_{n=1}^{\infty}$ tanúsorozat, hogy $\omega \in L$ akkor és csak akkor, ha $T(\omega, t_{|\omega|}) = 1$ (azaz ω-n a számítás ELFOGAD állapotba vezet).

(iii) $L \preceq_{\mathcal{P}}^{Turing} S$, *valamely $S \subset \Sigma^*$ alkalmas ritka nyelvre (definíciók alább).*

Definíció. Egy S nyelv ritka, ha benne az n hosszú szavak száma n-ben polinomiális (azaz nagy n-re jóval kevesebb, mint a teljes lehetőségek $|\Sigma|^n$ exponenciális száma). Formálsian van olyan $q(n)$ polinom, hogy $|S \cap \Sigma^n| \leq q(n)$.

Definíció (Turing-redukció). $L \preceq_{\mathcal{P}}^{\text{Turing}} S$ akkor és csak akkor van olyan polinomiális idejű Turing-gép, amely ω L-hez tartozását dönti el. Számolása alatt „?" állapotba mehet a gép, aminek van egy kérdező szalagja is. A speciális állapotba kerülés egy kérdés: a kérdező szalagra az előző kérdés óta felírt karaktersorozatról derül ki (egy lépésben) az S-hez tartozás. Az „egy lépésben kiderül" alatt azt értjük, hogy a rákövetkező konfigurációban BENNE-VAN vagy pedig a NINCS-BENNE állapotba lesz, aszerint, hogy a kérdező szalag tartalma S-beli vagy sem.

Az S-et szokták orákulumnak is nevezni. S tulajdonképpen egy „meg nem írt szubrutin", ami ha valamilyen módon megírható, mondjuk megírása könnyű, akkor L sem lehet nehéz.

Bizonyítás. (i)⇒(ii): Minden n-hez tartozik egy hossz, amelyben bitekkel kódoljuk az Σ^n elemeit. Legyen ez ν. Ha L nem-uniform polinomiális időben van, akkor tartozik hozzá egy $\{C_n\}$ hálózat-sorozat. Legyen t_n a C_ν hálózat kódja. A T Turing-gép a tanú-szalag tartalmából kiolvassa C_ν-t, az input-szalag tartalmát $0-1$ bitekkel kódolja, majd a C_ν-t ennek tartalmán kiértékeli. Ha 1-et kap ELFOGAD, ha 0-t ELVET állapotba kerül.

(ii)⇒(iii): Tegyük fel, hogy létezik egy L-et elfogadó nem determinisztikus, polinomiális gép az input hosszától függő tanúval.

Először definiálunk egy ritka nyelvet Legyen $S := \{(1^k, t_k^{\leq \ell}) : k \in \mathbb{N},\ \ell \leq t_k \text{ hossza}\}$. S a következő alakú: $\underbrace{1\ 1\ 1\ \ldots\ 1}_{k \text{ db}}$; „$t_k$ első ℓ' karaktere" ($\ell' \leq \ell$).

2. Állítás. *Az előbb definiált S nyelv ritka.*

Valóban: Az 1-es blokk elemszáma meghatározza, hogy melyik tanúnak a része következik.

3. Állítás. *Adott $\omega \in L$ esetén $\mathcal{P}$ időben az S-hez tartozásra kérdezéssel $t_{|\omega|}$ kiszámítható, aminek ismerete megoldja az L-hez tartozás kérdését.*

Valóban: Beírjuk az $1^{|\omega|}$ karaktert, majd megkérdezzük, hogy $\Sigma := \{\sigma_1, \ldots, \sigma_k\}$ valamely eleme S-beli-e.

$$\begin{aligned} 1^{|\omega|};&\quad \sigma_1 \stackrel{?}{\in} S \\ 1^{|\omega|};&\quad \sigma_2 \stackrel{?}{\in} S \\ &\quad\vdots \\ 1^{|\omega|};&\quad \sigma_k \stackrel{?}{\in} S \end{aligned}$$

Amikor igent kapunk, akkor megtaláltuk a tanúnak az első bitjét, b_1-et. Ugyanezen ötlet segítségével a második bit is megtalálható a következő kérdések után:

$$\begin{aligned} 1^{|\omega|};&\quad b_1\sigma_1 \stackrel{?}{\in} S \\ 1^{|\omega|};&\quad b_1\sigma_2 \stackrel{?}{\in} S \\ &\quad\vdots \\ 1^{|\omega|};&\quad b_1\sigma_k \stackrel{?}{\in} S \end{aligned}$$

Amikor igent kapunk, akkor megtaláltuk a tanúnak a második bitjét, b_2-t. Ha nem kapunk igent, akkor a teljes tanú elöttünk áll. Folytatva az eljárást megtaláljuk (a szükségszerűen polinomiális hosszú) tanút. Ezek után polinom időben eldönthető T-vel, ω L-hez tartozása.

(iii)⇒(i): Konstruálunk egy hálózat-sorozatot.

Egy determinisztikus/nem-determinisztikus Turing-gép esetén már láttuk, hogyan kell azt szimuláló hálózat-sorozatot konstruálni. Ugyanazt az utat követjük.

Az egyetlen különbség, hogy a hálózatnak kezelnie kell azt, hogy a konfigurációt kódoló bitsorozat egy szakaszáról eldöntse S-beli elemet kódol-e. Esetünkben ez polinomiálisan sok elem valamelyikével való megegyezésnek a tesztelését jelenti.

Ha S „szép" lenne, akkor akár egy uniform hálózattal megtehetnénk ezt. S azonban lehet bonyolult. A nem-uniformitás miatt azonban nincs probléma. Adott n inputméret esetén fel tudjuk sorolni az S nyelv szóbajövő $\sigma_1, \sigma_2, \ldots, \sigma_{q(n)}$ elemeit kódoló biteket. Legyen ξ a kérdéző szalag tartalma. Hálózatunkba a

$$(\xi = \sigma_1) \vee (\xi = \sigma_2) \vee \ldots \vee (\xi = \sigma_{q(n)})$$

formulát kell beírni, ahol $\xi = \sigma_i$ a

$$((\xi_1 \wedge (\sigma_i)_1) \vee (\neg\xi_1 \wedge \neg(\sigma_i)_1)) \wedge ((\xi_2 \wedge (\sigma_i)_2) \vee (\neg\xi_2 \wedge \neg(\sigma_i)_2)) \wedge \ldots$$

formulával fejezhető ki. A hálózathoz kell a ritka nyelvből kiolvasott $\sigma_1, \sigma_2, \ldots, \sigma_{q(n)}$ bitsorozatok, ami a nem-uniformitást eredményezi. ■

Most visszatérünk az erdeti kérdésünkre. SAT megoldható-e nem-uniorm polinomiális hálózat-sorozattal. Célunk, hogy belássuk, ha ez így lenne, akkor annak következményei lennének.

4. Tétel (Karp—Lipton, Sipser tétele). *Ha* $SAT \in \mathcal{P}^{\textit{nem-uniform}}$, *akkor*

$$\Pi_2\mathcal{P} \subset \Sigma_2\mathcal{P}.$$

Emlékeztető. $L \in \Pi_2$ akkor és csakis akkor, ha létezik polinomiális T Turing-gép, hogy $\omega \in L$ akkor és csakis akkor, ha

$$\forall\, x(\in \Sigma^{p(n)}), \exists y(\in \Sigma^{p(n)}) : T(\omega, x, y) = 1,$$

azaz T futása ELFOGAD állpottal ér véget.

$L \in \Sigma_2$ akkor és csakis akkor, ha létezik polinomiális T Turing-gép, hogy $\omega \in L$ akkor és csakis akkor, ha

$$\exists\, x, \forall\, y : T(\omega, x, y) = 1.$$

Megjegyzés. A tétel konklúziójából ($\Pi_2\mathcal{P} \subset \Sigma_2\mathcal{P}$) következik, hogy $\Sigma_2\mathcal{P} = \Pi_3\mathcal{P} = \Pi_4\mathcal{P} = \Sigma_4\mathcal{P} = \ldots$, tehát a polinomiális hierarchia a második szintre esik össze. Valóban: $\Pi_3\mathcal{P}$-beli nyelv jellemezhető mint azon ω-k halmaza, amelyekre $\forall\, x, \exists\, y, \forall\, z : T(\omega, x, y, z) = 1$. $\Pi_2\mathcal{P} \subset \Sigma_2\mathcal{P}$ alapján a $\forall$ és a $\exists$ kvantor felcserélhető, azaz nylvünk leírható úgy is, mint $\exists\, \widetilde{y}, \forall\, \widetilde{x}, \forall\, z : T(\omega, \widetilde{x}, \widetilde{y}, z) = 1$. Ez a leírás már Σ_2 elemeként írja le a $\Pi_3\mathcal{P}$-beli nyelvet.

Azt sejtik, hogy a polinomiális hierarchia valódi hierarchia. A sejtés jóval ero|sebb mint a $\mathcal{P} \subsetneq \mathcal{NP}$ sejtés, ami a hierarhia legaljáról szól. Aki ezt az erősebb sejtést is elfogadja, azok számára a tétel úgy interpretálható, hogy nem valószínű, hogy SAT megoldható legyen polinomiális méretű hálózat-sorozattal (akkor sem, ha ez hektikusan váltakozhat).

Bizonyítás. Legyen L egy tetszőleges nyelv $\Pi_2\mathcal{P}$-ből. Azaz legyen T egy polinomiális Turing-gép, $\omega \in L$ akkor és csakis akkor, ha

$$\forall\, x\ \exists\, y :\ T(\omega, x, y) = 1.$$

Bevezetve az $\omega, x := \omega^+$ jelölést, legyen $\widetilde{L} = \{\omega^+ : \exists\, y T(\omega^+, y) = 1\} \in \mathcal{NP}$. Speciálisan $\widetilde{L}$-et polinomiális redukcióval a SAT-ra vezetjük (a standard $\mathcal{NP}$-teljes nyelv). Feltevésünk szerint SAT benne van $\mathcal{P}^{\text{nem-uniform}}$-ban, így SAT kiszámolható egy $\{C_n\}$ hálózat-sorozattal. Az ω^+ inputból egy R redukciós algoritmus $R(\omega^+)$ CNF formulát gyárt: $\omega^+ \in \widetilde{L}$ akkor és csakis akkor, ha $R(\omega^+) \in SAT$, ami akkor és csakis akkor következik be, ha $C_n(\ulcorner R(\omega)^+ \urcorner) = 1$-et számolja ki, ahol n az $\ulcorner R(\omega)^+ \urcorner$ hossza.

Célunk: Az $\omega \in L$ információt átírjuk

$$\exists\, \widetilde{x}\ \forall\, \widetilde{y} :\ \widetilde{T}(\omega, \widetilde{x}, \widetilde{y}) = 1$$

alakra (olyan alakra, ami $\Sigma_2\mathcal{P}$-beliséget mutat).

Első próbálkozásban tippeljük meg C_n-et:

$$\exists\, \ulcorner C_n \urcorner\ \forall\, x \in \Sigma^{p(|\omega|)} :\ C_n(\ulcorner R(\omega, x) \urcorner) = 1,$$

ahol C_n olyan Turing-gép, ami $\ulcorner C_n \urcorner$-ből megkapja C_n-et, és kiértékeli azt, majd teszteli, hogy az 1 bitet számoltuk-e ki.

Ha a C_n tipp jó, akkor az ekvivalenciában minden stimmel. Ha a C_n tipp rossz, két eset lehetséges. A $C_n = 0$ esetén nincs baj. Problémát az okozhat, ha egy rossz tipp esetén $C_n = 1$-et kapunk. Az ötlet finomításra szorul.

5. Lemma. *Legyen C_n egy hálózat. Ekkor polinomiális időben kiszámolhatunk egy $\widetilde{C}_n$ hálózatot, ami a következőket tudja.*

(i) ha C_n a SAT-ot döntötte el, akkor $\widetilde{C}_n$ is,

(ii) ha $\widetilde{C}_n$ az 1 bitet számolja ki, akkor kielégíthető formulát kódol az input.

Lemma bizonyítása: Ha van egy hálózatunk, amivel a SAT-ot el tudjuk dönteni, továbbá adott egy formula minek van kielégítő kiértékelése, akkor ki is tudjuk számolni egy kielégítő kiértékelést.

A φ kódjából könnyen ki lehet számolni $\varphi|_{x_1=0}$ és $\varphi|_{x_1=1}$ két formula kódját. Mindkét formulát bevezethetjük hálózatunk inputjába. A két alkalmazás közül legalább az egyik esetben kielégíthető formulát kapunk. Legyen α_1 egy „jó" rögzítése x_1-nek. Kiszámoljuk a $\varphi|_{x_1=\alpha_1,x_2=0}$ és $\varphi|_{x_1=\alpha_1,x_2=1}$ két formula kódját. Mindkét formulát bevezethetjük hálózatunk inputjába. A két alkalmazás közül legalább az egyik esetben kielégíthető formulát kapunk. Legyen α_2 egy „jó" rögzítése x_2-nek. E"s így tovább, amíg $\alpha_1, \alpha_2, \ldots$ biteket meg nam kapjuk.

Amennyiben hálózatunk a SAT problémát számolta ki és fomrulánk kielégithető volt $(\alpha_1, \alpha_2, \ldots)$ egy kielégítő kiértékelés. Ha feltevésünk nem igaz, akkor $(\alpha_1, \alpha_2, \ldots)$ bitekről semmi információnk sincs.

A hálózat végén kiértékeljük formulánkat az $(\alpha_1, \alpha_2, \ldots)$ biteken. A kiértékelés adja az outputot.

Amennyiben hálózatunk a SAT problémát számolta ki és formulánk kielégithető volt a végső bit 1 lesz. Ha a feltételünk nem teljesül, akkor a végső kiértékelés biztosít arról, hogy a végső 1 bit esetén a kielégíthetőség garantált.

(Lemma bizonyítása)■

A tétel bizonyításának befejezése már nyilvánvaló. $\omega \in L$ akkor és csakis akkor, ha

$$\exists^{\lceil C_n \rceil}\ \forall x:\ \widetilde{C}_n(\lceil R(\omega, x)\rceil) = 1.$$

Az első próbálkozás zsákutcája nem lehetséges, ha C_n-et rosszul is tippeljük meg a lemma alapján elvégzett $\widetilde{C}_n$ átírás már nem hibázhat a rossz irányban. Azaz a fenti alak L $\Sigma_2\mathcal{P}$-beliségét igazolja. ■

A fenti tétel a $\mathcal{P}^{\text{nem-uniform}}$ jellemzései alapján a következő változatban is kimondható.

6. Tétel. *Ha $SAT \preceq_{\mathcal{P}}^{Turing} S$, ahol S ritka, akkor $\Pi_2\mathcal{P} \subseteq \Sigma_2\mathcal{P}$.*

A Karp-redukció felfogható Turing-redukcióként is: orákulum nélkül számolunk, egyetlen kérdést generálunk és csak ezt az egy kérdést tehetjük fel S-nek, amire kapott válasz az outputja. Mahaney vette észre, ha a Karp—Lipton, Sipser tételének feltételben szereplő Turing-redukaciónál erősebb Karp-redukciót tesszük fel, akkor erősebb következményt igazolhatunk.

7. Tétel (Mahaney-tétel). *Ha $SAT \preceq_{\mathcal{P}}^{Karp} S$, akkor $\mathcal{P} = \mathcal{NP}$.*

A bizonyítás előtt átfogalmazzuk a tétel feltételét és állítását. Ehhez néhány definícióra lesz szükségünk.

Definíció. $\{0,1\}^{\leq n} = \{\sigma \in \{0,1\}^* : |\sigma| \leq n\},\ x, y \in \{0,1\}^{\leq n},\ x \preceq y$ rendezés akkor és csakis akkor, ha

(i) Van olyan i, hogy $j < i$ esetén $x_j = y_j$, továbbá $x_i < y_i$ (x és y első eltérő bitje 0 az x-ben, míg 1 y-ban)

(ii) vagy y valódi kezdőszelete x-nek.

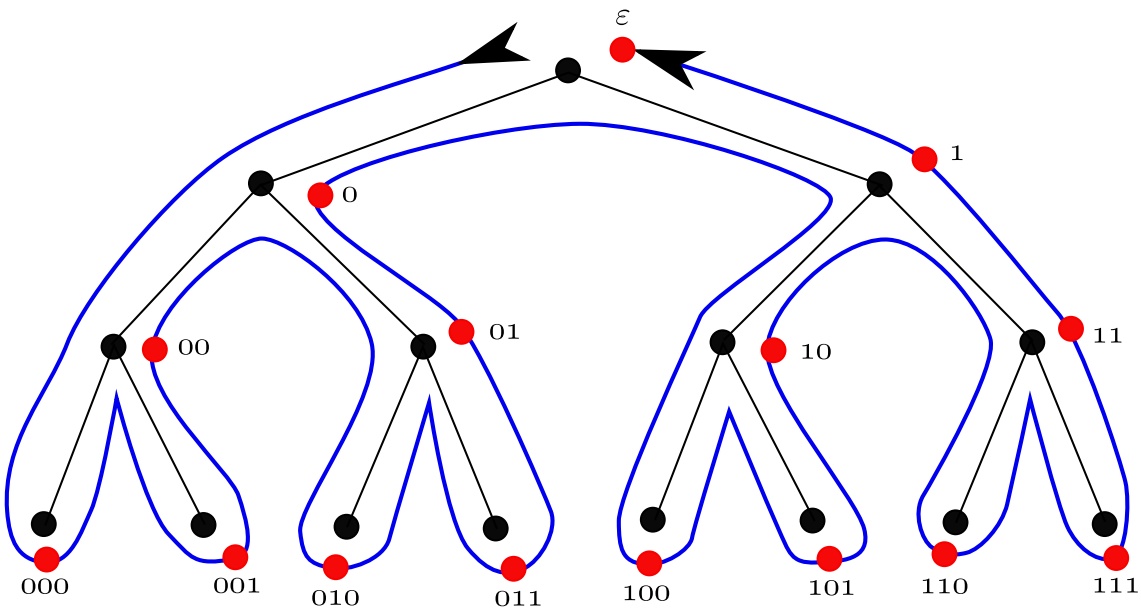
Példa. $\{0,1\}^{\leq 3}$ rendezett sorrendje:

$$000,\ 001,\ 00,\ 010,\ 011,\ 01,\ 0,\ 100,\ 101,\ 10,\ 110,\ 111,\ 11,\ 1,\ \varepsilon,$$

ahol ε a 0-hosszú, üres szó. Ez minden hosszra a rendezés utolsó eleme, hiszen minden más szónál hamarabb befejeződik

Azonos hosszúságú szavakra megszorítva rendezésünket a szokásos lexikografikus rendezést kapjuk.

Egy másik szemlélet a rendezésre: $\{0,1\}^{\leq n}$ elemeit egy bináris fában soroljuk fel. A gyökérben az ϵ üres szó. Minden csúcs bal-gyereke a 0 bittel való kiterjesztése, jobb-gyereke az 1 bittel való kiterjesztése. A fa mélysége n. A d mélységben éppen a d hosszú szavaknak megfelelő csúcsok vannak. Fánk természetes módon bejárható: mindig balra lelépünk, ha arra még nem jártunk; jobbra lelépünk, ha arra még nem jártunk; különben felfelé lépünk.



1. ábra.

Ez a bejárás a gyökérből indul és ott akad el. közben minden csúcsot meglátogatunk. A leveleket egyszer, a belső csúcsokat háromszor látogatjuk meg. Ha minden szót az utolsó látogatásnál írjuk le (postorder sorrend), akkor a fenti sorrendet kapjuk.

Definíció. A SAT^* inputja egy φ CNF-et (legyen n a formula változóinak száma) és egy $t \in \{0,1\}^{\leq n}$-et tartalmaz. Akkor kell elfogadni, ha van $k \in \{0,1\}^{\leq n}$ kielégítő értékadás, amire $k \preceq t$.

A $SAT^* \in \mathcal{NP}$ könnyen látható. Ebből következik, hogy $SAT^* \preceq_{\mathcal{P}}^{\text{Karp}} SAT$.
Átfogalmazott feltétel: $SAT^* \preceq_{\mathcal{P}}^{\text{Karp}} S$ (ritka nyelv).
Átfogalmazott állítás: $SAT \in \mathcal{P}$. ($\mathcal{P} = \mathcal{NP}$ átfogalmazása, hiszen SAT $\mathcal{NP}$-teljes.)

A bizonyítás befejezése a következő előadásra marad.