ALGORITMUSOK ÉS BONYOLULTSÁGELMÉLET

Matematika MSc hallgatók számára

9. Előadás

Előadó: Hajnal Péter
Jegyzetelő: Hajnal Péter

2010. április 12.

Alternáló polinomiális idő

Emlékeztető. $\Sigma_i\mathcal{P}$, $\Pi_i\mathcal{P}$

Definíció. Legyen T egy nem-detereminisztikus Turing-gép, amelyre $S \subset S_0 \times \{\exists, \forall\}$, azaz minden állapotának van egy komponense, ami egszisztenciális ($\exists$) vagy univerzális ($\forall$). Egy adott ω-n gépünk lehetséges futásai szerteágaznak és egy fát alkotnak, amely levelei az ELFOGAD/ELVET állapotokban van (ezek esetén az univerzális/egzisztenciális jelleg nem számít). Ekkor T-t *alternáló Turing-gépnek* nevezzük.

Akár a nem-determinisztikus gépek esetén a kiszámolt érték (elfogadás/elvetés) leírása problémás.

Definíció. Minden konfigurációhoz egy értéket rendelünk. A leálló konfigurációk esetén az állapot mondja meg a kiszámolt értéket. Nem leálló konfigurációk esetén megnézzük milyen értéket rendeltünk a rákövetkezőkhöz. Ha a konfiguráció állapota univerzális, ez akkor és csak akkor lesz elfogadó csúcs, ha minden rákövetkező konfigurációja elfogadó. Ha a konfiguráció állapota egzisztenciális, ez akkor és csak akkor lesz elfogadó csúcs, ha van olyan rákövetkező konfigurációja, amely elfogadó. A gép egy ω inputot akkor és csak akkor fogad el, ha a kiinduló konfiguráció elfogadó.

Definíció II. változat. Egy L nyelv $\Sigma_i\mathcal{P}$ nyelvosztályhoz tartozik, ha van olyan őt elfogadó T alternáló Turing-gép, amelyre teljesül:

(P) minden futása polinomiális az input hosszában,

(Σ) a kiinduló konfiguráció egzisztenciális,

(A_i) minden futása során az egzisztenciális és univerzális állapotok $i-1$-szer változnak (i darab blokkban következnek)

Definíció. Egy L nyelv $\Pi_i\mathcal{P}$ nyelvosztályhoz tartozik, ha van olyan őt elfogadó T alternáló Turing-gép, amelyre teljesül (P), (A_i) és

(Π) a kiinduló konfiguráció univerzális.

A következő tételt bizonyítás nélkül említjük meg.

1. Tétel. *A múltkor és a most bevezetett $\Sigma_i\mathcal{P}$ és $\Pi_i\mathcal{P}$ nyelvosztályok ugyanazok.*

Megjegyezzük, hogy azon konfigurációk, amelyeknek egyetlen rákövetkezője van címkézetlenek maradhatnak. Ugyanis akár $\exists$ akár $\forall$ címkét kapnak a kiértékelésnél egyetlen rákövetkezőjük eredményét öröklik, bármelyik kvantorral lássuk is el. Így beszélhetünk a $\Sigma_0\mathcal{P}$ és $\Pi_0\mathcal{P}$ nyelvosztályokról, amik „mögött" lévő Turing-gépek determinisztikusak.

Észrevétel. (i) $\Sigma_0\mathcal{P} = \Pi_0\mathcal{P} = \mathcal{P}$

(ii) $\Sigma_1\mathcal{P} = \mathcal{NP}$, $\Pi_i\mathcal{P} =$ co-$\mathcal{NP}$

(iii) co-$\Sigma_i\mathcal{P} = \Pi_i\mathcal{P}$, co-$\Pi_i\mathcal{P} = \Sigma_i\mathcal{P}$,

(iv) $\Sigma_i\mathcal{P} \subset \Sigma_{i+1}\mathcal{P}, \Pi_{i+1}\mathcal{P}$, $\Pi_i\mathcal{P} \subset \Pi_{i+1}\mathcal{P}, \Sigma_{i+1}\mathcal{P}$

Az utolsó észrevétel egy kissé másként is megfogalmazható egy definíció segítségével.

Definíció.

$$\Delta_i\mathcal{P} = \Sigma_i\mathcal{P} \cap \Pi_i\mathcal{P}.$$

Ekkor $\Sigma_i\mathcal{P} \subset \Delta_{i+1}\mathcal{P} \subset \Sigma_{i+1}\mathcal{P}$, $\Pi_i\mathcal{P} \subset \Delta_{i+1}\mathcal{P} \subset \Pi_{i+1}\mathcal{P}$.

Természetesen a fenti osztályok indexe az alternálást korlátozza. Vehetjük a korlátozás nélküli alternáló Turing-gépet.

Definíció. $\mathcal{AP}$ azokat az L nyelveket tartalmazza, amikhez van olyan L-et elfogadó alternáló Turing-gép, ami teljesíti (P)-t.

Elhagytuk az alternálás korlátozására vonatkozó feltételt. Persze a polinomiális futás korlátozza az alternálások számát is. A kiinduló konfiguráció típusára tett feltétel elhagyása természetes. Ha az alternálások száma nem korlátozott, akkor a kvantorok blokkjai elé/mögé egy új kvantor-blokk tehető az elfogadott nyelv megváltoztatása nélkül. Így elérhető, hogy egzisztenciális állapotból induljon a gép (mint ahogy az is, hogy univerzálisból).

Emlékeztető.

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

Az eddigi osztályok viszonyai könnyen szemléltethetők

Észrevétel. $\mathcal{P} \subset \mathcal{NP} \subset \mathcal{PH} \subset \mathcal{AP}$.

Bizonyítás nélkül megemlítjük a következő fontos alaptételt.

2. Tétel. *(i)* $\mathcal{AP} = \mathcal{PSPACE}$,

(ii) QBF, a KVNATIFIKÁLT-BOOLE-FORMULA, nyelv teljes a $\mathcal{AP}$ *és* $\mathcal{PSPACE}$ *osztályokra is.*

Nem-uniform polinomiális idő

Emlékeztető. $L \in \mathcal{P}$ akkor és csak akkor ha van olyan $\{C_n\}_{n-1}^{\infty}$ polinomiális méretű hálózat sorozat, amely

(K) C_n input bitjei Σ^n elemeit kódolják (ω input kódja $\lceil\omega\rceil$ bitsorozat), továbbá $\omega \in L$ akkor és csak akkor, ha $C_n(\lceil\omega\rceil) = 1$, azaz C_n ω kódján az 1 logikai értéket számolja ki,

(U) C_n hálózat kódja 1^n ismeretében $\mathcal{L}$-ben kiszámolható.

Az (U)-val jelölt feltételt *uniformitásnak* nevezzük. Felmerül az uniformitás szerepének fontossága. A (K) feltétel kombinatorikus jellegű. $\mathcal{P}$ fenti definíciójában a számítás ketté van osztva, (K)-ban adott inputméret esetén 0-1 bitekkel történő egyszerű operálás történik. Az, hogy a számítás minden inputméretre egységesen kódolt, az (U)-ban jelenik meg. A következő definíció motivációja, hogy megpróbálunk a kombinatorikus összetevőre koncentrálni.

Definíció. $L \in \mathcal{P}^{\text{nem-uniform}}$ akkor és csak akkor ha van olyan $\{C_n\}_{n-1}^{\infty}$ polinomiális méretű hálózat sorozat, amelyre (K) teljtesül.

A nyelvosztály egy kicsit furcsa az eddigiekhez képest. Kilóg az eddig ismert legbővebb kiszámíthatósággal kapcsolatos $\mathcal{S}$ osztályból, a felsorolható nyelvek osztályából. Ehhez legyen $NS \subset \mathbb{N}$ a természetes számok egy nem felsorolható részhalmaza. $L = \{1^\ell : \ell \in NS\} \subset \Sigma^*$ nyilván nem felsorolható. Másrészt nyilván hálózatsorozattal kiszámolható: ha $\ell \in NS$, akkor a hálózat azaz egyszerű hálózat amelyik az összes változót ÉS-sel köti össze, különben pedig egy ellentmondást (például $x \wedge \neg x$-et) kiszámoló hálózat. A trükk a hálózat sorozat rapszódikus, algoritmikusan leírhatatlan váltakozása.

Az alábbiakban a $\mathcal{P}^{\text{nem-uniform}}$ osztály alternatív leírásait közöljük

3. Tétel. *Az alábbiak ekvivalensek:*

(i) $L \in \mathcal{P}^{\textit{nem-uniform}}$,

(ii) *L-hez van olyan tanúszalagos Turing-gép, amely csak az input hosszától függő tanúval elfogadtatja L-et. Azaz van olyan* $t_n \in \Sigma^{p(n)}$ *és polinomiális Turing-gép, hogy akkor és csak akkor teljesül, hogy* $\omega \in L$, *ha* $T(\omega, t_{|\omega|})$ *elfogadó.*

(iii) *Alkalmas S ritka nyelvre* $L \prec_{\mathcal{P}}^{Turing} S$.

A tétel utolsó pontjában szereplő jelöléseket meg kell magyaráznunk.

Definíció. Egy S nyelv akkor *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).

$L \prec_{\mathcal{P}}^{Turing} L'$ a redukció egy erősített változata (az erdetihez képest, amit Karp nevével is szoktak fémjelezni).

Definíció. $L \prec_{\mathcal{P}}^{Turing} L'$ ha van olyan polinomiális Turing-gép, ami ω ismeretében eldönti az L-hez tartozást, amennyiben kérdéseket tehet fel az L' nyelvbeliséggel kapcsolatban. (Ezen kérdések feltétele után, a következő konfigurációban már tudjuk a választ, azaz 1 a költsége az időbonyolultság szempontjából.)

A fenti redukció megvalósítását úgy lehet elképzelni, hogy egy új, kérdező szalagot vezetünk be. Erre írhatjuk fel azt a szót, aminek L'-beliségét szeretnénk tudni. Egy új '?' állapotunk lesz. Ebbe jutva a gép a következő állapotban 'BENNE-VAN' vagy NINCS-BENNE ' állapotba jut, a szerint, hogy a kérdező-szalag tartalma L'-beli, vagy nem. Az ilyen gépeket L'-orákulumos gépeknek nevezik. Az ezek által elfogadott nyelvosztály jele $\mathcal{P}^{L'}$. Természetesen más nyelvosztályokra is definiálhatók orákulumos gépek.

Bizonyítás. $(i) \Rightarrow (ii)$ Legyen t_n a megfelelő hálózat kódja. A (ii)-t igazoló gép a tanú alapján rekonstruálja a hálózatot, kiértékeli ω kódján.

$(ii) \Rightarrow (iii)$ Legyen $S = \{t(n,i) : t_n$ első i karaktere$\}$ egy ritka nyelv. Könnyű S-hez tartozásra kérdézéssel ω-hoz kiszámítani $t_{|\omega|}$-t, aminek ismerete megoldja az L-hez tartozás kérdését.

$(iii) \Rightarrow (i)$ Legyen S a megfelelő ritka nyelv az (iii) feltételből. A polinomiális redukálógép időigényéből egy tudott inputméret esetén fel tudjuk sorolni a leheséges kérdések polinomiális hosszú listáját (a kérdező-szalagra nem írhatunk polinomnál hosszabb szót (nincs rá időnk)). A redukciót végrehajtó gép számolását kódoljuk hálózattal, ahogy azt korábban tettük $\mathcal{P}$-beli gépekkel (ezen része a konstukciónknak uniform). Az egyetlen probléma az orákulumhoz intézett kérdések szimulálása. A fentiek alapján ez egy polinomhosszú kódlistához tartozással egyenértékű. Ezt könnyű hálózattal szimulálni (VAGY-olni kell az „egyes szavakkal azonosnak lenni" leírását, ami a karakterenkénti egyezés ÉS-elése). Az L nyelvről csak a ritkaságot tettünk/használtunk fel. (Konstrukciónk ezen részénél nem beszélhetünk uniformitásról.) ■

Karp, Lipton—Sipser-tétel

A SAT nyelv $\mathcal{NP}$-teljességének egy megfogalmazása a következő állítás.

Emlékeztető. Ha $SAT \in \mathcal{P}$, akkor $\mathcal{P} = \mathcal{NP}$

Ennek interpretálása a $\mathcal{P} \neq \mathcal{NP}$ sejtés fényében, hogy „nem várható, hogy SAT polinomiális időben megoldható". De vajon a $\mathcal{P}^{\text{nem-uniform}}$ osztályhoz tartozással mi a helyzet? A következő tétel azt állítja, ha hisszük, hogy a polinomiális hierarchia nem esik össze, akkor SAT ehhez a nyelvosztályhoz sem tartozhat hozzá.

4. Tétel (Karp—Lipton, Sipser tétele). *Ha* $SAT \in \mathcal{P}^{\textit{nem-uniform}}$, *akkor* $\Pi_2\mathcal{P} \subset \Sigma_2\mathcal{P}$.

Bizonyítás. A feltétel alapján tudjuk, hogy SAT kiszámolható egy $\{C_n\}$ hálózatsorozattal.

A bizonyítandóhoz legyen $L \in \Pi_2\mathcal{P}$, azaz alkalmas polinomiális Turing-gépre $\forall x \in \Sigma^{p(|\omega|)} \exists y \in \Sigma^{p(|\omega|)} T(\omega, x, y)$ akkor és csak akkor teljesül, ha $\omega \in L$. Azt kell bizonyítanunk, hogy $L \in \Sigma_2\mathcal{P}$. A továbbiakban $n := |\omega|$.

Legyen $\omega^+ = \omega, x$. Az L-beli inputok $\Pi_2\mathcal{P}$-beli leírásának belseje

$$\exists y \in \Sigma^{p(|\omega|)} T(\omega^+, y),$$

ami egy $\mathcal{NP}$-beli leírás, ha ω^+-t tekintjük inputnak. Így ez a probléma viszavezethető SAT-ra: kiszámolhatunk egy $\varphi(\omega^+) = \varphi(\omega, x)$ CNF-formulát, ami akkor és

csak akkor lesz kielégíthető ($C_\ell(\lceil\varphi(\omega,x)\rceil)=1$, ahol ℓ az ω,x hossza, azaz $n+p(n)$), ha ω-hoz x olyan, hogy a belső formula igaz legyen.

Egy ÖTLET: Tippeljük meg C_ℓ kódját. Nem lesz jó az L nyelv leírására a következő?

$$\exists\lceil C_\ell\rceil\forall x\in\Sigma^{p(|\omega|)}C_\ell(\lceil\varphi(\omega,x)\rceil)=1,$$

ahol C_ℓ az a Turing-gép, ami $\lceil C_\ell\rceil$-ből kicsomagolja C_ℓ-et és kiértékeli azt, majd teszteli, hogy az 1 bitet számoltuk-e ki.

Ha $\omega\in L$, akkor minden rendben van: a jó tipp elfogadáshoz vezet. Ha $\omega\notin L$, akkor a C_ℓ tipp rossz, de ez lehet egy vad rosszság. Ötletünk finomításra szorul.

5. Lemma. *Legyen C_ℓ egy hálózat. Ekkor polinomiális időben kiszámolunk egy $\widetilde{C}_\ell$ hálózatot, ami a következőket tudja:*

(i) ha C_ℓ a SAT-ot oldotta meg, akkor $\widetilde{C}_\ell$ is,

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

A lemma bizonyítása után már készen vagyunk az L nyelv leírására $\Sigma_2\mathcal{P}$ nyelvvel:

$$\exists\lceil C_\ell\rceil\forall x\in\Sigma^{p(|\omega|)}\widetilde{C}_\ell(\lceil\varphi(\omega,x)\rceil)=1,$$

ahol $\widetilde{C}_\ell$ az a Turing-gép, ami $\lceil C_\ell\rceil$-ből kicsomagolja C_ℓ-et, a lemma alapján egyik irányban hibajavítóvá teszi és kiértékeli azt a redukált belső részen, majd bejelenti, hogy az 1 bitet számoltuk-e ki. ■

Mahaney-tétel

A Karp—Lipton, Sipser tételében szerepő feltételt úgy is megfogalmazhatjuk, hogy $SAT\prec_{\mathcal{P}}^{\text{Turing}}S$ valamely S ritka nyelvre. Mi történik, ha nem Turing, hanem az eredeti/Karp redukcióval dolgozunk? (Azaz erősebb feltételünk van.) Előadáson nem szerepelt, de természetes folytatás lett volna a következő tétel.

6. Tétel (Mahaney-tétel). *Tegyük fel, hogy $SAT\prec S$ valamely ritka nyelvre. Ekkor $\mathcal{P}=\mathcal{NP}$.*

Bizonyítás. Először a SAT egy élesítését fogalmazzuk meg. Ehhez definiálunk egy rendezést $\{0,1\}^{\le n}$ halmazon, a legfeljebb n hosszú 0-1 szavakon. (A pontosan n hosszú 0-1 szavakon van egy természetes rendezés: értelmezzük kettes számrendszerbeli számként és $\mathbb{N}$-ből örökítsük a rendezést, vagy másképpen az első (balról-jobbra menve) eltérő bit dönt.)

Definíció. $x,y\in\{0,1\}^{\le n}$ esetén $x<y$, ha x és y első eltérő bitje 0 az x-ben és 1 az y-ban vagy pedig y valódi kezdőszelete x-nek.

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

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

Definíció. Legyen SAT^* az a nyelv, amely inputja egy φ CNF-et (n a változó szám) és egy $t\in\{0,1\}^{\le n}$ szót tartalmaz. Pontosan akkor kell elfogadni, ha van olyan $b\in\{0,1\}^n$ kielégítő értékadás, amire $b\le t$.

Nyilván ha a (φ, ϵ) inputról döntjük el a SAT^*-hoz tartozását, akkor φ kielégíthetőséget döntjük el. De az is nyilvánvaló, hogy SAT^* is $\mathcal{NP}$-beli. Speciálisan $SAT^* \prec SAT$ (SAT $\mathcal{NP}$-teljes). Így feltételünk miatt $SAT^* \prec S$ a tétel feltételét bizonyító S ritka nyelvre.

Polinom időben megoldjuk a SAT problémát: A leírandó eljárás során majd a φ input és mellé írt legfeljebb n hosszú bitsorozaton futtatjuk a redukciós algoritmust. Ez alapján becsülhetjük a redukció során S-ből kapott értékeket $t(n)$-nel valamely t polinommal (ahol n az inputformula változószáma).

Algoritmusunk futása alatt mindig lesz egy polinomiális hosszú listánk $\{0,1\}^{\leq n}$ elemeiből, amelyek hossza mindig ugyanaz lesz (a futás során egyenletesen fog nőni). Kezdőlistánk tartalmazza azon hosszú 0-1 sorozatokat, amik ugyanolyan hosszúak és számuk éppen $t(n)$ felett van. (Listánk hossza legfeljebb $2t(n)$.) Mindegyik elemre végezzük el a Karp-redukciót. Ha egy érték többször fordul elő, akkor csak a legelső 0-1 sorozatot hagyjuk listánkon azok közül, amelyek ugyanazon sorozatra vezetődtek vissza. Ha listánk túl hosszú, akkor csak az utolsó $t(n)$ darab elemét tartjuk meg. Ezzel elértük, hogy listánk hossza legfeljebb $t(n)$ legyen.

Vegyük listánk elemeinek összes 1 bittel való kiterjesztettjét. Legfeljebb $2t(n)$ hosszú listához jutunk és ismételjük meg korábbi lépésünket.

Ahogy az ismételgetést végezzük 0-1 sorozataink hossza 1-gyel nő. Ismételgessük ezt addig, amíg teljes kiértékeléseink lesznek (legfeljebb $t(n)$ sok). Ezek mindegyikére ellenőrizzük kielégíti-e φ-t. Ha bármelyik kiélegítő értékadás, akkor elfogadjuk az inputot. Különben elvetjük.

Eljárásunk nyilván polinomiális. Csak helyességét kell ellenőrizni. Egyetlen módon tévedhet: Egy kielégíthető formulát elvet. Belátjuk, hogy ha φ-nek van kielégítő értékadása, akkor az első (lásd rendezésünket) ilyen értékadás megfelelő hosszú kezdőszelete listánkon lesz.

Ez kezdetben nyilván teljesült. A duplikátumok kihúzása során nem veszthettük el. Később csak úgy lehetett baj, ha nála nagyobb elem $t(n)$ elem volt a listánkon, amik különböző elemekre redukálódtak. Mindegyik ilyen elem nagyobb mint az első kielégítő értékadásunk. Így a redukciójuk S-beli elemhez kellett hogy vezessen. A kapható S-beli elemek száma viszont maximum $t(n)$ lehet. Az ellentmondás az algoritmus helyességét és így a tétel igaz mivoltát bizonyítja. ■