ALGORITMUSOK ÉS BONYOLULTSÁGELMÉLET
Matematika MSc hallgatók számára

4. Előadás

Előadó: Hajnal Péter
Jegyzetelő: Hajnal Péter
2012. február 29.

1. Szép idő- és tárfüggvények

Definíció. Egy $t(n) : \mathbb{N} \to \mathbb{N}$ függvényt szép időfüggvénynek nevezzük, ha minden n hosszú inputon pontosan $t(n)$ ideig fut.

Definíció. Egy $s(n) : \mathbb{N} \to \mathbb{N}$ függvényt szép tárfüggvénynek nevezzük, ha minden n hosszú inputon leáll és pontosan $s(n)$ mezőt érint a munkaszalagon.

A fentiek technikai feltételek. Azonban az eddig használt függvények mind szépek. (Például idő esetén a polinom függvények, 2^n, vagy tár esetén $\lceil \log_2 n \rceil$ is.)

A fenti szépség hasznát néhány példával világítjuk meg.

Példa. Feltesszük, hogy $t(n)$ szép időfüggvény. Legyen T egy tetszőleges Turing-gép. Legyen ω egy tetszőleges n hosszú input.

ω-t átmásoljuk egy munkaszalagra, majd a szemek/kezek balra visszamennek ($2n$ idő). Innen párhuzamosan szimuláljuk a T gép futását és a munkaszalag segítségével egy $t(n)$ szépségét mutató W gépet. A T gép ELFOGAD/ELVET állapota leállítja gépünket. W leálló állapotát CSÖRÖG-nek nevezzük. W-re úgy gondolunk mint egy órára. A CSÖRÖG állapot az egész gépet leállítja. Így az új gépünk garantáltan $2n + t(n)$ időben leáll ($t(n) + 2n$ és $t(n)$ nagyságrendileg megegyezik). T számításait elvégzi, ha azok a $t(n)$ időbe beférnek.

Példa. Feltesszük, hogy $s(n)$ szép tárfüggvény. Legyen T egy tetszőleges Turing-gép. Legyen ω egy tetszőleges n hosszú input.

Először szimuláljuk az $s(n)$ szépségét mutató W gépet. Majd a használt mezőket felültírjuk egy üres jellel és mögéjük teszünk egy EDDIG karaktert. Ezekután elkezdjük a T gép szimulálását. Ha az EDDIG karaktert olvassuk, akkor leállunk SOK-MEMÓRIA állapottal. A T gép ELFOGAD/ELVET állapota leállítja gépünket. Így az új gépünk garantáltan $1 + s(n)$ tárat használ ($s(n)$ és $s(n) + 1$ nagyságrendileg megegyezik). Az új gép T számításait elvégzi, ha azok a $s(n)$ tárba beférnek.

Példa. Legyen T egy nem-determinisztikus gép, amely $t(n)$ időigényű és az L nyelvet számolja ki. Ennek lehet sok olyan futása lehet, aminek idejéről nem tudunk semmit. A fenti példa alapján HA $t(n)$ szép, akkor feltehető, hogy gépünk minden futása $t(n) \approx t(n) + 2n$ lépés alatt megáll. A leállított futások nem változtatják meg az elfogadott nyelvet. Az időbonyolultsági feltétel miatt $\omega \in L$ esetén lesz olyan ElFOGAD állapothoz vezető futás, ami CSÖRÖG előtt végetér. Azaz a szimuláló gép is „észleli" ezt.

Példa. Legyen $s(n)$ egy szép tárfüggvény. Legyen T egy $s(n)$ tárigényű gép. Ekkor elérhetjük, hogy gépünknek pontosan kétféle leálló konfigurációja legyen:

Futtassuk T-t. A számítás végén azonban „tartsuk meg magunknak" az eredményt és az ELFOGAD/ELVET állapotok bejelentése előtt töröljük le a munkaszalagot $s(n)$ hosszban (a szépség miatt ez könnyen megtehető). Minden szem/kéz mozogjon balra. Ezek után érjük el a kiszámított eredménynek megfelelő leálló állapotot. Kétféle leálló konfiguráció („fénykép" a gépről) és természetesen gépünk ugyanazt számolja ki mint az erdeti.

Megjegyezzük, hogy a szép időfüggvényt (legyen W az ezt bizonyító Turing-gép) tárkijelölésre is használhatjuk. ω átmásolása után W szimulációja mellett a többi munkaszalagon a balra állított szem/kéz folyamatosan jobbra haladjon a CSÖRÖG állapotig. Ekkor az átmásolt inputon túli munkaszalagokon pontosan $t(n)$ mező lett kijelölve.

Emlékeztető. Az eddigi nyelvosztályokról a következő tartalmazások nyilvánvalók:

$$\begin{array}{ccccc}
 & & \mathcal{NP} & \subseteq & \mathcal{NEXP} \\
 & & \cup\!| & & \cup\!| \\
 & & \mathcal{P} & \subseteq & \mathcal{EXP} \\
 & & \cap\!| & & \cap\!| \\
\mathcal{L} & \subseteq & \mathcal{PSPACE} & \subseteq & \mathcal{EXPSPACE} \\
\cap\!| & & \cap\!| & & \cap\!| \\
\mathcal{NL} & \subseteq & \mathcal{NPSPACE} & \subseteq & \mathcal{NEXPSPACE} \\
 & & \cup\!| & & \cup\!| \\
 & & \mathcal{NP} & \subseteq & \mathcal{NEXP}
\end{array}$$

Az $\mathcal{L}, \mathcal{P}, \mathcal{EXP}$ nyelvosztályok determinisztikus gépekhez kapcsolódnak, ebből adódóan zártak a komplementerképzésre. Az $\mathcal{NL}, \mathcal{NP}, \mathcal{NPSPACE}, \mathcal{NEXP}$ osztályok nem-determinisztikus nyelvcsaládok (ezt hangsúlyozza a kezdő $\mathcal{N}$ betű). Ezek esetén a *co* $\mathcal{NL}$, *co* $\mathcal{NP}$, *co* $\mathcal{NPSPACE}$, *co* $\mathcal{NEXP}$ osztályok érdekesek.

Definíció.

$$\mathcal{TIME}(t(n)) = \{L : \text{létezik } L\text{-et elfogadó } T \text{ Turing-gép, amelynek futási ideje minden } \omega\text{-n legfeljebb } t(|\omega|)\}.$$

$$\mathcal{SPACE}(s(n))) = \{L : \text{létezik } L\text{-et elfogadó } T \text{ Turing-gép, amelynek tár igénye minden } \omega\text{-n legfeljebb } s(|\omega|)\}.$$

Célunk a következő tartalmazási lánc belátása:

$$\mathcal{L} \subset \mathcal{NL} \overset{(1)}{\subset} \mathcal{P} \subset \mathcal{NP} \overset{(2)}{\subset} \mathcal{PSPACE} \subset \mathcal{NPSPACE} \overset{(3)}{\subset} \mathcal{EXP} \subset \mathcal{NEXP}.$$

Megszámoztuk a még bizonyítatlan tartalmazásokat. Az alábbiakban belátjuk ezeket. Célunk azonban nem a minél rövidebb indoklás, hanem az eredmények összefoglalása és a későbbi módszerek bevezetése.

2. További tartalmazások bonyolultsági osztályok között

Észrevétel 0.a. $\mathcal{TIME}(t(n)) \subset \mathcal{SPACE}(t(n))$.

Valóban, az időkorlát korlátozza azt, hogy a munkaszalag szem/keze milyen messze tud elmozogni.

Észrevétel 0.b. (i) $\mathcal{TIME}(t(n)) \subset \mathcal{NTIME}(t(n))$.

(ii) $\mathcal{SPACE}(s(n)) \subset \mathcal{NSPACE}(s(n))$.

Valóban, a determinizmus felfogható, mint a nem-determinisztikusság egy speciális esete.

Észrevétel 1. $\mathcal{NTIME}(t(n)) \subset \mathcal{SPACE}(t(n))$, ahol $t(n)$ szép időfüggvény.

Bizonyítás. Legyen $L \in \mathcal{NTIME}(t(n))$. Ekkor megadható ezt bizonyító T tanúszalagos Turing-gép (azaz T az L nyelvet fogadja el és minden ω inputra $t(|\omega|)$ az időbonyolultsága. Ezt a továbbiakban $L \in_T \mathcal{NTIME}(t(n))$ jelöléssel írjuk le.

Az állítás bizonyításához megadunk (T-re alapulva) egy $\widetilde{T}$ egy determinisztikus Turing-gépet, amely ugyanazt a nyelvet fogadja el és tár korlátja $t(n)$ lesz. Ehhez megtartjuk T leírásához szükséges munkaszalagokat és hozzáadunk egyet, amely a tanú szalag szerepét tölti be és még egyet, ami egy óra szerepét tölti be ($t(n)$ szép időfüggvény). Persze az új gép a nem-determinisztikus gépek „zsenialitását"/tippelő tulajdonságát nem birtokolja. $\widetilde{T}$ működésének leírásához megadjuk, hogyan néz ki egy futása. Ebből az átmenetifüggvény (formális leírása) kiolvasható. Feltesszük, hogy az inputunk hossza n.

Inicializáló fázis: A tanúszalag szerepét betöltő munkaszalagon kijelölünk $t(n)$ számú mezőt, melyet egy Γ-beli speciális határolójellel lezárunk. Ez egy csak erre a célra használt karakter. Ezen karakter olvasásakor tudjuk, hogy a tár korlát betartása mellett nem léphetünk jobbra. ($t(n)$ szép időfüggvény, azaz vehetünk egy órát, ami $t(n)$ lépés után „csörög" (és persze újra felhúzható). Ennek segítségével a tárterület kijelölése könnyen megoldható: a munka szalag felett a csörgésig jobbra mozgunk.)

A tanúszalag szerepét betöltő munkaszalagra felírjuk az első lehetséges $t(n)$ karaktert, ami egy tanú-kezdet lehet (több karakterre nincs szükségünk mert $t(n)$ időkorlátos gép nem tud többet elolvasni).

Szimuláló fázis: A T Turing-gép munkaszalagjainak megfelelő szalagokon szimuláljuk T futását az első tanún $t(n)$ ideig. A szimuláció vagy ELFOGAD, vagy NEM-STIMMEL állapottal ér véget, vagy letelik az idő/kifutunk a $t(n)$ időből. Ez utóbbit is a tesztelt tanú elvetéseként (NEM-STIMMEL állapot) fogjuk fel.

Ha a szimuláció ELFOGAD álapotba jutott, akkor mi is ELFOGADjuk az inputot, $\widetilde{T}$ is leáll. Ha ELVET állapotba jutott, akkor a tanú szalag szerepét betöltő szalagon a következő lehetséges $t(n)$ hosszú tanúkezdettel írjuk felül eddigi tartalmát. A többi szalag tartalmát letöröljük. Megismételjük a Szimuláló fázist.

Ha a következő tanú-kezdet generálása nem lehetséges, mert az összes tanúkezdetet teszteltük (a tanúk kimerültek), akkor ELVET állapottal leállunk.

1. Állítás. *(i) $\widetilde{T}$ L-et számolja ki.*

(ii) $\widetilde{T}$ tárigénye legfeljebb $t(n)$.

Mindkét rész egyszerűen adódik az előzőekből. Ezzel az észrevételt igazoltuk. ■

Ezzel speciálisan adódott a (2)-vel jelölt tartalmazás.

Észrevétel 2. $\mathcal{SPACE}(s(n)) \subseteq \cup_{c\in\mathbb{N}}\mathcal{TIME}(c^{s(n)+\log(n+1)})$, ahol $s(n)$) szép tárfüggvény.

Bizonyítás. Legyen $L \in \mathcal{SPACE}(s(n))$. Ekkor megadható olyan T Turing-gép, amely eldönti L-et (speciálisan minden $\omega \in L$-en megáll), és a tárigénye legfeljebb $s(n)$.

Legyen $\kappa_0(\omega) \to \kappa_1(\omega) \to \kappa_2(\omega) \to \ldots \to \kappa_\ell(\omega)$ a futás ω-n. Azaz ez egy $\ell \geq 1$ hosszú, véges konfigurációsorozat, ahol az első konfiguráció ($\kappa_0(\omega)$) a kiinduló konfiguráció (ebben az állapot START) és az utolsó állapot ($\kappa_\ell(\omega)$) az első olyan konfigutáció a futás során, amleyben az állapot ELFOGAD/ELVET.

Könnyű látni, hogy a futás során nem ismétlődhet konfiguráció, azaz $i \neq j$ esetén $\kappa_i \neq \kappa_j$ teljesül. Valóban minden konfiguráció egyértelműen meghatározza a rákövetkezőt, így ismétlődés egy végtelen, periodikus konfigurációsorozathoz vezetne.

Hányféle konfiguráció léphet fel a fenti sorozatban rögzített ω esetén? Legyen $|\omega| = n$. Egy felső becslés a kérdésre adandó válaszra (α_T, β_T konstansok T-től függnek):

$$(n+2)\cdot|S|\cdot(s(n))+1)\cdot|\Gamma|^{s(n))} \leq \alpha_T(n+1)\beta_T^{s(n)} = \beta_T^{s(n)+\log(n+1)},$$

hiszen az input szem helyzete $n+2$-féle, a munka szem helyzete $s(n)+1$-féle, az input szalag tartalma $|\Gamma|^{s(n)}$-féle, az állapot $|S|$-féle lehet.

Összefoglalva: Ha a futási idő ${\beta_T}^{s(n))+\log n}$-nél hosszabb lenne, akkor a futás során a konfigurációk ismétlődnének, így a futás végtelen lenne.

Tudjuk, hogy nincs így. Tehát kaptuk, hogy T időigénye automatikusan megfelel az észrevételben szereplőkkel. ■

Észrevétel 3. $\mathcal{NSPACE}(s(n)) \subseteq \bigcup\limits_{c\in\mathbb{N}}\mathcal{TIME}(c^{s(n)+\log(n+1)})$, ahol $s(n)$ szép tárfüggvény.

Bizonyítás. Legyen $L \in_T \mathcal{NSPACE}(s(n))$. Azaz T (I. értelemben vett) nemdeterminisztikus Turing-gép, vagyis az átmeneti függvény nem-determinisztikus, a futás „szétágazó" lehet. T kiszámolja L-et és tárigénye $s(n)$. T-ről feltehető, hogy leálláskor az input- illetve a munkafej a szalag elejére áll, továbbá a munkaszalag első $s(n)$ karaktere üres (a gép leradirozza a munkaterületét). Így a leálláskor két konfiguráció fordulhat elő. Speciálisan elfogadó futás során tudjuk mi az utolsó konfiguráció.

Definíció. Redukált konfiguráció $s(n)$ tárigényű I. nem-determinisztikus Turing-gép esetén adott ω inputra nézve a következő komponenseket tartalmazza:

1) input- és munkafej pozíciója,

2) munkaszalag első $s(n)$ karaktere,

3) a gép állapota.

Tulajdonképpen csak az inputszalag tartalmát és a munkaszalag garantáltan olvasatlan/érintetlen részét takartuk le a (teljes) konfigurációból. A redukált konfigurációk halmaza legyen V. Ekkor

$$|V| \leq \alpha_T \cdot (n+1) \cdot \beta_T^{s(n)}.$$

A κ konfigurációból természetesen kiolvasható a megfelelő redukált $\rho = \mathrm{red}(\kappa)$ konfiguráció. Ha ismert az ω input, akkor megfordítva is igaz: ω és a ρ redukált konfiguráció meghatározza a $\kappa = \mathrm{konf}(\omega, \rho)$ teljes konfigurációt.

Definíció. Legyen T egy I. nemdeterminisztikus Turing-gép és ω egy inputja. Ekkor $\overrightarrow{G}_{\omega,T}$ a (T, ω)-hoz tartozó redukált konfigurációk gráfja. Ez egy irányított gráf, ahol a csúcsok halmaza a fenti V halmaz, továbbá $\overrightarrow{uv}$ akkor és csak akkor él, ha az $\mathrm{konf}(\omega, u)$ konfiguráció után az átmeneti függvény megengedi a $\mathrm{konf}(\omega, v)$ konfigurációt.

Legyen V speciális eleme $v_0 = \mathrm{red}(\kappa_0(\omega))$ a kezdő konfiguráció redukáltja. Legyen v_0 az elfogadó leállásnak megfelelő konfiguráció redukáltja.

Megjegyezzük, hogy determinisztikus gép esetén is bevezethetők a fenti fogalmak. Ekkor a definiált irányított gráf minden pontjának kifoka 1 lenne.

A következő állítás a fenti definíciók megértése után nyilvánvaló.

2. Állítás. *Az $\omega \in L$ pontosan akkor teljesül, ha $\overrightarrow{G}_{\omega,T}$-ben létezik $v_0 v_1$ irányított út.*

Valóban, $\omega \in L$ ekvivalens elfogadó futás létezésével ω-n. Az elfogadó futások párbaállíthatók a $v_0 v_1$ irányított utakkal.

Legyen $T_1(T)$ egy determinisztikus gép, ami az ω inputon kiszámítja a $\overrightarrow{G}_{\omega,T}, v_0, v_1$ hármas kódját.

3. Lemma. *$T_1(T)$ megvalósítható úgy, determinisztikus tárigénye*

$$\alpha_T(s(n) + \log(n+1))$$

legyen.

Bizonyítás. A megadott tár arra elegendő, hogy a munkaszalagon konstans számú redukált konfiguráció kódját írjuk le. Számunkra két redukált konfiguráció számára kell hely. A futás elején kijelölünk két blokkot a munkaszalag elején, amik egy-egy redukált konfiguráció tárolására szolgálnak ($s(n)$ szép tárfüggvény).

Az első blokkban felsoroljuk az összes lehetséges kódszót. Ezek a jelöltek arra, hogy gráfunk egy csúcsát kódolják. Mindegyik jelöltnél eldöntjük, hogy redukált konfiguráció kódja-e. (Egy természetes kódolási szabály lerögzítése után az a feladat könnyen megoldható.) Ha nem, akkor a következő jelöltre térünk át. Ha igen, akkor átmásoljuk az outputszalagra, majd egy ':' rakunk. Ezután fogjuk írnunk a ki-szomszédok sorozatát. A munkaszalag második blokkjában szintén elkezdjük a redukált konfigurációk felsorolását. Ha a munkaszalagon x és y redukált konfigurációk kódja szerepel, akkor eldöntjük, hogy x-ből vezet-e él y-hoz. Az inputszalagon ott van ω a T Turing-gép az egész kondtrukció alapja, iszmert számunkra. Ezen részfeladat implementációja ismét egyszerű. Ha azt kapjuk, hogy vezet él, akkor y-t az outputszalagra másoljuk Ha azt kapjuk, hogy nem vezet él, akkoregyből a következő y keresésére térünk. Ha az y kimerül, akkor az következő x keresésére térünk át. Ha az x-ek is kimerülnek, akkor kiszámoltuk $\overrightarrow{G}_{\omega,T}$ kódját.

v_0 és v_1 kódjának felírása az outputszalagra szintén könnyen megoldható.

A részfeladatok megoldását nem részleteztük. Megvalósításuk során az adott tárkorlátot nem kell túllépnünk. ■

Térjünk vissza a bonyolultsági osztályok tartalmazásának bizonyításához. $L \in_T \mathcal{NSPACE}(s(n))$. Futassuk $T_1(T)$-t ω-n és írjuk le $\overrightarrow{G}_{\omega,T}, v_0, v_1$ kódját. Ennek a determinisztikus eljárásnak a tárigényét előbb becsültük és így futási ideje is legfeljebb $2^{\beta_T(s(n)+\log(n+1))}$. Így a kiszámolt kódszó hossza is legfeljebb $2^{\beta_T(s(n)+\log(n+1))}$.

Döntsük el, hogy $\overrightarrow{G}_{\omega,T}$-ben van-e v_0v_1 irányított út. Erre számtalan megoldás létezik. A szélességi, illetve mélységi keresés módszere biztos szerepelt BSc Algoritmuselmélet kurzusban. Az algoritmus Turing-gépen megvalósítható (T_2). Futási ideje (különösebb ötlet nélkül) polinomiális az input hosszában.

$T_1(T)$ és T_2 együtt éppen az L nyelvet dönti el és időigénye $2^{\gamma_T(s(n)+\log(n+1))}$. Ez adja a bizonyítandót. ■

Észrevételünkből az (1) és (3) tartalmazás is adódik.

3. Előretekintés

A bizonyított tartalmazásokat kiegészíthetjük a nem-determinisztikus osztályok komplementer nyelveinek osztályaival:

$$\begin{array}{ccccccccccccc}
 & & \mathcal{NL} & & & & \mathcal{NP} & & & & \mathcal{NSPACE} & & \\
 & \rotatebox{45}{$\subseteq$} & & \rotatebox{-45}{$\subseteq$} & & \rotatebox{45}{$\subseteq$} & & \rotatebox{-45}{$\subseteq$} & & \rotatebox{45}{$\subseteq$} & & \rotatebox{-45}{$\subseteq$} & \\
\mathcal{L} & & & & \mathcal{P} & & & & \mathcal{PSPACE} & & & & \mathcal{EXP} \\
 & \rotatebox{-45}{$\subseteq$} & & \rotatebox{45}{$\subseteq$} & & \rotatebox{-45}{$\subseteq$} & & \rotatebox{45}{$\subseteq$} & & \rotatebox{-45}{$\subseteq$} & & \rotatebox{45}{$\subseteq$} & \\
 & & co\,\mathcal{NL} & & & & co\,\mathcal{NP} & & & & co\,\mathcal{NSPACE} & &
\end{array}$$

További összefüggések is vannak:

$$\mathcal{NL} = \mathrm{co}\mathcal{NL},$$

$$\mathcal{PSPACE} = \mathcal{NPSPACE} = \mathrm{co}\mathcal{NPSPACE}.$$

Ezeket az összefüggéseket később igazoljuk.

Az is igaz, hogy az idő, illetve tárkorlát lényeges emelésével bővebb osztályhoz jutunk:

$$\mathcal{L} \subsetneq \mathcal{PSPACE},$$

$$\mathcal{P} \subsetneq \mathcal{EXP}.$$

Ennél több azonban nem ismert. Azt a kérdést, hogy „A $\mathcal{P} \subset \mathcal{NP}$ tartalmazás valódi, vagy egyenlőség áll fenn?" sokan a XXI. századi matematika központi problémájának tartják.