ALGORITMUSOK ÉS BONYOLULTSÁGELMÉLET

Matematika MSc hallgatók számára

3. Előadás

Előadó: Hajnal Péter

Jegyzetelő: Pék Máté

2010. február 15.

Az előadáson két különböző nem-determinisztikus Turing-gépet definiáltunk.

Nem-determinisztikus Turing-gépek

Definíció (I. változat). A T nem-determinisztikus Turing-géphez kapcsolódó nyelveket hasonlóan definiáljuk (szalagok, „kezek”, „szemek”, ... ugyanaz mint a determinisztikusé). Az átmeneti függvény értelmezési tartománya sem változik: $(\Sigma \cup \{\triangleright, \triangleleft\}) \times (\Gamma \cup \{\triangleright, \_\}) \times S$. A különbség (a determinisztikushoz képest), hogy az értékkészlet a determinisztikus értékkészlet egy nem üres részhalmaza:

$$P(\{\leftarrow, ., \rightarrow\} \times \Gamma \times \{\leftarrow, ., \rightarrow\} \times (\{.\} \cup \Sigma) \times S) \backslash \emptyset.$$

Következmény-Definíció. κ konfigurációhoz hozzárendelhető a $K^+(\kappa)$ konfigurációhalmaz, amit úgy kapunk meg, hogy κ-ból látott részhez az átmeneti függvény által hozzárendelt értéknek minden eleme alapján felírjuk, a determinisztikus esetben leírt módon, milyen rákövetkező konfigurációba megy át gépünk. $K^+(\kappa)$ elemeit a lehetséges rákövetkező konfigurációknak nevezzük.

Megjegyzés. A determinisztikus esetben κ-hoz egyetlen κ^+ tartozott. Most a κ látható részéhez rendelt halmaz elemszáma sok lehetséges rákövetkező konfiguráció lesz. Az értékkészletből az üreshalmazt kizártuk. Így $K^+(\kappa)$ nem üres.

Következmény-Definíció. Legyen $\omega \in \Sigma^*$ input.

$$\Phi(\omega) = \{\{\kappa_i\}_{i=0}^{\infty} : \kappa_0 = \kappa_0(\omega)(\omega \text{ kezdőkonfigurációja}, \kappa_{i+1} \in K^+(\kappa_i)\},$$

a *lehetséges futások* halmaza az ω inputon.

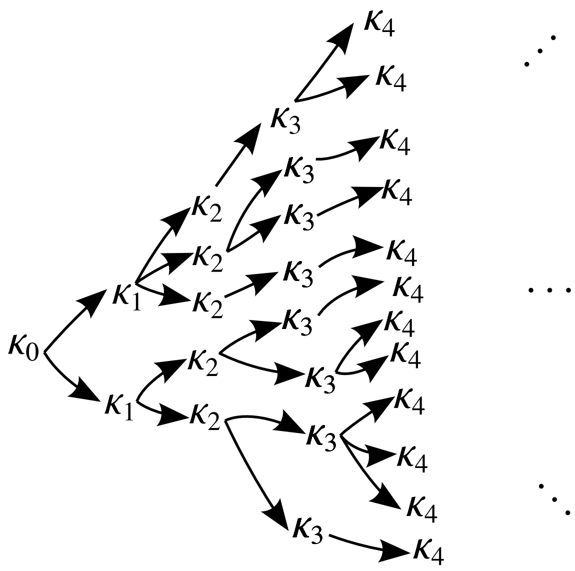
Emlékeztető. A determinisztikus esetben ω meghatározott egy konfiguráció sorozatot. $\kappa_0(\omega) \rightarrow \kappa_1 \rightarrow \kappa_2 \rightarrow \kappa_3 \rightarrow \dots$

A nem determinisztikus esetben a kiinduló konfigurációból egy fa írja le a gép lehetséges futásait (1. ábra).

$\kappa_0(\omega)$ a fa gyökere, a lehetséges futások a gyökérből induló, levélben véget érő utak. Levelek, azok a konfigurációk, ahol az állapot STOP/ELVET/ELFOGAD. A nem-determinisztikus gép futását is (ahogy a determinisztikus esetben) megállítjuk egy ilyen állapot elérésénél.

Definíció. Legyen T nem-determinisztikus Turing-gép. T elfogadja az L nyelvet, ha:

1) mindig leáll,



1. ábra. A nem-determinisztikus gép elágázó futás-fája

2) $\omega \in L$ esetén van olyan $\phi \in \Phi(\omega)$, ami elfogadó futás,

2') $\omega \notin L$ minden $\phi \in \Phi(\omega)$ elvető futás.

Tehát egyetlen elfogadó számítási sorozat (az összes lehetséges futások halmazában), bizonyítja az input elfogadhatóságát/nyelvhez tartozását.

Megjegyzés. A fenti definíció ismeretében talán jobb az ELVET ≡ „NEM-STIMMEL", ELFOGAD ≡ „BIZTOS-JÓ" alternatív szóhasználatot használnunk.

A nemdeterminisztikus gépek futási idejének vizsgálatához az alábbi fogalmakat vezetjük be.

Definíció. Legyen T nem-determinisztikus Turing-gép és $\omega \in \Sigma^*$. T időigényét külön definiáljuk az elfogadott L nyelv elemeire és nem elemeire:

$$NTIME(\omega, T) = \begin{cases} \min\{l : \{\kappa_i\}_{i=0}^{l} \text{ BIZTOS-JÓ futás}\}, & \omega \in L, \\ \min\{l : \{\kappa_i\}_{i=0}^{l} \text{ NEM-STIMMEL futás}\}, & \omega \notin L. \end{cases}$$

A legrosszabb eset analízis logikája elvezet a következő definícióhoz:

Definíció. $NTIME(n, T) = max\{NTIME(\omega; T) : \omega \in \Sigma^n\}$

$NSPACE(\omega, T)$ és $NSPACE(n, T)$ teljesen hasonlóan definiálható.
A determinisztikus esetben látott logika alapján bevezethetünk nyelvosztályokat:

Definíció. $L \in \mathcal{NTIME}(f(n))$ akkor és csak akkor, ha alkalmas T nem-determinisztikus Turing-gépre teljesül

(i) T L-et dönti el,

(ii) $NTIME(n, T) \leq f(n)$.

A determinisztikus átmeneti függvényt is tekinthetjük nem-determinisztikusnak: mindig egy eleme van a rákövetkező konfigurációk halmazának. A determinisztikus futás konfigurációsorozatát egy fának tekinthetjük, ahol minden csúcsnak egy fia van, valamint egy levele van a fának (ahol a kiszámított érték a levélkonfiguráció állapotában van elkódolva). Nyilván $\mathcal{TIME}(f(n)) \subseteq \mathcal{NTIME}(f(n))$.

A megfelelő nemdeterminisztikus tár-bonyolultsági osztályok:

Definíció. $L \in \mathcal{NSPACE}(f(n))$ akkor és csak akkor, ha alkalmas T nem-determinisztikus Turing-gépre teljesül

(i) T az L nyelvet dönti el,

(ii) $NSPACE(n, T) \leq f(n)$.

A robosztusabb nyelvosztályok:

Definíció.

$$\mathcal{NL} = \bigcup_{\alpha \in \mathbb{N}} \mathcal{NSPACE}(\alpha \log n),$$

$$\mathcal{NP} = \bigcup_{\alpha \in \mathbb{N}} \mathcal{NTIME}(\alpha n^{\alpha}),$$

$$\mathcal{NEXP} = \bigcup_{k \in \mathbb{N}} \mathcal{NTIME}(2^{n^k}).$$

Definíció. Az $f(n)$-t **szép-időfüggvény**nek nevezzük, ha $\exists T$ Turing-gép, hogy $\Sigma = \{1\}$ és $\omega = 1^n$-n pontosan $f(n)$ lépést tesz, és leáll.

A szépség előnyét a következő lemma mutatja.

1. Lemma. *Legyen T egy nem-determinisztikus Turing-gép. $NTIME(n; T) \leq t(n)$, ahol $t(n)$ szép-időfüggvény. Ekkor $\exists \widetilde{T}$ nem-determinisztikus Turing-gép:*

(i) ugyanazt a nyelvet fogadja el,

(ii) minden futása legfeljebb $t(|\omega|)$ idejű.

A tár bonyolultságnál a megfelelő „szépség” tár változata szükséges.

Definíció. Az $s(n)$-t **szép-tárfüggvény**nek nevezzük, ha $\exists T$ Turing-gép úgy, hogy $\omega \in \Sigma^n$-en pontosan $s(n)$ mező fölött halad át a futás során.

Az előzőhöz hasonló lemma kimondását és bizonyítását az olvasóra bízzuk.

Jelölés. Legyen $\mathcal{C}$ egy nyelvosztály (például: $\mathcal{P}, \mathcal{NL}, \mathcal{PSPACE}, \mathcal{EXP}, \ldots$), ekkor a nyelvosztály komplementere: *co*-$\mathcal{C} = \{\overline{L} : L \in \mathcal{C}\}$, ahol $\overline{L} = \Sigma^* \backslash L$.

A determinisztikus esetben az ELFOGAD/ELVET állapotok felcserélésével a komplementer nyelvet elfogadó gépet kapunk, amely bonyolultsága az eredeti gépével azonos. Azaz:

Észrevétel. *co*-$\mathcal{P} = \mathcal{P}$, *co*-$\mathcal{EXP} = \mathcal{EXP}$, *co*-$\mathcal{L} = \mathcal{L}$.

A nem-determinisztikus esetben nem ilyen egyszerű a helyzet. A $\mathcal{P}, \mathcal{NP}$ és $co\mathcal{NP}$ osztályok lehetséges kapcsolatairól a nyilvánvaló $\mathcal{P} \subset \mathcal{NP}, co\mathcal{NP}$ tartalmazáson kívül csak azt tudjuk, ha $\mathcal{NP}$ és $co\mathcal{NP}$ közt tartalmazás van, akkor a két osztály egyenlő. Minden más lehetőség a három osztály viszonyára olyan, hogy nem tudjuk kizárni:

$$\mathcal{P} \subsetneq \mathcal{NP} \cap co\mathcal{NP} \subsetneq \mathcal{NP}, co\mathcal{NP}, \qquad \mathcal{P} = \mathcal{NP} \cap co\mathcal{NP} \subsetneq \mathcal{NP}, co\mathcal{NP},$$

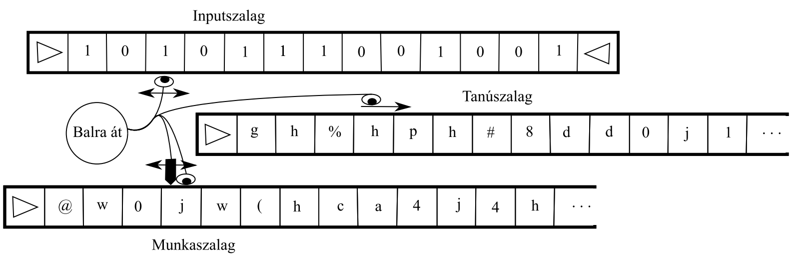
$$\mathcal{P} \subsetneq \mathcal{NP} = co\mathcal{NP}, \qquad \mathcal{P} = \mathcal{NP} = co\mathcal{NP}.$$

Hogy mi az igazság, senki sem tudja. Egy sejtést kiemelünk: $\mathcal{P} \subsetneq \mathcal{NP}$. Ez egyike a milleniumi problémáknak.

A nem-determinizmus, mint „bizonyítás ellenőrzés”

Az alábbi, második típusú, nem-determinisztikus Turing-gépet is csak döntési problémákra írjuk le:

Definíció (II. változat). Bővítjük a standard Turing-gépet egy csak olvasható, úgy nevezett tanúszalaggal/bizonyító szalaggal, ahol a szemmozgás csak jobbra haladhat, továbbá átírás nem lehetséges (nincs kéz).



2. ábra. Tanúszalagos gép egy példa konfigurációja

ω input és τ tanúszalag tartalomhoz tartozó kiindulási állapotból a futás: $\kappa_0(\omega, \tau) \to \kappa_1(\omega, \tau) \to \kappa_2(\omega, \tau) \to \ldots$

Ha csak ω-t rögzítjük, akkor a különböző τ tanúszalag-tartalmak különböző utakat definiálnak. Ha τ és τ' első karakterei megegyeznek, akkor a két futás eleje (amíg egyik gép tanú-szeme sem éri el az első különböző tanú-karaktert) megegyezik. Így ezek az utak összessége is egy fában foglalható össze.

Definíció. A T nem-determinisztikus Turing-gép elfogadja az L nyelvet, ha:

(i) mindig leáll,

(ii) $\omega \in L$ esetén mindig van olyan τ tanú, hogy T futása (ω, τ)-n ELFOGADÓ legyen, míg $\omega \notin L$ esetén tetszőleges τ tanúra T futása ELVETŐ.

Definíció. Legyen T nem-determinisztikus Turing-gép.

$$NTIME(\omega, T) = \begin{cases} \min\{TIME((\omega, \tau); T) : \tau \text{ elfogadó futásra vezet}\}, & \text{ha } \omega \in L \\ \min\{TIME((\omega, \tau); T)\}, & \text{ha } \omega \notin L \end{cases}$$

A tárbonyolultság fogalmai teljesen hasonlóan felépíthetőek. Ezt az olvasóra hagyjuk.

Bizonyítás nélkül megemlítjük, hogy a két fogalom ugyanahhoz a nyelvosztályokhoz vezet.

Eddig definiált bonyolultsági osztályok viszonya

Az időkorlát erősebb mint ugyanaz a tárkorlát. A nem-determinizmus „lazít” a determinizmuson. Ezek a nyilvánvaló észrevételek triviális tartalmazásokat adnak a bevezetett bonyolultsági osztályok között. Az alábbi diagram néhányat tartalmaz ezek közül.

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

Az alábbi tétel pontosabb viszonyokat leírását teszi lehetővé.

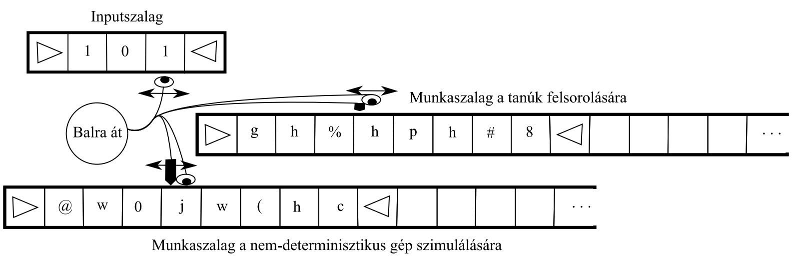
2. Tétel. *Legyen $t(n)$ szép-időfüggvény, $s(n)$ szép-tárfüggvény, ekkor*

(i) $\mathcal{NTIME}(t(n)) \subseteq \mathcal{SPACE}(t(n))$

(ii) $\mathcal{SPACE}(s(n)) \subseteq \bigcup_{\alpha\in\mathbb{N}} \mathcal{TIME}(\alpha^{s(n)+logn})$

(iii) $\mathcal{NSPACE}(s(n)) \subseteq \bigcup_{\alpha\in\mathbb{N}} \mathcal{TIME}(\alpha^{s(n)+logn})$

Bizonyítás. **(i) bizonyítása:** Legyen $L \in \mathcal{NTIME}(t(n))$, ekkor van olyan T II-nem-determinisztikus Turing-gép, ami L-et eldönti $t(n)$ időben. Egy determinisztikus Turing-gépet írunk le kettő munkaszalaggal. Mindkét munkaszalagon $t(n)$ szépsége alapján a munkaszalagokon kijelölhetünk $t(n)$ mezőt, az utolsóba egy-egy határoló jelet tehetünk (közben későbbi mezők fölé nem is kerülünk).



3. ábra. A nem-determinisztikus gép determinisztikus szimulálása alatti példa konfiguráció

Az első munkaszalagon a lehetséges tanúkat soroljuk fel egyesével ($t(n)$ hosszban, hiszen hosszabb tanúnak nincs értelme: gépünk nem tudja elolvasni). A szimulálás a másik szalagon történik úgy, hogy a kitett határoló jeleken való túllépést nem tesszük meg, "TÚL-FUT" állapotba jutunk. Tehát egy tanú vizsgálata háromféleképpen végződhet: NEM-STIMMEL, BIZTOS-JÓ, TÚL-FUT. Ha az utolsó tanút is teszteltük, akkor KIMERÜLT állapotba kerülünk. Ezen fontos állapotok esetén leírjuk mit tesz a gép:

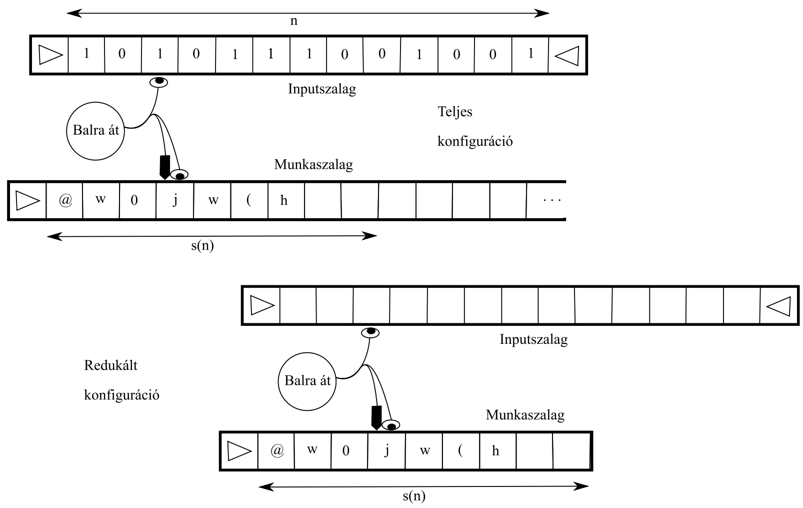
1. NEM-STIMMEL, ekkor a következő lehetőségre térünk át,
2. BIZTOS-JÓ, ekkor ELFOGAD állapotba kerülünk és leállunk,
3. TÚL-FUT, ekkor a következő lehetőségre térünk át,
4. KIMERÜLT, ekkor ELVET állapotba kerülünk és leállunk.

3. Lemma. *Ez a determinisztikus gép ugyanazt a nyelvet fogadja el, mint T és nem fut ki $t(n)$ tárból.*

A lemma és az észrevétel bizonyítja (i)-et.

(ii) bizonyítása: $L \in \mathcal{SPACE}(s(n))$. T nem-determinisztikus Turing-gép $s(n)$ tárkorláttal rendelkezik és L-et fogadja el.

Definíció. A redukált konfiguráció a teljes konfiguráció az inputszalag tartalma nélkül. (Az input részről csak az inputszem pozícióját tartalmazza.)



4. ábra. Egy konfiguráció és a hozzátartozó redukált konfiguráció

Egy adott ω-hoz tartozó ($n = |\omega|$) redukált konfigurációk száma legfeljebb

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

ahol konstansaink T-től függnek.

Ha a futás hosszabb lenne $\beta^{s(n)+logn}$-nél, akkor a futás során (redukált) konfigurációk ismétlődnének. Így a futás végtelen lenne, ami nem lehet. Így kaptuk, hogy T automatikusan olyan időigényű, ami (ii)-t igazolja.

(iii) bizonyítása:

Definíció. Legyen T nem-determinisztikus (I) Turing-gép. A G_ω konfiguráció gráf egy irányított gráf ($\omega \in \Sigma^n$ input), amely V csúcshalmaza a redukált konfigurációk halmaza. κ és λ csúcsok esetén κ-ból akkor és csak akkor vezet él λ-hoz, ha κ kiegészítve ω-val egy olyan konfiguráció, hogy az átmeneti függvény megengedje azt, hogy λ (ω-val kiegészítve) legyen a rákövetkező konfiguráció.

T-ről feltehető, hogy elfogadás „érzékelése” után mindent letöröl és a szalag elejére áll. Így az input elfogadása egy standard σ konfigurációban lesz gépünk

Definíció. A kiinduló konfigurációnak megfelelő csúcs legyen A (azaz $A \in V(G_\omega)$. A standard elfogadó konfiguráció felejen meg a Z csúcsnak G_ω-ban.

Könnyen ellenőrizhető, hogy $\omega \in L$ akkor és csak akkor, ha G_ω-ban van A-Z irányított út (igazából az A-Z irányított utak pontosan az elfogadó futásokkal leszoríthatók). Legyen T egy I-nem-determinisztikus Turing-gép, ami $L \in \mathcal{NSPACE}(s(n))$-t bizonyítja. L eldöntésére egy olyan Turing-gépet kell megadnunk ami időigénye legfeljebb $\alpha^{s(n)+logn}$.

A mi gépünk két fázisban működik:

1. fázis ω-t olvasva G_ω, A és Z egy kódját kiszámolja. Ezt megtervezhetjük úgy, hogy közben a munkaszalagon legfeljebb két redukált konfiguráció (G_ω csúcsa) szerepeljen, azaz tárigénye $\alpha_T(s(n)+logn)$. ((ii) alapján időigénye is korlátunkon belül marad).

2. fázis G_ω, A és Z ismeretében eldönti, hogy van-e $A-Z$ irányított út. Ez megoldható jól ismert algoritmuselméleti módszerek Turing-gépen való megvalósításával (például: mélységi keresés, szélességi keresés). Ezek az algoritmusok G_ω méretében ($\approx \beta_T^{s(n)+logn}$) polinomiális időben futnak.

A teljes futási idő az előírt korlát alatt marad. Ez (iii)-t bizonyítja. $\blacksquare$

Érdemes a bizonyítás egy fontos következményét és annak egy speciális esetén külön is megfogalmazni.

4. Lemma. *Legyen $L \in \mathcal{NSPACE}(s(n))$, ahol $s(n)$ egy szép-tár-függvény. Ekkor egy alkalmas Turing gép $\omega \in \Sigma^*$-ból $\mathcal{O}(s(n) + \log n)$ tárban kiszámol egy (G, a, z) hámast, ahol G egy irányított gráf, a, z két csúcsa úgy, hogy $\omega \in L$ akkor és csak akkor teljesül, ha G-ben van a-ból z-be vezető irányított út.*

5. Következmény. *Legyen $L \in \mathcal{NL}$. Ekkor egy alkalmas Turing gép $\omega \in \Sigma^*$-ból $\mathcal{O}(\log n)$ tárban kiszámol egy (G, s, t) hámast, ahol G egy irányított gráf, s, t két csúcsa úgy, hogy $\omega \in L$ akkor és csak akkor teljesül, ha G-ben van s-ből t-be vezető irányított út.*

Ezekután kiinduló ábránkat pontosíthatjuk:

$$\mathcal{L} \subset \mathcal{NL} \subset \mathcal{P} \subset \mathcal{NP} \subset \mathcal{PSPACE} \subset \mathcal{NPSPACE} \subset \mathcal{EXP} \subset$$
$$\subset \mathcal{NEXP} \subset \mathcal{EXPSPACE} \subset \mathcal{NEXPSPACE}.$$