ALGORITMUSOK ÉS BONYOLULTSÁGELMÉLET

Matematika MSc hallgatók számára

2. Előadás

Előadó: Hajnal Péter
Jegyzetelő: Bittner Emese 2010. február 8.

Turing-gép fogalmának változatai

Amikor a Turing-gép fogalmát leírtuk sok esetlegességet láthattunk (a szalagok száma, írhatóság, az output megjelenítése (miért kell új szalag hozzá), stb.). Valóban sokféle alternatív leírás található az irodalomban. Ezek azonban csak megállapodási különbségek. Mi két változatot ismertetünk.

Definíció. *k-szalagos Turing-gép* ($k \in \mathbb{N}$): az input- és az outputszalag mellett (az eredeti egy helyett) k darab munkaszalag van. Ennek megfelelően az átmeneti függvény értelmezési tartománya és értékkészlete is módosul. Ennek megfelelően a rákövetkező konfiguráció fogalma is máshogy néz ki. A gép által kiszámolt függvény, vagy eldöntött nyelv leírása azonban szószerint átmásolható.

A fenti definíció csak megemlít néhány változást, azokat nem adja meg. Ez az olvasóra van bízva.

Definíció. *Egyetlen szalagos Turing-gép*: egyetlen, egyik irányba végtelen szalagból áll, mely egyben az input-, a munka-, és az outputszalag is. Természetesen ez az egy szalag nem csak olvasható, írható is. A további módosításokat az olvasóra bízzuk. (Az output értelmézése maradhat: a szalag bal végét jelző jel és leálláskor fej alatt lévő mezők tartalma írja le.)

Az alternatív definíciók/megállapodások a kiszámíthatóság/eldönthetőség fogalmát nem változtatják meg. (Ez egy állítás, igazolását mindenki megpróbálhatja.)

Definíció. Legyen $\Sigma = \{0, 1\}$ és PALINDROM=$\{\omega \in \Sigma^*$palindrom szó$\}$, ahol egy $\omega = o_1 o_2 \dots o_n$ szóról akkor mondjuk, hogy palindrom, ha $o_i = o_{n+1-i}$, minden $i = 1, \dots, n$ esetén.

Készítsünk Turing-gépet annak eldöntésére, hogy egy adott szó benne van-e a PALINDROM nyelvben. A feladatra két mégoldást adunk két különböző modellben.

I) Standard modell

Az állapothalmazt nem soroljuk fel. Pontos leírása az algoritmus megértése után kidolgozható. Egy izelítőt adunk belőle: $S = \{$`"MÁSOLOK"`, `"ÁTMÁSOLTAM"`, `"TESZT"`,...$\}$

- 1. lépés: ω átmásolása a munkaszalagra (az inputszem az inputszalag végére kerül, a munkaszem a munkaterület valamelyik belső mezőjét szemléli);
- 2. lépés: a munkaszem szalag elejére állítása;

- 3. lépés: az inputszem balra, a munkaszem jobbra mozog. Vizsgáljuk az új mezőtartalmak egyezőségét. Ha valamely mezőnél nem egyeznek meg, akkor `ELVET`-jük, hogy $\omega \in PALINDROM$; ha mindenhol megegyeznek (azaz maradunk `TESZT` állapotban, amíg elérjük az inputszalag végét), akkor `ELFOGAD`-juk.

Vizsgálhatjuk a Turing-gép időigényét adott ω esetén. Minden $\omega \in \Sigma^*$ esetén

$$2|\omega| \leq TIME(\omega, T) \leq 3(|\omega| + 2).$$

Sőt minden a PALINDROM nyelvet eldöntő T_{tetsz} Turing-gép és minden $\omega \in \Sigma^*$ esetén

$$2|\omega| \leq TIME(\omega, T) \leq 3(|\omega| + 2).$$

II. Egyetlen szalagos modell

Legyen a munka-ábécé $\Gamma = 0, 1, 0^{\checkmark}, 1^{\checkmark}$. Az állapothalmazból ismét csak szemezgetünk: $S = \{$`"0-ÁT LÁTTAM, MEGYEK JOBBRA"`, `"1-ET LÁTTAM, MEGYEK JOBBRA"`, `"STIMMEL, MEGYEK BALRA"`, ... $\}$.

- 1. lépés: kipipálja az első pipálatlan o_i-t (felülírja o_i-t $o_i^{\checkmark}$-val), és megjegyzi o_i-t;
- 2. lépés: az utolsó pipálatlanra, o_{n+1-i}-re áll a szem;
- 3. lépés: A gép tesztel: megvizsgálja, hogy o_i és o_{n+1-i} megegyezik-e:
  - Amennyiben nem egyeznek meg, `ELVET`.
  - Amennyiben megegyeznek kipipálja, (felülírja a látott jelet pipált változatával), majd újra az 1. lépés (`STIMMEL, MEGYEK BALRA` állapotban).
  - Pipálatlan jelek elfogyása esetén `ELFOGAD`.

A Turing-gép időigénye adott $\omega \in PALINDROM$ esetén:

$$TIME(\omega, T) = \Theta(|\omega|^2).$$

A két algoritmus közül az elsőnél az időre adott felső becslés az input hosszában lineáris, a másodiknál négyzetes. Ez a különbség a modellekben rejlik.

1. Tétel. *Tegyük fel, hogy a T egyetlen szalagos Turing-gép eldönti PALINDROMot. Ekkor* $\exists\ \omega \in \sum^n$*, melyre*

$$TIME(\omega, T) \geq \alpha_T n^2.$$

Bizonyítás. Vizsgáljunk speciális inputokat, melyet elfogad a T Turing-gépünk (azaz palindrom szavak). Ezek alakja legyen $\omega = \omega_{eleje} 0^{\frac{n}{3}} \overleftarrow{\omega_{eleje}}$, ahol $\omega_{eleje} \in \{0, 1\}^{n/3}$ (feltesszük, hogy n osztható 3-mal, a fordított nyíl jobbról balra olvasást jelent). A kiemelt inputok száma $2^{n/3}$, ω_{eleje} meghatározza a speciális ω-t. A kijelölt inputokat indexelt I-vel jelöljük, I_j esetén a definiáló ω_{eleje} lesz j.

Az input középső harmadában lévő szomszédos mezők érintkezési oldalaira mint *ajtókra* hívatkozunk. A fej egy ajtónál fej felváltva fog egyszer az egyik, aztán pedig a másik irányba mozogni. Egy inputnál és ajtónál az inputon való futást figyelve egy állapotsorozatot rögzíthetünk. A szükséges fontos észrevételt kiemeljük.

Észrevétel. I_i és I_j különböző inputoknál ($i \neq j$) adott "p" ajtónál megfigyelt állapotsorozatok különbözőek.

Az észrevétel bizonyítása: Indirekten bizonyítunk. Tegyük fel, hogy $i \neq j$ esetén megegyeznek az állapotsorozatok.

Vegyünk egy új $\widetilde{I}$ inputot, melynek első fele I_i első harmadából, második fele I_j második harmadából áll (középen csupa 0-kat tartalmaz). A T Turing-gépünk $i \neq j$ esetén tehát nem fogadja el $\widetilde{I}$-ot, hiszen I_i első harmada visszafele olvasva nem egyezik meg I_j második harmadával ($i \neq j$).

Vizsgáljuk most hogy a p pozícióban lévő ajtótołbalra, illetve jobbra hogyan viselkedik a gépünk. A p ajtótól balra az I_i, jobbra pedig az I_j mozgását (illetve a szalagtartalom átírását) végzi a fej. (Az ajtó bal oldalán lévő megfigyelo, aki emlékszik az I_i-n történt futásra, azt tapasztalhatja, hogy most a fej több vagy kevesebb időt tölt az ajtó túl oldalán. Másban azonban nem érzékel eltérést: Ha visszatér a fej (indirekt feltevésünk alapján) ugyanabban az állapotban tér vissza mint eredetileg és így ugyanazt is teszi mint elózőleg.) tehát leálláskor is valamelyik futás állapotában lesz a gép. I_i-n és I_j-n is elfogadó volt a futás, tehát az új inputon is csak az lehet. Ez ellentmonda's. ■

A legfeljebb x hosszú állapotsorozatok száma felírható. Ebből egyszerűen adódik a következő állítás:

2. Lemma. *Ha* $1+|\Sigma|+|\Sigma|^2+...+|\Sigma|^x < \frac{1}{2}2^{\frac{n}{3}}$, *akkor legalább az inputok fele* x*-nél többször lép át egy* p *ajtón. Ekkor* $x+1 = \lfloor \frac{\frac{n}{3}-1}{log_2|\Sigma|} \rfloor$, *azaz* $x \sim \alpha_T n$.

Ezekután átlag futási idő speciális inputjainkon könnyen becsülhető:

$$\frac{1}{2^{\frac{n}{3}}} \sum_{i \in \{0,1\}^{n/3}} TIME(I_i, T) \geq \frac{1}{2^{\frac{n}{3}}} \sum_{p \text{ pozíciója}} \sum_{i \in \{0,1\}^{n/3}} (I_i \text{ átlépéseinek száma } p\text{-n}) \geq$$

$$\geq \frac{1}{2^{\frac{n}{3}}} \sum_{p \text{ pozíciója}} \frac{1}{2} 2^{\frac{n}{3}} \alpha_T n \geq \frac{n}{3}\frac{1}{2}\alpha_T n = \alpha'_T n^2,$$

ahol az első egyenlőtlenségben a teljes idő helyett csak a megfigyelt (középső ajtókon való áthaladásokat), a második egyenlőlenségben pedig csak a Lemma-beli inputokat (amelyek sokszor lépnek át egy adott ajtón) számoljuk.

A speciális inputokra vonatkozó átlag futásra vonatkozó alsó becslés alapján a leghoszabb futás hosszára nyilvánvaló a tétel alsó becslése. ■

Ezekután bevezetjük a következő bonyolultsági osztályokat. Előző tételünk arra figyelmeztet, hogy nem mindegy melyik modellel dolgozunk. A legelfogadottabb utat követjük: a definícióban szereplő Turing-gép valamely $k \in N$ esetén k-szalagos Turing-gép

Definíció. Legyen $f : \mathbb{N} \to \mathbb{R}$.

$$TIME(f(.)) = \{L \text{ nyelv: } \exists\, T \text{ Turing-gép, amely } (i) \text{ eldönti } L\text{-et} \quad (ii)\ TIME(\omega, T) \leq f(|\omega|)\},$$

$$SPACE(f(.)) = \{L \text{ nyelv: } \exists\, T \text{ Turing-gép, amely } (i) \text{ eldönti } L\text{-et} \quad (ii)\ SPACE(\omega, T) \leq f(|\omega|)\}.$$

Az idő és tár igény bevezetése természetesen vezet el a megfelelő nyelvosztályok definíciójához:

Definíció.

$$\mathcal{P} = \bigcup_{\alpha \in \mathbb{N}} TIME(\alpha n^{\alpha})$$

$$\mathcal{EXP} = \bigcup_{\alpha \in \mathbb{N}} TIME(2^{\alpha n^{\alpha}})$$

$$\mathcal{PSPACE} = \bigcup_{\alpha \in \mathbb{N}} SPACE(\alpha n^{\alpha})$$

$$\mathcal{EXPSPACE} = \bigcup_{\alpha \in \mathbb{N}} TIME(2^{\alpha n^{\alpha}})$$

$$\mathcal{L} = \bigcup_{\alpha \in \mathbb{N}} SPACE(\alpha log_2 n)$$

Megjegyzés. Megjegyezzük, hogy ezen osztályok definíciójában ismét elmosódik az, hogy a Turing-gép definíciójának mely változatával dolgozunk.

Megjegyzés. Az unió mögött szereplő bonyolultsági osztályok korlátai esetlegesek. Például $\mathcal{P}$-ben szereplő $\mathcal{A} = \{\alpha n^{\alpha} : \alpha \in \mathbb{N}\}$ polinomok halmaza azért jó, mert minden polinomhoz van $\mathcal{A}$-beli, ami felülről becsüli $\mathbb{N}$-re megszorítva. Ha egy T Turing-gép olyan, hogy $\mathcal{P}$-hez tartozást bizonyít, akkor azt mondjuk, hogy T polinomiális.

Megjegyzés. $\mathcal{L}$ helyett néha $\mathcal{LOGSPACE}$-t használunk. Mivel időbonyolultság esetén a logaritmikus korlátnak nincs sok értelme, ezért írunk a hosszú jelölés helyett általában csak $\mathcal{L}$-et.

Az osztályok között fennállnak bizonyos tartalmazási relációk. Ezek közül az alábbiak triviálisak: $\mathcal{L} \subseteq \mathcal{PSPACE} \subseteq \mathcal{EXPSPACE}$, $\mathcal{P} \subseteq \mathcal{PSPACE}$, $\mathcal{P} \subseteq \mathcal{EXP}$, $\mathcal{EXP} \subseteq \mathcal{EXPSPACE}$.

Később látni fogjuk: $\mathcal{L} \subseteq \mathcal{P}$, $\mathcal{PSPACE} \subseteq \mathcal{EXP}$.

Turing-gépek kódolása

Mint minden, a Turing-gépek is kódolhatók. Ehhez kell egy Σ_u = "univerzális abc", ami lehet például $\{0, 1\}$ és egy megállapodás, hogy a definíció egyes összetevőit, komponenseit hogyan írjuk le egy karaktersorozattal. Ezt nem részletezzük.

Turing-gép kódolása: $T(\Sigma, S, \Gamma, \delta) \longmapsto \ulcorner T \urcorner \in \Sigma_u^*$, $\omega \in \Sigma^* \longmapsto \ulcorner \omega \urcorner \in \Sigma_u^*$.

3. Tétel. *Legyen $\Sigma = \Sigma_u \cup \{\ ,\}$, jelöljük $T(\omega)$-val az ω-ból T által kiszámított outputot. Ekkor konstruálható olyan U Turing-gép, hogy U a $(\ulcorner T \urcorner, \ulcorner \omega \urcorner)$ inputon $\ulcorner T(\omega) \urcorner$-t számolja ki.*

A tételt bizonyító U gépet univerzális Turing-gépnek nevezzük. Persze egy konkrét univerzális Turing-gép felírásához sok (esetleges) megállapodást kell kötnünk és átmenetei függvénye különböző prograzmozási stílusban lehet megfogalmazva.

Ezek után leírjuk, hogy a bevezetett fogalmak milyen hétköznapi értelmezéssel bírnak. Egy szótárba foglaljuk eddigi fogalmainkat:

Bonyolultságelmélet:	*Programozás:*
Turing-gép	Algoritmus
Turing-gépek kódolása	Programozási nyelv
$\ulcorner T \urcorner$	Program
U	Számítógép

Turing-gépek kódolása után definiáljunk egy kapcsolódó nyelvet:

MEGÁLLÁS$=\{(\ulcorner T \urcorner, \ulcorner \omega \urcorner)$: T leáll ω-n, azaz `STOP` vagy `ELFOGAD/ELVET` állapotba kerül $\}$

Definíció.

A következő tétel elméleti szempontból nagyon jelentős.

4. Tétel (Turing-tétel). *A MEGÁLLÁS nyelv nem eldönthető.*

Bizonyítás. Bizonyítsunk indirekten. Tegyük fel, hogy létezik M Turing-gép, amely eldönti MEGÁLLÁS-t.

Ekkor leírunk egy új gépet: $\widetilde{M}$: ω inputról megnézi, hogy Turing-gépet kódol-e. Amennyiben ω nem kódol Turing-gépet bármit tehet gépünk, mondjuk egyből elveti ω-t. Ha $\omega = \ulcorner T \urcorner$, akkor kiszámolja $M(\ulcorner T \urcorner, omega)$-t (azaz megnézi T leáll-e saját kódján).

- ha M `ELFOGAD`-ja, akkor balra-jobbra lépked — végtelen ciklusba kerül;
- ha M `ELVET`, akkor `ELFOGAD`-ja.

Mit csinál a $\widetilde{M}$ gép, ha saját kódjával futtatjuk? Vagy leáll (ELFOGAD állapottal), vagy végtelen ciklusba kerül. Futása bárhogy is végződjön, ellentmondásra jutunk. ■

Észrevétel. MEGÁLLÁS felsorolható.