ALGORITMUSOK ÉS BONYOLULTSÁGELMÉLET

Matematika MSc hallgatók számára

3. Előadás

Előadó: Hajnal Péter
Jegyzetelő: Sallai Gyöngyi 2011. február 15.

1. Eldöntő Turing-gépek

Emlékeztető. $L \subseteq \Sigma^*$ nyelv pontosan akkor eldönthető, ha létezik T Turing-gép, amely minden $\omega \in L$ esetén ELFOGAD állapotba jut és minden $\omega \notin L$ esetén ELVET állapotba jut.

Ekkor az outputszalagra mindössze 1 bit kerül leírásra. Így az outputszalagra megspórolható, az output bejelentése megtörténhet az ELFOGAD és ELVET állapotokkal.

Jelölés. Az eldönthető nyelvek halmazát $\mathcal{D}$ jelöli, azaz

$$\mathcal{D} := \{L : L \text{ eldönthető}\} \subseteq \mathcal{P}(\Sigma^*).$$

Egy másik lehetőséget kapunk az outputbit „kódolására”, ha az elvetést egy külön állapot helyett végtelen futással jelöljük.

Definíció. $L \subseteq \Sigma^*$ nyelv pontosan akkor *felsorolható*, ha létezik T Turing-gép, amely bármely $\omega \in L$ esetén ELFOGAD állapottal áll le és bármely $\omega \notin L$ esetén végtelen futása lesz.

Ez az utóbbi definíció gyengébb az eldöntésnél, gyakorlati hátránya is van: ha túl sokáig fut az algoritmus, nem lehet tudni, hogy csak lassú, vagy épp végtelen futásban van.

Jelölés. A felsorolható nyelvek halmazát $\mathcal{S}$ jelöli, azaz

$$\mathcal{S} := \{L : L \text{ felsorolható}\} \subseteq \mathcal{P}(\Sigma^*).$$

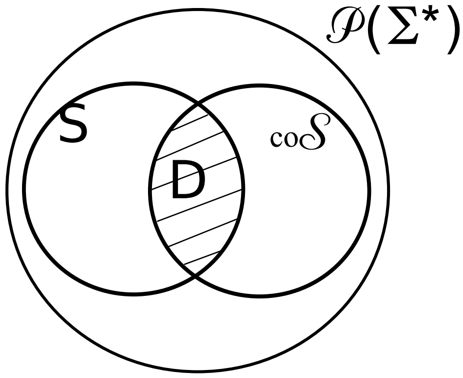
Definíció. Legyen $\mathcal{C} \subseteq \mathcal{P}(\Sigma^*)$ nyelvosztály, ekkor $co\ \mathcal{C} := \{\overline{L} : L \in \mathcal{C}\}$ jelöli $\mathcal{C}$ komplemeterét.

Példa. $co\ \mathcal{S} = \{L : L \text{ komplementere felsorolható}\}$

Észrevétel. $co\ \mathcal{D} = \mathcal{D}$. Valóban: Egy $L \in \mathcal{D}$-t bizonyító Turing-gép esetén az ELVET, ELFOGAD állapotok felcserélésével olyan géphez jutunk, ami $\overline{L}$ eldönthetőségét igazolja.

Észrevétel. A következő tartalmazásokat írhatjuk fel az előbb bevezetett nyelvhalmazok között:

- $\mathcal{S} \supseteq \mathcal{D}$, mivel az eldöntő Turing-gép esetében az ELVET állapotot helyett egy ∞ állapotot vehetünk fel, amelyben a gép ciklizál.



1. ábra.

- $co\ \mathcal{S} \supseteq \mathcal{D}$, ebben az esetben az ELFOGAD állapotot kell ∞ állapotra cserélni.
- $\mathcal{D} = \mathcal{S} \cap co\ \mathcal{S}$. Egyszerű állítás, lásd gyakorlat.

Megjegyzés 1.

- $|\mathcal{P}(\Sigma^*)| = c = 2^{\aleph_0}$. Nyilván, mert véges nem-üres Σ esetén $|\Sigma^*| = \aleph_0$.
- $|\mathcal{S}| = |co\ \mathcal{S}| = |\mathcal{D}| = \aleph_0$. Nyilván, mert olyan Turing-gépből, amelyre $\Gamma = [n]$ és $S = [m]$ csak véges sok van. Az ilyen Turing-gépekkel minden Turing-gép szimulálható.

Megjegyzés 2. $\mathcal{D} \subsetneq \mathcal{S}$ esetén $\mathcal{S} \neq co\ \mathcal{S}$ is teljesül.

2. Turing-gépek kódolása

A Turing-gépek is kódolhatóak, ehhez rögzítsünk egy Σ_0 univerzális ábécét. Például legyen most $\Sigma_0 = \{0, 1\}$. Rögzítsük Turing-gépeknek egy kódolását. Egy T Turing-gép esetén jelöljük kódját $\lceil T \rceil \in \Sigma_0^* = \{0, 1\}^*$. Az $\omega \in \Sigma^*$-hoz rendeljük hozzá az $\lceil \omega \rceil \in \Sigma_0^* = \{0, 1\}^*$ kódot. A $(\lceil T \rceil, \lceil \omega \rceil)$ pár kódja legyen $\lceil T, \omega \rceil$.

1. Állítás (Univerzális Turing-gép, Alan Turing). *Legyen $\Sigma_0 = \{0, 1\}$. Létezik U Turing-gép, ami az inputszalag tartalmát $\lceil T, \omega \rceil$-nak fogja fel, és T-t szimulálja ω-n. Azaz*

$$\begin{aligned} T\ \omega\text{-n ELFOGAD} &\Leftrightarrow U \text{ is ELFOGAD } \lceil T, \omega \rceil\text{-n} \\ T\ \omega\text{-n ELVET} &\Leftrightarrow U \text{ is ELVET } \lceil T, \omega \rceil\text{-n} \\ T\ \omega\text{-n } \infty \text{ futás} &\Leftrightarrow U \text{ is } \infty \text{ futás } \lceil T, \omega \rceil\text{-n.} \end{aligned}$$

Az állítás formális bizonyításához konkrétan meg kell állapodni Turing-gépek egy kódolásában és U átmenetifüggvényével le kellene írni, hogy egy $\lceil T \rceil$ kódból hogy olvassuk ki T lényegét és ezt hogyan szimuláljuk az inputból kiolvasott ω-n. Ez nem nehéz, de nagyon fáradságos munka.

$\star$

Az eddigi ismeretek alapján egy „szótárat” állíthatunk össze

Hétköznapi élet	Bonyolultságelmélet
Eldöntési probléma	L nyelv
Számítási probléma	$f : \Sigma^* \to \Sigma^*$
Algoritmus	T Turing-gép
Programozási nyelv	$T \mapsto \lceil T \rceil$
Számítógép/interpreter	U univerzális Turing-gép

Definíció.

$$\text{MEGÁLLÁS} = \{\lceil T, \omega \rceil : T \text{ leáll } \omega\text{-n, azaz STOP} \\ \text{vagy ELVET/ELFOGAD állapotba kerül}\}.$$

2. Tétel (Turing-tétel). *(i) MEGÁLLÁS $\in \mathcal{S}$,*

(ii) MEGÁLLÁS $\notin \mathcal{D}$.

Bizonyítás. A tétel első része következik az előző állításból. Az ott definiált univerzális Turing-gépet kell úgy módosítani, hogy a szimulálandó gép leállásánál, ne a leálló állapotnak megfelelő állapotba jusson, hanem csak a leállás ténye miatt ELFOGAD állapotba kerüljön.

A második állítás bizonyítása indirekten történik, azaz tegyük fel, hogy létezik I Turing-gép, amely eldönti a MEGÁLLÁS nyelvet. Továbbiakban az indirekt feltevés I gépére alapítva egy kissé módosított gépet írunk le.

Definíció. Legyen $\widetilde{I}$ egy Turing-gép, amely eldönti, hogy i input kódol-e Turing-gépet. Amennyiben nem a gép ELVET állapotba kerül. Amennyiben $i = \lceil T \rceil$ az I MEGÁLLÁS nyelvet eldöntő Turing-gépet futtatja (i, i)-n. Ha a futtatás ELFOGAD állapottal ér véget, akkor jobbra-balra lépegető végtelen ciklusba megy át, ha ELVET állapottal ér véget, akkor STOP állapottal leáll.

Kérdés: Mit csinál $\widetilde{I}$ gép $\lceil \widetilde{I} \rceil$-n? Azaz mi történik, ha a saját kódján futtatjuk az $\widetilde{I}$ gépet?

A definíció alapján „kibontja magát”, és a MEGÁLLÁS nyelvet eldöntő géppel eldönti, hogy a megadott inputon (ami esetünkben saját kódján) hogyan dolgozik. Ha I ELFOGAD állapotba kerül (a $\widetilde{I}$ gép $\lceil \widetilde{I} \rceil$-n megáll), akkor nem áll le (mármint $\widetilde{I}$ az $\lceil \widetilde{I} \rceil$-n), és ha I ELVET álpotba kerül (a $\widetilde{I}$ gép $\lceil \widetilde{I} \rceil$-n nem áll meg), akkor leáll. Mindenféleképpen ellentmondásra jutunk. ∎

A bizonyítás lényege hasonlít Cantor bizonyítására, hogy $[0, 1]$ nem felsorolható/megszámlálhatóan végtelen halmaz (átlós módszer), csak ebben az esetben számok és sorszámok helyett gépek, illetve inputok kódjai szerepelnek.

A kurzus továbbiakban részében a $\mathcal{D}$ halmaz nyelveivel dolgozunk. Célunk az eldöntési problémák összehasonlítása, a döntési feladatok nehézségének mérése.

3. Bonyolultsági osztályok

Definíció. T Turing-gép időigénye ω inputon:

$$TIME(T, \omega) = \begin{cases} \min\{i : \kappa_i \text{ állapota ELFOGAD/ELVET}\}, & \text{ha } T \text{ leáll } \omega\text{-n} \\ \infty, & \text{különben.} \end{cases}$$

Definíció. T Turing-gép tárigénye ω inputon:

$$SPACE(T,\omega) = \begin{cases} \max\{j : \quad \omega\text{-n a } T \text{ futása során } j \text{ távolságra kerül } \triangleright\text{-től}\}, \\ \qquad\qquad\qquad\qquad \text{ha halmazunk felülről nem korlátos,} \\ \infty, \qquad\qquad\qquad\quad \text{különben.} \end{cases}$$

Az idő- és tárigény definiálása természetesen vezet egy nyelv bonyolultságának bevezetéséhez és bonyolultsági nyelvosztályok bevezetéséhez.

Definíció. L nyelv pontosan akkor dönthető el polinomiális időben, ha létezik T Turing-gép, melyre a következő két feltétel teljesül,

1. L-et dönti el,
2. van olyan $i \in \mathbb{N}$, hogy minden $\omega \in \Sigma^*$ inputra $TIME(T,\omega) \leq |\omega|^i + i$.

$\mathcal{P}$ jelöli polinomiális időben eldönthető nyelvek osztályát.

Definíció. L nyelv pontosan akkor exponenciális idejű, ha létezik T Turing-gép, melyre a következő két feltétel teljesül,

1. L-et dönti el,
2. van olyan $i \in \mathbb{N}$, hogy minden $\omega \in \Sigma^*$ inputra $TIME(T,\omega) \leq 2^{|\omega|^i + i}$.

$\mathcal{EXP}$ jelöli az exponenciális időben eldönthető nyelvek osztályát.

Definíció. L nyelv pontosan akkor logaritmikus tárban eldönthető, ha létezik T Turing-gép, melyre a következő két feltétel teljesül,

1. L-et dönti el,
2. $\exists i \in \mathbb{N}\ \forall \omega \in \Sigma^*\ SPACE(T,\omega) \leq i \log{(n+i)}$.

$\mathcal{L}$ (egy bőbeszédűbb jelöléssel $\mathcal{LOGSPACE}$) a logaritmikus tárban eldönthető nyelvek osztálya.

Definíció. L nyelv pontosan akkor dönthető el polinomiális tárral, ha létezik T Turing-gép, melyre a következő két feltétel teljesül,

1. L-et dönti el,
2. van olyan $i \in \mathbb{N}$, hogy minden $\omega \in \Sigma^*$ inputra $SPACE(T,\omega) \leq |\omega|^i + i$.

$\mathcal{PSPACE}$ a polinomiális tárban eldönthető nyelvek osztálya.

Definíció. L nyelv pontosan akkor dönthető el exponenciális tárban, ha létezik T Turing-gép, melyre a következő két feltétel teljesül,

1. L-et dönti el,
2. van olyan $i \in \mathbb{N}$, hogy minden $\omega \in \Sigma^*$inputra $SPACE(T,\omega) \leq 2^{|\omega|^i + i}$.

$\mathcal{EXPSPACE}$ az exponenciális tárban eldönthető nyelvek osztálya.

Megjegyzés. A Turing-gép robosztusossága például azt jelenti, hogy a fenti definícióban leírt osztályok nem függnek a modell apró részleteitől. Alternatív megállapodások ugyanahhoz a $\mathcal{P}$ nyelvosztályhoz vezetnek.

Az osztályok között nyilvánvaló tartalmazási relációk állnak fent. Idő (és tár esetén is) a nagyobb korlát bővebb nyelvosztályhoz vezet, illetve bizonyos idő korlát a fej korlátos mozgása miatt tár korlátot is magával hoz:

$$\begin{array}{ccccc} & & \mathcal{P} & \subseteq & \mathcal{EXP} \\ & & \cap\!| & & \cap\!| \\ \mathcal{L} & \subseteq & \mathcal{PSPACE} & \subseteq & \mathcal{EXPSPACE} \end{array}$$

Későbbiekben további tartalmazásokat is felírunk.

4. Nem-determinisztikus Turing-gépek

A nem-determinisztikus Turing gépekre két alternatív definíciót is adunk.

Definíció (I. változat). Hasonlóan mint a determinisztikus Turing-gépeknél, itt is vannak szalagok, fejek, állapotok, stb. Azonban az átmeneti függvényt már máshogy definiáljuk:

$$\delta\colon \Sigma \times \Gamma \times S \to \mathcal{P}(\{\leftarrow, ., \to\} \times \Gamma \times \{\leftarrow, ., \to\} \times S) \setminus \{\emptyset\}.$$

Az ω inputhoz tartozó futás nem meghatározott (idegen szóval nem-determinisztikus), azaz a $\kappa_0(\omega)$ kezdőkonfigurációból több lehetséges konfiguráció felé mehetünk, és mindegyik konfigurációnál több rákövetkező konfiguráció lehetséges. Ily módon egy $\kappa_0(\omega)$-ban gyökereztetett fa írja le a gép lehetséges futásait.

Az ω inputot pontosan akkor fogadja el a T nem-determinisztikus Turing-gép, ha létezik ELFOGAD futása, és elveti, ha minden futása ELVET állapothoz vezet.

Definíció (II. változat). Ebben az esetben egy plusz szalagunk lesz az input- és munkaszalagok között, az úgynevezett tanú/bizonyítás szalag. Ez a szalag csak olvasható és a fej csak jobbra tud mozogni rajta. Az átmeneti függvényt ugyanúgy definiáljuk, mint a determinisztikus esetben, és a futás is determinisztikus lesz, azaz ω és τ (a tanúszalag tartalma) egyértelműen meghatároz egy konfigurációsorozatot:

$$\kappa_0 = \kappa_0(\omega, \tau) \to \kappa_1 \to \kappa_2 \to \dots.$$

Az ω-t pontosan akkor fogadja el egy nem-determinisztikus Turing-gép, ha van olyan τ tanúszalag tartalom, amelyre a futás ELFOGAD állapotba kerül.

A két változatban egy lényeges különbség, hogy az elsőben a nem-determinizmus a futás során „szétszórt”, az utolsó lépés előtt sem meghatározott a végső állapot. A második változatban a nem-determinizmus a τ választásával jelentkezik. A futás ezekután determinisztikus lesz.

Az alábbiakban a nemdeterminisztikus számításon alapuló nyelvosztályokat vezetjük be. A nem-determinizmus kétféle változata közül bármelyikre alapozzuk definícióinkat, ugyanahhoz a nyelvosztályhoz jutunk.

Definíció. Az L nyelv nem-determinisztikus polinomiális időben eldönthető, ha létezik T nem-determinisztikus Turing-gép, amelyre

1. L-et dönti el,

2. van olyan $i \in \mathbb{N}$, hogy minden $\omega \in \Sigma^*$ inputra alkalmas τ esetén

$$TIME(T, \omega, \tau) \leq |\omega|^i + i.$$

$\mathcal{NP}$ jelöli nem-determinisztikus polinomiális időben eldönthető nyelvek osztályát.

Definíció. Az L nyelv nem-determinisztikus exponenciális időben eldönthető, ha létezik T nem-determinisztikus Turing-gép, amelyre

1. L-et dönti el,

2. van olyan $i \in \mathbb{N}$, hogy minden $\omega \in \Sigma^*$ inputra alkalmas τ esetén

$$TIME(T, \omega, \tau) \leq 2^{|\omega|^i + i}.$$

$\mathcal{NEXP}$ jelöli nem-determinisztikus exponenciális időben eldönthető nyelvek osztályát.

Definíció. Az L nyelv nem-determinisztikus logaritmikus tárban eldönthető, ha van olyan nem-determinisztikus Turing-gép, amelyre

1. L-et dönti el,

2. van olyan $i \in \mathbb{N}$, hogy minden $\omega \in \Sigma^*$ inputra alkalmas τ esetén

$$SPACE(T, \omega, \tau) \leq i \log(|\omega| + i).$$

$\mathcal{NL}$ jelöli nem-determinisztikus logaritmikus tárban eldönthető nyelvek osztályát.

Definíció. Az L nyelv nem-determinisztikus polinomiális tárban eldönthető, ha van olyan nem-determinisztikus Turing-gép, amelyre

1. L-et dönti el,

2. van olyan $i \in \mathbb{N}$, hogy minden $\omega \in \Sigma^*$ inputra alkalmas τ esetén

$$SPACE(T, \omega, \tau) \leq |\omega|^i + i.$$

$\mathcal{NPSPACE}$ jelöli nem-determinisztikus polinomiális tárban eldönthető nyelvek osztályát.

Definíció. Az L nyelv nem-determinisztikus exponenciális tárban eldönthető, ha van olyan nem-determinisztikus Turing-gép, amelyre

1. L-et dönti el,

2. van olyan $i \in \mathbb{N}$, hogy minden $\omega \in \Sigma^*$ inputra alkalmas τ esetén

$$SPACE(T, \omega, \tau) \leq 2^{|\omega|^i + i}.$$

$\mathcal{NEXPSPACE}$ jelöli nem-determinisztikus exponenciális tárban eldönthető nyelvek osztályát.

Megjegyzés. Ismét megjegyezzük, hogy (a nem-determinizmus kétféle változatán túl) az alapmodell különféle megállpodásait használva a definiált osztályok nem változnak.

$\star \qquad \star \qquad \star$

Miért csak egy tanú esetén kívántuk meg az idő illetve tár korlátot? Szép függvények esetén ez lényegtelen.

Definíció. A $t(n)$ *szép idő-függvény*, ha van olyan T Turing-gép, amely minden $\omega \in \Sigma^n$ inputon pontosan $t(n)$ lépés után „LETELT” állapotba kerül.

A T Turing-gépre gondolhatunk úgy, mint egy „óra”, ami $t(n)$ lépés után „csörög”.

Definíció. Az $s(n)$ *szép tár-függvény*, ha van olyan T Turing-gép, amely minden $\omega \in \Sigma^n$ inputra a munkaszalag első $s(n)$ mezőjéből nem lép ki, és az $s(n)$-edik mezőre „VÉGE” jelet ír. Ez a jel speciális csak erre szolgál.

Azaz minden futás során a speciális karakter leírása kijelöl egy $s(n)$ hosszú kezdőszeletét a munkaszalagnak, ami után csak szűz karakterek vannak.

Észrevétel. A „barátságos” függvények szépek, mint például n, n^2, $\binom{n}{3}$, 2^n szép idő-függvények, és $\lceil \log n \rceil$, n, n^{2011} szép tár-függvények.

3. Lemma. *Ha $t(n)$ szép idő-függvény, akkor annak a feltevése, hogy alkalmas τ-ra a futás legfeljebb $t(n)$ hosszú, megegyezik azzal, hogy minden futás legfeljebb $t(n) + 2n + 2$ hosszú.*

Bizonyítás. Az inputot másoljuk át egy munkaszalagra, majd az input szemet és a másolást végző munka szemet is mozgassuk vissza a szalag elejére. Ehhez $2|\omega| + 2$ lépés kellett. Ezekután a nem-determinisztikus gépet és a $t(n)$-hez tartozó „órát” egyszerre/párhuzamosan szimuláljuk. Ha a szimulálás leáll, akkor ugyanazzal az állapottal mi is leállunk. Ha az óra „csörög” mielőtt a szimulációnak vége van, akkor leállunk ELVET állapottal. Ugyanazt a nyelvet ismertük fel. ■

Megjegyzés. Igazából az is kiderült, hogy elég csak $\omega \in L$ esetén feltenni az időkorlátot. Ha $\omega \notin L$ inputokra minden futás „hosszú” lenne, akkor a óra csörgése miatt jutnánk mindig ELVET állapotba (ahogy kell). Az is kiderült, hogy a tanúszalag tartalmából csak az első $t(n)$ karakter számít. Hoszabb tanút nem tudunk elolvasni az idő korláton belül. Azaz NINCS ÉRTELME hosszabb tanúval dolgozni.

Természetesen a tár esetén is igaz a megfelelő lemma.

4. Lemma. *Ha $s(n)$ szép tár-függvény, akkor annak a feltevése, hogy alkalmas τ-ra a futás legfeljebb $s(n)$ tárat igényel, megegyezik azzal, hogy minden futás legfeljebb $s(n)$ tárat igényel.*

$\star \qquad \star \qquad \star$

Ezekután kibővíthetjük a korábban felírt osztályok közötti tartalmazás listát:

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

Ennél persze több összefüggés is létezik. Ezek közül többet jövő héten igazolunk.

És persze jóval több bonyolultsági osztály létezik. Néhánnyal még találkozunk. A teljesség igénye azonban reménytelen. Akiket több osztály érdekel az keressen rá az interneten a „complexity zoo” kulcsszóra.