

**2. Előadás***Előadó: Hajnal Péter**Jegyzetelő: Görbe Tamás Ferenc*

2012. február 15.

**1. A standard modell és változatai**

Az előző előadásban bevezettünk egy számítási modellt, amelyet a későbbiek során *standard modellnek* fogunk nevezni. Eleveítsük fel a modell részeit!

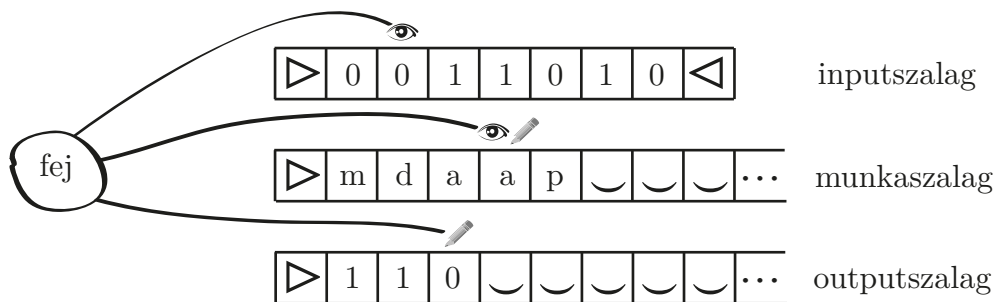
A standard modellben a számításokhoz három eltérő funkcióval rendelkező szalagot használunk. Ezek egy irányban (általában jobbra) végtelen mezősorozatok, a nevük: *inputszalag*, *munkaszalag*, *outputszalag*. A Turing-gép feje a szalagok jellegétől függően a mezők leolvasását és felülírását végzi. Az *inputszalag* a  $\Sigma$  ábécé karaktereiből álló véges hosszúságú *szót* tartalmazza  $\triangleright$  és  $\triangleleft$  szimbólumok között, amelyek az input kezdetét és végét jelzik. A fej karakterenként látja az inputot egy szemmel, amely az inputszalagon mindkét irányban képes mozogni. A *munkaszalag* első mezője szintén a szalag kezdetét jelölő  $\triangleright$  „bevésett” (nem felülírható) karaktert tartalmazza. A szalag mezőibe a  $\Gamma$ -val jelölt ún. *munkaábécé* karaktereit írhatjuk. A fej a munkaszalag mezőit egy jobbra-balra együtt mozgó szemmel és „ceruzát tartó” kézzel olvassa, ill. írja. Végül az ugyancsak  $\triangleright$  jellel kezdődő *outputszalagra* kerül a számítás eredménye, melyen a fej egy „ceruzát tartó” kézzel ír, a mezőkön egyenként jobbra haladva. A fejnek egy fontos jellemzője az állapota, amely a véges  $S$  állapothalmaz eleme.

A Turing-gép működését az  $f$  átmeneti függvény szabja meg. Ennek értelmezési tartományát az ún. *látható részek* alkotják, amelyek alapján  $f$  egy „update-elést” (frissítést) hajt végre. A látható részek a fej

- (1) inputszalagon futó szeme által látott karakter,
- (2) munkaszalagon futó szeme által látott karakter,
- (3) állapota.

Az átmeneti függvény megadja, hogy a látható részek alapján

- (4) milyen irányba lépjen az inputszalagot pásztázó szem,
- (5) milyen karakter kerüljön a munkaszalag aktuális mezőjébe,
- (6) merre mozduljon a munka szem-kéz,
- (7) mi kerüljön az outputszalagra,
- (8) végül pedig meghatározza, hogy a fej milyen állapotba kerüljön.

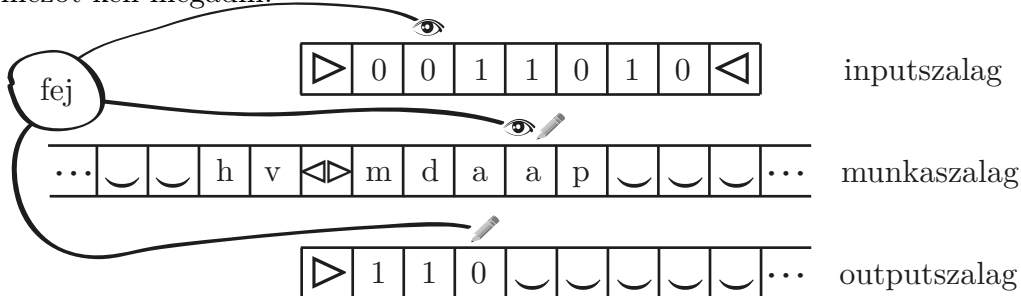


1. ábra. A standard modell.

$$f: \underbrace{(\Sigma \cup \{\triangleright, \triangleleft\})}_{(1)} \times \underbrace{(\Gamma \cup \{\triangleright, \smile\})}_{(2)} \times \underbrace{S}_{(3)} \rightarrow \underbrace{\{\leftarrow, \cdot, \rightarrow\}}_{(4)} \times \underbrace{\Gamma}_{(5)} \times \underbrace{\{\leftarrow, \cdot, \rightarrow\}}_{(6)} \times \underbrace{(\Sigma \cup \{.\})}_{(7)} \times \underbrace{S}_{(8)}$$

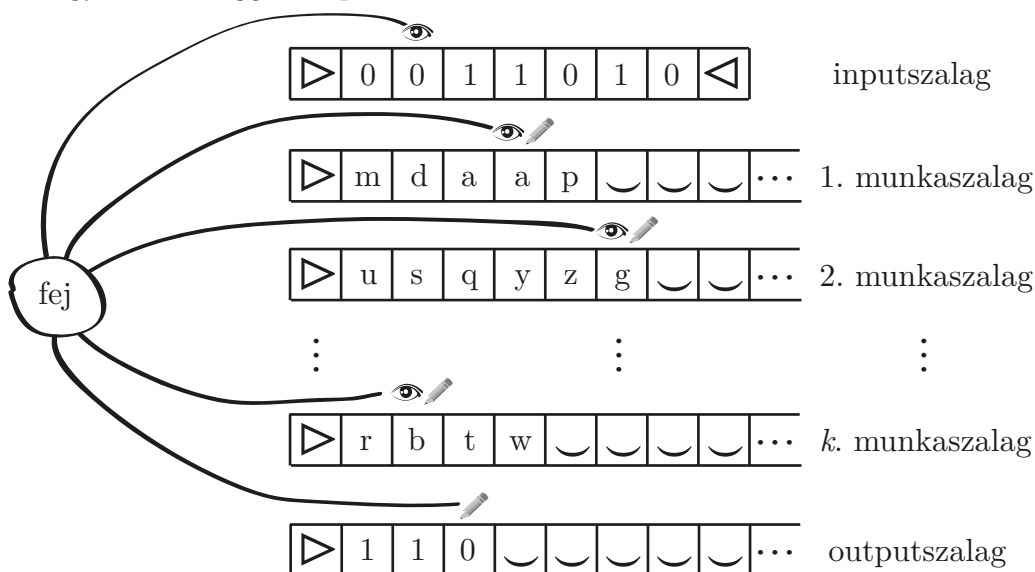
A fent leírt standard modell esetleges, hiszen például az, hogy egy irányban végtelen szalagokat használunk továbbá, hogy éppen három szalagon dolgozunk önkényes választások. Így természetesen léteznek más változatok is, íme néhány példa:

**Példa.** • **Két irányban végtelen munkaszalag:** A munkaszalag bal és jobb irányban is végtelen és nem tartalmaz 'szalag eleje' jelet, csupán egy kezdeti mezőt kell megadni.



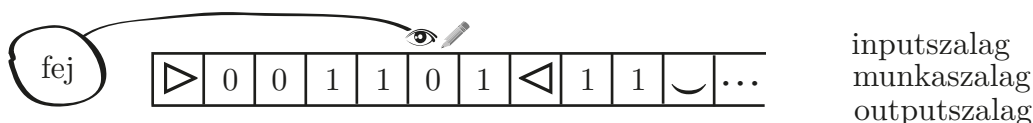
2. ábra. A két irányban végtelen munkaszalag.

•  **$k$ -szalagos modell:** Rögzített  $k \in \mathbb{N}$  számú munkaszalagot használunk, és minden egyes munkaszalaghoz tartozik egy szem-kéz páros. Fontos megjegyezni, hogy  $k$  nem függ az input méretétől.



3. ábra. A  $k$ -szalagos modell.

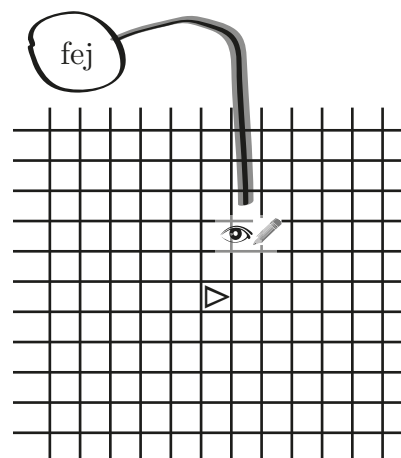
- **Egyszalagos modell:** Minden művelet egyetlen szalagon történik, amely így egyszerre input-, munka-, és outputszalag. A szalag jobbra végtelen és a kezdeti mezőket az input foglalja el  $\triangleright, \triangleleft$  jelek között. Ebben a modellben az input is felülírható és az output a szalag tartalma a  $STOP \in S$  állapot elérésekor.



4. ábra. Az egyszalagos modell.

- **Kétdimenziós munkaszalag:** Érdemes a munkaszalagot úgy elképzelni, mint egy végtelen négyzethálós füzetlapot (ez lehet a teljes síkot lefedő háló vagy csupán egy negyedsíknak megfelelő). Ebben az esetben is ki kell jelölnünk egy kezdeti mezőt.

Nyilvánvaló, hogy a különböző modellekhez más látható részek tartoznak, ebből adódóan pedig eltérők a modellek átmeneti függvényei. Emellett egy természetesen felmerülő kérdés, hogy függ-e a modell megválasztásától egy adott számítási feladat megoldásának „nehézsége”? Ugyanazt a problémát megoldó, de eltérő modellek esetleg különbözhetnek futásidő, tárigény tekintetében? A válasz: igen, a modell megválasztása rendkívül fontos és meghatározó döntés. Illusztrációként lássunk egy példát:



5. ábra. Munkaszalag a síkon.

**Példa.** Az előző előadás során két algoritmust adtunk a *PALINDROM* nyelv eldöntésére. Az első algoritmus során az inputot átmásoltuk a munkaszalagra. Ennek oka az volt, hogy az inputszalagon található karaktereket a standard modellben nem lehet felülírni. Azonban, ha az egyszalagos modellt használjuk, vagyis írhatunk az inputszalagra, akkor a munkaszalagra nincs szükség. Az outputszalag döntési feladat lévén eleve felesleges. Tehát az első algoritmus működik az egyszalagos modellben is.

A második algoritmus fontos eleme volt, hogy kettő szemet használjunk, így ez az egyszalagos modellben **nem** valósítható meg.

Az algoritmusok időigényét is meghatároztuk:

1. algoritmus ( $T_1$  Turing-gép) négyzetes rendű:  $TIME(\omega, T_1) \leq |\omega|^2$ ,
2. algoritmus ( $T_2$  Turing-gép) lineáris futásidejű:  $TIME(\omega, T_2) \leq 3|\omega| + 3$ .

**Megjegyzés.** ◦ A két irányban végtelen munkaszalag modelljében a tárigényt úgy tudjuk megadni, hogy mérjük a szem kilengéseit. Ez nehezebb feladat a standard modellben dolgozó Turing-gép tárigényének meghatározásánál.

◦ A  $k$ -szalagos modell által megoldott számítási probléma megoldható a standard modellben is, habár az egy munkaszalagra történő redukálás sokszor nehéz technikai feladat.

Láttuk, hogy a második algoritmus, amely kisebb futásidejű, nem valósítható meg az egyszalagos modellen. A következő tétel szerint, ha egyszalagos modellt használó algoritmussal akarjuk eldönteni a *PALINDROM* nyelvet, akkor a négyzetes időigény lényegében nem javítható.

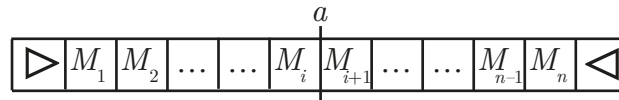
**1. Tétel.** *Ha  $T$  egy olyan Turing-gép, amely az egyszalagos modellben eldönti a *PALINDROM* nyelvet, akkor  $\forall n, \exists \omega \in \Sigma^n$ :*

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

valamilyen  $\alpha_T$  pozitív konstansra.

A tétel bizonyításához szükségünk lesz néhány új fogalomra.

**Definíció.** Az inputszalag két szomszédos mezőjének közös határát *ajtónak* nevezük. Ha az inputszalag mezőit úgy képzeljük el, mint végtelen egymásba nyíló szobasorozatot, akkor azt mondhatjuk, hogy a fej csak az ajtókon át tud közlekedni.



6. ábra. Az  $M_i$  és  $M_{i+1}$  mezőket elválasztó  $a$  ajtó.

Tekintsük egy  $T$  Turing-gép  $\omega$  inputon való futását. Ekkor egy

$$\kappa_0(\omega) \rightarrow \kappa_1 \rightarrow \kappa_2 \rightarrow \dots \rightarrow \kappa_\ell$$

konfigurációsorozatot kapunk, ahol

$$\ell := \min\{n \mid T \text{ állapota } \kappa_n\text{-ben } ELFOGAD \text{ vagy } ELVET\}$$

az időpont, amelyben eldöntjük, hogy  $\omega$  eleme-e a *PALINDROM* nyelvnek. Ez az időpont véges, hiszen a  $T$  Turing-gép eldönti a *PALINDROM* nyelvet. A *PALINDROM* nyelvet eldöntő Turing-gépek halmaza nem üres, hiszen eleme például a korábban említett  $T_1$  algoritmus.

Most vegyük azokat a  $\kappa_j, \kappa_{j+1}$  konfigurációkat, amelyekben az inputszem egyszer az  $M_i$ , másszor az  $M_{i+1}$  mezőt nézi. Jelölje  $s_j$  a mezőket elválasztó  $a$  ajtón történő átlépéskor a Turing-gép állapotát. Ezen  $s_j$  állapotok sorozatát  $\sigma(a, \omega)$  jelöli. Szemléletesen úgy képzelhetjük a  $\sigma(a, \omega)$  sorozatot, hogy az  $a$  ajtóban egy őr áll, amely a fej minden áthaladásakor feljegyezi annak állapotát.

Nyilvánvaló, hogy a  $\sigma(a, \omega)$  sorozatból kiolvasható az ajtón való áthaladás iránya, hiszen a fej balról érkezik, ezért a páratlan sorszámú állapotok a balról-jobbra ( $\rightarrow$ ) történő átlépéskor, míg a páros sorszámú állapotok a jobbról-balra ( $\leftarrow$ ) történő átlépéskor kerülnek feljegyzésre.

Jelölje  $\omega \overset{a}{|}$  az  $\omega$  input  $a$  ajtó előtti (tőle balra található) részét és  $\overset{a}{|}\omega$  az  $\omega$  input  $a$  ajtó utáni (tőle jobbra található) részét. Ha például az  $a$  ajtó a 6. ábrának megfelelően helyezkedik el, akkor  $\omega \overset{a}{|} = \omega_1 \dots \omega_i$  és  $\overset{a}{|}\omega = \omega_{i+1} \dots \omega_n$ . Természetesen a két rész kiadja a teljes  $\omega$  inputot:  $\omega = \left(\omega \overset{a}{|}\right) \left(\overset{a}{|}\omega\right)$ .

**Észrevétel.** Ha ismerjük az  $\omega \overset{a}{|}$  inputrészletet és a  $\sigma(a, \omega)$  állapotsorozatot, akkor meg tudjuk mondani, hogy a Turing-gép „hogyan működik”, amikor a szem az  $a$  ajtó előtti mezőket pásztázza. Azt nem tudhatjuk, hogy a szem meddig volt  $a$  jobb oldalán, de amint átlépi az ajtót (és amíg a bal oldalon marad) képesek vagyunk  $T$  futását leírni. Természetesen ugyanez elmondható az  $\overset{a}{|}\omega$  darabra is; ekkor az ajtó jobb oldalán ismerjük  $T$  lépéseit. Az alábbi animáció illusztrálja az inputszem mozgását és az ajtó elválasztó szerepét. A megfigyelt mezők pirosak, ha  $\omega \overset{a}{|}$  elemei és kékek, ha az  $\overset{a}{|}\omega$  részei. A  $\sigma(a, \omega)$  állapotsorozat a színváltások előtti állapotokból tevődik össze.

**2. Következmény.** Legyenek  $\omega, \omega' \in \Sigma^n$  tetszőleges inputok és a egy ajtó. Tegyük fel, hogy  $\sigma(a, \omega) = \sigma(a, \omega')$  és a  $T$  Turing-gép futásának eredménye megegyezik a két inputon. Ekkor az  $\tilde{\omega} = \left(\omega \overset{a}{|}\right) \left(\overset{a}{|}\omega'\right)$  inputon is ugyanazt számolja ki  $T$ .

**Megjegyzés.** Valójában az előbbinél többet is mondhatunk, hiszen az ajtón történő áthaladások számától függ, hogy melyik inputon ( $\omega$ -n vagy  $\omega'$ -n) fejeződik be a futás. Például, ha  $\sigma(a, \omega) = \sigma(a, \omega')$  páros hosszú állapotsorozat, akkor  $\tilde{\omega}$ -n  $T$  ugyanazt az eredményt adja, mintha  $\omega$ -n futtattuk volna. Ha az állapotsorozat páratlan hosszú, akkor  $T$  az  $\tilde{\omega}$  inputon az  $\omega'$ -n való számolás eredményével egyezik meg.

**Definíció.** Tegyük fel, hogy  $3 \mid n$ . Legyen

$$I_0 := \{\alpha 0^{\frac{n}{3}} \overleftarrow{\alpha} : \alpha \in \Sigma^{\frac{n}{3}}\} \subseteq \text{PALINDROM} \cap \Sigma^n.$$

Tehát  $I_0$  azon  $n$  hosszú palindrom szavakból áll, amelyekben egy  $n/3$  hosszú szó és fordítottja  $n/3$  nullát fog közre.

**Megjegyzés.** Mivel egy  $\alpha 0^{\frac{n}{3}} \overleftarrow{\alpha} \in I_0$  elemet egyértelműen meghatároz az  $\alpha$  szó, ezért  $|I_0| = |\Sigma|^{\frac{n}{3}} = |\{0, 1\}|^{\frac{n}{3}} = 2^{\frac{n}{3}}$ .

**3. Következmény.** Legyenek  $\omega, \omega' \in I_0$  különböző szavak és a egy középső ajtó (azaz valamelyik a középső  $n/3$  nullát elválasztó ajtó közül). Ekkor  $\sigma(a, \omega) \neq \sigma(a, \omega')$ .

**Bizonyítás.** Indirekt módon tegyük fel, hogy  $\sigma(a, \omega) = \sigma(a, \omega')$ . Ekkor az előző következmény alapján, ha mindkét inputon  $ELFOGAD$  állapottal áll le a Turing-gép, akkor az  $\tilde{\omega} = \left(\omega \overset{a}{|}\right) \left(\overset{a}{|}\omega'\right) = \alpha 0^{\frac{n}{3}} \overleftarrow{\alpha'}$  inputot is elfogadja, vagyis  $\tilde{\omega} \in I_0$ . Azonban  $\omega \neq \omega'$ , így  $\alpha \neq \alpha'$ , amiből  $\tilde{\omega} \notin I_0$  következik. Ez az ellentmondás igazolja az állítást. ■

**Észrevétel.** Ha feltételezzük, hogy  $|\Sigma| \geq 2$  és  $|S| \geq 3$ , akkor a  $t$ -nél rövidebb állapotsorozatok számára az

$$1 + |S| + \dots + |S|^{t-1} = \frac{|S|^t - 1}{|S| - 1} < |S|^t - 1 < |S|^t$$

felső becslés adható.

**4. Következmény.** Ha  $|I_0| = |\Sigma|^{\frac{n}{3}} \geq |S|^t$ , akkor  $\exists \omega \in I_0$ , hogy  $\sigma(a, \omega)$  hossza legalább  $t$ , ahol  $a$  egy középső ajtó.

**Bizonyítás.** Indirekt módon tegyük fel, hogy nincs ilyen  $\omega$ . Ekkor bármely  $\omega \in I_0$  és  $a$  középső ajtó esetén  $\sigma(a, \omega)$  hossza kisebb, mint  $t$ . Ez  $|I_0| > |S|^t$  állapot sorozatot jelent. Azonban a  $t$ -nél rövidebb állapot sorozatok számára adott becslés alapján, mint  $|I_0| < |S|^t$ . Ez ellentmondás, tehát az állításban szereplő  $\omega$  input létezik. ■

**Megjegyzés.** Az előbbi következményből az is leolvasható, hogy  $|I_0| = |\Sigma|^{\frac{n}{3}} \geq |S|^t$  esetén  $t \sim \beta_T \cdot n$ .

**5. Következmény.** Ha  $|I_0| \geq 2|S|^t$ , akkor létezik legalább  $|I_0|/2$  olyan szó  $I_0$ -ban, amelyre  $|\sigma(a, \omega)| \geq t$  és mindegyik különböző állapot sorozatot ad, amikor kiszámítjuk. Ezek halmazát jelölje  $I_1$ . Tehát

$$I_1 = \{\omega \in I_0 : |\sigma(a, \omega)| \geq t\} \subseteq I_0.$$

Az előbbi megjegyzés alapján  $t \sim \gamma_T \cdot n$ , alkalmas  $\gamma_T$  konstansra.

Ezek után térjünk rá az 1. Tétel bizonyítására

**Bizonyítás.** Azok az időpontok nyilván nem relevánsak, amikor a fej nem mozdul, ezért az alábbi alsó becslés adható

$$\sum_{\omega \in I_0} TIME(\omega, T) \geq \sum_{\omega \in I_0} \sum_{a \text{ középső ajtó}} |\{t : t \text{ időpillanatban a fej átlép } a\text{-n}\}|$$

az összegben megjelenő halmaz számossága  $\sigma(a, \omega)$  definíciója alapján  $|\sigma(a, \omega)|$ ,

$$= \sum_{\omega \in I_0} \sum_{a \text{ középső ajtó}} |\sigma(a, \omega)| = \sum_{a \text{ középső ajtó}} \sum_{\omega \in I_0} |\sigma(a, \omega)|$$

ahol kettős összegben a szokásos sorrendcsere történt. Az 5. Következmény alapján

$$\geq \sum_{a \text{ középső ajtó}} \sum_{\omega \in I_1} t = \sum_{a \text{ középső ajtó}} |I_1| \cdot t \geq \sum_{a \text{ középső ajtó}} \frac{|I_0|}{2} \cdot t$$

mivel a középső ajtók száma  $n/3$ , ezért ez

$$= \frac{n}{3} \cdot \frac{|I_0|}{2} \cdot t = \frac{\gamma_T}{6} \cdot n^2 \cdot |I_0|$$

ahonnan  $|I_0|$ -val való osztás után

$$\frac{1}{|I_0|} \sum_{\omega \in I_0} TIME(\omega, T) \geq \frac{\gamma_T}{6} \cdot n^2$$

adódik. Tehát azt kaptuk, hogy az átlagos futásidő négyzetesen függ az input hosszától. ■

**Megjegyzés.** A tételben szereplő konstans függ a Turing-gép megválasztásától. Ez nem „kerülhető ki”, vagyis minden  $\varepsilon > 0$  esetén megadható olyan Turing-gép, amely eldönti a *PALINDROM* nyelvet és futási ideje legalább  $\varepsilon n^2$ . Tehát a négyzetes jelleg megmarad. A futásidő ilyenén történő csökkentése az állapothalmaz bővítésével érhető el, melynek köszönhetően a gép hosszabb inputrészeket képes megjegyezni. Ennek részletes leírását az 1. Előadás 4. Példájában találhatjuk.

## 2. Bonyolultsági osztályok (nyelvekre)

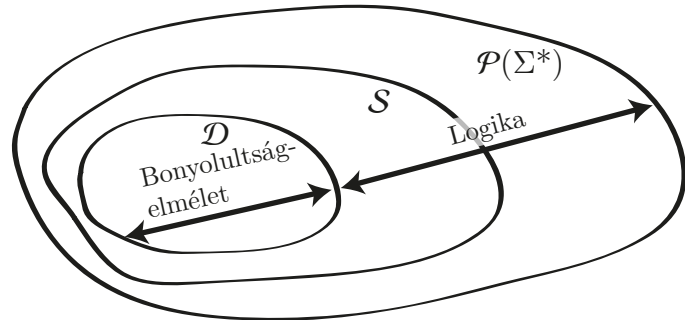
Korábban már definiáltuk a  $\Sigma$  ábécé feletti összes nyelv halmazát, mint a  $\mathcal{P}(\Sigma^*)$  hatványhalmazt, ahol  $\Sigma^* = \bigcup_{n \in \mathbb{N}} \Sigma^n$ . Az eldöntő Turing-gépek tárgyalásakor bevezettünk két nyelvosztályt:

- az eldönthető nyelvek  $\mathcal{D} := \{L \in \mathcal{P}(\Sigma^*) : \exists T \text{ Turing-gép, amely eldönti } L\text{-et}\}$ .  
A  $T$  Turing-gép *eldönti* az  $L$  nyelvet, ha az  $\omega \in L$  input esetén *ELFOGAD* állapottal,  $\omega \notin L$  input esetén pedig *ELVET* állapottal áll le.
- a felsorolható nyelvek  $\mathcal{S} := \{L \in \mathcal{P}(\Sigma^*) : \exists T \text{ Turing-gép, amely felsorolja } L\text{-et}\}$ .  
A  $T$  Turing-gép *felsorolja* az  $L$  nyelvet, ha az  $\omega \in L$  input esetén *ELFOGAD* állapottal leáll, míg  $\omega \notin L$  input esetén  $T$  futása végtelen.

Fontos megjegyezni, hogy a „létezik  $T$  Turing-gép” kifejezés alatt egy  $k$ -szalagos modell létezését értjük, azaz egy  $\Gamma$  munkaábécé,  $k \in \mathbb{N}$  munkaszalagszám,  $S$  állapot-halmaz és  $f$  átmeneti függvény által meghatározott  $k$ -szalagos Turing-gépet. Egy érdekesség, hogy ha más modellben definiáljuk az eldönthető és felsorolható nyelvek halmazát – például egyszalagos Turing-gépekre vagy kvantum Turing-gépekre – akkor is a fentiekkel megegyező  $\mathcal{D}$  és  $\mathcal{S}$  halmazokat nyerjük.

Az eldönthető és felsorolható nyelvekre beláttuk, hogy  $\mathcal{D} \subseteq \mathcal{S}$ . Megjegyezzük, hogy mindkét nyelvosztály számszáma megszámlálhatóan végtelen:  $|\mathcal{D}| = |\mathcal{S}| = \aleph_0$ , míg  $|\mathcal{P}(\Sigma^*)| = c$ .

A bonyolultságelmélet feladata a  $\mathcal{D}$  nyelvosztály vizsgálata.



7. ábra. Nyelvosztályok viszonya.

Most idézzük fel az időigény és tárigény definícióját!

**Definíció.** Egy  $T$  Turing-gép *időigénye* egy  $\omega$  inputon  $\ell := TIME(\omega, T)$ , ha futása  $\{\kappa_i\}_{i=0}^{\ell}$ , vagyis az  $\ell$ -edik konfigurációban kerül először *STOP* állapotba.

**Definíció.** Egy  $T$  Turing-gép *tárigénye* egy  $\omega$  inputon  $s := SPACE(\omega, T)$ , ha futása során a munkaszem/kéz alatti mező legnagyobb indexe  $s$ .

Mivel a Turing-gép legfeljebb annyi mezőt látogathat meg a munkaszalagon, amennyit mozog, ezért ezen jellemzők között fennáll a nyilvánvaló

$$SPACE(\omega, T) \leq TIME(\omega, T)$$

összefüggés. A futás idő- és tárigénye nyilván függ az input hosszától. A következő definícióval ezen függést tudjuk kifejezni.

**Definíció.** Legyen  $f: \mathbb{N} \rightarrow \mathbb{R}$  egy tetszőleges függvény. Azt mondjuk, hogy egy  $L$  nyelv eleme a  $TIME(f(n))$  halmaznak, ha létezik olyan  $T$  Turing-gép, amely

- eldönti  $L$ -et, és
- $TIME(\omega, T) \leq f(|\omega|)$ .

**Megjegyzés.** A korábban felidézett  $T_2$  algoritmus időigényére nyert eredményünket az előbbi definíció segítségével így írhatjuk:  $PALINDROM \in TIME(3n + 3)$ . Azonban fontos megjegyezni, hogy ha a „létezik  $T$  Turing-gép” kifejezésnek más jelentést tulajdonítunk (és nem a  $k$ -szalagos modellre gondolunk), akkor az előbbi tartalmazás nem biztos, hogy igaz.

Az imént bevezetett  $TIME(f(n))$  jelölés új nyelvosztályok definiálását teszi lehetővé.

**Definíció.**  $\circ$  *Polinomiális időben eldönthető nyelvek:*

$$\mathcal{P} := \bigcup_{a \in \mathbb{N}} TIME(an^a + a) = \bigcup_{p \in \mathbb{R}[x]} TIME(p(n)),$$

az  $a$  szorzó az 1 hosszúságú inputokon szükséges 2 lépés végett szükséges, így írhatnánk  $a$  helyett  $2$ -est is. A két unió, azért egyezik meg, mert bármely  $p \in \mathbb{R}[x]$  polinomhoz létezik olyan  $a \in \mathbb{N}$  természetes szám, melyre  $p(n) \leq an^a + a$ , ahol  $n \in \mathbb{N}$ .

$\circ$  *Exponenciális időben eldönthető nyelvek:*

$$\mathcal{EXP} := \bigcup_{a \in \mathbb{N}} TIME(2^{an^a + a}).$$

$\circ$  *Polinomiális tárral eldönthető nyelvek:*

$$\mathcal{PSPACE} := \bigcup_{a \in \mathbb{N}} SPACE(an^a + a).$$

$\circ$  *Exponenciális tárral eldönthető nyelvek:*

$$\mathcal{EXPSPACE} := \bigcup_{a \in \mathbb{N}} SPACE(2^{an^a + a}).$$

$\circ$  *Logaritmikus tárral eldönthető nyelvek:*

$$\mathcal{L} := \bigcup_{a \in \mathbb{N}} SPACE(a \log n),$$

itt megjegyezzük, hogy  $\log n$  az  $n = 0$  esetben nyilván nem értelmezhető, azonban ettől az elfajuló (nulla hosszúságú input) esettől eltekintünk. Továbbá  $\mathcal{L}$  miatt szükséges az inputszalag és a munkaszalag elkülönítése is, mert máskülönben nehéz lenne a logaritmikus tárat mérni/kimutatni.

**Megjegyzés.** Az iménti nyelvosztályok definíciója „robosztus”. Ez alatt azt értjük, hogy függetlenek a Turing-gép modelljének megválasztásától, azaz más-más modellel alapozva ugyanezen nyelvosztályokhoz jutunk.

Könnyen belátható, hogy a nyelvosztályok között az alábbi tartalmazások teljesülnek:

$$\mathcal{L} \subseteq \underbrace{\mathcal{PSPACE}}_{\cup \mathcal{P}} \subseteq \underbrace{\mathcal{EXPSPACE}}_{\cup \mathcal{EXP}} \subseteq \mathcal{D}$$

Természetesen további kapcsolatok is vannak, ezekről a későbbiek során esik szó. Nyelvosztályokból is jelentősen több létezik, mint amit itt megadtunk, a részletekért lásd a Qwiki oldalát: [http://qwiki.stanford.edu/index.php/Complexity\\_Zoo](http://qwiki.stanford.edu/index.php/Complexity_Zoo).



### 3. Univerzális Turing-gép

Az eldöntő Turing-gépek kódolhatóak. Ehhez rögzítsünk egy  $\Sigma_0$  ún. *univerzális ábécét*. Például legyen  $\Sigma_0 = \{0, 1\}$ . Rögzítsük Turing-gépeknek egy kódolását. Egy  $T = (\Sigma, \Gamma, S, k, f)$  Turing-gép esetén jelölje  $T$  kódját  $[T] \in \Sigma_0^* = \{0, 1\}^*$ . Az  $\omega \in \Sigma^*$  inputhoz rendeljük hozzá annak  $[\omega] \in \Sigma_0^*$  kódját. A  $([T], [\omega])$  párt jelölje  $[T, \omega]$ .

**6. Tétel.** *A fenti megállapodások esetén létezik olyan  $U$  Turing-gép, amely az inputszalagján a  $[T, \omega]$  kód alapján szimulálja a  $T$  Turing-gépnek az  $\omega$  inputon való futását, azaz*

$T$   $\omega$ -n ELFOGAD állapottal áll le  $\iff U$   $[T, \omega]$ -n ELFOGAD állapottal áll le

$T$   $\omega$ -n ELVET állapottal áll le  $\iff U$   $[T, \omega]$ -n ELVET állapottal áll le

$T$   $\omega$ -n  $\infty$  ideig fut (nem áll le)  $\iff U$   $[T, \omega]$ -n  $\infty$  ideig fut (nem áll le)

A tétel bizonyításához a Turing-gépek kódolását kellene definiálni, majd az  $U$  Turing-gép átmeneti függvényének konstrukcióját kellene megadni. Ezek hosszadalmas előkészületek, ezért a bizonyítástól eltekintünk.

Végül az eddigi ismereteket összegezve egy kisebb szótárat készíthetünk:

SZÓTÁR	
számítási probléma	$f: \Sigma^* \rightarrow \Sigma^*$ függvény
döntési probléma	$L \subseteq \Sigma^*$ nyelv
algoritmus	$T$ Turing-gép
programozási nyelv	$T \mapsto [T]$ Turing-gép kódolás
számítógép/interpreter	$U$ univerzális Turing-gép