

11. Veremfordító, függvényt számoló Turing-gép. Idő és tár

1. Legyen G a következő nyelvtan: $S \rightarrow aSbS \mid ab$. Ha egy $w \in L(G)$ szó bal-levezetésében alkalmazott szabályok sorszámait sorban felírjuk, akkor egy $\{1, 2\}^*$ -beli szót kapunk. A fordítás álljon ezekből a szó-sorszámsorozat párokból.

- (a) Adjon meg ehhez egy veremfordítót!
 (b) Adjon meg hozzá egy szintakszisvezérelt fordítási sémát!

Megoldás: Kezdjük a (b)-vel, abból a tanult eljárással egyszerű veremfordítót készíteni!

(b) A 2. nyelvtanban a sorszámot a változók elé kell írni, hogy helyes sorrendben kapjuk meg majd ezeket: $S \rightarrow aSbS; 1SS \mid ab; 2$

- (a) A verem alját jelölő karakter legyen S . A veremautomatának egyetlen, q állapota lesz. Az átmenetek:

$$\begin{aligned}(q, \varepsilon, S) &\mapsto \{(q, a1SbS, \varepsilon), (q, ab2, \varepsilon)\} \\ (q, a, a) &\mapsto \{(q, \varepsilon, \varepsilon)\} \\ (q, b, b) &\mapsto \{(q, \varepsilon, \varepsilon)\} \\ (q, \varepsilon, 1) &\mapsto \{(q, \varepsilon, 1)\} \\ (q, \varepsilon, 2) &\mapsto \{(q, \varepsilon, 2)\}\end{aligned}$$

Ez a tanultak szerint jó.

2. Az alábbi fordítási sémából készítsen veremfordítót:

$$S \rightarrow aAbS; 10AS1|b; 1 \quad A \rightarrow baA; A11|b; 1$$

Mi az $(ab)^3b(ab)^2bb$ szó fordítása?

Megoldás: A verem alját jelölő karakter legyen S . A veremautomatának egyetlen, q állapota lesz. Az átmenetek:

$$\begin{aligned}(q, \varepsilon, S) &\mapsto \{(q, a10AbS1, \varepsilon), (q, b1, \varepsilon)\} \\ (q, \varepsilon, A) &\mapsto \{(q, baA11, \varepsilon), (q, b1, \varepsilon)\} \\ (q, a, a) &\mapsto \{(q, \varepsilon, \varepsilon)\} \\ (q, b, b) &\mapsto \{(q, \varepsilon, \varepsilon)\} \\ (q, \varepsilon, 0) &\mapsto \{(q, \varepsilon, 0)\} \\ (q, \varepsilon, 1) &\mapsto \{(q, \varepsilon, 1)\}\end{aligned}$$

Csak egy nem elakadó számítási út van:

$(q, (ab)^3b(ab)^2bb, S) \mapsto (q, (ab)^3b(ab)^2bb, a10AbS1) \mapsto (q, b(ab)^2b(ab)^2bb, 10AbS1) \mapsto$
 2 kiíró lépés eredménye **10**, utána megmarad: $(q, b(ab)^2b(ab)^2bb, AbS1) \mapsto$
 $(q, b(ab)^2b(ab)^2bb, baA11bS1) \mapsto$ 2 olvasó lépés után: $(q, babb(ab)^2bb, A11bS1) \mapsto$
 $(q, babb(ab)^2bb, baA1111bS1) \mapsto$ 2 olvasó lépés után: $(q, bb(ab)^2bb, A1111bS1) \mapsto$
 $(q, bb(ab)^2bb, b11111bS1) \mapsto (q, b(ab)^2bb, 11111bS1) \mapsto$ 5 kiíró lépés eredménye **11111**, utána megmarad:
 $(q, b(ab)^2bb, bS1) \mapsto (q, (ab)^2bb, S1) \mapsto (q, (ab)^2bb, a10AbS11) \mapsto (q, babbb, 10AbS11) \mapsto$
 2 kiíró lépés eredménye **10**, utána megmarad: $(q, babbb, AbS11) \mapsto (q, babbb, baA11bS11) \mapsto$
 2 olvasó lépés után: $(q, bbb, A11bS11) \mapsto (q, bbb, b111bS11) \mapsto (q, bb, 111bS11) \mapsto$
 3 kiíró lépés eredménye **111**, utána megmarad: $(q, bb, bS11) \mapsto (q, b, S11) \mapsto (q, b, b111) \mapsto (q, \varepsilon, 111) \mapsto$
 és itt még kiír**111**

Vagy: a nem elakadó számítási útnak, az 1. nyelvtanbeli levezetésnek felel meg.

$$S \Rightarrow aAbS; 10AS1 \Rightarrow a \text{ } bA \text{ } bS; 10 \text{ } A11 \text{ } S1 \Rightarrow (aba \text{ } ba)A \text{ } bS; 10A11 \text{ } 1^2S1 \Rightarrow (ab)^3bS; 101^5S1 \Rightarrow (ab)^3baAbS; 101^5 \text{ } 10AS1 \text{ } 1 \Rightarrow (ab)^3ba \text{ } bA \text{ } bS; 101^60 \text{ } A11S11 \Rightarrow (ab)^3bababbS; 101^60111S11 \Rightarrow (ab)^3b(ab)^2bb; 101^601^6$$

Tehát a kimenet az 101^601^6 .

3. Legyen

$S \rightarrow$	$a0A \mid a0 \mid b0B \mid b0$		a	b	0	1
$A \rightarrow$	$a0A \mid a0 \mid b1B \mid b1$	h_1	a	b	ε	ε
$B \rightarrow$	$a1A \mid a1 \mid b0B \mid b0$	h_2	ε	ε	0	1

- (a) Mi a fordítás, aminek ez egy jellemző nyelvtana a h_1, h_2 leképezésekkel?
 (b) Megvalósítható-e véges fordítóval?
 (c) Adjon meg hozzá egy fordító automatát!

Megoldás:

(a) Értelmezési tartomány $= h_1(L)$. Ezt úgy kapjuk, hogy a nyelvtanra alkalmazzuk a leképezést, azaz kimarad a 0 és az 1. Látszik, hogy minden nem üres $\{a, b\}$ feletti szó ebből megkapható, az értelmezési tartomány az $(a + b)^*$

A fordításkor az a és b karakterek lecserélődnek 0-ra és 1-re. Az S szabályai szerint az első karakter helyett biztos 0 lesz. A továbbiakhoz azt kell észre venni, hogy attól függően szerepel az A vagy B változó a sorozatban, hogy az utolsó generált karakter a , vagy b volt. 1 pedig akkor keletkezik, ha változik a karakter (amikor A után B -re vagy vissza váltunk). Tehát a fordításban ott lesz 1, ahol a karakter különbözik az előző karaktertől.

(b) Igen. Egyrészt egyszerű átalakítani a nyelvtant regulárisrá, és akkor tétel szerint van hozzá véges fordító.

$$\begin{aligned} S &\rightarrow aX \mid bY \\ A &\rightarrow aX \mid bZ \\ B &\rightarrow aT \mid bY \\ X &\rightarrow 0A \mid 0 \\ Y &\rightarrow 0B \mid 0 \\ Z &\rightarrow 1B \mid 1 \\ T &\rightarrow 1A \mid 1 \end{aligned}$$

Közvetlenül megadni egy véges fordítót se nehéz (korábbi feladatsorban láttunk erre Mealy-automatát is).

4. Adjon meg egy jellemző nyelvtant (a szükséges leképezésekkel együtt) az 9/13 feladatbeli fordításhoz!

Megoldás: Az előző feladatbeli nyelvtant módosítsuk úgy, hogy a b -ről a -ra váltáskor 0 szerepeljen és ne 1,

a két függvény maradhat ugyanaz:

$$\begin{aligned} S &\rightarrow a0A \mid a0 \mid b0B \mid b0 \\ A &\rightarrow a0A \mid a0 \mid b1B \mid b1 \\ B &\rightarrow a0A \mid a0 \mid b0B \mid b0 \end{aligned}$$

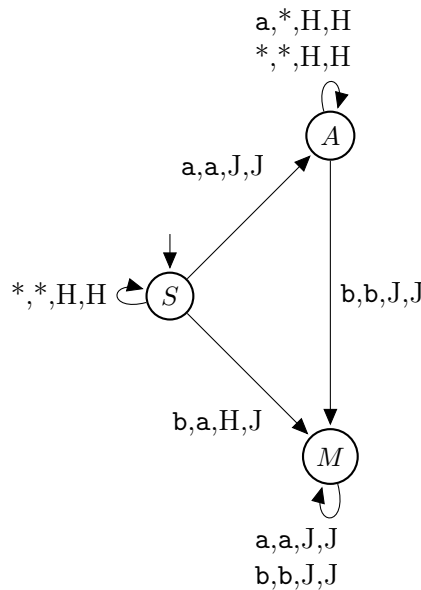
5. Legyen $\Sigma = \{a, b\}$ és az $f : \Sigma^* \rightarrow \Sigma^*$ függvény a következő:

$$f(x) = \begin{cases} x & \text{ha az } x \text{ szó } ab\text{-vel kezdődik} \\ ax & \text{ha az } x \text{ szó } b\text{-vel kezdődik} \\ \text{nincs definiálva} & \text{egyébként} \end{cases}$$

Adjon meg egy Turing-gépet, ami az f függvényt számolja ki!

Megoldás: 2 szalagos gépet adunk, ahol az 1. szalag nem írható, a 2. nem olvasható, ezért a mozgási irányok mellett elég az 1. szalag esetében csak az olvasott, a 2. szalagnál csak az írott karaktert feltüntetni.

A TG elég magától értetődő, arra kell vigyázni, hogy ahol a függvény nincs definiálva, ott a TG ne álljon meg. b -vel kezdődő szavaknál előbb kiír egy a -t, utána lemásolja a szót (M állapot). Ha a az első karakter, kiír egy a betűt, ami a helyes első karakter, ha egy b -vel tovább tud lépni az M állapotba, ahol már csak a maradékot másolja. Ha nem b -vel folytatódik a szó, akkor a végtelen ciklus miatt ez a már kiírt karakter nem számít kimenetnek.



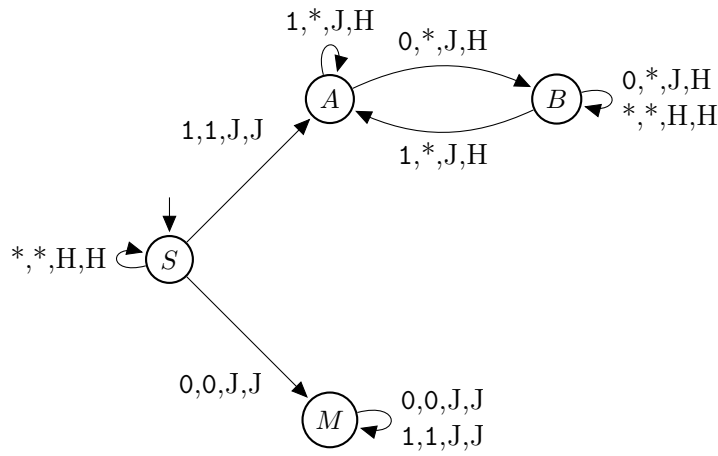
6. Legyen $\Sigma = \{0, 1\}$ és az $f : \Sigma^* \rightarrow \Sigma^*$ függvény a következő:

$$f(x) = \begin{cases} 1 & \text{ha az } x \text{ szó első és utolsó karaktere is } 1 \\ x & \text{ha az } x \text{ szó } 0\text{-val kezdődik} \\ \text{nincs definiálva} & \text{egyébként} \end{cases}$$

Adjon meg egy Turing-gépet, ami az f függvényt számolja ki!

Megoldás: A gondolat: 0-val 0-t kiírva átlépünk a további karakterek másolását végző M állapotba.

A kezdő 1 karakterkor kiírjuk az 1-t innen már csak arra kell vigyázni, hogy ha az utolsó nem 1, akkor a számítás ne álljon le (és ezen az ágon máskor ne írjunk ki semmit).



7. Igazolja, hogy az EXPTIME és a PSPACE nyelv osztály is zárt az

- (a) unióra
- (b) metszetre
- (c) konkatenálásra
- (d) tranzitív lezárra!

Megoldás: Nézzük az EXPTIME esetét! Legyen $L_1, L_2 \in \text{EXPTIME}$. Ekkor $L_1 \in \text{TIME}(2^{n^k})$ és $L_2 \in \text{TIME}(2^{n^\ell})$ valamilyen $k, \ell \geq 1$ pozitív egészekre. Azaz van hozzájuk M_1 , illetve M_2 DTG, és $c_1, c_2 > 0$ konstans, amikre teljesül, hogy $T_{M_1}(n) \leq c_1 \cdot 2^{n^k}$ és $T_{M_2}(n) \leq c_2 \cdot 2^{n^\ell}$, ha n elég nagy.

(a) Az M gépnek legyen 3 szalagja. Először M átmásolja a bemenetet a 2. és a 3. szalagra (egy-egy, a szalag elejét jelző karakter felírása után), majd visszamegy ennek a két szalagnak az elejére. Ez után M_1 -et futtatja a 2. szalagon. Ha ez elfogad, akkor M álljon meg elfogadó állapotban. Különben átlép M_2 kezdőállapotába, és ezt a gépet futtatja a 3. szalagon. Ha M_2 elfogad, akkor M is fogadjon el, ha M_2 elutasítva áll meg, akkor tegyen M is így.

Ez az M azokat a szavakat fogja elfogadni, amelyeket legalább az egyik gép elfogadott, azaz $L(M) = L(M_1) \cup L(M_2) = L_1 \cup L_2$.

Az M lépésszáma $T_M(n) \leq (n+1) + (n+1) + T_{M_1}(n) + T_{M_2}(n) \leq 2n+2 + c_1 \cdot 2^{n^k} + c_2 \cdot 2^{n^\ell}$, ahol az első két tag a másolás és a szalagok elejére visszamenés ideje. Legyen $c = \max\{c_1, c_2\}$ és $m = \max\{k, \ell\}$. Ekkor $T_M(n) \leq 2n+2 + 2c \cdot 2^{n^m} \leq 3c \cdot 2^{n^m}$, tehát $L_1 \cup L_2 \in \text{TIME}(2^{n^m}) \subset \text{EXPTIME}$.

(b) Az előző konstrukción annyit kell csak változtatni, hogy ha M_1 elutasít, akkor M is tegyen így, különben futtassa M_2 -t és ez esetben akkor fogadjon el, ha M_2 is elfogad.

Az időigényre alkalmazhatjuk ugyanazt a becslést, mint előbb, $L_1 \cap L_2 \in \text{TIME}(2^{n^m}) \subset \text{EXPTIME}$.

(c) A 2. és 3. szalagot most is az M_1 és M_2 szalagjaként használjuk. Ezekre M átmásolja az aktuális kezdőszeletet, illetve a végét a szónak. Most is egymás után futtatja a két gépet a megfelelő szalagokon. Ha mindkettő elfogadó állapotban áll meg, akkor M álljon meg elfogadóban, különben az eggyel hosszabb kezdőszeletet másolja a 2. szalagra, a maradékot a 3-ra, stb.

Ehhez persze az is kell, hogy valahogy M tudja, melyik szétvágás kipróbálásánál tart. A legegyszerűbb talán, ha van még 2 szalagja: kezdetben a 4. szalagra egy, az elejét jelző karakter után átmásolja a teljes bemenetet (hogy nyugodtan lehessen rajta visszafelé is menni). Az 5. szalagon mindig annyi 1 lesz, amilyen hosszú kezdőszeletnél tartunk. Azaz kezdetben az 5. szalag üres, és ezért a 2. szalagra 0 hosszú kezdőszeletet másol, a 3-ra meg az egész szót és visszamegy a 2-5. szalagok elejére. Amikor egy felosztás tesztelése elfogadás nélkül véget ért, akkor a 2. szalagra az 5. első üres mezőjének eléréséig másolja a karaktereket a 4. szalagról, az 5-re kiír még egy 1-et, és a 3-ra folytatja a másolást, végül visszaáll a 2-5. szalagok elejére.

Ennek az M gépnek az időigénye a korábbiakhoz képest úgy változik, hogy lényegében $(n+1)$ -szer futtatjuk az M_i gépeket. A másolások, előre-hátra mozgások nem befolyásolják a lépésszám nagyságrendjét: $T_M(n) \leq (n+1) \cdot 3c \cdot 2^{n^m}$ és ezért $L_1 L_2 \in \text{EXPTIME}$.

(d) Most már csak vázlatosan: az M gépnek a bemenet összes lehetséges felosztását ki kell próbálni, hogy valamelyikre teljesül-e, hogy a szó mindegyik darabja L_1 -ben van. Ezt az előzőhöz hasonlóan meg lehet tenni. Mivel bármely két karakter között lehet a felosztásnak egy vágása, ezért 2^{n-1} lehetőség van-

Ha ezeket mind egymás után kipróbáljuk, akkor az M_1 gépet kevesebb, mint $2^{n-1} \cdot n$ részszoán futtatjuk. Az adminisztráció költsége (éppen hol tartunk, a megfelelő részszoavak másolása) minden részszoánál lineáris, ezért $T_M(n) \leq n \cdot 2^n \cdot c_1 \cdot 2^{n^k}$, azaz $L_1^* \in \text{EXPTIME}$.

A PSPACE esete: Ehhez elég azt észrevenni, hogy az előző módszereink tárigénye $S_M(n) \leq S_{M_1}(n) + S_{M_2}(n) + c'n$. Ha M_1 és M_2 is polinom táorkorlátos, akkor ezért $S_M(n)$ is az lesz.

8. Legyen $L = \{xxx : x \in \Sigma^*\}$. Igazolja, hogy $L \in \text{SPACE}(\log n)$!

Megoldás: A logaritmikus tár nem teszi lehetővé azt az egyszerű megoldást, amikor lemásoljuk vagy felülírjuk a bemenetet. Mivel a tár kevesebb, mint n , ezért biztos kell a bemenetin kívül még szalag, és ennek a használatát korlátozzuk. Legyen mondjuk 2 további szalag. A 2-on a bemenetet olvasva számláljuk a karaktereket, azaz, amikor a bemenet végére érünk, akkor a 2. szalagon a bemenet hossza (n) álljon bináris alakban. Sőt még egyszerűbb ha ennek harmadát írjuk fel (minden 3. lépésnél adunk hozzá egyet). Ha a bemenet hossza nem osztható 3-mal, akkor el kell utasítani. Utána a 3. szalagra egymás után az $i = 1, 2, \dots, n/3$ számokat írjuk (bináris alakban!), és mindegyiknél ellenőrizzük, hogy a bemenet i -edik, $(i+n/3)$ -edik és $(i+2n/3)$ -edik karaktere megegyezik-e. Ha valamelyik hármas nem azonos, akkor a TG álljon meg elutasítva, különben, ha a végére ért, akkor fogadjon el.

9. Bizonyítsa be, hogy az $L = \{(G, k) : \text{a } G \text{ gráfban nincs } k \text{ független pont}\}$ nyelvre $L \in \text{PSPACE}$ és $L \in \text{EXPTIME}$ is teljesül!

Megoldás: Gondolkozzunk algoritmusban. Az kell, hogy ez polinomiális méretű tárat, illetve exponenciális időt használjon. Vegyük észre, hogy a triviális algoritmus jó: felsoroljuk a részhalmazokat, és mindegyikre ellenőrizzük, hogy független-e. Egy m csúcsú gráf esetén, ha ez pl. a szomszédossági mátrixával adott, akkor a bemenet hossza $n = m^2$. A lehetséges részhalmazok száma $< 2^m$ és minden részhalmaz ellenőrzése $\leq m^2$ lépés. Egy részhalmaz előállítása az előzőből megy a pontszámban lineáris, $O(m)$ időben, ezért az eljárás $O(m^2 \cdot 2^m)$ időkorlátos, ami miatt $L \in \text{TIME}(m^2 \cdot 2^m) \subset \text{TIME}(2^n) \subset \text{EXPTIME}$.

A tárigénynél csak arra kell figyelni, hogy ne akarjuk a lehetséges részhalmazok mindegyikét egyszerre eltárolni, hanem amikor előállítunk egyet, akkor történjen meg ennek ellenőrzése, és azután a részhalmazt írjuk felül a következővel. Így akkor csak az aktuális részhalmazt és azt kell tárolni, éppen melyik pontpárjánál tartunk az ellenőrzésében, ami csak lineáris tárat igényel, ezért $L \in \text{PSPACE}$ is teljesül.

10. Igazolja, hogy az $L \in \text{SPACE}(2^{121 \log n})$ feltevésből $L \in \text{P}$ is következik!

Megoldás: Használjuk a tár-idő tételt: ez alapján $L \in \text{TIME}(c^{a \cdot 2^{121 \log n}})$ teljesül valamilyen, az L -től függő c konstanssal. Mivel $c^{a \cdot 2^{121 \log n}} = n^{a \cdot 2^{121 \log c}}$, ezért van olyan k konstans, hogy $L \in \text{TIME}(n^k) \subseteq \text{P}$.

11. Rekurzív-e az $L_1 \cap L_2$ nyelv, ha $L_1 \in \text{TIME}(n)$ és $L_2 \in \text{SPACE}(2^n)$?

Megoldás: Mivel L_1 -hez a feltétel szerint van időkorlátos Turing-gép, ezért $L_1 \in \text{R}$. A tár-idő tétel miatt L_2 -höz is van időkorlátos Turing-gép, ezért $L_2 \in \text{R}$. Tudjuk, hogy R zárt a metszetre, ezért $L_1 \cap L_2 \in \text{R}$.

(Még rövidebben: volt, hogy $\text{TIME}(n) \subseteq \text{R}$, a tár-idő tétel következményeként volt hogy $\text{SPACE}(2^n) \subseteq \text{R}$, és hogy R zárt a metszetre, tehát a válasz igen.)