11. Idő és tár. Moore-automaták

1. Igazolja, hogy a P nyelvosztály zárt az unióra, metszetre, komplementerre, különbségre, konkatenálásra!

   Megoldás: Ha van két polinom időkorlátos determinisztikus Turing-gép, akkor ezekez a szokásos módon, egymás után vagy párhuzamosan futtatva kaphatunk az uniót, a metszetet, a különbséget elfogadó Turing-gépet. Könnyen látszik, hogy ezek is polinom időkorlátosak (a bemenetnek a szalagokra átmásolása, a szalagok elejére visszaállás lineáris idő, ehhez jön még a két időkorlát összege vagy maximuma, a lépésszám ezek összege, ami továbbra is felülről becsülhető egy polinommal.

   Mivel a különbségre igaz az álljtás, és a teljes nyelv (Σ^*) is P-ben van, ezért a komplementerre is zárt az osztály.

   Konkatenálás: egy n hosszú a bemeneti szót $n+1$ féleképpen tudunk kettévágni (az is jó, ha az egyik darab az üres szó). Ha van egy $t_1(n)$ és egy $t_2(n)$ időkorlátos M_1 és M_2 Turing-gép az L_1 és L_2 nyelvre, akkor a különböző szétvágások első felén az M_1-et, a második felén az M_2-t futtatva, a lépésszám legfeljebb $(t_1(0)+t_2(n))+(t_1(1)+t_2(n))+\ldots+(t_1(n)+t_2(0))$.

   Mindkét gép polinom időkorlátos, azaz $t_i(k)\le a_i\cdot k^{c_i}$, ezért az előző összeg legfeljebb $(n+1)\cdot(a_1\cdot n^{d_1}+a_2\cdot n^{d_2})\le (a_1+a_2)\cdot(n+1)\cdot n^{\max\{d_1,d_2\}}\in O(n^{1+\max\{d_1,d_2\}})$, ami n-nek polinomja.

   A két gép futtatásán kívül itt is szükség van arra, hogy a bemenet megfelelő darabját mindig odamásoljuk a megfelelő szalagra, de ez is csak polinom számú lépést jelent.

2. Igazolja, hogy az NP nyelvosztály zárt az unióra, metszetre, konkatenálásra!

   Megoldás: Az unióra, metszetre itt is hasonlóan járhatunk el, mint a determinisztikus esetben, de a konkatenálásra van egy egyszerűbb mód is, ha kihasználjuk a nemdeterminisztikus lehetőséget: ekkor nem kell minden lehetséges szétvágást kipróbálni, hanem nemdeterminisztikusan ágazzon el a számítás, pl. hogy a 0 hosszú kezdőszeletet vizsgálja vagy nem azt, ha nem, akkor elágazik, aszerint, hogy az 1 hosszút nézi vagy nem , stb. az ág végén csak azt az egy felosztást ellenőrzi. így minden számítási ágon csak egyszer kell futtatni az eredeti gépeket, a determinisztikus lépésszámbecslés-beli összeg helyett itt elég az $a_1\cdot n^{d_1}+a_2\cdot n^{d_2}$, meg még az elágazások és a másolás igényelte $O(n)$ lépés, összeségében, ez is polinom.

   Megjegyzés: Az nem ismert, hogy az NP zárt-e a komplementerre. A zártságból NP=co NP következne, ami egy ismert nyitott kérdés.

3. Igazolja, hogy az EXPTIME és a PSPACE nyelvosztály is zárt az
   (a) unióra
   (b) metszetre
   (c) komplementerre
   (d) konkatenálásra
   (e) tranzitív lezártra!

   Megoldás: Nézzük az EXPTIME esetét! Legyen $L_1,L_2\in$ EXPTIME. Ekkor $L_1\in \mathrm{TIME}(2^{n^k})$ és $L_2\in \mathrm{TIME}(2^{n^\ell})$ valamilyen $k,\ell\ge 1$ pozitív egészekre. Azaz van hozzájuk M_1, illetve M_2 DTG,és $c_1,c_2>0$ konstans, amikre teljesül, hogy $T_{M_1}(n)\le c_1\cdot 2^{n^k}$ és $T_{M_2}(n)\le c_2\cdot 2^{n^\ell}$, 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 $T_M(n)\le(n+1)+(n+1)+T_{M_1}(n)+T_{M_2}(n)\le 2n+2+c_1\cdot 2^{n^k}+c_2\cdot 2^{n^\ell}$, 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 $T_M(n)\le 2n+2+2c\cdot 2^{n^m}\le 3c\cdot 2^{n^m}$, tehát $L_1\cup L_2\in \mathrm{TIME}(2^{n^m})\subset$ 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, $L_1 \cap L_2 \in \mathrm{TIME}(2^{n^m}) \subset \mathrm{EXPTIME}$.

(c) Legyen M ugyanaz, mint M_1, csak az elfogadó állapotai legyenek M_1 nem elfogadó állapotai. Ekkor $L(M) = \overline{L(M_1)}$ és nyilván $T_M(n) = T_{M_1}(n) \in \mathrm{EXPTIME}$.

(d) 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. és 5. szalag elejére is rak egy-egy speciális karaktert (hogy nyugodtan lehessen rajtuk visszafelé is menni). Ez a karakter után a 4. szalagra átmásolja a teljes bemenetet, az 5-en tartja nyilván, melyik szétvágást vizsgálja, ezen 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 $L_1 L_2 \in \mathrm{EXPTIME}$.

(e) 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 $2^{n-1} \cdot n$ részszón futtatjuk. Az adminisztráció költsége (éppen hol tartunk, a megfelelő részszavak másolása) minden részszónál lineáris, ezért $T_M(n) \leq n \cdot 2^n \cdot c_1 \cdot 2^{n^k}$, azaz $L_1^* \in \mathrm{EXPTIME}$.

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 polinom tárkorlátos, akkor $S_M(n)$ is az lesz.

4. Igazolja, hogy az $L \in \mathrm{SPACE}(2023 \log n)$ feltevésből $L \in \mathrm{P}$ is következik!

   Megoldás: Használjuk a tár-idő tételt: ez alapján $L \in \mathrm{TIME}(c^{a \cdot 2121 \log n})$ teljesül valamilyen, az L-től függő c konstanssal. Mivel $c^{a \cdot 2121 \log n} = n^{a \cdot 2121 \log c}$, ezért van olyan k konstans, hogy $L \in \mathrm{TIME}(n^k) \subseteq \mathrm{P}$.

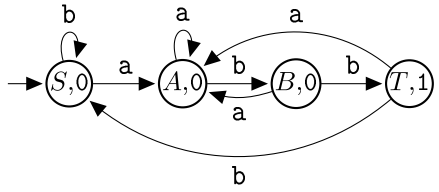
5. Rekurzív-e az $L_1 \cap L_2$ nyelv, ha $L_1 \in \mathrm{TIME}(n)$ és $L_2 \in \mathrm{SPACE}(2^n)$?

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

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

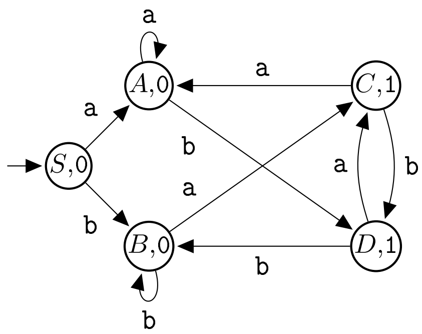
6. Adjon meg egy olyan Moore-automatát, aminek bemeneti ábécéje $\Sigma = \{\mathtt{a}, \mathtt{b}\}$, kimeneti ábécéje $\Gamma = \{\mathtt{0}, \mathtt{1}\}$. Az automata általában `0`-t ír ki, de amikor az `abb` részszó megjelent a bemenetben, akkor `1`-t (azaz olyan `b`-nél, ami előtt `ab` volt).

   Megoldás: 4 állapotot használunk, amelyek azt jelzik, hol tartunk az `abb` szóban.



7. Adjon meg egy Moore-automatát, ami minden $w \in \{\mathtt{a}, \mathtt{b}\}^*$ szóból egy olyan 0-1 sorozatot készít, amiben mindig ott van 1, ahol az aktuális karakter különbözik az előzőtől!

Megoldás: A kezdőállapoton kívül 4 állapot van, melyek azt mutatják, hogy mi az utolsó karakter, és hogy ez ugyanaz-e, mint az előző (A és C, illetve B és D).



. . . .