8. R, RE

1. Legyen $L = \{w : \exists M_w$ és ennek a szalagon levő feje soha nem megy az 5. mezőnél távolabbra$\}$. Igaz-e, hogy $L \in \mathrm{R}$?

   Megoldás: Az állítás igaz. Ennek belátásához egy olyan M TG-t kell vázolni, ami mindig megáll és pont az ilyen Turing-gépek kódjait fogadja el. Felületesen, ehhez azt kell ellenőriznie, hogy mindegy, hogy az M_w gépet milyen szón indítjuk el, az soha nem lép túl az 5. mezőn. A két probléma: ehhez első ránézésre az M_w gépet végtelen sok bemeneten kell ellenőrizni és ezek bármelyikén akár végtelen ideig is futhat, az M-nek meg véges időben meg kell állnia.

   Az első probléma egyszerű: ha M_w nem lép túl az 5. mezőn, akkor a bemenetének csak az első 5 karaktere számíthat – tehát csak véges sok lehetőséget kell ellenőrizni. De még ezeken is futhat végtelen ideig. Ezt kicsit trükkösebb észrevenni: Ha csak 5 karaktert írhat/olvashat a szalagon akkor csak véges sok különböző konfigurációban lehet (állapot, szalagtartalom, fej helyzete). Ha egy ilyen ismétlődik, akkor tudhatjuk, hogy M_w végtelen ciklusba került.

   Tehát M működése egy w bemeneten: ellenőrzi hogy w egy TG kódja-e. Ha nem, akkor M megáll elutasító állapotban. Különben pedig a 2. szalagon sorban generálja a legfeljebb 5 hosszú bemeneti szavait M_w-nek. Az aktuális szón lépésenként futtatja az M_w gépet. Ha M_w átlépne a 6. mezőre, akkor M megáll, elutasít. Különben M feljegyzi (egy további szalagra) az M_w pillanatnyi konfigurációját és ellenőrzi, hogy a korábbi feljegyzések között nem fordult-e már elő ugyanilyen. Ha nem, akkor folytatja és elvégzi M_w következő lépését. Ha viszont már volt ilyen, akkor M tovább megy a következő szóra. Akkor is tovább megy a következő szóra, ha M_w egy adott konfigurációban elakad.

   Ha M elutasítás nélkül végére ért az összes, legfeljebb 5 hosszú bemenetnek, akkor álljon meg elfogadó állapotban.

   Vegyük észre, hogy ha M_w valamilyen bemeneten átlép a 6. mezőre, akkor ez az M elutasítja a w bemenetet, azaz $w \notin L(M)$. Másrészt, ha M_w nem ilyen, akkor menet közben M nem áll meg elutasító állapotban. Viszont ilyenkor az eljárás véges (véges sok szón véges sok lépésig használja M_w-t, és a szükséges "adminisztráció" is véges idejű). A végén pedig el fogja fogadni a w szót, $w \in L(M)$.

   Az M valóban mindig megáll, mert ha w egy olyan TG kódja, ami tovább lép, akkor M az adott szónál megáll (elutasítva). Ha meg egy nem ilyen TG kódja, akkor a már leírt okok miatt a lehetőségeket kimerítve megáll.

2. Az L nyelv álljon azokból a $w\#s$ alakú szavakból, amelyeknél w egy Turing-gép kódja és a w kódú Turing-gép az s bemeneten 100 lépésen belül megáll. Igaz-e, hogy
   (a) $L \in \mathrm{R}$?
   (b) $L \in \mathrm{RE}$?
   (c) $L \in \mathrm{co\,RE}$?

   Megoldás: Mindhárom állítás igaz.

   Definiáljuk a következő M TG-t:

   - M ellenőrzi, hogy a bemenet $w\#s$ alakú. Ha nem ilyen, akkor M álljon meg elutasítva.
   - M ellenőrzi, hogy w egy TG -e. Ha nem, akkor M álljon meg elutasítva.
   - M futtatja az M_w gépet az s bemeneten 101 lépésig. Ha közben M_w elakad, akkor M álljon meg elfogadó állapotban, különben meg nem elfogadó állapotban. (Csak amikor a 101. lépésre térne, akkor derül ki, hogy a 100. lépés után már nem tud lépni.)

   Ez az M TG mindig megáll, hiszen M_w mozgását legfeljebb 101 lépésig követi, és akkor fogad el, ha ez alatt M_w megállt, azaz $L(M) = L$.

   Ebből tehát látjuk, hogy $L \in \mathrm{R}$, amiből $L \in \mathrm{RE}$ és $L \in \mathrm{co\,RE}$ is következik.

3. Az L nyelv álljon az olyan Turing-gépek w kódjából, hogy a w kódú Turing-gép minden bemeneten 100 lépésen belül megáll. Igazolja, hogy $L \in \mathrm{co\,RE}$.

Megoldás: $L \in \mathrm{co\,RE}$ pontosan akkor teljesül, ha $\overline{L} \in \mathrm{RE}$, azaz ha van TG, ami $\overline{L}$-et ismeri fel. $\overline{L}$ az olyan w-kből áll, amelyek vagy nem TG kódok vagy van hozzájuk olyan s bemenet, amin M_w nem áll meg 100 lépésen belül. Vázolunk ehhez egy TG-t, ami lényegében egy ilyen s szót keres.

Az M TG egy w bemeneten

- ellenőrzi, hogy w egy TG kódja-e. Ha nem, akkor álljon meg elfogadó állapotban.
- generálja egymás után a szavakat és sorban mindegyiken futtatja az M_w gépet 101 lépésig. Ha M_w eközben elakad, akkor veszi a következő szót. Ha viszont nem akad el, akkor M álljon meg elfogadó állapotban.

Ez az M akkor fog egy TG kódot elfogadni, amikor talál hozzá egy bemenetet, amin M_w nem áll meg 100 lépésben – és pont ezt kéri a feladat.

Megjegyzés: elég lenne csak a legfeljebb 100 hosszú bemeneteken futtatni M_w-t, hiszen ha ezek mindegyikén megáll 100 lépésben, akkor egy hosszabb bemenetből sem fog 100-nál több karaktert elolvasni.

4. Az L nyelv álljon az olyan Turing-gépek w kódjából, hogy a w kódú Turing-gép egyetlen bemeneten sem áll meg. Igaz-e, hogy $L \in \mathrm{co\,RE}$?

Megoldás: Az állítás igaz.

$L \in \mathrm{co\,RE}$ pontosan akkor teljesül, ha $\overline{L} \in \mathrm{RE}$, azaz ha van TG, ami $\overline{L}$-et ismeri fel. $\overline{L}$ az olyan w-kből áll, amelyek vagy nem TG kódok vagy van hozzájuk olyan s bemenet, amin M_w megáll. Vázolunk ehhez egy TG-t, ami lényegében egy megfelelő s bemenetet keres.

Az M TG egy w bemeneten

- ellenőrzi, hogy w egy TG kódja-e. Ha nem, akkor álljon meg elfogadó állapotban.
- A szokott módon sorban generálja az (s,i) szó–lépésszámkorlát párokat és mindegyiknél futtatja az M_w gépet az s bemeneten i lépésig. Ha M_w eközben elakad, akkor M álljon meg elfogadó állapotban. Ha viszont nem akad el, akkor vegye a következő párt.

Ez az M akkor fog egy TG kódot elfogadni, amikor eljut egy (s,i) párhoz, amire $M_w(s)$ megáll i lépésben. Ha ilyen nincs, akkor M nem áll meg, Ez utóbbi pontosan akkor történik, ha M_w nem áll meg egyetlen szón sem. Tehát $L(M) = \overline{L}$ és ezért $L \in \mathrm{co\,RE}$

Megjegyzés: itt már valóban végtelen sok s között kell keresni egy megfelelőt, nem lehet egy véges halmazra megszorítani. Ki kell használnunk a lehetőséget, hogy M-nek nem kell megállnia.

5. Rekurzív-e az $L = \{w : w \text{ Turing-gép kód és } L(M_w) = L_u\}$ nyelv?

Megoldás: Az, hogy egy nyelv az L_u, egy nyelvi tulajdonság, $T = \{L : L = L_u\}$. Nem triviális, hiszen $L_1 = L_u \in \mathrm{RE}$ rendelkezik a tulajdonsággal, míg mondjuk $L_2 = \emptyset \in \mathrm{RE}$ nem. Ezért a Rice-tételből következik, hogy $L = L_T \notin \mathrm{R}$.

6. Rekurzív-e az $L = \{w : w \text{ Turing-gép kód és } |L(M_w)| = 5\}$ nyelv?

Megoldás: A T nyelvi tulajdonság itt az, hogy a nyelvnek pontosan 5 szava van, $T = \{L : |L| = 5\}$. Ez nemtriviális nyelvi tulajdonság, hiszen pl. $L_1 = \{0^k : 0 \le k \le 4\} \in T$ egy pozitív példa, és mivel L_1 véges, ezért reguláris, és így persze $L_1 \in \mathrm{RE}$. Negatív példának most is jó az $L_2 = \emptyset$ választás. A Rice-tételből következik, hogy $L = L_T \notin \mathrm{R}$.

7. Álljon az L nyelv azokból a w szavakból, melyekre a w kódú Turing-gép létezik és az általa elfogadott nyelvben van legalább egy csupa `0`-ból álló szó. Igaz-e, hogy ez a nyelv rekurzívan felsorolható? Igaz-e, hogy ez a nyelv rekurzív?

Megoldás: Ahhoz, hogy rekurzívan felsorolható, egy M TG-t kell vázolni, ami ezt a nyelvet fogadja el, azaz M lényegében adott w-hez egy olyan $\mathtt{0}^k$ szót keres, amire $\mathtt{0}^k \in L(M_w)$. M egy w bemeneten:

- ellenőrzi, hogy w TG kód-e. Ha nem, akkor M álljon meg elutasítva.
- A (k,i) párokon sorba menve M futtatja M_w-t a $\mathtt{0}^k$ szón i lépésig. Ha ez alatt M_w elfogad, akkor M álljon meg elfogadó állapotban, különben menjen tovább a következő párra.

Ha van $\mathtt{0}^k \in L(M_w)$, akkor M_w ezt valahány lépésben elfogadja, és ha i legalább ennyi, akkor M megáll elfogadó állapotban. Ha viszont nincs csupa 0 szó az M_w nyelvében, akkor M keresése nem fog megállni, és így persze nem is fogadja el w-t. Tehát valóban $L(M) = L$, azaz $L \in \mathrm{RE}$.

Annak belátásához, hogy L nem rekurzív, vegyük észre, hogy az, hogy egy nyelv tartalmaz csupa 0-ból álló szót egy nyelvi tulajdonság, $T = \{L : \exists k, \mathtt{0}^k \in L\}$. Nem triviális, mert pl. $L_1 = \{\mathtt{0},\mathtt{1}\}^* \in \mathrm{RE}$ egy pozitív, míg $L_2 = \emptyset \in \mathrm{RE}$ egy negatív példa. Ezért $L = L_T$ a Rice-tétel miatt nem rekurzív.

8. Álljon az L nyelv az olyan Turing-gépek kódjaiból, amelyek csak páros hosszú szavakat fogadnak el. Igaz-e, hogy L

   a) rekurzív?

   b) rekurzívan felsorolható?

   c) co RE-ben van?

   Megoldás:

   a) Nem igaz. Legyen $T = \{L : w \in L \implies |w| \text{ páros } \}$. Ez a nyelvi tulajdonság nem triviális, hiszen van rá pozitív példa, pl. $L_1 = \{\mathtt{00}\}$, ami valóban RE-ben is van, hiszen egy véges nyelv reguláris, és ezért rekurzív, ami miatt RE-ben is benne van. (Közvetlenül sem nehéz megadni hozzá egy TG-t.) Negatív példának jó pl. az $L_2 = \{\mathtt{0}\} \in \mathrm{RE}$ hasonló okok miatt.

   Megmutatjuk, hogy c) igaz, és ebből már következik, hogy b) nem igaz, hiszen ha egy nyelv RE-ben és co RE-ben is benne van, akkor rekurzív is, ami az a) szerint nem teljesül.

   c) $L \in \mathrm{co\,RE}$ pontosan akkor, ha $\overline{L} \in \mathrm{RE}$, azaz van olyan TG, ami az $\overline{L}$ nyelvet ismeri fel. $\overline{L}$ elemei a nem TG kódok és azok a w TG kódok, melyekre M_w elfogad legalább egy páratlan hosszú szót is. Az M gép lényegében egy ilyen páratlan hosszú szót keres. M egy w bemeneten

   - ellenőrzi, hogy w TG kód-e. Ha nem, akkor M álljon meg elfogadó állapotban.
   - A szokott módon sorban előállítja az (s,i) párokat, ahol s páratlan hosszú szó, $i = 1, 2, \ldots$. Az M_w gépet az s szón i lépésig futtatja. Ha ez alatt M_w elfogad, akkor M álljon meg elfogadó állapotban, különben vegye a következő párt.

   Ha M_w elfogad egy páratlan hosszú s szót, akkor, amint M az (s,i) párhoz ér egy megfelelő i értékkel, akkor M elfogadja a w szót. Különben meg, ha nincs ilyen s szó, akkor M nem áll meg. Ezért $L(M) = L$.

9. Legyen $L \subseteq \{x\#y : x, y \in \{0{,}1\}^*\}$ rekurzívan felsorolható. Következik-e ebből, hogy az

   $$L_1 = \{x \in \{0{,}1\}^* : \text{ van olyan } y \in \{0{,}1\}^*, \text{ hogy } x\#y \in L\}$$

   nyelv is rekurzívan felsorolható?

   Megoldás: Megmutatjuk, hogy $L_1 \in \mathrm{RE}$. Ehhez egy M_1 TG-t kell vázolni, amire $L(M_1) = L_1$. Az M_1 gép egy x bemeneten lényegében olyan y-t keres, amire $x\#y \in L$. Ehhez a szokásos módon sorban veszi az (s,i) szó-lépésszámkorlát párokat, és egy ilyen párnál az L-et felismerő M gépet az $x\#s$ bemeneten futtatja i lépésig. Ha ez alatt M elfogad, akkor M_1 álljon meg elfogadó állapotban, különben lépjen a következő párra.

   Mivel $x \in L_1$ pontosan akkor teljesül, ha van olyan y, amire $x\#y \in L$, azaz $M(x\#y)$ elfogad mondjuk j lépésben. Amikor a pároknál M_1 először ér egy olyan (y,i) párhoz, ahol $i \geq j$, akkor M_1 meg fog állni és elfogad. Ha viszont $x \notin L_1$, akkor nincs $x\#y$, amit M elfogadna, tehát M_1 nem fog megállni, azaz ilyenkor $x \notin L(M_1)$. Ezért $L(M_1) = L$.

10. Tekintsük a dominóproblémának azt a változatát, amikor minden dominóknak a vízszintes és a függőleges tengelyre vett tükörképe is használható. Rekurzív-e az így módosított problémához tartozó nyelv?

    Megoldás: Igen, mert így minden nem üres készlet jó. Igazából egyetlen típus is elég, mert akármilyen (a,b,c,d) dominó esetén egy vízszintes sáv lefedhető a dominó és a függőleges tengelyre vett tükörkép váltakozásával.

Ez alá egy olyan sáv jöhet, amik a mindig a felettük levők vízszintes tengelyre vett tükörképei. És ezt a következő sávokra is alkalmazhatjuk.

Megjegyzés: Itt azt használtuk, hogy minden használt dominót szabad tükrözni is. Mi van, ha csak az eredeti készletet lehet?