Bonyolultságelmélet gyakorlatok IV/1 és IV/3 megoldása.

IV. 1/a *Döntsük el, hogy rekurzív, illetve rekurzívan felsorolható-e a következő probléma: Adott egy M Turing-gép, igaz-e, hogy M megáll az üres szón.*

Megoldás. Legyen

$$\text{ÜRESEN MEGÁLLÁS} := \{M \mid M(\varepsilon) \neq\nearrow\}.$$

ÜRESEN MEGÁLLÁS **rekurzívan felsorolható**, mert van rá féligeldöntő algoritmus:

Szimuláljuk M működését az üres szón (ε-on). Ha a szimuláció során azt tapasztaljuk, hogy M az üres szón megáll, akkor a válasz „IGEN".

Ez valóban egy félig eldöntő algoritmus, mert az igen példányokon mindig megáll igen válasszal, a nem példányokon, pedig végtelen sokáig fut, pontosan úgy, ahogy azt a rekurzívan felsorolhatóság (más néven a „féligeldöntés") definíciójában megköveteltük.

Megmutatjuk, hogy ÜRESEN MEGÁLLÁS **nem rekurzív**, mert a megállási probléma (jele: H) visszavezethető **rá**.

Valóban, legyen M_0 és x_0 egy tetszőleges példánya H-nak. Azaz M_0 egy Turing-gép, x_0 pedig M_0 egy bemenete (természetesen a H példányainak megadásához használt kódolás szerint.) Készítsünk el M_0 és x_0 ismeretében egy olyan M' Turing gépet, mely a következők szerint működik:

1. M' először megvizsgálja, hogy az inputja az üres szó-e, ha nem akkor megáll, különben az input szalagjára felmásolja x_0-at. (Egy véges karakterláncot a programjában előre el tudunk tárolni.)
2. M' ezután visszamegy az input elejére (egy új állapot kell hozzá).
3. Majd M' szimulálja M_0 működését az input szalagon (azaz x_0-on). Ez egyszerűen megoldható úgy, hogy M_0 szabályait M'-be másoljuk, és a 2. lépés teljesítése után M_0 kezdőállapotának a másolatára adjuk a vezérlést.

Röviden, elkészíthető olyan M', melyre:

$$M'(y) = \begin{cases} M_0(x_0) & \text{ha } y = \varepsilon, \\ \textbf{Állj!} & \text{ha } y \neq \varepsilon \end{cases} \qquad \forall y \text{ bemenetre.}$$

Most definiáljuk az f függvény a következő módon: $f(M_0; x_0) := M'$. Két dolgot kell bebizonyítanuk:

I. *f kiszámítható függvény.* Valóban, könnyen ellenőrizhető, hogy M_0 és x_0 ismeretében M' (pontosabban M_0 és x_0 kódja ismeretében M' kódja) algoritmikusan megkonstruálható. Először néhány olyan állapotot generálunk, melyek az 1. és 2. lépést végrehajtják, majd M_0 programját alkalmasan beillesztjük M'-éba.

II. *f visszavezeti* H*-t* ÜRESEN MEGÁLLÁS*-ra.* Ennek bizonyítást a definíciók felhasználásával kapjuk:

$$M_0; x_0 \in \text{H} \implies M_0(x_0) \neq\nearrow \implies M'(\varepsilon) \neq\nearrow \implies f(M_0, x_0) = M' \in \text{ÜRESEN MEGÁLLÁS}.$$

$$M_0; x_0 \notin \text{H} \implies M_0(x_0) =\nearrow \implies M'(\varepsilon) =\nearrow \implies f(M_0, x_0) = M' \notin \text{ÜRESEN MEGÁLLÁS}.$$

Ezzel bebizonyítottuk, hogy H $\leq_{rek}$ ÜRESEN MEGÁLLÁS, azaz H rekurzívan visszavezethető ÜRESEN MEGÁLLÁS-ra. Ezért ÜRESEN MEGÁLLÁS legalább olyan nehéz mint H, így mivel H eldönthetetlen ÜRESEN MEGÁLLÁS is eldönthetetlen.

IV. 1/b *Döntsük el, hogy rekurzív, illetve rekurzívan felsorolható-e a következő probléma: Adott egy M Turing-gép, van-e olyan bemenet, amelyen M megáll.*

Megoldás. Legyen

$$\text{VALAMIKOR MEGÁLLÁS} := \{M \mid \exists x : M(x) \neq\nearrow\}.$$

VALAMIKOR MEGÁLLÁS **rekurzívan felsorolható**, mert van rá féligeldöntő algoritmus: Párhuzamosan szimuláljuk M működését az összes lehetséges inputon. Először csak a rövidebb input szavakon, majd folyamatosan a hosszabb szavakon is elindítva egy-egy folyamatot. Részletesebben: rendezzük hossz szerint, majd az azonos hosszúak között ábécé rendbe az összes lehetséges inputot. Indítsunk el M egy szimulációját az első szón. Majd ismételjük a következőt: miután végrehajtottuk az összes futó szimuláció egy-egy lépését indítsunk el egy újabb szimulációt, a következő szón. Ha valamelyik szimuláció során azt tapasztaljuk, hogy M megáll, akkor a válasz ,,IGEN".

VALAMIKOR MEGÁLLÁS **nem rekurzív**, mert H visszavezethető **rá**.

Valóban, legyen M_0 és x_0 egy tetszőleges példánya H-nak. Készítsünk el M_0 és x_0 ismeretében egy olyan M' Turing gépet, mely a következők szerint működik.

1. M' először megvizsgálja, hogy a bemenete megegyezik-e x_0-lal. Ha nem, végtelen ciklusba kerül.
2. Ha igen, akkor M' szimulálja M_0 működését az input szalagon (azaz x_0-on).

Röviden, elkészíthető olyan M', mely az alábbi módon működik:

$$M'(y) = \begin{cases} M_0(x_0) & \text{ha } y = x_0, \\ \nearrow & \text{ha } y \neq x_0 \end{cases} \qquad \forall y \text{ bemenetre.}$$

Most definiáljuk az f függvény a következő módon: $f(M_0; x_0) := M'$.

I. f *kiszámítható függvény.* Könnyen látható, hogy ilyen M' létezik és M_0 és x_0 ismeretében egy ilyen M' algoritmikusan megkonstruálható.

II. f *visszavezeti* H-*t* VALAMIKOR MEGÁLLÁS-*ra:*

$$f(M_0, x_0) = M' \in \text{VALAMIKOR MEGÁLLÁS} \Leftrightarrow \exists x : M'(x) \neq\nearrow \Leftrightarrow M_0(x_0) \neq\nearrow \Leftrightarrow M_0; x_0 \in H.$$

Ezzel bebizonyítottuk, hogy H $\leq_{rek}$ VALAMIKOR MEGÁLLÁS, azaz H rekurzívan visszavezethető VALAMIKOR MEGÁLLÁS-ra. Ezért VALAMIKOR MEGÁLLÁS legalább olyan nehéz mint H, így mivel H eldönthetetlen VALAMIKOR MEGÁLLÁS is eldönthetetlen.

IV. 1/c *Döntsük el, hogy rekurzív, illetve rekurzívan felsorolható-e a következő probléma: Adott egy M Turing-gép, van-e olyan bemenet, amelyen M valamely lépésben a σ szimbólumot írja a szalagra.*

Megoldás. Legyen

$$\text{ÍR SZIGMÁT} := \{M \mid \exists x : M \text{ } x\text{-en futtatva valamikor } \sigma\text{-t ír ki.}\}$$

ÍR SZIGMÁT **rekurzívan felsorolható**, mert van rá féligeldöntő algoritmus, az összes bemeneten párhuzamosan kell szimulálni M működését.

De **nem rekurzív**, mert H visszavezethető rá. Adott M_0 és x_0 esetén konstruáljunk meg egy következő M'-t:

$$M'(y) = \begin{cases} \textbf{Írj ki } \sigma\textbf{-t!} & \text{ha } y = x_0 \text{ és } M_0(x_0) \neq\nearrow, \\ \nearrow & \text{különben} \end{cases} \qquad \forall y \text{ bemenetre.}$$

Feltehetjük, hogy M' az $M_0(x_0)$ szimuláció során sohasem ír ki σ-t, például a szimulációt kódolva végzi. Belátható, hogy ilyen M' létezik, (ehhez szükséges, hogy a második sor elejére $\nearrow$ kerüljön, hiszen az esetszétválasztás $M_0(x_0) \neq \nearrow$ feltételének tesztelése nem rekurzív subrutin!) Továbbá könnyen látható, hogy az M' az M_0 és x_0 ismeretében algoritmikusan megkonstruálható. Így $f(M_0; x_0) := M'$ rekurzív függvény.

Végül f visszavezeti H-t ÍR SZIGMÁT-ra, mert

$$f(M_0, x_0) = M' \in \text{ÍR SZIGMÁT} \iff \exists x : M'(x) \text{ ír ki } \sigma\text{-t} \iff M_0(x_0) \neq \nearrow \iff M_0; x_0 \in H.$$

IV. 1/d *Döntsük el, hogy rekurzív, illetve rekurzívan felsorolható-e a következő probléma: Adott egy M Turing-gép, van-e olyan bemenet, amelyen M valamely lépésben az éppen olvasottól különbző szimbólumot ír ki.*

A probléma **eldönthető**. A bizonyítás vázlata a következő: Adott $M = (K, \Sigma, \delta, s)$ Turing-gép esetén, annak minden ,,rossz", azaz $\delta(p, \sigma) = (q, \rho, D)$ alakú szabályát, ahol $p, q \in K$, $\sigma, \rho \in \Sigma$, $D \in \{\leftarrow, -, \rightarrow\}$ és $\rho \neq \sigma$, cseréljük ki a $\delta(p, \sigma) = (\text{,,HIBA''}, \sigma, -)$ szabályra, ahol ,,HIBA" egy új állapot. Az így módosított Turing-gép, már nem írhat a szalagra, pontosan úgy viselkedik, mint egy *kétirányú véges automata*. Mivel a kétirányú véges automaták szimulálhatók hagyományos (egyirányú) véges automatákkal (Ld. Hopcroft–Ullman: *Introduction to Automata Theory, Languages and Computation*, 40. oldal), így az eredeti Turing-gép akkor és csak akkor igen példány, azaz változtat a szalagja tartalmán, ha a belőle készített automata a ,,HIBA" állapottal, mint egyetlen végállapottal nem az üres nyelvet ismeri fel. Ismert azonban, hogy ez az utóbbi probléma eldönthető, (ha egy automata felismer valamit, akkor legalább egy, az állapotszámánál kevesebb betűből álló szót is felismer, azért elég csak véges sok szóra tesztelni az automatát). Így összességében a feladat problémája visszavezethető arra a problémára, hogy egy véges automata az üres nyelvet ismeri-e fel. Ezért a feladat problémája nem lehet nehezebb, így eldönthető.

IV. 1/e *Döntsük el, hogy rekurzív, illetve rekurzívan felsorolható-e a következő probléma: Adott egy M Turing-gép, igaz-e, hogy a Turing-gép semmit sem ismer fel, vagyis $L(M) = \emptyset$.*

Megoldás. Legyen

$$\text{SIF} := \{M \mid L(M) = \emptyset\}$$

A probléma **még csak nem is féligeldönthető**, mert a megállási probléma komplementere

$$co\text{H} = \{M_0; x_0 \mid M_0(x_0) = \nearrow\}$$

visszavezethető rá. Valóban, egy tetszőleges M_0 Turing-gép és annak x_0 bemenete ismeretében konstruáljunk, olyan M'-t, melyre

$$M'(y) = \begin{cases} \textbf{,,igen''} & \text{ha } y = x_0 \text{ és } M_0(x_0) \neq \nearrow, \\ \nearrow & \text{különben} \end{cases} \qquad \forall y \text{ bemenetre.}$$

Könnyen látható, hogy ekkor $f(M_0; x_0) := M'$ rekurzív függvény, és

$$f(M_0, x_0) = M' \in \text{SIF} \Leftrightarrow \forall y : M'(y) \neq \text{,,igen''} \Leftrightarrow \forall y : M'(y) = \nearrow \Leftrightarrow M_0(x_0) = \nearrow \Leftrightarrow M_0; x_0 \in co\text{H}.$$

Így mivel coH nem rekurzívan felsorolható és visszavezethető SIF-re, SIF sem rekurzívan felsorolható.

IV. 1/f *Döntsük el, hogy rekurzív, illetve rekurzívan felsorolható-e a következő probléma: Adott egy M Turing-gép, igaz-e, hogy a Turing-gép véges nyelvet is,mer fel, azaz $L(M)$ véges halmaz.*

Megoldás. Nem rekurzívan felsorolható, mert a $M'(y) := M_0(x_0)$ (azaz M' az y-tól függetlenül mindig úgy működik mint M_0 az x_0-on) hozzárendeléssel definiált $f(M_0; x_0) := M'$ függvény visszavezeti coH-t a problémára (HF!).

IV. 1/g *Döntsük el, hogy rekurzív, illetve rekurzívan felsorolható-e a következő probléma: Adottak az M és M' Turing-gépek, igaz-e, hogy ekvivalensek, azaz $L(M) = L(M')$.*

Megoldás. Nem rekurzívan felsorolható, mert az alábbi függvény visszavezeti *co*H-t rá:

Adott M_0 és x_0 esetén, legyen $M'(y) := \nearrow$, $\forall y$-ra, azaz egy soha meg nem álló Turing-gép és $M'(y) := M_0(x_0)$, bármely y bemenetre. Az így definiált két Turing-gép akkor és csak akkor ekvivalens, ha $M_0(x_0) = \nearrow$, vagyis $M_0; x_0 \in co\mathrm{H}$.

IV. 1/h *Döntsük el, hogy rekurzív, illetve rekurzívan felsorolható-e a következő probléma: Adottak az M Turing-gép, igaz-e, hogy van olyan x bemenet, hogy M az x-en öt lépésen belül megáll.*

Megoldás. Rekurzív (eldönthető). Világos, hogy M működésének első öt lépése, csak a bemenet első öt betűjétől függ. Ezért az összes 5 betűs szó kipróbálásával (véges sok eset van) eldönthető, hogy M igen példány-e vagy sem.

IV. 1/i *Döntsük el, hogy rekurzív, illetve rekurzívan felsorolható-e a következő probléma: Adottak az M Turing-gép, igaz-e, hogy nincs olyan x bemenet, hogy M az x-en öt lépésen belül megáll.*

Megoldás. Eldönthető. Az előző probléma komplementere, és rekurzív nyelv komplementere is rekurzív.

IV. 3 *Az előző feladat mely pontjainak eldönthetetlenségét tudjuk közvetlenül bizonyítani a Rice-tétel segítségével?*

Emlékeztető:

Rice-tétel. A rekurzívan felsorolható nyelvek egyetlen nemtriviális[1] tulajdonsága sem dönthető el algoritmikusan. Azaz bármely $\emptyset \subsetneq \mathcal{C} \subsetneq \mathbf{RE}$ esetén az $\{M \mid L(M) \in \mathcal{C}\}$ nyelv eldönthetetlen.

Ezek után a problémák eldönthetetlenégének belátásához definiáljuk a $\mathcal{C}$ osztályt az alábbi módon:

a) $\mathcal{C} := \{L \in \mathbf{RE} \mid \varepsilon \in L\}$

b) $\mathcal{C} := \{L \in \mathbf{RE} \mid L \neq \emptyset\} = \mathbf{RE}\setminus\{\emptyset\}$, feltéve, hogy előtte M-et átalakítjuk, hogy mindig ,,igen" állapotban álljon meg.

c) Nem alkalmazható a Rice-tétel, a probléma magától a Turing-géptől és nem csak az általa felismert nyelvtől függ.

d) Mint c).

e) $\mathcal{C} := \{\emptyset\}$, csak az üres nyelvből álló osztály.

f) $\mathcal{C} := \{L \mid |L| < \infty\}$ a véges nyelvek.

g) Átfogalmazva: M, M' ekvivalensek $\iff L(M') \in \mathcal{C}_M := \{L(M)\}$.

h-i) Nem alkalmazható a Rice-tétel, a probléma magától a Turing-géptől és nem csak az általa felismert nyelvtől függ.

Így közvetlenül belátható, hogy az a) b) e) f) és g) problémák nem rekurzívak.
(Ugyanakkor a rekurzív felsorolhatóságukról nem nyilatkozik a Rice-tétel!)

[1]nem minden rek. felsorolható nyelv, de nem is egyik se.