Bevezetés a kiszámíthatóság és algoritmusok bonyolultságának elméletébe

Ésik Zoltán

1 Bevezetés

A kiszámíthatóság elmélet a matematikai logikán belül, a matematikai logikával együtt alakult ki. 1900-ban, a századforduló alkalmából, Hilbert 23 problémát vetett fel a XX. század matematikusai számára. Ezen problémák egyike, a 10., olyan algoritmus kifejlesztését, létezését kérdezi, amellyel tetszőleges

$$f(x_1, \ldots, x_n) \quad = \quad g(x_1, \ldots, x_n)$$

diofantikus egyenletről (tehát $f(x_1, \ldots, x_n)$ és $g(x_1 \ldots, x_n)$ egész együtthatós polinomok) eldönthető, van-e (egész értékű) megoldása. Megj.: itt egész helyett mondhatunk természetes számot is. E problémát Matijasevič oldotta meg 1971-ben felhasználva Robinson, Davis és Putnam korábbi eredményeit: a kérdéses algoritmus nem létezik. Ugyancsak Hilbert, 1928-ban, a matematikai logika fő problémáját abban fogalmazta meg, hogy olyan algoritmust kellene találni, amely az elsőrendű kalkulus (függvénykalkulus) tetszőleges kijelentésére eldönti, hogy kielégíthető-e. Ha ilyen algoritmus létezne, akkor elvileg a matematika összes problémája algoritmikusan megoldható lenne. A 30-as évek közepén Church és Turing megmutatták, hogy ilyen algoritmus nem létezik. Ehhez szükség volt arra, hogy megalkossák az algoritmus intuitív fogalmának matematikai megfelelőjét.

A függvények azon osztályát, amelyet ma primitív rekurzív függvényeknek nevezünk, Gödel vezette be 1931-ben. Az 1934-es év tavaszán Gödel Princetonba látogatott és az Institute of Advanced Study szemináriumán tartott előadássorozatot. Az előadásokon részt vett Church, Rosser és Kleene is. Itt, Herbrand javaslatára, Gödel lényegében definiálta az általánosabb rekurzív függvény fogalmát, és megfogalmazta azt az elképzelését, hogy a rekurzív függvény fogalma megfelel a “mechanikusan” kiszámítható függvény fogalmának. Megjegyezte, hogy ez heurisztikus elvként szolgálhat, annak ellenére, hogy nem bizonyítható, hiszen a véges (mechanikus) kiszámítás fogalma matematikai értelemben nem definiált. Mindazonáltal úgy gondolta, hogy az emberi elme képes nem csak mechanikus kiszámítások elvégzésére is.

Az 1930-as évek elején Church, Kleene és Rosser a λ-kalkuluson dolgozott, amely egy olyan logikai rendszer, amely a függvény fogalmán és a függvények változók értékeire való alkalmazásán alapszik. A λ-kalkuluson belül megalkották a λ-definiálható (vagy λ-kiszámítható) függvény fogalmát. Egy Gödellel folytatott beszélgetés közben Church kifejtette azt a nézetét, amelyet Gödel akkor nem tartott megfelelőnek, hogy a λ-definiálható függvények megfelenek az effektíven kiszámítható függvényeknek.

1935 áprilisában az Amerikai Matematikai Társaság New York-ban tartotta összejövetelét. Itt Church megfogalmazta azt a róla elnevezett tézist, hogy a Gödel által 1934-ben javasolt általános rekurzivitás megfelel az effektíven vagy algoritmikusan kiszámítható függvény fogalmának. Az előadás összefoglalójában Church utal arra, hogy más ekvivalens fogalmak is létezhetnek, pld. a λ-definiálhatóság. Azt, hogy minden λ-definiálható függvény általánosan rekurzív, Church és Kleene egymástól függetlenül bizonyították körülbelül ebben az időben. A fordított irányú állítás Kleene 1936-ban megjelent munkájában nyert bizonyítást.

A kiszámítható függvény fogalmának matematikailag egyik legszebb és legmeggyőzőbb analízise azonban Turing nevéhez fűződik. Turing Angliában dolgozott, a Princeton-i iskolától függetlenül.

1936-ban jelent meg az a cikke, amelyben a róla elnevezett gépet definiálja, és megfogalmazza azt a tézist, hogy a Turing géppel kiszámítható függvények megfelelnek az algoritmikusan kiszámítható függvény intuitív fogalmának. A cikk közlésre való leadásakor ismerte meg a λ-definiálható függvény fogalmát, és egy rövid appendixben vázolta annak bizonyítását, hogy a két fogalom ekvivalens.

A Church-Turing tézist tapasztalati tények és matematikai eredmények támasztják alá. Egyrészt minden intuitíven algoritmikusan megoldható problémáról, algoritmikusan kiszámítható függvényről sikerült megmutatni, hogy azok matematikai értelemben is megoldhatóak, kezelhetőek az említett modellekben. Másrészt a matematikusok (pld. Post, Lambek, Minsky, Markov, Shepherdson, Sturgis,...) más egyéb modelleket is javasoltak, és ezek mindegyikéről kiderült, hogy ekvivalensek az egyéb modellekkel. Ezen eredmények a Church tézis legfőbb bizonyítékai.

Az előadásban több oldalról is felépítjük a kiszámítható (rekurzív) függvény fogalmát, megismerkedünk több olyan matematikai modellel, amelyekben az algoritmikus problémák vizsgálhatóak. Megismerkedünk továbbá a kiszámíthatóság elméletének, a rekurzív függvények elméletének alapvető eredményeivel.

2 Programok és kiszámítható függvények

Bevezetünk egy $\mathcal{L}$ programozási nyelvet a nemnegatív egész számok $\mathbf{N}$ halmaza felett, amely bár igen egyszerű, mégis elég gazdag ahhoz, hogy minden kiszámítható függvényt előállítson.

Szintaxis

Változók:

- bemenő változók: $X_1, X_2, \ldots$
- munka változók: $Z_1, Z_2, \ldots$
- kimenő változó: Y

Címkék: $A_1, B_1, C_1, A_2, B_2, C_2, \ldots$

Utasítások:

- $V \leftarrow V$
- $V \leftarrow V + 1$
- $V \leftarrow V - 1$
- `if` $V \neq 0$ `goto` L

ahol V változó, L címke. (A $V \leftarrow V$ alakú utasítások fölöslegesek, hiszen mint látni fogjuk, hatásuk ekvivalens a $V \leftarrow V + 1$, $V \leftarrow V - 1$ utasításokkal. Technikai szempontból mégis előnyös lesz a $V \leftarrow V$ alakú utasításokat alaputasításként tekinteni.) Megengedett az utasítások címkézése.

Program: utasítások véges, esetleg üres sorozata.

Szemantika Legyen P egy n utasításból álló program. A P egy σ állapotán $V = k$ ($k \in \mathbf{N}$) alakú egyenletek egy olyan halmazát értjük, amelyben minden változó pontosan egyszer fordul elő. Egy σ állapot iniciális, ha σ-ban teljesül $Y = 0$ és $Z = 0$, valahányszor Z munka változó. A P egy pillanatfelvételén egy (i, σ) párt értünk, ahol $1 \leq i \leq n+1$ és σ egy állapot. Iniciális pillanatfelvétel egy olyan (i, σ) pillanatfelvétel, amelyre $i = 1$, σ pedig iniciális állapot. Terminális egy (i, σ) pillanatfelvétel ha $i = n+1$. Legyen (i, σ) olyan pillanatfelvétel, amely nem terminális. Ennek (j, τ) rákövetkezőjét az alábbi módon definiáljuk, ahol $V = k$ a σ állapotban van:

i) P i-edik utasítása $V \leftarrow V$. Ekkor $j = i + 1$, $\tau = \sigma$.

ii) P i-edik utasítása $V \leftarrow V + 1$. Ekkor $j = i + 1$ és $\tau = (\sigma - \{V = k\}) \cup \{V = k + 1\}$.

iii) P i-edik utasítása $V \leftarrow V - 1$. Ekkor $j = i+1$ és $\tau = (\sigma - \{V = k\}) \cup \{V = k'\}$, ahol $k' = k - 1$ vagy $k' = 0$, attól függően, hogy $k > 0$ vagy $k = 0$.

iv) P i-edik utasítása `if` $V \neq 0$ `goto` L. Ekkor $\tau = \sigma$ és két aleset lehetséges.

Ha $k = 0$ akkor $j = i + 1$.

Ha $k > 0$ akkor, ha van P-ben L címkéjű utasítás, j az első ilyen sorszáma. Különben $j = n+1$. (Következésképpen nem definiált címkére való ugrás a programból való kilépést fog eredményezni.)

A P egy kiszámítási sorozata pillanatfelvételek olyan $s_1, s_2, \ldots, s_k$ ($k \geq 1$) sorozata, amelyben

s_1 iniciális,

s_{i+1} az s_i rákövetkezője ($i = 1, \ldots, k-1$),

s_k terminális.

Ha s_1 olyan iniciális pillanatfelvétel, amelyben $X_j = 0$ valahányszor $j > m$, ahol $m \geq 1$ adott egész szám, akkor $s_1, s_2, \ldots, s_k$ m-változós kiszámítási sorozat.

Definíció 2.1 *Legyen P program, $m \geq 1$ rögzített egész szám. A P által kiszámított m-változós parciális függvényen azt a $\psi_P^{(m)} : \mathbf{N}^m \longrightarrow \mathbf{N}$ parciális függvényt értjük, amelyre $\psi_P^{(m)}(r_1, \ldots, r_m) = r$ akkor és csakis akkor, ha létezik olyan $s_1, \ldots, s_k$ kiszámítási sorozat, amelyre*

1) s_1 m-változós, és benne $X_1 = r_1, \ldots, X_m = r_m$,

2) s_k-ban $Y = r$.

Tehát egy és ugyanazon program felhasználható különböző $m \geq 1$ egész számokra m-változós parciális függvény kiszámítására.

Definíció 2.2 *Legyen $f : \mathbf{N}^m \longrightarrow \mathbf{N}$ ($m \geq 1$) parciális függvény. f parciális rekurzív (vagy parciális kiszámítható) ha $f = \psi_P^{(m)}$ valamely P programra. Ha f teljesen definiált is, akkor f rekurzív (vagy kiszámítható). Parciális rekurzív predikátum egy olyan parciális rekurzív függvény, mely értékk'eszletében csak a 0 (*`hamis`*) és 1 (*`igaz`*) számok fordulnak elő. Rekurzív predikátum egy teljesen definiált parciális rekurzív predikátum.*

3 Néhány technikai megjegyzés

A `goto` L feltétlen ugrás programozása:

$$\begin{array}{l} Z \leftarrow Z + 1 \\ \texttt{if } Z \neq 0 \texttt{ goto } L \end{array}$$

ahol Z új munkaváltozó.

Az `if` $V = 0$ `goto` L programozása:

$$\begin{array}{ll} & \texttt{if } V \neq 0 \texttt{ goto } L' \\ & \texttt{goto } L \\ L' & \end{array}$$

ahol L' új címke.

Változó beállítása (azaz $V \leftarrow k$ programozása, ahol k egy konstans):

$$\begin{array}{ll} L & V \leftarrow V - 1 \\ & \texttt{if } V \neq 0 \texttt{ goto } L \\ & \left.\begin{array}{l} V \leftarrow V + 1 \\ \quad\vdots \\ V \leftarrow V + 1 \end{array}\right\} \quad k\text{-szor} \end{array}$$

ahol L új címke.

Értékadás (azaz $V_1 \leftarrow V_2$ programozása):

$$\begin{array}{ll} & V_1 \leftarrow 0 \\ L_1 & \texttt{if } V_2 = 0 \texttt{ goto } L_2 \\ & V_2 \leftarrow V_2 - 1 \\ & V_1 \leftarrow V_1 + 1 \\ & Z \leftarrow Z + 1 \\ & \texttt{goto } L_1 \\ L_2 & \texttt{if } Z = 0 \texttt{ goto } E \\ & Z \leftarrow Z - 1 \\ & V_2 \leftarrow V_2 + 1 \\ & \texttt{goto } L_2 \end{array}$$

ahol L_1, L_2 új címkék, Z új munkaváltozó.

4 A makró elv

Legyen $f : \mathbf{N}^n \to \mathbf{N}$ parciális rekurzív függvény. Világos, hogy $f = \psi_P^{(n)}$ egy olyan P programra is, amelyben mondjuk legfeljebb az

$$Y, X_1, \ldots, X_n, Z_1, \ldots, Z_k, E, A_1, \ldots, A_l$$

változók illetve címkék fordulnak elő, ahol E az exit címke - azaz az egyetlen olyan definiálatlan címke, amelyre van vezérlés átadás. (E tetszőleges, ha ilyen nincs.) A

$$P_m \quad = \quad P(Z_m, \ldots, Z_{m+n+k}, A_m, \ldots, A_{m+l})$$

programon azt a programot értjük, amelyet úgy kapunk P-ből, hogy P változóit és címkéit a megfelelő

$$Z_m, \ldots, Z_{m+n+k}, A_m, \ldots, A_{m+l}$$

változóval, illetve címkével helyettesítjük. Ezek után ha egy programban a

$$W \leftarrow f(V_1, \ldots, V_n)$$

makró utasítást írjuk, ezen az alábbi programrészletet értjük:

$$\begin{array}{ll} & Z_m \leftarrow 0 \\ & Z_{m+1} \leftarrow V_1 \\ & \quad\vdots \\ & Z_{m+n} \leftarrow V_n \\ & Z_{m+n+1} \leftarrow 0 \\ & \quad\vdots \\ & Z_{m+n+k} \leftarrow 0 \\ & P_m \\ A_m & W \leftarrow Z_m \end{array}$$

ahol m elég nagy ahhoz, hogy a program többi részében $Z_m, \ldots, Z_{m+n+k}, A_m \ldots, A_{m+l}$ ne forduljon elő. Hasonlóan, ha P n-változós parciális rekurzív predikátum, akkor az

$$\texttt{if } P(V_1, \ldots, V_n) \texttt{ goto } L$$

makró utasításon a

$$\begin{array}{l} W \leftarrow P(V_1, \ldots, V_n) \\ \texttt{if } W \neq 0 \texttt{ goto } L \end{array}$$

programot értjük, ahol W új munka változó.

5 Primitív rekurzív függvények

Definíció 5.1 *Legyenek $f : \mathbf{N}^k \longrightarrow \mathbf{N}$ és $g_1, \ldots, g_k : \mathbf{N}^n \longrightarrow \mathbf{N}$ (parciális) függvények. Azt a $h : \mathbf{N}^n \longrightarrow \mathbf{N}$ (parciális) függvényt, amelyre*

$$h(x_1, \ldots, x_n) = f(g_1(x_1, \ldots, x_n), \ldots, g_k(x_1, \ldots, x_n))$$

az f-ből és a $g_1, \ldots, g_k$ függvényekből előálló összetett (parciális) függvénynek nevezzük.

Állítás 5.2 *Ha $f, g_1, \ldots, g_k$ (parciális) rekurzív függvények, akkor a h összetett függvény is az.*

Bizonyítás. Tekintsük az alábbi programot:

$$\begin{array}{l} Z_1 \leftarrow g_1(X_1, \ldots, X_n) \\ \quad\vdots \\ Z_k \leftarrow g_k(X_1, \ldots, X_n) \\ Y \leftarrow f(Z_1, \ldots, Z_k) \quad \square \end{array}$$

Definíció 5.3 *Legyenek $f : \mathbf{N}^n \longrightarrow \mathbf{N}$ és $g : \mathbf{N}^{n+2} \longrightarrow \mathbf{N}$ (teljesen definiált) függvények. Azt az egyértelműen meghatározott $h : \mathbf{N}^{n+1} \longrightarrow \mathbf{N}$ függvényt, amelyre*

$$\begin{aligned} h(x_1, \ldots, x_n, 0) &= f(x_1, \ldots, x_n) \\ h(x_1, \ldots, x_n, t+1) &= g(t, h(x_1, \ldots, x_n, t), x_1, \ldots, x_n), \end{aligned}$$

az f-ből és g-ből primitív rekurzióval kapott függvénynek nevezzük.

(A változók sorrendje érdektelen a későbbiekben bevezetésre kerülő PRZ osztályok és a primitív rekurzív függvény fogalma szempontjából. Az $n = 0$ eset is megengedett: ekkor f egy szám.)

Állítás 5.4 *Ha f és g rekurzívak, akkor a primitív rekurzióval nyert h függvény is az.*

Bizonyítás. Tekintsük az alábbi programot:

$$\begin{array}{ll} & Y \leftarrow f(X_1 \ldots, X_n) \\ A & \texttt{if } X_{n+1} = 0 \texttt{ goto } E \\ & Y \leftarrow g(Z, Y, X_1, \ldots, X_n) \\ & Z \leftarrow Z + 1 \\ & X_{n+1} \leftarrow X_{n+1} - 1 \\ & \texttt{goto } A \quad \square \end{array}$$

Definíció 5.5 *Alapfüggvények az $s : \mathbf{N} \longrightarrow \mathbf{N}$, $s(x) = x+1$ rákövetkezési függvény és az $u_i^n : \mathbf{N}^n \longrightarrow \mathbf{N}$, $u_i^n(x_1, \ldots, x_n) = x_i$, projekciós függvények, ahol $n \geq 1$, $1 \leq i \leq n$. A természetes számok feletti $\mathcal{C}$ függvényosztály PRZ-osztály, ha tartalmazza az alapfüggvényeket, továbbá zárt az összetett függvény képzésre és a primitív rekurzióra.*

Tétel 5.6 *A rekurzív függvények osztálya PRZ-osztály.*

Bizonyítás. Az s függvény kiszámítható az $Y \leftarrow X_1 + 1$ programmal, az u_i^n projekciós függvény pedig az $Y \leftarrow X_i$ programmal. Másrészt a rekurzív függvények osztálya zárt az összetett függvény képzésre és a primitív rekurzióra. □

Definíció 5.7 *A legszűkebb PRZ-osztályt a primitív rekurzív függvények osztályának nevezzük.*

Tehát egy függvény akkor és csakis akkor primitív rekurzív, ha előáll az alapfüggvényekből a függvényösszetétel és a primitív rekurzió véges sokszori alkalmazásával.

Példa 5.8 1. Minden konstans függvény:

$$\begin{aligned} k(0) &= k \\ k(x+1) &= k(x) \end{aligned}$$

2. $x + y$

$$\begin{aligned} f(x,0) &= x \\ f(x,y+1) &= f(x,y)+1 \end{aligned}$$

3. $x \cdot y$

$$\begin{aligned} f(x,0) &= 0 \\ f(x,y+1) &= f(x,y)+x \end{aligned}$$

4. $x!$

$$\begin{aligned} 0! &= 1 \\ (x+1)! &= (x!) \cdot (x+1) \end{aligned}$$

5. x^y

$$\begin{aligned} x^0 &= 1 \\ x^{y+1} &= x^y x \end{aligned}$$

6. $p(x) = \begin{cases} x-1 & \text{ha } x \neq 0 \\ 0 & \text{ha } x = 0 \end{cases}$

$$\begin{aligned} p(0) &= 0 \\ p(x+1) &= x \end{aligned}$$

7. $x \dot{-} y = \begin{cases} x-y & \text{ha } x \geq y \\ 0 & \text{ha } x < y \end{cases}$

$$\begin{aligned} x \dot{-} 0 &= x \\ x \dot{-} (y+1) &= p(x \dot{-} y) \end{aligned}$$

8. $|x-y|$

$$|x-y| \quad = \quad (x \dot{-} y)+(y \dot{-} x)$$

9. $\alpha(x)=\begin{cases} 1 & \text{ha } x=0 \\ 0 & \text{ha } x \neq 0 \end{cases}$

$$\begin{aligned} \alpha(0) &= 1 \\ \alpha(x+1) &= 0 \end{aligned}$$

Definíció 5.9 *Egy olyan predikátumot, amelyik primitív rekurzív függvény, primitív rekurzív predikátumnak nevezünk.*

Példa 5.10 1. $x=y$

$$\alpha(|x-y|)$$

2. $x \geq y$

$$\alpha(y \dot{-} x)$$

Állítás 5.11 *Legyen $\mathcal{C}$ PRZ-osztály. Ha a P és Q predikátumok $\mathcal{C}$-hez tartoznak, akkor a $\sim P$, $P \vee Q$ és $P \wedge Q$ predikátumok is.*

Következmény 5.12 *Ha P, Q (primitív) rekurzívak, akkor $\sim P$, $P \vee Q$ és $P \wedge Q$ is azok.*

Bizonyítás.

$$\begin{aligned} \sim P &= \alpha(P) \\ P \wedge Q &= P \cdot Q \\ P \vee Q &= \sim(\sim P \wedge \sim Q). \quad \square \end{aligned}$$

Állítás 5.13 *Legyen $\mathcal{C}$ PRZ-osztály, és tegyük fel, hogy az f, g n-változós függvények, valamint a P n-változós predikátum $\mathcal{C}$-hez tartozik. Akkor az alábbi h is $\mathcal{C}$-ben van:*

$$h(x_1,\ldots,x_n) \quad = \quad \begin{cases} f(x_1,\ldots,x_n) & \text{ha } P(x_1,\ldots,x_n)=1 \\ g(x_1,\ldots,x_n) & \text{ha } P(x_1,\ldots,x_n)=0 \end{cases}$$

Következmény 5.14 *Ha f, g és P (primitív) rekurzívak, akkor a fenti h is az.*

Bizonyítás.

$$h(x_1,\ldots,x_n) \quad = \quad P(x_1,\ldots,x_n) \cdot f(x_1,\ldots,x_n) + (\sim P(x_1,\ldots,x_n)) \cdot g(x_1,\ldots,x_n). \quad \square$$

Állítás 5.15 *Legyenek f és g n-változós parciális rekurzív függvények, P pedig n-változós parciális rekurzív predikátum. Ekkor a*

$$h(x_1,\ldots,x_n) = \begin{cases} f(x_1,\ldots,x_n) & \text{ha } P(x_1,\ldots,x_n)=1 \\ g(x_1,\ldots,x_n) & \text{ha } P(x_1,\ldots,x_n)=0 \\ \uparrow & \text{különben} \end{cases}$$

függvény is parciális rekurzív.

6 Korlátozott kvantorok és minimalizálás

Állítás 6.1 *Legyen $\mathcal{C}$ PRZ-osztály. Ha $f(t,x_1,\ldots,x_n)$ $\mathcal{C}$-ben van, akkor az alábbi g és h is:*

$$\begin{aligned} g(y,x_1,\ldots,x_n) &= \sum_{t=0}^{y} f(t,x_1,\ldots,x_n) \\ h(y,x_1,\ldots,x_n) &= \prod_{t=0}^{y} f(t,x_1,\ldots,x_n) \end{aligned}$$

Következmény 6.2 *Ha f (primitív) rekurzív, akkor g és h is az.*

Bizonyítás.

$$\begin{aligned} g(0,x_1,\ldots,x_n) &= f(0,x_1,\ldots,x_n) \\ g(t+1,x_1,\ldots,x_n) &= g(t,x_1,\ldots,x_n)+f(t+1,x_1,\ldots,x_n). \quad \square \end{aligned}$$

Megjegyzés 6.3 A fenti állítás akkor is igaz, ha az összegzés (szorzás) 1-től indul. (Az üres összeg 0, az üres szorzat 1.)

Definíció 6.4 *Legyen $P(t,x_1\ldots,x_n)$ predikátum. Definiáljuk a $Q(y,x_1,\ldots,x_n)$ és $R(y,x_1,\ldots,x_n)$ predikátumokat az alábbi módon:*

$$\begin{aligned} Q(y,x_1,\ldots,x_n) &\Leftrightarrow (\forall t)_{\le y} P(t,x_1,\ldots,x_n) \\ R(y,x_1,\ldots,x_n) &\Leftrightarrow (\exists t)_{\le y} P(t,x_1,\ldots,x_n) \end{aligned}$$

Állítás 6.5 *Ha P egy $\mathcal{C}$ PRZ-osztályhoz tartozik, akkor Q és R is.*

Következmény 6.6 *Ha P (primitív) rekurzív, akkor Q és R is.*

Bizonyítás.

$$(\forall t)_{t\leq y} P(t, x_1, \ldots, x_n) \quad \Leftrightarrow \quad \prod_{t=0}^{y} P(t, x_1, \ldots, x_n) = 1$$

$$(\exists t)_{t\leq y} P(t, x_1, \ldots, x_n) \quad \Leftrightarrow \quad \sum_{t=0}^{y} P(t, x_1, \ldots, x_n) \neq 0 \quad \square$$

Megjegyzés 6.7 Hasonló állítás igaz a $(\forall t)_{t<y} P(t, x_1, \ldots, x_n)$ és $(\exists t)_{t<y} P(t, x_1, \ldots, x_n)$ predikátumokra is.

Példa 6.8

1. $y|x$

$$y|x \quad \Leftrightarrow \quad (\exists t)_{\leq x}(y \cdot t = x)$$

   (Tehát $0|0$ igaz.)

2. $\text{prím}(x)$

$$\text{prím}(x) \quad \Leftrightarrow \quad x > 1 \wedge (\forall t)_{t\leq x}[t = 1 \ \vee \ t = x \vee \ \sim (t|x)]$$

Definíció 6.9 *Legyen $P(t, x_1, \ldots, x_n)$ tetszőleges predikátum. Ekkor a $\min_{t\leq y} P(t, x_1, \ldots, x_n)$ $n+1$-változós függvényen azt a függvényt értjük, mely értéke adott $y, x_1, \ldots, x_n$ mellett a legkisebb olyan $t \leq y$ amelyre $P(t, x_1, \ldots, x_n) = 1$, ha ilyen van, és amely értéke 0, ha ilyen nincs.*

Állítás 6.10 *Ha $P(t, x_1, \ldots, x_n)$ egy $\mathcal{C}$ PRZ-osztályhoz tartozik, akkor a $\min_{t\leq y} P(t, x_1, \ldots, x_n)$ függvény is $\mathcal{C}$-ben van.*

Következmény 6.11 *Ha $P(t, x_1, \ldots, x_n)$ (primitív) rekurzív, akkor $\min_{t\leq y} P(t, x_1, \ldots, x_n)$ is (primitív) rekurzív.*

Bizonyítás. Legyen

$$g(y, x_1, \ldots, x_n) \quad = \quad \sum_{u=0}^{y} \prod_{t=0}^{u} \sim P(t, x_1, \ldots, x_n)$$

Ekkor g $\mathcal{C}$-ben van, és

$$\min_{t\leq y} P(t, x_1, \ldots, x_n) \quad = \quad \begin{cases} g(y, x_1, \ldots, x_n) & \text{ha } (\exists t)_{\leq y} P(t, x_1, \ldots, x_n) \\ 0 & \text{különben.} \end{cases} \qquad \square$$

Példa 6.12

1. $\lfloor x/y \rfloor$

$$\lfloor x/y \rfloor \quad = \quad \min_{t\leq x}[(t+1) \cdot y > x]$$

   (Tehát $\lfloor x/0 \rfloor = 0$.)

2. $R(x,y)$ (x/y maradéka)

$$R(x,y) = x \dot{-} (y \cdot \lfloor x/y \rfloor)$$

(Tehát $R(x,0) = x$.)

3. p_n: az n-dik prím, pontosabban: $p_0 = 0$, $p_1 = 2$, $p_2 = 3, \dots$

$$\begin{aligned} p_0 &= 0 \\ p_{n+1} &= \min_{t \le p_n!+1} [\text{prím}(t) \wedge t > p_n] \end{aligned}$$

Definíció 6.13 *Legyen $P(x_1, \dots, x_n, y)$ tetszőleges predikátum. Legyen $f(x_1, \dots, x_n)$ a legkisebb olyan y, amelyre $P(x_1, \dots, x_n, y) = 1$, ha ilyen van. Különben $f(x_1, \dots, x_n)$ nem definiált. Jelölés: $f(x_1, \dots, x_n) = \min_y P(x_1, \dots, x_n, y)$.*

Állítás 6.14 *Ha $P(x_1, \dots, x_n, y)$ rekurzív, akkor $\min_y P(x_1, \dots, x_n, y)$ parciális rekurzív.*

Bitzonyítás.

$$\begin{array}{ll} A & \texttt{if } P(X_1, \dots, X_n, Y) \texttt{ goto } E \\ & Y \leftarrow Y + 1 \\ & \texttt{goto } A \quad \square \end{array}$$

Megjegyzés 6.15 Később látni fogjuk, hogy van olyan $P(x_1, \dots, x_n, y)$ primitív rekurzív predikátum, amelyre $\min_y P(x_1, \dots, x_n, y)$ teljesen definiált, tehát rekurzív is, de nem primitív rekurzív. Ezzel szemben ha $P(x_1, \dots, x_n, y)$ rekurzív és $\min_y P(x_1, \dots, x_n, y)$ teljesen definiált, akkor $\min_y P(x_1, \dots, x_n, y)$ nyilván rekurzív. A korlátlan minimalizálás kiterjeszthető parciális predikátumokra is. Legyen $P(x_1, \dots, x_n, y)$ $n+1$-változós parciális predikátum. Legyen $f(x_1, \dots, x_n)$ a legkisebb olyan y, amelyre $P(x_1, \dots, x_n, y) = 1$, és amelyre $z < y$ esetén $P(x_1, \dots, x_n, z) = 0$. Ha ilyen y nincs, $f(x_1, \dots, x_n)$ nem definiált. Ezzel a definícióval érvényes az előző állítás megfelelője. Jelölés: $f(x_1, \dots, x_n) = \min_y P(x_1, \dots, x_n, y)$.

Állítás 6.16 *Ha $P(x_1, \dots, x_n, y)$ parciális rekurzív predikátum, akkor*

$$\min_y P(x_1, \dots, x_n, y)$$

parciális rekurzív függvény.

7 Párképző függvények

Definíció 7.1 *A $\langle -, - \rangle : \mathbf{N}^2 \to \mathbf{N}$ függvényt párképző függvénynek nevezzük, ha primitív rekurzív, bijektív, és $x, y \le \langle x, y \rangle$ teljesül minden x, y esetén.*

Állítás 7.2 *Legyen* $\langle -, - \rangle$ *párképző függvény. Ekkor egyértelműen léteznek olyan* $l, r : \mathbf{N} \longrightarrow \mathbf{N}$ *függvények, amelyekre:*

1. $l(\langle x, y\rangle) = x, \quad r(\langle x, y\rangle) = y$
2. $\langle l(z), r(z)\rangle = z$
3. $l(z), r(z) \leq z.$

Továbbá l *és* r *primitív rekurzívak.*

Bizonyítás.

$$l(z) \;=\; \min_{x\leq z}[(\exists y)_{\leq z}\langle x, y\rangle = z]. \quad \square$$

Állítás 7.3 *Az* $\langle x, y\rangle = 2^x(2y+1) - 1$ *hozzárendeléssel definiált függvény párképző függvény.*

Bizonyítás.

$$l(z) \;=\; \max_{t\leq z} 2^t | (z+1)$$

$$r(z) \;=\; \frac{(z+1)/2^{l(z)} \dot{-} 1}{2}$$

$\square$

Állítás 7.4 *A*

$$\langle x, y\rangle \;=\; \frac{x^2 + 2xy + y^2 + 3x + y}{2}$$

függvény párképző függvény.

A párképző függvény segítségével definiálhatunk $\mathbf{N}^3 \longrightarrow \mathbf{N}$, $\mathbf{N}^4 \longrightarrow \mathbf{N}, \ldots$ primitív rekurzív bijekciókat is. Pld.:

$$\langle x, y, z\rangle \;=\; \langle x, \langle y, z\rangle\rangle$$

és

$$\langle x, y, z\rangle \;=\; \langle\langle x, y\rangle, z\rangle$$

$\mathbf{N}^3 \longrightarrow \mathbf{N}$ primitív rekurzív bijekciók. Adott u esetén az $\langle x, y, z\rangle = u$ egyenlőségnek eleget tevő x, y, z számok az u primitív rekurzív függvényeként állíthatóak elő, és $x, y, z \leq u$.

A párképzési függvényt felhasználva létesíthetünk egy bijekciót az $\mathbf{N}^2 \longrightarrow \mathbf{N}$ (parciális) függvények, és az $\mathbf{N} \longrightarrow \mathbf{N}$ (parciális) függvények között.

Definíció 7.5

$$\begin{aligned} f:\mathbf{N}^2\to\mathbf{N} &\mapsto f^\bullet:\mathbf{N}\to\mathbf{N}\\ f^\bullet(x) &= f(l(x),r(x))\\ g:\mathbf{N}\to\mathbf{N} &\mapsto g^\circ:\mathbf{N}^2\to\mathbf{N}\\ g^\circ(x,y) &= g(\langle x,y\rangle)\end{aligned}$$

Állítás 7.6

$$f^{\bullet\circ}=f,\quad g^{\circ\bullet}=g.$$

Bizonyítás.

$$\begin{aligned} f^{\bullet\circ}(x,y) &= f^\bullet(\langle x,y\rangle)\\ &= f(l(\langle x,y\rangle),r(\langle x,y\rangle))\\ &= f(x,y),\end{aligned}$$

$$\begin{aligned} g^{\circ\bullet}(x) &= g^\circ(l(x),r(x))\\ &= g(\langle l(x),r(x)\rangle)\\ &= g(x).\quad \square\end{aligned}$$

Állítás 7.7 *(i) f (parciális) rekurzív $\Leftrightarrow$ $f^\bullet$ (parciális) rekurzív. g (parciális) rekurzív $\Leftrightarrow$ g° (parciális) rekurzív.*

(ii) Legyen $\mathcal{C}$ PRZ-osztály. Ekkor

$$\begin{aligned} f\in\mathcal{C} &\Leftrightarrow f^\bullet\in\mathcal{C}\\ g\in\mathcal{C} &\Leftrightarrow g^\circ\in\mathcal{C}.\end{aligned}$$

Hasonló dolgok igazolhatóak többváltozós függvényekre is.

Definíció 7.8 *Legyenek $i=1,2$ esetén $f_i:\mathbf{N}^n\to\mathbf{N}$, $g_i:\mathbf{N}^{n+3}\to\mathbf{N}$ adott függvények. Definiáljuk a $h_i:\mathbf{N}^{n+1}\to\mathbf{N}$ függvényeket $i=1,2$ esetén az alábbi összefüggésekkel:*

$$\begin{aligned} h_i(x_1,\ldots,x_n,0) &= f_i(x_1,\ldots,x_n)\\ h_i(x_1,\ldots,x_n,t+1) &= g_i(t,h_1(x_1,\ldots,x_n,t),h_2(x_1\ldots,x_n,t),x_1\ldots,x_n).\end{aligned}$$

Ekkor azt mondjuk, hogy a h_i függvények előállnak az f_i és g_i függvényekből párhuzamos primitív rekurzióval.

Állítás 7.9 *Minden $\mathcal{C}$ PRZ-osztály zárt a párhuzamos primitív rekurzióra.*

Bizonyítás.

$$\begin{aligned}
k(x_1,\ldots,x_n,0) &= \langle f_1(x_1,\ldots,x_n), f_2(x_1,\ldots,x_n)\rangle \\
k(x_1,\ldots,x_n,t+1) &= \langle g_1(t, l(k(x_1,\ldots,x_n,t)), r(k(x_1,\ldots,x_n,t)), x_1,\ldots,x_n), \\
&\quad\; g_2(t, l(k(x_1,\ldots,x_n,t)), r(k(x_1,\ldots,x_n,t)), x_1,\ldots,x_n)\rangle.
\end{aligned}$$

Mivel k a $\mathcal{C}$-ben van, valahányszor f_i és g_i, $i = 1,2$, $\mathcal{C}$-beli függvények, és mivel

$$\begin{aligned}
h_1(x_1,\ldots,x_n,t) &= l(k(x_1,\ldots,x_n,t)), \\
h_2(x_1,\ldots,x_n,t) &= r(k(x_1,\ldots,x_n,t)),
\end{aligned}$$

következik, hogy $\mathcal{C}$ zárt a párhuzamos primitív rekurzióra. □

Most a primitív rekurzió egy másik általánosításával foglalkozunk.

Állítás 7.10 *Legyen $\mathcal{C}$ PRZ-osztály, és legyenek $f_0, f_1 : \mathbf{N}^n \to \mathbf{N}$, $g : \mathbf{N}^{n+3} \to \mathbf{N}$ $\mathcal{C}$-beli függvények. Definiáljuk a $h : \mathbf{N}^{n+1} \to \mathbf{N}$ függvényt az alábbi szabályokkal:*

$$\begin{aligned}
h(x_1,\ldots,x_n,0) &= f_0(x_1,\ldots,x_n) \\
h(x_1,\ldots,x_n,1) &= f_1(x_1,\ldots,x_n) \\
h(x_1,\ldots,x_n,t+2) &= g(t, h(x_1,\ldots,x_n,t), h(x_1,\ldots,x_n,t+1), x_1,\ldots,x_n).
\end{aligned}$$

Ekkor h $\mathcal{C}$-ben van.

Bizonyítás. Legyen $h'(x_1,\ldots,x_n,t) = h(x_1,\ldots,x_n,t+1)$. Ekkor párhuzamos primitív rekurzióval:

$$\begin{aligned}
h(x_1,\ldots,x_n,0) &= f_0(x_1,\ldots,x_n) \\
h'(x_1,\ldots,x_n,0) &= f_1(x_1,\ldots,x_n) \\
h(x_1,\ldots,x_n,t+1) &= h'(x_1,\ldots,x_n,t) \\
h'(x_1,\ldots,x_n,t+1) &= g(t, h(x_1,\ldots,x_n,t), h'(x_1,\ldots,x_n,t), x_1,\ldots,x_n). \quad \Box
\end{aligned}$$

8 A sorozat-függvény

Definíció 8.1 *Legyen $n > 0$, $(a_1,\ldots,a_n) \in \mathbf{N}^n$. Ekkor:*

$$[a_1,\ldots,a_n] = \prod_{i=1}^{n} p_i^{a_i},$$

ahol p_i az i-dik prím. (Emlékeztetőül: $p_0 = 0$, $p_1 = 2, \ldots$)

Állítás 8.2 *A* $[-,\dots,-]:\mathbf{N}^n\longrightarrow\mathbf{N}$ *függvények primitív rekurzívak. Továbbá:*

$$[a_1,\dots,a_n]=[b_1,\dots,b_n]\quad\Leftrightarrow\quad a_1=b_1,\dots,a_n=b_n.$$

Megjegyzés 8.3 Ezzel szemben: $[a_1,\dots,a_n,0]=[a_1,\dots,a_n]$. Azért, hogy ez érvényben maradjon az üres sorozatra is, vagyis az $n=0$ esetben, definiáljuk: $[\,]=1$.

Definíció 8.4 *Legyen* $x\in\mathbf{N}$. *Rögzítsünk egy* $i\in\mathbf{N}$ *számot. Ekkor:*

$$\begin{aligned}(x)_i &= \min_{t\le x}(p_i^{t+1}\not| x)\\ Lt(x) &= \min_{i\le x}((x)_i\neq 0\wedge(\forall j)_{\le x}((x)_j\neq 0\Rightarrow j\le i))\\ &= \min_{i\le x}((x_i)\neq 0\wedge(\forall j)_{\le x}((x)_j=0\vee j\le i)).\end{aligned}$$

Tehát $(x)_0=0$ mindex x-re. Ha $i\geq 1$, akkor $(x)_i$ az a kitevő, amellyel p_i előfordul x prímtényezős felbontásában (ez 0, ha p_i nem fordul elő). $Lt(0)=Lt(1)=0$. Egyébként $Lt(x)$ a legnagyobb olyan i szám, amelyre p_i előfordul x prímtényezős felbontásában.

Állítás 8.5

$$\begin{aligned}([a_1,\dots,a_n])_i &= \begin{cases}a_i & \textit{ha } 1\le i\le n\\ 0 & \textit{különben}\end{cases}\\ [(x)_1,\dots,(x)_n] &= x,\quad \textit{ha } n\geq Lt(x).\end{aligned}$$

Továbbá, a $(-)_i$ *és* Lt *függvények primitív rekurzívak, sőt, a kétváltozós* $(-)_{(-)}:(x,i)\mapsto(x)_i$ *függvény is primitív rekurzív.*

9 Programok Gödel-száma

A fenti segédfüggvényeket felhasználva definiáljuk egy P program $\#(P)$ Gödel számát, amelyből a P program egyértelműen visszanyerhető. Rögzítjük a változók és címkék egy-egy felsorolását:

$$\begin{gathered}Y,X_1,Z_1,X_2,Z_2,\dots\\ A_1,B_1,C_1,A_2,B_2,C_2,\dots\end{gathered}$$

Tehát minden második változó bemenő változó. Legyen V változó, L címke. Ekkor:

1. $\#(V)$: a V sorszáma a változók listáján.
2. $\#(L)$: az L sorszáma a címkék listáján.

(A sorszámozás 1-gyel indul.) Legyen most I utasítás. Ekkor

$$\#(I) \quad = \quad \langle a, \langle b, c \rangle \rangle$$

ahol:

Ha I címkétlen $a = 0$, különben $a = \#(L)$, ahol L az I címkéje.

Ha V az I-ben szereplő változó, akkor $c = \#(V) - 1$.

Ha I a $V \leftarrow V$, $V \leftarrow V + 1$, $V \leftarrow V - 1$ utasítások valamelyike, akkor $b = 0, 1$, illetve 2, ebben a sorrendben.

Ha I az `if` $V \neq 0$ `goto` L utasítás, akkor $b = \#(L) + 2$.

Világos, hogy minden $q \in \mathbf{N}$ számhoz egyértelműen létezik olyan I, amelyre $\#(I) = q$. I megkapható a fenti eljárás megfordításával:

Számítsuk ki $l(q)$-t. Ha $l(q) = 0$, az I utasítás címkétlen, különben a címkéje az $l(q)$-dik címke.

Számítsuk ki az $r(r(q)) + 1$ számot. Ezzel lokalizáljuk a hivatkozott változót, legyen ez V.

Számítsuk ki $l(r(q))$-t. Ha ez $0, 1$ vagy 2, az utasítás megfelelően: $V \leftarrow V$, $V \leftarrow V + 1$, $V \leftarrow V - 1$.

Ha $l(r(q)) > 2$, akkor az utasítás `if` $V \neq 0$ `goto` L, ahol L az $(l(r(q)) - 2)$-dik címke.

Vegyük észre, hogy $I : Y \leftarrow Y$ esetén: $\#(I) = \langle 0, \langle 0, 0 \rangle \rangle = 0$. Végül legyen P egy program, mondjuk P az $I_1, \ldots, I_n$ utasításokból áll. Ekkor:

$$\#(P) \quad = \quad [\#(I_1), \ldots, \#(I_n)] - 1$$

Speciálisan, ha P üres program (azaz $n = 0$), akkor $\#(P) = 0$.

Vegyük észre, hogy minden q számhoz van olyan P, amelyre $\#(P) = q$. Most azonban P egyértelműsége nem biztosított: ha P végéhez csatoljuk az $Y \leftarrow Y$ utasítást, akkor az előálló P'-re: $\#(P) = \#(P')$. Ez azonban nem baj, mert P és P' ekvivalensek. (Itt használjuk a $V \leftarrow V$ alakú utasításokat egyedül!) A továbbiakban ezt az esetet is kizárjuk: csak olyan programokat tekintünk, amelyek nem végződnek $Y \leftarrow Y$ alakú utasítással. Így P egyértelműen visszanyerhető a $\#(P)$ számból. (Speciálisan: az üres P program az egyetlen, melyre $\#(P) = 0$.)

10 Programok megállási problémája

Definíció 10.1 *Tetszőleges $y \in \mathbf{N}$ számra legyen P az a program, amelyre $\#(P) = y$. A* HALT(x, y) *predikátumot definiáljuk az alábbi módon:*

$$\mathrm{HALT}(x, y) \quad \Leftrightarrow \quad \psi_P^{(1)}(x) \downarrow$$

azaz

$$\mathrm{HALT}(x,y) \quad = \quad \begin{cases} 1 & \text{\it ha } \psi_P^{(1)}(x)\downarrow \\ 0 & \text{\it különben.} \end{cases}$$

Tétel 10.2 $\mathrm{HALT}(x,y)$ *nem rekurzív.*

Bizonyítás. Tegyük fel, hogy $\mathrm{HALT}(x,y)$ rekurzív. Akkor $\mathrm{HALT}(x,x)$ is az. Definiáljuk az alábbi $f : \mathbf{N} \to \mathbf{N}$ parciális függvényt:

$$f(x) \quad = \quad \begin{cases} 0 & \text{ha } \sim \mathrm{HALT}(x,x) \\ \uparrow & \text{különben.} \end{cases}$$

$f(x)$ parciális rekurzív: az alábbi P programra $f = \psi_P^{(1)}$:

$$A \quad \texttt{if}\ \mathrm{HALT}(X,X)\ \texttt{goto}\ A$$

Legyen $y_0 = \#(P)$. Látjuk, hogy

$$\mathrm{HALT}(x,y_0) \quad \Leftrightarrow \quad \sim \mathrm{HALT}(x,x)$$

Így $x = y_0$ választással: $\mathrm{HALT}(y_0,y_0) \Leftrightarrow \sim \mathrm{HALT}(y_0,y_0)$, ami ellentmondás. □

Következmény 10.3 *A* $\mathrm{HALT}(x,x)$ *predikátum nem rekurzív.*

Átfogalmazás. Nem létezik olyan algoritmus, amely tetszőleges P programra és x-re eldönti, hogy P végrehajtása befejeződik-e x mellett. Hasonlóan adódik, hogy tetszőleges $n \geq 1$ számra, a

$$\mathrm{HALT}(x_1,\ldots,x_n,y) \quad \Leftrightarrow \quad \psi_P^{(n)}(x_1,\ldots,x_n)\downarrow$$

predikátum nem rekurzív, ahol $y = \#(P)$.

11 Univerzális program

Tetszőleges $n \geq 1$ esetén legyen:

$$\varphi^{(n)}(x_1,\ldots,x_n,y) \quad = \quad \psi_P^{(n)}(x_1,\ldots,x_n)$$

ahol $y = \#(P)$. Meg fogjuk mutatni, hogy $\varphi^{(n)}$ parciális rekurzív, azaz létezik olyan U_n program, amely által kiszámított $n+1$ változós függvény éppen $\varphi^{(n)}$. Az ilyen U_n program univerzális. Úgy működik, mint egy interpreter. Adott $y, x_1, \ldots, x_n$ esetén kiszámítja az y sorszámú P program által kiszámított n-változós függvény értékét az $x_1, \ldots, x_n$ adatokon, azaz szimulálja P

működését az $x_1, \ldots, x_n$ adatokon. Az univerzális program a szimulálandó P pillanatfelvételeit egy K, S változópárban állítja elő: K-utasításszámláló, S-állapot.

$$\begin{aligned} &Z \leftarrow X_{n+1} + 1 \\ &S \leftarrow \prod_{i=1}^{n} (p_{2i})^{X_i} \\ &K \leftarrow 1 \end{aligned}$$

Így ha $P : I_1, \ldots, I_l$, Z értéke $[\#(I_1), \ldots, \#(I_l)]$. S kezdőértéke: $[0, X_1, \ldots, 0, X_n]$, K kezdőértéke 1. Az univerzális program következő utasítása:

$$C \quad \texttt{if } K = Lt(Z) + 1 \vee K = 0 \texttt{ goto } F$$

Ha a szimulációnak vége, ugrás az F cimkére. (A $K = 0$ esetet később magyarázzuk meg.)

$$\begin{aligned} &U \leftarrow r((Z)_K) \\ &P \leftarrow p_{r(U)+1} \end{aligned}$$

$(Z)_K = \langle a, \langle b, c \rangle \rangle$ a K-dik utasítás Gödel száma. Így $U = \langle b, c \rangle$ a végrehajtandó utasítás kódja, $r(U) + 1$ megadja a hivatkozott változó sorszámát, P értéke, a $p_{r(U)+1}$ prímszám, az ennek megfelelő prím. Az $r(U) + 1$-dik változó aktuális értéke a $p_{r(U)+1}$ kitevője S tartalmának prímtényezős felbontásában.

$$\begin{aligned} &\texttt{if } l(U) = 0 \texttt{ goto } N \\ &\texttt{if } l(U) = 1 \texttt{ goto } A \\ &\texttt{if } P \not| S \texttt{ goto } N \\ &\texttt{if } l(U) = 2 \texttt{ goto } M. \end{aligned}$$

Ha $l(U) = 0$, vagyis $b = 0$, a végrahajtandó utasítás az üres $V \leftarrow V$. Ugrás a következő utasítás végrehajtásának előkészítésére. Ha $l(U) = 1$, vagyis $b = 1$, akkor a hivatkozott változó értékét 1-gyel növeljük az A címkén. Ha $l(U) > 1$ de a hivatkozott változó értéke 0, a következő utasítás végrehajtását készítsük elő. Ha $l(U) = 2$, vagyis $b = 2$, és a hivatkozott változó értéke nem 0, akkor ugrás az M címkére, ahol 1-et vonunk ki a hivatkozott változó értékéből. Különben feltételes utasításról van szó és a hivatkozott változó értéke nem 0.

$$\begin{aligned} &K \leftarrow \min_{i \leq Lt(Z)} [l((Z)_i) + 2 = l(U)] \\ &\texttt{goto } C \end{aligned}$$

(Tehát $K = 0$ lesz, ha nincs olyan utasítás, amely címkéje a hivatkozott címke!) A hátra lévő rész:

$$\begin{array}{ll} M & S \leftarrow \lfloor S/P \rfloor \\ & \texttt{goto } N \\ A & S \leftarrow S \cdot P \\ N & K \leftarrow K + 1 \\ & \texttt{goto } C \\ F & Y \leftarrow (S)_1 \end{array}$$

Ezzel beláttuk:

Tétel 11.1 *Tetszőleges $n \geq 1$ esetén, $\varphi^{(n)}$ parciális rekurzív.*

A továbbiakban használjuk a

$$\varphi_y^{(n)}(x_1, \ldots, x_n) \quad = \quad \varphi^{(n)}(x_1, \ldots, x_n, y)$$

jelölést is, és az n felső indexét is gyakran elhagyjuk.

Kis módosítással adódik az alábbi:

Tétel 11.2 *Tetszőleges $n \geq 1$ esetén $\mathrm{STP}^{(n)}(x_1, \ldots, x_n, y, t)$ rekurzív predikátum, ahol*

$$\mathrm{STP}^{(n)}(x_1, \ldots, x_n, y, t) \quad \Leftrightarrow \quad \begin{array}{l} \textit{az } y \textit{ Gödel számú program legfeljebb } t \textit{ lépésben} \\ \textit{megáll az } x_1, \ldots, x_n \textit{ inputon.} \end{array}$$

Megjegyzés 11.3 Belátható, hogy $\mathrm{STP}^{(n)}$ primitív rekurzív predikátum is.

Legyen $\#(P) = y$. Definiáljuk:

$$s(y, x_1, \ldots, x_n, t) \quad = \quad \begin{cases} \text{a } t\text{-dik lépés után végrehajtandó utasítás sorszáma, ha ilyen van} \\ k+1, \text{ különben,} \end{cases}$$

ahol k a P hossza.

$$m(y, x_1, \ldots, x_n, t) \quad = \quad \prod_{i=1}^{\infty} p_i^{\alpha_i},$$

ahol α_i az i-dik változó értéke a t-dik lépés végrehajtása után. Belátjuk, hogy s és m primitív rekurzív függvények. Felhasználjuk az alábbi rövidítéseket. Először is $s(y, x_1, \ldots, x_n, t)$ helyett s-et, $m(y, x_1, \ldots, x_n, t)$ helyett m-et írunk. Továbbá:

$u = (y+1)_s$	(t-dik lépés után végrehajtandó utasítás kódja)
$a = l(u), \ b = l(r(u)), \ c = r(r(u))$	(a t-dik lépés után végrehajtandó utasítás mezőinek kódjai)
$v = c + 1$	(a hivatkozott változó)
$z = b \dot{-} 2$	($b \geq 3$ esetén a hivatkozott címke)

Az $s(y, x_1, \ldots, x_n, t)$ és $m(y, x_1, \ldots, x_n, t)$ meghatározására párhuzamos primitív rekurziót alkalmazunk.

$$\begin{aligned} s(y, x_1, \ldots, x_n, 0) &= 1 \\ m(y, x_1, \ldots, x_n, 0) &= \prod_{i=1}^{n} p_{2i}^{x_i} \\ s(y, x_1, \ldots, x_n, t+1) &= \begin{cases} s+1 & \text{ha } b \leq 2 \ \vee \ (p_v \nmid m) \\ r = \min_{i \leq Lt(y+1)}[l((y+1)_i) = z] & \text{ha } b > 2 \ \wedge \ (p_v | m) \ \wedge \ r \neq 0 \\ Lt(y+1) + 1 & \text{különben} \end{cases} \\ m(y, x_1, \ldots, x_n, t+1) &= \begin{cases} m & \text{ha } b = 0 \ \vee \ (b = 2 \ \wedge \ (p \nmid m)) \ \vee \ b > 2 \\ m \cdot p_v & \text{ha } b = 1 \\ m / p_v & \text{különben} \end{cases} \end{aligned}$$

Ezek után:

$$\mathrm{STP}(x_1,\dots,x_n,y,t) \quad\Leftrightarrow\quad (\exists r)_{\le t}[s(y,x_1,\dots,x_n,r) = Lt(y+1)+1].$$

12 Rekurzív és rekurzívan felsorolható halmazok

Definíció 12.1 *Legyen $A \subseteq \mathbf{N}$. Az A halmaz karakterisztikus függvénye a*

$$P_A(x) \quad = \quad \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$$

predikátum. A rekurzív, ha karakterisztikus függvénye rekurzív. A rekurzívan felsorolható, ha van olyan $f : \mathbf{N} \longrightarrow \mathbf{N}$ parciális rekurzív függvény, amelynek A az értelmezési tartománya. Jelölés: $A = \{x \in \mathbf{N} | f(x)\downarrow\}$.

Állítás 12.2 *Legyen $A \subseteq \mathbf{N}$. Az A halmaz akkor és csak akkor rekurzívan felsorolható, ha*

$$A \quad = \quad \{x \in \mathbf{N} | P(x) = 1\} \tag{1}$$

valamely $P(x)$ parciális rekurzív predikátumra, vagy

$$A \quad = \quad \{x \in \mathbf{N} | f(x) = k\} \tag{2}$$

valamely $f(x)$ parciális rekurzív függvényre és k számra.

Bizonyítás. Ha $A = \{x | f(x)\downarrow\}$, ahol f parciális rekurzív függvény, akkor $A = \{x | P(x) = 1\}$, ahol $P(x)$ a $h(f(x))$ parciális rekurzív predikátum. Itt $h(x)$ a konstans 1 értékű függvény. Fordítva, ha A megadható (2) alakban, akkor nyilván megadható (1) alakban is. Továbbá, ha (1) érvényes, ahol $P(x)$ parciális rekurzív predikátum, akkor legyen $Q(x)$ az azonosan igaz, egyváltozós predikátum. Ekkor $A = \{x | Q(P(x))\downarrow\}$. □

Állítás 12.3 *Minden véges $A \subseteq \mathbf{N}$ rekurzív. A rekurzív halmazok zártak az egyesítés, metszet és komplementáció műveleteire.*

Bizonyítás. Legyen A véges, $A = \{a_1,\dots,a_k\}$. Az A karakterisztikus függvénye az

$$x = a_1 \vee x = a_2 \vee \ldots \vee x = a_k$$

primitív rekurzív predikátum. Legyenek A, B rekurzívak a P_A és P_B karakterisztikus függvényekkel. Ekkor

$$\begin{array}{rcl} P_A \wedge P_B & \text{az} & A \cap B \\ P_A \vee P_B & \text{az} & A \cup B \\ \sim P_A & \text{az} & \overline{A} \end{array}$$

karakterisztikus függvénye. □

Tétel 12.4 *Legyen $A \subseteq \mathbf{N}$. Az A halmaz akkor és csak akkor rekurzív, ha A és $\overline{A}$ is rekurzívan felsorolható.*

Bizonyítás. Szükségesség. Ha A rekurzív, akkor $\overline{A}$ is rekurzív. Így elegendő belátnunk, hogy ha A rekurzív, akkor A rekurzívan felsorolható. Ez nyilvánvaló a 12.2 Állításból.

Elegendőség. Legyen

$$\begin{aligned} A &= \{x \in \mathbf{N} | f(x) \downarrow\} \\ \overline{A} &= \{x \in \mathbf{N} | g(x) \downarrow\}, \end{aligned}$$

ahol

$$\begin{aligned} f(x) &= \varphi^{(1)}(x, p) \\ g(x) &= \varphi^{(1)}(x, q). \end{aligned}$$

Tekintsük az alábbi programot:

$$\begin{array}{ll} L & \texttt{if } \mathrm{STP}^{(1)}(X, p, T) \texttt{ goto } F \\ & \texttt{if } \mathrm{STP}^{(1)}(X, q, T) \texttt{ goto } E \\ & T \leftarrow T + 1 \\ & \texttt{goto } L \\ F & Y \leftarrow 1 \end{array}$$

E program által kiszámított 1-változós függvény éppen A karakterisztikus függvénye. □

Állítás 12.5 *Ha A és B rekurzívan felsorolható, akkor $A \cup B$ és $A \cap B$ is az.*

Bizonyítás. Legyen

$$\begin{aligned} A &= \{x | f(x) \downarrow\} \\ B &= \{x | g(x) \downarrow\}, \end{aligned}$$

ahol

$$\begin{aligned} f(x) &= \varphi^{(1)}(x, p) \\ g(x) &= \varphi^{(1)}(x, q). \end{aligned}$$

Tekintsük az alábbi két R és R' programot:

$$\begin{array}{lcll} Y \leftarrow f(X) & \qquad & L & \texttt{if } \mathrm{STP}^{(1)}(X, p, T) \texttt{ goto } E \\ Y \leftarrow g(X) & & & \texttt{if } \mathrm{STP}^{(1)}(X, q, T) \texttt{ goto } E \\ & & & T \leftarrow T + 1 \\ & & & \texttt{goto } L \end{array}$$

Ekkor

$$\begin{aligned} A \cap B &= \{x | \psi_R(x) \downarrow\} \\ A \cup B &= \{x | \psi_{R'}(x) \downarrow\} \quad \Box \end{aligned}$$

Definíció 12.6 $W_n = \{x \in \mathbf{N} | \varphi^{(1)}(x, n) \downarrow\} = \{x \in \mathbf{N} | \varphi_n(x) \downarrow\}$.

Világos, hogy egy $A \subseteq \mathbf{N}$ halmaz akkor és csakis akkor rekurzívan felsorolható, ha $A = W_n$ valamely n-re.

Tétel 12.7 *Legyen $K = \{n \in \mathbf{N} | n \in W_n\}$. K rekurzívan felsorolható, de nem rekurzív.*

Bizonyítás. Világos, hogy

$$n \in K \quad \Leftrightarrow \quad \mathrm{HALT}(n, n).$$

Így K nem rekurzív, hiszen a $\mathrm{HALT}(x, x)$ perdikátum nem rekurzív. Másrészt

$$K = \{n | \varphi^{(1)}(n, n) \downarrow\}$$

és $\varphi^{(1)}(n, n)$ parciális rekurzív. $\Box$

Következmény 12.8 *A rekurzívan felsorolható halmazok osztálya nem zárt a komplementer képzésre.*

Tétel 12.9 *Legyen $A \subseteq \mathbf{N}$, $A \neq \emptyset$. Az alábbi állítások ekvivalensek.*

(i) Létezik olyan $f : \mathbf{N} \to \mathbf{N}$ primitív rekurzív függvény, hogy $A = \{f(x) | x \in \mathbf{N}\}$.

(ii) Létezik olyan $f : \mathbf{N} \to \mathbf{N}$ rekurzív függvény, hogy $A = \{f(x) | x \in \mathbf{N}\}$.

(iii) Létezik olyan $f : \mathbf{N} \to \mathbf{N}$ parciális rekurzív függvény, hogy $A = \{f(x) | x \in \mathbf{N}\}$.

(iv) A rekurzívan felsorolható.

Következmény 12.10 *Legyen $A \subseteq \mathbf{N}$. A akkor és csakis akkor rekurzívan felsorolható, ha egy parciális rekurzív függvény értékkészlete.*

Bizonyítás. (iv) $\Rightarrow$ (i): Tegyük fel, hogy $A = W_p$. Definiáljuk:

$$f(z) = \begin{cases} l(z) & \text{ha } \mathrm{STP}(l(z), p, r(z)) \\ x_0 & \text{különben,} \end{cases}$$

ahol x_0 az A rögzített eleme.

(iii) $\Rightarrow$ (iv): Legyen $f = \varphi_p$. Megadunk egy olyan Q programot, hogy A a $g = \psi_Q^{(1)}$ parciális rekurzív függvény értelmezési tartománya.

$$\begin{array}{ll} L & \texttt{if } \mathrm{STP}(l(T), p, r(T)) \texttt{ goto } F \\ & T \leftarrow T + 1 \\ & \texttt{goto } L \\ F & Z \leftarrow f(l(T)) \\ & \texttt{if } X_1 \neq Z \texttt{ goto } L \end{array}$$

Ekkor

$$g(x) = \begin{cases} 0 & \text{ha } \exists u \ f(u) = x \\ \uparrow & \text{különben.} \end{cases} \qquad \Box$$

Az előző eredményhez hasonlóan igazolható az alábbi tétel is.

Tétel 12.11 *Legyen $A \subseteq \mathbf{N}$ végtelen halmaz. Az alábbi állítások ekvivalensek:*

(i) Létezik olyan szigorúan monoton növő rekurzív függvény, hogy $A = \{f(x) | x \in \mathbf{N}\}$.

(ii) A rekurzív.

Megjegyzés 12.12 A fenti tétel nem igaz akkor, ha az (i) feltételben rekurzív függvény helyett primitív rekurzív függvényt mondunk.

A rekurzívan felsorolható, illetve rekurzív halmaz fogalma kiterjeszthető az $\mathbf{N}^n$ részhalmazaira, ahol $n \geq 2$ tetszőleges. Az $\mathbf{N}^n$ rekurzívan felsorolható részhalmazait rekurzívan felsorolható relációknak, rekurzív részhalmazait pedig rekurzív relációknak is nevezzük. Definíció szerint $A \subseteq \mathbf{N}^n$ rekurzívan felsorolható, ha létezik olyan $f(x_1, \ldots, x_n)$ parciális rekurzív függvény, hogy

$$A = \{(x_1, \ldots, x_n) | f(x_1, \ldots, x_n) \downarrow\},$$

és A rekurzív, ha karakterisztikus függvénye rekurzív. A rekurzívan felsorolható illetve a rekurzív relációkra a rekurzívan felsorolható illetve rekurzív halmazokra vonatkozó tételek megfelelői érvényesek. Ezek származtathatóak korábbi eredményeinkből, hiszen érvényes az alábbi:

Állítás 12.13 *Legyen $A \subseteq \mathbf{N}^n$. Definiáljuk*

$$B = \{\langle x_1, \ldots, x_n \rangle | (x_1, \ldots, x_n) \in A\}.$$

(i) A akkor és csak akkor rekurzívan felsorolható, ha B rekurzívan felsorolható.

(ii) A akkor és csak akkor rekurzív, ha B rekurzív.

13 Normál forma tételek (Kleene)

Tétel 13.1 *Legyen $A \subseteq \mathbf{N}$. Az A halmaz akkor és csak akkor rekurzívan felsorolható, ha van olyan $R(x,t)$ (primitív) rekurzív predikátum, hogy*

$$A = \{x \mid (\exists t)R(x,t)\}. \tag{3}$$

Bizonyítás. Legyen $R(x,t) = \mathrm{STP}(x, y_0, t)$, ahol $A = W_{y_0}$. Fordítva, ha R olyan rekurzív predikátum, hogy (3) érvényes, akkor

$$A = \{x \mid f(x)\downarrow\},$$

ahol $f(x) = \min_y R(x,y)$ parciális rekurzív függvény. □

Következmény 13.2 *Valamely $A \subseteq \mathbf{N}^n$ akkor és csakis akkor rekurzívan felsorolható reláció, ha létezik olyan $R(x_1,\ldots,x_n,t)$ (primitív) rekurzív predikátum, hogy*

$$A = \{(x_1,\ldots,x_n) \mid (\exists t)R(x_1,\ldots,x_n,t)\}.$$

Bizonyítás. Tegyük fel, hogy A rekurzívan felsorolható. Ekkor a

$$B = \{\langle x_1,\ldots,x_n\rangle \mid (x_1,\ldots,x_n) \in A\}$$

halmaz is rekurzívan felsorolható. Így

$$B = \{x \mid (\exists t)R(x,t)\}$$

valamely $R(x,t)$ primitív rekurzív predikátumra. Tehát

$$A = \{(x_1,\ldots,x_n) \mid (\exists t)R(\langle x_1,\ldots,x_n\rangle, t)\}.$$

Fordítva, ha

$$A = \{(x_1,\ldots,x_n) \mid (\exists t)R(x_1,\ldots,x_n,t)\},$$

ahol R (primitív) rekurzív predikátum, akkor az előzőekben definiált B halmazra

$$B = \{x \mid (\exists t)R((x)_1,\ldots,(x)_n,t)\}.$$

Így az előző tétel szerint B, és ezért A is, rekurzívan felsorolható. □

Következmény 13.3 *Legyen R $m+n$-változós rekurzív predikátum. Ekkor az*

$$A = \{(x_1,\ldots,x_m) \mid (\exists y_1)\ldots(\exists y_n)R(x_1,\ldots,x_m,y_1,\ldots,y_n)\}$$

reláció rekurzívan felsorolható.

Bizonyítás. A bizonyítást $m = 1$-re végezzük el. Legyen

$$S(x,z) \quad \Leftrightarrow \quad R(x,(z)_1,\ldots,(z)_n).$$

Ekkor $S(x,z)$ rekurzív, és

$$A \quad = \quad \{x|(\exists z)S(x,z)\},$$

tehát A rekurzívan felsorolható. $\square$

Következmény 13.4 *Ha $A \subseteq \mathbf{N}^{n+1}$ rekurzívan felsorolható, akkor*

$$B \quad = \quad \{(x_1,\ldots,x_n)|(\exists t)(x_1,\ldots,x_n,t) \in A\}$$

is rekurzívan felsorolható.

Tétel 13.5 *Az $f : \mathbf{N}^n \to \mathbf{N}$ parciális függvény akkor és csakis akkor parciális rekurzív, ha az*

$$A \quad = \quad \{(x_1,\ldots,x_n,y)|y = f(x_1,\ldots,x_n)\}$$

reláció, vagy ami azzal ekvivalens, a

$$B \quad = \quad \{\langle x_1,\ldots,x_n,y\rangle|y = f(x_1,\ldots,x_n)\}$$

halmaz rekurzívan felsorolható.

Bizonyítás. Tegyük fel, hogy f parciális rekurzív. Akkor a

$$g(z) \quad = \quad f((z)_1,\ldots,(z)_n)$$

függvény és a

$$\langle z, g(z)\rangle$$

függvény is parciálisan rekurzív. Másrészt

$$B \quad = \quad \{\langle z, g(z)\rangle|z \in \mathbf{N}\}.$$

(Most az

$$\langle x_1,\ldots,x_n\rangle \quad = \quad \langle\ldots\langle\langle x_1,x_2\rangle,x_3\rangle,\ldots,x_n\rangle$$

definíciót használjuk.) Fordítva, tegyük fel, hogy A rekurzívan felsorolható. Ekkor létezik olyan $R(x_1,\ldots,x_n,y,t)$ primitív rekurzív predikátum, hogy

$$A \quad = \quad \{(x_1,\ldots,x_n,y)|(\exists t)R(x_1,\ldots,x_n,y,t)\}.$$

Ekkor

$$f(x_1,\ldots,x_n) \quad = \quad l(\min_z R(x_1,\ldots,x_n,l(z),r(z)). \quad \square$$

Az előző tétel bizonyítása közben beláttuk az alábbi norma forma tételt is:

Tétel 13.6 *Az $f : \mathbf{N}^n \to \mathbf{N}$ parciális függvény akkor és csakis akkor parciális rekurzív, ha valamely $R(x_1, \ldots, x_n, z)$ primitív rekurzív predikátumra*

$$f(x_1, \ldots, x_n) \quad = \quad l(\min_z R(x_1, \ldots, x_n, z)).$$

Következmény 13.7 *A parciális rekurzív függvények osztálya a leszűkebb olyan függvényosztály, amely tartalmazza az alapfüggvényeket és zárt a függvényösszetételre, a primitív rekurzióra és a minimalizálásra.*

14 A paraméter (vagy s-m-n) tétel

A paraméter tétel egy olyan fontos technikai tétel, amely a $\varphi^{(n)}(x_1, \ldots, x_n, y)$ függvények között állít fel összefüggést.

Tétel 14.1 *Bármely $n, m \geq 1$ számokhoz létezik olyan S^n_m primitív rekurzív függvény, amelyre:*

$$\varphi^{(m+n)}(x_1, \ldots, x_m, u_1, \ldots, u_n, y) \quad = \quad \varphi^{(m)}(x_1, \ldots, x_m, S^n_m(u_1, \ldots, u_n, y)).$$

Bizonyítás. Rögzített $u_1, \ldots, u_n, y$ mellett

$$\varphi^{m+n}(x_1, \ldots, x_m, u_1, \ldots, u_n, y)$$

m-változós parciális rekurzív függvény. Van tehát olyan P program, amely ezt számítja ki. Legyen $u = S^n_m(u_1, \ldots, u_n, y)$ egy ilyen program Gödel száma, ekkor a fenti összefüggés fennáll. Ez az okoskodás azonban nem ad primitív rekurzív függvényt (sőt, még rekurzívat sem). A tételben éppen az az érdekes, hogy S^n_m választható primitív rekurzívnak. A bizonyítást n szerinti teljes indukcióval végezzük.

$n = 1$. Be kell látnunk, hogy

$$\varphi^{(m+1)}(x_1, \ldots, x_m, u, y) \quad = \quad \varphi^{(m)}(x_1, \ldots, x_m, S^1_m(u, y))$$

valamely S^1_m primitív rekurzív függvényre. Adott u és y mellett legyen P az a program, amelyre $\#(P) = y$. Készítsük el az alábbi Q programot:

$$\begin{array}{l} \left.\begin{array}{l} X_{m+1} \leftarrow X_{m+1} + 1 \\ \qquad\vdots \\ X_{m+1} \leftarrow X_{m+1} + 1 \end{array}\right\} \; u\text{−szor} \\ P \end{array}$$

Világos, hogy $\psi^{(m)}_Q$ az az m-változós parciális függvény, amelyet úgy kapunk, hogy $\varphi^{(m+1)}$ utolsó két változóját rögzítjük u-val illetve y-nal. Ha tehát $S^1_m(u, y) = \#(Q)$, fennáll a kívánt összefüggés.

Ha $\#(Q)$ primitív rekurzív függvénnyel kapható u-ból és $y = \#(P)$-ből, kész vagyunk. De:

$$\begin{aligned}
\#(Q) &= [\overbrace{\langle 0, \langle 1, 2m+1\rangle\rangle, \ldots, \langle 0, \langle 1, 2m+1\rangle\rangle}^{u-\text{szor}}, (y+1)_1, \ldots, (y+1)_{Lt(y+1)}] \dot{-} 1 \\
&= \prod_{j=1}^{u} p_j^{\langle 0, \langle 1, 2m+1\rangle\rangle} \cdot \prod_{j=1}^{Lt(y+1)} p_{u+j}^{(y+1)_j} \dot{-} 1.
\end{aligned}$$

Legyen most $n \geq 1$, és tegyük fel, hogy az állítás igaz n-re.

$$\begin{aligned}
\varphi^{(m+n+1)}(x_1, \ldots, x_m, u_1, \ldots, u_{n+1}, y) &= \varphi^{(m+n)}(x_1, \ldots, x_m, u_1, \ldots, u_n, S^1_{m+n}(u_{n+1}, y)) \\
&= \varphi^{(m)}(x_1, \ldots, x_m, S^n_m(u_1, \ldots, u_n, S^1_{m+n}(u_{n+1}, y))) \\
&= \varphi^{(m)}(x_1, \ldots, x_m, S^{n+1}_m(u_1, \ldots, u_{n+1}, y)). \quad \square
\end{aligned}$$

Megjegyzés 14.2 Az S^n_m függvények választhatóak injektívnek is. Tekintsük az $m = n = 1$ esetet. Legyenek adottak az u és y számok, és legyen P azon program, amelyre $\#(P) = y$. Ekkor definíciónk szerint $S^1_1(u, y) = \#(Q)$, ahol Q az alábbi program:

$$\begin{array}{l}
\left.\begin{array}{l} X_2 \leftarrow X_2 + 1 \\ \quad\vdots \\ X_2 \leftarrow X_2 + 1 \end{array}\right\} u-\text{szor} \\
P
\end{array}$$

Módosítsuk Q-t egy $X_2 \leftarrow X_2$ utasítás beillesztésével az alábbi módon:

$$\begin{array}{l}
\left.\begin{array}{l} X_2 \leftarrow X_2 + 1 \\ \quad\vdots \\ X_2 \leftarrow X_2 + 1 \end{array}\right\} u-\text{szor} \\
X_2 \leftarrow X_2 \\
P
\end{array}$$

Megjegyzés 14.3 A paraméter tétel alkalmazásáról. A paraméter tételt gyakran használjuk bizonyításokban a következő alakban: Legyen

$$h(x_1, \ldots, x_m, y_1, \ldots, y_n)$$

parciális rekurzív függvény. Ekkor létezik olyan $f(y_1, \ldots, y_n)$ injektív primitív rekurzív függvény, hogy

$$\varphi^{(m)}_{f(y_1,\ldots,y_n)}(x_1, \ldots, x_m) = h(x_1, \ldots, x_m, y_1, \ldots, y_n).$$

Valóban, ha

$$h(x_1, \ldots, x_m, y_1, \ldots, y_n) = \varphi^{(m+n)}(x_1, \ldots, x_m, y_1, \ldots, y_n, e),$$

akkor legyen

$$f(y_1, \ldots, y_n) = S^n_m(y_1, \ldots, y_n, e).$$

15 A rekurziós tétel (Kleene)

Tekintsünk egy

$$f(x_1, \ldots, x_n) = G(f, x_1, \ldots, x_n) \tag{4}$$

fixpont egyenletet, ahol G adott „funkcionál". Amennyiben f megoldás, azt is mondjuk, hogy (4) implicite definiálja f-et. Persze itt nem közönséges definícióról van szó, hanem olyan f-et szeretnénk találni, amely megoldása a (4) egyenletnek. Amennyiben $G(f, x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$, minden n-változós f függvény megoldás. Amennyiben $G(f, x_1, \ldots, x_n) = f(x_1, \ldots, x_n) + 1$, csak a seholsem definiált függvény megoldás. Azt szeretnénk belátni, hogy amennyiben G parciális rekurzív, úgy (4) mindig megoldható a parciális rekurzív függvények között. Ehhez azonban az egyenletben f-et indexével kell helyettesítenünk. Így jutunk a következő tételhez:

Tétel 15.1 *Legyen g $m+1$-változós parciális rekurzív függvény. Létezik olyan e szám, hogy*

$$\varphi_e^{(m)}(x_1, \ldots, x_m) = g(e, x_1, \ldots, x_m).$$

Bizonyítás. Minden rögzített e-re $g(e, x_1, \ldots, x_m)$ parciális rekurzív függvény, így létezik olyan e', hogy

$$g(e, x_1, \ldots, x_m) = \varphi_{e'}^{(m)}(x_1, \ldots, x_m).$$

A nemtriviális dolog az, hogy létezik olyan e, amelyre $e' = e$.

Tekintsük az alábbi h függvényt:

$$h(x_1, \ldots, x_m, v) = g(S_m^1(v, v), x_1, \ldots, x_m).$$

Mivel h $m+1$-változós parciális rekurzív függvény, ezért létezik olyan z_0, amelyre

$$h(x_1, \ldots, x_m, v) = \varphi^{(m+1)}(x_1, \ldots, x_m, v, z_0).$$

Így:

$$\begin{aligned} g(S_m^1(v, v), x_1, \ldots, x_m) &= \varphi^{(m+1)}(x_1, \ldots, x_m, v, z_0) \\ &= \varphi^{(m)}(x_1, \ldots, x_m, S_m^1(v, z_0)). \end{aligned}$$

A $v = z_0$ választással az $e = S_m^1(z_0, z_0)$ számra adódik a kívánt összefüggés. □

Megjegyzés 15.2 Ha adott a g indexe (azaz $g = \varphi_{e_0}^{(m+1)}$), akkor az e index az e_0 injektív primitív rekurzív függvénye. Valóban:

$$h(x_1, \ldots, x_m, v) = \varphi_{e_0}^{(m+1)}(S_m^1(v, v), x_1, \ldots, x_m).$$

A jobboldal tekinthető $e_0, v, x_1, \ldots, x_m$ parciális rekurzív függvényként. Így, a paraméter tétel szerint létezik olyan $f(e_0)$ injektív primitív rekurzív függvény, hogy

$$\varphi_{e_0}^{(m+1)}(S_m^1(v,v), x_1, \ldots, x_m) \quad = \quad \varphi_{f(e_0)}^{(m+1)}(x_1, \ldots, x_m, v).$$

Az előző jelölésekkel $z_0 = f(e_0)$. Ezek után

$$e \;=\; S_m^1(z_0, z_0) \;=\; S_m^1(f(e_0), f(e_0))$$

az e_0 injektív primitív rekurzív függvénye.

Érvényes a rekurziós tétel alábbi általánosítása is:

Tétel 15.3 *Legyen $g(v, y_1, \ldots, y_n, x_1, \ldots, x_m)$ parciális rekurzív függvény. Ekkor létezik egy olyan $e(y_1, \ldots, y_n)$ injektív primitív rekurzív függvény, hogy*

$$g(e(y_1, \ldots, y_n), y_1, \ldots, y_n, x_1, \ldots, x_m) \quad = \quad \varphi_{e(y_1,\ldots,y_n)}^{(m)}(x_1, \ldots, x_m).$$

Bizonyítás. Legyen

$$h(x_1, \ldots, x_m, y_1, \ldots, y_n, v) \quad = \quad g(S_m^{n+1}(y_1, \ldots, y_n, v, v), y_1, \ldots, y_n, x_1, \ldots, x_m).$$

Létezik olyan z_0 szám, amelyre:

$$\begin{aligned} h(x_1, \ldots, x_m, y_1, \ldots, y_n, v) \quad &= \quad \varphi^{(m+n+1)}(x_1, \ldots, x_m, y_1, \ldots, y_n, v, z_0) \\ &= \quad \varphi_{S_m^{n+1}(y_1,\ldots,y_n,v,z_0)}^{(m)}(x_1, \ldots, x_m). \end{aligned}$$

Legyen

$$e(y_1, \ldots, y_n) \quad = \quad S_m^{n+1}(y_1, \ldots, y_n, z_0, z_0). \quad \square$$

Megjegyzés 15.4 Csakúgy mint korábban, ha adott a g e_0 indexe, $e(y_1, \ldots, y_n)$ tekinthető az $e_0, y_1, \ldots, y_n$ injektív primitív rekurzív függvénynek is.

Példa 15.5 Létezik olyan e, hogy: $\varphi_e(x) = e$.

Valóban, legyen $g(z, x) = u_1^2(z, x) = z$. A rekurziós tétel szerint létezik olyan e, hogy $\varphi_e(x) = g(e, x) = e$.

Példa 15.6 Legyen g 1-változós, h pedig 3-változós függvény. Definiáljuk az f függvényt az alábbi szabályokkal:

$$\begin{aligned} f(x, 0) \quad &= \quad g(x) \\ f(x, t+1) \quad &= \quad h(t, f(2x, t), x) \end{aligned}$$

Amennyiben g és h rekurzívak, h is az. Ez az alábbi módon látható be egyszerűen. Legyen

$$k(y, x, t) \quad = \quad \begin{cases} g(x) & \text{ha } t = 0 \\ h(t-1, \varphi_y(2x, t-1), x) & \text{ha } t > 0. \end{cases}$$

A rekurziós tétel szerint van olyan e, hogy $\varphi_e(x, t) = k(e, x, t)$. De $\varphi_e = f$, így f rekurzív.

16 A fixpont tétel (Kleene)

Tétel 16.1 *Legyen $f : \mathbf{N} \to \mathbf{N}$ rekurzív függvény. Ekkor létezik olyan e, amelyre $\varphi_e(x) = \varphi_{f(e)}(x)$.*

Bizonyítás. Legyen $g(z, x) = \varphi_{f(z)}(x)$. A rekurziós tétel szerint létezik olyan e, hogy

$$\varphi_e(x) \;=\; g(e, x) \;=\; \varphi_{f(e)}(x).$$

□

A tétel általánosabban is kimondható.

Tétel 16.2 *Legyen $f(u, y_1, \ldots, y_n)$ rekurzív függvény, $m \geq 1$. Létezik olyan $e(y_1, \ldots, y_n)$ injektív primitív rekurzív függvény, hogy*

$$\varphi^{(m)}_{f(e(y_1,\ldots,y_n),y_1,\ldots,y_n)} \;=\; \varphi^{(m)}_{e(y_1,\ldots,y_n)}.$$

Bizonyítás. A bizonyítást csak $m = 1$-re végezzük el. Legyen

$$g(u, y_1, \ldots, y_n, x) \;=\; \varphi_{f(u,y_1,\ldots,y_n)}(x).$$

A rekurziós tétel szerint létezik az $e(y_1, \ldots, y_n)$ injektív primitív rekurzív függvény úgy, hogy

$$\begin{aligned}\varphi_{e(y_1,\ldots,y_n)}(x) &= g(e(y_1, \ldots, y_n), y_1, \ldots, y_n, x) \\ &= \varphi_{f(e(y_1,\ldots,y_n),y_1,\ldots,y_n)}(x). \quad \square\end{aligned}$$

Megjegyzés 16.3 Az előző két tétel akkor is érvényes, ha f parciális rekurzív függvény. Ha a 16.1 Tételben f a sehol sem definiált függvény, akkor e olyan program indexe, amely soha nem áll meg. Hasonlóan, $\varphi_{f(e)}(x) = \varphi(x, f(e))$ az x változó egyetlen értékére sem definiált.

Megjegyzés 16.4 Amennyiben f az e_0 indexével adott, $e(y_1, \ldots, y_n)$ tekinthető az e_0 függvényének is, a tétel érvényben marad. Ennek az az oka, hogy g valamely indexe az f indexének injektív primitív rekurzív függvénye. Valóban:

$$g(u, y_1, \ldots, y_n, x) \;=\; \varphi_{\varphi_{e_0}(u,y_1,\ldots,y_n)}(x),$$

a jobboldal az $e_0, u, y_1, \ldots, y_n, x$ parciális rekurzív függvénye. A paraméter tétel miatt akkor létezik olyan $p(e_0)$ injektív primitív rekurziós függvény, hogy

$$\varphi_{p(e_0)}(u, y_1, \ldots, y_n, x) \;=\; \varphi_{\varphi_{e_0}(u,y_1,\ldots,y_n)}(x).$$

Megjegyzés 16.5 A fixpont tétel, és az általunk rekurziós tételnek nevezett eredmények lényegében ekvivalensek, és a fixpont tételt is szokás rekurziós tételnek nevezni. Megmutatjuk, hogyan nyerhető a rekurziós tétel a fixpont tételből. Legyen $g(v, y_1, \ldots, y_n, x_1, \ldots, x_m)$ parciális rekurzív függvény. A paraméter tétel szerint létezik olyan $f(v, y_1, \ldots, y_n)$ injektív primitív rekurzív függvény, hogy

$$\varphi^{(m)}_{f(v,y_1,\ldots,y_n)}(x_1, \ldots, x_m) \quad = \quad g(v, y_1, \ldots, y_n, x_1, \ldots, x_m).$$

A fixpont tétel szerint létezik olyan $e(y_1, \ldots, y_n)$ injektív primitív rekurzív függvény, hogy

$$\varphi^{(m)}_{f(e(y_1,\ldots,y_n),y_1,\ldots,y_n)}(x_1, \ldots, x_m) \quad = \quad \varphi^{(m)}_{e(y_1,\ldots,y_n)}(x_1, \ldots, x_m).$$

Így:

$$\varphi^{(m)}_{e(y_1,\ldots,y_n)}(x_1, \ldots, x_m) \quad = \quad g(e(y_1, \ldots, y_n), y_1, \ldots, y_n, x_1, \ldots, x_m).$$

17 Erős reducibilitási fogalmak

Legyenek A és B az $\mathbf{N}$ részhalmazai, és legyen $f : \mathbf{N} \to \mathbf{N}$ függvény. Amennyiben tetszőleges x-re

$$x \in A \quad \Leftrightarrow \quad f(x) \in B,$$

az $A \leq_f B$ jelölést használjuk.

Definíció 17.1 *(i) $A \leq_1 B$ akkor és csakis akkor, ha létezik olyan f injektív rekurzív függvény, hogy $A \leq_f B$.*

(ii) $A \leq_m B$ akkor és csakis akkor, ha létezik olyan f rekurzív függvény, hogy $A \leq_f B$.

(iii) $A \equiv_1 B$ akkor és csakis akkor, ha $A \leq_1 B$ és $B \leq_1 A$.

(iv) $A \equiv_m B$ akkor és csakis akkor, ha $A \leq_m B$ és $B \leq_m A$.

Megjegyzés 17.2 Amennyiben $A \leq_1 B$, akkor $A \leq_m B$ is teljesül. A fordított irányú állítás nem igaz.

Megjegyzés 17.3 Legyen $r = 1, m$. Ekkor $A \leq_r B$ és $B \leq_r C$ esetén $A \leq_r C$. Ha $A \equiv_r B$ és $B \equiv_r C$, akkor $A \equiv_r C$. Továbbá $A \leq_r A$, $A \equiv_r A$, és ha $A \equiv_r B$ akkor $B \equiv_r A$, valamint $A \leq_r B$ akkor és csak akkor, ha $\overline{A} \leq_r \overline{B}$.

Állítás 17.4 *Tegyük fel, hogy $A \leq_m B$. Ha B rekurzív, akkor A is rekurzív. Ha B rekurzívan felsorolható, akkor A is rekurzívan felsorolható.*

Bizonyítás. Tegyük fel, hogy $A \leq_f B$, ahol f rekurzív függvény. Ha P_B jelöli a B karakterisztikus függvényét, akkor az A karakterisztikus függvényére fennáll a

$$P_A(x) = P_B(f(x))$$

összefüggés. Így ha P_B rekurzív, P_A is rekurzív. Ha B a g parciális rekurzív függvény értelmezési tartománya, akkor

$$A = \{x | g(f(x)) \downarrow\}$$

rekurzívan felsorolható. □

18 Rekurzív permutációk és Myhill izomorfizmus tétele

Definíció 18.1 *Rekurzív permutáció egy olyan $f : \mathbf{N} \longrightarrow \mathbf{N}$ rekurzív függvény, amely bijekció.*

Állítás 18.2 *A rekurzív permutációk csoportot alkotnak.*

Bizonyítás. Legyen f rekurzív permutáció. Ekkor az f

$$g(x) = \min_y [f(y) = x]$$

inverz függvénye is rekurzív. □

Definíció 18.3 *Legyenek $A, B \subseteq \mathbf{N}$. Azt mondjuk, hogy A és B rekurzívan izomorfak, $A \equiv B$, ha létezik olyan $f : \mathbf{N} \longrightarrow \mathbf{N}$ rekurzív permutáció, hogy $f(A) = B$.*

Világos, hogy $A \equiv B$ esetén $A \equiv_1 B$. A fordított irányú állítás is érvényes.

Tétel 18.4 *Legyenek $A, B \subseteq \mathbf{N}$. Ekkor $A \equiv B \Leftrightarrow A \equiv_1 B$.*

Bizonyítás. Tegyük fel, hogy $f, g : \mathbf{N} \longrightarrow \mathbf{N}$ olyan injektív rekurzív függvények, amelyekre:

$$\begin{aligned} x \in A &\Leftrightarrow f(x) \in B \\ y \in B &\Leftrightarrow g(y) \in A. \end{aligned}$$

Előállítunk az f és g felhasználásával egy olyan $h : \mathbf{N} \longrightarrow \mathbf{N}$ bijektív rekurzív függvényt, amelyre

$$x \in A \Leftrightarrow h(x) \in B.$$

A h függvényt

$$h = \bigcup_{s=0}^{\infty} h_s$$

alakban keressük, ahol mindegyik h_s injektív parciális függvény, mely értelmezési tartománya véges. Továbbá, $h_0 \subseteq h_1 \subseteq \ldots$, és ha $h_s(x)\downarrow$, akkor

$$x \in A \quad \Leftrightarrow \quad h_s(x) \in B.$$

Legyen h_0 a sehol sem definiált függvény. Tegyük fel, hogy h_s már definiált.

1. eset. $s+1 = 2x+1$. Ha $h_s(x)\downarrow$, legyen $h_{s+1} = h_s$. Ha $h_s(x)\uparrow$, akkor képezzük az

$$f(x), f(h_s^{-1}(f(x))), \ldots, f((h_s^{-1}f)^n(x)), \ldots$$

sorozat első olyan tagját, amely nem esik a h_s értékkészletébe. Ha ez y, akkor legyen $h_{s+1}(x) = y$. Továbbá $x' \neq x$ esetén legyen $h_{s+1}(x') = h_s(x')$.

2. eset. $s+1 = 2y+2$. Ha $h_s^{-1}(y)\downarrow$, akkor legyen $h_{s+1} = h_s$. Ha $h_s^{-1}(y)\uparrow$, képezzük a

$$g(y), g(h_s(g(y))), \ldots, g((h_s g)^n(y)), \ldots$$

sorozat első olyan tagját, amely nem eleme a h_s értelmezési tartományának. Ha ez x, legyen $h_{s+1}(x) = y$, és $x' \neq x$ esetén legyen $h_{s+1}(x') = h_s(x')$.

A már említett tulajdonságokon kívül a h_s függvénysorozat rendelkezik az alábbi tulajdonságokkal is:

a) $\forall x \exists s \exists y\ h_s(x) = y$.

b) $\forall y \exists s \exists x\ h_s(x) = y$.

Ezen tulajdonságokat felhasználva kapjuk, hogy h bijektív rekurzív függvény, amelyre

$$x \in A \quad \Leftrightarrow \quad h(x) \in B. \quad \square$$

19 Rice tétele

Definíció 19.1 *Legyen $A \subseteq \mathbf{N}$. Az A halmazt index halmaznak nevezzük, ha tetszőleges x és y esetén*

$$x \in A \ \wedge\ \varphi_x = \varphi_y \quad \Rightarrow \quad y \in A.$$

Tehát minden index halmaz meghatározza a parciális rekurzív függvények egy tulajdonságát, és fordítva, a parciális rekurzív függvények minden egyes tulajdonságához tartozik egy index halmaz.

Tétel 19.2 *Legyen C az egyváltozós parciális rekurzív függvények tetszőleges halmaza. Ekkor a*

$$\Gamma_C = \{n|\varphi_n \in C\}$$

index halmaz akkor és csakis akkor rekurzív, ha C triviális, azaz $C = \emptyset$ vagy C az összes parciális rekurzív függvény halmaza.

Röviden: A parciális rekurzív függvények egyetlen nemtriviális tulajdonsága sem dönthető el.

Rice tétele következik az alábbi eredményből:

Tétel 19.3 *Legyen A nemtriviális index halmaz, azaz $A \neq \emptyset$, $A \neq \mathbf{N}$. Ekkor $K \leq_1 A$ vagy $K \leq_1 \overline{A}$.*

Bizonyítás. Legyen e_0 olyan szám, hogy φ_{e_0} sehol sem definiált. Tegyük fel, hogy $e_0 \notin A$. Belátjuk, hogy ekkor $K \leq_1 A$. Legyen $e_1 \in A$. Tekintsük a

$$\psi(x, y) = \begin{cases} \varphi_{e_1}(y) & \text{ha } x \in K \\ \uparrow & \text{különben} \end{cases}$$

parciális függvényt, amely parciális rekurzív:

$$\psi(x, y) = \varphi_{e_1}(y) \cdot p(x),$$

ahol $p(x) = 1$, ha $x \in K$, különben $p(x) \uparrow$. A paraméter tétel felhasználásával adódik egy olyan injektív (primitív) rekurzív f függvény, hogy $\psi(x, y) = \varphi_{f(x)}(y)$.

$$x \in K \quad \Rightarrow \quad \varphi_{f(x)} = \varphi_{e_1} \quad \Rightarrow \quad f(x) \in A$$

$$x \notin K \quad \Rightarrow \quad \varphi_{f(x)} = \varphi_{e_0} \quad \Rightarrow \quad f(x) \in \overline{A}.$$

Tehát $K \leq_f A$.

Ha $e_0 \in A$, hasonlóan bizonyítható a $K \leq_1 \overline{A}$ összefüggés. □

Megjegyzés 19.4 Rice tételének érvényes az alábbi, a rekurzívan felsorolható halmazokra vonatkozó alakja is: A rekurzívan felsorolható halmazok egyetlen nemtriviális index halmaza sem rekurzív. (Természetesen egy A halmaz a rekurzívan felsorolható halmazok indexhalmaza, ha

$$x \in A \ \wedge \ W_x = W_y \quad \Rightarrow \quad y \in A.)$$

Ez következik abból, hogy a rekurzívan felsorolható halmazok minden indexhalmaza a parciális rekurzív függvények indexhalmaza is.

A tétel alkalmazásával adódik, hogy az alábbi halmazok egyike sem rekurzív:

1. $\{x|\varphi_x \text{ rekurzív}\}$.
2. $\{x|W_x \neq \emptyset\}$.
3. $\{x|W_x \text{ véges}\}$.
4. $\{x|\varphi_x = \varphi_{x_0}\}$, ahol x_0 tetszőleges rögzített szám.

20 Teljes, produktív és kreatív halmazok

Definíció 20.1 *Legyen $r = 1$ vagy $r = m$. Az $A \subseteq \mathbf{N}$ halmaz r-teljes, ha rekurzívan felsorolható, és $W \leq_r A$ teljesül minden W rekurzívan felsorolható halmazra.*

Ebben a fejezetben megadjuk a teljes halmazok egy jellemzését. A K halmazt már definiáltuk, $K = \{x|x \in W_x\}$.

Definíció 20.2

$$\begin{aligned} K_0 &= \{\langle x, y\rangle | x \in W_y\} \\ K_1 &= \{x|W_x \neq \emptyset\}. \end{aligned}$$

Állítás 20.3 *A K, K_0, K_1 halmazok 1-teljesek, és így m-teljesek is.*

Bizonyítás. A K, K_0, K_1 halmazok rekurzíven felsorolhatók. Legyen $W = W_{y_0}$ rekurzívan felsorolható halmaz. Akkor $W \leq_f K_0$, ahol $f(x) = \langle x, y_0\rangle$. Tehát $W \leq_1 K_0$. Most definiáljuk a $g(x)$ injektív primitív rekurzív függvényt az alábbi összefüggéssel:

$$W_{g(x)} = \begin{cases} \mathbf{N} & \text{ha } x \in W \\ \emptyset & \text{különben.} \end{cases}$$

Ekkor:

$$\begin{aligned} x \in W &\quad\Rightarrow\quad g(x) \in W_{g(x)} = \mathbf{N} \quad\Rightarrow\quad g(x) \in K \cap K_1 \\ x \notin W &\quad\Rightarrow\quad W_{g(x)} = \emptyset \quad\Rightarrow\quad g(x) \notin K \cup K_1. \quad \square \end{aligned}$$

Következmény 20.4

$$K \equiv K_0 \equiv K_1.$$

Definíció 20.5 *(Post) A $P \subseteq \mathbf{N}$ halmaz produktív, ha létezik olyan $\psi(x)$ parciális rekurzív függvény (amelyet a P produktív függvényének nevezünk), amelyre*

$$(\forall x)[W_x \subseteq P \quad \Rightarrow \quad \psi(x) \downarrow \wedge\, \psi(x) \in P - W_x].$$

A $C \subseteq \mathbf{N}$ halmaz kreatív, ha rekurzívan felsorolható, és $\overline{C}$ produktív.

Példa 20.6 K kreatív, hiszen $\overline{K}$ egy produktív függvénye az identikus függvény, és K rekurzívan felsorolható. Valóban, ha $W_x \subseteq \overline{K}$, akkor $x \notin W_x$ és $x \in \overline{K}$.

Tétel 20.7 *Minden produktív halmaznak létezik injektív rekurzív produktív függvénye.*

Bizonyítás. Legyen P produktív a ψ produktív függvénnyel. Először megmutatjuk, hogy létezik a P halmaznak rekurzív, tehát teljesen definiált produktív függvénye. Legyen $g(x)$ olyan rekurzív függvény, amelyre

$$W_{g(x)} \quad = \quad \begin{cases} W_x & \text{ha } \psi(x)\downarrow \\ \emptyset & \text{különben.} \end{cases}$$

Legyen továbbá

$$q(x) \quad = \quad \begin{cases} \psi(x) & \text{ha } \psi(x) \text{ hamarabb konvergál, mint } \psi(g(x)) \\ \psi(g(x)) & \text{ha } \psi(g(x)) \text{ konvergál hamarabb.} \end{cases}$$

A $q(x)$ függvény teljesen definiált. Valóban, ha $\psi(x)\uparrow$, akkor $W_{g(x)} = \emptyset$. Mivel $W_{g(x)} = \emptyset \subseteq P$, ezért $\psi(g(x))\downarrow$. Ezek után könnyen látható, hogy $q(x)$ rekurzív függvény. Tegyük fel, hogy $W_x \subseteq P$. Ekkor $\psi(x)\downarrow$, és így $W_{g(x)} = W_x$.

a) $q(x) = \psi(x)$. Ekkor $\psi(x) \in P - W_x$ miatt $q(x) \in P - W_x$.

b) $q(x) = \psi(g(x))$. Ekkor $\psi(g(x)) \in P - W_{g(x)}$ és $W_x = W_{g(x)}$ miatt $q(x) \in P - W_x$.

Tehát

$$W_x \subseteq P \quad \Rightarrow \quad q(x) \in P - W_x.$$

Most a $q(x)$ rekurzív függvény felhasználásával készítünk egy olyan injektív rekurzív függvényt, amely szintén a P produktív függvénye. Létezik olyan $h(x)$ rekurzív függvény, amelyre

$$W_{h(x)} \quad = \quad W_x \cup \{q(x)\}.$$

Ekkor

$$W_x \subseteq P \quad \Rightarrow \quad W_{h(x)} \subseteq P,$$

hiszen q a P produktív függvénye. Így teljes indukcióval:

$$W_x \subseteq P \quad \Rightarrow \quad W_{h^n(x)} = W_x \cup \{q(x), qh(x), \ldots, qh^{n-1}(x)\} \subseteq P,$$

minden $n \geq 0$ esetén. Most legyen $p(0) = q(0)$, és ha $p(0), \ldots, p(x)$ már definiáltak, legyen $p(x+1)$ a

$$q(x+1),\ qh(x+1),\ qh^2(x+1), \ldots \tag{5}$$

sorozat első olyan eleme, mondjuk $y = qh^n(x+1)$, amely nincs a $\{p(0), \ldots, p(x)\}$ halmazban, amennyiben ilyen létezik. Ha $W_{x+1} \subseteq P$, akkor $y \in P$, a fentiek alapján. Másrészt mivel $W_{h^n(x+1)} \subseteq P$, $y \notin W_{h^n(x+1)}$, így $y \notin W_{x+1}$. Ha a (5) sorozat minden tagja a $\{p(0), \ldots, p(x)\}$ halmazba esik, akkor $W_{x+1} \not\subseteq P$. Ellenkező esetben ugyanis kapnánk egy olyan $W_{h^n(x+1)}$ halmazba eső számot, amely q melletti képe valamely $m < n$ esetén a $W_{h^m(x+1)}$ halmazba, így a $W_{h^n(x+1)}$ halmazba esne. Ez ellentmondana a

$$W_{h^n(x+1)} \subseteq P \quad \Rightarrow \quad qh^n(x+1) \notin W_{h^n(x+1)}$$

feltételnek. (Vö.: $W_{h^n(x+1)} = \{q(x+1), \ldots, qh^{n-1}(x+1)\} \cup W_{x+1}$). A második esetben így a $p(x+1)$ számot választhatjuk a legkisebb olyan számnak, amely nem esik a $\{p(0), \ldots, p(x)\}$ halmazba. Az így definiált p függvény a P halmaz injektív rekurzív produktív függvénye. □

Tétel 20.8 *Legyen P produktív halmaz. Ekkor*

(i) P nem rekurzívan felsorolható.

(ii) P tartalmaz végtelen rekurzívan felsorolható részhalmazt.

(iii) Ha $P \leq_m A$, akkor A is produktív.

Bizonyítás. Legyen $p(x)$ a P injektív rekurzív produktív függvénye.

(i) Tegyük fel, hogy $P = W_{x_0}$ valamely x_0 számra. Ekkor $p(x_0) \in P - W_{x_0} = \emptyset$. Ez nyilván ellentmondás.

(ii) Létezik olyan $h(x)$ rekurzív függvény, hogy

$$W_{h(x)} \quad = \quad W_x \cup \{p(x)\}.$$

Ha most $W_{x_0} \subseteq P$, akkor minden $m \geq 1$ esetén

$$W_{h^m(x_0)} = W_{x_0} \cup \{p(x_0), ph(x_0), \ldots, ph^{m-1}(x_0)\} \subseteq P.$$

Speciálisan, ha $W_n = \emptyset$, akkor

$$W_{h^m(n)} = \{p(n), ph(n), \ldots, ph^{m-1}(n)\} \subseteq P.$$

Így $W \subseteq P$, ahol

$$W = \{p(n), ph(n), ph^2(n), \ldots\}.$$

Könnyen látható, hogy W egy rekurzív függvény értékkészlete. Így W rekurzívan felsorolható. Az is világos, hogy

$$ph^m(n) \notin W_{h^m(n)},$$

hiszen $ph^m(n) \in P - W_{h^m(n)}$. Tehát W végtelen halmaz.

(iii) Legyen $P \leq_f A$, ahol f rekurzív. Létezik olyan $g(x)$ rekurzív függvény, amelyre

$$W_{g(x)} = f^{-1}(W_x).$$

Belátjuk, hogy az $f(p(g(x)))$ függvény az A produktív függvénye. Valóban:

$$W_x \subseteq A \quad \Rightarrow \quad W_{g(x)} \subseteq P \quad \Rightarrow \quad p(g(x)) \in P - W_{g(x)} \quad \Rightarrow \quad f(p(g(x))) \in A - W_x. \quad \square$$

Tétel 20.9 *(Myhill)*

(i) Ha P produktív, akkor $\overline{K} \leq_1 P$.

(ii) Ha C kreatív, akkor C 1-teljes.

Bizonyítás. Tegyük fel, hogy P produktív. Legyen $p(x)$ a P injektív rekurzív produktív függvénye. Létezik olyan $f(x,y)$ rekurzív függvény, hogy

$$W_{f(x,y)} = \begin{cases} \{p(x)\} & \text{ha } y \in K \\ \emptyset & \text{különben.} \end{cases}$$

A fixpont tétel szerint létezik olyan injektív (primitív) rekurzív $n(y)$ függvény, amelyre

$$W_{n(y)} = W_{f(n(y),y)} = \begin{cases} \{p(n(y))\} & \text{ha } y \in K \\ \emptyset & \text{különben.} \end{cases}$$

Így

$$\begin{array}{llllllll} y \in K & \Rightarrow & W_{n(y)} = \{pn(y)\} & \Rightarrow & W_{n(y)} \not\subseteq P & \Rightarrow & pn(y) \in \overline{P} \\ y \in \overline{K} & \Rightarrow & W_{n(y)} = \emptyset & \Rightarrow & W_{n(y)} \subseteq P & \Rightarrow & pn(y) \in P. \end{array}$$

(Az előző sorban a második implikáció kontrapozícióval adódik: Ha $W_{n(y)} \subseteq P$, akkor $pn(y) \in P - W_{n(y)}$, így $pn(y) \notin W_{n(y)}$.)

A második állítás következik az első felhasználásával. Ha C kreatív, akkor $\overline{C}$ produktív, így $\overline{K} \leq_1 \overline{C}$, és $K \leq_1 C$. Mivel C rekurzívan felsorolható, és mivel K 1-teljes, adódik, hogy C is 1-teljes. $\square$

Következmény 20.10 *Tetszőleges P halmazra ekvivalensek az alábbi állítások.*

(i) P *produktív.*

(ii) $\overline{K} \leq_1 P$.

(iii) $\overline{K} \leq_m P$.

Következmény 20.11 *Tetszőleges C halmazra ekvivalensek az alábbi állítások.*

(i) C *kreatív.*

(ii) C *rekurzívan felsorolható és* $K \leq_1 C$.

(iii) C *rekurzívan felsorolható és* $K \leq_m C$.

Következmény 20.12 *Tetszőleges C halmazra ekvivalensek az alábbi állítások.*

(i) C *kreatív.*

(ii) C 1*-teljes.*

(iii) C m*-teljes.*

(iv) $C \equiv K$.

Tehát a teljes halmazok rekurzívan izomorfak.

21 Relatív kiszámíthatóság, alapfogalmak

Módosítsuk eredeti $\mathcal{L}$ programozási nyelvünket az alábbi módon. A $V \leftarrow V$ alakú alaputasítás helyett tekintsük a

$$V \leftarrow O(V)$$

(orákulum) utasítást. Egyébként a programok hasonlóan épülnek fel, mint az $\mathcal{L}$ programozási nyelvben. Azt kiköthetjük, hogy programok nem végződhetnek $Y \leftarrow O(Y)$ utasításra, ahol Y a kimenő változó, ilyen utasítást ugyanis helyettesíthetnénk a

$$\begin{aligned} &V \leftarrow Y \\ &V \leftarrow O(V) \\ &Y \leftarrow V \end{aligned}$$

programrészlettel, ahol V új munkaváltozó.

Ahhoz, hogy a programoknak jelentést tulajdonítsunk, tekintsünk egy $G : \mathbf{N} \longrightarrow \mathbf{N}$ teljesen definiált függvényt. A szemantika megadásakor valamely (i, σ) pár rákövetkezőjét ugyanúgy definiáljuk, mint korábban, az alábbi eltéréssel. Amennyiben a P program i-dik utasítsa $V \leftarrow O(V)$ és σ tartalmazza a $V = k$ egyenletet, akkor (i, σ) rákövetkezője az $(i+1, \tau)$ pár, ahol $\tau = (\sigma - \{V = k\}) \cup \{V = G(k)\}$.

Jelölés: Legyen P egy program és $m \geq 1$. Ekkor

$$\psi_{P,G}^{(m)}(x_1, \ldots, x_m)$$

jelöli a P által G-kiszámított m-változós parciális függvényt.

Definíció 21.1 *Valamely $f : \mathbf{N}^m \longrightarrow \mathbf{N}$ parciális függvény parciális G-rekurzív, ha létezik olyan P program, amelyre*

$$f = \psi_{P,G}^{(m)}.$$

Ha f teljesen definiált is, akkor f G-rekurzív.

Definíció 21.2 *Legyen $A \subseteq \mathbf{N}$. Az $f : \mathbf{N}^m \longrightarrow \mathbf{N}$ (parciális) függvény (parciális) A-rekurzív, ha (parciális) P_A-rekurzív, ahol P_A az A karakterisztikus függvénye.*

Állítás 21.3 *(i) Ha f (parciális) rekurzív, akkor tetszőleges G függvényre (parciális) G-rekurzív.*

(ii) Valamely f függvény akkor és csak akkor (parciális) rekurzív, ha (parciális) I-rekurzív, ahol $I : \mathbf{N} \longrightarrow \mathbf{N}$ az identikus függvény.

Állítás 21.4 *Legyen $G : \mathbf{N} \longrightarrow \mathbf{N}$ függvény. Ekkor:*

(i) $G(x)$ G-rekurzív.

(ii) A G-rekurzív függvények osztálya PRZ-osztály.

Állítás 21.5 *Legyen f (parciális) G-rekurzív, G pedig H-rekurzív. Ekkor f (parciális) H-rekurzív.*

Bizonyítás. Legyen P olyan program, amelyre

$$f = \psi_{P,G}^{(m)},$$

és legyen Q olyan program, amelyre

$$G = \psi_{Q,H}^{(1)}.$$

Helyettesítsünk P-ben minden $V \leftarrow O(V)$ utasítást makró szerűen olyan, a Q programból kapott programrészlettel, amely H-kiszámítja a G függvényt. Az így előálló R programra

$$f = \psi_{R,H}^{(m)}. \quad \square$$

Tétel 21.6 *Legyen G rekurzív függvény. Ekkor egy f (parciális) függvény akkor és csak akkor (parciális) rekurzív, ha (parciális) G-rekurzív.*

Bizonyítás. Ha f (parciális) rekurzív, akkor f minden G függvényre (parciális) G-rekurzív, hiszen f kiszámítható olyan, az $\mathcal{L}$ nyelvben felírt programmal, amelyben nem szerepel $V \leftarrow O(V)$ alakú utasítás.

Fordítva, ha G rekurzív függvény, akkor G I-rekurzív is. Ha f (parciális) G-rekurzív, akkor az előző állítás szerint f (parciális) I-rekurzív, tehát (parciális) rekurzív. $\square$

22 Néhány korábbi eredmény relativizált alakja

Az orákulum utasítások bevezetésével kapott programozási nyelv P programjaihoz ugyanúgy rendelhetünk egy $\#(P)$ Gödel számot, mint az eredeti $\mathcal{L}$ programozási nyelv esetén. Tetszőleges $G : \mathbf{N} \to \mathbf{N}$ függvény esetén definiáljuk a $\varphi_G^{(n)}$ függvényeket a

$$\varphi_G^{(n)}(x_1, \ldots, x_n, y) \quad = \quad \psi_{P,G}^{(n)}(x_1, \ldots, x_n)$$

összefüggéssel, ahol P az a program, amelyre $\#(P) = y$. (Gyakran fogjuk használni a $\varphi_{G,y}^{(n)}$ jelölést is. Amennyiben $n = 1$, a felső indexet elhagyva csupán $\varphi_{G,y}(x)$-et írunk.)

Tétel 22.1 *A $\varphi_G^{(n)}(x_1, \ldots, x_n, y)$ függvény parciális G-rekurzív.*

Tetszőleges $n \geq 1$ esetén definiáljuk az $\mathrm{STP}_G^{(n)}$ $n + 2$-változós predikátumot az alábbi módon:

$$\mathrm{STP}_G^{(n)}(x_1, \ldots, x_n, y, t) \quad \Leftrightarrow \quad \begin{array}{l}\text{az } y \text{ Gödel számú program legfeljebb } t \text{ lépésben} \\ \text{megáll az } x_1, \ldots, x_n \text{ adatokon.}\end{array}$$

Tétel 22.2 $\mathrm{STP}_G^{(n)}(x_1, \ldots, x_n, y, t)$ *G-rekurzív predikátum.*

Érvényes a paraméter tétel is.

Tétel 22.3 *Minden $m, n \geq 1$ esetén létezik olyan S_m^n injektív primitív rekurzív függvény, hogy*

$$\varphi_G^{(m+n)}(x_1, \ldots, x_m, u_1, \ldots, u_n, y) \quad = \quad \varphi_G^{(m)}(x_1, \ldots, x_m, S_m^n(u_1, \ldots, u_n, y)).$$

23 G-rekurzív és G-rekurzívan felsorolható halmazok

Definíció 23.1 *Legyen G egy teljesen definiált függvény. Az $A \subseteq \mathbf{N}$ halmazt G-rekurzívnak nevezzük, ha A karakterisztikus függvénye G-rekurzív. Speciálisan, ha G a $B \subseteq \mathbf{N}$ halmaz karakterisztikus függvénye és A G-rekurzív, akkor az A halmazt B-rekurzívnak is nevezzük.*

Definíció 23.2 *Legyen $G : \mathbf{N} \longrightarrow \mathbf{N}$ függvény és $A \subseteq \mathbf{N}$. Az A halmaz G-rekurzívan felsorolható, ha létezik olyan parciálisan G-rekurzív $f : \mathbf{N} \longrightarrow \mathbf{N}$ függvény, hogy*

$$A = \{x | f(x) \downarrow\}.$$

Amennyiben A P_B-rekurzívan felsorolható, ahol P_B a $B \subseteq \mathbf{N}$ halmaz karakterisztikus függvénye, akkor A B-rekurzívan felsorolható.

Állítás 23.3 *Legyenek G és H függvények. Minden rekurzív halmaz G-rekurzív, és minden rekurzívan felsorolható halmaz G-rekurzívan felsorolható. Tegyük fel, hogy G H-rekurzív. Akkor minden G-rekurzív (G-rekurzívan felsorolható) halmaz H-rekurzív (H-rekurzívan felsorolható).*

Állítás 23.4 *Legyen G rekurzív függvény. Valamely A halmaz akkor és csak akkor rekurzív (rekurzívan felsorolható), ha G-rekurzív (G-rekurzívan felsorolható).*

Definíció 23.5 *Legyen $A, B \subseteq \mathbf{N}$. Az A halmaz Turing-reducibilis B-re, $A \leq_T B$, ha A B-rekurzív. Ha $A \leq_T B$ és $B \leq_T A$ is teljesül, akkor A és B Turing-ekvivalensek: $A \equiv_T B$.*

Állítás 23.6 *$A \leq_T$ reláció kvázi-rendezés, a $\equiv_T$ reláció pedig ekvivalencia reláció. Továbbá $\overline{A} \leq_T A$ és $A \leq_T \overline{A}$ teljesül tetszőleges $A \subseteq \mathbf{N}$ halmazra. Tetszőleges A és B esetén A akkor és csak akkor B-rekurzív (B-rekurzívan felsorolható), ha $\overline{B}$-rekuzív ($\overline{B}$-rekurzívan felsorolható).*

Állítás 23.7 *Ha $A \leq_m B$, akkor $A \leq_T B$. Ha $A \equiv_m B$, akkor $A \equiv_T B$.*

Bizonyítás. Tegyük fel, hogy $A \leq_f B$, ahol f rekurzív függvény. Legyen Q az alábbi program:

$$\begin{array}{l} X_1 \leftarrow f(X_1) \\ X_1 \leftarrow O(X_1) \\ Y \leftarrow X_1 \end{array}$$

Ekkor $\psi^{(1)}_{Q,B}$ az A karakterisztikus függvénye.

Korábbi eredményeink kiterjeszthetőek. Nevezetesen

(a) A G-rekurzív halmazok zártak az egyesítésre, a metszetre, és a komplementációra.

(b) A G-rekurzívan felsorolható halmazok zártak az egyesítésre és a metszetre, de nem zártak a komplementációra.

(c) Egy A halmaz akkor és csak akkor G-rekurzív, ha A és $\overline{A}$ G-rekurzívan felsorolható.

(d) Létezik olyan G-rekurzívan felsorolható halmaz, amely nem G-rekurzív.

A legutolsó állításhoz definiáljuk tetszőleges n számra és $G : \mathbf{N} \to \mathbf{N}$ függvényre:

$$W_n^G = \{x \in \mathbf{N} | \varphi_{G,n}(x) \downarrow\}.$$

Tehát a W_n^G halmazok az összes G-rekurzívan felsorolható halmazok.

Definíció 23.8

$$G' = \{n | n \in W_n^G\}.$$

Speciálisan, ha G a B halmaz karakterisztikus függvénye, W_n^G helyett a W_n^B, G' helyett a B' jelölést használjuk.

Tétel 23.9 *G' G-rekurzívan felsorolható, de nem G-rekurzív.*

Korábbi, a teljességre vonatkozó tételünk is kiterjeszthető:

Tétel 23.10 *Tetszőleges $A \subseteq \mathbf{N}$ halmazra ekvivalensek az alábbi állítások.*

i) A G-rekurzívan felsorolható

ii) $A \leq_m G'$

iii) $A \leq_1 G'$.

Vegyük észre, hogy $K \equiv \emptyset'$, illetve $K \equiv G'$, valahányszor G rekurzív függvény. Tehát ha A rekurzív halmaz, akkor $A' \equiv K$.

A rekurzívan felsorolható halmazokra vonatkozó normál forma tétel megfelelője az alábbi eredmény.

Tétel 23.11 *Legyen $G : \mathbf{N} \to \mathbf{N}$ teljesen definiált függvény. Valamely $A \subseteq \mathbf{N}$ halmaz akkor és csak akkor G-rekurzívan felsorolható, ha létezik olyan B G-rekurzív halmaz, hogy*

$$A = \{x | \exists y \langle x, y \rangle \in B\}. \tag{6}$$

Következmény 23.12 *Legyen B G-rekurzívan felsorolható halmaz és*

$$A = \{x | \exists y \langle x, y\rangle \in B\}.$$

Akkor A G-rekurzívan felsorolható.

Tétel 23.13 *Legyen $A, B \subseteq \mathbf{N}$. Ekkor:*

(i) $A \leq_T B$ akkor és csak akkor, ha $A' \leq_1 B'$.

(ii) $A \equiv_T B$ akkor és csak akkor, ha $A' \equiv_1 B'$.

Bizonyítás.

(i) Mivel A' A-rekurzívan felsorolható, ha $A \leq_T B$, akkor A' B-rekurzívan felsorolható. De B' teljes a B-rekurzívan felsorolható halmazok osztályában, így $A' \leq_1 B'$. Fordítva, $A, \overline{A} \leq_T A$, tehát A és $\overline{A}$ A-rekurzívan felsorolhatóak. Így $A, \overline{A} \leq_1 A'$, és ha $A' \leq_1 B'$, akkor $A, \overline{A} \leq_1 B'$. Ezért A és $\overline{A}$ B-rekurzívan felsorolhatóak, tehát A B-rekurzív: $A \leq_T B$.

(ii) Nyilvánvaló (i)-ből. □

Következmény 23.14 *Legyenek A, B az $\mathbf{N}$ részhalmazai. Ha $A \leq_1 B$ akkor $A' \leq_1 B'$.*

Következmény 23.15 *Legyenek A, B az $\mathbf{N}$ részhalmazai.*

(i) $A \leq_T B$ akkor és csakis akkor, ha $A' \leq_m B'$.

(ii) $A \equiv_T B$ akkor és csakis akkor, ha $A' \equiv_m B'$, illetve ha $A' \equiv B'$.

24 Az aritmetikai hierarchia

Definíció 24.1

$$\begin{aligned} \Sigma_0 &: \textit{rekurzív halmazok} \\ \Sigma_{n+1} &: \textit{azon halmazok, amelyek } B\textit{-rekurzívan} \\ & \quad \textit{felsorolhatóak valamely } B \in \Sigma_n \textit{ halmazra.} \end{aligned}$$

Definíció 24.2

$$\begin{aligned} \Pi_n &= co - \Sigma_n = \{A \subseteq \mathbf{N} \mid \overline{A} \in \Sigma_n\} \\ \Delta_n &= \Sigma_n \cap \Pi_n. \end{aligned}$$

Megjegyzés 24.3 Tetszőleges A halmazra, $A \in \Sigma_{n+1}$ akkor és csak akkor, ha A B-rekurzívan felsorolható valamely $B \in \Pi_n$ halmazra. (Lásd egy korábbi állítást.) Hasonlóan, $A \in \Pi_{n+1}$ akkor és csak akkor, ha $\overline{A}$ B-rekurzívan felsorolható valamely $B \in \Sigma_n$ (vagy valamely $B \in \Pi_n$) halmazra. Így $A \in \Delta_{n+1}$ akkor és csak akkor, ha A B-rekurzívan felsorolható és $\overline{A}$ C-rekurzívan felsorolható valamely $B, C \in \Sigma_n$ (vagy $B, C \in \Pi_n$) halmazokra. Később belátjuk, hogy ez tovább ekvivalens azzal, hogy $A \leq_T B$ valamely $B \in \Sigma_n$ (vagy $B \in \Pi_n$) halmazra. Nyílván igaz az is, hogy $\Sigma_0 = \Pi_0 = \Delta_0 = \Delta_1$. Továbbá Σ_1 a rekurzívan felsorolható, Π_1 pedig a co-rekurzívan felsorolható halmazok osztálya. A Σ_n, Π_n és Δ_n osztályok mindegyike zárt a $\leq_T$ reducibilitásra: Ha $A \leq_T B$ és B ezen osztályok valamelyikébe esik, akkor A ugyanabban az osztályban van. Így Σ_n, Π_n és Δ_n zártak a $\leq_1$ és $\leq_m$ visszavezetésekre is.

Megjegyzés 24.4 Minden $n \geq 0$ esetén érvényesek a következő tartalmazások: $\Sigma_n \subseteq \Sigma_{n+1}$, $\Sigma_n \subseteq \Pi_{n+1}$, $\Pi_n \subseteq \Pi_{n+1}$, $\Delta_n \subseteq \Delta_{n+1}$, $\Sigma_n \cup \Pi_n \subseteq \Delta_{n+1}$.

Állítás 24.5 *Legyen $A \subseteq \mathbf{N}$. $A \in \Sigma_{n+1}$ akkor és csak akkor teljesül, ha létezik olyan $B \in \Sigma_n$, hogy $A \leq_1 B'$, vagy $A \leq_m B'$.*

Bizonyítás. Ez következik abból, hogy B' teljes a B-rekurzívan felsorolható halmazok osztályában. $\square$

Definíció 24.6 *Valamely B halmaz Σ_n-teljes (Π_n-teljes), ha $B \in \Sigma_n$ ($B \in \Pi_n$), és $A \leq_1 B$ teljesül valahányszor $A \in \Sigma_n$ ($A \in \Pi_n$).*

Megjegyzés 24.7 B akkor és csak akkor Σ_n-teljes (Π_n-teljes), ha $B \in \Sigma_n$ ($B \in \Pi_n$), és $A \leq_m B$ teljesül valahányszor $A \in \Sigma_n$ ($A \in \Pi_n$).

Megjegyzés 24.8 Egy B halmaz akkor és csak akkor Σ_n-teljes, ha $\overline{B}$ Π_n-teljes.

Definíció 24.9

$$\emptyset^{(0)} = \emptyset, \quad \emptyset^{(n+1)} = (\emptyset^{(n)})', n \geq 0.$$

Lemma 24.10 *Minden $n \geq 0$ esetén $\emptyset^{(n)} \in \Sigma_n$.*

Bizonyítás. $\emptyset^{(0)} = \emptyset \in \Sigma_0$. Ha $\emptyset^{(n)} \in \Sigma_n$, akkor mivel $\emptyset^{(n+1)}$ $\emptyset^{(n)}$-rekurzívan felsorolható, fennáll a $\emptyset^{(n+1)} \in \Sigma_{n+1}$ tartalmazás is. $\square$

Tétel 24.11 *Legyen $n \geq 0$.*

(i) Ha $n > 0$, akkor valamely A halmaz akkor és csak akkor Σ_n-halmaz, ha $A \leq_1 \emptyset^{(n)}$. Így ha $n > 0$, akkor $\emptyset^{(n)}$ Σ_n-teljes.

(ii) $A \in \Sigma_{n+1}$ *akkor és csak akkor, ha A* $\emptyset^{(n)}$*-rekurzívan felsorolható.*

(iii) $A \in \Pi_{n+1}$ *akkor és csak akkor, ha* $\overline{A}$ $\emptyset^{(n)}$*-rekurzívan felsorolható.*

(iv) $A \in \Delta_{n+1}$ *akkor és csak akkor, ha* $A \leq_T \emptyset^{(n)}$, *azaz A* $\emptyset^{(n)}$*-rekurzív halmaz.*

Következmény 24.12 *Ha* $n > 0$, *akkor a* $\overline{\emptyset^{(n)}}$ *halmaz* Π_n*-teljes.*

Következmény 24.13 *Legyen* $A \subseteq \mathbf{N}$ *és* $n \geq 0$. *Ekkor* $A \in \Delta_{n+1}$ *akkor és csakis akkor, ha* $A \leq_T B$ *valamely* $B \in \Sigma_n$ *(vagy* $B \in \Pi_n$*) halmazra.*

Bizonyítás.

(i) A bizonyítást n szerinti teljes indukcióval végezzük el. Az $n = 1$ esetben $\emptyset' \equiv K$, és így $A \in \Sigma_1$ akkor és csak akkor, ha $A \leq_1 \emptyset'$. Legyen most $n \geq 1$ és tegyük fel, hogy $\emptyset^{(n)}$ Σ_n-teljes. Így $\overline{\emptyset^{(n)}}$ Π_n-teljes.

$$\begin{aligned} A \in \Sigma_{n+1} \quad &\Leftrightarrow \quad \text{van olyan } B\ \Sigma_n\text{-halmaz melyre } A\ B\text{-rekurzívan felsorolható} \\ &\Leftrightarrow \quad A \leq_1 B' \text{ valamely } B\ \Sigma_n\text{-halmazra} \\ &\Leftrightarrow \quad A \leq_1 (\emptyset^{(n)})' \\ &\Leftrightarrow \quad A \leq_1 \emptyset^{(n+1)}. \end{aligned}$$

(ii) Nyilvánvaló (i)-ből és a teljességi tételből.

(iii) Következik (ii)-ből.

(iv) Következik (ii)-ből és (iii)-ból. Továbbá $A \in \Delta_{n+1}$ akkor és csakis akkor, ha $A \leq_T B$ valamely $B \in \Sigma_n$ vagy $B \in \Pi_n$ halmazra. □

Megjegyzés 24.14 Az előző tétel akkor is érvényben marad, ha benne a $\leq_m$ reducibilitást használjuk.

Tétel 24.15 *(A hierarchia tétel) Legyen* $n \geq 0$.

(i) Ha $n > 0$, *akkor* $\Delta_n \subset \Sigma_n$ *és* $\Delta_n \subset \Pi_n$.

(ii) $\Sigma_n \subset \Sigma_{n+1}$ *és* $\Pi_n \subset \Pi_{n+1}$.

(iii) Ha $n > 0$, *akkor* $\Sigma_n \cup \Pi_n \subset \Delta_{n+1}$.

Bizonyítás.

(i) Definíció szerint $\Delta_n \subseteq \Sigma_n$ és $\Delta_n \subseteq \Pi_n$. Egy korábbi tételünk szerint $\emptyset^{(n)} \in \Sigma_n - \Delta_n$, hiszen a $\emptyset^{(n)}$ Σ_n-teljes, de nem $\emptyset^{(n-1)}$-rekurzív. Így $\overline{\emptyset^{(n)}} \in \Pi_n - \Delta_n$.

(ii) A $\Sigma_n \subseteq \Sigma_{n+1}$ és $\Pi_n \subseteq \Pi_{n+1}$ tartalmazások nyílvánvalóak. Mivel $\Sigma_n \cup \Pi_n \subseteq \Delta_{n+1} \subset \Sigma_{n+1}$, fennáll a $\Sigma_n \subset \Sigma_{n+1}$ tartalmazás.

(iii) Nyilvánvalóan $\Sigma_n \cup \Pi_n \subseteq \Delta_{n+1}$. Be kell látnunk, hogy létezik olyan $A \in \Delta_{n+1}$, melyre $A \notin \Sigma_n \cup \Pi_n$. Legyen

$$A \;=\; \emptyset^{(n)} \oplus \overline{\emptyset^{(n)}} \;=\; \{2x | x \in \emptyset^{(n)}\} \cup \{2x+1 | x \in \overline{\emptyset^{(n)}}\}$$

Ekkor

$$A \leq_T \emptyset^{(n)} \quad \text{és} \quad \emptyset^{(n)}, \overline{\emptyset^{(n)}} \leq_1 A.$$

Tegyük fel, hogy $A \in \Sigma_n$. Akkor $\overline{\emptyset^{(n)}} \in \Sigma_n$. Ezért $\emptyset^{(n)} \in \Delta_n$. Hasonlóan, ha $A \in \Pi_n$, akkor $\emptyset^{(n)} \in \Pi_n$ és így $\emptyset^{(n)} \in \Delta_n$. Tehát $A \notin \Sigma_n \cup \Pi_n$, ugyanakkor $A \leq_T \emptyset^{(n)}$ szerint $A \in \Delta_{n+1}$.

□

Következmény 24.16 *Ha $n > 0$, akkor $\Sigma_n \neq \Pi_n$. Valahányszor A Σ_n-teljes, $A \notin \Pi_n$. Hasonlóan, ha A Π_n-teljes, akkor $A \notin \Sigma_n$.*

Lemma 24.17 *Legyen $n \geq 0$.*

(i) Ha $A \in \Sigma_{n+1}$, akkor

$$\{x | \exists y \langle x, y\rangle \in A\}$$

Σ_{n+1}-halmaz.

(ii) Ha $A \in \Pi_{n+1}$, akkor

$$\{x | \forall y \langle x, y\rangle \in A\}$$

Π_{n+1}-halmaz.

Bizonyítás. Legyen A Σ_{n+1}-halmaz. Akkor A B-rekurzívan felsorolható valamely B Σ_n-halmazra, és így az $\{x | \exists y \langle x, y\rangle \in A\}$ halmaz is B-rekuzívan felsorolható. A második állítás következik az elsőből.

□

Tétel 24.18 *Legyen $n \geq 0$.*

(i) $A \in \Sigma_{n+1}$ akkor és csakis akkor, ha létezik olyan $B \in \Pi_n$, hogy

$$A \;=\; \{x \mid \exists y \, \langle x, y\rangle \in B\}. \tag{7}$$

(ii) $A \in \Pi_{n+1}$ akkor és csakis akkor, ha létezik olyan $B \in \Sigma_n$, hogy

$$A \;=\; \{x \mid \forall y \, \langle x, y\rangle \in B\}. \tag{8}$$

Bizonyítás. Először megmutatjuk, hogy ha (i) teljesül valamely n számra akkor ugyanerre a számra (ii) is teljesül. Tegyük fel, hogy (i) igaz valamely $n \geq 0$-ra. Ha $A \in \Pi_{n+1}$, akkor, lévén $\overline{A} \in \Sigma_{n+1}$, létezik olyan $B \in \Pi_n$, melyre

$$\overline{A} \;=\; \{x \mid \exists y\, \langle x,y\rangle \in B\}.$$

Tehát

$$A \;=\; \{x \mid \forall y\, \langle x,y\rangle \in \overline{B}\},$$

ahol $\overline{B} \in \Sigma_n$. Másrészt ha valamely $B \in \Sigma_n$ halmazra fennáll a (8) egyenlőség, akkor az

$$\overline{A} \;=\; \{x \mid \exists y\, \langle x,y\rangle \in \overline{B}\}$$

halmaz B-rekurzívan felsorolható, tehát $A \in \Pi_{n+1}$. Ezzel beláttuk, hogy az (i) feltételből következik (ii).

Most megmutatjuk, hogy (i) teljesül minden $n \geq 0$-ra. Ha $B \in \Pi_n$ és az A halmazt a (7) egyenlőséggel definiáljuk, akkor nyilvánvaló, hogy A $\overline{B}$-rekurzívan felsorolható, tehát $A \in \Sigma_{n+1}$. Fordítva, tegyük fel, hogy $A \in \Sigma_{n+1}$. Akkor A C-rekurzívan felsorolható valamely $C \in \Sigma_n$, $C \neq \emptyset$ halmazra, mondjuk

$$A \;=\; W_e^C \;=\; \{x \mid \exists y\, \mathrm{STP}_C(x,e,y)\}.$$

Ha $n = 0$, akkor C rekurzív és a

$$B \;=\; \{\langle x,y\rangle \mid \mathrm{STP}_C(x,e,y)\} \;\in\; \Pi_0$$

halmazra fennáll a (7) egyenlőség. Tegyük fel, hogy $n > 0$. A fenti B halmazra most is fennáll a (7) egyenlőség, viszont csak annyi biztos, hogy B Δ_{n+1}-ben van, nem Π_n-ben. Másik B halmazt kell tehát keresnünk.

Mivel C D-rekurzívan felsorolható valamely $D \in \Sigma_{n-1}$ halmazra, létezik egy $\gamma : \mathbf{N} \to \mathbf{N}$ D-rekurzív függvény, hogy

$$C \;=\; \{\gamma(0), \gamma(1), \ldots\}.$$

Minden $k \geq 0$-ra legyen

$$C_k \;=\; \{\gamma(0), \ldots, \gamma(k)\},$$

továbbá STP' legyen az a négyváltozós predikátum, melyre $\mathrm{STP}'(x,y,k,m)$ pontosan akkor igaz, ha az e Gödel-számú program a C_k orákulummal az x inputon legfeljebb y lépésben megáll úgy, hogy közben az orákulumot csak m-nél nem nagyobb számokra hívja meg. A bizonyítás befejezéséhez szükségünk lesz két segédtételre.

Lemma 24.19 *Az* STP' *predikátum* D*-rekurzív.*

Bizonyítás. Mivel a γ függvény D-rekurzív, megadható egy D orákulumot használó program, amely szimulálja az e Gödel-számú program működését a C_k orákulummal az x inputon legfeljebb y lépésig úgy, hogy közben azt is figyeli, hogy a C_k orákulumhoz intézett kérdések egyike sem legyen m-nél nagyobb. □

Lemma 24.20 *Minden $x, y \in \mathbf{N}$-re*

$$\mathrm{STP}_C(x, e, y) \quad \Leftrightarrow \quad \exists m\, \exists t\, \forall k\, [k \geq t \,\Rightarrow\, \mathrm{STP}'(x, y, k, m)].$$

Bizonyítás. Először tegyük fel, hogy $\mathrm{STP}_C(x, e, y)$ teljesül valamely x, y számokra. Ez azt jelenti, hogy az e Gödel-számú program az x inputon a C orákulumot használva legfeljebb y lépésben megáll, így működése közben csak véges sok (legfeljebb y darab) kérdést intéz az orákulumhoz. Legyen m a legnagyobb olyan szám, melyre a program meghívja az orákulumot. Mivel

$$C \quad = \quad \bigcup_{k \geq 0} C_k,$$

létezik egy olyan t index, hogy az m-nél nem nagyobb számok mindegyike pontosan akkor van benne C-ben, ha benne van C_t-ben. Vegyük észre, hogy $k \geq t$ esetén ugyanez igaz a C_k halmazra is. Ez azt jelenti, hogy ha az e Gödel-számú programot a C_k orákulummal futtatjuk le az x inputon valamely $k \geq t$-re, akkor az pontosan ugyanúgy fog viselkedni, mintha a C orákulummal futtatnánk. A fordított irányú implikáció igazolásához tegyük fel, hogy valamely x, y, t, m számokra igaz a

$$\forall k\, [k \geq t \,\Rightarrow\, \mathrm{STP}'(x, y, k, m)]$$

állítás. Legyen $k \geq t$ olyan, hogy a C_k halmaz tartalmazza C összes m-nél nem nagyobb elemét. Akkor feltevésünk szerint $\mathrm{STP}'(x, y, k, m)$ teljesül, amiből egyrészt $\mathrm{STP}_{C_k}(x, e, y)$, másrészt pedig az következik, hogy az e Gödel-számú program a C orákulumot használva ugyanúgy viselkedik az x inputon, mint amikor C helyett a C_k orákulomot használja. Ezért $\mathrm{STP}_C(x, e, y)$. □

Visszatérve a tétel bizonyítására, mivel az STP' predikátum D-rekurzív, a

$$B' \quad = \quad \{\langle x, y, t, m, k\rangle \mid k \geq t \Rightarrow STP'(x, y, k, m)\}$$

halmaz Δ_n-ben van. Így

$$B'' \quad = \quad \{\langle x, y, t, m\rangle \mid \forall k[k \geq t \Rightarrow STP'(x, y, k, m)]\}$$

és

$$B \quad = \quad \{\langle x, z\rangle \mid \langle x, (z)_1, (z)_2, (z)_3\rangle \in B''\}$$

Π_n-halmazok. Végül

$$A \quad = \quad \{x \mid \exists z \langle x, z\rangle \in B\}.$$

□

Következmény 24.21 *Legyen $A \subseteq \mathbf{N}$ és $n \geq 0$. Az alábbi állítások ekvivalensek.*

(i) *A Σ_n-halmaz.*

(ii)
$$A = \{x|(\exists y_1)(\forall y_2)\dots(Qy_n)\langle x, y_1, \dots, y_n\rangle \in B\}$$
alakú valamely B rekurzív halmazra, ahol Q a $\exists$ vagy $\forall$ kvantor, attól függően, hogy n páratlan vagy páros.

(iii)
$$A = \{x|(\exists \mathbf{y}_1)(\forall \mathbf{y}_2)\dots(Q\mathbf{y}_n)\langle x, \mathbf{y}_1, \dots, \mathbf{y}_n\rangle \in B\}$$
alakú valamely B rekurzív halmazra, ahol Q a $\exists$ vagy $\forall$ kvantor, attól függően, hogy n páratlan vagy páros, továbbá az $\mathbf{y}_i$ változók vektor változók.

Definíció 24.22 *Legyen $R \subseteq \mathbf{N}^m$ reláció és $n \geq 0$. Azt mondjuk, hogy R Σ_n-reláció (Π_n-reláció), ha a*
$$\{\langle x_1, \dots, x_m\rangle|(x_1, \dots, x_m) \in R\}$$
halmaz Σ_n-halmaz (Π_n-halmaz).

Következmény 24.23 *Legyen $R \subseteq \mathbf{N}^m$ és $n \geq 0$. Az alábbi állítások ekvivalensek.*

(i) *R Σ_n-reláció.*

(ii)
$$R = \{(x_1, \dots, x_m)|(\exists y_1)(\forall y_2)\dots(Qy_n)(x_1, \dots, x_m, y_1, \dots, y_n) \in S\}$$
alakú valamely S rekurzív relációra, ahol Q a $\exists$ vagy $\forall$ kvantor, attól függően, hogy n páratlan vagy páros.

(iii)
$$R = \{(x_1, \dots, x_m)|(\exists \mathbf{y}_1)(\forall \mathbf{y}_2)\dots(Q\mathbf{y}_n)(x_1, \dots, x_m, \mathbf{y}_1, \dots, \mathbf{y}_n) \in S\}$$
alakú valamely S rekurzív relációra, ahol Q a $\exists$ vagy $\forall$ kvantor, attól függően, hogy n páratlan vagy páros, továbbá az $\mathbf{y}_i$ változók vektor változók.

Az előző két következményhez hasonló állítások érvényesek a Π_n-halmazokra ill. relációkra is. Itt azonban a kvantorok váltakozó sorozata univerzális kvantorral indul.

Megjegyzés 24.24 Egy $R \subseteq \mathbf{N}^m$ reláció aritmetikai, ha első rendben definiálható az $\langle \mathbf{N}, 0, 1, +, \cdot, \leq\rangle$ struktura felett. A fenti eredmények szerint R akkor és csak akkor aritmetikai, ha valamely n esetén Σ_n-reláció (vagy Π_n-reláció).

25 Σ_n-teljes halmazok

Már láttuk, hogy a K, K_0, K_1 halmazok Σ_1-teljesek. Minden $n \geq 1$ esetén $\emptyset^{(n)}$ Σ_n-teljes. Most megadunk néhány természetes Σ_n-teljes halmazt $n = 2$ és $n = 3$ esetén.

Definíció 25.1

$$\begin{aligned} \mathrm{FIN} &= \{x \in \mathbf{N} | W_x \text{ } \textit{véges}\} \\ \mathrm{INF} &= \{x \in \mathbf{N} | W_x \text{ } \textit{végtelen}\} \\ \mathrm{TOT} &= \{x \in \mathbf{N} | W_x = \mathbf{N}\} = \{x \in \mathbf{N} | \varphi_x \text{ } \textit{teljesen definiált}\} \end{aligned}$$

Állítás 25.2 FIN $\in \Sigma_2$, INF, TOT $\in \Pi_2$.

Bizonyítás.

$$x \in \mathrm{FIN} \quad \Leftrightarrow \quad \exists s \forall t \forall z [z \geq s \ \Rightarrow \sim \mathrm{STP}(z, x, t)]$$

Tehát FIN $\in \Sigma_2$ és INF $\in \Pi_2$. Másrészt

$$x \in \mathrm{TOT} \quad \Leftrightarrow \quad \forall z \exists t \mathrm{STP}(z, x, t).$$

Ezért TOT $\in \Pi_2$. □

Állítás 25.3 FIN Σ_2*-teljes,* INF *és* TOT Π_2*-teljesek.*

Bizonyítás. Legyen $A \in \Sigma_2$. Ekkor $\overline{A} \in \Pi_2$ és így létezik olyan $R(x, y, z)$ rekurzív predikátum, hogy

$$x \in \overline{A} \quad \Leftrightarrow \quad (\forall y)(\exists z) R(x, y, z).$$

A paraméter tétel szerint létezik olyan $f(x)$ injektív rekurzív függvény, hogy

$$\varphi_{f(x)}(u) \quad = \quad \begin{cases} 0 & \text{ha } (\forall y)_{\leq u} (\exists z) R(x, y, z) \\ \uparrow & \text{különben.} \end{cases}$$

Így

$$\begin{aligned} x \in \overline{A} \quad &\Rightarrow \quad W_{f(x)} = \mathbf{N} \quad \Rightarrow \quad f(x) \in \mathrm{TOT} \subseteq \mathrm{INF} \\ x \in A \quad &\Rightarrow \quad W_{f(x)} \text{ véges} \quad \Rightarrow \quad f(x) \in \mathrm{FIN}. \quad \square \end{aligned}$$

Következmény 25.4

$$\mathrm{INF} \equiv \mathrm{TOT}.$$

Definíció 25.5

$$\begin{aligned} \mathrm{REC} &= \{x \in \mathbf{N} | W_x \textit{ rekurzív}\} \\ \mathrm{COF} &= \{x \in \mathbf{N} | W_x \textit{ co-véges}\}. \end{aligned}$$

Állítás 25.6 REC *és* COF Σ_3*-teljesek.*

Következmény 25.7

$$\mathrm{REC} \equiv \mathrm{COF}.$$

26 Szavak numerikus reprezentációja

Legyen A egy véges nemüres halmaz, melyen egy rögzített lineáris rendezést tételezünk fel. Mondjuk $A = \{s_1, \ldots, s_n\}$. Egy $w = s_{i_k} \ldots s_{i_0}$, $k > 0$, szónak feleltessük meg az

$$x = i_k n^k + \ldots + i_0$$

számot, a $w = \lambda$ szónak pedig az $x = 0$ számot. Így egy bijekciót létesítettünk az A^* és az $\mathbf{N}$ halmazok között, ha $w \mapsto x$ akkor w az x szám n-alapú reprezentációja és x a w szó által n-alapban jelölt szám. (Vegyük észre, hogy $n = 1$ esetén $x = |w|$, azaz x a w hossza.) Legyen

$$R^+(x,y) = \begin{cases} R(x,y) & \text{ha } y \not| x \\ y & \text{ha } y | x \end{cases} \qquad (\text{Így: } R^+(x,0) = x.)$$

$$Q^+(x,y) = \begin{cases} \lfloor x/y \rfloor & \text{ha } y \not| x \\ \lfloor x/y \rfloor \dot{-} 1 & \text{ha } y | x \end{cases} \qquad (\text{Így: } Q^+(x,0) = 0.)$$

Ha $y \neq 0$, érvényes:

$$x/y = Q^+(x,y) + R^+(x,y)/y.$$

Legyen most $x, n \in \mathbf{N}$ tetszőleges. Primitív rekurzióval definiáljuk a $g : \mathbf{N}^3 \to \mathbf{N}$ függvényt:

$$\begin{aligned} g(0,n,x) &= x \\ g(m+1,n,x) &= Q^+(g(m,n,x),n). \end{aligned}$$

Legyen $x, n \geq 1$, és $w = s_{i_k} \ldots s_{i_0}$ az x n-alapú reprezentációja. Ekkor:

$$\begin{aligned} g(0,n,x) &= i_k n^k + \ldots + i_0, \\ g(1,n,x) &= i_k n^{k-1} + \ldots + i_1, \\ &\vdots \end{aligned}$$

$$
\begin{aligned}
g(k,n,x) &= i_k,\\
g(k+1,n,x) &= 0\\
g(k+2,n,x) &= 0\\
&\vdots
\end{aligned}
$$

Legyen $h(m,n,x) = R^+(g(m,n,x),n)$. Így, az $x,n \geq 1$ feltevéssel élve:

$$
\begin{aligned}
h(0,n,x) &= i_0\\
&\vdots\\
h(k,n,x) &= i_k\\
h(k+1,n,x) &= 0\\
&\vdots
\end{aligned}
$$

Ha $x = 0$, akkor tetszőleges m-re és n-re $h(m,n,x) = 0$. Tehát, adott x-re a h primitív rekurzív függvény segítségével "meghatározható" az x n-alapú reprezentációja.

27 Parciális rekurzív függvények szavakon

Jelölés: Legyen $\tau : A^* \longrightarrow \mathbf{N}$ az a függvény, amely tetszőleges $w \in A^*$-hoz hozzárendeli a w szó által n-alapban jelölt számot.

Definíció 27.1 *(a) Legyen $f : A^{*m} \longrightarrow A^*$ m-változós (parciális) függvény. f (parciális) rekurzív, ha az alábbi $\widehat{f}$ (parciális) rekurzív:*

$$
\begin{tikzcd}
\mathbf{N}^m \arrow[r, "\widehat{f}"] \arrow[d, "(\tau^{-1})^m"'] & \mathbf{N} \arrow[d, "\tau^{-1}"] \\
A^{*m} \arrow[r, "f"'] & A^*
\end{tikzcd}
$$

(Itt a $(\tau^{-1})^m$ jelölést használtuk az $(x_1,\ldots,x_m) \mapsto (\tau^{-1}(x_1),\ldots,\tau^{-1}(x_m))$ bijekcióra is.) Világos, hogy $f : A^{*m} \longrightarrow A^*$ akkor és csakis akkor rekurzív, ha parciális rekurzív és mindenhol értelmezett.

(b) Legyen $f : A^{*m} \longrightarrow A^*$ egy függvény. f primitív rekurzív, ha $\widehat{f}$ primitív rekurzív. Világos, hogy minden primitív rekurzív függvény rekurzív.

(c) Legyen $f : A^{*m} \longrightarrow A^*$ egy függvény. f predikátum, ha $\widehat{f}$ az. Másszóval ez azt jelenti, hogy f értékkészletében csak λ és s_1 fordulnak elő. Legyen $f : A^{*m} \longrightarrow A^*$ predikátum. f rekurzív (primitív rekurzív), ha rekurzív (primitív rekurzív) függvény is.

(d) *Legyen $B \subseteq A^*$. B rekurzívan felsorolható (rekurzív), ha $\tau(B)$ rekurzívan felsorolható (rekurzív). Könnyen látható, hogy $B \subseteq A^*$ akkor és csakis akkor rekurzívan felsorolható, ha $B = \{w \in A^* | f(w) \downarrow\}$ valamely $f : A^* \longrightarrow A^*$ parciális rekurzív függvényre. Továbbá, valamely $B \subseteq A^*$ rekurzív $\Leftrightarrow$ karakterisztikus függvénye rekurzív.*

Megjegyzés 27.2 A fenti definíciók kiterjeszthetők $f : A^{*m} \longrightarrow \mathbf{N}$ és $\mathbf{N}^m \longrightarrow A^*$ alakú parciális függvényekre is.

Példa 27.3 1. $f(u) = |u| \quad (A^* \longrightarrow \mathbf{N}$ alakú)

$$|u| \quad = \quad \min_{x \leq u} \left[\sum_{j=0}^{x} n^j > x \right] \qquad \text{(mint } \mathbf{N} \longrightarrow \mathbf{N} \text{ alakú)}$$

2. $CONCAT_n(u, v) = uv \quad (A^{*2} \longrightarrow A^*$ alakú)

$$CONCAT_n(u, v) \quad = \quad u \cdot n^{|v|} + v \qquad \text{(mint } \mathbf{N}^2 \longrightarrow \mathbf{N} \text{ alakú)}$$

3. $CONCAT_n^{(m)}(u_1, \ldots, u_m)$

$$CONCAT_n^{(m)} \quad = \quad CONCAT_n(u_1, CONCAT_n(u_2, \ldots (CONCAT_n(u_{m-1}, u_m))$$

4. $RTEND_n(w) = \begin{cases} w \text{ utolsó betűje} & \text{ha } w \neq \lambda \\ \lambda & \text{különben} \end{cases}$

$$RTEND_n(w) \quad = \quad h(0, n, w)$$

5. $LTEND_n(w) = \begin{cases} w \text{ első betűje} & \text{ha } w \neq \lambda \\ \lambda & \text{különben} \end{cases}$

$$LTEND_n(w) \quad = \quad h(|w| \dot{-} 1, n, w)$$

6. $RTRUNC_n(w) = \begin{cases} \text{a } w \text{ utolsó betűje elhagyásával keletkezett szó} & \text{ha } w \neq \lambda \\ \lambda & \text{különben} \end{cases}$

$$RTRUNC_n(w) \quad = \quad g(1, n, w)$$

Megjegyzés: A w^- jelölést is használjuk.

7. $LTRUNC_n(w) = \begin{cases} \text{a } w \text{ első betűjének elhagyásával keletkezett szó} & \text{ha } w \neq \lambda \\ \lambda & \text{különben} \end{cases}$

$$LTRUNC_n(w) \quad = \quad w \dot{-} LTEND_n(w) \cdot n^{|w| \dot{-} 1})$$

8. Legyen $1 \leq n \leq l$, $A = \{s_1, \ldots, s_n\}$, $B = \{s_1, \ldots, s_l\}$, tehát $A \subseteq B$. Legyen $x \in \mathbf{N}$, és tekintsük az x numerikus reprezentációját A^*-ban, legyen ez w. Legyen $y \in \mathbf{N}$ az a szám, amelyet w reprezentál, ha w-t mint B^* elemét tekintjük. Az $x \mapsto y$ hozzárendeléssel kapott $\mathbf{N} \longrightarrow \mathbf{N}$ alakú függvényt az $UPCHANGE_{n,l}$ szimbólummal jelöljük. Tehát:

$$UPCHANGE_{n,l}(x) \quad = \quad y$$

Fordítva, legyen $x \in \mathbf{N}$, és legyen w az x reprezentációja B^*-ban. Legyen w' az a szó, amelyet úgy kapunk, hogy töröljük a w-ben előforduló $B - A$-beli betűket. Végül legyen y az a szám, amelyet w' mint A^* eleme reprezentál. Ekkor:

$$DOWNCHANGE_{n,l}(x) \quad = \quad y$$

Bár be lehetne látni, hogy a fenti függvények primitív rekurzívak, csak annyit mutatunk meg, hogy rekurzívak.

$$\begin{array}{ll} A & \texttt{if } X = 0 \texttt{ goto } E \\ & Z \leftarrow LTEND_n(X) \\ & X \leftarrow LTRUNC_n(X) \\ & Y \leftarrow l \cdot Y + Z \\ & \texttt{goto } A \end{array} \qquad \begin{array}{ll} A & \texttt{if } X = 0 \texttt{ goto } E \\ & Z \leftarrow LTEND_l(X) \\ & X \leftarrow LTRUNC_l(X) \\ & \texttt{if } Z > n \texttt{ goto } A \\ & Y \leftarrow n \cdot Y + Z \\ & \texttt{goto } A \end{array}$$

28 Egy programozási nyelv szavakon

Jelölje $\mathcal{L}$ az ($\mathbf{N}$-felett) eddig bevezetett programozási nyelvet. Szavakon értelmezett függvények kiszámítására az $\mathcal{L}$ nyelv egyáltalán nem természetes. Ezért most bevezetünk a szavak fölött egy programozási nyelvet, pontosabban minden $n > 0$-ra egy $\mathcal{L}_n$ programozási nyelvet az A n-elemű halmaz felett. Az utasítások:

$$\begin{array}{ll} V \leftarrow \sigma V & (\sigma \in A = \{s_1, \ldots, s_n\}) \\ V \leftarrow V^- & \\ \texttt{if } V \texttt{ ends } \sigma \texttt{ goto } L & \end{array}$$

A szintaxis és szemantika szabályai az $\mathcal{L}$-hez hasonlóak.

Definíció 28.1 *Legyen $f : A^{*m} \longrightarrow A$ parciális függvény. f parciális rekurzív $\mathcal{L}_n$-ben, ha van olyan $\mathcal{L}_n$-beli program, amely által kiszámított m-változós függvény éppen f. Ha emellett még f teljesen definiált is, akkor f rekurzív $\mathcal{L}_n$-ben.*

Néhány makró $\mathcal{L}_n$-ben:

1. $\texttt{if } V \neq \lambda \texttt{ goto } L$ (vagy másképpen: $\texttt{if } V \neq 0 \texttt{ goto } L$)

$$\begin{array}{c} \texttt{if } V \texttt{ ends } s_1 \texttt{ goto } L \\ \vdots \\ \texttt{if } V \texttt{ ends } s_n \texttt{ goto } L \end{array}$$

2. $V \leftarrow \lambda$ (vagy másképpen: $V \leftarrow 0$)

$$\begin{array}{ll} A & V \leftarrow V^- \\ & \texttt{if } V \neq \lambda \texttt{ goto } A \end{array}$$

3. goto L

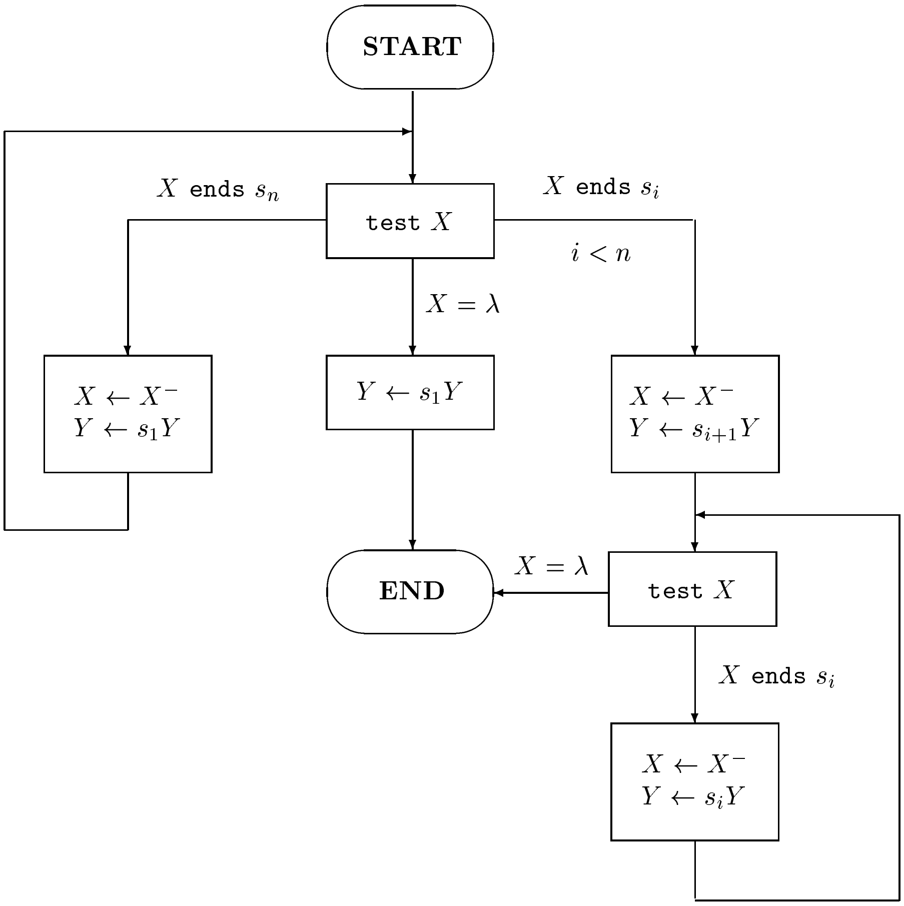
$$
\begin{array}{l}
Z \leftarrow \lambda \\
Z \leftarrow s_1 Z \\
\texttt{if } Z \texttt{ ends } s_1 \texttt{ goto } L
\end{array}
$$

4. $V' \leftarrow V$

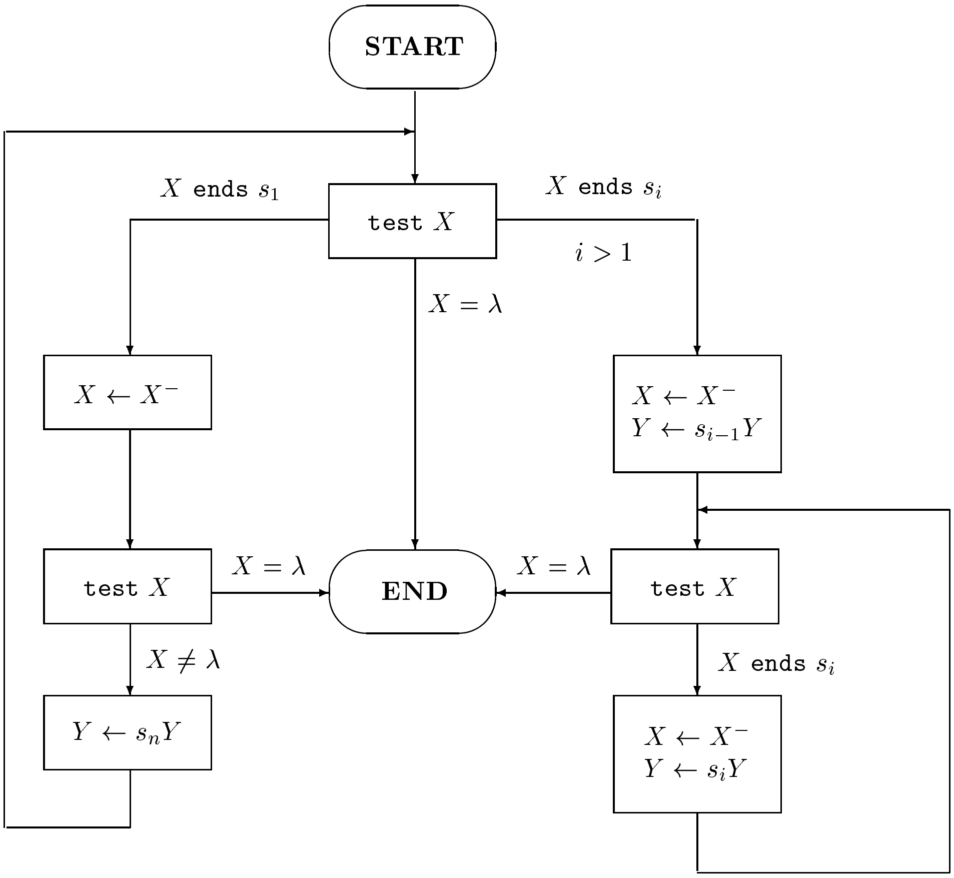
$$
\begin{array}{lll}
 & Z \leftarrow \lambda & \\
 & V' \leftarrow \lambda & \\
 & \texttt{goto } C & \\
A & \texttt{if } V \texttt{ ends } s_i \texttt{ goto } B_i & (i = 1, \ldots, n) \\
B_i & \left.\begin{array}{l} V \leftarrow V^- \\ V' \leftarrow s_i V' \\ Z \leftarrow s_i Z \\ \texttt{goto } A \end{array}\right\} & (i = 1, \ldots, n) \\
C & \texttt{if } Z \texttt{ ends } s_i \texttt{ goto } D_i & (i = 1, \ldots, n) \\
 & \texttt{goto } E & \\
D_i & \left.\begin{array}{l} Z \leftarrow Z^- \\ V \leftarrow s_i V \\ \texttt{goto } C \end{array}\right\} & (i = 1, \ldots, n)
\end{array}
$$

Két példa $\mathcal{L}_n$-ben kiszámítható függvényre:

1. $x + 1$



2. $x \dot- 1$



29 $\mathcal{L}$ szimulálása $\mathcal{L}_n$-ben

Tétel 29.1 *Egy $f : A^{*m} \longrightarrow A^*$ parciális függvény akkor és csakis akkor (parciális) rekurzív, ha (parciális) rekurzív $\mathcal{L}_1$-ben.*

Bizonyítás. $\mathcal{L}$ és $\mathcal{L}_1$ ekvivalensek:

$$\begin{aligned} V \leftarrow V+1 \quad &\sim \quad V \leftarrow s_1 V \\ V \leftarrow V-1 \quad &\sim \quad V \leftarrow V^- \\ \texttt{if } V \neq 0 \texttt{ goto } L \quad &\sim \quad \texttt{if } V \texttt{ ends } s_1 \texttt{ goto } L \quad \square \end{aligned}$$

Tétel 29.2 *Ha egy $f : A^{*m} \to A^*$ parciális függvény (parciális) rekurzív, akkor (parciális) rekurzív $\mathcal{L}_n$-ben.*

Bizonyítás. Láttuk, hogy $V \leftarrow V + 1$ és $V \leftarrow V - 1$ megvalósíthatók $\mathcal{L}_n$-beli makrókkal, az `if` $V \neq 0$ `goto` L utasításnak pedig megfelel az `if` $V \neq \lambda$ `goto` L makró. □

30 Post-Turing programok

Bevezetünk egy újabb programozási nyelvet. Az eddigiekkel szemben ezen $\mathcal{T}$ programozási nyelv nem tartalmaz változókat: az összes feldolgozandó információ egy lineáris szalagon van, amely potenciálisan végtelen. A szalag mezőkre van osztva, és minden mező egyetlen szimbólumot tartalmaz. Az utasítások:

$$\begin{array}{ll} \texttt{print}\ \sigma & \sigma \in \{s_0, s_1, \ldots\} \\ \texttt{if}\ \sigma\ \texttt{goto}\ L & \\ \texttt{right} & \\ \texttt{left} & \end{array}$$

Az s_0 szimbólumot `blank`-nek hívjuk, és B-vel is jelöljük. A szalag mindig csak véges sok nem `blank` szimbólumot tartalmaz. Ennek megfelelően egy konfiguráció egy

$$\begin{array}{c} s_{i_1} \ldots s_{i_j} \ldots s_{i_k} \\ \uparrow \end{array}$$

szó: az s_{i_1}-től balra lévő szimbólumok mindegyike `blank`, az s_{i_k}-tól jobbra lévő szimbólumok mindegyike szintén `blank`, $\uparrow$ pedig az éppen feldolgozás alatt lévő karakterre mutat.

Definíció 30.1 *Legyen $A = \{s_1, \ldots, s_n\}$, $f : A^{*m} \to A^*$ parciális függvény, P pedig egy program. Azt mondjuk, hogy a P által kiszámított függvény f, ha tetszőleges $x_1, \ldots, x_m \in A^*$ esetén a*

$$\begin{array}{l} Bx_1Bx_2 \ldots Bx_m \\ \uparrow \end{array}$$

konfigurációval indulva P akkor és csakis akkor áll meg, ha $f(x_1, \ldots, x_m)$ értelmezett, és megálláskor a szalagról a nem A-beli karakterek leolvasásával előálló szó éppen $f(x_1, \ldots, x_m)$. (Megengedett ugyanis az, hogy P tartalmazzon olyan utasításokat, amelyek az $s_0, s_1, \ldots, s_n$ szimbólumoktól különböző szimbólumra hivatkoznak. Azt mondjuk, hogy P szigorú értelemben kiszámítja f-et, ha kiszámítja és:

(i) P utasításaiban legfeljebb az $s_0, \ldots, s_n$ szimbólumokra történik hivatkozás,

(ii) ha P megáll, akkor a szalagkonfiguráció $\begin{array}{l} By \\ \uparrow \end{array}$ alakú, ahol $y \in \{s_1, \ldots, s_n\}^*$ (tudniillik $y = f(x_1, \ldots, x_m)$).

Tétel 30.2 *Ha f parciális rekurzív $\mathcal{L}_n$-ben, akkor létezik olyan Post-Turing program, amely szigorú értelemben kiszámítja.*

Bizonyítás. Legyen P olyan $\mathcal{L}_n$-beli program, amely f-et számítja ki. Feltehető, hogy P legfeljebb az $X_1, \ldots, X_m, Z_1, \ldots, Z_k, Y$ változókat használja. Legyen $l = m + k + 1$, és írjuk ezeket a változókat mint $V_1, \ldots, V_l$. Készítünk egy Q Post-Turing programot, amely lépésenként szimulálja P működését. P-nek azt az állapotát, amelyre a változók értéke rendre $u_1, \ldots, u_l$, az alábbi szalagkonfiguráció fogja megadni:

$$\begin{array}{l} Bu_1Bu_2 \ldots Bu_l \\ \uparrow \end{array}$$

Tehát kezdetben a szalagkonfiguráció:

$$\begin{array}{l} Bx_1Bx_2 \ldots Bx_m. \\ \uparrow \end{array}$$

A szimuláció során felhasználjuk az alábbi makrókat:

$$\texttt{goto } L \sim \left\{ \begin{array}{ll} & \texttt{if } s_0 \texttt{ goto } L \\ & \quad \vdots \\ & \texttt{if } s_n \texttt{ goto } L \end{array} \right.$$

$$\texttt{right to next blank} \sim \left\{ \begin{array}{ll} A & \texttt{right} \\ & \texttt{if } B \texttt{ goto } E \\ & \texttt{goto } A \end{array} \right.$$

$$\texttt{left to next blank} \sim \left\{ \begin{array}{ll} A & \texttt{left} \\ & \texttt{if } B \texttt{ goto } E \\ & \texttt{goto } A \end{array} \right.$$

$$\texttt{move block right} \sim \left\{ \begin{array}{lll} C & \texttt{left} & \\ & \texttt{if } s_i \texttt{ goto } A_i & (i = 0, \ldots, n) \\ A_i & \left.\begin{array}{l} \texttt{right} \\ \texttt{print } s_i \\ \texttt{left} \\ \texttt{goto } C \end{array}\right\} & (i = 1, \ldots, n) \\ A_0 & \texttt{right} & \\ & \texttt{print } B & \\ & \texttt{left} & \end{array} \right.$$

$$\texttt{erase block} \sim \left\{ \begin{array}{ll} A & \texttt{right} \\ & \texttt{if } B \texttt{ goto } E \\ & \texttt{print } B \\ & \texttt{goto } A \end{array} \right.$$

Ezután a P-t szimuláló Q programot úgy határozzuk meg, hogy P minden egyes utasítását helyettesítjük sorrendben egy utasítássorozattal.

$$V_j \leftarrow s_i V_j \quad \sim \quad \left\{ \begin{array}{ll} & \texttt{right to next blank} \quad [l] \\ & \texttt{move block right} \quad [l-j+1] \\ & \texttt{right} \\ & \texttt{print } s_i \\ & \texttt{left to next blank} \quad [j] \end{array} \right.$$

$$V_j \leftarrow V_j^- \quad \sim \quad \left\{ \begin{array}{ll} & \texttt{right to next blank} \quad [j] \\ & \texttt{left} \\ & \texttt{if } B \texttt{ goto } C \\ & \texttt{right} \\ & \texttt{print } s_0 \\ & \texttt{move block right} \quad [j] \\ & \texttt{right} \\ C & \texttt{left to next blank} \quad [j-1] \end{array} \right.$$

$$\texttt{if } V_j \texttt{ ends } s_i \texttt{ goto } L \quad \sim \quad \left\{ \begin{array}{ll} & \texttt{right to next blank} \quad [j] \\ & \texttt{left} \\ & \texttt{if } s_i \texttt{ goto } C \\ & \texttt{goto } D \\ C & \texttt{left to next blank} \quad [j] \\ & \texttt{goto } L \\ D & \texttt{right} \\ & \texttt{left to next blank} \quad [j] \end{array} \right.$$

Végül Q-t az alábbi módon zárjuk:

$$\texttt{erase block} \quad [l-1]$$

□

31 $\mathcal{T}$ szimulálása $\mathcal{L}$-ben

Tétel 31.1 *Ha egy $f : A^{*m} \longrightarrow A^*$ parciális függvény kiszámítható $\mathcal{T}$-beli P programmal, akkor f parciális rekurzív. (Emlékeztetőül: f definíció szerint parciális rekurzív, ha mint $\mathbf{N}^m \longrightarrow \mathbf{N}$ parciális függvény parciális rekurzív.)*

Tétel 31.2 *Legyen $A = \{s_1, \ldots, s_n\}, f : A^{*m} \longrightarrow A^*$ parciális függvény. Az alábbiak ekvivalensek:*

1. *f parciális rekurzív*
2. *f parciális rekurzív $\mathcal{L}_n$-ben*

3. f (szigorú értelemben) kiszámítható egy Post-Turing programmal.

A 31.2 Tétel az 31.1 Tétel és korábbi eredményeink következménye. A 31.2 Tételhez hasonló állítás mondható ki $\mathbf{N}^m \longrightarrow \mathbf{N}$ függvényekre is.

Következmény 31.3 *Legyen $f : \mathbf{N}^m \longrightarrow \mathbf{N}$ parciális függvény, $A = \{s_1, \ldots, s_n\}$, $n \geq 1$. f-et felfoghatjuk, mint $A^{*m} \longrightarrow A^*$ parciális függvényt. Az alábbiak ekvivalensek:*

1. *f parciális rekurzív*
2. *f parciális rekurzív $\mathcal{L}_n$-ben*
3. *f (szigorú értelemben) kiszámítható Post-Turing programmal.*

A 31.1 Tétel bizonyítása. Feltevésünk szerint f kiszámítható a $\mathcal{T}$-beli P programmal. Elkészítünk egy olyan Q programot $\mathcal{L}$-ben, mely szimulálja P-t. Tegyük fel, hogy P az $s_1, \ldots, s_n, \ldots, s_r, B$ szimbólumokat használja legfeljebb. Legyen $A' = \{s_1, \ldots, s_r, B\}$ ebben az elrendezésben, most az $s_{r+1} = B$ jelölést is használjuk. A P szalagkonfigurációit három változóban tárolja Q:

$$\begin{array}{ccl} L & \sim & \text{a fejtől balra lévő szó numerikus értéke } r+1 \text{ alapban} \\ R & \sim & \text{a fejtől jobbra lévő szó numerikus értéke } r+1 \text{ alapban} \\ H & \sim & \text{a fej alatt lévő jel numerikus értéke } r+1 \text{ alapban.} \end{array}$$

(A szavakat mindig egy olyan pontig tekintjük, ahonnan már csupa B áll. Ez nem feltétlen az első ilyen!) Az utasítások helyettesítése:

$$\begin{array}{rcl} \texttt{print } s_i & \sim & H \leftarrow i \\ \texttt{if } s_i \texttt{ goto } L & \sim & \texttt{if } H = i \texttt{ goto } L \\ \texttt{right} & \sim & \left\{ \begin{array}{l} L \leftarrow CONCAT_{r+1}(L, H) \\ H \leftarrow \left\{ \begin{array}{ll} LTEND_{r+1}(R) & \text{ha } R \neq 0 \\ r+1 & \text{ha } R = 0 \end{array} \right. \\ R \leftarrow LTRUNC_{r+1}(R) \end{array} \right. \\ \texttt{left} & \sim & \left\{ \begin{array}{l} L \leftarrow RTRUNC_{r+1}(L) \\ H \leftarrow \left\{ \begin{array}{ll} RTEND_{r+1}(L) & \text{ha } L \neq 0 \\ r+1 & \text{ha } L = 0 \end{array} \right. \\ R \leftarrow CONCAT_{r+1}(H, R) \end{array} \right. \end{array}$$

Ezek után a Q programot a következő módon állítjuk össze.

1. Kezdetben P szalagkonfigurációja $\underset{\uparrow}{B}x_1 \ldots Bx_m$ alakú. A szalagtartalmat fel kell jegyezni az

L, H, R változ'okban $r+1$ alapban. Persze P inputja az $X_1, \ldots, X_m$ változókban van.

$$
\begin{aligned}
& L \leftarrow 0 \\
& H \leftarrow r+1 \\
& Z_1 \leftarrow UPCHANGE_{n,r+1}(X_1) \\
& \vdots \\
& Z_m \leftarrow UPCHANGE_{n,r+1}(X_m) \\
& R \leftarrow CONCAT_{r+1}(Z_1, r+1, \ldots, r+1, Z_m)
\end{aligned}
$$

2. P minden utasítását helyettesítjük a megfelelő programrésszel.

3. Végül:

$$
\begin{aligned}
& Z \leftarrow CONCAT_{r+1}(L, H, R) \\
& Y \leftarrow DOWNCHANGE_{n,r+1}(Z)
\end{aligned}
$$

32 Turing gépek

Rögzítjük jeleknek (vagy betűknek) egy $\{s_0, s_1, s_2, \ldots\}$ halmazát és állapotoknak egy $\{q_1, q_2, \ldots\}$ halmazát. Az s_0-ra ismét használjuk a B jelölést.

Definíció 32.1 *Négyesekben megadott Turing gép véges sok szabályból áll, amelyek az alábbi formájúak lehetnek:*

1. $q_i s_j s_k q_l$
2. $q_i s_j R q_l$
3. $q_i s_j L q_l$

A szabályok értelmezése:

1. *ha q_i állapotban s_j az olvasott jel, akkor írjunk s_k-t és gyerünk a q_l állapotba,*
2. *ha q_i állapotban s_j az olvasott jel, az író -olvasó fejet mozgassuk egy beosztással jobbra és gyerünk a q_l állapotba*
3. *Mint* (2) *azzal a különbséggel, hogy a fej mozgatása balra történik.*

Kikötés még, hogy bármely (q_i, s_j) párra legfeljebb egy szabály lehet (determinisztikus Turing gép). Valamely Turing gép mindig a q_1 állapotból indul, és működése akkor fejeződik be, ha olyan q_i állapotba ér, hogy az olvasott s_j jelre nem létezik $q_i s_j$ kezdetű szabály. A Post-Turing programokhoz hasonlóan definiálhatjuk egy Turing-gép által valamely $A = \{s_1, \ldots, s_n\}$ felett kiszámított m-változós és a szigorú értelemben kiszámított m-változós parciális függvény fogalmát.

Tétel 32.2 *Ha egy $f : A^{*m} \longrightarrow A^*$ függvény parciális rekurzív (azaz kiszámítható Post-Turing programmal = szigorú értelemben kiszámítható Post-Turing programmal), akkor f kiszámítható (szigorú értelemben kiszámítható) Turing géppel.*

Bizonyítás. Egy P Post-Turing programot kell szimulálnunk egy M Turing géppel. Legyenek P utasításai $I_1, \ldots, I_k$, és legyen $s_0, \ldots, s_l$ a betűk olyan listája, amely tartalmazza az összes P-ben és A-ban előforduló betűt. Ezután M szabályait a következő eljárással kapjuk:

$$\begin{array}{lcll}
I_i : \texttt{print } s_r & \sim & q_i s_j s_r q_{i+1} & j = 0, \ldots, l \\
I_i : \texttt{right} & \sim & q_i s_j R q_{i+1} & j = 0, \ldots, l \\
I_i : \texttt{left} & \sim & q_i s_j L q_{i+1} & j = 0, \ldots, l \\
I_i : \texttt{if } s_r \texttt{ goto } L & \sim & q_i s_r s_r q_t & \\
 & & q_i s_j s_j q_{i+1} & j = 0, \ldots, l, \; j \neq r,
\end{array}$$

ahol t az első olyan utasítás sorszáma, melynek címkéje L, ha ilyen van, különben $t = k + 1$. □

Definíció 32.3 *Egy ötösökkel megadott Turing gép szabályai az alább alakúak lehetnek:*

1. $q_i s_j s_k R q_l$
2. $q_i s_j s_k L q_l$

(1) jelentése: ha q_i állapotban s_j az olvasott jel, írjunk s_k-t, majd gyerünk jobbra és a q_l állapotba. (2) jelentése hasonló. Ismét kikötés az, hogy nem lehet két különböző $q_i s_j$ kezdetű szabály. Valamely ötösökkel adott Turing gép által kiszámított (szigorúan értelmezett kiszámított) parciális függvény fogalmát értelem szerűen definiáljuk.

Tétel 32.4 *Ha egy $f : A^{*m} \longrightarrow A^*$ parciális függvény kiszámítható négyesekkel adott Turing géppel akkor kiszámítható ötösökkel adott Turing géppel is. Ugyanez igaz szigorú értelemben vett Turing gépre is.*

Bizonyítás. Legyen M négyesekkel adott Turing gép. Legyen $\{s_0, s_1, \ldots, s_n\}$ olyan A-t tartalmazó betűhalmaz, hogy M-ben nem fordul elő $s_0, \ldots, s_n$-en kívüli betű. Legyenek $q_1, \ldots, q_K$ az M szabályaiban előforduló állapotok. Készítünk egy M' ötösökkel adott Turing gépet, mely szimulálja M-et. M' állapotai: $q_1, \ldots, q_K, q_{K+1}, \ldots, q_{2K}$.

$$\begin{array}{lcll}
q_i s_j R q_l & \sim & q_i s_j s_j R q_l & \\
q_i s_j L q_l & \sim & q_i s_j s_j L q_l & \\
q_i s_j s_k q_l & \sim & q_i s_j s_k R q_{K+i} & \\
 & & q_{K+i} s_t s_t L q_l & t = 0, \ldots, n
\end{array}$$

Tétel 32.5 *Ha egy $f : A^{*m} \longrightarrow A^*$ parciális függvény kiszámítható (szigorú értelemben kiszámítható) ötösökkel adott Turing géppel, akkor az kiszámítható (szigorú értelemben kiszámítható) Post-Turing programmal.*

Bizonyítás. Legyen M ötösökkel adott Turing gép, mely állapotai $q_1, \ldots, q_k$, és amely legfeljebb az $s_0, \ldots, s_n$ betűket használja. Tegyük fel, hogy $A \subseteq \{s_1, \ldots, s_n\}$. Minden q_i állapothoz hozzárendelünk egy A_i címkét és minden (q_i, s_j) párhoz egy B_{ij} címkét.

$$\begin{array}{ll} A_i & \texttt{if } s_0 \texttt{ goto } B_{i0} \\ & \quad\vdots \\ & \texttt{if } s_n \texttt{ goto } B_{in} \end{array}$$

Ha M tartalmazza a $q_i s_j s_k R q_l$ szabályt, akkor

$$\begin{array}{ll} B_{ij} & \texttt{print } s_k \\ & \texttt{right} \\ & \texttt{goto } A_l \end{array}$$

Ha M tartalmazza a $q_i s_j s_k L q_l$ szabályt, akkor

$$\begin{array}{ll} B_{ij} & \texttt{print } s_k \\ & \texttt{left} \\ & \texttt{goto } A_l \end{array}$$

Végül, ha nincs $q_i s_j$ kezdetű szabály, akkor

$$B_{ij} \ \ \texttt{goto } E.$$

□

Következmény 32.6 *Az alábbi állítások ekvivalensek tetszőleges $f : A^{*m} \to A^*$ parciális függvényre:*

1. *f parciális rekurzív*
2. *f (szigorú értelemben) kiszámítható négyesekkel adott Turing géppel*
3. *f (szigorú értelemben) kiszámítható ötösökkel adott Turing géppel.*

Következmény 32.7 *Az alábbi állítások ekvivalensek tetszőleges $f : \mathbf{N}^m \to \mathbf{N}$ parciális függvényre és $A = \{s_1, \ldots, s_n\}$ halmazra*

1. *f parciális rekurzív*
2. *f (szigorú értelemben) kiszámítható olyan Turing géppel, amely legfeljebb az $s_0, \ldots, s_n$ karaktereket használja*

2-ben négyesekkel és ötösökkel adott Turing gépre is gondolhatunk.

33 Univerzális Turing gépek

Láttuk, hogy a $\varphi^{(m)}(x_1,\ldots,x_m,z)$ függvények parciális rekurzívak. Így $\varphi^{(m)}$ kiszámítható Turing géppel, bármelyik, így pl. az egybetűs $A=\{s_1\}$ halmaz felett is. A $\varphi^{(m)}$-et kiszámító Turing gépeket univerzális Turing gépeknek nevezzük. Az univerzális Turing gépet felfoghatjuk a számítógépek egy matematikai modelljeként.

34 Turing géppel felismert nyelvek

Definíció 34.1 *Legyen M Turing gép és $f : A^* \longrightarrow A^*$ az M által kiszámított (szigorú értelemben kiszámított) parciális függvény. Legyen L az f értelmezési tartománya. L az M által (A felett) felismert (szigorú értelemben felismert) nyelv.*

A Turing géppel kiszámítható függvények egybeesése a parciálisan rekurzív függvényekkel adja az alábbi eredményt.

Következmény 34.2 *Az alábbi állítások ekvivalensek tetszőleges $L \subseteq A^*$-ra.*

1. *L rekurzívan felsorolható,*
2. *L felismerhető Turing géppel,*
3. *L szigorú értelemben felismerhető Turing géppel.*

Következmény 34.3 *Legyen $B \subseteq \mathbf{N}$ és tekintsük B-t úgy is, mint A^* részhalmazát. Az alábbi állítások ekvivalensek:*

1. *B rekurzívan felsorolható,*
2. *B felismerhető Turing géppel,*
3. *B szigorú értelemben felismerhető Turing géppel.*

35 Turing gépek megállási problémája

Tétel 35.1 *Létezik olyan Turing gép az $\{s_1\}$ halmaz felett (tehát legfeljebb az s_0, s_1 betűk fordulhatnak elő a szabályokban), amely megállási problémája algoritmikusan eldönthetetlen.*

Bizonyítás. Legyen $U \subseteq \mathbf{N}$ olyan rekurzívan felsorolható halmaz, amely nem rekurzív. U mint ${s_1}^*$ részhalmaza rekurzívan felsorolható, de nem rekurzív. Legyen M olyan Turing gép, amely U-t (szigorú értelemben) felismeri. M megállási problémája algoritmikusan megoldhatatlan. $\square$

36 A Turing gép néhány variánsa

1. Turing gép egyirányban végtelen szalaggal
2. Turing gép több író olvasó fejjel
3. "Off-line" Turing gép
4. Több szalagos Turing gép
5. Több dimenziós Turing gép
6. Nemdeterminisztikus Turing gép.

Tétel 36.1 *Egy L nyelv akkor és csak akkor ismerhető fel (szigorú értelemben) nemdeterminisztikus Turing géppel, ha felismerhető determinisztikus Turing géppel.*

37 Thue rendszerek

Definíció 37.1 *Legyen A egy véges, nemüres halmaz. A feletti egyirányú Thue rendszer véges sok*

$$g \longrightarrow \overline{g}$$

átírási szabály halmaza, ahol $g, \overline{g} \in A^*$.

Definíció 37.2 *Legyen Π egyirányú Thue rendszer az A felett. Legyen $u, v \in A^*$. Ekkor:*

$$\begin{array}{lll} u \Rightarrow v & \Leftrightarrow & \exists u_1, u_2, g, \overline{g} : \ u = u_1 g u_2, \ v = u_1 \overline{g} u_2, \ g \longrightarrow \overline{g} \in \Pi. \\ u \overset{*}{\Rightarrow} v & \Leftrightarrow & \exists u_0, \ldots, u_n : \ u = u_0 \Rightarrow u_1 \Rightarrow \ldots \Rightarrow u_n = v. \end{array}$$

Definíció 37.3 *Thue rendszer egy olyan egyirányú Π Thue rendszer, hogy valahányszor $g \longrightarrow \overline{g}$ a Π-ben van, mindig teljesül, hogy a fordított $\overline{g} \longrightarrow g$ szabály is Π-ben van.*

A Thue rendszereket néha kétirányú Thue rendszereknek is nevezik.

38 Nemdeterminisztikus Turing gép szimulálása egyirányú Thue rendszerrel

Legyen M egy Turing gép, mely

állapotai: $q_1, \ldots, q_n$
felhasznált betűk: $s_0, \ldots, s_m$.

Elkészítünk az $A = \{q_0, \ldots, q_{n+1}, s_0, \ldots, s_m, h\}$ halmaz felett egy olyan $\Sigma(M)$ egyirányú Thue-rendszert, mely bizonyos értelemben szimulálja M működését.

Definíció 38.1 *Post szó: huq_ivh alakú szó, ahol $u, v \in \{s_0, \ldots, s_m\}^*$. Minden Post szónak megfelel az M egy konfigurációja, pld.:*

$$\begin{array}{rcl} hs_0s_1s_1q_is_2h & \mapsto & \ldots Bs_0s_1s_1\underset{\substack{\uparrow \\ q_i}}{s_2}B\ldots \\ hs_0q_ih & \mapsto & \ldots Bs_0\underset{\substack{\uparrow \\ q_i}}{B}\ldots \end{array}$$

$\Sigma(M)$ Post-szavakon keresztül fogja szimulálni M-et. Az átírási szabályok:

$$\begin{array}{rcl} q_is_js_kq_l & \sim & q_is_j \longrightarrow q_ls_k \\ q_is_jRq_l & \sim & \left\{ \begin{array}{ll} q_is_js_k \longrightarrow s_jq_ls_k & k = 0, \ldots, m \\ q_is_jh \longrightarrow s_jq_ls_0h & \end{array} \right. \\ q_is_jLq_l & \sim & \left\{ \begin{array}{ll} s_kq_is_j \longrightarrow q_ls_ks_j & k = 0, \ldots, m \\ hq_is_j \longrightarrow hq_ls_0s_j & \end{array} \right. \\ \textit{nincs } q_is_j \textit{ kezdetű szabálya } M\textit{-nek} & \sim & q_is_j \longrightarrow q_{n+1}s_j. \end{array}$$

Végül mindig vesszük az összes alábbi átírási szabályt:

$$\begin{array}{ll} q_{n+1}s_i \longrightarrow q_{n+1} & i = 0, \ldots, m \\ q_{n+1}h \longrightarrow q_0h & \\ s_iq_0 \longrightarrow q_0 & i = 0, \ldots, m. \end{array}$$

Tétel 38.2 *$hq_1s_0uh \overset{*}{\Rightarrow}_{\Sigma(M)} hq_0h \Leftrightarrow M$ elfogadja u-t. ($u \in \{s_1, \ldots, s_m\}^*$)*

Definíció 38.3 *Egyegyirányú Thue rendszer inverzét úgy kapjuk, hogy minden szabályát megfordítjuk.*

Következmény 38.4 *Legyen $\Omega(M)$ a $\Sigma(M)$ inverze. $hq_0h \overset{*}{\Rightarrow}_{\Omega(M)} hq_1s_0uh \Leftrightarrow M$ elfogadja U-t.*

Következmény 38.5 *Nem létezik olyan algoritmus, amely tetszőleges (egyirányú) Thue rendszerre és u, v szavakra eldöntené, hogy $u \overset{*}{\Rightarrow} v$. Sőt, létezik egy olyan egyirányú Thue rendszer, amelyre ez a probléma eldönthetetlen.*

Ilyenkor azt is mondjuk, hogy az egyirányú Thue rendszer szóproblémája eldönthetetlen.

Hasonlóan bizonyítható az alábbi állítás is.

Következmény 38.6 *Létezik olyan (kétirányú) Thue rendszer, amely szóproblémája eldönthetetlen.*

BIZONYÍTÁS. Tekintsük a $\Theta(M) = \Sigma(M) \cup \Omega(M)$ Thue rendszert, ahol az M által felismert nyelv nem rekurzív. □

39 Nyelvtanok

Definíció 39.1 *Nyelvtan egy olyan egyirányú Thue rendszer, melynek ábécéje két diszjunkt részhalmazra (nemterminálisok és ternimálisok) van particionálva úgy, hogy minden átírási szabály baloldala tartalmaz nemterminálist. A nemterminálisok halmazát általában V, a terminálisokét pedig T fogja jelölni. Kikötés az, hogy V és T egyike sem üres. Minden nyelvtan tartalmaz még egy kitüntetett nemterminálist is, amit általában S jelöl. S a kezdőszimbólum.*

Definíció 39.2 *Legyen $G = (V, T, S, P)$ nyelvtan, ahol P a szabályok halmaza.*

$$L(G) \quad = \quad \{u \in T^* | S \stackrel{*}{\Rightarrow} u\}.$$

Tétel 39.3 *Legyen $L \subseteq T^*$. Tegyük fel, hogy létezik az L nyelvet felismerő M nemdeterminisztikus Turing gép. Akkor L generálható nyelvtannal.*

Bizonyítás. Legyen $L = L(M)$, ahol M nemdeterminisztikus Turing gép. Legyen $T = \{s_1, \ldots, s_n\}$, és legyen $\{s_1, \ldots, s_m\}$ olyan halmaz, amelyre $n \leq m$, és amelyre teljesül az, hogy minden olyan betű, amely előfordul M szabályaiban, szerepel az $s_0, \ldots, s_m$ listán. Legyen M állapotainak halmaza $\{q_1 \ldots, q_k\}$. Az $\{s_0, \ldots, s_m, q_0, \ldots, q_{k+1}, h\}$ felett már megkonstruáltunk egy $\Omega(M)$ egyirányú Thue rendszert, hogy tetszőleges $u \in T^*$-ra:

$$u \in L(M) \quad \Leftrightarrow \quad hq_0h \stackrel{*}{\Rightarrow} hq_1s_0uh.$$

Legyen most $V = \{s_0, s_{n+1}, \ldots, s_m, q_0, \ldots, q_{k+1}, h, q, S\}$, és bővítsük a szabályokat az alábbiakkal:

$$\begin{array}{ll} S \rightarrow hq_0h & \\ hq_1s_0 \rightarrow q & \\ qs \rightarrow sq, & \text{minden } s \in T\text{-re} \\ qh \rightarrow \lambda. & \end{array}$$

Belátható, hogy az így kapott G nyelvtanra: $L(G) = L(M)$. □

Legyen most G egy rögzített nyelvtan, melyre $T = \{s_1, \ldots, s_n\}$, $V = \{V_1, \ldots, V_k\}$, $S = V_1$, és amelyben P a szabályok halmaza, mondjuk $P : g_1 \rightarrow h_1, \ldots, g_l \rightarrow h_l$. Az $\{s_1, \ldots, s_n, V_1, \ldots, V_k\}$ feletti nyelveket úgy tekintjük, mint olyan szavakat, melyek $n+k$ alapban jelölnek számokat. Célunk annak megmutatása, hogy $L(G)$, mint $\mathbf{N}$ részhalmaza, rekurzívan felsorolható.

Lemma 39.4 *Az $u \Rightarrow v$ predikátum primitív rekurzív.*

Bizonyítás.

$$PROD_i(u,v) \quad \Leftrightarrow \quad (\exists r,s)_{\leq u}(u = CONCAT(r,g_i,s) \wedge v = CONCAT(r,h_i,s)). \quad \square$$

Világos, hogy

$$u \Rightarrow v \quad \Leftrightarrow \quad PROD_1(u,v) \vee \ldots \vee PROD_l(u,v).$$

Így $u \Rightarrow v$ is primitív rekurzív. Legyen $DERIV(u,y) \Leftrightarrow \exists m, y = [u_1, \ldots, u_m, 1]$, amelyre $u_1, \ldots, u_m$ az u derivációja S-ből. (Az 1-et azért tettük még a végére, hogy az $u_m = 0$ esetén előálló komplikációkat elkerüljük.)

Lemma 39.5 *$DERIV(u,y)$ primitív rekurzív.*

Bizonyítás.

$$\begin{aligned} DERIV(u,y) \quad \Leftrightarrow \quad & (\exists m)_{\leq y}(m+1 = Lt(y) \wedge (y)_1 = n+1 \wedge (y)_m = u \wedge (y)_{m+1} = 1 \wedge \\ & \wedge (\forall j)_{<m}(j = 0 \vee ((y)_j \Rightarrow (y)_{j+1}))). \end{aligned}$$

Tétel 39.6 *$L(G)$ rekurzívan felsorolható.*

Bizonyítás. A bizonyításban felhasználjuk még azt az egyszerű tényt, hogy T^*, mint $\mathbf{N}$ részhalmaza, rekurzív (sőt, primitív rekurzív is). Világos, hogy

$$S \overset{*}{\Rightarrow} u \quad \Leftrightarrow \quad \exists y DERIV(u,y) \downarrow,$$

ahol $u \in \mathbf{N}$ tetszőleges (vagyis $u \in \{s_1, \ldots, s_n, V_1, \ldots, V_k\}^*$). Másrészt

$$L(G) \quad = \quad T^* \cap \{u | S \Rightarrow^* u\}. \quad \square$$

Tétel 39.7 *Legyen $L \subseteq T^*$ tetszőleges nyelv. Az alábbiak ekvivalensek.*

1. *L rekurzívan felsorolható*
2. *L felismerhető determinisztikus Turing géppel*
3. *L felismerhető nemdeterminisztikus Turing géppel*
4. *Létezik olyan G nyelvtan, amelyre $L = L(G)$.*

40 A bonyolultság Blum-féle elmélete

Definíció 40.1 *A $C(x,i) = C_i(x)$ kétváltozós parciális függvény bonyolultsági mérték, ha kielégíti az alábbi Blum-féle axiómákat:*

- (i) $C_i(x)\downarrow \quad \Leftrightarrow \quad \varphi_i(x)\downarrow$.
- (ii) *A $C_i(x) \leq y$ (háromváltozós) predikátum rekurzív.*

Megjegyzés 40.2 A $C_i(x) \leq y$ predikátum természetesen hamis, ha $C_i(x)\uparrow$. Ilyenkor azt is mondjuk, hogy $C_i(x) = \infty$.

Példa 40.3

1. Legyen $C_i(x)$ a P program futásának lépésszáma x input mellett, ahol $\#(P) = i$, ha P megáll x-en. Különben legyen $C_i(x) = \infty$. Ekkor $C_i(x)$ bonyolultsági mérték.
2. Legyen $M_i(x)$ a fenti P változói által felvett értékek maximuma, ha P megáll x-en. Különben legyen $M_i(x) = \infty$. Ekkor $M_i(x)$ bonyolultsági mérték.

Megjegyzés 40.4 Az előző definícióban adott második tulajdonság ekvivalens az alábbiak bármelyikével.

- $(\mathrm{ii})_1$ A $C_i(x) = y$ predikátum rekurzív.
- $(\mathrm{ii})_1$ A $C_i(x) < y$ predikátum rekurzív.

Valóban, ha a $C_i(x) \leq y$ predikátum rekurzív, akkor a

$$C_i(x) = y \quad \Leftrightarrow \quad C_i(x) \leq y \wedge (y = 0 \vee \neg(C_i(x) \leq y - 1))$$

összefüggés miatt $(\mathrm{ii})_1$ teljesül. Ha pedig $(\mathrm{ii})_1$ teljesül, akkor

$$C_i(x) < y \quad \Leftrightarrow \quad (\exists z)_{<y}(C_i(x) = y)$$

miatt a $C_i(x) < y$ predikátum rekurzív. Végül ha ez utóbbi teljesül, akkor

$$C_i(x) \leq y \quad \Leftrightarrow \quad C_i(x) < y + 1$$

miatt (ii) teljesül.

Állítás 40.5 *Minden bonyolultsági mérték parciális rekurzív függvény.*

Bizonyítás. Döntsük el $y = 0, 1, 2, \ldots$ esetén végtelen ciklusban, hogy $C_i(x) = y$ teljesül-e. □

Állítás 40.6 *Legyen $C_i(x)$ bonyolultsági mérték, $r(x,y)$ pedig olyam rekurzív függvény, hogy $r(x,y) \leq r(x,y+1)$ és $\lim_{y\longrightarrow\infty} r(x,y) = \infty$. Ekkor $D_i(x) = r(x, C_i(x))$ is bonyolultsági mérték.*

Bizonyítás. Világos, hogy $D_i(x) \downarrow \Leftrightarrow C_i(x) \downarrow \Leftrightarrow \varphi_i(x) \downarrow$. Másrészt az $r(x, C_i(x)) \leq y$ predikátum rekurzív, hiszen $r(x, C_i(x)) \leq y$ akkor és csakis akkor teljesül, ha $C_i(x) < [\min_z r(x,z) > y]$. □

Tétel 40.7 A bonyolultsági mértékek rekurzív kapcsolata. *Legyenek C és D bonyolultsági mértékek. Ekkor létezik olyan $r(x,y)$ rekurzív függvény, hogy $r(x,y) < r(x,y+1)$ és tetszőleges i esetén*

1. *$C_i(x) \leq r(x, D_i(x))$ majdnem minden x-re,*
2. *$D_i(x) \leq r(x, C_i(x))$ majdnem minden x-re.*

Megjegyzés 40.8 Megegyezünk abban, hogy $r(x, \infty) = \infty$. Így a fenti egyenlőtlenségek fennállnak akkor, ha $C_i(x) = \infty$, ami akkor és csak akkor teljesül, ha $D_i(x) = \infty$.

Bizonyítás. Előszöris vegyük észre, hogy $C_i(x) = \infty$ akkor és csak akkor, ha $D_i(x) = \infty$, így ebben az esetben a fenti egyenlőtlenségek érvényesek, bármi is az r függvény.

Legyen most

$$r(x,y) \quad = \quad y + \max\{C_j(x), D_j(x) | j \leq x \ \wedge \ (C_j(x) \leq y \ \vee \ D_j(x) \leq y)\}.$$

Tegyük fel, hogy valamely i-re és x-re $C_i(x) \downarrow$. Ekkor $D_i(x) \downarrow$, és $x \geq i$ esetén

$$\begin{aligned} r(x, D_i(x)) &= D_i(x) + \max\{C_j(x), D_j(x) | j \leq x \ \wedge \ (C_j(x) \leq D_i(x) \ \vee \ D_j(x) \leq D_i(x))\} \\ &\geq D_i(x) + \max\{C_i(x), D_i(x)\} \\ &\geq C_i(x). \end{aligned}$$

Nyilvánvaló, hogy $r(x,y) < r(x,y+1)$. Azt kell még belátnunk, hogy $r(x,y)$ rekurzív. De $r(x,y)$ kiszámításához először azt kell meghatároznunk, hogy mely $j \leq x$ számokra érvényes a $C_j(x) \leq y$ vagy $D_j(x) \leq y$ egyenlőtlenség. A második Blum-féle axióma miatt ez eldönthető. Ezek után minden egyes ilyen j számra ki kell számítanunk a $C_j(x)$ és $D_j(x)$ értékeket, amelyet azért tehetünk meg, mert C és D parciális rekurzív függvények és értelmezettek az adott j, x párra. Végül vennünk kell a véges sok kiszámított $C_j(x)$ és $D_j(x)$ érték maximumát, és ehhez hozzá kell adnunk y-t. (Amennyiben nincs olyan $j \leq x$ amelyre $C_j(x) \leq y$ vagy $D_j(x) \leq y$, akkor a maximum 0.) □

Az előző tétel jelentősége abban áll, hogy segítségével gyakran következtethetünk arra, hogy amennyiben egy állít'as érvényes egy konkrét bonyolultsági mértékre, akkor érvényes tetszőleges bonyolultsági mértékre is.

Tétel 40.9 Hézag tétel. *Legyen $g(x,y)$ tetszőleges olyan rekurzív függvény, amelyre $y < g(x,y)$, és legyen C bonyolultsági mérték. Megadható olyan $t(x)$ rekurzív függvény, hogy $x > i$ esetén $C_i(x) \leq t(x)$ vagy $C_i(x) > g(x, t(x))$.*

Bizonyítás. Legyen

$$P(x,y) \quad \Leftrightarrow \quad (\forall i)_{<x}(C_i(x) \leq y \ \vee \ g(x,y) < C_i(x)).$$

Ekkor $P(x,y)$ rekurzív predikátum. Be kell látnunk, hogy $\forall x \exists y = t(x)$ amelyre $P(x,y)$, és azt, hogy y az x rekurzív függvénye, azaz $t(x)$ rekurzív.

Legyen $t(x) = \min_y P(x,y)$. Ekkor $t(x)$ parciális rekurzív függvény, és ha mindenhol értelmezett, akkor rekurzív is. Tehát azt kell csak megmutatnunk, hogy $\forall x \exists y P(x,y)$. Adott x mellett legyen

$$Q \quad = \quad \{C_i(x) | i < x \ \wedge \ \varphi_i(x) \downarrow\}.$$

Legyen továbbá $y = \max Q$ (tehát $y = 0$, ha $Q = \emptyset$). Ha valamely $i < x$ számra $C_i(x) \downarrow$, akkor $\varphi_i(x) \downarrow$, és így $C_i(x) \leq y$. Ha pedig $C_i(x) \uparrow$, akkor természetesen $g(x,y) < C_i(x)$. Tehát $P(x,y)$ teljesül. □

A TÉTEL INTERPRETÁLÁSA.

Legyenek $C_i(x)$ és $D_i(x)$ bonyolultsági mértékek. Tudjuk, hogy létezik olyan $r(x,y)$ rekurzív függvény, amelyre $r(x,y) < r(x,y+1)$, és tetszőleges rögzített i mellett $C_i(x) \leq r(x, D_i(x))$ és $D_i(x) \leq r(x, C_i(x))$ majdnem minden x-re. Legyen $g(x,y) = r(x,y) + y + 1$. Ekkor létezik olyan $t(x)$ rekurzív függvény, amelyre tetszőleges i esetén

$$C_i(x) \leq t(x) \quad \Leftrightarrow \quad C_i(x) \leq g(x, t(x))$$

majdnem minden x-re. Legyen most i_0 olyan, hogy $D_{i_0}(x) \leq t(x)$ majdnem minden x-re. Ekkor majdnem minden x-re

$$C_{i_0}(x) \quad \leq \quad r(x, D_{i_0}(x)) \quad \leq \quad g(x, t(x)),$$

így, a hézag tétel szerint, $C_{i_0}(x) \leq t(x)$, majdnem minden x-re.

Tekintsünk most két olyan számítógépet, amelyen programjaink futtathatóak. Legyen

- $C_i(x) =$ az i Gödel számú program futási ideje az első, „lassú" számító gépen,
- $D_i(x) =$ az i Gödel számú program futási ideje a második, „gyors" számító gépen.

Feltételezhetjük, hogy $C_i(x)$ és $D_i(x)$ bonyolultsági mértékek. Mármost az előzőek szerint $t(x)$-re teljesül, hogy az i_0 Gödel számú program akkor és csak akkor fut $t(x)$ időben a gyors számítógépen, ha a lassún is.

Most egy másik meglepő eredményt ismertetünk.

Tétel 40.10 Blum felgyorsítási tétele. *Legyen $g(x,y)$ rekurzív függvény, $C_i(x)$ pedig tetszőleges bonyolultsági mérték. Ekkor van olyan $f(x)$ rekuzív függvény, hogy bármely olyan i számra, amelyre $\varphi_i(x) = f(x)$, létezik olyan j, hogy*

1. $\varphi_j(x) = f(x)$,
2. $g(x, C_j(x)) \leq C_i(x)$, *majdnem minden x-re.*

A TÉTEL INTERPRETÁLÁSA. Legyen $g(x, y) = 2^y$, és legyen $f(x)$ a tételben szereplő rekurzív függvény. Akárhogyis veszünk egy olyan P programot, amely f-et számítja ki, megadható egy olyan Q program is, amely szintén az f-et számítja ki, csak sokkal hatékonyabban: Ha i jelöli a P és j a Q Gödel számát, akkor $2^{C_j(x)} \leq C_i(x)$, majdnem minden x-re.

41 P és NP-problémák

Legyenek $f, g : \mathbf{N} \to \mathbf{N}$ függvények. Azt mondjuk, hogy $f(n) = O(g(n))$, ha van olyan c konstans és n_0 küszöbszám, hogy $f(n) \leq cg(n)$ valahányszor $n \geq n_0$.

Állítás 41.1 *Legyenek f, g $\mathbf{N} \to \mathbf{N}$ függvények.*

1. *Ha $\lim_{n \to \infty} f(n)/g(n) = \beta$, ahol β pozitív valós szám, akkor $f(n) = O(g(n))$.*
2. *Ha $\lim_{n \to \infty} f(n)/g(n) = \infty$, akkor $f(n)$ gyorsabban nő, mint $f(n)$, azaz $g(n) = O(f(n))$ de $f(n) \neq O(g(n))$.*

Állítás 41.2 *Legyen p olyan r-ed fokú polinom, hogy $p(n) \in \mathbf{N}$ valahányszor $n \in \mathbf{N}$. Ha $m < r$, akkor $p(n)$ gyorsabban nő, mint az n^m polinom. Ha $r = m$, akkor $p(n)$ ugyanolyan gyorsan nő, mint az n^m polinom, azaz $p(n) = O(n^m)$ és $n^m = O(p(n))$. Továbbá tetszőleges $k \geq 1$ esetén k^n gyorsabban nő, mint $p(n)$.*

Definíció 41.3 *Egy $L \subseteq A^*$ nyelvet polinomidőben eldönthetőnek nevezünk, ha létezik olyan M Turing-gép és $p(n)$ polinom, hogy $L = L(M)$, és valahányszor $x \in L$, M legfeljebb $p(|x|)$ lépésben megáll x-en.*

A továbbiakban a polinomidőben eldönthető nyelvek osztályát **P**-vel jelöljük.

Definíció 41.4 *Egy $L \subseteq A^*$ nyelv az* **NP** *osztályba esik, ha létezik olyan M nemdeterminisztikus Turing-gép és $p(n)$ polinom, hogy $L = L(M)$, és valahányszor $x \in L$, M valamely kiszámítási sorozatával legfeljebb $p(|x|)$ lépésben megáll x-en.*

P a polinomidőben megoldható problémák, **NP** pedig a poinomidőben verifikálható problémák osztályát reprezentálja. Polinomidőben megoldható problémák: összefüggőség, legrövidebb út, legnagyobb közös osztó, 2-SAT. Az **NP** osztályba eső problémák: SAT, Hamilton út, teljes részgráf probléma, stb. (A SAT és 2-SAT definícióját ld. lejjebb.) Azt, hogy egy probléma a **P** 0sztályba esik, általában úgy bizonyítjuk, hogy probléma megoldására adunk egy polinomiedjű algoritmust. Ezt persze nem Turing-gép szinten specifikáljuk. A futási idő analíziséhez az algoritmust lépésekre osztjuk úgy, hogy a lépésszám az input hosszának polinomiális függvénye legyen, és minden egyes lépés szimulálható legyen polinom időkorlátos determinisztikus Turing-géppel. Ez utóbbit a legritkább esetben bizonyítjuk formálisan.

Példaként tekintsük az alábbi problémát.

Adott: Egy G véges irányított gráf az $x_1, \ldots, x_m$ csúcsokkal, és két különböző x_{i_1} és x_{i_2} csúcs.

Kérdés: Létezik-e x_{i_1}-ből x_{i_2}-be vezető irányított út?

Módszer: Képezzük az $M_0, \ldots, M_m$ mátrixokat, ahol

$$\begin{aligned} (M_0)_{i,j} &= \begin{cases} 0 & \text{ha } i = j \\ 1 & \text{ha } (i,j) \text{ él és } i \neq j \\ \infty & \text{különben.} \end{cases} \\ (M_{k+1})_{i,j} &= \min\{(M_k)_{i,j}, (M_k)_{i,k+1} + (M_k)_{k+1,j}\} \end{aligned}$$

Világos, hogy $(M_k)_{i,j} < \infty$ akkor és csak akkor, ha létezik x_i-ből olyan út x_j-be, mely minden belső pontja az $\{x_1, \ldots, x_k\}$ halmazba esik, és ekkor $(M_k)_{i,j}$ a legrövidebb ilyen út hossza. Tehát akkor és csak akkor létezik x_{i_1}-ből x_{i_2}-be vezető út, ha $(M_m)_{i_1,i_2} < \infty$.

Az algoritmus lépésszáma $O(m)$, és minden lépés realizálható polinom időkorlátos determinisztikus Turing-géppel, ahol n az input hossza. Mivel m lineárisan függ n-től, a probléma a **P** osztályban van.

Az, hogy egy probléma a **P** osztályba esik-e, független attól, hogyan kódoljuk az inputot, amennyiben „megfelelően tömör" kódolást alkalmazunk: számok esetén $d > 1$ alapú kódolást, halmaz elemeit felsoroljuk mint pld. $\{x1, \ldots, xn\}$, ahol i az i szám kódját is jelöli, stb. Láttuk, hogy a kiszámítás (szekvenciális) modellje ekvivalensek. Egy pontosabb analízissel azt is megmutathattuk volna, hogy „polinomiálisan ekvivalensek". Így a **P** osztály akkor sem változna, ha definíciójában a kiszámítás más modelljét vennénk alapul. A gyakorlatban előforduló legtöbb **P**-beli probléma rendje az input méretében legfeljebb 4-5.

A **P** osztályt a gyakorlatban megoldható problémák osztályának tekintjük. Ez ellen persze több ellenvetés tehető. Rögzített k esetén az a probléma, hogy adott n csúcsú gráf tartalmaz-e k csúcsú teljes részgráfot, a triviális algoritmussal megolgható $O(n^k)$ időben. Nem ismert, hogy létezik-e ennél jobb algoritmus. A konstansok nagyságát sem veszi figyelembe a **P** osztály definíciója, így kis méretű inputon sokszor egy exponenciális algoritmus is jobban működhet, mint egy polinomiális. Bizonyos esetekben (pld. csomómentesség) csak annyit tudunk, hogy a probléma **P**-be esik, de nem ismert polinomiális algoritmus.

Azt, hogy egy probléma az **NP** osztályba esik, általában úgy bizonyítjuk, hogy megadunk olyan nemdeterminisztikus polinomiális algoritmust, amely megoldja a problémát.

Adott: Egy G gráf és egy k szám.

Kérdés: Tartalmaz-e G legalább egy k csúcsú teljes részgráfot.

Módszer: Nemdeterminisztikusan generáljuk G csúcsainak egy k-elemű részhalmazát, és ellenőrizzük, hogy ez teljes részgráfot feszít-e ki.

Ez az algoritmus megvalósítható polinom időkorlátos nemdeterminisztikus Turing-géppel.

Állítás 41.5 $\mathbf{P} \subseteq \mathbf{NP}$, *továbbá minden* $\mathbf{NP}$*-beli nyelv rekurzív.*

Definíció 41.6 *Egy* $f : A^* \longrightarrow B^*$ *függvényt polinomidőben kiszámítható függvénynek nevezünk, ha létezik olyan Turing-gép és* $p(n)$ *polinom, hogy* M *kiszámítja* f*-et és tetszőleges* $x \in A^*$ *szóra legfeljebb* $p(|x|)$ *lépésben megáll* x*-en.*

Amennyiben $p(n)$ r-edfokú polinom, azt is mondjuk, hogy M $O(n^r)$ időben számítja ki f-et.

Definíció 41.7 *Legyen* $L \subseteq A^*$ *és* $L' \subseteq B^*$*. Az mondjuk, hogy* L *polinomidőben visszavezethető (vagy reducibilis)* L'*-re, ha létezik olyan* $f : A^* \longrightarrow B^*$ *polinomidőben kiszámítható függvény, hogy bármely* $x \in A^*$ *szóra*

$$x \in L \quad \Leftrightarrow \quad f(x) \in L'.$$

Lemma 41.8 *A polinomidőben való visszavezetés előrendezés, tehát reflexív és tranzitív reláció. Továbbá* $\mathbf{P}$ *és* $\mathbf{NP}$ *zártak a polinomidőben való visszavezetésre, azaz érvényesek az alábbiak:*

1. *Ha* $L \leq_p L'$ *és* $L' \in \mathbf{P}$*, akkor* $L \in \mathbf{P}$*.*
2. *Ha* $L \leq_p L'$ *és* $L' \in \mathbf{NP}$*, akkor* $L \in \mathbf{NP}$*.*

Bizonyítás. Csak az első állítást bizonyítjuk. Az világos, hogy $\leq_p$ reflexív. Tegyük fel, hogy $L \leq_p L'$ és $L' \leq_p L''$, ahol $L \subseteq A^*$, $L' \subseteq B^*$ és $L'' \subseteq C^*$. Ekkor léteznek olyan $f : A^* \longrightarrow B^*$ és $g : B^* \longrightarrow C^*$ polinomidőben kiszámítható függvények, hogy bármely $x \in A^*$ és $y \in B^*$ esetén

$$\begin{aligned} x \in L \quad &\Leftrightarrow \quad f(x) \in L' \\ y \in L' \quad &\Leftrightarrow \quad g(y) \in L''. \end{aligned}$$

Legyen $h : A^* \longrightarrow C^*$ a $h(x) = g(f(x))$ összefüggéssel definiált összetett függvény. Ha f $O(n^r)$ és g $O(n^s)$ időben kiszámítható, ahol $r, s \geq 1$, akkor h kiszámítható $O(n^{r+s})$ időben. Továbbá,

$$x \in L \quad \Leftrightarrow \quad h(x) \in L''. \quad \square$$

42 NP-teljes nyelvek

Definíció 42.1 *Egy* $L \subseteq A^*$ *nyelvet* $\mathbf{NP}$*-teljesnek nevezünk, ha* $L \in \mathbf{NP}$ *és valahányszor* $L' \in \mathbf{NP}$*, mindig fennáll, hogy* $L' \leq_p L$*.*

Állítás 42.2 *Tegyük fel, hogy L egy* **NP***-teljes nyelv.* $\mathbf{P} = \mathbf{NP}$ *akkor és csak akkor teljesül, ha* $L \in \mathbf{P}$.

Bizonyítás. Ha $\mathbf{P} = \mathbf{NP}$, akkor $L \in \mathbf{NP} = \mathbf{P}$. Fordítva, tegyük fel, hogy $L \in \mathbf{P}$. Ha $L' \in \mathbf{NP}$, akkor $L' \leq_p L$. Így mivel **P** zárt a polinomidőben való visszavezetésre, $L' \in \mathbf{P}$. Tehát $\mathbf{NP} \subseteq \mathbf{P}$, és mivel a fordított irányú tartalmazás nyilvánvaló, $\mathbf{P} = \mathbf{NP}$. □

Állítás 42.3 *Tegyük fel, hogy L* **NP***-teljes, $L' \in \mathbf{NP}$, és $L \leq_p L'$. Akkor L' is* **NP***-teljes.*

Bizonyítás. Legyen $L'' \in \mathbf{NP}$. Mivel L **NP**-teljes, $L'' \leq_p L$. Így a tranzitivitás miatt $L'' \leq_p L'$. □

Definíció 42.4 *A kielégíthetőségi probléma (*SAT*) a következő.*

Adott: Egy konjunktív normálformában felírt logikai kifejezés.

Kérdés: Kielégíthető-e a kifejezés.

A SAT problémát fefoghatjuk úgy is, mint egy nyelvet. Ehhez egy konjunktív normálformában adott kifejezéseket kódolnunk kell. A kódolás lényegében tetszőleges, annak figyelembevételével, hogy megfelelően tömör legyen. A kódolásra hasznáhatunk kételemű halmazt, mondjuk az $\{a, b\}$ halmazt is. A zárójelek kódja legyen ba és ba^2, a $\wedge$ és $\vee$ műeveleti jeleké ba^3 és ba^4. Amennyiben egy kifejezésben az $x_1, \ldots, x_n$ változók fordunak elő, minden x_i változót helytettesítsünk a $bb\bar{i}$ szóval, ahol $\bar{i}$ az i szám bináris alakja.

Lemma 42.5 SAT $\in$ **NP**.

Proof. Az M nemdeterminisztikus Turing-gép működjön az alábbi módon. Adott szóra először döntse el, hogy a szó konjunktív normálformájú kifejezés kódja-e. Ha nem, akkor M végtelen ciklusba esik. Ha igen, akkor nemdeterminisztikusan generálja a kifejezésben szereplő változók egy kiértékelését, majd determinisztikusan ellenőrzi, hogy a generált kiértékelés kielégíti-e a kifejezést. Ha kielégíti, akkor M megáll, ellenben végtelen ciklusba esik. □

Tétel 42.6 Cook tétele. SAT **NP***-teljes.*

Bizonyítás. Azt már beláttuk, hogy SAT az **NP** osztályban van. Most be kell bizonyítanunk, hogy $L \leq_p$ SAT teljesül teszőleges $L \in \mathbf{NP}$ nyelv esetén.

Legyen $L = L(M)$, ahol M nemdeterminisztikus Turing-gép és létezik olyan $p(n)$ polinom, hogy $u \in L$ esetén M valamely kiszámítási sorozata az u szón legfeljebb $p(|u|)$ hosszú. Megadunk egy olyan polimom időben kiszámítható $u \mapsto \delta_u$ függvényt, ahol δ_u konjunktív normálformájú kifejezés, hogy $u \in L$ akkor és csak akkor, ha δ_u kielégíthető.

Legyen $t = p(|u|)$. Mivel $u \in L$ akkor és csakis akkor, ha M legfeljebb t lépésben elfogadja u-t, ezért elegendő M működését t lépésig figyelni. E közben M legfeljebb $2t+1$ mezőt használ a szalagon. Ezért M működését nyomon követhetjük egy $(t+1) \times (2t+1)$-es T táblázaton, amely a szalag tartalmát adja meg kezdetben és az egyes lépések után.

Legyenek M állapotai $q_1, q_2, \ldots, q_m$, ahol q_1 a kezdőállapot. Feltehető, hogy q_m az egyetlen olyan állapot, amelyben megállhat a gép, és hogy a q_m állapotban a gép mindig meg is áll. Legyen $s_0, s_1, \ldots, s_r$ az M abc-je. A kiszámítási sorozat definícióját módosítsuk úgy, hogy megengedünk „üres" átmeneteket is. Így M akkor és csak akkor fogadja el az u szót, ha valamely kiszámítási sorozatával pontosan t lépésben elfogadja.

A δ_u kifejezést az alábbi változók felhasználásával építjük fel.

$$\{\sigma_{i,j,k} | 0 \le i \le r,\ 1 \le j \le 2t+1,\ 0 \le k \le t\}$$
$$\{\rho_{h,j,k} | 1 \le h \le m,\ 1 \le j \le 2t+1,\ 0 \le k \le t\}$$

Amennyiben $u \in L$, δ_u igaz lesz a változók alábbi v kiértékelése mellett. Válasszuk az M egy olyan futását az u szón, amelyben t lépésben elfogadja u-t. Legyen

- $v(\sigma_{i,j,k}) = 1$ akkor és csak akkor, ha a k-dik lépésben a szalag j-dik pozícióján s_i áll.
- $v(\rho_{h,j,k}) = 1$ akkor és csak akkor, ha a k-dik lépésben M a q_h állapotban a szalag j-dik mezőjét olvassa.

Tehát a változók száma $O(t^2)$, azaz $O(q(n))$, ahol $q(n) = (p(n))^2$ és n az u hosszát jelöli.

Jelölés: Teszőleges $x_1, \ldots, x_\ell$ változókra,

$$\Delta_{e=1}^{\ell} x_e \quad = \bigwedge_{1 \le e < f \le \ell} (\overline{x}_e \vee \overline{x}_f) \wedge \bigwedge_{e=1}^{\ell} x_e.$$

Tehát $\Delta_{e=1}^{\ell} x_e$ akkor és csak akkor igaz a változók egy kiértékelése mellett, ha pontosan egy olyan e létezik, amelyre x_e igaz.

A δ_u kifejezést 5 részformula konjunkciójaként adjuk meg. Minden részformula hossza t, és így az u hosszának polinom függvénye lesz.

1. RÉSZFORMULA. Ez azt fejezi ki, hogy a kiszámítás minden egyes lépésében M pontosan egy állapotban van és pontosan egy mezőt olvas.

$$\bigwedge_{k=0}^{t} \Delta_{1 \le h \le m,\ 1 \le t \le 2t+1} \rho_{h,j,k}$$

2. RÉSZFORMULA. Ez azt fejezi ki, hogy minden egyes lépésben, a szalag minden egyes mezője pontosan egy szimbólumot tartalmaz.

$$\bigwedge_{0 \le k \le t,\ 1 \le j \le 2t+1} \Delta_{i=0}^{r} \sigma_{i,j,k}$$

3. RÉSZFORMULA Kezdetben a szalagon az u szó van felírva, és M a q_1 állapotban az u szó előtti s_0 szimbólumot olvassa.

$$\bigwedge_{j=1}^{t+1} \sigma_{0,j,0} \wedge \bigwedge_{j=1}^{n} \sigma_{i_j,t+j+1,0} \wedge \bigwedge_{j=1}^{t-n} \sigma_{0,t+n+j+1,0} \wedge \rho_{0,t+1,0}$$

ahol $u = s_{i_1} \dots s_{i_n}$.

4. RÉSZFORMULA. A t-dik lépésben M a q_m állapotban van (és így elfogadja az u szót).

$$\bigvee_{j=1}^{2t+1} \rho_{m,j,t}$$

5. RÉSZFORMULA. A lépések M szabályainak megfelelően követik egymást. Legyenek a Turing-gép szabályai

$$\begin{aligned} q_{i_a}\ s_{j_a}\ s_{k_a}\ q_{\ell_a}, &\quad a = 1, \dots, \overline{a} \\ q_{i_b}\ s_{j_b}\ R\ q_{\ell_b}, &\quad b = 1, \dots, \overline{b} \\ q_{i_c}\ s_{j_c}\ L\ q_{\ell_c}, &\quad c = 1, \dots, \overline{c} \end{aligned}$$

Ekkor a kifejezés

$$\bigwedge_{1 \le j \le 2t+1,\ 0 \le k \le t-1} F(j,k)$$

ahol $F(j,k)$ az alábbi formulával ekvivalens konjunktív normálformájú kifejezés.

$$\begin{aligned} F(j,k) &= A(j,k) \vee B(j,k) \vee C(j,k) \vee I(j,k) \vee N(j,k) \\ A(j,k) &= \bigvee_{a=1}^{\overline{a}} (\rho_{i_a,j,k} \wedge \sigma_{j_a,j,k} \wedge \sigma_{k_a,j,k+1} \wedge \rho_{\ell_a,j,k+1}) \\ B(j,k) &= \bigvee_{b=1}^{\overline{b}} (\rho_{i_b,j,k} \wedge \sigma_{j_b,j,k} \wedge \sigma_{j_b,j,k+1} \wedge \rho_{\ell_b,j+1,k+1}) \\ C(j,k) &= \bigvee_{c=1}^{\overline{c}} (\rho_{i_c,j,k} \wedge \sigma_{j_c,j,k} \wedge \sigma_{j_c,j,k+1} \wedge \rho_{\ell_c,j-1,k+1}) \\ I(j,k) &= \bigvee_{1 \le h \le m, 0 \le i \le r} (\rho_{h,j,k} \wedge \sigma_{i,j,k} \wedge \rho_{h,j,k+1} \wedge \sigma_{i,j,k+1}) \\ N(j,k) &= \bigvee_{i=0}^{r} (\sigma_{i,j,k} \wedge \sigma_{i,j,k+1}) \wedge \bigwedge_{h=0}^{m} \overline{\rho}_{h,j,k}. \end{aligned}$$

Amennyiben $j = 2t+1$, $B(j,k)$ legyen egy kielégíthetetlen formula. Hasonlóan, a $j = 1$ esetben $C(j,k)$ legyen kielégíthetetlen.

A konstrukció alapján világos az, hogy amennyiben $u \in L$, δ_u kielégíthető, a fentiekben megadott v kiértékelés mellett δ_u igaz. Fordítva, ha valamely kiértékelés mellett δ_u igaz, akkor mind az 5 részformulája igaz. Így bármely k-ra a $\rho_{i,j,k}$ változók közül pontosan egy igaz, amiből meghatározható, hogy a k-dik lépésben mely állapotban van a gép és melyik mezőt olvassa. Mivel a második részformula is igaz, ezért egyértelműen kitölthető a szalag mezőinek tartalma. A harmadik részformula miatt kezdetben u van a szalagon, és a gép az u szót megelőző s_0 szimbólumot olvassa a q_1 állapotban. A negyedik részformula miatt a gép végül a q_m állapotban van. Végül a lépések a gép szabályainak megfelelően követik egymást. Tehát $u \in L$. Mivel az $u \mapsto \delta_u$ függvény nyilván kiszámíthtató polinom időben, ezért $L \leq_p$ SAT. □

43 Néhány további NP-teljes probléma

Teszőleges $k \geq 2$ esetén k-SAT ugyanaz a probléma, mint SAT, azzal a különbséggel, hogy a konjunktív normálformájú kifejezés minden tagja k literál diszjunkciója. Belátható, hogy 2-SAT **P**-ben van.

Állítás 43.1 *A* 3-SAT *probléma* **NP***-teljes.*

Bizonyítás. Mivel 3-SAT a SAT speciális esete, nyilvánvaló, hogy 3-SAT is **NP**-ben van. A teljességhez azt mutatjuk meg, hogy SAT $\leq_p$ 3-SAT.

Tekintsünk agy konjunktív normálformájú kifejezést. Ennek minden tagját helyettesítsük olyan konjunktív normálformájú kifejezéssel, amelyben minden tag három literál diszjunkciója, és amely akkor és csak akkor kielégíthető, ha az adott tag az. Az így előálló kifejezés akkor és csak akkor lesz kielégíthető, ha a kiindulási kifejezés az.

Minden egy literálból álló ℓ tagot helyettesítsünk az $\ell \vee \ell \vee \ell$ taggal, minden $\ell_1 \vee \ell_2$ tagot az $\ell_1 \vee \ell_1 \vee \ell_2$ taggal, minden $\ell_1 \vee \ell_2 \vee \ell_3$ tagot önmagával, és $m > 3$ esetén minden $\ell_1 \vee \ldots \vee \ell_m$ tagot az

$$(\ell_1 \vee \ell_2 \vee x) \wedge (\overline{x} \vee \ell_3 \vee y) \wedge (\overline{y} \vee \ell_4 \vee z) \wedge \ldots \wedge (\overline{u} \vee \ell_{m-1} \vee \ell_m)$$

formulával, ahol $x, y, z, \ldots, u$ új változók. □

A lefogási probléma azt kérdezi, hogy adott G (véges) gráf és $m \geq 0$ számra létezik-e G éleinek legfeljebb m pontból álló lefogása.

Állítás 43.2 *A lefogási probléma* **NP***-teljes.*

Bizonyítás. Az nyilvánvaló, hogy a lefogási probléma **NP**-ben van. A teljességhez a 3-SAT problémát vezetjük vissza a lefogási problémára.

Legyen adott egy φ konjunktív normálformájú kifejezés, amelyben minden tag három literál diszjunkciója. Legyenek $x_1, \ldots, x_n$ a kifejezésben szereplő változók és $c_1, \ldots, c_s$ a tagok, ahol $c_j =$

$\ell_{j1} \vee \ell_{j2} \vee \ell_{j3}$. A G gráf tartalmazzon minden x_i változóhoz 2 csúcsot, amelyeket az x_i-vel és a $\overline{x}_i$-vel azonosítunk. Minden i-re kössük össze az x_i csúcsot a $\overline{x}_i$ csúccsal. Minden c_j taghoz tekintsünk még 3 új csúcsot, amelyeket az ℓ_{jk} literálokkal azonosítunk és amelyek egy háromszöget alkotnak. Ezenkívül kössünk még össze minden ℓ_{jk} literál-csúcsot az x változó-csúccsal vagy az $\overline{x}$ csúccsal annak megfelelően, hogy $\ell_{jk} = x$ vagy $\ell_{jk} = \overline{x}$ teljesül. Az így kapott gráfnak $2n + 3s$ csúcsa és $n + 6s$ éle van. Legyen $m = n + 2s$. Megmutatjuk, hogy G élei akkor és csak akkor foghatóak le legfeljebb m éllel, ha φ kielégíthető. Tegyük fel először azt fel, hogy φ kielégíthető, mondjuk v kielégítő kiértékelés. Minden egyes x_i változóra vegyük az x_i csúcsot, amennyiben $v(x_i) = 1$, és a $\overline{x}_i$ csúcsot, amennyiben $v(x_i) = 0$. Továbbá minden egyes c_j tagra rögzítsük a tag egy olyan literálját, amely értéke 1, és tekintsük a másik két literál-csúcsot. Az így kapott m csúcs lefogja az éleket.

Fordítva, ha a H legfeljebb m elemű halmazban lévő csúcsok lefogják az éleket, akkor minden x_i-re az x_i-hez vagy az $\overline{x}_i$-hez tartozó csúcs H-ban van. Továbbá minden c_j esetén a taghoz tartozó háromszög legalább két csúcsa is H-ban van. Mivel ez már m csúcs, H ezekből a csúcsokból áll. Legyen $v(x_i) = 1$ akkor és csak akkor, ha az x_i csúcs H-ban van. Könnyen látható, hogy v kileégíti a φ kifejezést. □

A független ponthalmaz probléma azt kérdezi, hogy adott G gráf és $k \geq 0$ szám esetén létezik-e G-nek k pontból álló független részhalmaza.

Állítás 43.3 *A független ponthalmaz probléma* **NP***-teljes.*

Bizonyítás. A probléma nyilván **NP**-ben van. A teljességhez vegyük észre, hogy egy n-ed rendű G gráf élei akkor és csak akkor foghatóak le k ponttal, ha létezik $n - k$ független pont. □

A teljes részgráf probléma azt kérdezi, hogy adott G gráfnak létezik-e k pontú teljes részgráfja.

Következmény 43.4 *A teljes részgráf probléma* **NP***-teljes.*

További **NP**-teljes problámák: gráfok 3-színezhetősége, Hamiltion út létezése, stb.

Folytatás következik