Számítási modellek

11. előadás

Fejlődő rendszerek

Aristid Lindenmayer (1925-1989), magyar származású holland biológus, matematikus a vörös algák (callithamnion roseum) fejlõdésének tanulmányozása során írta le a később róla elnevezett párhuzamos fejlődő rendszert.

Az algák sejtjeinek 9 állapota van. Csak ez határozza meg a következő állapotát. Bizonyos állapotokban lévő sejtek növelhetik az algát, sőt oldalágat is növeszthetnek.

Matematikailag így írható le, kezdetben egyetlen 1-es állapotú sejt van.

1→23	4→25	7→8	[→ [
2→2	5→65	8→9 [3]	] →]
3→24	6→7	9→9	

([és] oldalágat határol.)

Vörös algák

1→23,2→2,3→24,4→25, 5→65,6→7,7→8,8→9[3],9→9, [→[,]→]

Az 1 axiómából induló levezetés elemei:

1, 23, 224, 2225, 22265, 222765, 2228765, 2229[3]8765

2229[24]9[3]8765, 2229[225]9[24]9[3]8765

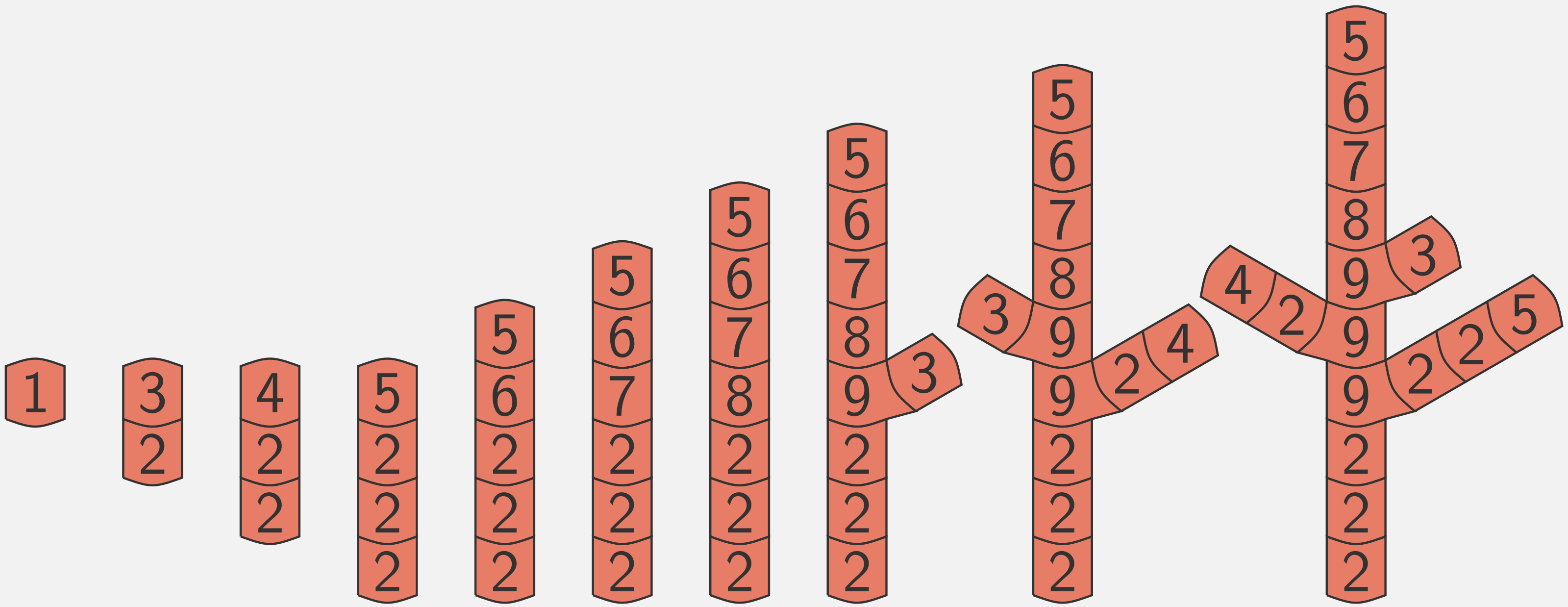
2229[2265]9[225]9[24]9[3]8765,

2229[22765]9[2265]9[225]9[24]9[3]8765,

2229[228765]9[22765]9[2265]9[225]9[24]9[3]8765

2229[229[3]8765]9[228765]9[22765]9[2265]9[225]9[24]9[3]8765

2229[229[24]9[3]8765]9[229[3]8765]9[228765]9[22765]9[2265]9[225]9[24]9[3]8765



0L-rendszer

Definíció

0L-rendszer (Lindenmayer-rendszer, 0L-grammatika, L-rendszer) alatt egy $G = (V, P, \omega)$ rendezett hármast értünk, ahol

- V egy véges ábécé,
- P környezetfüggetlen V feletti átírási szabályok véges halmaza, feltesszük, hogy minden $a \in V$-re létezik szabály P-ben,
- $\omega \in V^+$ pedig az axióma.

Definíció

$z_1, z_2 \in V^*$ szavak esetében $z_1 \Longrightarrow_G z_2$ írható ha $z_1 = a_1 a_2 \cdots a_r$ $(a_i \in V, 1 \leqslant i \leqslant r)$, $z_2 = x_1 x_2 \cdots x_r$ $(x_i \in V^*, 1 \leqslant i \leqslant r)$ és minden $1 \leqslant i \leqslant r$ -re $a_i \to x_i \in P$.

(G elhagyható, ha világos, hogy melyik G-ről van szó.)

A $\Longrightarrow^*$ reláció a $\Longrightarrow$ reláció reflexív, tranzitív lezártja.

0L-rendszer által generált nyelv

Definíció

A $G = (V, P, \omega)$ **0L-rendszer által generált nyelv** $L(G) = \{u \in V^* \mid \omega \Longrightarrow^* u\}$.

Példa: Legyen $G = (V, P, \omega)$ egy 0L-rendszer, ahol $V = \{a\}, P = \{a \to a^2\}$, és $\omega = a^3$.

Ekkor $L(G) = \{a^{3 \cdot 2^i} \mid i \geqslant 0\}$.

Definíció

Amennyiben a $G = (V, P, \omega)$ 0L-rendszerben minden $a \in V$-re pontosan egy P-beli szabálya van G-nek, akkor **D0L-rendszerről** beszélünk (determinisztikus 0L-rendszer).

Ilyenkor P valójában egy $h : V \to V^*$ homomorfizmus, a $\langle h^t(\omega),\ t \geqslant 0\rangle$ sorozat $(h^0(\omega) := \omega, h^t(\omega) = h(h^{t-1}(\omega)), t \geqslant 0)$ a G D0L rendszer növekedési sorozata, $f(t) := |h^t(\omega)|, (f : \mathbb{N} \to \mathbb{N})$, a növekedési függvénye.

0L-rendszer – Példák

Példa: $G_{\mathrm{FIB}} = (\{a, b\}, \{a \to b, b \to ab\}, a)$.

G_{FIB} D0L-rendszer, azaz valójában egy h homomorfizmus. Jelölje ω_n az n. átírás után kapott szót. $\omega_0 = a, \omega_1 = b, \omega_2 = ab = \omega_0\omega_1$. Teljes indukcióval belátjuk, hogy $n \geqslant 2$-re $\omega_n = \omega_{n-2}\omega_{n-1}$:

$\omega_{n+1} = h(\omega_n) = h(\omega_{n-2}\omega_{n-1}) = h(\omega_{n-2})h(\omega_{n-1}) = \omega_{n-1}\omega_n$. Tehát az $f(n)$ növekedési sorozatra $f(0) = 1, f(1) = 1$,

$f(n) = f(n-1) + f(n-2)$ rekurzió teljesül, ami megegyezik a Fibonacci sorozatra vonatkozó rekurzióval, eggyel eltolva. Tehát $f(n) = F_{n+1}$, ahol F_n jelöli az n. Fibonacci számot.

Teljes indukcióval hasonlóan látható, hogy $|\omega_n|_b = F_n$ $(n \geqslant 0)$ és $|\omega_n|_a = F_{n-1}(n \geqslant 1)$.

Példa: Nem létezik olyan G 0L rendszer, melyre $L(G) = \{a, a^2\}$. Ugyanis, ha $\omega = a$, akkor $a \Rightarrow^* a^2$, és így $a^2 \Rightarrow^* a^4$, tehát $a^4 \in L(G)$. Míg ha $\omega = a^2$, akkor $a^2 \Rightarrow^* a$, de ekkor $a \Rightarrow^* \varepsilon$, mert az átírási szabályok környezetfüggetlenek (az egyik a-ból a-t a másikból ε-t vezettük le), tehát $\varepsilon \in L(G)$.

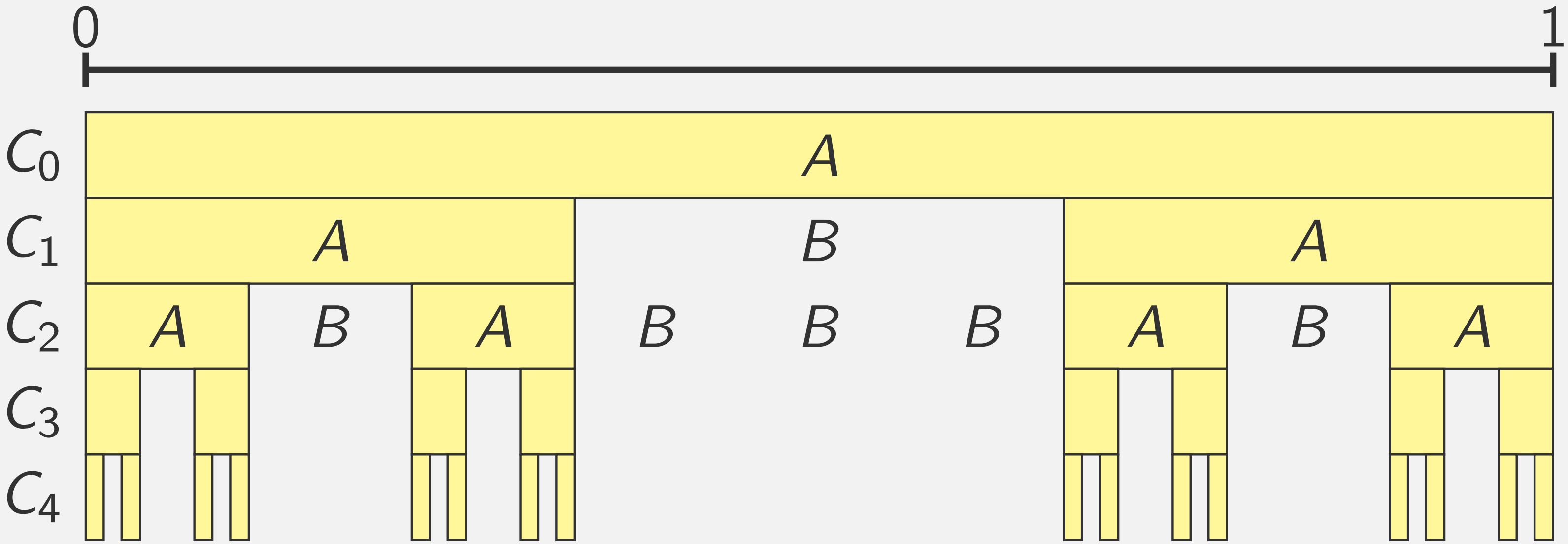
0L rendszer –Példák

Példa: $G_{\mathrm{CANTOR}} = (\{A, B\}, \{A \to ABA, B \to BBB\}, A)$.

Legyen $\langle \omega_n, n \geqslant 0 \rangle$ a G_{CANTOR} D0L-rendszer által generált szó-sorozat és $\omega_{n,i}$ $(1 \leqslant i \leqslant 3^n)$ az n. szó i. betűje. Ekkor

$$C_n := \bigcup_{i:\omega_{n,i}=A} \left[\frac{(i-1)}{3^n}, \frac{i}{3^n} \right]$$

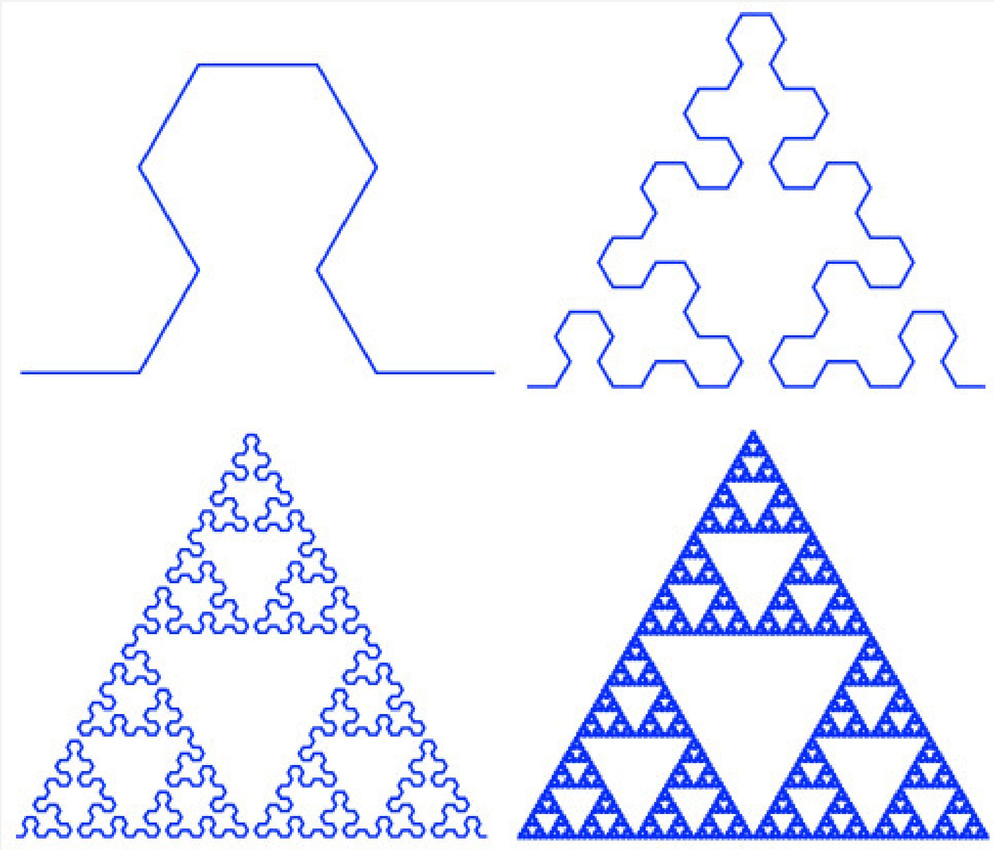
a jól ismert Cantor-halmaz közelítésének n-edik iterációja.



0L rendszer –Példák

$$G_{\text{SIERPIŃSKI}} = (\{A, B, +, -\}, \{A \to B - A - B, B \to A + B + A, + \to +, - \to -\}, A).$$

A és B: teknőcgrafika rajzoljon ki egy egységnyi vonalat abban az irányban, amerre a teknős áll. + és - jelentése: forduljon balra illetve jobbra 60 fokkal.Az első páros iterációk ezeket adják:



Kiterjesztett 0L rendszer

Definíció

A $G = (V, P, \omega)$ 0L rendszert szaporodó (propagating) Lindenmayer-rendszernek (**P0L-rendszernek**) nevezzük, ha nincs ε-szabálya.

Definíció

Egy $G = (V, T, P, \omega)$ rendszert ($T \subseteq V$) kiterjesztett (extended) Lindenmayer-rendszernek (**E0L-rendszernek**) nevezünk, ha G V feletti 0L-rendszer. Egy G kiterjesztett Lindenmayer-rendszer által **generált nyelv** $L(G) = \{u \in T^* \mid \omega \Longrightarrow^* u\}$).

G kiterjesztett determinisztikus Lindenmayer-rendszer (**ED0L-rendszer**), ha V feletti D0L-rendszer.

T neve: terminális ábécé.

E0L rendszer – Példa

Példa: $G_{\mathrm{SYNC}} = (\{A, B, C, A', B', C', F, a, b, c\}, \{a, b, c\}, P, ABC)$
P szabályai:

$$\begin{array}{llllll}
A \to AA' & A \to a & A' \to A' & A' \to a & a \to F & \\
B \to BB' & B \to b & B' \to B' & B' \to b & b \to F & \\
C \to CC' & C \to c & C' \to C' & C' \to c & c \to F & F \to F
\end{array}$$

Az F nemterminálisnak szinkronizáló szerepe van. Ugyanis ha néhány átírási lépés után maradt még nemterminális a szóban, akkor kell még legalább egy átíró lépés. Ekkor viszont a már meglévő terminálisok megváltoztathatatlanul átíródnak F-re, így végül nem kapunk terminális szót.

Ha n átírás után a kapott szó nem tartalmaz terminálist akkor az csak $A(A')^n B(B')^n C(C')^n$ lehet. Tehát
$L(G_{\mathrm{SYNC}}) = \{a^n b^n c^n \mid n \geqslant 1\}$.

Észrevétel: E0L-rendszerrel minden környezetfüggetlen nyelv generálható. Adjuk hozzá ugyanis az $a \to a$ E0L-szabályokat minden $a \in N \cup T$-re a nyelvet generáló grammatika szabályaihoz.

Táblázatos 0L-rendszer

Definíció

Egy $G = (V, P_1, \ldots, P_m, \omega)$ rendszert táblázatos Lindenmayer rendszernek (**T0L-rendszernek**) nevezünk, ha minden $1 \leqslant i \leqslant m$ esetén $G_i := (V, P_i, \omega)$ 0L rendszer.

Definíció

A $G = (V, P_1, \ldots, P_m, \omega)$ T0L rendszerben $z_1 \Longrightarrow_G z_2$ írható, ha létezik $1 \leqslant i \leqslant m$ melyre $z_1 \Longrightarrow_{G_i} z_2$.

(G elhagyható, ha világos, hogy melyik G-ről van szó.)

A $\Longrightarrow^*$ reláció a $\Longrightarrow$ reláció reflexív, tranzitív lezártja.

Definíció

A $G = (V, P_1, \ldots, P_m, \omega)$ **T0L-rendszer által generált nyelv** $L(G) = \{u \in V^* \,|\, \omega \Longrightarrow^* u\}$.

Kiterjesztett táblázatos 0L-rendszer

Definíció

Egy $G = (V, T, P_1, \ldots, P_m, \omega)$ rendszert kiterjesztett táblázatos Lindenmayer rendszernek (**ET0L-rendszernek**) nevezünk, ha minden $1 \leqslant i \leqslant m$ esetén $G_i := (V, T, P_i, \omega)$ E0L rendszer.

Definíció

A $G = (V, P_1, \ldots, P_m, \omega)$ T0L rendszerben $z_1 \Longrightarrow_G z_2$ írható, ha létezik $1 \leqslant i \leqslant m$ melyre $z_1 \Longrightarrow_{G_i} z_2$.

A $\Longrightarrow^*$ reláció a $\Longrightarrow$ reláció reflexív, tranzitív lezártja.

A $G = (V, P, \omega)$ **ET0L-rendszer által generált nyelv** $L(G) = \{u \in T^* \mid \omega \Longrightarrow^* u\}$.

Analóg módon definiáljuk a TD0L, ETD0L, EPT0L, stb. rendszereket.

Kiterjesztett táblázatos 0L-rendszer

Példa:

$G_{\mathrm{SYNC2}} = (\{A, B, C, A', B', C', a, b, c\}, \{a, b, c\}, P_1, P_2, ABC)$

P_1 szabályai:		P_2 szabályai:	
$A \to AA'$	$A' \to A'$	$A \to a$	$A' \to a$
$B \to BB'$	$B' \to B'$	$B \to b$	$B' \to b$
$C \to CC'$	$C' \to C'$	$C \to c$	$C' \to c$

$L(G_{\mathrm{SYNC2}}) = \{a^n b^n c^n \mid n \geqslant 1\}$.

Példa:

Legyen $G_{\mathrm{DADOG}} = (\{S, A, B, a, b, c\}, \{a, b, c\}, P_1, P_2, SSS)$, ahol

P_1 szabályai:		P_2 szabályai:	
$S \to A$	$a \to a$	$S \to B$	$a \to a$
$A \to Sa$	$b \to b$	$B \to Sb$	$b \to b$
$B \to c$	$c \to c$	$A \to c$	$c \to c$

$L(G_{\mathrm{DADOG}}) = \{cwcwcw \mid w \in \{a, b\}^*\}$.

ET0L-rendszerek

Definíció

Az E0L,D0L,ET0L, stb. rendszerek által generálható nyelvek nyelvosztályát rendre $\mathcal{L}(\text{E0L}), \mathcal{L}(\text{D0L}), \mathcal{L}(\text{ET0L})$, stb. jelöli.

A következőkben bizonyítás nélül ismertetjük az $\mathcal{L}(\text{ET0L})$ nyelvosztály néhány érdekes tulajdonságát.

D0L, DT0L rendszerek nem tudnak minden véges nyelvet generálni.

Tétel

Minden ET0L rendszert lehet 2 táblás ET0L rendszerrel szimulálni.

Tétel

Az $\mathcal{L}(\text{ET0L})$ és $\mathcal{L}(\text{EDT0L})$ zárt az unióra, konkatenációra, Kleene-lezártra, homomorfizmusra, reguláris nyelvvel való metszetre.

ET0L nyelvek egy tulajdonsága

Jelölés: Legyen $u = t_1 \dots t_n$, ahol $t_i \in X$ $(1 \leqslant i \leqslant n)$, továbbá $Y \subseteq X$, ekkor

$$|u|_Y := \Big| \{ \, i \mid (1 \leqslant i \leqslant n) \wedge (t_i \in Y) \, \} \Big|.$$

Az alábbi lemma (nem bizonyítjuk) egy szükséges (de nem elégséges !!!) feltételt ad egy nyelv ET0L rendszerrel való generálhatóságára.

Lemma

Legyen $L \subseteq T^*$ egy $\mathcal{L}(\text{ET0L})$-beli nyelv. Ekkor bármely $\varnothing \neq S \subseteq T$-re $\exists k > 0$ egész, hogy $\forall u \in L$-re

(i) vagy $|u|_S \leqslant 1$

(ii) vagy u-nak van olyan w részszava amelyre $|w| \leqslant k$ $|w|_S \geqslant 2$

(iii) vagy van L-nek végtelen sok olyan w szava, amelyre $|w|_S = |u|_S$.

0L-rendszerek számítási ereje

Példa: $\{(ab^n)^m \mid m \geqslant n \geqslant 1\} \notin \mathcal{L}(\mathrm{ET0L})$.

Legyen $S = \{a\}$, $k > 0$ és $u = (ab^{k+1})^{k+1}$.

- Mivel $k + 1 \geqslant 2$ ezért (i) nem teljesül.
- Az a-k u-ban k-nál távolabb vannak egymástól, így (ii) sem.
- A $k + 1$ darab a-t tartalmazó L-beli szavak: $\{(ab)^{k+1}, (ab^2)^{k+1}, \ldots, (ab^{k+1})^{k+1}\}$. Ezek száma véges, így (iii) sem.

Mivel $u \in L$, a Lemmából következik az állítás.

Megjegyzés: $\{(ab^n)^m \mid m \geqslant n \geqslant 1\} \in \mathrm{CS} \setminus \mathrm{CF}$

Tétel

$\mathrm{CF} \subset \mathcal{L}(\mathrm{E0L}) = \mathcal{L}(\mathrm{EP0L}) \subset \mathcal{L}(\mathrm{ET0L}) = \mathcal{L}(\mathrm{EPT0L}) \subset \mathrm{CS}$

Megjegyzés: Vannak 1L-rendszerek, 2L-rendszerek, stb.
1L-rendszer: egy bal- vagy jobboldalon jelenlévő szimbólumtól függ az átírás alkalmazhatósága (azaz az átírási szabályok környezetfüggőek).

Watson-Crick 0L-rendszerek

Alapötlet: Vizsgáljuk meg, hogy milyen nukleotidsorozatokat kaphatunk, ha a DNS evolúciójára fejlődő rendszerként tekintünk. DNS specifikus feltevés, hogy mindig rendelkezésre áll a generált sztring Watson-Crick komplemense. Így ha „rossz” sztringet kapunk áttérhetünk, a komplemens sztringe azzal folytatva a számítást.

Az áttérés lehet szabad vagy valamilyen kontroll hatására történő.

A Lindenmayer rendszerek minden betűre párhuzamosan átírják az aktuális sztringet a szabályaik szerint. A Watson-Crick komplemensre való áttérés is valójában egy Lindenmayer-típusú átírás, így természetes módon illeszkedik 0L rendszerekhez.

DNS-típusú ábécé, Watson-Crick morfizmus

Definíció

DNS-típusú ábécén egy $2n$ ($n \geqslant 1$) darab betűből álló, $\Sigma = \{a_1, \ldots, a_n, \bar{a}_1, \ldots, \bar{a}_n\}$ ábécét értünk. Az a_i és $\bar{a}_i$ $(1 \leqslant i \leqslant n)$ betűket egymás komplemenseinek hívjuk, továbbá a nem felülhúzott betűket **purinnak**, a felülhúzottakat pedig **pirimidinnek** nevezzük.

Ez a terminológia az eredeti DNS ábécéből, $\{A, G, T, C\}$-ből ered, ahol az A és G betűk ténylegesen purinokat, a T és C betűk pedig pirimidineket jelölnek. T az A, C pedig a G komplemense.

Definíció

h_W-vel jelöljük azt a DNS-típusú Σ ábécé feletti betűnkénti endomorfizmust, amely minden betűhöz a komplemensét rendeli. h_W-t **Watson-Crick morfizmusnak** is nevezzük.

Példa

$\Sigma = \{a, g, t, c\}$, $h_W(a) = t, h_W(g) = c, h_W(t) = a, h_W(c) = g.$

Watson-Crick 0L-rendszerek

Definíció

Watson-Crick 0L rendszer (W0L rendszer) alatt egy olyan $W = (\Sigma, P, \omega, \varphi)$ rendezett 4-est értünk, ahol

- Σ DNS-típusú ábécé egy h_W Watson-Crick morfizmussal,
- $G_W = (\Sigma, P, \omega)$ egy 0L rendszer,
- $\varphi : \Sigma^* \to \{0, 1\}$ egy olyan leképezés, amelyre $\varphi(\omega) = \varphi(\epsilon) = 0$ és minden $u \in \Sigma^*$ szóra, amelyre $\varphi(u) = 1$ igaz, $\varphi(h_W(u)) = 0$ teljesül.

A komplementerre váltást előidéző feltétel (**trigger**) az, hogy $\varphi(u) = 1$, amit úgy értelmezünk, hogy a szó **„rossz”** lett. Amíg $\varphi(u) = 0$ teljesül, addig a szó **„jó”**. Feltesszük, hogy minden „rossz” szó WC-komplementere „jó”, fordítva, nem feltétlen kell így legyen.

Ha az 0L rendszer D0L,E0L, stb, akkor Watson-Crick D0L, E0L stb. rendszerről beszélünk (röviden WD0L, WE0L stb. rendszerek).

Watson-Crick 0L-rendszerek – levezetés

Definíció

$z_1, z_2 \in V^*$ szavak esetében $z_1 \Longrightarrow_W z_2$ írható, ha

- $z_1 \Longrightarrow_{G_W} z_2$ és $\varphi(z_2) = 0$ vagy
- $z_1 \Longrightarrow_{G_W} h_W(z_2)$ és $\varphi(h_W(z_2)) = 1$.

Példa:

$\Sigma = \{a, g, t, c\}$, $h_W(a) = t, h_W(g) = c, h_W(t) = a, h_W(c) = g$.
$P = \{a \to \varepsilon, a \to ca, g \to cat, c \to ta, t \to \varepsilon\}, \quad \omega = gac$.

A $\varphi(u) = 1 \Leftrightarrow catcat \subseteq u$ (*catcat* részszava *u*-nak) trigger nem teljesíti W0L rendszer φ-re vonatkozó feltételét. Ilyenkor blokkolhat a rendszer, pl. a *catcatgtagta* szóra nem lehetne tovább haladni. Ezért szükséges ez a feltétel.

Most viszont nem generálható blokkoló szó (*g* csak trigger után lehet a szóban, de ekkor nincs *c*). Maga az axióma blokkolhat.

Egy levezetés: $gac(\Rightarrow catcata) \stackrel{\text{trigger}}{\Longrightarrow} gtagtat(\Rightarrow catcatca) \stackrel{\text{trigger}}{\Longrightarrow}$
$gtagtagt(\Rightarrow (cat)\varepsilon\varepsilon(cat)\varepsilon(ca)(cat)\varepsilon) \stackrel{\text{trigger}}{\Longrightarrow} gtagtagtgta.$

Standard Watson-Crick 0L-rendszerek

Egy szó kielégíti a standard trigger komplemens átírásra vonatkozó feltételét (azaz a szó rossz), ha több pirimidint (felülhúzott betűt) tartalmaz, mint purint (nem felülhúzottat). Formálisan:

Definíció

$\Sigma = \{a_1, \ldots, a_n, \bar{a}_1, \ldots, \bar{a}_n\}$,
$\Sigma_{\mathrm{PUR}} = \{a_1, \ldots, a_n\}$, $\Sigma_{\mathrm{PYR}} = \{\bar{a}_1, \ldots, \bar{a}_n\}$.
Ekkor

$$\varphi_{\mathrm{STD}}(u) = \begin{cases} 0 & \text{ha } \sum_{a \in \Sigma_{\mathrm{PUR}}} |w|_a \geqslant \sum_{a \in \Sigma_{\mathrm{PYR}}} |w|_a \\ 1 & \text{ha } \sum_{a \in \Sigma_{\mathrm{PUR}}} |w|_a < \sum_{a \in \Sigma_{\mathrm{PYR}}} |w|_a \end{cases}$$

definiálja a standard komplementerre váltási (trigger) feltételt.

Standard Watson-Crick E0L-rendszer – Példa

Példa: $W = (V, T, P, \omega, \varphi_{\mathrm{STD}})$, ahol
$V = \{S, S_1, F, E, A, B, X, X_1, X_2, a, b, \bar{S}, \bar{S}_1, \bar{F}, \bar{E}, \bar{A}, \bar{B}, \bar{X}, \bar{X}_1, \bar{X}_2, \bar{a}, \bar{b}\}$,

$$\begin{aligned} P = \{&S \to SF\bar{E} \mid \bar{S}_1, \quad F \to F, \quad \bar{E} \to \bar{E}, \\ &S_1 \to \bar{X}_1 A, \quad \bar{F} \to \bar{X}_1\bar{X}_1 X \mid \bar{X}_1 X, \quad E \to A, \\ &\bar{X}_1 \to \bar{X}_1, \quad X \to XX_2, \quad A \to A\bar{B}, \quad \bar{B} \to \bar{B}, \quad X_2 \to X_2, \\ &X_1 \to \varepsilon, \quad \bar{X} \to \varepsilon, \quad \bar{X}_2 \to \varepsilon, \quad \bar{A} \to a, \quad B \to b\}, \end{aligned}$$

$T = \{a, b\}$, $\omega = S$.

$$S \overset{(m-1)}{\Longrightarrow} S(F\bar{E})^{m-1} (\Longrightarrow \bar{S}_1(F\bar{E})^{m-1}) \overset{\text{trigger}}{\Longrightarrow} S_1(\bar{F}E)^{m-1}$$
$$\Longrightarrow \bar{X}_1 A(\bar{X}_1 X A)^{n-1}(\bar{X}_1\bar{X}_1 X A)^{m-n} \qquad (1 \leqslant n \leqslant m)$$

Lehet más a sorrendjük. Most $2m-1$ purin és $2m-n$ pirimidin van. Az $\bar{X}_1 \to \bar{X}_1, X \to XX_2, A \to A\bar{B}, \bar{B} \to \bar{B}, X_2 \to X_2$ szabályokkal $m-1$ új purinnal és m pirimidinnel nő a szó hossza. Így éppen n lépés után fordít át a standard trigger.

$$\overset{(n-1)}{\Longrightarrow} \alpha (\Longrightarrow \bar{X}_1 A\bar{B}^n(\bar{X}_1 X X_2^n A\bar{B}^n)^{n-1}(\bar{X}_1\bar{X}_1 X X_2^n A\bar{B}^n)^{m-n}) \overset{\text{trigger}}{\Longrightarrow}$$
$$X_1\bar{A}B^n(X_1\bar{X}\bar{X}_2^n\bar{A}B^n)^{n-1}(X_1X_1\bar{X}\bar{X}_2^n\bar{A}B^n)^{m-n} \Longrightarrow (ab^n)^m.$$

Tehát $L(W) = \{(ab^n)^m \mid m \geqslant n \geqslant 1\}$. (ET0L-lel nem generálható.)

Watson-Crick D0L rendszerek hálózatai

Watson-Crick D0L rendszerek hálózatai egy közös DNS típusú ábécé feletti véges sok páronként összeköttetésben lévő D0L rendszerből állnak. A rendszer determinisztikus és számításának egy üteme a következőkből áll:

1. a hálózat csúcsaiban elhelyezkedő determinisztikus Lindenmayer rendszerek az aktuális szóhalmazon szinkronizált módon átírást végeznek. A csúcsokban determinisztikus átírást alkalmazunk, azaz valójában minden csúcshoz egy homomorfizmus van hozzárendelve,
2. majd az így nyert szavakat a Watson-Crick komplementaritás elvét alapul vevő valamilyen kommunikációs protokollt használva egymásnak közvetítik.

A csúcsokból szavak sosem törlődnek, így a számítás eredménye egy kitüntetett csúcsban keletkező szavak összessége (azaz a $\leqslant n$ lépésű számítások limesze, $n \to \infty$).

Watson-Crick D0L rendszerek hálózatai

Definíció

$\Gamma = (\Sigma, P_1, \omega_1, \dots, P_r, \omega_r, \varphi)$ **N_rWD0L rendszer** ($r \geqslant 1$ Watson-Crick D0L rendszer hálózata) ha a következők teljesülnek:

- $\Sigma = \{a_1, \dots, a_n, \bar{a}_1, \dots, \bar{a}_n\}$ DNS-típusú ábécé
- $G_i = (\Sigma, P_i, \omega_i)$ D0L rendszer ($1 \leqslant i \leqslant r$),
- $\varphi : \Sigma^* \to \{0, 1\}$ egy olyan leképezés (a rendszer triggere), amelyre $\varphi(\omega_i) = \varphi(\epsilon) = 0$ ($1 \leqslant i \leqslant r$) és minden $u \in \Sigma^*$ szóra, amelyre $\varphi(u) = 1$ igaz, $\varphi(h_W(u)) = 0$ teljesül.

G_i a rendszer i-edik komponense (csúcsa) ($1 \leqslant i \leqslant r$), jelölje g_i a G_i által meghatározott homomorfizmust.

Az 1-es számú komponens kitüntetett, **mesterkomponensnek** nevezzük.

Definíció

Egy N_rWD0L rendszert **standardnak** nevezünk, ha $\varphi = \varphi_{\mathrm{STD}}$.

Watson-Crick D0L rendszerek állapotai

Definíció

Egy $\Gamma = (\Sigma, P_1, \omega_1, \ldots, P_r, \omega_r, \varphi)$ N_rWD0L rendszer **állapota** egy olyan $(L_1, \ldots, L_r)$ rendezett r-es, ahol $\forall 1 \leqslant i \leqslant r$ esetén $L_i \subseteq \Sigma^*$ és $\forall w \in L_i$-re $\varphi(w) = 0$.
$(\{\omega_1\}, \ldots, \{\omega_r\})$ a Γ hálózat **kezdeti állapota**.

NWD0L rendszerek kommunikációs protokolljai

Definíció

$s_1 = (L_1, \ldots, L_r)$-ből **közvetlenül levezethető** $s_2 = (L'_1, \ldots, L'_r)$ az **(a) protokoll szerint** (jelölése $s_1 \Longrightarrow_{(a)} s_2$), ha

$L'_i = C'_i \cup \bigcup_{j=1}^{r} h_W(B'_j)$, ahol

$C'_i = \{g_i(v) \mid v \in L_i, \varphi(g_i(v)) = 0\}$ és

$B'_j = \{g_j(u) \mid u \in L_j, \varphi(g_j(u)) = 1\}$.

Definíció

$s_1 = (L_1, \ldots, L_r)$-ből **közvetlenül levezethető** $s_2 = (L'_1, \ldots, L'_r)$ a **(b) protokoll szerint** (jelölése $s_1 \Longrightarrow_{(b)} s_2$), ha

$L'_i = h_W(B'_i) \cup \bigcup_{j=1}^{r} C'_j$, ahol

$B'_i = \{g_i(u) \mid v \in L_i, \varphi(g_i(u)) = 1\}$ és

$C'_j = \{g_j(v) \mid u \in L_j, \varphi(g_j(v)) = 0\}$.

NWD0L rendszerek kommunikációs protokolljai

Tehát mindkét protokoll esetében, miután WD0L módon alkalmaztuk a derivációs lépést, a csúcspont megtartja a helyes és a kijavított szavakat (a helytelen szavak komplemensét), és

- az (a) protokoll esetében az összes kijavított szó másolatát küldi el a többi csúcsnak,
- a (b) protokoll esetében pedig az összes helyes szóét.

A két protokollt két különböző kommunikációs stratégia motiválja:

- az (a) protokoll szerint a csúcspontok a detektált hibák kijavításáról informálják egymást,
- a (b) protokoll szerint a kapott helyes szavakról.

NWD0L rendszerek kommunikációs protokolljai

1. Példa: $r = 3$

$\Sigma_{\mathrm{PUR}} = \{a, g\}, \Sigma_{\mathrm{PYR}} = \{t, c\}$, $a \xleftrightarrow{h_W} t$, $g \xleftrightarrow{h_W} c$, φ_{STD} trigger.

(a) protokoll szerint:

$L_1' = C_1' \cup h_W(B_1') \cup h_w(B_2') \cup h_W(B_3')$.

$L_2' = h_W(B_1') \cup C_2' \cup h_w(B_2') \cup h_W(B_3')$.

$L_3' = h_W(B_1') \cup h_w(B_2') \cup C_3' \cup h_W(B_3')$.

$a \to ac$	$a \to c$	$a \to a$
$g \to g$	$g \to g$	$g \to ga$
$t \to t$	$t \to tt$	$t \to t$
$c \to c$	$c \to c$	$c \to c$

$L_1 = \{acg, taa\}$
$C_1' = \{accg\}$
$B_1' = \{tacac\}$
$h_W(B_1') = \{atgtg\}$
$L_1' = \{accg, atgtg, a, cgg\}$

$L_2 = \{a, gaa\}$
$C_2' = \{\}$
$B_2' = \{c, gcc\}$
$h_W(B_2') = \{a, cgg\}$
$L_2' = \{atgtg, a, cgg\}$

$L_3 = \{atg, aa\}$
$C_3' = \{atga, aa\}$
$B_3' = \{\}$
$h_W(B_3') = \{\}$
$L_3' = \{atga, aa, atgtg, a, cgg\}$

NWD0L rendszerek kommunikációs protokolljai

2. Példa: $r = 3$

$\Sigma_{\mathrm{PUR}} = \{a, g\}, \Sigma_{\mathrm{PYR}} = \{t, c\}$, $a \overset{h_W}{\longleftrightarrow} t$, $g \overset{h_W}{\longleftrightarrow} c$, φ_{STD} trigger.

(b) protokoll szerint:

$L'_1 = C'_1 \cup h_W(B'_1) \cup C'_2 \cup C'_3.$

$L'_2 = C'_1 \cup C'_2 \cup h_w(B'_2) \cup C'_3.$

$L'_3 = C'_1 \cup C'_2 \cup C'_3 \cup h_W(B'_3).$

$a \to ac$	$a \to c$	$a \to a$
$g \to g$	$g \to g$	$g \to ga$
$t \to t$	$t \to tt$	$t \to t$
$c \to c$	$c \to c$	$c \to c$

$L_1 = \{acg, taa\}$
$C'_1 = \{accg\}$
$B'_1 = \{tacac\}$
$h_W(B'_1) = \{atgtg\}$
$L'_1 = \{accg, atgtg, atga, aa\}$

$L_2 = \{a, gaa\}$
$C'_2 = \{\}$
$B'_2 = \{c, gcc\}$
$h_W(B'_2) = \{a, cgg\}$
$L'_2 = \{accg, atga, a, cgg, aa\}$

$L_3 = \{atg, aa\}$
$C'_3 = \{atga, aa\}$
$B'_3 = \{\}$
$h_W(B'_3) = \{\}$
$L'_3 = \{accg, atga, aa\}$

Az NWD0L rendszer által generált nyelv

Adott (x) protokoll szerint ($x \in \{a, b\}$) **többlépéses levezetést** a szokásos módon, $\Longrightarrow_{(x)}$ reflexív, tranzitív lezártjaként definiáljuk és $\Longrightarrow^*_{(x)}$-gal jelöljük.

Definíció

A $\Gamma = (\Sigma, P_1, \omega_1, \ldots, P_r, \omega_r, \varphi)$ NWD0L rendszer által (x) protokollal ($x \in \{a, b\}$) **generált nyelv**
$L_{(x)}(\Gamma) = \bigcup \{L_1 \mid (\{\omega_1\}, \ldots, \{\omega_r\}) \Longrightarrow^*_{(x)} (L_1, \ldots, L_r)\}$.

Azaz a generált nyelv a működés során valamikor a mesterkomponensbe bekerülő szavak halmaza. A generált nyelvhez tartozó szavak halmaza iterációról iterációra folyamatosan bővül.

W0L és NWD0L rendszerek számítási ereje

Néhány érdekesebb eredmény:

- A WD0L rendszereket V. Michalche és A. Salomaa vezették be (1997), az NWD0L-rendszereket Csuhaj Varjú E. és A. Salomaa (2003).
- Minden Turing-kiszámítható függvény kiszámítható WD0L rendszerrel (P. Sosík)
- A standard Watson-Crick E0L rendszerek illetve a standard Watson-Crick ETD0L rendszerek számítási ereje megegyezik a Turing gépekével. (Csima J., Csuhaj Varjú E., A. Salomaa)
- A standard NWD0L rendszerek számítási ereje megegyezik a Turing gépekével (Csuhaj Varjú E.) Sőt, a Turing teljességhez elegendő a generált szavak nem üres prefixeit kommunikálni. (Csuhaj Varjú E.)
- NWD0L rendszerek segítségével a rendszer masszív párhuzamosságát kihasználva lineáris időben oldhatók meg NP-teljes problémák (SAT, Hamilton kör). (Csuhaj Varjú E., A. Salomaa)