

# Számítási modellek

## 10. előadás

# A DNS számítások

DNS-számítások alatt olyan számítási modelleket és módszereket értünk, amelyeket a DNS szálak struktúrája, tulajdonságai, a szálakon végezhető műveletek, valamint a szálak viselkedése motivál.

Tom Head (1987): a DNS szálak rekombináns viselkedésének formális nyelvi modellezése (elméleti megközelítés)

Leonard M. Adleman (1994): kísérlet a Hamilton-út probléma egy példányának megoldására DNS szálakat manipuláló kémcsőrendszerek segítségével (gyakorlati megközelítés)

# A DNS szerkezete

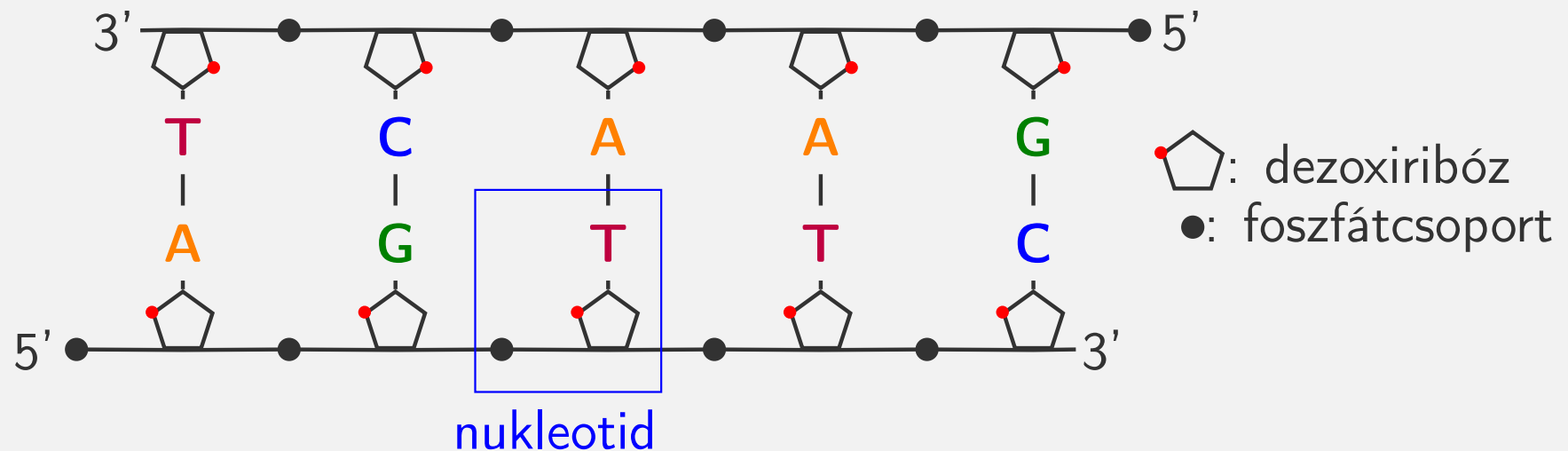
A DNS egy nukleinsav. A nukleinsavak ismétlődő nukleotid egységekből álló nagy méretű molekulák (polimerek). Minden nukleotid három egymáshoz kapcsolódó elemből áll:

- ▶ egy nitrogéntartalmú szerves bázisból (ez lehet adenin(A), timin(T), citozin(C), guanin(G))
- ▶ egy pentózcukorból (dezoxiribóz)
- ▶ egy foszfátcsoporthoz

A polimer váza a nukleotidok foszfodiészter kötéssel egymáshoz kapcsolódó dezoxiribóz részeiből áll. A foszfodiészter kötés az egyik nukleotid cukor komponensének 3'OH-csoportja és a következő cukorkomponensének 5'OH-ja között található egy foszfátcsoport közbeiktatásával.

A szerkezet változó része az egymást követő nukleotidok bázisainak a sorrendje, ez a bázissorrend határozza meg az információt.

# A DNS szerkezete



A 4 nukleotidbázis közül kettő 9 atomos (adenin és guanin), ezek a **purinok**, kettő 6 atomos (timin és citozin), ezek a **pirimidinek**.

A DNS-ben két hosszú nukleotidszál kapcsolódik egymáshoz a bázispárok segítségével. A párok mindig A-T és G-C. A DNS a térben kettős spirál szerkezetű.

Az egyik szál egyik végén a cukor 5. szénatomjához végéhez még csatlakozik egy foszfátcsoport (röviden a nukleotidlánc 5' vége), a másik végén a cukor 3. szénatomjához egy OH csoport csatlakozik (a nukleotidlánc 3' vagy 3'OH vége). A komplementer szálon éppen fordítva vannak az 5' és 3' végek.

# DNS rekombináció

A restrikciós enzimek (vagy restrikciós endonukláázok) képesek felismerni egy rövid nukleotidszekvenciát (részszót) a kétszálú DNS-en belül és a DNS-t adott helyeken elvágni.

Példa:

T C G A  
A G C T

G C G C  
C G C G

G C G C  
C G C G

A létrejött fragmensek:

5' – C C C T C G A C C C – 3'  
3' – G G G A G C T G G G – 5'

5' – C C C T C G A C C C – 3'  
3' – G G G A G C T G G G – 5'

5' – A A A G C G C A A A – 3'  
3' – T T T C G C G T T T – 5'

5' – A A A G C G C A A A – 3'  
3' – T T T C G C G T T T – 5'

5' – T T T G C G C T T T – 3'  
3' – A A A C G C G A A A – 5'

5' – T T T G C G C T T T – 3'  
3' – A A A C G C G A A A – 5'

# DNS rekombináció

A különböző restrikciós (vágó) enzimek különféle alakú vágási végeket eredményeznek. Az egymáshoz illeszkedő vágási végek ún. ligáz (ragasztó) enzimek hatására a kémiai feltételek megléte esetén összeilleszthetők.

A egyik lehetőség persze, hogy az imént elvágott molekulát egy ligáz újra, ugyanúgy illeszti össze. Izgalmasabb azonban, ha az összeillő vágási végek összeragasztásával új, hibrid molekulák keletkeznek.

Az előző példában az első négy fragmens rekombinációja a következő új DNS-eket eredményezheti:

5' – C C C T  C G C A A A – 3'  
3' – G G G A G C  G T T T – 5'

5' – A A A G  C G A C C C – 3'  
3' – T T T C G C  T G G G – 5'

Ugyanakkor az utolsó 2 fragmens nem tud az első 4 fragmans egyikével se kombinálódni.

# Leonard M. Adleman kísérlete

Az egyik első kísérlet biocomputer építésére. Ötlet: használjuk ki a DNS rekombináció jelenségében rejlő masszív párhuzamosságot egy nem hatékonyan kiszámítható probléma, például az NP-teljes Hamilton-út probléma egy példányának megoldására.

Adott néhány város (Adlemannál 7), közülük néhányat nonstop repülőjárat köt össze. Létezik-e olyan út Brisbane és Alice Springs között, amely mindegyik várost pontosan egyszer érinti?

**Bemenet:**  $G$  irányított gráf  $n$  csúccsal és két kitüntetett csúccsal, amelyeket  $v_{\text{start}}$ -tal és  $v_{\text{end}}$ -del jelölünk.

**1. lépés:**  $G$  -beli sétákat generálunk véletlenszerűen, nagy mennyiségben.

**2. lépés:** Csak azokat a sétákat tartjuk meg, amelyek  $v_{\text{start}}$ -tal kezdődnek és  $v_{\text{end}}$ -del végződnek.

**3. lépés:** Csak az  $n$  hosszú sétákat tartjuk meg.

**4. lépés:** Csak azokat a sétákat tartjuk meg, amik utak, azaz minden csúcsot érintenek.

**Kimenet:** YES, ha marad út, és NO egyébként.

## Adleman kísérlete

A program inputjának kódolásához a gráf mindegyik  $i$  csúcsához rendeljünk hozzá egy véletlen 20-mert (húsz nukleotid hosszúságú egyszeres DNS szálrészletet), amelyet  $s_i$  -vel,  $(1 \leq i \leq n)$ , jelölünk.  $s_1$  legyen a  $v_{\text{start}}$ -hoz,  $s_n$  a  $v_{\text{end}}$ -hez tartozó 20-mer.

Pl.  $i = 2, 3, 4$ -re Adleman a következő 20 hosszúságú oligonukleotidokat (nukleotidlánc töredékeket) használta:

$s_2 = \text{TATCGGATCGGTATATCCGA},$

$s_3 = \text{GCTATTTCGAGCTTAAAGCTA},$

$s_4 = \text{GGCTAGGTACCAGCATGCTT}.$

Ezen oligonukleotidok a  $5' \rightarrow 3'$  irányt követik.



# Adleman kísérlete – Watson-Crick morfizmus

A  $h$  **Watson-Crick morfizmus** a 4 DNS bázist a Watson-Crick komplementére képezi:

$$h(A) = T, h(T) = A, h(C) = G, h(G) = C.$$

$h$  kiterjeszthető egy  $\{C, T, A, G\}^* \rightarrow \{C, T, A, G\}^*$  homomorfizmussá.  $h$ -t betűnként alkalmazhatjuk a DNS szálra, pl.:  
 $h(CATTAG) = GTAATC$ .

Így  $h$  a DNS szálak Watson-Crick komplementét adja vissza ( $h$  megváltoztatja az orientációt: az eredeti szál az  $5' \rightarrow 3'$  irányba mutat, míg a Watson-Crick komplement a  $3' \rightarrow 5'$  irányba.)

Pl.

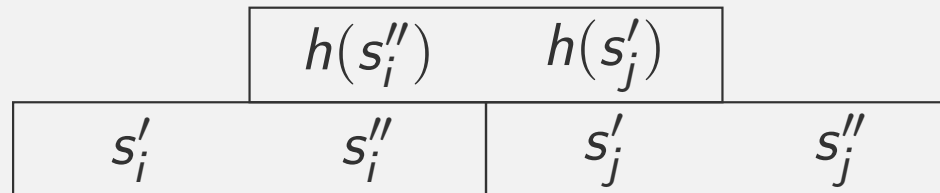
$$\begin{aligned} s_2 &= TATCGGATCGGTATATCCGA, \\ h(s_2) &= ATAGCCTAGCCATATAGGCT, \end{aligned}$$

$$\begin{aligned} s_3 &= GCTATTTCGAGCTTAAAGCTA, \\ h(s_3) &= CGATAAGCTCGAATTTTCGAT. \end{aligned}$$

# Adleman kísérlete

Mindegyik  $s_i$ -t két 10-hosszú szálra bontjuk:  $s_i = s'_i s''_i$ .  
 $s'_i$ -t illetve  $s''_i$ -t úgy tekintjük, mint  $s_i$  első illetve második felét.

Vegyük észre, hogy a  $e_{i \rightarrow j} := h(s''_i s'_j)$  oligonukleotid illeszkedik az  $s_i s_j$  40-merhez.



Pl.

$e_{2 \rightarrow 3} = \text{CATATAGGCTCGATAAGCTC} ,$

$e_{3 \rightarrow 2} = \text{GAATTTTCGATATAGCCTAGC} ,$

$e_{3 \rightarrow 4} = \text{GAATTTTCGATCCGATCCATG} .$

Vegyük észre, hogy ez a konstrukció megkülönbözteti az élek orientációját.  $e_{2 \rightarrow 3}$  és  $e_{3 \rightarrow 2}$  különböző 20-merek.

# Adleman kísérlete

**1. lépés:** Minden  $i$  csúcsra és  $i \rightarrow j$  élre a gráfban,  $s_i$  és  $e_{i \rightarrow j}$  oligonukleotidokat keverünk össze nagy mennyiségben egy egyszerű ligációs reakcióban. Így a DNS molekulává kapcsolódás során ha  $i \rightarrow j$  éle a gráfnak, akkor  $s_i$  és  $s_j$  következhet egymás után egy nukleotid láncban. Mivel az oligonukleotidok kellően hosszúak, ezért a ligációs reakció során kialakuló láncok megfelelnek a véletlen sétáknak.

**2. lépés:** Szálakra bontjuk az így kapott DNS darabokat. A polimeráz-láncreakció (PCR) nevű eljárás során kellő mennyiségben rendelkezésre álló építőkö (dezinukleotid-trifoszfát) és mesterséges kémiai feltételek mellett egy minta DNS szálát akár többmillió példányban lehet megsokszorozni. Ez a DNS szál két végéhez komplementesen illeszkedő ún. primer DNS szál darabok nagy számban való hozzáadásával történik. Ha az  $s_1$  és  $h(s_n)$  primerekkel PCR-t alkalmazunk az 1. lépés eredményére akkor ezután a kémcső szinte csak  $v_{\text{start}}$ -ból  $v_{\text{end}}$ -be menő sétákat fog tartalmazni, igen nagy számban.

# Adleman kísérlete

**3. lépés:** A gélelektroforézis alapelve, hogy a töltéssel rendelkező molekulák elektromos térben, össztöltésüknek megfelelően, az ellentétes töltésű elektróda felé vándorolnak. A rövidek gyorsabban, a hosszabbak lassabban. A gélelektroforézist a DNS szálak hossz alapján történő elválasztására használják. Csak a  $20n$  nukleotidpárból álló kettős spirálokat hagyjuk meg, ezek  $n$  hosszú sétákat kódolnak.

**4. lépés:** Az ún. affinity purification (vonzalom alapú szűrés) olyan technikai eljárás, mely során kiszűrhetjük egy adott bázissorozatot tartalmazó nukleotidláncokat. A DNS negatív töltésű. Az  $x$  részsztót tartalmazó szálak egy  $h(x)$ -et tartalmazó fémgolyóhoz tapadnak. Ezek után mágneses mező hatására az  $x$ -et nem tartalmazó szálak kiszűrhetők.

Ezt tegyük meg sorra  $h(s_1), \dots, h(s_n)$ -nel, rendre eldobva a nem illeszkedő láncokat. Akkor és csak akkor van Hamilton út  $v_{\text{start}}$ -ból  $v_{\text{end}}$ -be, ha az eljárás végén a kémcső legalább egy DNS-molekulát tartalmaz.

# DNS számítások vs Neumann-elvű számítógép

- ▶ Adleman kísérlete 7 napig tartott egy 7 csúcsú gráfra, csekély volt a gyakorlati jelentősége.
- ▶ A DNS számításokkal potenciálisan nagyságrendekkel nagyobb párhuzamosság érhető el, mint Neumann típusú computerekkel.
- ▶ A molekulák kis mérete miatt potenciálisan nagyobb tárhelykapacitás érhető el.
- ▶ A kémiai reakciók lezajlását ki kell várni, így a DNS számítások jelenleg sokkal lassabbak, mint egy modern számítógép számításai. Ezt kéne a masszív párhuzamosságnak kompenzálnia.
- ▶ Egy Neumann típusú számítás eredménye sokkal könnyebben értelmezhető, mint egy kémiai kísérlet eredménye.
- ▶ Bár Adleman kísérlete óta történtek gyakorlati előrelépések, hatékony biocomputerek építése egyelőre gyerekcipőben jár.

# A DNS-rekombináció matematikai modelljei

A DNS-rekombináció matematikai megfelelője a **splicing** (összeragasztás) művelete. Legyen  $V$  egy ábécé (például a DNS esetében  $V = \{c, g, a, t\}$ ).

- ▶ Tom Head modellje (1987): Adott  $B$  és  $C$   $(\alpha, \mu, \beta)$  alakú rendezett 3-asok halmaza, ahol  $\alpha, \mu, \beta \in V^*$ . Legyenek  $(\alpha, \mu, \beta)$  és  $(\alpha', \mu, \beta')$  vagy mindketten  $B$ -beli, vagy mindketten  $C$ -beli 3-asok továbbá  $u\alpha\mu\beta v$  és  $p\alpha'\mu\beta'q$  két  $V$  feletti szó. Ekkor a splicing művelet eredménye  $u\alpha\mu\beta'q$ .

(Itt  $B$  és  $C$  a  $3' \rightarrow 5'$  és  $5' \rightarrow 3'$  orientációknak felel meg, míg a  $\mu$  **keresztveződés** a ligációhoz szükséges illeszkedésének.)

- ▶ Gheorghe Păun modelljében (1996) a szabályok  $u_1 \# u_2 \$ u_3 \# u_4$  alakúak és az  $x_1 u_1 u_2 x_2$  és  $y_1 u_3 u_4 y_2$   $V$  feletti szavakra alkalmazva a művelet eredménye  $x_1 u_1 u_4 y_2$ .
- ▶ Dennis Pixton modelljében (1996) a szabályok  $(\alpha, \alpha'; \beta)$  alakúak és az  $u\alpha v$  és  $p\alpha'q$   $V$  feletti szavakra  $u\beta q$  a művelet eredménye.

# A DNS-rekombináció matematikai modelljei

## Megjegyzések:

- ▶ Păun modellje általánosabb Head modelljénél. Az  $(\alpha, \mu, \beta)$  és  $(\alpha', \mu, \beta')$  Head-féle szabálpárnak megfelel az  $\alpha\mu\#\beta\$ \alpha'\mu\#\beta'$  szabály.
- ▶ Pixton modellje általánosabb mindkét modellnél.  
Az  $(\alpha, \mu, \beta)$  és  $(\alpha', \mu, \beta')$  Head-féle szabálpárnak megfelel az  $(\alpha\mu\beta, \alpha'\mu\beta'; \alpha\mu\beta')$  Pixton-szabály.  
A  $u_1\#u_2\$u_3\#u_4$  Păun-féle szabálynak megfelel az  $(u_1u_2, u_3u_4; u_1u_4)$  Pixton-szabály.
- ▶ Megmutatható, hogy van olyan reguláris nyelv, mely előállítható véges nyelvből Pixton-féle szabályokkal, de nem állítható elő Păun-féle szabályokkal, illetve olyan ami előállítható Păun-féle szabályokkal, de nem állítható elő Head-féle szabályokkal.
- ▶ Néha nem 1 hanem 2 szó a művelet eredménye. Például:  
 $p\alpha'\mu\beta v$  is (Head modell)  
 $y_1u_3u_2x_2$  is (Păun modell)

# A splicing művelet

A továbbiakban ha külön nem említjük a Păun-féle változatra szorítkozunk.

## Definíció

Egy  $(V$  ábécé feletti)  $r = u_1 \# u_2 \$ u_3 \# u_4$  szabályt **splicing szabálynak** nevezünk, ha  $u_i \in V^*$ ,  $1 \leq i \leq 4$ , valamint  $\#$  és  $\$$  speciális szimbólumok, amelyek bem  $V$ -beliek.

## Definíció

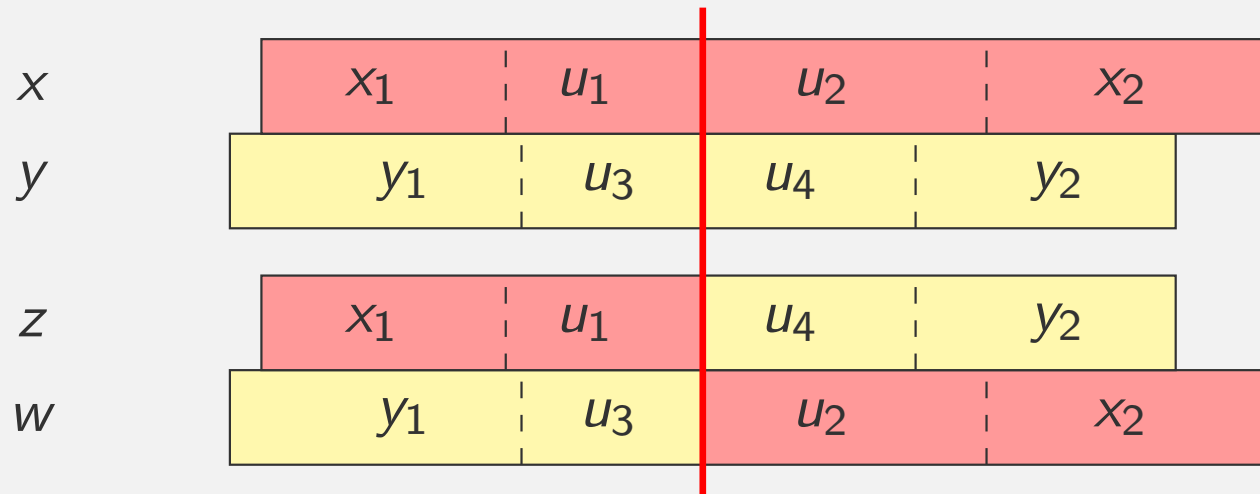
Az  $r = u_1 \# u_2 \$ u_3 \# u_4$  splicing szabályra és  $x, y, z, w \in V^*$  sztringekre

- ▶  $(x, y) \vdash_r z$  akkor és csak akkor áll fenn, ha  $x = x_1 u_1 u_2 x_2$ ,  $y = y_1 u_3 u_4 y_2$  és  $z = x_1 u_1 u_4 y_2$ . valamely  $x_1, x_2, y_1, y_2 \in V^*$  -ra (**1-splicing reláció**),
- ▶  $(x, y) \vdash_r (z, w)$  akkor és csak akkor áll fenn, ha  $x = x_1 u_1 u_2 x_2$ ,  $y = y_1 u_3 u_4 y_2$ ,  $z = x_1 u_1 u_4 y_2$  és  $w = y_1 u_3 u_2 x_2$ , valamely  $x_1, x_2, y_1, y_2 \in V^*$ -ra (**2-splicing reláció**).



# A splicing művelet

Az  $r = u_1 \# u_2 \$ u_3 \# u_4$  2-splicing szabály illusztrálása az  $x = x_1 u_1 u_2 x_2$  és  $y = y_1 u_3 u_4 y_2$  szavakra:



- ▶ 1-splicing: a művelet eredménye egyetlen szó ( $z = x_1 u_1 u_4 y_2$ ), ilyenkor  $(x, y) \vdash_r z$
- ▶ 2-splicing: a művelet eredménye két szó ( $z = x_1 u_1 u_4 y_2$  és  $w = y_1 u_3 u_2 x_2$ )

Ha világos, hogy melyik szabályról van szó, akkor  $\vdash_r$  helyett röviden  $\vdash$  írható.

# H-rendszer – nemiterált eset

## Definíció

**Splicing rendszernek** (vagy **H-rendszernek**) nevezzük az  $S = (V, A, R)$  rendezett 3-ast, ahol  $V$  egy ábécé,  $A \subseteq V^*$  a kezdeti sztringek (axiómák) nyelve és  $R$  ( $V$  feletti) splicing szabályok halmaza.

Ekkor definiálhatjuk az egylépéses 1-splicing műveletet egy  $L \subseteq V^*$  nyelvre, melynek eredménye az  $L$ -beli sztringeken minden lehetséges módon végrehajtott  $R$  szerinti 1-splicingok eredményeinek halmaza.

## Definíció

Legyen  $S = (V, A, R)$  egy H-rendszer és  $L \subseteq V^*$ . Ekkor az  $L$  nyelv **1-splicingja**  $\sigma_1(L) = \{z \in V^* \mid (x, y) \vdash_r z, \text{ ahol } x, y \in L, r \in R\}$ .

(Az 1 alsó index az 1-splicingra utal.)

**Megjegyzés:** Ha  $\sigma_1(x, y) = \{z \in V^* \mid (x, y) \vdash_r z, \text{ ahol } r \in R\}$ , akkor  $\sigma_1(L) = \bigcup_{x, y \in L} \sigma_1(x, y)$ .

# H-rendszer számítási ereje – nemiterált eset

Ha az  $S = (V, A, R)$  H-rendszerre és FL1, FL2 nyelvcsaládokra  $A \in \text{FL1}$  és  $R \in \text{FL2}$ , akkor  $S$ -et **(FL1, FL2) típusúnak** mondjuk.

## Jelölés

Adott FL1 és FL2 nyelvcsaládokra használjuk a következő jelölést:

$$S_1(\text{FL1}, \text{FL2}) := \{L \mid L = \sigma_1(A) \text{ valamely } S = (V, A, R) \\ \text{H-rendszerre ahol } A \in \text{FL1} \text{ és } R \in \text{FL2}\}.$$

## Jelölés:

- FIN: véges nyelvek nyelvcsaládja,
- REG: reguláris nyelvek nyelvcsaládja,
- LIN: lineáris nyelvek nyelvcsaládja (olyan CF grammatikával generálható, ahol a szabályok jobboldalán a terminálisokon kívül  $\leq 1$  nemterminális áll).
- CF: környezetfüggetlen nyelvek nyelvcsaládja,
- CS: környezetfüggő nyelvek nyelvcsaládja,
- RE: rekurzíve felsorolható nyelvek nyelvcsaládja.

# H-rendszer számítási ereje – nemiterált eset

**Tételek**  $S_1(\text{FL1}, \text{FL2})$ -ről a következőket tudjuk:

FL1\FL2	FIN	REG	LIN	CF	CS	RE
FIN	FIN	FIN	FIN	FIN	FIN	FIN
REG	REG	REG	REG,LIN	REG,CF	REG,RE	REG,RE
LIN	LIN, CF	LIN,CF	RE	RE	RE	RE
CF	CF	CF	RE	RE	RE	RE
CS	RE	RE	RE	RE	RE	RE
RE	RE	RE	RE	RE	RE	RE

(Ha egy cellában két nyelvcsalád szerepel: szigorúan köztük van.)

# H-rendszer – iterált eset

## Definíció

Legyen  $S = (V, A, R)$  egy H-rendszer és  $L \subseteq V^*$ . Ekkor az  $L$  nyelv  $\sigma_1^*(L)$  **iterált 1-splicingját** a következőképpen definiáljuk:

$$\sigma_1^*(L) := \bigcup_{i \geq 0} \sigma_1^i(L),$$

ahol

$$\sigma_1^0(L) := L, \quad \sigma_1^{i+1}(L) := \sigma_1(\sigma_1^i(L)), \quad i \geq 0.$$

Tehát  $\sigma_1^*(L)$  a legkisebb nyelv, amely tartalmazza  $L$ -et és zárt az 1-splicing műveletére nézve.

## Definíció

Az  $S = (V, A, R)$  H-rendszer által **generált 1-splicing nyelv** alatt  $L_1(S) := \sigma_1^*(A)$ -t értjük.

# H-rendszer által generált nyelv – példák

1. Legyen  $S = (\{a\}, \{a\}, \{a\#\$a\})$  Könnyen látható, hogy  $L_1(S) = a^+$ .
2. Legyen  $S = (V, A, R)$ , ahol  $\{a, b, c\}$ ,  
 $A = \{cabb, cabab, cabaab\}$  és  $R = \{caba\#a\$cab\#a\}$ . Mivel  $cab$  mindig csak a szavak elejére illeszkedhet ezért  $L_1(S) = caba^*b$  lesz.
3. Véges  $A$  axiómarendszerre  $(aa)^* \neq \sigma_1^*(A)$ .  
Indirekt, tegyük fel, hogy  $L_1(S) = (aa)^*$ , ahol  $S = (V, A, R)$  és nyilván  $V = \{a\}$ ,  $A \subseteq V^*$  és  $R \neq \emptyset$ . Legyen  $r \in R$ , ekkor  $r = a^k\#a^\ell\$a^m\#a^n$  valamely  $k, \ell, m, n \in \mathbb{N}$ -re. Tekintsük az  $u = a^{k+\ell+x+1}$  és a  $v = a^{m+n+y}$  szavakat  $x, y \in \{0, 1\}$  értékeket úgy választva, hogy mindkét szó hossza (azaz  $k + \ell + x + 1$  és  $m + n + y$ ) páros legyen. Ekkor  $u, v \in L_1(S)$ , és az  $r$  szabály alkalmazható az  $(u, v)$  párra: (1)  $u$ -t a  $k$ .,  $v$ -t az  $m + y$ . betű után vágjuk el, ekkor  $(u, v) \vdash_r a^{k+n}$ , (2)  $u$ -t a  $k + 1$ .,  $v$ -t az  $m + y$ . betű után vágjuk el, ekkor  $(u, v) \vdash_r a^{k+n+1}$  adódik. Egyik ptl. hosszú, ellentmondás.

# H-rendszer számítási ereje – iterált eset

## Jelölés

Adott FL1 és FL2 nyelvcsaládokra használjuk a következő jelölést:

$$H_1(\text{FL1}, \text{FL2}) := \{L \mid L = \sigma_1^*(A) \text{ valamely } S = (V, A, R) \\ \text{H-rendszerre ahol } A \in \text{FL1 és } R \in \text{FL2}\}.$$

**Tételek**  $H_1(\text{FL1}, \text{FL2})$ -ről a következőket tudjuk:

FL1\FL2	FIN	REG	LIN	CF	CS	RE
FIN	FIN,REG	FIN,RE	FIN,RE	FIN,RE	FIN,RE	FIN,RE
REG	REG	REG,RE	REG,RE	REG,RE	REG,RE	REG,RE
LIN	LIN, CF	LIN,RE	LIN,RE	LIN,RE	LIN,RE	LIN,RE
CF	CF	CF,RE	CF,RE	CF,RE	CF, RE	CF,RE
CS	CS,RE	CS,RE	CS,RE	CS,RE	CS,RE	CS,RE
RE	RE	RE	RE	RE	RE	RE

(Ha egy cellában két nyelvcsalád szerepel: szigorúan köztük van.)

## 2-splicing

### Definíció

Legyen  $S = (V, A, R)$  egy H-rendszer és  $L \subseteq V^*$ . Ekkor az  $L$  **2-splicingja**  $\sigma(L) = \{w \in V^* \mid (x, y) \vdash_r (w, z) \text{ vagy } (x, y) \vdash_r (z, w), \text{ ahol } x, y \in L, r \in R\}$ .

### Definíció

Legyen  $S = (V, A, R)$  egy H-rendszer és  $L \subseteq V^*$ . Ekkor az  $L$  nyelv  $\sigma^*(L)$  **iterált 2-splicingját** a következőképpen definiáljuk:

$$\sigma^*(L) := \bigcup_{i \geq 0} \sigma^i(L),$$

ahol

$$\sigma^0(L) := L, \quad \sigma^{i+1}(L) := \sigma(\sigma^i(L)), \quad i \geq 0.$$

### Definíció

Az  $S = (V, A, R)$  H-rendszer által **generált 2-splicing nyelv** alatt  $L(S) := \sigma^*(A)$ -t értjük.



## 2-splicing

### Jelölés

Adott FL1 és FL2 nyelvcsaládokra vezessük be az alábbi jelöléseket:

$S(\text{FL1}, \text{FL2}) := \{L \mid L = \sigma(A) \text{ valamely } S = (V, A, R)$   
H-rendszerre ahol  $A \in \text{FL1}$  és  $R \in \text{FL2}\}$ .

$H(\text{FL1}, \text{FL2}) := \{L \mid L = \sigma^*(A) \text{ valamely } S = (V, A, R)$   
H-rendszerre ahol  $A \in \text{FL1}$  és  $R \in \text{FL2}\}$ .

**Megjegyzés:** Az 1-splicing erősebb, mint a 2-splicing. Ha egy  $(V, A, R)$  1-splicing rendszer minden  $u_1 \# u_2 \$ u_3 \# u_4 \in R$  szabálya esetén az  $u_3 \# u_4 \$ u_1 \# u_2 \in R$  teljesül, akkor ez valójában ekvivalens egy 2-splicing rendszerrel.

**Állítás:** Van olyan nyelv, ami 1-splicing rendszerrel generálható, de 2-splicing rendszerrel nem.

# Univerzalitási alaplemma

## Tétel

Minden  $L \in \text{RE}$ ,  $L \subseteq T^*$  nyelv felírható a következő alakban:  
 $L = L_0 \cap T^*$ , valamely  $L_0 \in H_1(\text{FIN}, \text{REG})$ -re.

**Bizonyítás** Tekintsük a  $G = (N, T, P, S)$  0-típusú grammatikát és alkalmazzuk az  $U = N \cup T \cup \{B\}$  jelölést, ahol  $B$  új szimbólum.

Megkonstruáljuk az  $S = (V, A, R)$  H-rendszert, ahol  
 $V = N \cup T \cup \{X, X', B, Y, Z\} \cup \{Y_\alpha \mid \alpha \in U\}$ .

$R$  a következő 1-splicing szabályokat tartalmazza.

# Univerzalitási alaplemma

A grammatikai szabályok szimulálása:

1)  $X\omega\#uY\$Z\#vY$ , ahol  $u \rightarrow v \in P, \omega \in U^*$ ,

A mondatforma elforgatása:

2)  $X\omega\#\alpha Y\$Z\#Y_\alpha$ , ahol  $\alpha \in U, \omega \in U^*$ ,

3)  $X'\alpha\#Z\$X\#\omega Y_\alpha$ , ahol  $\alpha \in U, \omega \in U^*$ ,

4)  $X'\omega\#Y_\alpha\$Z\#Y$ , ahol  $\alpha \in U, \omega \in U^*$ ,

5)  $X\#Z\$X'\#\omega Y$ , ahol  $\omega \in U^*$ ,

Terminálás:

6)  $\#ZY\$XB\#\omega Y$ , ahol  $\omega \in T^*$ ,

7)  $\#Y\$XZ\#$ .

Legyen végül az axiómák halmaza

$$A = \{XBSY, ZY, XZ\} \cup \{ZvY \mid u \rightarrow v \in P\} \cup \{ZY_\alpha, X'\alpha Z \mid \alpha \in U\}.$$

Azt fogjuk belátni, hogy  $L = \sigma_1^*(A) \cap T^*$ .

# Univerzalitási alaplemma

Az állítás bebizonyításához megvizsgáljuk, hogy  $S$  hogyan működik, vi. azt, hogy hogyan kaphatunk  $T^*$ -beli sztringeket.

Egyik  $A$  -beli sztring sem  $T^*$ -beli. Viszont minden  $R$ -beli szabály tartalmaz  $Z$ -t. Az 1-splicing szabályaink viszont olyanok, hogy nem keletkezik  $Z$ -t tartalmazó új sztring. Ezért minden lépésben egy  $XBSY$ -ből származtatott és  $A - \{XBSY\}$ -beli axiómát kell használnunk. (Két  $A - \{XBSY\}$ -beli se kombinálódhat.)

**Állítás:** minden új sztring  $\beta w_1 B w_2 Y$  vagy  $\beta w_1 B w_2 Y_\alpha$  alakú, ahol  $\beta \in \{X, X'\}$  és  $w_2 w_1$  illetve  $w_2 w_1 \alpha$   $G$  aktuális mondatformája és ezek a sztringek elő is állíthatók.

A 2)-5) típusú szabályokkal olyan új sztringeket kaphatunk, mely egy korábbi,  $X$  és  $Y$  által határolt  $\gamma$  szót ciklikusan elforgat. Ha  $\gamma$ -t addig forgatjuk, hogy egy  $G$ -beli szabály baloldala kerüljön a szó végére ( $Y$  elé), akkor az 1) típusú szabályokkal a mondatformából újabb (esetleg ciklikusan elforgatott) mondatformát kaphatunk.

# Univerzalitási alaplemma

Kicsit részletesebben:

- ▶ A 2-es típusú szabály előállít olyan sztringet egy korábbiából, hogy az utolsó betűt ( $\alpha$ -t) törli, a törölt betűt azonban megjegyzi  $Y$  indexében
- ▶ az előző sztringet csak 3-es típusú szabállyal lehet folytatni, ez az  $\alpha$ -t beírja  $X$  elé (és  $X$ -ből  $X'$  lesz).
- ▶ az előző sztringet csak 4-es típusú szabállyal lehet folytatni, ez most törli az  $Y$  indexében feleslegessé vált információt
- ▶ az előző sztringet csak 5-ös típusú szabállyal lehet folytatni, ez  $X'$ -ből újra  $X$ -et csinál.

Összeségében egy betű a szó végéről (a közvetlenül  $Y$  előtti pozícióból) a szó elejére kerül át (közvetlenül  $X$  mögé).

Vegyük észre, hogy az aktuális ciklikus mondatformához mindig tartozik pontosan egy  $B$  marker, amelyik a ciklikus szóban a tényleges első betűt jelöli, így nem veszik el az az információ, hogy valójában hol kezdődik a mondatforma.

# Univerzalitási alaplemma

Az egyetlen mód, hogy a nem  $T$ -beli szimbólumokat eltávolítsuk a  $\sigma$  alkalmazásával kapott sztringekből az az, hogy a 6) és 7) csoport szabályait használjuk. Pontosabban az  $XB$  szimbólumokat csak akkor tudjuk eltávolítani, ha a következő feltételek teljesülnek:

- ▶  $Y$  jelen van (ha először a 7. csoportba tartozó szabályt használnánk, akkor  $Y$  eltávolítása miatt további splicing műveletben a sztring nem vehetne részt, de nem is lenne  $T^*$ -beli),
- ▶  $B$  az  $X$  után, baloldalon van (a szó vissza van forgatva), és
- ▶ az  $XB$  és  $Y$  által közrefogott aktuális sztring csak  $T$ -beli szimbólumokból áll (azaz nincsenek  $N$ -beli jelek).

$X$  és  $B$  eltávolítása után,  $Y$ -t szintén eltávolíthatjuk. A fenti megfontolások alapján nyilván az így kapott  $T^*$ -beli sztringek  $L(G)$ -beliek (azaz  $\sigma_1^*(A) \cap T^* \subseteq L(G)$ ) és fordítva, minden  $L(G)$ -beli sztring előállítható ilyen módon (azaz  $L(G) \subseteq \sigma_1^*(A) \cap T^*$ ). Ezekből  $\sigma_1^*(A) \cap T^* = L(G)$  adódik, amit bizonyítani kellett. □

# Kiterjesztett H rendszer

El tudnánk-e érni más módon a Turing-unverzalitást (bármely RE-beli nyelv generálhatóságát) véges axiómahalmaz és véges szabályrendszer esetén?

## Definíció

**Kiterjesztett H-rendszer** alatt a  $S = (V, T, A, R)$  négyest értjük, ahol

- ▶  $V$  egy ábécé,
- ▶  $T \subseteq V$  a terminális ábécé,
- ▶  $A \subseteq V^*$  a kezdeti sztringek (axiómák) halmaza, és
- ▶  $R \subseteq V^* \# V^* \$ V^* \# V^*$  splicing szabályok halmaza, ahol  $\$, \# \notin V$ .

## Definíció

Az  $S = (V, T, A, R)$  **kiterjesztett H-rendszer által generált 1-splicing nyelv**  $L_1^T(S) := \sigma_1^*(A) \cap T^*$ .

# Kiterjesztett H-rendszerek számítási ereje

## Jelölés

FL1 és FL2 nyelvcsaládokra,  $EH_1(\text{FL1}, \text{FL2})$  jelöli azon  $L_1^T(S)$  alakú nyelvek családját, amelyeket az  $S = (V, T, A, R)$  kiterjesztett H rendszer generál, ahol  $A \in \text{FL1}$  és  $R \in \text{FL2}$ .

**Tételek:**  $EH_1(\text{FL1}, \text{FL2})$ -ről a következőket tudjuk:

FL1\FL2	FIN	REG	LIN	CF	CS	RE
FIN	REG	RE	RE	RE	RE	RE
REG	REG	RE	RE	RE	RE	RE
LIN	LIN, CF	RE	RE	RE	RE	RE
CF	CF	RE	RE	RE	RE	RE
CS	RE	RE	RE	RE	RE	RE
RE	RE	RE	RE	RE	RE	RE

(Ha egy cellában két nyelvcsalád szerepel: szigorúan köztük van.)

$EH(\text{FL1}, \text{FL2})$  jelentése hasonló, csak 2-splicingra.



# Kémcsőrendszerek

1996-ban Csuhaj Varjú E., L. Kari és Gh. Păun vezették be a kémcsőrendszerek (test tube systems) számítási modelljét.

## Definíció

$n$ -edfokú ( $n \geq 1$ ) **kémcsőrendszer** alatt egy  $\Gamma = (V, (V_1, A_1, R_1), (V_2, A_2, R_2), \dots, (V_n, A_n, R_n))$  konstrukciót értünk, ahol

- ▶  $V$  egy ábécé,
- ▶  $V_i \subseteq V$  ( $1 \leq i \leq n$ ),
- ▶  $A_i \subseteq V^*$  ( $1 \leq i \leq n$ ),
- ▶  $R_i \subseteq V^* \# V^* \$ V^* \# V^*$  ( $1 \leq i \leq n$ ).

A rendszer  $(V_i, A_i, R_i)$  komponensének neve **kémcső**.

$V_i$ -t az  $i$ -edik kémcső **szelektorának** hívjuk.

Jelölés:  $B = V^* \setminus \bigcup_{i=1}^n V_i^*$ .

**Észrevétel:** Minden  $1 \leq i \leq n$ -re  $S_i = (V, A_i, R_i)$  egy H-rendszer.

# Kémcsőrendszerek– konfiguráció

## Definíció

Az  $(L_1, \dots, L_n)$ ,  $L_i \subseteq V^*$ ,  $1 \leq i \leq n$ , rendezett  $n$ -est a rendszer **konfigurációjának**,  $L_i$ -t pedig az  $i$ -edik kémcső tartalmának hívjuk.

## Jelölés:

Legyen  $\Gamma = (V, (V_1, A_1, R_1), (V_2, A_2, R_2), \dots, (V_n, A_n, R_n))$  egy kémcsőrendszer. Minden  $1 \leq i \leq n$ -re jelölje  $\sigma_i(L)$  az  $L \subseteq V^*$  nyelv  $S_i = (V, A_i, R_i)$  H-rendszer szerinti 2-splicing nyelvét, azaz

$$\sigma_i(L) := \{w \in V^* \mid (x, y) \vdash_r (w, z) \text{ vagy } (x, y) \vdash_r (z, w), \text{ ahol } x, y \in L, r \in R_i\}.$$

$\sigma_i^*(L)$  jelöli az iteratív 2-splicing nyelvet.

# Kémcsőrendszerek– tranzíció

## Definíció

Az  $(L_1, \dots, L_n)$  és  $(L'_1, \dots, L'_n)$  konfigurációkra definiáljuk a **tranzíciót** a következőképpen:

$(L_1, \dots, L_n) \implies (L'_1, \dots, L'_n)$  akkor és csak akkor, ha minden  $1 \leq i \leq n$ -re

$$L'_i = \bigcup_{j=1}^n (\sigma_j^*(L_j) \cap V_i^*) \cup (\sigma_i^*(L_i) \cap B)$$

Minden kémcső tartalmát a saját splicing szabályhalmaza szerint kombináljuk egymással, amíg lehetséges,  $(L_i$ -ről  $\sigma_i^*(L_i)$ -re változtatjuk a tartalmat,  $1 \leq i \leq n$ ), az eredményt pedig szétosztjuk az  $n$  kémcső között a  $V_1, \dots, V_n$  szelektoroknak megfelelően (egy sztringet több kémcső is megkaphat!)

Azok a részek, amelyeket nem tudunk elosztani (nem tartoznak egyetlen  $V_i^*$ -hoz sem,  $1 \leq i \leq n$ ) a kémcsőben maradnak.

# Kémcsőrendszerek – számítás

## Definíció

Egy  $(k$  hosszú,  $k \geq 0$ ) **számításon**  $\Gamma$ -ban konfigurációk egy  $C = \{(L_1^{(t)}, \dots, L_n^{(t)}) \mid 0 \leq t \leq k\}$  sorozatát értjük, úgy hogy

- ▶  $(L_1^{(0)}, \dots, L_n^{(0)}) = (A_1, \dots, A_n)$  és
- ▶  $(L_1^{(t)}, \dots, L_n^{(t)}) \implies (L_1^{(t+1)}, \dots, L_n^{(t+1)})$  minden  $0 \leq t \leq k-1$ -re.

Ha van olyan  $C$  számítás, amellyel  $(L_1, \dots, L_n)$ -ből  $(L'_1, \dots, L'_n)$ -be juthatunk, akkor enne jelölése  $(L_1, \dots, L_n) \implies^* (L'_1, \dots, L'_n)$ .

## Definíció

$\Gamma$  által **generált nyelv**:

$L(\Gamma) = \{w \in V^* \mid \text{létezik } (A_1, \dots, A_n) \implies^* (L_1^{(t)}, \dots, L_n^{(t)}), t \geq 0 \text{ számítás, amelyre } w \in L_1^{(t)}\}$ .

Tehát az 1-es számú, „főkémcsőben” gyűlik a számítás eredménye.

# Kémcsőrendszerek – példa

**Példa:** Tekintsük a  $\Gamma =$

$(\{a, b, c, d, e\}, (\{a, b, c\}, \{cabc, ebd, dae\}, \{b\#c\$e\#bd, da\#e\$c\#a\}), (\{a, b, d\}, \{ec, ce\}, \{b\#d\$e\#c, c\#e\$d\#a\}))$  kémcsőrendszert.

**Állítás:**  $L(\Gamma) \cap ca^+b^+c = \{ca^n b^n c \mid n \geq 1\}$  (Így  $L(\Gamma) \notin \text{REG.}$ )

Teljes indukcióval látható, hogy minden iterációs lépésben az 1-es kémcsőben mindig 2 lépésre van lehetőség (a sorrendjük kötetlen):

- ▶  $(\alpha a^i b^j | c, e | bd) \vdash (\alpha a^i b^{j+1} d, ec) \quad \alpha \in \{c, d\},$
- ▶  $(da | e, c | a^i b^j \alpha) \vdash (da^{i+1} b^j \alpha, ce) \quad \alpha \in \{c, d\}.$

Ezek után nem lehet több splicing műveletet végezni, a kapott  $da^{i+1} b^{j+1} d$  átkerül a 2-es kémcsőbe. Itt is 2 lehetőség van (a sorrend ismét kötetlen):

- ▶  $(\alpha a^i b^j | d, e | c) \vdash (\alpha a^i b^j c, ed) \quad \alpha \in \{c, d\},$
- ▶  $(c | e, d | a^i b^j \alpha) \vdash (ca^i b^j \alpha, de) \quad \alpha \in \{c, d\}.$

Ezek után nem lehet több splicing műveletet végezni, a kapott  $ca^i b^j c$  átkerül a 2-es kémcsőbe. Ebből az állítás következik.

# Kémcsőrendszerek számítási ereje

$TT_n(\text{FL1}, \text{FL2})$  jelöli azon  $L(\Gamma)$  nyelvek családját, melyekre  $\Gamma = (V, (V_1, A_1, R_1), (V_2, A_2, R_2), \dots, (V_m, A_m, R_m))$  valamely  $m \leq n$ -re és  $A_i \in \text{FL1}$ ,  $R_i \in \text{FL2}$  minden  $1 \leq i \leq m$  esetén.

$\Gamma$ -t ilyenkor  $(\text{FL1}, \text{FL2})$  típusúnak mondjuk.

$$TT_*(\text{FL1}, \text{FL2}) = \bigcup_{n \geq 1} TT_n(\text{FL1}, \text{FL2})$$

Érdekesebb tételek (Csuhaj Varjú, Kari, Păun, 1996):

## Tétel

- ▶  $TT_2(\text{FIN}, \text{FIN}) \setminus \text{REG} \neq \emptyset$ ,  $TT_4(\text{FIN}, \text{FIN}) \setminus \text{CF} \neq \emptyset$ ,  
 $TT_6(\text{FIN}, \text{FIN}) \setminus \text{CS} \neq \emptyset$ ,  $TT_6(\text{FIN}, \text{FIN}) \setminus \text{R} \neq \emptyset$ ,
- ▶  $TT_1(\text{FIN}, \text{FIN}) \subset \text{REG} \subset TT_2(\text{FIN}, \text{FIN}) \subseteq TT_3(\text{FIN}, \text{FIN})$   
 $\subseteq \dots \subseteq TT_*(\text{FIN}, \text{FIN}) = TT_*(\text{FL1}, \text{FL2}) = \text{RE}$ , ahol  $\text{FIN} \subseteq \text{FL1}$ ,  $\text{FL2} \subseteq \text{RE}$  tetszőlegesen,
- ▶  $H(\text{FL1}, \text{FL2}) = TT_1(\text{FL1}, \text{FL2})$  minden  $\text{FL1}$ -re és  $\text{FL2}$ -re,
- ▶  $\text{EH}(\text{FL1}, \text{FL2}) \subseteq TT_2(\text{FL1}, \text{FL2})$  minden  $\text{FL1}$ -re és  $\text{FL2}$ -re.