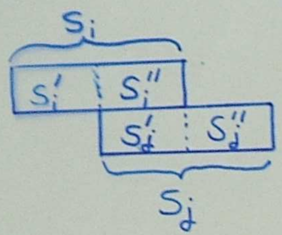
Bemerkungen:

- Die Sequenzen s_i und e_{ij} müssen "genügend zufällig" sein:

  z.B. sollte $s_i'' \neq \overline{s_j'}$ gelten, sonst ist unerwünschte Ligation möglich:



- Adlemans Experiment benötigte ca
  7 Tage.
  4 Stunden Ligation
  $\approx 10^{13}$ Kopien der s_i und e_{ij}

Kap 3. Sticker Systeme

Grammatiken zur Kombination von unvollständiger DNS (Dominos)

wie z.B

```
      CGTAAG             TTC
ATTGCA              TTCAAGTCA
```

Verallgemeinerungen gegenüber biologischer Realität:

- $\{A, T, C, G\}$ wird durch beliebiges endliches Alphabet V ersetzt.
- Watson-Crick-Komplementarität $\{(A,T), (T,A), (C,G), (G,C)\}$ wird durch beliebige symmetrische Relation $\rho \subseteq V \times V$ ersetzt
  $((a,b) \in \rho \Leftrightarrow (b,a) \in \rho)$

Definitionen

- $V \neq \emptyset$ endliches Alphabet
- $\rho \subseteq V \times V$ symmetrische Relation (Komplementarität)
- V^* = Menge aller endlichen Wörter über V (inkl. leerem Wort λ)
- Ein Paar $(x, y) \in V^* \times V^*$ schreiben wir als $\binom{x}{y}$
- Analog: für $L, K \subseteq V^*$ sei $\binom{L}{K} = L \times K$
- Konkatenation von Paaren: $\binom{x_1}{x_2}\binom{y_1}{y_2} = \binom{x_1 y_1}{x_2 y_2}$
- $\begin{bmatrix} V \\ V \end{bmatrix}_\rho = \left\{ \begin{bmatrix} a \\ b \end{bmatrix} \mid a, b \in V, (a, b) \in \rho \right\}$

$WK_\rho(V) = \begin{bmatrix} V \\ V \end{bmatrix}_\rho^*$ - Watson-Crick Domain.

$\begin{bmatrix} a_1 \\ b_1 \end{bmatrix} \begin{bmatrix} a_2 \\ b_2 \end{bmatrix} \cdots \begin{bmatrix} a_n \\ b_n \end{bmatrix} \in WK_\rho(V)$

wird identifiziert mit

$\begin{bmatrix} a_1 a_2 \cdots a_n \\ b_1 b_2 \cdots b_n \end{bmatrix}$ (doppelsträngige Sequenz, Molekül)

Beachte: Wenn wir für $u, v \in V^*$ die Schreibweise $\begin{bmatrix} u \\ v \end{bmatrix}$ verwenden, so bringen wir dadurch implizit zum Ausdruck, dass $|u| = |v|$ und (i-tes Symbol von u, i-tes Symbol von v) $\in \rho$.
Dies wird durch die Schreibweise $\binom{u}{v}$ nicht ausgedrückt.

Wir betrachten $WK_\rho(V)$ als Teilmenge von $V^* \times V^* = \binom{V^*}{V^*}$

$$L_\rho(V) = \left(\binom{\lambda}{V^*} \cup \binom{V^*}{\lambda}\right) \overbrace{\begin{bmatrix} V \\ V \end{bmatrix}_\rho^*}^{= WK_\rho(V)}$$

$$R_\rho(V) = \begin{bmatrix} V \\ V \end{bmatrix}_\rho^* \left(\binom{V^*}{\lambda} \cup \binom{\lambda}{V^*}\right)$$

$$LR_\rho(V) = \left(\binom{\lambda}{V^*} \cup \binom{V^*}{\lambda}\right) \begin{bmatrix} V \\ V \end{bmatrix}_\rho^+ \left(\binom{\lambda}{V^*} \cup \binom{V^*}{\lambda}\right)$$

$$W_\rho(V) = L_\rho(V) \cup R_\rho(V) \cup LR_\rho(V)$$

Menge der unvollständigen Moleküle. (Dominos)

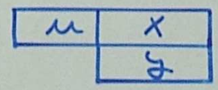
$$W_\rho(V) \setminus \left(\binom{\lambda}{V^*} \cup \binom{V^*}{\lambda}\right) = \text{Menge}$$

der well-started Moleküle:

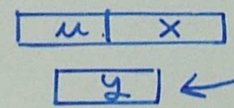
haben mind eine Position (oder Spalte) $\begin{bmatrix} a \\ b \end{bmatrix}$ $= LR_\rho(V)$

Beachte: $\binom{u}{\lambda}\begin{bmatrix} x \\ y \end{bmatrix}$ beinhaltet mehr Information als $\binom{ux}{y}$

$\binom{u}{\lambda}\begin{bmatrix} x \\ y \end{bmatrix}$:



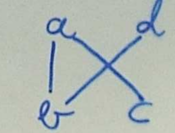
$\binom{ux}{y}$:



keine Ausrichtung

Beispiele: $V = \{a, b, c, d\}$

$\rho = \{(a,b), (b,a), (a,c), (c,a), (b,d), (d,b)\}$



$\begin{bmatrix} adcba \\ cbadb \end{bmatrix} abcd \in R_\rho(V)$

$\begin{bmatrix} adcbd \\ cbaab \end{bmatrix} dab \in R_\rho(V)$

$b\,c\,a\,b\,d\,a\,d\begin{bmatrix} a\,d\,a\,d \\ b\,b\,b\,b \end{bmatrix} \in L_\rho(V)$

$\parallel$

$b\,c\,a\,b\begin{bmatrix} d\,a\,d\,a \\ b\,b\,b\,b \end{bmatrix} d\,a\,d \in LR_\rho(V)$

$\begin{pmatrix} \lambda \\ abcdab \end{pmatrix} \in L_\rho(V) \cap R_\rho(V)$

$\parallel$

$\begin{pmatrix} abcdab \\ \lambda \end{pmatrix} \in L_\rho(V) \cap R_\rho(V)$

$\uparrow$

$\notin LR_\rho(V)$

$a\,b\,c\,c\begin{bmatrix} a\,b\,a \\ c\,d\,c \end{bmatrix} a\,b\,a \in LR_\rho(V)$

Partielle Konkatenation von Dominos (Sticking / Ligation / Annealing)

Seien $x, y \in W_\rho(V)$ mit x well-started

$\hookrightarrow x = x_1 x_2 x_3$ mit

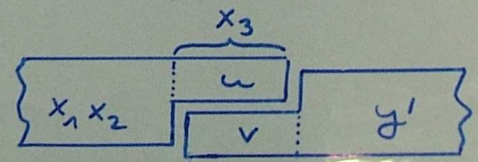
$x_2 \in WK_\rho(V) \setminus \left\{ \begin{bmatrix} \lambda \\ \lambda \end{bmatrix} \right\} = \begin{bmatrix} V \\ V \end{bmatrix}_\rho^+$

$x_1, x_3 \in \begin{pmatrix} V^* \\ \lambda \end{pmatrix} \cup \begin{pmatrix} \lambda \\ V^* \end{pmatrix}$ (überhängende Enden)

Wir definieren $\mu(x, y)$ (Erweiterung von x am ~~linken~~ rechten Ende um y) wie folgt:

1. Fall: $x_3 = \begin{pmatrix} u \\ \lambda \end{pmatrix}$, $y = \begin{pmatrix} \lambda \\ v \end{pmatrix} y'$,
$u, v \in V^*$, $\begin{bmatrix} u \\ v \end{bmatrix} \in WK_\rho(V)$, $y' \in R_\rho(V)$

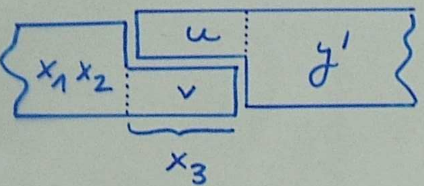
$\Rightarrow \mu(x, y) = x_1 x_2 \begin{bmatrix} u \\ v \end{bmatrix} y'$



2. Fall $\quad x_3 = \begin{pmatrix} \lambda \\ v \end{pmatrix}, \; y \in \begin{pmatrix} u \\ \lambda \end{pmatrix} y'$

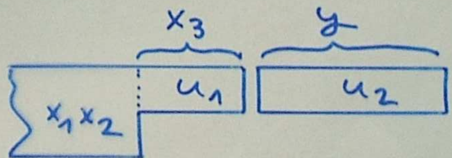
$u, v \in V^*, \; \begin{bmatrix} u \\ v \end{bmatrix} \in WK_p(V), \; y' \in R_p(V)$

$\rightarrow \mu(x,y) = x_1 x_2 \begin{bmatrix} u \\ v \end{bmatrix} y'$



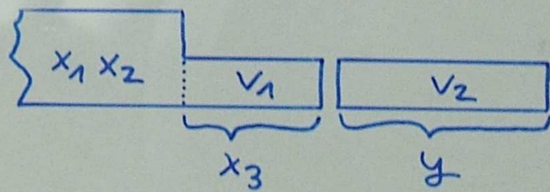
3. Fall: $x_3 = \begin{pmatrix} u_1 \\ \lambda \end{pmatrix}, \; y = \begin{pmatrix} u_2 \\ \lambda \end{pmatrix}, \; u_1, u_2 \in V^*$

$\Rightarrow \mu(x,y) = x_1 x_2 \begin{pmatrix} u_1 u_2 \\ \lambda \end{pmatrix}$



4. Fall: $x_3 = \begin{pmatrix} \lambda \\ v_1 \end{pmatrix}, \; y = \begin{pmatrix} \lambda \\ v_2 \end{pmatrix} \; v_1, v_2 \in V^*$

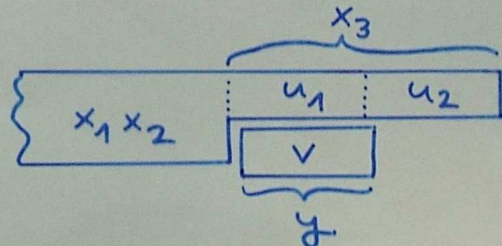
$\Rightarrow \mu(x,y) = x_1 x_2 \begin{pmatrix} \lambda \\ v_1 v_2 \end{pmatrix}$



5. Fall: $x_3 = \begin{pmatrix} u_1 u_2 \\ \lambda \end{pmatrix}, \; y = \begin{pmatrix} \lambda \\ v \end{pmatrix}$

$u_1, u_2, v \in V^*, \; \begin{bmatrix} u_1 \\ v \end{bmatrix} \in WK_p(V)$

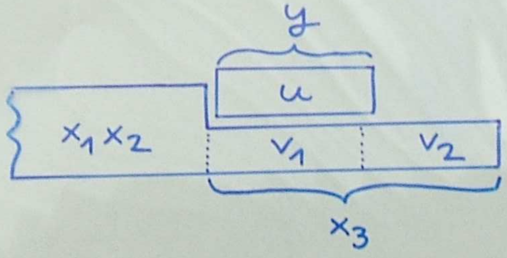
$\Rightarrow \mu(x,y) = x_1 x_2 \begin{bmatrix} u_1 \\ v \end{bmatrix} \begin{pmatrix} u_2 \\ \lambda \end{pmatrix}$



6. Fall: $x_3 = \begin{pmatrix} \lambda \\ v_1 v_2 \end{pmatrix}, \; y = \begin{pmatrix} u \\ \lambda \end{pmatrix}$

$u_1, u_2, v \in V^*, \; \begin{bmatrix} u \\ v_1 \end{bmatrix} \in WK_p(V)$

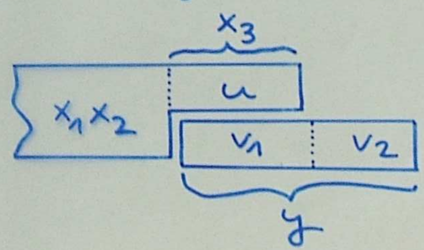
$\Rightarrow \mu(x,y) = x_1 x_2 \begin{bmatrix} u \\ v_1 \end{bmatrix} \begin{pmatrix} \lambda \\ v_2 \end{pmatrix}$



<u>7. Fall</u>: $x_3 = \binom{u}{\lambda}$, $y = \binom{\lambda}{v_1 v_2}$

$u, v_1, v_2 \in V^*$, $\begin{bmatrix} u \\ v_1 \end{bmatrix} \in WK_\rho(V)$

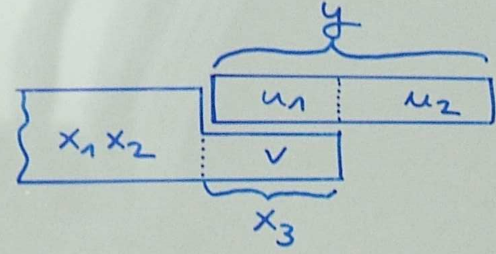
$\Rightarrow \mu(x,y) = x_1 x_2 \begin{bmatrix} u \\ v_1 \end{bmatrix} \binom{\lambda}{v_2}$



<u>8. Fall</u>: $x_3 = \binom{\lambda}{v}$, $y = \binom{u_1 u_2}{\lambda}$,

$u_1, u_2, v \in V^*$, $\begin{bmatrix} u_1 \\ v \end{bmatrix} \in WK_\rho(V)$

$\Rightarrow \mu(x,y) = x_1 x_2 \begin{bmatrix} u_1 \\ v \end{bmatrix} \binom{u_2}{\lambda}$



- In allen anderen Fällen ist $\mu(x,y)$ undefiniert.
- Analog kann $\mu(x,y)$ für $x, y \in W_\rho(V)$ mit y well-started definiert werden (Erweiterung von y am linken Ende durch x)
- <u>Beachte</u>: Sind x und y beide well-started, so gilt:

  Erweiterung von y am linken Ende durch x

  $=$

  Erweiterung von x am rechten Ende durch y

- $\mu'(x,y)$ = Eingeschränktes Sticking von x und y für x oder y well-started

  Analog zu $\mu(x,y)$ definiert, nur Fall 3 und Fall 4 in der Def. vom $\mu(x,y)$ werden weggelassen (d.h. Komplementarität ρ muss eingesetzt werden.

- Für $x = x_1 x_2 x_3 \in W_\rho(V)$ mit $x_2 \in WK_\rho(V) - \left\{ \left[\begin{smallmatrix}\lambda\\ \lambda\end{smallmatrix}\right] \right\}$ und $x_1, x_3 \in \binom{V^*}{\lambda} \cup \binom{\lambda}{V^*}$ ist

  $delay(x) = max\{|x_1|, |x_3|\}$

  Für $x = \binom{u}{\lambda}$ oder $x = \binom{\lambda}{u}$ mit $u \in V^*$ sei $delay(x) = |u|$

Definition (Sticker-System)

Ein Sticker-System ist ein Tupel

$$\gamma = (V, \rho, A, D)$$

mit:
- V ist ein endliches Alphabet
- $\rho \subseteq V \times V$ ist die symmetrische Komplementaritätsrelation.
- $A \subseteq LR_\rho(V)$ ist eine endliche Menge von Axiomen
- $D \subseteq W_\rho(V) \times W_\rho(V)$ endlich

- Sei $\gamma = (V, \rho, A, D)$ ein Sticker-System und seien $x, y \in LR_\rho(V)$.
  Es gilt $x \Longrightarrow_\gamma y$ genau dann, wenn

  $$y = \mu(u, \mu(x, v)) = \mu(\mu(u, x), v)$$

  für ein $(u, v) \in D$ gilt.

- Eine Folge

$$x_1 \Rightarrow_\gamma x_2 \Rightarrow_\gamma x_3 \cdots \Rightarrow_\gamma x_{k-1} \Rightarrow_\gamma x_k$$

mit $x_1 \in A$ ist eine Berechnung in γ, sie ist vollständig, falls $x_k \in WK_p(V)$.

- $$LM_n(\gamma) = \{ w \in WK_p(V) \mid \exists x \in A : x \overset{*}{\Rightarrow}_\gamma w \}$$

D.h. $w \in \overset{LM_n(\gamma)}{\cancel{WK_p(V)}}$ genau dann, wenn es eine vollständige Berechnung in γ gibt, die mit w endet.

LM : Language of Molekules
n : non-restricted

$$L_n(\gamma) = \{ w \in V^* \mid \exists w' \in V^* : \begin{bmatrix} w \\ w' \end{bmatrix} \in LM_n(\gamma) \}$$

- Unterschied zwischen $L_n(\gamma)$ und $LM_n(\gamma)$

Seien $K \subseteq \Gamma^*$, $L \subseteq \Sigma^*$ Sprachen.
K ist eine Kodierung von L genau dann, wenn ein Homomorphismus $f : \Sigma^* \to \Gamma^*$ existiert mit:

- $f(a) \in \Gamma$ für $a \in \Sigma$
- $K = f(L) = \{ f(w) \mid w \in L \}$

Offensichtlich ist $L_n(\gamma)$ eine Kodierung von $LM_n(\gamma)$:

Sei $f : WK_p(V) \to V^*$ der Morphismus mit $f(\begin{bmatrix} a \\ b \end{bmatrix}) = a$

$\hookrightarrow f(LM_n(\gamma)) = L_n(\gamma)$

- Eine vollständige Berechnung
  $A \ni x_1 \Rightarrow_\gamma x_2 \cdots \Rightarrow_\gamma x_{k-1} \Rightarrow_\gamma x_k$
  heißt:
  - <u>primitiv</u>, falls kein $i \in \{1, \dots, k-1\}$ mit $x_i \in WK_\rho(V)$ existiert.
  - <u>mit delay d</u>, falls $(d \in \mathbb{N})$ $delay(x_i) \leq d$ für alle $i \in \{1, \dots, k\}$

- $L_p(\gamma) = \{ w \in L_m(\gamma) \mid w \text{ hat primitive Berechnung} \}$

- $L_d(\gamma) = \{ w \in L_m(\gamma) \mid w \text{ hat Berechnung mit delay } d \}$

<u>Beachte</u>: $L_p(\gamma) \subseteq L_m(\gamma)$

$L_{d_1}(\gamma) \subseteq L_{d_2}(\gamma) \subseteq L_m(\gamma)$ falls $d_1 \leq d_2$

- Sei $\gamma = (V, \rho, A, D)$ ein Sticker-System
  - γ hat <u>beschränkten delay</u>, falls ein d mit $L_d(\gamma) = L_m(\gamma)$ existiert.
  - γ ist <u>einseitig</u>, falls für alle $(u,v) \in D$ gilt: $u = \binom{\lambda}{\lambda}$ oder $v = \binom{\lambda}{\lambda}$
  - γ ist <u>regulär</u>, falls für alle $(u,v) \in D$ gilt: $u = \binom{\lambda}{\lambda}$
  - γ ist <u>simpel</u>, falls für alle $(u,v) \in D$ gilt: $u, v \in \binom{V^*}{\lambda}$ oder $u, v \in \binom{\lambda}{V^*}$

Klassen von Sticker-Sprachen

$$ASL(\alpha) = \{ L_\alpha(\gamma) \mid \gamma \text{ ein Sticker-System} \}$$

für $\alpha \in \{n, p\}$

(A für arbitrary, SL für sticker language)

$$ASL(b) = \{ L_n(\gamma) \mid \gamma \text{ ein Sticker-System mit beschränkten delay} \}$$

Wir ersetzen das A in $ASL(\alpha)$ durch

$$\left\{\begin{array}{l} O \\ R \\ S \\ SO \\ SR \end{array}\right\} \text{ falls Einschränkung auf } \left\{\begin{array}{l} \text{einseitige} \\ \text{reguläre} \\ \text{simple} \\ \text{simple u. einseitige} \\ \text{simple u. reguläre} \end{array}\right\}$$

Sticker-Systeme

(O steht für one-sided = einseitig)

Offensichtliche Beziehungen zwischen Stickersprachen:

$$\begin{tikzcd}[arrows=dash]
 & ASL(\alpha) \arrow[dl] \arrow[dr] & & \\
SSL(\alpha) \arrow[dr] & & OSL(\alpha) \arrow[dl] \arrow[dr] & \\
 & SOSL(\alpha) \arrow[dr] & & RSL(\alpha) \arrow[dl] \\
 & & SRSL(\alpha) &
\end{tikzcd}$$

$XSL(b) \subseteq XSL(n)$ für alle

$X \in \{A, O, R, S, SO, SR\}$

Lemma: $XSL(\alpha) \subseteq CS$ (= Klasse der kontextsensitiven Sprachen) für alle X und α wie oben.

__Beispiel__: $\gamma = (V, \rho, A, D)$ mit

$V = \{a, b, c\}$

$\rho = \{(a,a), (b,b), (c,c)\}$

$A = \{\begin{bmatrix} a \\ a \end{bmatrix}\}$

$D = \{ (\binom{b}{\lambda}, \binom{b}{\lambda}), (\binom{c}{\lambda}, \binom{\lambda}{\lambda}),$

$(\binom{\lambda}{b}, \binom{\lambda}{\lambda}), (\binom{\lambda}{c}, \binom{\lambda}{b}) \}$

γ ist simpel !

$$LM_n(\gamma) = \{ x \begin{bmatrix} a \\ a \end{bmatrix} \begin{bmatrix} b \\ b \end{bmatrix}^m \mid m \geq 0, x \in \{\begin{bmatrix} b \\ b \end{bmatrix}, \begin{bmatrix} c \\ c \end{bmatrix}\}^*, x \text{ enthält genau } m \text{ Vorkommen von } \begin{bmatrix} b \\ b \end{bmatrix} \text{ sowie } \begin{bmatrix} c \\ c \end{bmatrix} \}$$

$$L_n(\gamma) = \{ y a b^m \mid m \geq 0, y \in \{b, c\}^*, |y|_b = |y|_c = m \}$$

- $L_n(\gamma)$ nicht kontextfrei, denn wäre $L_n(\gamma)$ kontextfrei, so wäre auch $L_n(\gamma) \cap c^+ b^+ a b^+ = \{c^m b^m a b^m \mid m \geq 1\}$ kontextfrei $\lightning$ zu Pumping-Lemma.

- $LM_p(\gamma) = LM_n(\gamma)$:

  Jedes Molekül $w \in LM_n(\gamma)$ kann abgeleitet werden, indem zuerst nur die Regeln

  $$(\binom{b}{\lambda}, \binom{b}{\lambda}), (\binom{c}{\lambda}, \binom{\lambda}{\lambda})$$

  angewendet werden, und dann nur die Regeln

  $$(\binom{\lambda}{b}, \binom{\lambda}{\lambda}), (\binom{\lambda}{c}, \binom{\lambda}{b})$$

  angewendet werden

  $\Rightarrow L_p(\gamma) = L_n(\gamma)$ nicht kontextfrei.

Andererseits:

$\forall d \geq 1 : L_d(\gamma) \subsetneq L_m(\gamma)$

Betrachte $w = \begin{bmatrix} c \\ c \end{bmatrix}^m \begin{bmatrix} b \\ b \end{bmatrix}^m \begin{bmatrix} a \\ a \end{bmatrix} \begin{bmatrix} b \\ b \end{bmatrix}^m$

$\in LM_m(\gamma)$ für $m = d+1$

Jede Ableitung von w hat die Form

$$\begin{bmatrix} a \\ a \end{bmatrix} \overset{*}{\Rightarrow}_\gamma \underbrace{\begin{pmatrix} b^k \\ \lambda \end{pmatrix} \begin{bmatrix} b^{m-k} \\ b^{m-k} \end{bmatrix} \begin{bmatrix} a \\ a \end{bmatrix} \begin{pmatrix} b^m \\ \lambda \end{pmatrix}}_{delay \, = \, m > d}$$

$\overset{*}{\Rightarrow}_\gamma w$ für ein $0 \leq k \leq m$

Beachte: Es können zunächst nur die Regeln $\left(\begin{pmatrix} b \\ \lambda \end{pmatrix}, \begin{pmatrix} b \\ \lambda \end{pmatrix}\right)$ und $\left(\begin{pmatrix} \lambda \\ b \end{pmatrix}, \begin{pmatrix} \lambda \\ \lambda \end{pmatrix}\right)$ angewendet werden.

$\Rightarrow c^m b^m a b^m \in L_m(\gamma) \setminus L_d(\gamma)$

Konsequenz aus obigem Beispiel

<u>Satz 1</u> $SSL(m)$ und $SSL(p)$ enthalten nicht-kontextfreie Sprachen.

<u>Satz 2</u> $OSL(m) \subseteq REG$ (Klasse der regulären Sprachen)

zum Beweis von Satz 2 benutzen wir:

<u>Lemma 1</u> Sei $\gamma = (V, \rho, A, D)$ ein einseitiges Sticker System.

Sei $w \in LM_m(\gamma)$. Dann existiert eine Ableitung

$$A \ni w_0 \Rightarrow_\gamma w_1 \Rightarrow_\gamma \cdots \Rightarrow_\gamma w_{m-1} \Rightarrow_\gamma w_m = w$$

mit $delay(w_i) \leq \underline{maxdelay(\gamma)}$.

$maxdelay(\gamma)$ ist wie folgt definiert:

Sei $S := A \cup \{u, v \mid (u,v) \in D\}$

$\Rightarrow maxdelay(\gamma) := \max\{delay(w) \mid w \in S\}$

Beweis von Lemma 1

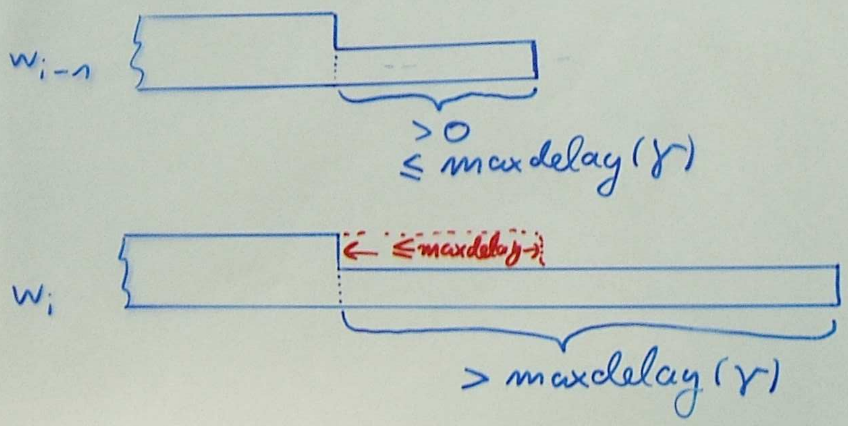
Sei $A \ni w_0 \Rightarrow_\gamma w_1 \Rightarrow_\gamma \cdots \Rightarrow_\gamma w_m = w$ (*)

eine Ableitung für w, und sei $1 \leq i < m$ so, daß gilt:

$delay(w_i) > maxdelay(\gamma)$

$delay(w_j) \leq maxdelay(\gamma)$

für alle $0 \leq j < i$



Durch Umordnen der Ableitungsschritte in (*) können wir $delay(w_i) \leq maxdelay(\gamma)$ erreichen

□

Idee zum Beweis vom Satz 2

Sei $\gamma = (V, \rho, A, D)$ einseitiges Sticker System.

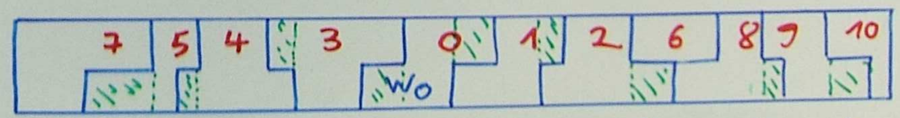
Sei $d = maxdelay(\gamma)$.

Nach Lemma 1 gibt es für jedes $w \in LM_m(\gamma)$ eine Ableitung

$A \ni w_0 \Rightarrow_\gamma w_1 \Rightarrow_\gamma \cdots \Rightarrow_\gamma w_m = w$

mit $delay(w_i) \leq maxdelay(\gamma) = d$

Simuliere diese Ableitung durch eine kontextfreie Grammatik G, wobei w von innen nach außen erzeugt wird:



Die "Überhänge" aus $\binom{V^{\leq d}}{\lambda} \cup \binom{\lambda}{V^{\leq d}}$ bilden die Nichtterminale wobei $V^{\leq d} = \{ v \in V^* \mid |v| \leq d \}$

Die Grammatik G wird außerdem die Eigenschaften aus Aufg. 2, Blatt 2 erfüllen.

$\Rightarrow L(G)$ regulär

Details: Tafel bzw. Buch

Satz 3: $OSL(p) \subseteq REG$

zum Beweis von Satz 3 benötigen wir

Lemma 2 Sei $\gamma = (V, \rho, A, D)$ einseitiges Sticker System.

Sei $w \in LM_p(\gamma)$.

Dann existiert eine primitive Ableitung

$$A \ni w_0 \Rightarrow_\gamma w_1 \Rightarrow_\gamma \cdots \Rightarrow_\gamma w_{m-1} \Rightarrow_\gamma w_m = w$$

mit $delay(w_i) \leq 2 \cdot maxdelay(\gamma)$,

für $0 \leq i \leq m$.

Beweis von Lemma 2

Sei $A \ni w_0 \Rightarrow_\gamma w_1 \Rightarrow_\gamma \cdots w_{m-1} \Rightarrow_\gamma w_m = w$ (*)

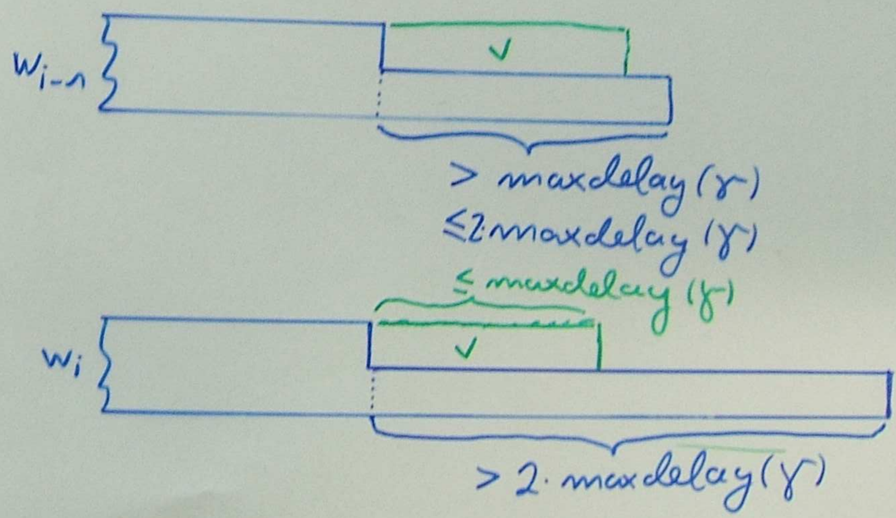
eine primitive Ableitung, und sei $1 \leq i < m$ so, daß gilt:

$delay(w_i) > 2 \cdot maxdelay(\gamma)$

$delay(w_j) \leq 2 \cdot maxdelay(\gamma)$

für $0 \leq j < i$

$\Rightarrow delay(w_{i-1}) > maxdelay(\gamma)$



Durch Umordnen der Ableitungsschritte in (*) können wir $delay(w_i) \leq 2 \cdot maxdelay(\gamma)$ erreichen.

Dabei bleibt die Ableitung primitiv wegen $delay(w_{i-1}) > maxdelay(\gamma)$. $\square$

<u>Beweis von Satz 3</u>

Gleiche Konstruktion wie in Beweis von Satz 2, nur:

- $d := 2 \cdot maxdelay(\gamma)$ anstatt $d = maxdelay(\gamma)$
- In Produktionen vom Typ-2 und Typ-3 verlangen wir zusätzlich $(u_1 \neq \varepsilon$ oder $u_2 \neq \varepsilon)$

Nach Lemma 2 kann die resultierende kontextfreie Grammatik beliebige primitive Ableitungen von γ simulieren. $\square$

<u>Korollar</u>: $SOSL(\alpha) \subsetneq SSL(\alpha)$ für $\alpha \in \{n, p\}$

<u>Beweis</u>: $SSL(n)$ und $SSL(p)$ enthalten nicht-kontextfreie Sprachen (Satz 1). $SOSL(\alpha) \subseteq REG$ für $\alpha \in \{n, p\}$ nach Satz 1 und Satz 3. $\square$

<u>Satz 4</u>. $ASL(b) \subseteq LIN$ (Klasse der linearen Sprachen)

<u>Idee zum Beweis</u>:

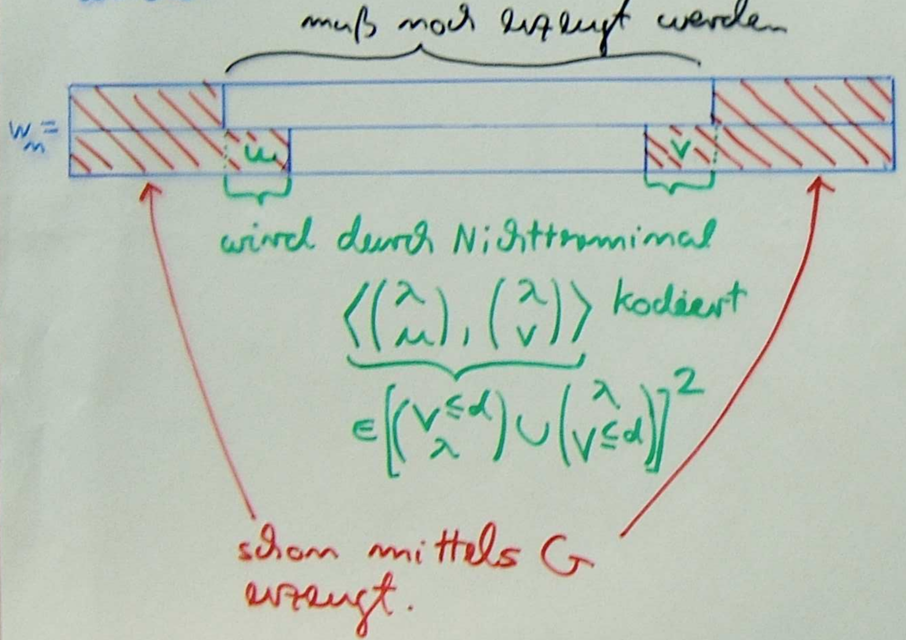
Sei $\gamma = (V, \rho, A, D)$ ein beschränktes Sticker-System und sei $d \in \mathbb{N}$ so, dass $L_d(\gamma) = L_n(\gamma)$.

Simuliere eine Berechnung $delay(w_i) \leq d$

$$A \ni w_0 \Rightarrow_\gamma w_1 \Rightarrow_\gamma w_2 \cdots \Rightarrow_\gamma w_n \in WK_\rho(V)$$

durch eine lineare kontextfreie Grammatik G, wobei w_n von außen nach innen erzeugt wird.

Da $\mathrm{delay}(w_i) \leq d$ für alle $0 \leq i \leq n$, können die Überhängenden Enden durch ein Nichtterminal kodiert werden:



<u>Korollar aus Satz 4</u>: Für jedes Sticker System γ und jede Zahl $d \in \mathbb{N}$ gilt $L_d(\gamma) \in LIN$

<u>Satz 5</u> $REG \subseteq RSL(b) \cap RSL(p)$

<u>Idee zum Beweis</u>:

Sei $M = (K, V, s_0, F, \delta)$ ein endlicher deterministischer Automat.

$K = \{s_0, s_1, \ldots, s_k\}$ = Menge der Zustände
V = Alphabet
s_0 = Anfangszustand
$F \subseteq K$ Menge der Endzustände
$\delta : K \times V \to K$ Transitionsfunktion

Simuliere Berechnung von M durch ein reguläres Sticker-System

$$\gamma = (V, \rho, A, D).$$

Kodiere dabei den Zustand s_i $(0 \leq i \leq k)$ durch ein überhängendes Ende der Länge $\underbrace{i+1}_{\leq k+1} \geq 1$ im oberen Strang.

Satz 6 Für $\alpha \in \{m, p, lr\}$ gilt
$RSL(\alpha) = OSL(\alpha) = REG$.

Beweis:

Satz 2 $\rightarrow$ $OSL(m) \subseteq REG$
Satz 3 $\rightarrow$ $OSL(p) \subseteq REG$

$\Rightarrow OSL(lr) \subseteq OSL(m) \subseteq REG$
$\Rightarrow RSL(\alpha) \subseteq OSL(\alpha) \subseteq REG$ für $\alpha \in \{m, p, lr\}$

Satz 5 $\rightarrow$ $REG \subseteq RSL(lr) \subseteq RSL(m)$
$REG \subseteq RSL(p)$

also $REG \subseteq RSL(\alpha)$ für $\alpha \in \{m, p, lr\}$

□

Satz 7 $LIN \subseteq ASL(lr) \cap ASL(p)$

Idee zum Beweis

Sei $G = (N, T, S, P)$ eine lineare kontextfreie Grammatik.
O.B.d.A enthalte P nur Produktionen der Form

$X \rightarrow aY$, $X \rightarrow Ya$, $X \rightarrow a$
für $X, Y \in N$, $a \in T$

Sei $N = \{X_1, \ldots, X_k\}$ $(k \geq 1)$

Konstruiere Sticker-System

$\gamma = (T, \rho, A, D)$

mit $L_m(\gamma) = L_k(\gamma) = L_p(\gamma) = L(G)$

$\rho = \{(a,a) \mid a \in T\}$

$$A = \left\{ \begin{bmatrix} x \\ x \end{bmatrix} \;\middle|\; x \in L(G),\ |x| \leq 3k+1 \right\}$$

$$\cup \left\{ \binom{u}{\lambda}\begin{bmatrix} x \\ x \end{bmatrix} \;\middle|\; |ux| \leq 3k+1,\ |x| \geq 1,\ |u| = i\ (1 \leq i \leq k),\ X_i \overset{*}{\Rightarrow} ux \right\}$$

$$\cup \left\{ \begin{bmatrix} x \\ x \end{bmatrix}\binom{u}{\lambda} \;\middle|\; |xu| \leq 3k+1,\ |x| \geq 1,\ |u| = i\ (1 \leq i \leq k),\ X_i \overset{*}{\Rightarrow} ux \right\}$$

Regeln D des Sticker Systems γ aus dem Beweis von Satz 6:

(1) $\left(\binom{u}{\lambda}\begin{bmatrix} x \\ x \end{bmatrix}\binom{\lambda}{v}, \begin{bmatrix} z \\ z \end{bmatrix} \right)$ falls

$1 \leq |u|, |v| \leq k$, $|ux| = k+1$

$0 \leq |z| \leq k$, $X_{|u|} \overset{*}{\Rightarrow} ux X_{|v|} z$

(2) $\left(\begin{bmatrix} x \\ x \end{bmatrix}\binom{\lambda}{v}, \begin{bmatrix} z \\ z \end{bmatrix}\binom{u}{\lambda} \right)$ falls

$1 \leq |u|, |v|, |x| \leq k$, $|zu| = k+1$,

$X_{|u|} \overset{*}{\Rightarrow} x X_{|v|} zu$

(3) $\left(\begin{bmatrix} x \\ x \end{bmatrix}\binom{\lambda}{v}, \begin{bmatrix} z \\ z \end{bmatrix} \right)$ falls $1 \leq |v| \leq k$,

$|x| \geq 1$, $|xz| \leq 2k+1$, $S \overset{*}{\Rightarrow} x X_{|v|} z$

(4) $\left(\begin{bmatrix} z \\ z \end{bmatrix}, \binom{\lambda}{v}\begin{bmatrix} x \\ x \end{bmatrix}\binom{u}{\lambda} \right)$ falls

$1 \leq |u|, |v| \leq k$, $|xu| = k+1$

$0 \leq |z| \leq k$, $X_{|u|} \overset{*}{\Rightarrow} z X_{|v|} xu$

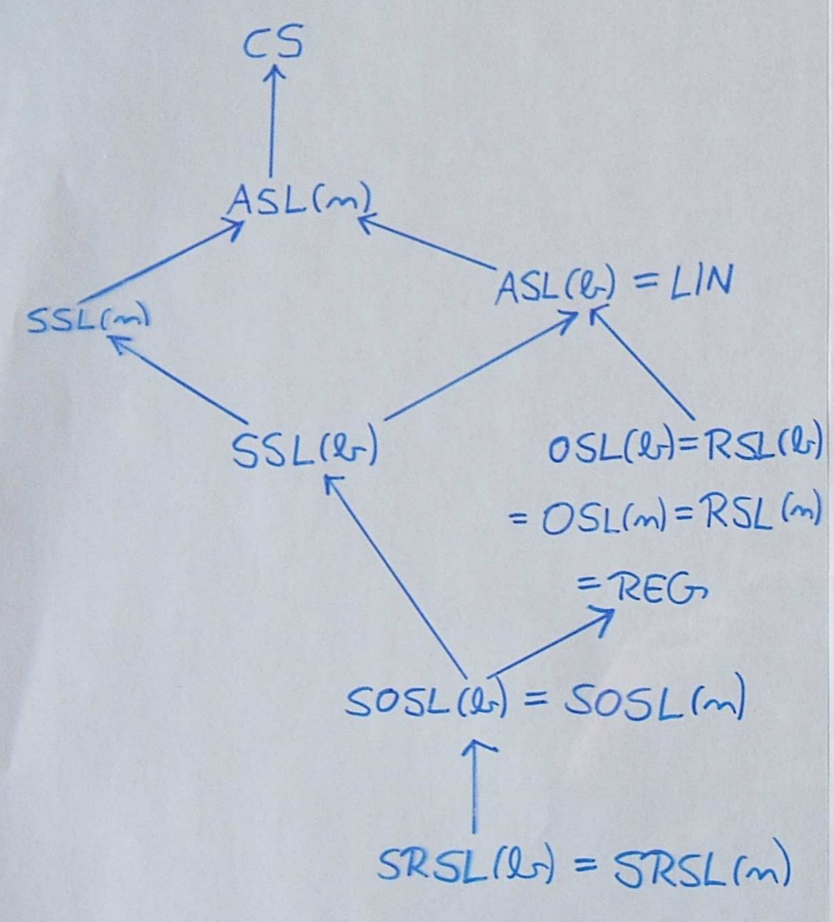
(5) $\left(\binom{u}{\lambda}\begin{bmatrix} z \\ z \end{bmatrix}, \binom{\lambda}{v}\begin{bmatrix} x \\ x \end{bmatrix} \right)$ falls

$1 \le |u|, |v|, |x| \le k, \quad |uz| = k+1$

$X_{|u|} \overset{*}{\Rightarrow} uz\, X_{|v|}\, x$

(6) $\left(\begin{bmatrix} z \\ z \end{bmatrix}, \binom{\lambda}{v}\begin{bmatrix} x \\ x \end{bmatrix} \right)$ falls $1 \le |v| \le k$,

$|x| \ge 1, \quad |zx| \le 2k+1, \quad S \overset{*}{\Rightarrow} z\, X_{|v|}\, x$



Repräsentation von rekursiv aufzählbaren Sprachen durch Sticker-Systeme

Eine Projektion $h: \Sigma^* \to \Gamma^*$ (wobei $\Gamma \subseteq \Sigma$) ist ein Homomorphismus mit

$$h(a) \in \{a, \lambda\}$$

für alle $a \in \Sigma$

Satz 11 Sei $L \in RE$ (Klasse der rekursiv aufzählbaren Sprachen). Dann existiert $L' \in SSL(m) \cap SSL(p)$ und eine Projektion h mit $L = h(L')$.

Zum Beweis von Satz 11 benötigen wir einen Hilfssatz.

- Ein Homomorphismus $h: \Sigma^* \to \Gamma^*$ heißt <u>λ-frei</u>, falls gilt:
  $h(a) \neq \lambda$ für alle $a \in \Sigma$

- Für zwei Homomorphismen $h_1: \Sigma^* \to \Gamma^*$ und $h_2: \Sigma^* \to \Gamma^*$ sei:
  $EQ(h_1, h_2) = \{w \in \Sigma^* : h_1(w) = h_2(w)\}$

<u>Satz 12:</u> Sei $L \subseteq T^*$ rekursiv aufzählbar. Dann existieren λ-frei Homomorphismen h_1 und h_2 sowie eine reguläre Sprache R und eine Projektion π mit: $L = \pi(h_1(EQ(h_1, h_2)) \cap R)$

<u>Beachte:</u> $h_1(EQ(h_1, h_2)) = h_2(EQ(h_1, h_2))$

<u>Beweis:</u> Sei $G = (N, T, S, P)$ eine Typ-0 Grammatik mit $L = L(G)$.

- Wir beschränken uns hier auf den Fall, dass $\lambda \notin L = L(G)$

$\to$ O.B.d.A. gelte $u \neq \lambda \neq v$ für jede Produktion $(u \to v) \in P$.

- Produktionen aus P seien eindeutig mit Markierungen aus einer Menge Lab versehen ($|P| = |Lab|$).

- Sei $T' = \{a' \mid a \in T\}$
  $T'' = \{a'' \mid a \in T\}$
  $Lab' = \{p' \mid p \in Lab\}$

$d: (N \cup T)^* \to (N \cup T')^*$ Morphismus mit $d(A) = A$ für $A \in N$ und $d(a) = a'$ für $a \in T$.

d bijektiv $\to d^{-1}$ existiert.

- Sei $V_1 = N \cup T \cup T' \cup \{B, F, c\}$

  $V_2 = N \cup T \cup T' \cup T'' \cup Lab \cup Lab' \cup \{B, F, c, c'\}$

- Die λ-freien Morphismen $h_1, h_2 : V_2^* \to V_1^*$ seien wie folgt def.:

$h_1(B) = BSc$	$h_2(B) = B$	
$h_1(c) = c$	$h_2(c) = c$	
$h_1(p) = d(v)$	$h_2(p) = d(u)$	falls $p: u \to v \in P$
$h_1(p') = v$	$h_2(p') = d(u)$	falls $p: u \to v \in P$ und $v \in T^+$
$h_1(A) = A$	$h_2(A) = A$	falls $A \in N$
$h_1(a') = a'$	$h_2(a') = a'$	falls $a' \in T'$
$h_1(a'') = a$	$h_2(a'') = a'$	falls $a'' \in T''$
$h_1(a) = F$	$h_2(a) = a$	falls $a \in T$
$h_1(c') = F$	$h_2(c') = c$	
$h_1(F) = F$	$h_2(F) = FF$	

- Sei R die reguläre Sprache

  $R = BS(c(N \cup T')^*)^* c T^* F^+$

- Schließlich sei $\pi : V_1^* \to T^*$ die Projektion mit

  $\pi(x) = x$ für $x \in T$ und
  $\pi(x) = \lambda$ für $x \notin T$.

Dann gilt: $L(G) = \pi(h_1(EQ(h_1, h_2)) \cap R)$ (siehe Tafel) $\square$

<u>Zum Beweis von Satz 11</u>

Sei $L \in RE \xrightarrow{\text{Satz 12}}$

$$L = \pi(h_1(EQ(h_1, h_2)) \cap R)$$

R: $\subseteq V_2^*$, regulär

π: Projektion von V_2^* auf T^* $(T \subseteq V_2)$

h_1, h_2: λ-frei Morphismen von V_1^* nach V_2^*

- Sei $M = (K, V_2, s_0, F, \delta)$ ein deterministischer endlicher Automat mit $L(M) = R$.

- Wir konstruieren ein simples Sticker-System

$$\gamma = (V, \rho, A, D)$$

über dem Alphabet

$$V = V_2 \cup \overline{V_2} \cup K \cup \{\$, E, E', C, Z\},$$

so dass gilt:

$$h(L_n(\gamma)) = h(L_p(\gamma)) = \pi(h_1(EQ(h_1, h_2)), R)$$

Projektion von V^* auf $T^* \subseteq V_2^* \subseteq V^*$

siehe Tafel.