10.1 Sticker Systeme

In this section the basic operation is polymerase, i.e., we glue together some parts of double strands according to the Watson-Crick complementarity; e. g. from the pieces

```
AACGTAGCGATTT          and       GGCCAATAGGGAAACC

    CATCGCTAAACCGG                      TTATCCCT
```

we obtain the double strand

```
AACGTAGCGATTTGGCCAATAGGGAAACC

    CATCGCTAAACCGGTTATCCCT
```

In order to describe the double strands with overhangs we introduce the following notations. Let V be an alphabet, and let $\varrho \subset V \times V$ be a symmetric relation (i.e., $(a, b) \in \varrho$ implies $(b, a) \in \varrho$[1]. We say that a and b are complementary if $(a, b) \in \varrho$.

We set

$$\begin{bmatrix} V \\ V \end{bmatrix}_\varrho = \{ \begin{bmatrix} a \\ b \end{bmatrix} \mid a, b \in V,\ (a, b) \in \varrho \}$$

and consider this set as an alphabet. In the sequel the word $\begin{bmatrix} a_1 \\ b_1 \end{bmatrix} \begin{bmatrix} a_2 \\ b_2 \end{bmatrix} \dots \begin{bmatrix} a_n \\ b_n \end{bmatrix}$ over $\begin{bmatrix} V \\ V \end{bmatrix}_\varrho$ will often be written as $\begin{bmatrix} a_1 a_2 \dots a_n \\ b_1 b_2 \dots b_n \end{bmatrix}$.

The elements of $\begin{bmatrix} V \\ V \end{bmatrix}_\varrho^*$ describe the double strands where the upper and lower part are letter by letter in the relation ϱ. In order to describe the overhangs we set

$$\begin{pmatrix} V^* \\ \lambda \end{pmatrix} = \{ \begin{pmatrix} w \\ \lambda \end{pmatrix} \mid w \in V^* \}$$

and

$$\begin{pmatrix} \lambda \\ V^* \end{pmatrix} = \{ \begin{pmatrix} \lambda \\ w \end{pmatrix} \mid w \in V^* \}\,.$$

The elements of these sets describe single upper strands and single lower strands, respectively. Now we define

$$\begin{array}{rcl}
L_\varrho(V) & = & (\begin{pmatrix} V^* \\ \lambda \end{pmatrix} \cup \begin{pmatrix} \lambda \\ V^* \end{pmatrix}) \begin{bmatrix} V \\ V \end{bmatrix}_\varrho^*, \\
R_\varrho(V) & = & \begin{bmatrix} V \\ V \end{bmatrix}_\varrho^* \begin{pmatrix} V^* \\ \lambda \end{pmatrix} \cup \begin{pmatrix} \lambda \\ V^* \end{pmatrix}), \\
LR_\varrho(V) & = & (\begin{pmatrix} V^* \\ \lambda \end{pmatrix} \cup \begin{pmatrix} \lambda \\ V^* \end{pmatrix}) \begin{bmatrix} V \\ V \end{bmatrix}_\varrho^+ (\begin{pmatrix} V^* \\ \lambda \end{pmatrix} \cup \begin{pmatrix} \lambda \\ V^* \end{pmatrix}), \\
W_\varrho(V) & = & L_\varrho(V) \cup R_\varrho(V) \cup LR_\varrho(V).
\end{array}$$

[1]If one considers the biologically interesting case of $V = \{A, C, G, T\}$, then ϱ is the relation given by the Watson-Crick-complementarity.

Obviously, $L_\varrho(V)$, $R_\varrho(V)$, and $LR_\varrho(V)$ are constructs which describe double strands with overhangs to the left side, to the right side, and to both sides. The three strands given in the beginning of this section can be presented as

$$\binom{\mathtt{AAC}}{\lambda}\begin{bmatrix}\mathtt{GTAGCGATTT}\\ \mathtt{CATCGCTAAA}\end{bmatrix}\binom{\lambda}{\mathtt{CCGG}},$$
$$\binom{\mathtt{GGCC}}{\lambda}\begin{bmatrix}\mathtt{AATAGGGA}\\ \mathtt{TTATCCCT}\end{bmatrix}\binom{\mathtt{AACC}}{\lambda}$$
$$\binom{\mathtt{AAC}}{\lambda}\begin{bmatrix}\mathtt{GTAGCGATTTGGCCAATAGGA}\\ \mathtt{CATCGCTAAACCGGTTATCCCT}\end{bmatrix}\binom{\mathtt{AACC}}{\lambda}$$

We note that $\binom{w}{\lambda}\left[\begin{smallmatrix}w_1\\ w_2\end{smallmatrix}\right]$ cannot be written as (a pair) $\binom{ww_1}{w_2}$ since we loose the information which letters are in the relation ϱ.

Let $x \in LR_\varrho(V)$. Then x can be decomposed as

$$x = x_1x_2x_3 \text{ with } x_2 \in \begin{bmatrix}V\\ V\end{bmatrix}_\varrho^+ \text{ and } x_1, x_3 \in (\binom{V^*}{\lambda} \cup \binom{\lambda}{V^*}). \tag{10.1}$$

Thus

$$x_1 = \binom{w_1}{\lambda} \text{ or } x_1 = \binom{\lambda}{w_1} \quad \text{and} \quad x_3 = \binom{w_3}{\lambda} \text{ or } x_3 = \binom{\lambda}{w_3}$$

for some $w_1 \in V^*$ and $w_3 \in V^*$. We define the delay of x by

$$d(x) = |w_1| + |w_3|.$$

The delay of a word is the sum of its overhangs to the right and to the left. Obviously, a delay can be defined for the elements of $L_\varrho(V)$ and $R_\varrho(V)$, too.

We now define the *sticking operation* $\mu_r : LR_\varrho(V) \times W_\varrho(V) \to LR_\varrho(V)$ which allows the prolongation of an element of $LR_\varrho(V)$ to the right by an element of $W_\varrho(V)$. Let $x \in LR_\varrho(V)$ be decomposed as $x = x_1x_2x_3$ as in (11.6). Let $y \in W_\varrho(V)$. Then we define $\mu_r(x, y)$ as

1. $x_1x_2\left[\begin{smallmatrix}u\\ v\end{smallmatrix}\right]y'$ if $x_3 = \binom{u}{\lambda}$ and $y = \binom{\lambda}{v}y'$ for some $u, v \in V^*$ and $y' \in R_\varrho(V)$,
2. $x_1x_2\left[\begin{smallmatrix}u\\ v\end{smallmatrix}\right]y'$ if $x_3 = \binom{\lambda}{v}$ and $y = \binom{u}{\lambda}y'$ for some $u, v \in V^*$ and $y' \in R_\varrho(V)$,
3. $x_1x_2\left[\begin{smallmatrix}u\\ v\end{smallmatrix}\right]\binom{u'}{\lambda}$ if $x_3 = \binom{uu'}{\lambda}$ and $y = \binom{\lambda}{v}$ for some $u, v, u' \in V^*$ and $y' \in R_\varrho(V)$,
4. $x_1x_2\left[\begin{smallmatrix}u\\ v\end{smallmatrix}\right]\binom{\lambda}{v'}$ if $x_3 = \binom{u}{\lambda}$ and $y = \binom{\lambda}{vv'}$ for some $u, v, v' \in V^*$ and $y' \in R_\varrho(V)$,
5. $x_1x_2\binom{uv}{\lambda}$ if $x_3 = \binom{u}{\lambda}$ and $y = \binom{v}{\lambda}$ for some $u, v \in V^*$,
6. $x_1x_2\left[\begin{smallmatrix}v\\ u\end{smallmatrix}\right]\binom{\lambda}{u'}$ if $x_3 = \binom{\lambda}{uu'}$ and $y = \binom{v}{\lambda}$ for some $u, v, u' \in V^*$,
7. $x_1x_2\left[\begin{smallmatrix}v\\ u\end{smallmatrix}\right]\binom{v'}{\lambda}$ if $x_3 = \binom{\lambda}{u}$ and $y = \binom{vv'}{\lambda}$ for some $u, v, v' \in V^*$,
8. $x_1x_2\binom{uv}{\lambda}$ if $x_3 = \binom{\lambda}{u}$ and $y = \binom{\lambda}{v}$ for some $u, v \in V^*$.

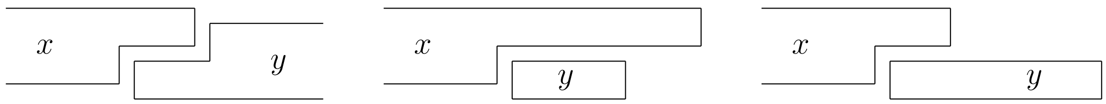

Figure 10.1: Pictorial representation of the operation μ_r in the Cases 1, 3 and 4.

The pictures in Figure 11.14 illustrate the Cases 1, 3 and 4. The reader may verify that the given cases record all possible cases of a continuation to the right (note that $x_3 = \binom{\lambda}{\lambda}$ is allowed).

Obviously, in an analogous way we can define the prolongation to the left by an operation μ_l. We omit the details.

Definition 10.1 *i) A* sticker system *is a quadruple* $G = (V, \varrho, A, D)$ *where*

– V *is an alphabet,*

– $\varrho \subset V \times V$ *is a symmetric relation on* V*,*

– A *is a finite subset of* $LR_\varrho(V)$*, and*

– D *is a finite subset of* $W_\varrho(V) \times W_\varrho(V)$*.*

ii) We say that $y \in LR_\varrho(V)$ *is derived by* $x \in LR_\varrho(V)$ *in one step (written as* $x \Longrightarrow y$*) iff*

$$y = \mu_l(\mu_r(x, y_2), y_1) \text{ for some } (y_1, y_2) \in D\,.$$

(Note that $\mu_l(\mu_r(x, y_2), y_1) = \mu_r(\mu_l(x, y_1), y_2)$ *since the prolongation to the right and to the left are independent from each other.) By* $\stackrel{*}{\Longrightarrow}$ *we denote the reflexive and transitive closure of* $\Longrightarrow$*.*

iii) The molecule language $ML(G)$ *and the word language* $wL(G)$ *generated by* G *are defined by*

$$ML(G) = \{z \mid x \Longrightarrow s{*}z,\ x \in A,\ z \in \begin{bmatrix} V \\ V \end{bmatrix}_\varrho^+ \}$$

and

$$wL(G) = \{w \mid \begin{bmatrix} w \\ v \end{bmatrix} \in ML(G) \text{ for some } v \in V^+\},$$

respectively.

By definition the molecule language of G consists of all double strands without overhangs which can be obtained from the elements of A by simultaneous prolongations to the left and to the right by elements of D. If we restrict to the upper strand of the molecules of the molecule language, then we obtain the word language of G. Obviously, the upper strands can be obtained from the molecules by the homomorphism which maps $\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]$ to a. Thus the word language of a sticker systems is a homomorphic images of its molecule language.

Example 10.2 We consider the sticker system

$$G = (\{a,b,c\}, \{(a,a),(b,b),(c,c)\}, \{\begin{bmatrix} a \\ a \end{bmatrix}\}, D)$$

where

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

We are only interested in molecules in

$$M = ML(G) \cap \begin{bmatrix} c \\ c \end{bmatrix}^* \begin{bmatrix} b \\ b \end{bmatrix}^* \begin{bmatrix} a \\ a \end{bmatrix} \begin{bmatrix} b \\ b \end{bmatrix}^*.$$

Any word in M has a derivation of the following form

$$\begin{aligned}
\begin{bmatrix} a \\ a \end{bmatrix} & \Longrightarrow \binom{b}{\lambda} \begin{bmatrix} a \\ a \end{bmatrix} \binom{b}{\lambda} \Longrightarrow \binom{b^2}{\lambda} \begin{bmatrix} a \\ a \end{bmatrix} \binom{b^2}{\lambda} \Longrightarrow \ldots \Longrightarrow \binom{b^n}{\lambda} \begin{bmatrix} a \\ a \end{bmatrix} \binom{b^n}{\lambda} \\
& \Longrightarrow \binom{b^{n-1}}{\lambda} \begin{bmatrix} ba \\ ba \end{bmatrix} \binom{b^n}{\lambda} \Longrightarrow \binom{b^{n-2}}{\lambda} \begin{bmatrix} b^2a \\ b^2a \end{bmatrix} \binom{b^n}{\lambda} \Longrightarrow \ldots \Longrightarrow \begin{bmatrix} b^na \\ b^na \end{bmatrix} \binom{b^n}{\lambda} \\
& \Longrightarrow \binom{c}{\lambda} \begin{bmatrix} b^na \\ b^na \end{bmatrix} \binom{b^n}{\lambda} \Longrightarrow \binom{c^2}{\lambda} \begin{bmatrix} b^na \\ b^na \end{bmatrix} \binom{b^n}{\lambda} \Longrightarrow \ldots \Longrightarrow \binom{c^n}{\lambda} \begin{bmatrix} b^na \\ b^na \end{bmatrix} \binom{b^n}{\lambda} \\
& \Longrightarrow \binom{c^{n-1}}{\lambda} \begin{bmatrix} cb^nab \\ cb^nab \end{bmatrix} \binom{b^{n-1}}{\lambda} \Longrightarrow \binom{c^{n-2}}{\lambda} \begin{bmatrix} c^2b^nab^2 \\ c^2b^nab^2 \end{bmatrix} \binom{b^{n-2}}{\lambda} \Longrightarrow \ldots \\
& \Longrightarrow \begin{bmatrix} c^nb^nab^n \\ c^nb^nab^n \end{bmatrix}
\end{aligned}$$

(first we add n times to the left as well as to the right a b in the upper strand, then we add n times to the left a b in the lower strand, then we add m times to the left a c in the upper strand, then we add m times simultaneously c to the left and b to the right in the lower strand; obviously, since we want to generate a double strand without overhangs $n = m$ has to hold). In a certain sense this derivation is the unique one for $\begin{bmatrix} c^nb^nab^n \\ c^nb^nab^n \end{bmatrix}$; the only change which is allowed concerns the order of the generation of the letter c in the upper strand and of the letter b in the lower part, which have not to be generated in the sequence given above, it can also happen in a mixed form, but we have to generate n times c and n times b; also the application of $(\binom{\lambda}{c}, \binom{\lambda}{b})$ can be done earlier, if c is already present in the upper overhang to the left and all bs added to the left have already their counterpart in the lower strand.

Therefore we get for the word language

$$wL(G) \cap \{c\}^*\{b\}^*\{a\}\{b\}^* = \{c^nb^nab^n \mid n \geq 1\}.$$

It is easy to show (e.g. by a pumping lemma) that $wL(G)$ is not a context-free language.

We now present four special types of sticker systems or requirements to the derivations in the systems.

Definition 10.3 *A sticker system $G = (V, \varrho, A, D)$ is called*

– one-sided *if, for each pair* $(u,v) \in D$, $u = \binom{\lambda}{\lambda}$ *or* $v = \binom{\lambda}{\lambda}$ *hold,*
– regular *if, for each pair* $(u,v) \in D$, $u = \binom{\lambda}{\lambda}$ *holds,*
– simple *if, for each pair* $(u,v) \in D$, $uv \in \binom{V^*}{\lambda}$ *or* $uv \in \binom{\lambda}{V^*}$ *hold.*

Obviously, the sticker system given in Example 11.34 is not one-sided and not regular since it contains the element $(\binom{\lambda}{c}, \binom{\lambda}{b})$ in its set D. On the other hand, G of Example 11.34 is simple since we use for the prolongations only pairs which prolong to both sides only the upper strand or only the lower strand of the molecule.

From the definition it can be seen that regular sticker systems have an analogy to regular grammars since the molecule and the string can only be prolonged to the right, respectively.

Definition 10.4 *i) For a sticker system* $G = (V, \varrho, A, D)$ *and a natural number* $d \geq 1$, *we define the language* $ML_d(G)$ *as the set of all molecules which have a derivation*

$$x = x_0 \Longrightarrow x_1 \Longrightarrow x_2 \Longrightarrow \ldots \Longrightarrow x_k \text{ with } x_k \in \begin{bmatrix} V \\ V \end{bmatrix}_\varrho^* \text{ with } d(x_i) \leq d \text{ for } 0 \leq i \leq k.$$

ii) We say that a molecule language $L \subset \begin{bmatrix} V \\ V \end{bmatrix}_\varrho^*$ *or a word language* $L' \subset V^*$ *can be generated with* bounded delay, *if there are a sticker system* $G = (V, \varrho, A, D)$ *and a natural number* $d \geq 1$ *such that* $ML(G) = ML_d(G)$ *and* $L = ML(G)$ *and* $L' = wL(G)$, *respectively, are valid.*

The words of the language $ML_d(G)$ can be generated by a derivation where the length of the overhangs is bounded by d. Thus it is obvious that $ML_d(G) \subseteq ML(G)$. If all words of $ML(G)$ can be generated by a derivation where the length of the overhangs is bounded, then $ML(G) = ML_d(G)$.

We mention that the generation of $\begin{bmatrix} c \\ c \end{bmatrix}^n \begin{bmatrix} b \\ b \end{bmatrix}^n \begin{bmatrix} a \\ a \end{bmatrix} \begin{bmatrix} b \\ b \end{bmatrix}^n$ and $c^n b^n a b^n$ by the sticker system given in of Example 11.34 requires an overhang of length n to the right. This follows from the fact that the shortening of the right overhang is only possible if the sub-molecule $\begin{bmatrix} b \\ b \end{bmatrix}^n$ between the c-part and the a-part has already been produced. We shall see below that the languages generated in Example 11.34 cannot be derived with bounded delay since the word language is not context-free (see Theorem 11.44).

We denote the families of word languages generated by arbitrary sticker systems, one-sided sticker systems and regular sticker systems by $\mathcal{L}(ASL)$, $\mathcal{L}(OSL)$ and $\mathcal{L}(RSL)$, respectively. If we allow only simple systems, we add the letter S before X with $X \in \{A, O, R\}$. Moreover, if we restrict to languages which can be generated by bounded delay, we add (b) after SL. Furthermore, we combine these restriction. Thus $\mathcal{L}(ASL(b))$ and $\mathcal{L}(SRSL)$ are the families of languages which can be generated by arbitrary sticker systems with bounded delay and by regular simple sticker systems, respectively.

We now investigate the generative power of sticker systems. The first two statements follow directly from the definitions.

Lemma 10.5 *For* $y \in \{(b), \lambda\}$, *the diagram given in Figure 11.15 is valid (if* X *and* Y *are connected by a line and* Y *has an upper position than* X, *then* $X \subseteq Y$*).* □

Lemma 10.6 *For* $X \in \{O, R, SO, SR\}$, $\mathcal{L}(XSL(b)) = \mathcal{L}(XSL)$.

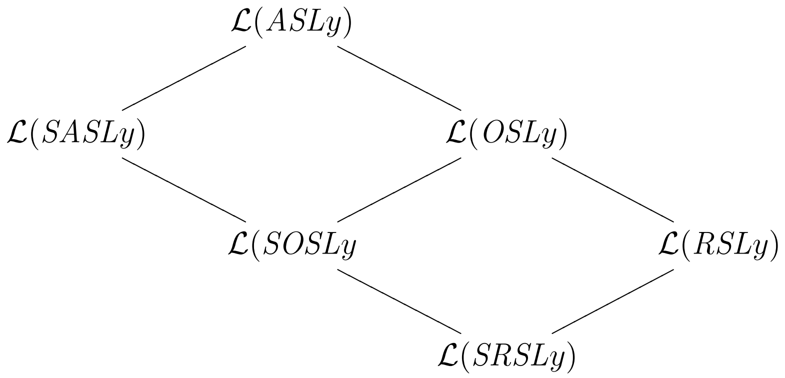

Figure 10.2: Hierarchy of (bounded) language families generated by sticker systems

Proof. Let $G = (V, \varrho, A, D)$ be a (simple) one-sided or (simple) regular sticker system. Let

$$d = \max\{d(x) \mid x \in A \text{ or } (x,u) \in D \text{ or } (u,x) \in D \text{ for some } u\}.$$

Now assume that there is a derivation of some molecule z with an upper overhang at the right end which is longer than d. This situation can only occur if in the last step some upper single strand has been added. Then the elements of D which result in a prolongation to the right have to be simple. If there are upper strands the delay is increased; if it is a lower strand, then the delay is decreased. Obviously, the order in which we apply the single strands can be arbitrarily chosen. Finally we have to reach a molecule without overhangs. Therefore we can choose the order of the simple addings in such a way that the overhang is always smaller than d. Thus any molecule can be generated by a derivation where all intermediate steps have a delay $\leq d$. Hence $L(G) = L_d(G)$.

Therefore any language generated by a (simple) one-sided or (simple) regular sticker system, is a language with bounded delay. This implies the statements. $\square$

Lemma 10.7 $\mathcal{L}(ASL) \subseteq \mathcal{L}(CS)$.

Proof. Let $G = (V, \varrho, A, D)$ be a sticker system. We consider the alphabet consisting of all pairs (a, c) with $a, c \in V \cup \{\lambda\}$ and $(a, c) \in \varrho$ if a and c are both non-empty words. It is easy to construct a phrase structure grammar which simulates the sticking of x to the left and the sticking of y to the right where $(x, y) \in D$. Since no erasing is performed during the simulations, the grammar is a context-sensitive one. $\square$

Lemma 10.8 $\mathcal{L}(OSL \subseteq \mathcal{L}(REG)$.

Proof. Let $G = (V, \varrho, A, D)$ be a one-sided sticker system. By the proof of Lemma 11.38, $L(G) = L_d(G)$ for some d. We now construct the context-free grammar $G' = (N, T, P, S)$ with

$$\begin{aligned} N &= \left\{ \left\langle \begin{matrix} u \\ \lambda \end{matrix} \right\rangle_l, \left\langle \begin{matrix} u \\ \lambda \end{matrix} \right\rangle_r, \left\langle \begin{matrix} \lambda \\ u \end{matrix} \right\rangle_l, \left\langle \begin{matrix} \lambda \\ u \end{matrix} \right\rangle_r \mid u \in V^*, 0 \leq |u| \leq d \right\} \cup \{S\}, \\ T &= \begin{bmatrix} V \\ V \end{bmatrix}_\varrho \end{aligned}$$

(the nonterminals store the existing overhangs to the left or to the right),and P consisting of all rules of the form

$$S \to \left\langle \begin{matrix} u_1 \\ u_2 \end{matrix} \right\rangle_l \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \left\langle \begin{matrix} v_1 \\ v_2 \end{matrix} \right\rangle_r \text{ with } \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \in A$$

(by these rules we generate all elements of A),

$$\left\langle \begin{matrix} u_1 \\ u_2 \end{matrix} \right\rangle_l \to \left\langle \begin{matrix} u_1' \\ u_2' \end{matrix} \right\rangle_l \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \text{ such that } \begin{bmatrix} x_1 y_1 u_1 \\ x_2 y_2 u_2 \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$
$$\text{for some } \left(\begin{pmatrix} u_1' \\ u_2' \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \begin{pmatrix} \lambda \\ \lambda \end{pmatrix} \right) \in D$$

(if the left end is $\binom{u_1}{u_2}$ we add to the left the sticker $\binom{u_1'}{u_2'} \left[\begin{smallmatrix} y_1 \\ y_2 \end{smallmatrix}\right] \binom{y_1}{y_2}$ according to an element of D to the left end and get $\binom{u_1'}{u_2'} \left[\begin{smallmatrix} x_1 y_1 u_1 \\ x_2 y_2 u_2 \end{smallmatrix}\right]$),

$$\left\langle \begin{matrix} u_1 \\ u_2 \end{matrix} \right\rangle_r \to \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \left\langle \begin{matrix} u_1' \\ u_2' \end{matrix} \right\rangle_r \text{ such that } \begin{bmatrix} u_1 y_1 x_1 \\ u_2 y_2 x_2 \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$
$$\text{for some } \left(\begin{pmatrix} \lambda \\ \lambda \end{pmatrix}, \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \begin{pmatrix} u_1' \\ u_2' \end{pmatrix} \right) \in D$$

(we extend to the right),

$$\left\langle \begin{matrix} \lambda \\ \lambda \end{matrix} \right\rangle_l \to \lambda \text{ and } \left\langle \begin{matrix} \lambda \\ \lambda \end{matrix} \right\rangle_r \to \lambda$$

(if there is no overhang, then we finish the derivation).

It is easy to see that $L(G') = ML(G) = ML_d(G)$.

Moreover, any derivation starts with a rule $S \to \left\langle \begin{smallmatrix} u_1 \\ u_2 \end{smallmatrix} \right\rangle_l \left[\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right] \left\langle \begin{smallmatrix} v_1 \\ v_2 \end{smallmatrix} \right\rangle_r$ with $\binom{u_1}{u_2} \left[\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right] \binom{v_1}{v_2} \in A$. Then we extend from $\left\langle \begin{smallmatrix} u_1 \\ u_2 \end{smallmatrix} \right\rangle_l$ to the left by rules of the form $A \to Bz$ and from $\left\langle \begin{smallmatrix} v_1 \\ v_2 \end{smallmatrix} \right\rangle_r$ to the right by rules of the form $A \to zB$ where $w \in T^*$ and $A, B \in N$. Therefore $L(G')$ is a finite union of languages of the form $X\{w\}Y$ where X and Y are generated by rules of the form $A \to zB$ or $A \to Bz$ and $w \in T^+$. Hence X and Y are regular, which implies that all $X\{w\}Y$ and thus $L(G')$ are regular, too.

In order to get the word language, we only consider the upper strands, which can be obtained by a homomorphism from $L(G')$. By the closure properties of $\mathcal{L}(REG)$, $wL(G')$ is regular, too. □

Lemma 10.9 $\mathcal{L}(REG) \subseteq \mathcal{L}(RSL(b))$.

Proof. Let L be a regular language. Then $L = L(\mathcal{A})$ for some deterministic finite automaton $\mathcal{A} = (X, Z, z_1, F, \delta)$. Let $Z = \{z_1, z_2, \ldots, z_k\}$.

We construct a sticker system $G = (X, \varrho, A, D)$ with $\varrho = \{(a, a) \mid a \in X\}$ (because we are only interested in the word language it is sufficient to consider only molecules of the form $\left[\begin{smallmatrix} w \\ w \end{smallmatrix}\right]$). With any state z_j we associate the words $\left[\begin{smallmatrix} w \\ w \end{smallmatrix}\right] \binom{u}{\lambda}$ with $|wu| = k + 1$ and $|u| = j$. Or conversely, if w is a word of length $k+1$, and we want to remember a state z_j, then we choose x and u as the prefix and suffix of w of lengths $k+1-j$ and j, respectively. A word $z \in L$ can be written as $w = w_1 w_2 \dots w_r$ where $|w_i| = k+1$ for $1 \le i \le r-1$ and $1 \le |w_r| \le k+1$. We consider the states $s_i = \delta(z_0, w_1 w_2 \dots w_i) = \delta(s_{i-1}, w_i)$. By a partition of w_i into x_i and u_i as mentioned above we can remember the state s_i.

We now define A and D by

$$\begin{aligned}
A_1 &= \left\{ \begin{bmatrix} x \\ x \end{bmatrix} \mid x \in L,\ 0 \le |x| \le k+1 \right\}, \\
A_2 &= \left\{ \begin{bmatrix} x \\ x \end{bmatrix} \begin{pmatrix} u \\ \lambda \end{pmatrix} \mid |xu| = k+1, |u| = j, \delta(z_0, xu) = z_j \right\}, \\
A &= A_1 \cup A_2
\end{aligned}$$

(any word $x \in L$ of length at most $k+1$ is in $L(G)$ by $A_1 \subseteq L(G)$; otherwise we consider the prefix of the word, i.e., w_1 in the above notation and remember $z_j = s_1$),

$$\begin{aligned}
D_1 &= \left\{ \left(\begin{pmatrix} \lambda \\ \lambda \end{pmatrix}, \begin{pmatrix} \lambda \\ v \end{pmatrix} \begin{bmatrix} x \\ x \end{bmatrix} \begin{pmatrix} u \\ \lambda \end{pmatrix} \right) \mid |v| = j, |xu| = k+1, |u| = i, \delta(z_j, xu) = z_i \right\}, \\
D_2 &= \left\{ \left(\begin{pmatrix} \lambda \\ \lambda \end{pmatrix}, \begin{pmatrix} \lambda \\ v \end{pmatrix} \begin{bmatrix} x \\ x \end{bmatrix} \right) \mid |v| = j, 1 \le |x| \le k+1, \delta(z_j, x) \in F \right\}, \\
D &= D_1 \cup D_2
\end{aligned}$$

(by the rules of D_1, we extend the word by xu, which leads from the remembered state z_j to the state z_i which is stored by u of length i; by the rules of D_2, we read the last subword of the partition we add the word without an overhang and stop the generation if an accepting state is reached; otherwise we have no applicable rule).

It is easy to see by theses explanations that $L = wL(G)$.

Obviously, G is a regular sticker system. Moreover, by our construction, all overhangs are bounded by k. Therefore, $ML(G) = ML_k(G)$ which shows that $wL(G)$ is of bounded delay. $\square$

Lemma 10.10 $ASL(b) = \mathcal{L}(LIN)$.

Proof. We first prove that $\mathcal{L}(ASL(b)) \subseteq \mathcal{L}(LIN)$ holds.

In a sticker system, a word is generated by adding to the left and to the right elements of $W_\varrho(V)$, i.e., looking only on the upper strand a derivation has the form

$$z \Longrightarrow p_1 z q_1 \Longrightarrow p_2 p_1 z q_1 q_2 \Longrightarrow \dots \Longrightarrow p_n p_{n-1} \dots p_2 p_1 z q_1 q_2 \dots q_{n-1} q_n. \tag{10.2}$$

In a linear grammar, the situation is opposite since a derivation has the form

$$\begin{aligned}
S &\Longrightarrow p'_1 A_1 q'_1 \Longrightarrow p'_1 p'_2 A_2 q'_2 q'_1 \Longrightarrow \dots \Longrightarrow p'_1 p'_2 \dots p'_n A_n q'_n q'_{n-1} \dots q'_1 \\
&\Longrightarrow p'_1 p'_2 \dots p'_n A z' q'_n q'_{n-1} \dots q'_1.
\end{aligned}$$

Thus the idea of the linear grammar is to start with the elements added in the last step of the generation in the sticker system, to move "backwards" in the generation and to stop with a generation of an element of the start set of the sticker system.

We now give the formalization of this idea. Let $G = (V, \varrho, A, D)$ be a sticker system. Let $L = ML_d(G)$ for some constant d. Then the delays are bounded by d. We construct the linear grammar $G' = (N, T, P, S)$ with

$$\begin{aligned} N &= \left\{ \left\langle \binom{u_1}{u_2}, \binom{v_1}{v_2} \right\rangle \mid \binom{u_1}{u_2}, \binom{v_1}{v_2} \in \binom{\lambda}{V} \cup \binom{V}{\lambda},\ |u_1|, |u_2|, |v_1|, |v_2| \le d \right\} \cup \{S\}, \\ T &= \begin{bmatrix} V \\ V \end{bmatrix}_{\varrho} \end{aligned}$$

(by the nonterminals we store the overhangs by going from the outer part to the inner part) and P consisting of all rules of the following forms

$$S \to \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \left\langle \binom{u_1}{u_2}, \binom{v_1}{v_2} \right\rangle \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \text{ where } \left(\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \binom{u_1}{u_2}, \binom{v_1}{v_2} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \right) \in D$$

(we generate the outer elements p_n and q_n of the derivation (11.7),

$$\begin{aligned} \left\langle \binom{u_1}{u_2}, \binom{v_1}{v_2} \right\rangle &\to \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \left\langle \binom{u_1'}{u_2'}, \binom{v_1'}{v_2'} \right\rangle \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \\ &\text{where} \left(\binom{x_1}{x_2} \begin{bmatrix} x_1' \\ x_2' \end{bmatrix} \binom{u_1'}{u_2'}, \binom{v_1'}{v_2'} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \binom{y_1'}{y_2'} \right) \in D, \\ &\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} u_1 x_1 x_1' \\ u_2 x_2 x_2' \end{bmatrix}, \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} y_1 y_1' v_1 \\ y_2 y_2' v_2 \end{bmatrix}, \end{aligned}$$

(we proceed to the "middle"),

$$\begin{aligned} &\left\langle \binom{u_1}{u_2}, \binom{v_1}{v_2} \right\rangle \to \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \\ &\text{where } \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \neq \begin{bmatrix} \lambda \\ \lambda \end{bmatrix}, \ \binom{w_1'}{w_2'} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \binom{z_1'}{z_2'} \in A, \ \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} u_1 w_1' \\ u_2 w_2' \end{bmatrix}, \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} z_1' v_1 \\ z_2' v_2 \end{bmatrix}, \end{aligned}$$

(if the overhangs of the nonterminal fit to some element of A, we finish the derivation),

$$S \to \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \text{ where } \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \in A$$

(we generate directly the elements from A which belong to the language $ML(G)$). By these explanations it is easy to see that $L(G') = wL_d(G)$. □

Lemma 10.11 *There is a regular language which is not in SOSL.*

Proof. We consider the language $L = \{b\}\{a\}^+\{b\}$, which is regular since it is given as a regular expression. Let us assume that $L = wL(G)$ for some simple one-sided sticker system $G = (V, \varrho, A, D)$. Because A is finite, and L is infinite, one needs upper and lower strands which can generate in the upper part an arbitrary number of as and the corresponding letter in the lower part, i.e., D contains at least one pair of one of the forms

$$\left(\binom{\lambda}{\lambda}, \binom{y_1}{\lambda} \right) \text{ and } \left(\binom{\lambda}{\lambda}, \binom{\lambda}{y_2} \right) \quad \text{or} \quad \left(\binom{y_1}{\lambda}, \binom{\lambda}{\lambda} \right) \text{ and } \left(\binom{\lambda}{y_2}, \binom{\lambda}{\lambda} \right)$$

with $y_1 \in \{a\}^+$ and $y_2 \in (V')^+$, where V' consists of all letters c with $(a, c) \in \varrho$.

We only discuss the first case; the other one can be handled analogously.

Let $|y_1| = k_1$ and $|y_2| = k_2$. Then, for ba^2b, we have a derivation

$$\binom{x_1}{x_2}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}\binom{z_1}{z_2} \Longrightarrow^* \begin{bmatrix} ba^2b \\ w_1 \end{bmatrix}$$

for some $\binom{x_1}{x_2}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}\binom{z_1}{z_2} \in A$ and some $w_1 \in (V'')^+$ where V'' consists of all letters c with $(a, c) \in varrho$ or $(b, c) \in varrho$. However, this molecule can be extended to the right by adding k_2 times $\binom{y_1}{\lambda}$ and k_1 times $\binom{\lambda}{y_2}$. Because we have $k_2|y_1| = k_1|y_2| = k_1k_2$, we get the result

$$\begin{bmatrix} ba^2b \\ w_1 \end{bmatrix}\binom{y_1}{\lambda}^{k_2}\binom{\lambda}{y_2}^{k_1} = \begin{bmatrix} ba^2ba^{k_1k_2} \\ w_2 \end{bmatrix}$$

for some $w_2 \in (V')^+$ of length $k_1k_2 + 4$. Hence $ba^2ba^{k_1k_2} \in wL(G)$, but $ba^2ba^{k_1k_2} \notin L$ in contrast to $L = wL(G)$. □

If we combine the Lemmas 11.37 – 11.43 and the fact that $\mathcal{L}(SASL)$ contains a non-context-free language by Example 11.34, we obtain the following hierarchy.

Theorem 10.12 *The diagram of Figure 11.16 holds (where an arrow $X \to Y$ is used for the proper inclusion $X \subset Y$; if X and Y are connected by a line and Y has an upper position than X, then $X \subseteq Y$).* □

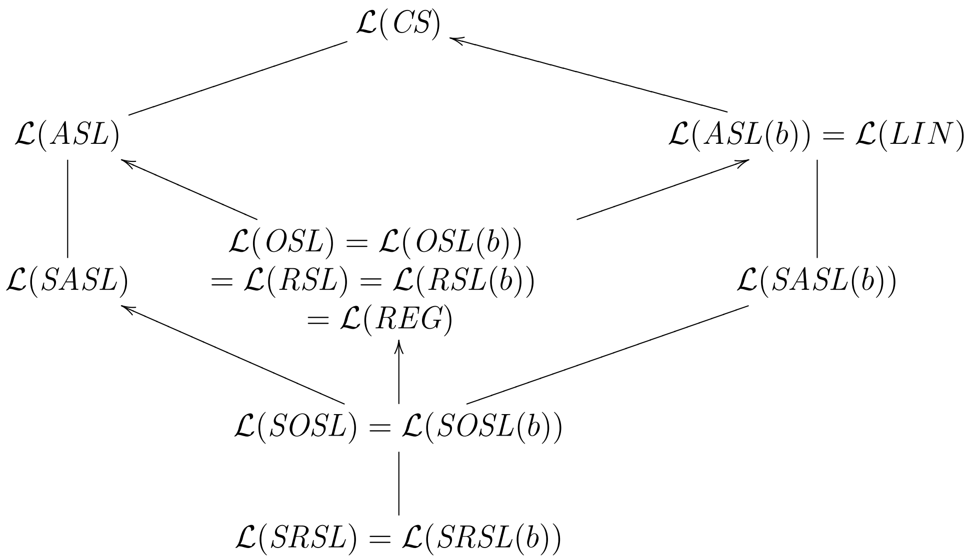

Figure 10.3: Hierarchy of language families generated by sticker systems