BIDIRECTIONAL STICKER SYSTEMS

Rudolf FREUND
Department of Computer Science, Vienna University of Technology
Karlsplatz 13, A-1040 Wien, Austria
email: rudi@logic.at

Gheorghe PĂUN
Institute of Mathematics of the Romanian Academy
PO Box 1 – 764, 70700 Bucureşti, Romania
email: gpaun@imar.ro

Grzegorz ROZENBERG
Department of Computer Science, Leiden University
PO Box 9512, 2300 RA Leiden, The Netherlands
email: rozenber@wi.leidenuniv.nl

Arto SALOMAA
Academy of Finland and Turku University
Department of Mathematics, 20500 Turku, Finland
email: asalomaa@sara.utu.fi

We introduce two-sided *sticker systems*, the two-sided variant of a computability model introduced[5] as an abstraction of Adleman's style of DNA computing[1] and of the matching of the so-called Watson-Crick complements. Several types of sticker systems are shown to have the same power as regular grammars, one variant is found to represent the linear languages, and another one is proved to be able to represent any recursively enumerable language. From this result we infer that any recursively enumerable language can be represented as the projection of the intersection of two minimal linear languages.

1 Introduction

Sticker systems[5] are language generating devices based on the *sticker operation*, which, in turn, is a model of the techniques used by L. Adleman in his successful experiment of deciding the existence of a Hamiltonian path in a graph by using DNA[1]. We recall some biological details in order to see the roots of the models of sticker systems[5] and of bidirectional sticker systems introduced in this paper. DNA sequences are double stranded (helicoidal) structures composed of four nucleotides, A (adenine), C (cytosine), G (guanine), and T (thymine), paired A–T, C–G according to the so-called Watson-Crick complements. If we have a single stranded sequence of A, C, G, T nucleotides, together with a single stranded sequence composed of the complementary nucleotides, the two sequences will be "glued" together (by hydrogen bonds), forming a double stranded DNA sequence. This matching of complementary nucleotides now is the constraint which has to be fulfilled when we prolong to the left and to the right a sequence of (single or double) symbols by using given single stranded strings or even more complex dominoes with sticky ends, gluing these ends together with the sticky ends of the current sequence according to a complementarity relation.

A formal model for this *sticker operation* is defined in the following section. According to this sticker operation a generative mechanism can be defined: We start from a given set of (incomplete) double-stranded sequences (axioms), plus a set of pairs of double-stranded complementary sequences allowing us to prolong the initial sequences to the left and to the right, respectively. Iterating these prolongations we get "computations" of possibly arbitrary length. A *codification* procedure (a projection) finally gives "meaning" to blocks of symbols, in this way yielding the "decoded" language.

The generative power of several variants of such mechanisms is investigated in the following. Several types of bidirectional sticker systems are shown to represent regular languages and one variant is found to represent the linear languages. Another variant is proved to be able to represent any recursively enumerable language, which reminds the results obtained in a series of papers [2,4,6,7,8] about the possibility of designing universal (and programmable) DNA "computers" based on the operation of *splicing*[3], introduced as a model of the recombinant behavior of DNA under the influence of restriction enzymes and ligases. Moreover, this universality result also allows us to establish an optimal representation result for recursively enumerable languages: Any recursively enumerable language L, $L \subseteq \Sigma^*$, can be written as $L = h\left(L_1 \cap L_2\right)$, where L_1 and L_2, $L_1, L_2 \subseteq \Sigma_1^*$, are minimal linear languages (i.e. they are generated by a linear grammar with only one non-terminal symbol) and h is a projection from Σ_1 on Σ.

2 Prerequisites

We first specify a few notions and notations from formal language theory [9]. For an alphabet V, by V^* we denote the free monoid generated by V under the operation of concatenation; the empty string is denoted by λ. Moreover, we denote $V^+ = V^* \setminus \{\lambda\}$; $|x|$ is the length of the string x. By *REG*, *LIN*, and *RE* we denote the families of regular, linear, and recursively enumerable languages, respectively.

Take an alphabet V (a finite set of abstract symbols) endowed with a relation ρ (of *complementarity*), $\rho \subseteq V \times V$; moreover, consider a special symbol, #, not in V, denoting an empty space (the *blank* symbol). Using the elements of $V \cup \{\#\}$ we construct the following sets of *composite* symbols:

$$\binom{V}{V}_\rho = \left\{ \binom{a}{b} \mid a, b \in V, (a,b) \in \rho \right\},$$

$$\binom{\#}{V} = \left\{ \binom{\#}{a} \mid a \in V \right\}, \quad \binom{V}{\#} = \left\{ \binom{a}{\#} \mid a \in V \right\}.$$

Moreover, we denote $W_\rho(V) = W_\rho^s(V) \cup S(V)$ (the set of *well-formed sequences* or *dominoes*) and $W_\rho^s(V) = S(V)\binom{V}{V}_\rho^+ S(V)$ (the set of *well-started sequences* or *dominoes*), where $S(V) = \binom{\#}{V}^* \cup \binom{V}{\#}^*$. Stated otherwise, the elements of $W_\rho^s(V)$ in the middle have pairs of symbols in V, as selected by the complementarity relation, and at the left and at the right they end either by pairs $\binom{\#}{a}$ or by pairs $\binom{b}{\#}$, for $a, b \in V$ (the symbols $\binom{\#}{a}$, $\binom{b}{\#}$ are not mixed). The elements of $S(V)$ at the left and at the right end of a domino in $W_\rho^s(V)$ are called *sticky ends*; if such an element is λ, then this end is called a *blunt end.*

The *sticker operation,* denoted by μ, is a partially defined mapping from $W_\rho(V) \times W_\rho(V)$ to $W_\rho(V)$; for $x, y, z \in W_\rho(V)$, we write $\mu(x,y) = z$ if and only if one of the following cases holds:

1. $x = w\binom{a_1}{\#}\cdots\binom{a_r}{\#}\binom{a_{r+1}}{\#}\cdots\binom{a_{r+p}}{\#}$, $w \in S(V)\binom{V}{V}_\rho^+$,

   $y = \binom{\#}{c_1}\cdots\binom{\#}{c_r}$, $z = w\binom{a_1}{c_1}\cdots\binom{a_r}{c_r}\binom{a_{r+1}}{\#}\cdots\binom{a_{r+p}}{\#}$, or

   $y = \binom{a_{r+p}}{\#}\cdots\binom{a_{r+1}}{\#}\binom{a_r}{\#}\cdots\binom{a_1}{\#}w$, $w \in \binom{V}{V}_\rho^+ S(V)$,

   $x = \binom{\#}{c_r}\cdots\binom{\#}{c_1}$, $z = \binom{a_{r+p}}{\#}\cdots\binom{a_{r+1}}{\#}\binom{a_r}{c_r}\cdots\binom{a_1}{c_1}w$, for $r \geq 0$,

   $p \geq 0$, $a_j \in V, 1 \leq j \leq r+p$, $c_i \in V, 1 \leq i \leq r$, $(a_i, c_i) \in \rho, 1 \leq i \leq r$;

2. $x = w\binom{\#}{c_1}\cdots\binom{\#}{c_r}\binom{\#}{c_{r+1}}\cdots\binom{\#}{c_{r+p}}$, $w \in S(V)\binom{V}{V}_\rho^+$,

   $y = \binom{a_1}{\#}\cdots\binom{a_r}{\#}$, $z = w\binom{a_1}{c_1}\cdots\binom{a_r}{c_r}\binom{\#}{c_{r+1}}\cdots\binom{\#}{c_{r+p}}$, or

   $y = \binom{\#}{c_{r+p}}\cdots\binom{\#}{c_{r+1}}\binom{\#}{c_r}\cdots\binom{\#}{c_1}w$, $w \in \binom{V}{V}_\rho^+ S(V)$,

   $x = \binom{a_r}{\#}\cdots\binom{a_1}{\#}$, $z = \binom{\#}{c_{r+p}}\cdots\binom{\#}{c_{r+1}}\binom{a_r}{c_r}\cdots\binom{a_1}{c_1}w$, for $r \geq 0$,

   $p \geq 0, a_i \in V, 1 \leq i \leq r$, $c_j \in V, 1 \leq j \leq r+p$, $(a_i, c_i) \in \rho, 1 \leq i \leq r$;

3. $x = w\binom{a_1}{\#}\cdots\binom{a_r}{\#}$, $w \in S(V)\binom{V}{V}_\rho^+$, $y = \binom{\#}{c_1}\cdots\binom{\#}{c_r}\cdots\binom{\#}{c_{r+p}}$,

$z = w \binom{a_1}{c_1} \ldots \binom{a_r}{c_r} \binom{\#}{c_{r+1}} \ldots \binom{\#}{c_{r+p}}$, or

$y = \binom{a_r}{\#} \ldots \binom{a_1}{\#} w$, $w \in \binom{V}{V}^+_\rho S(V)$, $x = \binom{\#}{c_{r+p}} \ldots \binom{\#}{c_r} \ldots \binom{\#}{c_1}$,

$z = \binom{\#}{c_{r+p}} \ldots \binom{\#}{c_{r+1}} \binom{a_r}{c_r} \ldots \binom{a_1}{c_1} w$, for $r \geq 0, p \geq 0,\ r+p \geq 1$,

$a_i \in V, 1 \leq i \leq r,\ c_j \in V, 1 \leq j \leq r+p,\ (a_i, c_i) \in \rho, 1 \leq i \leq r$;

4. $x = w \binom{\#}{c_1} \ldots \binom{\#}{c_r}$, $w \in S(V) \binom{V}{V}^+_\rho$, $y = \binom{a_1}{\#} \ldots \binom{a_r}{\#} \ldots \binom{a_{r+p}}{\#}$,

$z = w \binom{a_1}{c_1} \ldots \binom{a_r}{c_r} \binom{a_{r+1}}{\#} \ldots \binom{a_{r+p}}{\#}$, for

$y = \binom{\#}{c_r} \ldots \binom{\#}{c_1} w$, $w \in \binom{V}{V}^+_\rho S(V)$, $x = \binom{a_{r+p}}{\#} \ldots \binom{a_r}{\#} \ldots \binom{a_1}{\#}$,

$z = \binom{a_{r+p}}{\#} \ldots \binom{a_{r+1}}{\#} \binom{a_r}{c_r} \ldots \binom{a_1}{c_1} w$, for $r \geq 0, p \geq 0,\ r+p \geq 1$,

$a_j \in V, 1 \leq j \leq r+p,\ c_i \in V, 1 \leq i \leq r,\ (a_i, c_i) \in \rho, 1 \leq i \leq r$;

5. $x = w \binom{a_1}{\#} \ldots \binom{a_r}{\#}$, $w \in S(V) \binom{V}{V}^+_\rho$, $y = \binom{a_{r+1}}{\#} \ldots \binom{a_{r+p}}{\#}$,

$z = w \binom{a_1}{\#} \ldots \binom{a_r}{\#} \binom{a_{r+1}}{\#} \ldots \binom{a_{r+p}}{\#}$, or

$y = \binom{a_r}{\#} \ldots \binom{a_1}{\#} w$, $w \in \binom{V}{V}^+_\rho S(V)$, $x = \binom{a_{r+p}}{\#} \ldots \binom{a_{r+1}}{\#}$,

$z = \binom{a_{r+p}}{\#} \ldots \binom{a_{r+1}}{\#} \binom{a_r}{\#} \ldots \binom{a_1}{\#} w$, for $r \geq 0, p \geq 0,\ r+p \geq 1$,

$a_i \in V, 1 \leq i \leq r+p$;

6. $x = w \binom{\#}{c_1} \ldots \binom{\#}{c_r}$, $w \in S(V) \binom{V}{V}^+_\rho$, $y = \binom{\#}{c_{r+1}} \ldots \binom{\#}{c_{r+p}}$,

$z = w \binom{\#}{c_1} \ldots \binom{\#}{c_r} \binom{\#}{c_{r+1}} \ldots \binom{\#}{c_{r+p}}$, or

$y = \binom{\#}{c_r} \ldots \binom{\#}{c_1} w$, $w \in \binom{V}{V}^+_\rho S(V)$, $x = \binom{\#}{c_{r+p}} \ldots \binom{\#}{c_{r+1}}$,

$z = \binom{\#}{c_{r+p}} \ldots \binom{\#}{c_{r+1}} \binom{\#}{c_r} \ldots \binom{\#}{c_1} w$, for $r \geq 0, p \geq 0,\ r+p \geq 1$,

$c_i \in V, 1 \leq i \leq r+p$;

7. $x = w \binom{a_1}{\#} \ldots \binom{a_r}{\#}$, $w \in S(V) \binom{V}{V}^*_\rho$, $y = \binom{\#}{c_1} \ldots \binom{\#}{c_r} v$,

$v \in \binom{V}{V}^*_\rho S(V)$, $wv \in W^s_\rho(V)$, $z = w \binom{a_1}{c_1} \ldots \binom{a_r}{c_r} v$, or

$x = w \binom{\#}{c_1} \ldots \binom{\#}{c_r}$, $w \in S(V) \binom{V}{V}^*_\rho$, $y = \binom{a_1}{\#} \ldots \binom{a_r}{\#} v$,

$v \in \binom{V}{V}_\rho^* S(V)$, $wv \in W_\rho^s(V)$, $z = w\binom{a_1}{c_1}\dots\binom{a_r}{c_r}v$, for $r \geq 0$,

$a_i \in V, 1 \leq i \leq r,\ c_i \in V, 1 \leq i \leq r,\ (a_i, c_i) \in \rho, 1 \leq i \leq r.$

In case 1 (case 2) we add complementary symbols on the lower (upper) level of x without completing all the blank spaces. In case 3 and case 4 we not only complete the blank spaces, but may even prolong the strand. In case 5 and case 6 the sticky end of x itself is prolonged. In case 7 we combine two dominoes whose sticky ends exactly fit together. Of course, for dominoes x, y which do not satisfy any of the previous conditions, $\mu(x, y)$ is not defined. Note that in all cases we allow the prolongation of "blunt" ends of strings in $W_\rho^s(V)$; moreover we have $\mu(\lambda, x) = \mu(x, \lambda) = x$ for every $x \in W_\rho^s(V)$; finally, observe that if all (intermediate) results are defined, then for dominoes $x, y, z \in W_\rho(V)$ we have $\mu(\mu(x, y), z) = \mu(x, \mu(y, z))$, i.e. the sticker operation is associative.

3 Bidirectional sticker systems

Using the sticker operation we can define a generating mechanism as follows:

A *bidirectional sticker system* is a construct $\gamma = ((V, \rho), A, P)$, where V is an alphabet, $\rho \subseteq V \times V$ is a relation on V, A is a finite subset of $W_\rho^s(V)$ (of axioms), and P is a finite set of pairs (D_l, D_r), where D_l is the domino to be adjoined at the left-hand side of the current object and D_r is the domino to be adjoined at the right-hand side of the current object, repectively; $D_l, D_r \in W_\rho(V)$.

The idea behind considering such a machinery is the following: We start with the objects in A and we prolong them to the left and to the right according to the sticker operation by using one of the pairs (D_l, D_r). When no sticky end is present any more, we may stop the derivation.

Formally, for two objects $x, z \in W_\rho(V)$ we write

$$x \Longrightarrow_\gamma z \text{ if and only if } z = \mu(\mu(D_l, x), D_r) \text{ for some } (D_l, D_r) \in P.$$

By $\Longrightarrow_\gamma^*$ we denote the reflexive and transitive closure of the relation $\Longrightarrow_\gamma$.

A sequence $x_0 \Longrightarrow_\gamma x_1 \Longrightarrow_\gamma \dots \Longrightarrow_\gamma x_k$, $x_0 \in A$, is called a *computation* in γ (of length k). A computation as above is *complete* when $x_k \in \binom{V}{V}_\rho^+$. The language generated by γ, denoted by $L(\gamma)$, is defined by

$$L(\gamma) = \left\{ w \mid w \in \binom{V}{V}_\rho^+,\ x \Longrightarrow_\gamma^* w \text{ for some } x \in A \right\}.$$

Therefore, only the complete computations are taken into account when defining $L(\gamma)$. Note that a complete computation can be continued, because we allow prolongations starting from blunt ends.

A complete computation $x_0 \Longrightarrow_\gamma x_1 \Longrightarrow_\gamma \ldots \Longrightarrow_\gamma x_k$, $x_0 \in A$, $x_k \in \binom{V}{V}_\rho^+$, with respect to γ, is said to be of *bounded delay* d, if the sticky ends at the right-hand side and at the left-hand side of each x_i, $0 \leq i \leq k$, are never longer than d. The corresponding subset of $L(\gamma)$ is denoted by $L_d(\gamma)$.

A bidirectional sticker system is called *regular,* if for every pair (D_l, D_r) in P either $D_l = \lambda$ or $D_r = \lambda$, i.e. in each step we can prolong the objects to the left only or to the right only.

By BSL, $RBSL$ we denote the families of languages generated by arbitrary and regular bidirectional sticker systems; the corresponding families of languages generated by arbitrary and regular bidirectional sticker systems with computations of bounded delay are denoted by $DBSL$, $DRBSL$.

4 Characterizing the regular and the linear languages

We now begin our investigations concerning the generative capacity of bidirectional sticker systems. We will first show that the basic variants characterize the regular languages.

Lemma 1. *Every regular language is the projection of a language in DRBSL.*

Proof. Let $G = (N, T, P, S)$ be a regular grammar, i.e. we may assume that each production in P is of the form $B \to bC$ or $B \to b$ with $B, C \in N$ and $b \in T$. We now define a regular bidirectional sticker system with $L(G) = g_T(L(\gamma))$ by $\gamma = ((V, \rho), A, P)$, where $V = N \cup T$, $\rho = \{(x, x) \mid x \in N \cup T\}$, $A = \left\{ \begin{smallmatrix} S & \# \\ S & S \end{smallmatrix} \right\}$, and P contains the following pairs of dominoes:

$\left(\lambda, \begin{smallmatrix} B & B & b \\ \# & \# & \# \end{smallmatrix}\right)$, $\left(\lambda, \begin{smallmatrix} \# & \# \\ B & b \end{smallmatrix}\right)$ for $B \to b$ in P with $B \in N$ and $b \in T$.

$\left(\lambda, \begin{smallmatrix} B & B & b \\ \# & \# & \# \end{smallmatrix}\right)$, $\left(\lambda, \begin{smallmatrix} \# & \# & \# \\ B & b & C \end{smallmatrix}\right)$ for $B \to bC$ in P with $B, C \in N$ and $b \in T$.

Moreover, we define $g_T : \binom{V}{V}_\rho^* \to T^*$ by $g_T\left(\binom{x}{x}\right) = \begin{cases} x, & \text{if } x \in T, \\ \lambda, & \text{if } x \in N. \end{cases}$

An object generated by this regular bidirectional sticker system γ represents the derivation for the corresponding string generated by the regular grammar G, i.e. if $A_1 \Longrightarrow_G a_1 A_2 \Longrightarrow_G \ldots \Longrightarrow_G a_1 \ldots a_{n-1} A_n \Longrightarrow_G a_1 \ldots a_{n-1} a_n$ is a derivation generating the string $a_1 \ldots a_n$, where $A_1, \ldots, A_n \in N$ and $A_1 = S$, then a corresponding derivation in γ is:

$$\begin{smallmatrix} A_1 & \# \\ A_1 & A_1 \end{smallmatrix} \Longrightarrow_\gamma \begin{smallmatrix} A_1 & A_1 & A_1 & a_1 \\ A_1 & A_1 & \# & \# \end{smallmatrix} \Longrightarrow_\gamma \begin{smallmatrix} A_1 & A_1 & A_1 & a_1 & \# \\ A_1 & A_1 & A_1 & a_1 & A_2 \end{smallmatrix} \Longrightarrow_\gamma \cdots$$

$$\begin{smallmatrix} A_1 & A_1 & A_1 & a_1 & A_2 & A_2 & & a_{n-1} & A_n & A_n & a_n \\ A_1 & A_1 & A_1 & a_1 & A_2 & A_2 & \cdots & a_{n-1} & A_n & A_n & a_n \end{smallmatrix}$$

By applying g_T to the resulting object of this derivation in γ, we obviously obtain $a_1 \ldots a_n$. Hence, we conclude $L(G) = g_T(L(\gamma))$. $\square$

The proof of the opposite direction is based on the fact that in regular bidirectional sticker systems the adjoinings of dominoes to the right and to the left are done in an independent way:

Lemma 2. *$RBSL \subseteq REG$.*

Proof. In a regular bidirectional sticker system γ, $\gamma = ((V,\rho), A, P)$, $L(\gamma) = L_m(\gamma)$, where m is the maximum length of the dominoes D_l and D_r in the rules (D_l, D_r) in P and of the sticky ends of the axioms in A, because for any derivation in γ generating an object w there is a derivation of bounded delay m in γ generating w, too. Now we can represent $L(\gamma)$ as the union and concatenation of languages generated by "subsystems" of γ : Let us write the axioms in A in the form $x_i y_i z_i$, $x_i, z_i \in S(V)$, $y_i \in \binom{V}{V}_\rho^+$, $1 \leq i \leq k$, $k \geq 1$, and $P = P_l \cup P_r$, where $P_l = \{(D_l, \lambda) \mid (D_l, \lambda) \in P\}$ and $P_r = \{(\lambda, D_r) \mid (\lambda, D_r) \in P\}$. Then $L(\gamma) = \cup_{i=1}^k L(\gamma_{i,l})\, g_i(L(\gamma_{i,r}))$, where $\gamma_{i,l} = ((V,\rho), \{x_i y_i\}, P_l)$, $\gamma_{i,r} = ((V,\rho), \{y_i z_i\}, P_r)$, and g_i is the generalized sequential machine erasing the string y_i from the beginning of a string, $1 \leq i \leq k$. Moreover, it is easy to verify that $L(\gamma_{i,l}) = \left(L\left(\gamma_{i,l}^r\right)\right)^r$, where the superscript r denotes the mirror image and $\gamma_{i,l}^r = ((V,\rho), \{(x_i y_i)^r\}, P_l^r)$, $P_l^r = \{(\lambda, (D_l)^r) \mid (D_l, \lambda) \in P_l\}$.

As REG is closed under arbitrary gsm mappings, union, concatenation, and mirror image, in the following without loss of generality we now may assume that A contains only one axiom from $\binom{V}{V}_\rho^+ S(V)$ and that in all the rules (D_l, D_r) in P we have $D_l = \lambda$. Then we can construct a regular grammar G, $G = (N, T_G, P_G, S)$, generating $L_m(\gamma)$ in the following way:

$$N = \left\{ \left[\binom{x_1}{\#} \dots \binom{x_m}{\#}\right] \mid \text{for some } k \leq m: \begin{array}{l} x_i \in V \text{ for } 1 \leq i \leq k \text{ and} \\ x_j = \# \text{ for } k+1 \leq j \leq m \end{array} \right\} \cup$$
$$\left\{ \left[\binom{\#}{y_1} \dots \binom{\#}{y_m}\right] \mid \text{for some } k \leq m: \begin{array}{l} y_i \in V \text{ for } 1 \leq i \leq k \text{ and} \\ y_j = \# \text{ for } k+1 \leq j \leq m \end{array} \right\};$$

$T_G = \left\{ \binom{a}{b} \mid a, b \in V,\ (a,b) \in \rho \right\}$;

P contains the following productions (in any case, we assume $a_i, b_i \in V$):

$$S \to \binom{a_1}{b_1} \dots \binom{a_k}{b_k} \left[\binom{a_{k+1}}{\#} \dots \binom{a_n}{\#} \binom{\#}{\#}^{m-n+k} \right],$$

if $A = \left\{ \binom{a_1}{b_1} \dots \binom{a_k}{b_k} \binom{a_{k+1}}{\#} \dots \binom{a_n}{\#} \right\}$ for some $k \geq 1$, $n \geq k$, and

$$S \to \binom{a_1}{b_1} \dots \binom{a_k}{b_k} \left[\binom{\#}{b_{k+1}} \dots \binom{\#}{b_n} \binom{\#}{\#}^{m-n+k} \right],$$

if $A = \left\{ \binom{a_1}{b_1} \dots \binom{a_k}{b_k} \binom{\#}{b_{k+1}} \dots \binom{\#}{b_n} \right\}$ for some $k \geq 1$, $n \geq k$.

Moreover, we include $X \to \binom{a_1}{b_1} \dots \binom{a_k}{b_k} Y$, where $X, Y \in N$ and $(a_i, b_i) \in \rho$ for $1 \leq i \leq k$, if for some l, n with $l \leq k \leq n$ we have:

1. $X = \left[\binom{a_1}{\#} \dots \binom{a_l}{\#} \binom{\#}{\#}^{m-l}\right]$, $Y = \left[\binom{a_{k+1}}{\#} \dots \binom{a_n}{\#} \binom{\#}{\#}^{m-n+k}\right]$,

   $\left(\lambda, \begin{smallmatrix} \# \\ b_1 \end{smallmatrix} \cdots \begin{smallmatrix} \# \\ b_l \end{smallmatrix} \begin{smallmatrix} a_{l+1} \\ b_{l+1} \end{smallmatrix} \cdots \begin{smallmatrix} a_k \\ b_k \end{smallmatrix} \begin{smallmatrix} a_{k+1} \\ \# \end{smallmatrix} \cdots \begin{smallmatrix} a_n \\ \# \end{smallmatrix}\right) \in P$, or

2. $X = \left[\binom{a_1}{\#} \dots \binom{a_l}{\#} \binom{\#}{\#}^{m-l}\right]$, $Y = \left[\binom{\#}{b_{k+1}} \dots \binom{\#}{b_n} \binom{\#}{\#}^{m-n+k}\right]$,

   $\left(\lambda, \begin{smallmatrix} \# \\ b_1 \end{smallmatrix} \cdots \begin{smallmatrix} \# \\ b_l \end{smallmatrix} \begin{smallmatrix} a_{l+1} \\ b_{l+1} \end{smallmatrix} \cdots \begin{smallmatrix} a_k \\ b_k \end{smallmatrix} \begin{smallmatrix} \# \\ b_{k+1} \end{smallmatrix} \cdots \begin{smallmatrix} \# \\ b_n \end{smallmatrix}\right) \in P$, or

3. $X = \left[\binom{\#}{b_1} \dots \binom{\#}{b_l} \binom{\#}{\#}^{m-l}\right]$, $Y = \left[\binom{\#}{b_{k+1}} \dots \binom{\#}{b_n} \binom{\#}{\#}^{m-n+k}\right]$,

   $\left(\lambda, \begin{smallmatrix} a_1 \\ \# \end{smallmatrix} \cdots \begin{smallmatrix} a_l \\ \# \end{smallmatrix} \begin{smallmatrix} a_{l+1} \\ b_{l+1} \end{smallmatrix} \cdots \begin{smallmatrix} a_k \\ b_k \end{smallmatrix} \begin{smallmatrix} \# \\ b_{k+1} \end{smallmatrix} \cdots \begin{smallmatrix} \# \\ b_n \end{smallmatrix}\right) \in P$, or

4. $X = \left[\binom{\#}{b_1} \dots \binom{\#}{b_l} \binom{\#}{\#}^{m-l}\right]$, $Y = \left[\binom{a_{k+1}}{\#} \dots \binom{a_n}{\#} \binom{\#}{\#}^{m-n+k}\right]$,

   $\left(\lambda, \begin{smallmatrix} a_1 \\ \# \end{smallmatrix} \cdots \begin{smallmatrix} a_l \\ \# \end{smallmatrix} \begin{smallmatrix} a_{l+1} \\ b_{l+1} \end{smallmatrix} \cdots \begin{smallmatrix} a_k \\ b_k \end{smallmatrix} \begin{smallmatrix} a_{k+1} \\ \# \end{smallmatrix} \cdots \begin{smallmatrix} a_n \\ \# \end{smallmatrix}\right) \in P$.

   $\left[\binom{\#}{\#}^m\right] \to \lambda$ is the final production.

Obviously, using these productions we can simulate the adjoining of dominoes to the right in any derivation of γ that is of bounded delay m, hence $L(G) = L_m(\gamma)$, which completes the proof. $\square$

If we consider bidirectional sticker systems with bounded delay, but allow to use pairs of adjoining rules, we can represent the linear languages.

Lemma 3. *Every linear language is the projection of a language in DBSL.*

Proof. Let $G = (N, T, P, S)$ be a linear grammar, i.e. without loss of generality we can assume that $N = N' \cup \{E\}$, $E \notin N'$, $P = P' \cup \{E \to \lambda\}$ and each production in P' is of the form $B \to uCv$ with $B \in N'$, $C \in N$, and $u, v \in T^*$. Moreover, let $m = max\{|u|, |v| \mid B \to uCv \text{ is a production in } P'\} + 1$. We now construct a bidirectional sticker system γ with $L(G) = g_T(L_m(\gamma))$ in such a way that the derivation of a word w in G is simulated by γ in the reversed order: $\gamma = ((V, \rho), A, P)$, $V = N \cup T$, $\rho = \{(x, x) \mid x \in N \cup T\}$, $A = \left\{\begin{smallmatrix} \# & E & \# \\ E & E & E \end{smallmatrix}\right\}$ and P contains $\left(\binom{S}{\#}, \binom{S}{\#}\right)$ and for any production $B \to uCv$ in P with $B \in N'$, $C \in N$, and $u, v \in T^*$ the following pairs of dominoes:

$$\begin{pmatrix} B & u & C \\ \# & \#^{|u|} & \# \end{pmatrix}, \begin{pmatrix} C & v & B \\ \# & \#^{|v|} & \# \end{pmatrix}, \begin{pmatrix} \# & \# & \#^{|u|} \\ B & B & u \end{pmatrix}, \begin{pmatrix} \#^{|v|} & \# & \# \\ v & B & B \end{pmatrix}.$$

The projection $g_T : \binom{V}{V}_\rho^* \to T^*$ is defined by $g_T\left(\binom{x}{x}\right) = \begin{cases} x, \text{ if } x \in T, \\ \lambda, \text{ if } x \in N. \end{cases}$

An object generated by this bidirectional sticker system represents the corresponding derivation for the string generated by the linear grammar G, i.e. if $A_1 \Longrightarrow_G u_1 A_2 v_1 \Longrightarrow_G \ldots \Longrightarrow_G u_1 \ldots u_{n-1} u_n v_n v_{n-1} \ldots v_1$ is the derivation generating the string $u_1 \ldots u_n v_n \ldots v_1$, where $A_1, \ldots, A_n \in N'$, $A_{n+1} = E$, and $A_1 = S$, then the corresponding derivation in γ is:

$$\begin{matrix} \# & E & \# \\ E & E & E \end{matrix} \Longrightarrow_\gamma \begin{matrix} A_n & u_n & E & E & E & v_n & A_n \\ \# & \#^{|u_n|} & E & E & E & \#^{|v_n|} & \# \end{matrix} \Longrightarrow_\gamma \cdots$$

$$\begin{matrix} A_1 & A_1 & u_1 & A_2 & \ldots & u_n & E & E & E & v_n & \ldots & A_2 & v_1 & A_1 & A_1 \\ A_1 & A_1 & u_1 & A_2 & \ldots & u_n & E & E & E & v_n & \ldots & A_2 & v_1 & A_1 & A_1 \end{matrix}$$

By applying g_T to result of this derivation in γ, we obviously obtain $u_1 \ldots u_n v_n \ldots v_1$. Hence, we conclude $g_T(L_m(\gamma)) = L(G)$. □

As we shall show in the following section, bidirectional sticker systems without the restriction of bounded delay derivations are very powerful and can represent every recursively enumerable language, yet as long as we only consider such derivations with bounded lengths of sticky ends at both sides, we do not go beyond linear languages.

Lemma 4. *DBSL* $\subseteq$ *LIN*.

Proof. Let $m \geq 1$ and γ be a bidirectional sticker system, $\gamma = ((V, \rho), A, P)$. We now construct a linear grammar G, $G = (N, T, P_G, S)$, generating the objects of $L_m(G)$ in the reversed order:

$N = \left\{ \left[\binom{x_1}{\#} \ldots \binom{x_{2m}}{\#} \right] \mid \text{for some } l, r \text{ with } 0 \leq l \leq m,\ m \leq r \leq 2m: \right.$
$x_i \in V$ for all $i \in \{j \mid 1 \leq j \leq l \text{ or } r+1 \leq j \leq 2m\}$,
$\left. x_j = \# \text{ for all } j \text{ with } l+1 \leq j \leq r \right\} \cup$
$\left\{ \left[\binom{\#}{y_1} \ldots \binom{\#}{y_{2m}} \right] \mid \text{for some } l, r \text{ with } 0 \leq l \leq m,\ m \leq r \leq 2m: \right.$
$y_i \in V$ for all $i \in \{j \mid 1 \leq j \leq l \text{ or } r+1 \leq j \leq 2m\}$,
$\left. y_j = \# \text{ for all } j \text{ with } l+1 \leq j \leq r \right\} \cup$
$\left\{ \left[\binom{x_1}{y_1} \ldots \binom{x_{2m}}{y_{2m}} \right] \mid \text{for some } l, r \text{ with } 0 \leq l \leq m,\ m \leq r \leq 2m: \right.$
$x_i \in V$ for $1 \leq i \leq l$ and $x_j = \#$ for $l+1 \leq j \leq 2m$,
$\left. y_i \in V \text{ for } r+1 \leq i \leq 2m,\ y_j = \# \text{ for } 1 \leq j \leq r \right\} \cup$
$\left\{ \left[\binom{x_1}{y_1} \ldots \binom{x_{2m}}{y_{2m}} \right] \mid \text{for some } l, r \text{ with } 0 \leq l \leq m,\ m \leq r \leq 2m: \right.$
$y_i \in V$ for $1 \leq i \leq l$ and $y_j = \#$ for $l+1 \leq j \leq 2m$,
$\left. x_i \in V \text{ for } r+1 \leq i \leq 2m,\ x_j = \# \text{ for } 1 \leq j \leq r \right\}.$

i.e. we use non-terminal symbols remembering at most $2m$ positions in the middle of the object derived so far.

$T = \left\{ \binom{a}{b} \mid a, b \in V,\ (a, b) \in \rho \right\}$;

P contains productions of the following kinds:

If $\binom{a_1}{\#}\cdots\binom{a_l}{\#}\binom{a_{l+1}}{b_{l+1}}\cdots\binom{a_k}{b_k}\binom{a_{k+1}}{\#}\cdots\binom{a_n}{\#}\in A$
for some $l,k,n\geq 0$ with $l<k\leq n$, then
$X\to\binom{a_1}{b_1}\cdots\binom{a_l}{b_l}\binom{a_{l+1}}{b_{l+1}}\cdots\binom{a_k}{b_k}\binom{a_{k+1}}{b_{k+1}}\cdots\binom{a_n}{b_n}$ in P_G with $X\in N$,
$X=\left[\binom{\#}{b_1}\cdots\binom{\#}{b_l}\binom{\#}{\#}\cdots\binom{\#}{\#}\binom{\#}{b_{k+1}}\cdots\binom{\#}{b_n}\right]$, $a_i,b_i\in V$, $(a_i,b_i)\in\rho$ for $1\leq i\leq n$, terminates a derivation in G at an axiom of γ, which started the corresponding derivation in γ. The remaining variants of the axiom and the variable X, respectively, are obvious and therefore left to the reader.

If $(D_l,D_r)\in P$ with
$D_l=\begin{smallmatrix}a_1\\\#\end{smallmatrix}\cdots\begin{smallmatrix}a_l\\\#\end{smallmatrix}\begin{smallmatrix}a_{l+1}\\b_{l+1}\end{smallmatrix}\cdots\begin{smallmatrix}a_k\\b_k\end{smallmatrix}\begin{smallmatrix}\#\\b_{k+1}\end{smallmatrix}\cdots\begin{smallmatrix}\#\\b_n\end{smallmatrix}$, $D_r=\begin{smallmatrix}\#\\d_1\end{smallmatrix}\cdots\begin{smallmatrix}\#\\d_r\end{smallmatrix}\begin{smallmatrix}c_{r+1}\\d_{r+1}\end{smallmatrix}\cdots\begin{smallmatrix}c_s\\d_s\end{smallmatrix}\begin{smallmatrix}c_{s+1}\\\#\end{smallmatrix}\cdots\begin{smallmatrix}c_t\\\#\end{smallmatrix}$,
for some $l,k,n,r,s,t\geq 0$ with $l\leq k\leq n$ and $r\leq s\leq t$, then
$X\to\binom{a_1}{b_1}\cdots\binom{a_k}{b_k}Y\binom{c_{r+1}}{d_{r+1}}\cdots\binom{c_t}{d_t}$ in P_G, where $X,Y\in N$,
$X=\left[\binom{\#}{b_1}\cdots\binom{\#}{b_l}\binom{\#}{\#}\cdots\binom{\#}{\#}\binom{\#}{d_{s+1}}\cdots\binom{\#}{d_t}\right]$,
$Y=\left[\binom{\#}{b_{k+1}}\cdots\binom{\#}{b_n}\binom{\#}{\#}\cdots\binom{\#}{\#}\binom{\#}{d_1}\cdots\binom{\#}{d_r}\right]$,
$(a_i,b_i)\in\rho$ for $1\leq i\leq k$, $(c_j,d_j)\in\rho$ for $1\leq j\leq s$, simulates a derivation step of γ in G in the reversed order. The remaining variants of the rule (D_l,D_r) and the variables X,Y, respectively, again are rather obvious and therefore left to the reader.

According to the construction given above, any derivation in γ that is of bounded delay m and generates an object w, $w\in\binom{V}{V}_\rho^+$, can be simulated by a corresponding derivation in G that generates w in the reversed order. Hence, we conclude $L(G)=L_m(\gamma)$. □

5 Characterizing the recursively enumerable languages

We now are going to show our main result, i.e. the representation of recursively enumerable languages by bidirectional sticker systems:

Lemma 5. *For any recursively enumerable language L, $L\subseteq T^*$, there exist a bidirectional sticker system γ and a morphism g such that $L=g(L(\gamma))$.*

Proof. We use the following characterization of L [5]:

For each recursively enumerable language L, $L\subseteq T^*$, there exist two non-erasing morphisms $h_1,h_2:\Sigma_2^*\to\Sigma_1^*$, a regular language $R\subseteq\Sigma_1^*$, and a projection $h_T:\Sigma_1^*\to T^*$ defined by $h_T(a)=\begin{cases}a, & \text{if } a\in T,\\ \lambda, & \text{if } a\in\Sigma_1\setminus T,\end{cases}$ such that

$L = h_T(h_1(E(h_1, h_2)) \cap R)$, *where* $E(h_1, h_2) = \{w \in \Sigma_2^* \mid h_1(w) = h_2(w)\}$.

Now let L be given according to the preceding characterization as $h_T(h_1(E(h_1, h_2)) \cap R$, and R be given by the (deterministic) finite automaton $M = (Q, \Sigma_1, \delta, q_0, F)$. We now define a bidirectional sticker system $\gamma = ((V, \rho), A, P)$ with $L = g(L(\gamma))$ by $V = \Sigma_1 \cup \tilde{\Sigma}_1 \cup Q \cup \{C, E, F, X, Z\}$, $\rho = \{(x, x) \mid x \in V\}$, $A = \left\{ \begin{smallmatrix} q_0 & X & Z \\ \# & X & \# \end{smallmatrix} \right\}$ and P containing the following pairs of adjoining rules:

For any $a \in \Sigma_2$, $h_1(a) = b_1 \ldots b_k$, $h_2(a) = c_1 \ldots c_m$, for any arbitrary states q_{i_j} from Q, $0 \leq j \leq m+1$, such that $\delta(q_{i_j}, c_j) = q_{i_{j+1}}, 0 \leq j \leq m$:

$$\left(\begin{matrix} q_{i_{m+1}} & \tilde{c}_m & q_{i_m} & \ldots & q_{i_2} & \tilde{c}_2 & q_{i_1} & q_{i_1} & \tilde{c}_1 & q_{i_0} \\ \# & \# & \# & \ldots & \# & \# & \# & \# & \# & \# \end{matrix} \right. ,$$
$$\left. \begin{matrix} b_1 & C & Z & b_2 & C & \ldots & Z & b_k & C & Z \\ \# & \# & \# & \# & \# & \ldots & \# & \# & \# & \# \end{matrix} \right).$$

To the left we produce the reversed image of a word $w \in \Sigma_2^*$ under h_2 at the same time guessing a valid path through the finite automaton M, whereas to the right we produce the image of w under h_1 the symbols of which separated by the characters C, Z.

$\left(\begin{smallmatrix} \# \\ \tilde{b} \end{smallmatrix}, \begin{smallmatrix} \# & \# \\ b & C \end{smallmatrix} \right)$ for all $b \in \Sigma_1$; by these rules, the correspondence (equality) of symbols on the left-hand side and on the right-hand side is checked.

$\left(\begin{smallmatrix} \# & \# \\ q & q \end{smallmatrix}, \begin{smallmatrix} \# \\ Z \end{smallmatrix} \right)$ for all $q \in Q$; by these rules, the validity of the previously guessed paths through the finite automaton M is checked.

$\left(\begin{smallmatrix} F & q_f \\ \# & \# \end{smallmatrix}, \begin{smallmatrix} E \\ \# \end{smallmatrix} \right)$ for all $q_f \in F$; $\left(\begin{smallmatrix} \# \\ F \end{smallmatrix}, \begin{smallmatrix} \# \\ E \end{smallmatrix} \right)$. If the checks described above are all successful and M has reached a final state, the sticky ends can be completed, which for the first time yields a configuration with blunt ends on both sides. If such an object is prolonged, it never can yield another object with blunt ends at both sides any more.

Moreover, we define $g : \binom{V}{V}_\rho^* \rightarrow T^*$ by $g\left(\binom{x}{x}\right) = \begin{cases} x, & \text{if } x \in T, \\ \lambda, & \text{if } x \in V \setminus T. \end{cases}$

From these constructions described above the reader can easily verify the equality $L = g(L(\gamma))$. □

Using the construction given in the preceding proof, we can derive a representation of recursively enumerable languages as the projection of the intersection of two minimal linear languages.

A *minimal linear grammar* is a linear grammar containing only one nonterminal symbol. Any language that can be generated by a minimal linear grammar is called a *minimal linear laguage.*

Corollary 1. *For any recursively enumerable language L, $L \subseteq T^*$, there exist two minimal linear languages $L_1, L_2 \subseteq ,^*$ and a projection $g_T : ,^* \to T^*$ such that $L = g_T\left(L_1 \cap L_2\right)$.*

Proof. We take the construction of the bidirectional sticker system given in the preceding proof and consider the upper row as the element of L_1 and the lower row as the element of L_2. Hence we construct the following two minimal linear grammars G_1 and G_2 :

$$\begin{array}{rcl} G_i & = & \left(\{S\}, , , P_i, S\right),\ i \in \{1,2\}, \\ , & = & \Sigma_1 \cup \tilde{\Sigma}_1 \cup Q \cup \{C, E, F, X, Z\}, \\ P_1 & = & \big\{S \to q_{i_{m+1}} \tilde{c}_m q_{i_m} \dots q_{i_2} \tilde{c}_2 q_{i_1} q_{i_1} \tilde{c}_1 q_{i_0} S b_1 C Z b_2 C \dots Z b_k C Z \mid a \in \Sigma_2, \\ & & h_1(a) = b_1 \dots b_k, h_2(a) = c_1 \dots c_m, \delta\left(q_{i_j}, c_j\right) = q_{i_{j+1}}, 0 \leq j \leq m\big\} \cup \\ & & \{S \to F q_f S E \mid q_f \in F\} \cup \{S \to q_0 X Z\}, \\ P_2 & = & \{S \to qqSZ \mid q \in Q\} \cup \Big\{S \to \tilde{b} S b C \mid b \in \Sigma_1\Big\} \cup \{S \to FSE, S \to X\}. \end{array}$$

The projection $g_T : ,^* \to T^*$ is defined by $g_T(a) = \begin{cases} a, & \text{if } a \in T, \\ \lambda, & \text{if } a \in , \setminus T. \end{cases}$

The result now follows immediately from the proof of Lemma 6. □

References

1. L. M. Adleman, Molecular computation of solutions to combinatorial problems, *Science*, **226**, 1021 – 1024 (Nov. 1994).
2. E. Csuhaj-Varjú, R. Freund, L. Kari, Gh. Păun, DNA computing based on splicing: universality results, *First Annual Pacific Symposium on Biocomputing*, Hawaii (Jan. 1996).
3. T. Head, Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors, *Bull. Math. Biology*, 737 – 759 (1987).
4. T. Head, Gh. Păun, D. Pixton, Language theory and molecular genetics. Generative mechanisms suggested by DNA recombination, in [9] (1997).
5. L. Kari, Gh. Păun, G. Rozenberg, A. Salomaa, S. Yu, DNA computing, sticker systems, and universality, *submitted,* (1996).
6. Gh. Păun, Splicing. A challenge to formal language theorists, *Bulletin EATCS*, **57**, 183 – 194 (1995).
7. Gh. Păun, Regular extended H systems are computationally universal, *J. Automata, Languages and Combinatorics*, **1**, 1, 27 – 36 (1996).
8. Gh. Păun, A. Salomaa, DNA computing based on the splicing operation, *Mathematica Japonica*, **43**, 3, 607 – 632 (1996).
9. G. Rozenberg, A. Salomaa, eds., *Handbook of Formal Languages,* Springer (1997).