Sticker Expressions[1]

José M Sempere
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia,
Camino de Vera s/n 46022 Valencia (Spain)
email:jsempere@dsic upv.es

Languages defined by DNA strands have been a key point in the recent developments on DNA computing and bioinformatics. It is commonly accepted that DNA strands and other biosequences form an unknown formal language which can be transformed by introducing new operations which can be carried out in the laboratory. In the recent times, there have been some proposals to describe in a formal framework such languages. So, Li [1] introduced an algebraic framework for DNA molecules and van Vliet *et al.* [2] proposed DNA expressions to model DNA molecules with *nicks* and *gaps*.

In this work we propose a formalism inspired by regular expressions in order to define *sticker languages*. We can say that sticker languages are sets of double stranded strings, subjected to a complementarity relation inspired by some new DNA computing models. The formalism that we propose to represent such languages is based on *linear expressions*, which were introduced to represent linear languages [3]. The advantages of such representation are clear: we can use methods to synthesize DNA computing models from them, we can study the descriptive complexity of such models in a compact framework and we can easily describe biological features of nature by introducing some sequences of symbols in the expressions (i.e. describing gene sequences in the expression in order to deduce computational models).

Given an alphabet Σ, a *sticker* over Σ will be the pair (x, y) such that $x = x_1 v x_2$, $y = y_1 w y_2$ with $x, y \in \Sigma^*$ and $\rho(v) = w$, where $\rho(v)$ means the complementary string of v. The sticker (x, y) will be denoted by $\binom{x}{y}$. Any sticker $\binom{x}{y}$ can be represented by $x\#y^r$ where $\# \notin \Sigma$. The product of two stickers $s = \binom{x}{y}$ and $p = \binom{z}{w}$ will be denoted by $sp = \binom{xz}{yw}$. Observe that this approach is different to the one proposed in [4], where Kari *et al.* proposed sticker systems as a generating device based on stickers, and the *sticking operation* which requires a complementarity relation between sequential parts of the products of every string. The sticker product leaves the complementarity relation free.

[1]Work supported by the Spanish Ministerio de Educación y Ciencia under project TIN2007-60769

An alternative representation for stickers can be proposed by introducing *indexed alphabets*. Let $\Sigma = \{a_1, a_2, \ldots, a_n\}$ and $\Delta = \{b_1, b_2, \ldots, b_m\}$ be two alphabets. We define the alphabet Σ indexed by Δ, denoted as Σ_Δ, as the new alphabet $\{a_{1_{b_1}}, \ldots, a_{1_{b_m}}, \ldots, a_{n_{b_1}}, \ldots, a_{n_{b_m}}\}$. Then, the sticker $\binom{x_1 x_2 \ldots x_m}{y_1 y_2 \ldots y_n}$ over Σ can be represented by a string defined over the indexed alphabet $\Sigma_{\{u,l\}}$ as

$$(x_1)_u(x_2)_u \ldots (x_m)_u(y_1)_l(y_2)_l \ldots (y_n)_l$$

Here, u denotes the upper strand and l the lower one.

Let $\Sigma_{\{l,u\}}$ be an indexed alphabet and $x \in \Sigma^*_{\{l,u\}}$. Now we can inductively define the *image of x over Σ*, denoted by $im_\Sigma(x)$, as follows

1. If $x = \lambda$ then $im_\Sigma(\lambda) = \binom{\lambda}{\lambda}$
2. If $x = a_u w$, with $w \in \Sigma^*_{\{l,u\}}$, then $im_\Sigma(a_u w) = \binom{a}{\lambda} \cdot im_\Sigma(w)$
3. If $x = a_l w$, with $w \in \Sigma^*_{\{l,u\}}$, then $im_\Sigma(a_l w) = \binom{\lambda}{a} \cdot im_\Sigma(w)$

Definition Let the $\Sigma_{\{l,u\}}$ be an indexed alphabet. Then a *sticker* expression over $\Sigma_{\{l,u\}}$ is defined inductively as follows

1. $\emptyset$ is a sticker expression
2. λ is a sticker expression
3. $(\forall a \in \Sigma)$ a_l and a_u are sticker expressions
4. If r and s are sticker expressions then so are $(r)+(s)$, $(r)(s)$ and $(r)^*$

References

[1] Z. Li. Algebraic Properties of DNA operations. *Biosystems* 52, pp 55-61 1999.

[2] R. van Vliet, H.J. Hoogeboom, G. Rozenberg. Combinatorial Aspects of Minimal DNA Expressions. In *Proceedings of the 10th International Workshop on DNA Computing DNA10 (June, 2004)*, pp 375-388. LNCS 3384. Springer 2005.

[3] J.M. Sempere. On a class of regular-like expressions for linear languages. *Journal of Automata, Languages and Combinatorics* Vol. 5, Number 3, pp 343-354. 2000.

[4] L. Kari, Gh. Păun, G. Rozenberg, A. Salomaa, S. Yu. DNA computing, sticker systems, and universality. *Acta Informatica* 35, pp 401-420. 1998.