Families of Languages Associated with SN P Systems: Preliminary Ideas, Open Problems

Gheorghe Păun[1], José M. Sempere[2]

[1] Institute of Mathematics of the Romanian Academy
Bucharest, Romania
gpaun@us.es
[2] Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia, Spain
jsempere@dsic.upv.es

Summary. By considering various encodings of the spike train generated by a spiking neural (SN) P system, several related languages are obtained. A few proposals on how to define such a family of languages, some preliminary results, and a series of suggestions for further research are briefly presented.

1 Basic Definitions, Basic Problems

The reader is supposed to be familiar with SN P systems (in general, with membrane computing area), so that we pass directly to introducing the families of languages mentioned above.

The idea is simple: to take the binary language $L_1(\Pi)$ generated by an SN P system Π (the set of all binary strings – also called spike trains – describing halting computations of Π: a symbol 0 is associated with a step when no spike is sent to the environment, and a symbol 1 is associated with a step when at least one spike is sent to the environment, until the system halts), and to encode blocks of k digits, for various natural numbers k, in such a way that various languages $L_k(\Pi)$ are obtained.

Of course, we have to take care of the case when the spike train is not of a length which is a multiple of the considered k. In this case, we add symbols 0 so that the obtained binary string is of a length divisible by k.

More formally, let $B = \{0, 1\}$ be the binary alphabet, let $k \geq 1$ be a natural number, and V_k be an alphabet. Consider a mapping $\varphi_k : B^k \longrightarrow V_k$. For each string $w \in B^*$ we consider the string ${}_kw = w0^t$, where $t = \min\{n \geq 0 \mid |w0^n| \text{ is a multiple of } k\}$.

The string ${}_kw$ can be written in the form ${}_kw = x_1x_2 \dots x_s$, such that $|x_j| = k$ for all $j = 1, 2, \dots, s$. Then, φ_k can be extended to $(B^k)^*$ in the natural way: $\varphi_k(y_1y_2 \dots y_t) = \varphi_k(y_1)\varphi_k(y_2) \dots \varphi_k(y_t)$ for all $y_i \in B^k, 1 \leq i \leq t, t \geq 0$.

Thus, for an SN P system Π and an encoding φ_k as above, we can define the language

$$L_{\varphi_k}(\Pi) = \{\varphi_k({}_kw) \mid w \in L_1(\Pi)\}.$$

This language depends on the encoding φ_k, hence a family of languages can be associated with Π by varying k and the mapping φ_k.

Already at this very general level there appear several research issues. Consider classes of mappings φ_k with various properties and investigate the properties (size, closure, decidability, etc.) of the corresponding families of languages generated by SN P systems. How these properties depend on the SN P systems? Following suggestions and importing notions and questions from the grammar forms area (a very active research area some decades ago, starting with the pioneering paper [2] – see also the corresponding chapter from the first volume of [5]) can be useful.

2 The One-to-one Case

Actually, the present research idea occurred in a framework related to classical communication channels with encoded information, where with every SN P system different languages can be associated that depend on a parameter that fixes a time window to analyze the spikes train at the output.

A natural possibility is to order in a precise way, e.g., lexicographically, the strings in B^k, and to associate with each of them a distinct symbol from an alphabet Σ_k with 2^k elements.

The fact that the encoding is one-to-one is rather restrictive: the passing from the binary language $L_1(\Pi)$ to a given $L_k(\Pi)$ (we omit mentioning the mapping φ) can be done by means of a sequential transducer (a gsm, in the usual terminology, [5]). Conversely, the passage from $L_k(\Pi)$ to $L_1(\Pi)$ is done by an one-to-one (non-erasing) morphism, which implies that the converse passage is done by an inverse morphism.

This observation is interesting enough to be formally formulated:

Proposition 1. *If $L_1(\Pi) \in FL$, where FL is a family of languages closed under gsm mappings or under inverse morphisms, then $L_k(\Pi) \in FL$, for all $k \geq 1$. If FL is closed under non-erasing morphisms and $L_k(\Pi) \in FL$, then also $L_1(\Pi) \in FL$.*

Families as FL above are REG, LIN, CF in the Chomsky hierarchy, hence if $L_1(\Pi)$ is regular/linear/context-free, then also all languages $L_k(\Pi)$ are regular/linear/context-free, respectively, and conversely.

This means that each family $F(\Pi) = \{L_k(\Pi) \mid k \geq 1\}$ contains only languages of the same type in the Chomsky hierarchy (for instance, it is not possible to have a context-free non-regular language $L_k(\Pi)$ together with a regular language $L_j(\Pi)$, for some $k \neq j$.

Of course, if $L_1(\Pi)$ is finite, then all languages $L_k(\Pi)$ are finite, hence the family $F(\Pi)$ is finite, up to a renaming of symbols of alphabets Σ_k.

If, instead, $L_1(\Pi)$ is infinite, then $F(\Pi)$ can be an infinite family, because the alphabet of $L_{k+1}(\Pi)$ might be larger than the alphabet of $L_k(\Pi)$.

This is the case, for instance, for the SN P system Π generating $L_1(\Pi) = \{1^n01^m \mid n, m \geq 1\}$ (which is an infinite regular language).

This assertion seems to be true (*conjecture*) for all SN P systems Π with infinite $L_1(\Pi)$.

3 The Non-injective Case

The previous type-preserving Proposition 1 does not hold in the case of using encodings which are not one-to-one.

Here is an example: Consider Π such that $L_1(\Pi) = \{1^n01^n \mid n \geq 1\}$ (SN P systems are universal, [4], hence any language can be taken as the starting language). Of course, $L_1(\Pi)$ is context-free non-regular.

Consider the encoding $\varphi_k : B^k \longrightarrow \{a, b\}$ defined by $\varphi_k(w) = a$ if $|w|_0 \leq 1$, and $\varphi_k(w) = b$ if $|w|_0 \geq 2$. We get

$$L_k(\Pi) = a^+ \cup a^*b, \text{ for } k \geq 4,$$

and

$$L_k(\Pi) = a^+ \cup a^+b, \text{ for } k = 2, 3.$$

Clearly, the languages $L_k(\Pi), k \geq 2$, are regular, in spite of the fact that $L_1(\Pi)$ is (context-free) non-regular.

The properties of the encoding is crucial for the properties of the obtained language families (this is true in other frameworks, see, e.g., [3] and its references), hence this issue deserves further research efforts.

4 Final Comments

The idea of associating a family of languages with a given P system is rather natural. We have illustrated it here with the case of SN P systems, but the same strategy can be applied for any type of P systems producing a language (such that cell-like P systems with external output, SN P systems generating trace languages [1], etc.).

A more systematic study of this idea is of interest, starting with relevant examples, continuing with "standard" formal language theory questions, and ending with possible applications of this approach (as languages generated by the same P system are "genetically" related, maybe in this way one can capture biological connections/dependencies or other types of relationships).

Of course, further ways to associate a family of languages to a given SN P system, to a given P system in general, remain to be found.

References

1. H. Chen, M. Ionescu, A. Păun, Gh. Păun, B. Popa: On trace languages generated by spiking neural P systems, *Eighth International Workshop on Descriptional Complexity of Formal Systems* (DCFS 2006), June 21-23, 2006, Las Cruces, New Mexico, USA, 94–105.
2. A.B. Cremers, S. Ginsburg: Context-free grammar forms. *J. Computer System Sci.*, 11 (1975), 86–116.
3. E. Csuhaj-Varjú, G. Vaszil: On counter machines versus dP automata. *Membrane Computing. 14th Intern. Conf., CMC 2013, Chişinău, August 2013* (A. Alhazov et al., eds.), LNCS 8340, Springer, Berlin, 2014, 138–150.
4. M. Ionescu, Gh. Păun, T. Yokomori: Spiking neural P systems. *Fundamenta Informaticae*, 71, 2-3 (2006), 279–308.
5. G. Rozenberg, A. Salomaa, eds.: *Handbook of Formal Languages*, 3 vols., Springer-Verlag, Berlin, 1997.