A Note on P Systems with Small-Size Insertion and Deletion*

Artiom Alhazov[1,2], Alexander Krassovitskiy[3], Yurii Rogozhin[1,3], Sergey Verlan[4]

[1] Institute of Mathematics and Computer Science
Academy of Sciences of Moldova
Academiei 5, Chişinău MD-2028 Moldova
{artiom,rogozhin}@math.md

[2] IEC, Department of Information Engineering, Graduate School of Engineering
Hiroshima University,
Higashi-Hiroshima 739-8527 Japan

[3] Research Group on Mathematical Linguistics, Rovira i Virgili University
Av. Catalunya, 35, Tarragona 43002 Spain
alexander.krassovitskiy@estudiants.urv.cat

[4] LACL, Département Informatique, Université Paris Est
61 av. Général de Gaulle, 94010 Créteil, France
verlan@univ-paris12.fr

Summary. We present an overview of recent results on small size insertion-deletion P systems. Together with the ordinary definition we consider systems with priority of deletion over insertion. In both cases, obtained P systems are strictly more powerful than ordinary insertion-deletion systems, and in most of the cases they are computationally complete. We list several such results. When using the priority relation, the computational completeness may be achieved by context-free insertion and deletion of one symbol only.

1 Introduction and definitions

Insertion and the deletion operations originate from the language theory, being introduced mainly with linguistic motivation. In general form, an insertion operation means adding a substring to a given string in a specified (left and right) context, while a deletion operation means removing a substring of a given string from a specified (left and right) context. More precisely, an *insertion* rule (u, α, v) corre-

* A.A. acknowledges the support of the Japan Society for the Promotion of Science, and the Grant-in-Aid, project 20·08364. A.A., Y.R. and S.V. acknowledge the support by the Science and Technology Center in Ukraine, project 4032. Y.R. acknowledges the support of the European Commission, project MolCIP, MIF1-CT-2006-021666. A.K. acknowledges the support PIF program of University Rovira i Virgili, and project no. MTM2007-63422 from the Ministry of Science and Education of Spain.

sponds to the rewriting rule $uv \to u\alpha v$, and a *deletion* rule (u, α, v) corresponds to the rewriting rule $u\alpha v \to uv$.

An *insertion-deletion system* is a construct $ID = (V, T, A, I, D)$, where V is an alphabet, $T \subseteq V$, A is a finite language over V, and I, D are finite sets of insertion and deletion rules, respectively. The language $L(ID)$ generated by ID is defined as $\{w \in T^* \mid A \ni x \Longrightarrow^* w\}$, where $\Longrightarrow$ is the relation defined by an insertion or deletion rule.

The *size* of an InsDel system $ID = (V, T, A, I, D)$ is defined as $(n, m, m'; p, q, q')$, where

$$n = \max_{(u,\alpha,v)\in I} |\alpha|,\ m = \max_{(u,\alpha,v)\in I} |u|,\ m' = \max_{(u,\alpha,v)\in I} |v|,$$
$$p = \max_{(u,\alpha,v)\in D} |\alpha|,\ q = \max_{(u,\alpha,v)\in D} |u|,\ q' = \max_{(u,\alpha,v)\in D} |v|.$$

The corresponding families of languages are denoted by $INS_n^{m,m'}DEL_p^{q,q'}$. Insertion-deletion systems of a "sufficiently large" size characterize RE.

An *insertion-deletion P system* is a tuple $\Pi = (O, T, \mu, M_1, \ldots, M_n, R_1, \cdots, R_n)$, where O is a finite alphabet, $T \subseteq O$ is the terminal alphabet, μ is the (tree) structure of n membranes, it can be represented by a string of correctly nested labeled parentheses. Region i is delimited by membrane i, $1 \leq i \leq n$. The set M_i is a finite language of initial objects, and R_i is a set of insertion and deletion rules of region i, of the following forms: $(u, x, v; tar)_a$, where (u, x, v) is an insertion rule, and $(u, x, v; tar)_e$, where (u, x, v) is a deletion rule, and the *target indicator* tar is from the set $\{here, in_j, out \mid 1 \leq j \leq n\}$. The configurations, transitions and computations of the system are defined in the standard way. The result $L(\Pi)$ of generated by Π consists of strings over T ever sent out of the system during its computations.

We denote by $ELSP_k(ins_p^{m,m'}, del_p^{q,q'})$ the family of languages $L(\Pi)$ generated by insertion-deletion P systems with $k \geq 1$ membranes and insertion and deletion rules of size at most $(n, m, m'; p, q, q')$. We omit letter E if $T = O$. If deletion rules have a priority over insertion rules, the corresponding class is denoted as $(E)LSP_k(ins_p^{m,m'} < del_p^{q,q'})$. Letter "t" is inserted before P to denote classes for the tissue case (defined similarly to the membrane case). We write $*$ if corresponding parameter is unbounded.

We consider register machines with three types of instructions: $p : (\mathrm{ADD}(k), q, s)$, $p : (\mathrm{SUB}(k), q, s)$ and $p : (\mathrm{WRITE}(A), q)$, where p, q, s are states, k is a register, and A is a symbol. The last form of instruction writes a symbol on the output tape. Register machines generate $PsRE$. RE is generated if WRITE instructions are used.

2 Minimal context-free insertion-deletion P systems

Systems in $INS_1^{0,0}DEL_*^{0,0}$ only generate strings obtained by inserting any number of specific symbols anywhere in words of a finite language ([6]); this is included in the regular languages family; strictly as, e.g., for $\{a^*b^*\}$ the system has no control on the place of insertion or deletion in the string and the initial language is finite. Therefore, $INS_1^{0,0}DEL_1^{0,0} \subset REG$.

Adding a membrane structure, by mutual simulation of blind register machines (which do not have the zero check in the decrement instruction) it can be obtained:

Theorem 1. *$PsStP_*(ins_1^{0,0}, del_1^{0,0}) = PsMAT$.*

However, $\{a^*b^*\}$ cannot be generated, while non-context-free languages are generated even without priorities and deletion. Therefore,

Theorem 2. *$REG \backslash LStP_*(ins_n^{0,0} < del_1^{0,0}) \neq \emptyset$, for any $n > 0$, and $LStP_*(ins_1^{0,0}, del_0^{0,0}) \setminus CF \neq \emptyset$.*

A more general inclusion holds:

Theorem 3. *$ELStP_*(ins_n^{0,0}, del_1^{0,0}) \subset MAT$, for any $n > 0$.*

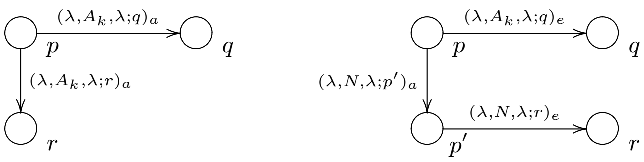

Fig. 1. Simulating $p : (ADD(k), q, r)$(left) and $p : (SUB(k), q, r)$ (right).

Nevertheless, minimal context-free insertion-deletion systems with priorities do generate $PsRE$. This is especially clear for the tissue P systems: jumping to an instruction corresponds to sending a string to the associated region, and the entire construction is a composition of graphs shown in Figure 1. Notice the use of priority of deletion over insertion in decrement.

A more sophisticated proof can be done for the tree-like membrane structure [2].

Theorem 4. *$PsSP_*(ins_1^{0,0} < del_1^{0,0}) = PsRE$.*

3 Small contextual insertion-deletion P systems

Although Theorem 4 shows that the systems from the previous section are quite powerful, they cannot generate RE without control on the place where a symbol is inserted. Once we allow a context in insertion or deletion rules, they can.

Theorem 5. *$LSP_*(ins_1^{0,1} < del_1^{0,0}) = RE$.*

A similar result holds if contextual deleting operation is allowed.

Theorem 6. *$LSP_*(ins_1^{0,0} < del_1^{1,0}) = RE$.*

Proof. As in Theorem 5, we use the construction from Theorem 4. However, $7k$ additional membranes are needed to simulate k writing instructions.

The WRITE instruction is simulated by inserting symbols to be written (together with some temporary marking symbols) in the string and performing a check that they are located at the end by using trap rules and the priority of the deletion. If the corresponding symbol is not inserted at the rightmost position, then it would be able to delete the symbol that just follows it. In such a way the trapping mechanism is realized.

Corollary 1. $LSP_*(ins_1^{1,0} < del_1^{0,0}) = LSP_*(ins_1^{0,0} < del_1^{0,1}) = RE.$

The contextual deletion can be replaced by a context-free deletion of two symbols.

Theorem 7. $LSP_*(ins_1^{0,0} < del_2^{0,0}) = RE.$

We mention that the counterpart of Theorem 7 obtained by interchanging parameters insertion and deletion rules is not true, see Theorem 2.

If one considers a context dependency in both insertion and deletion rules, then the priority relation can be avoided. The following results are from [3].

Theorem 8. $ELSP_5(ins_1^{1,0}del_1^{1,0}) = ELSP_5(ins_1^{1,0}del_2^{0,0}) =$
$REELSP_5(ins_2^{0,0}del_1^{1,0}) = RE.$

Note that corresponding insertion-deletion systems cannot generate RE. In particular, the last two systems cannot generate the language $L = (ba)^+$, [3]. In the same article, a characterization of the class $INS_1^{1,0}DEL_1^{1,0}$ in terms of context-free grammars is also given.

Finally, we remark that the context is important when no priorities are used:

Theorem 9. $REG \setminus ELSP_*(ins_2^{0,0}del_2^{0,0}) \neq \emptyset.$

As it was shown in [3], the language $L_{ab} = a^*b \notin ELSP_k(ins_2^{0,0}del_2^{0,0})$, for any $k \geq 1$.

References

1. A. Alhazov, A. Krassovitskiy, Y. Rogozhin, S. Verlan: P Systems with Minimal Insertion and Deletion. In R. Gutiérrez-Escudero, et al eds., *Proc. of Seventh Brainstorming Week on Membrane Computing Sevilla, February 2-6, 2009*, Fénix Editora, Sevilla, Vol. I, 9–21.
2. A. Alhazov, A. Krassovitskiy, Y. Rogozhin, S. Verlan: P Systems with Minimal Insertion and Deletion, submitted. Extended version of [1].
3. A. Krassovitskiy, Yu. Rogozhin, S. Verlan: Computational Power of Insertion-Deletion (P) Systems with Rules of Size Two, submitted. Extended version of [4, 5].
4. A. Krassovitskiy, Yu. Rogozhin, S. Verlan: One-Sided Insertion and Deletion: Traditional and P Systems Case, *CBM* 2008, Vienna, 53–64.
5. A. Krassovitskiy, Yu. Rogozhin, S. Verlan: Computational Power of P Systems with Small Size Insertion and Deletion Rules. *CSP* 2008, Cork, 137–148.
6. S. Verlan: On Minimal Context-Free Insertion-Deletion Systems. *Journal of Automata, Languages and Combinatorics* **12**, 2007, 1/2, 317-328.