On Restricted Insertion-Deletion Systems

István Katsányi

The insertion grammars (or semi-contextual grammars) were introduced in [1] (see also [2]) as a model of the constructions of natural languages. It is an important model of formal languages of its own right, but it gained even more significance by the emerging of the field of DNA computing, since using a standard laboratory technique called *PCR site-specific oligonucleotide mutagenesis* insertions or deletions of nucleotide sequences into or from the strands of DNA molecules are possible. Hence by inspecting the practical applicability of the formal models we may gain functioning molecular computers. In the past few years several papers showed, that various generative mechanisms in formal language theory that use insertion and deletion operations are capable of generating any recursively enumerable languages [3, 4, 5, 6, 7, 8, 9, 10, 11]. Since such systems are also models of molecular computing, for practical reasons it is important to examine these systems in a restricted case, in which the number of symbols in the model of the alphabet is limited. In [6] it is showed that we can define the generated language of an insertion-deletion system in such a way, that a two-letter alphabet is enough to generate any recursively enumerable language. In this work we complete this result by showing that the same generative capacity can be obtained even if we define the generated language the traditional way.

An *insertion-deletion system* (or shortly an *insdel system*) is a construct $\gamma = (V, T, A, I, D)$, where V is a finite *alphabet*, $T \subseteq V$ is the *terminal alphabet*, $A \subseteq V^*$ is the finite set of *axioms* and $I, D \subseteq V^* \times V^* \times V^*$ are the finite sets of *insertion and deletion rules*, respectively. For two words $x, y \in V^*$ the relation $x \Longrightarrow_\gamma y$ holds when either $x = x_1uvx_2$, $y = x_1uzvx_2$, $x_1, x_2 \in V^*$ and $(u, z, v) \in I$, or $x = x_1uzvx_2$, $y = x_1uvx_2$, $x_1, x_2 \in V^*$ and $(u, z, v) \in D$. Let $\Longrightarrow_\gamma{}^*$ be the reflexive, transitive closure of $\Longrightarrow_\gamma$. The language generated by γ is $L(\gamma) = \{w \in T^* \mid x \Longrightarrow_\gamma{}^* w, x \in A\}$. In papers [3, 4, 6, 8, 9, 11] we can find different proofs for even stronger variants of the following theorem:

Theorem. 0.1. *The family of languages generated by insertion-deletion systems equals to the family of recursively enumerable languages.*

In the paper we define a new kind of insertion-deletion system which has additional constraints comparing to the regular model described earlier, but more general as the restrained system defined in [6]. We also show that in spite of the restrictions it is capable of universal computation. A *restricted insertion-deletion system* is a construct $\gamma = (V, T, h, A, I, D)$, where V is an *alphabet* consisting of two letters, T is a finite alphabet called the *terminal alphabet*, $h : T^* \to V^*$ is a λ-free morphism, A is a finite subset of V^*, the set of *axioms*, I and D are finite subsets of $V^* \times V^* \times V^*$, the *insertion and deletion rules*, respectively. The role of V, A, I and D coincides with the regular model. The relation $\Longrightarrow_\gamma$ is also defined the usual way. The morphism h is needed to define languages over an arbitrary finite alphabet. The language generated by γ is $L(\gamma) = h^{-1}(\{w \in V^* \mid z \Longrightarrow^* w, \text{ where } z \in A\})$.
In the paper we give two different proofs of the following theorem:

Theorem. 0.2. *The family of languages generated by restricted insertion-deletion systems equals to the family of recursively enumerable languages.*

References

[1] B. S. Galiukschov. Semicontextual grammars (in russian). *Mat. Logica i Mat. Ling.*, pages 38–50, 1981.

[2] S. Marcus. Contextual grammars. *Rev. Roum. Math Pures Appl.*, 14:1525–1534, 1969.

[3] Carlos Martin-Vide, Gheorghe Păun, and Arto Salomaa. Characterizations of recursively enumerable languages by means of insertion grammars. *Theoretical Computer Science*, 205:195–205, 1998.

[4] Lila Kari and Gabriel Thierrin. Contextual insertions / deletions and computability. *Information and Computation*, 131(1):47–61, November 25 1996.

[5] Mark Daley, Lila Kari, Greg Gloor, and Rani Siromoney. Circular contextual insertions / deletions with applications to biomolecular computation. In *SPIRE/CRIWG*, pages 47–54, 1999.

[6] Lila Kari, Gheorghe Păun, Gabriel Thierrin, and Sheng Yu. At the crossroads of DNA computing and formal languages: Characterizing recursively enumerable languages using insertion-deletion systems. In *Proceedings of the 3rd DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June 23 – 25, 1997*, pages 318–333.

[7] Lila Kari. *On Insertion and Deletion in Formal Languages*. PhD thesis, University of Turku, 1991.

[8] Gheorghe Păun, Grzegorz Rozenberg, and Arto Salomaa. *DNA Computing. New Computing Paradigms*. Springer-Verlag, Berlin, 1998.

[9] A. Takahara and T.Yokomori. On the computational power of insertion-deletion systems. *Proceedings of the 8th International Workshop on DNA-based Computers, Sapporo, Japan June 10–13, 2002, also in LNCS vol. 2568, 2003, pages 269–280*.

[10] Maurice Margenstern, Gheorghe Păun, and Yurii Rogozhin. On the power of (molecular) crowd: Set-conditional string processing. *Proceedings of AFL'02, Budapest, Hungary, August 13–18, 2002*.

[11] Maurice Margenstern, Gheorghe Păun, Yurii Rogozhin, and Sergey Verlan. Context-free insertion-deletion systems. *Proceedings of DCFS 2003, the 5th Workshop on Descriptional Complexity of Formal Systems, Budapest, Hungary, July 12–14, 2003*.