

# Graph Transformation in Molecular Biology<sup>\*</sup>

Francesc Rosselló<sup>1</sup> and Gabriel Valiente<sup>2</sup>

<sup>1</sup> Department of Mathematics and Computer Science,  
Research Institute of Health Science (IUNICS), University of the Balearic Islands,  
Palma de Mallorca, Spain

<sup>2</sup> Department of Software, Technical University of Catalonia, Barcelona, Spain  
`cesc.rossello@uib.es`, `valiente@lsi.upc.edu`

**Abstract.** In the beginning, one of the main fields of application of graph transformation was biology, and more specifically morphology. Later, however, it was like if the biological applications had been left aside by the graph transformation community, just to be moved back into the mainstream these very last years with a new interest in molecular biology. In this paper, we review several fields of application of graph grammars in molecular biology, including: the modelling of higher-dimensional structures of biomolecules, the description of biochemical reactions, and the study of biochemical pathways.

## 1 Introduction

Once upon a time, biology was one of the main fields of application of graph transformation, as it is proved by the maiden name (back in 1978) “Workshop on Graph Grammars and Their Application to Computer Science and Biology” of the current “International Conference on Graph Transformation.” Those early applications of graph rewriting in biology mostly belonged to the field of morphogenesis.

It is common knowledge that graphs describe structures in a simplified but explicit way. In such descriptions, nodes correspond to substructures and arcs represent relations among substructures. These arcs can be directed if the relation is so, labelled if one wants to record the kind of relation they stand for, and so on. On their turn, nodes may be labelled to make explicit what they symbolize, with labels that may be not only raw names, but also graphs themselves, or other higher-order objects that can be used to abstract the details of the substructure represented by the nodes in hierarchical structures. In any case, the actual meaning of the nodes and the arcs will depend on the actual application. Under this graphical representation of structures, the evolution of the latter can be described by graph rewriting mechanisms, where one or several subgraphs are replaced by other graphs in a way determined by evolution rules specified in a graph grammar.

It was soon noticed that the development states of an organism can be described as graphs in this way, with nodes representing for instance cells, body

---

<sup>\*</sup> This work has been partially supported by the Spanish CICYT, project MAVERISH (TIC2001-2476-C03-01) and by the Spanish DGES and the EU program FEDER, project BFM2003-00771 ALBIOM.

segments, or tissues, and arcs representing spatial or biological relations among nodes. The nodes' labels may be used to denote their type and the arcs' labels the type of interaction they stand for. The rules governing some aspect of the development of such an organism can be described in this framework as graph rewriting rules and gathered in a graph grammar. In a given application, these rules can be fired simultaneously, in a synchronized way, or following some priority order. It was precisely the possibility of modelling the development of organisms where changes and segmentations take place simultaneously at different places that lead to the notion of parallel graph grammars, also called *graph L-systems*, as a generalization of string L-systems. They were introduced about thirty years ago by K. Culik and A. Lindenmayer [12], previously hinted by B. Mayoh [42], and they have been used since then in many applications of graph rewriting in morphogenesis.

This was the first kind of applications of graph rewriting in biology, and, as a matter of fact, the use of graph grammars as models of the development of organisms is still alive. For instance, Beck, Benkő et al [2] have proposed recently the use of graph transformation as an alternative to standard morphospace representations and geometric morphometrics in the field of theoretical morphology, while Tomita, Kurokawa and Murata [63] have introduced a new type of graph rewriting systems, *graph automata*, as an alternative to graph L-systems in the description of self-reproducing complex systems.

The success of graph grammars in the description of development pathways can be seen as a simple instance of their pattern handling power. According to D. Gernert [33], as soon as patterns are represented as graphs, graph grammars are a natural tool to describe the fundamental operations related to patterns: pattern generation, pattern transfer (the duplication of a certain subpattern and its insertion in a different location), pattern recognition, pattern interpretation (the influence of certain subpatterns on the behavior of whole system) and pattern application (the transmission of a certain pattern to another location). A type of graph grammars specifically tailored to handle patterns was proposed in [41].

Patterns that are conveniently modelled as graphs are found everywhere in biology, and not only in morphology. Molecular biology is no exception: the inner structure of chemical compounds [9], the tridimensional structure of nucleic acids and proteins [64], the chemical reactions [31], the biochemical networks [45], most formal components of molecular biology can be represented as graphs. This fact must be added to what is called in sociology of science “the phenomenon of the earlier tool” [33]: when some branch of mathematics reaches a high standard or it becomes fashionable, then it will be surely used in many other sciences<sup>1</sup>.

---

<sup>1</sup> Historians of science put more emphasis on the converse phenomenon, when a problem in some science gives rise to new a branch of mathematics or gives new life to an already existing branch; for instance, the theory of Abstract Data Types gave a boost to universal algebra. . . and H. Ehrig [24] had his share of guilt! Graph grammars can also be seen as an example of this phenomenon, as they were born to solve the problem of specifying the transformation of non linear structures in software systems.