ASPECTS OF CONCURRENCY
IN GRAPH GRAMMARS

Hartmut Ehrig*
Technische Universität Berlin,
D-1000 Berlin 10, Franklinstr. 28/29

ABSTRACT: This paper is a written version of the overview lecture on "Concurrency in Graph Grammars" given at the "2nd International Workshop on Graph Grammars and their Applications to Computer Science, 1982". The intention of that lecture and this paper is to show that a number of results in the transformational theory of graph grammars can be considered already as contributions towards a theory of concurrency in graph grammars. Simulations of Petri nets within graph grammars are reviewed and simulations of algebraic specifications within graph grammars are introduced to cover also abstract data type concepts. More general concepts of concurrency are considered to be studied in the framework of graph grammars which go essentially beyond those in Petri nets. Finally it is proposed to combine the new approach of Ugo Montanari for synchronization with the aspects of concurrency reviewed above to obtain a graph grammar based model for concurrent and distributed systems.

ACKNOWLEDGEMENTS: The results summarized in this paper are based on contributions including the following coauthors: Annegret Habel, Dirk Jannsens, Hans-Jörg Kreowski, Andrea Maggiolo-Schettini, Peter Padawitz, Barry Rosen, Grzegorz Rozenberg, Josef Winkowski. Thanks to all of them and especially to Annegret Habel and Horst Hansen for fruitful discussions on the new material in this paper. For excellent typing we are once again grateful to Helga Barnewitz.

INTRODUCTION

Problems of concurrency have been studied for about 20 years in the framework of Petri nets. Starting from a different point of view, namely evaluation of recursive equations represented as acyclic graphs and construction of canonical derivation sequences for graph grammars, several aspects of parallelism and concurrency have also been studied in the framework of graph grammars since about 10 years. The corresponding theoretical results obtained in the algebraic approach to graph grammars have been applied to problems of concurrency in data base systems and computer processed control.

More recently these concepts of concurrency were also studied in the framework of Node-Label-Controlled (NLC) grammars leading to a new phenomenon of "concurrency" which is not recaptured in the algebraic approach up to now nor in Petri nets. Concurrency of firing of transitions in Petri nets, however, corresponds exactly to parallelism of graph grammar derivations, once the graph grammars are specialized to model Petri nets. On the other hand graph grammars seem to be more flexible than Petri nets concerning dynamical changes of the structure and not only of the labelling. Since Petri nets have been used already with good success for the modeling of concurrent systems graph grammars should be even more suitable for this purpose. It remains the problem how far graph grammars are suitable to model data structures.
Data structures, especially abstract data types, are not only a main concept within programming languages but have turned out to be a fundamental tool for the modeling of software systems. Especially algebraic specification techniques are very useful for this purpose, at least concerning data types and software systems without concurrency concepts. At the present state of development graph grammars seem to be more suitable to cover concurrency concepts than algebraic specifications, but less suitable concerning data structures. This means from the graph grammar point of view that it is desirable to simulate Petri nets as well as algebraic specifications within graph grammars to cover concurrency as well as data structure aspects.

There are already at least two approaches simulating Petri nets within graph grammars which we are going to discuss in this paper but there is no approach up to now simulating algebraic specifications within graph grammars. In this paper we are going to propose two simulations, the simulation of the operational semantics of algebraic specifications within graph grammars and also that of the initial algebra semantics, more precisely the simulation of the canonical term algebra of algebraic specifications within graph grammars. We hope that this is a useful step in order to extend algebraic specification techniques to graph grammars. From the algebraic specification point of view it seems also advisable to simulate graph grammars, especially the concurrency concepts in graph grammars, within algebraic specifications. But this is another topic not to be discussed in this paper.

Finally we propose some new concepts within graph grammars in order to model different aspects of concurrency and relate these concepts to each other and to those studied already in the literature.

The ideas introduced above are organized within the following sections of this paper:
1. General remarks on graph grammar developments including aspects of concurrency.
2. Basic results in the theory of graph transformations.
3. Basic formalisms used in the theory of graph transformations.
4. Triangle Simulations.
5. Simulations of algebraic specifications within graph grammars.
6. Towards a theory of concurrency in graph grammars.

A final remark:
We will show in section 1 that the theory of graph grammars has been developed mainly in two directions: A theory of graph languages following mainly the lines of formal language theory in the string case and a theory of graph transformations developed from the various fields of applications of graph grammars in Computer Science and Biology. This paper can also be considered as an overview of the second case, graph transformation theory, at least as far as the algebraic approach, NLC (Node-Label-Controlled) and the GH (Generalized Handle)-approach are concerned. The new concepts proposed in this paper are also given in informal style, the technical presentations of those ideas will be given in subsequent papers.