Chapter II

Rule-Based Transformation of Graphs and the Product Type

Renate Klempien-Hinrichs, University of Bremen, Germany

Hans-Jörg Kreowski, University of Bremen, Germany

Sabine Kuske, University of Bremen, Germany

Abstract

This chapter presents rule-based graph transformation as a framework for modeling data-processing systems. It recalls the structuring concept of graph transformation units which allows for transforming graphs in a rule-based, modularized, and controlled way. In order to get a flexible typing mechanism and a high degree of parallelism, this structuring concept is extended to the product of transformation units. Moreover, it is demonstrated how the product type can be used to transform graph transformation units.
The authors advocate rule-based graph transformation for all applications where data, knowledge, and information can be modeled as graphs and their transformation can be specified by means of rules in an adequate way.

Introduction

The area of graph transformation brings together the concepts of rules and graphs with various methods from the theory of formal languages and from the theory of concurrency, and with a spectrum of applications (Figure 1).

Graphs are important structures in computer science and beyond to represent complex system states, networks, and all kinds of diagrams. The application of rules provides graphs with a dynamic dimension yielding a rich methodology of rule-based graph transformation. The three volumes of the *Handbook of Graph Grammars and Computing by Graph Transformation* give a good overview of the state of the art in theory and practice of graph transformation (Rozenberg, 1997; Ehrig, Engels, Kreowski & Rozenberg, 1999; Ehrig, Kreowski, Montanari & Rozenberg, 1999).

Although one encounters quite a large number of different approaches to graph transformation in the literature, nearly all of them contain five basic features.

- **Graphs** to represent complex relations among items in an intuitive but mathematically well-understood way.
- **Rules** to describe possible changes and updates of graphs in a concise way.
- **Rule applications** to perform the possible changes and updates on graphs explicitly as they are embodied in the rules.
- **Graph class expressions** to specify special classes of graphs to be used as initial as well as terminal graphs.
- **Control conditions** to regulate the applications of rules such that the inherent non-determinism of rule application can be cut down.

*Figure 1. Main ingredients of graph transformation*