Graph Transformation Units with Interleaving Semantics

Hans-Jörg Kreowski and Sabine Kuske
University of Bremen, Department of Mathematics and Computer Science, Bremen, Germany

Keywords: Graph transformation, Transformation unit, Structuring, Formal methods, Interleaving semantics

Abstract. The aim of the paper is to introduce the notion of a transformation unit together with its interleaving semantics and to study it as a means of constructing large graph transformation systems from small ones in a structured and systematic way. A transformation unit comprises a set of rules, descriptions of initial and terminal graphs, and a control condition. Moreover, it may import other transformation units for structuring purposes. Its semantics is a binary relation between initial and terminal graphs which is given by interleaving sequences. As a generalization of ordinary derivations, an interleaving sequence consists of direct derivation steps interleaved with calls of imported transformation units. It must obey the control condition and may be seen as a kind of structured derivation. The introduced framework is independent of a particular graph transformation approach and, therefore, it may enhance the usefulness of graph transformations in many contexts.

1. Introduction

The significance of graphs and rules in many areas of computer science is evident: On the one hand, graphs constitute appropriate means for the description of complex relationships between objects. Trees, forests, Petri nets, circuit diagrams, finite automata, flow charts, data flow graphs, and entity-relationship diagrams are
some typical examples. On the other hand, rules are used to describe “permitted”
manipulations on objects as, for example, in the areas of functional and logic
programming, formal languages, algebraic specification, theorem proving, and
rule-based systems.

The intention of bringing graphs and rules together – motivated by several
application areas – has led to the theory of graph grammars and graph trans-
formation (see the three volumes of the Handbook of Graph Grammars and
Computing by Graph Transformation [Roz97, EEK99, EKM99] for a survey).
A wide spectrum of approaches exists within this theory and some of them are
implemented (see, for example, PROGRES [Sch91a, Sch91b], GraphEd [Him91],
Dactl [GKS91], and AGG [LöB93, TaB94]).

With the aim of enhancing the usefulness of graph transformation, we propose
a new approach-independent structuring method for building up large systems of
graph transformation rules from small pieces. The method is based on the notion
of a transformation unit and its interleaving semantics. A transformation unit is
allowed to use other units such that a system of graph transformation rules can
be structured hierarchically and existing transformation units can be re-used. The
transformation unit is a basic concept of the new graph and rule centered language
GRACE that is being developed by researchers from Berlin, Bremen, Erlangen,
München, Oldenburg, and Paderborn (see also [AEH99, Sch96]). Nevertheless,
the notion is meaningful in its own right because – independently of GRACE – it can be employed as a structuring principle in most graph transformation
approaches one encounters in the literature where graph transformation is often
called graph rewriting.

The paper is organized as follows. Section 2 introduces the notion of a trans-
formation unit together with its interleaving semantics. In Section 3, the concepts
of a transformation unit are illustrated with an example. Section 4 presents how
some operations on binary relations can be modelled by suitable operations on
transformation units. In Section 5, some normal forms of transformation units
are considered. The paper ends with some concluding remarks. To avoid wrong
expectations, we would like to point out that the goal of the paper is to shed
some light on the usefulness of the introduced structuring method rather than to
come up with deep theory.

A short draft version of this paper without proofs is published as [KrK96].

2. Transformation Units with Interleaving Semantics

The key operation in graph transformation approaches is the direct derivation be-
ing the transformation of a graph into a graph by applying a rule. In other words,
each rule yields a binary relation on graphs. Hence, each set of rules specifies a
binary relation on graphs by iterated rule applications. This derivation process is
highly non-deterministic in general and runs on arbitrary graphs which is both
not always desirable. For example, if one wants to generate graph languages, one
may start in a particular axiom and end with certain terminal objects only. Or
if a more functional behaviour is required, one may prefer to control the deriva-
tion process and to cut down its non-determinism. The latter can be achieved
by control mechanisms for the derivation process like application conditions or
programmed graph transformation (see, e.g., [Bun79, Nag79, EhH86, KrR90,
MaW91, Sch91a, Scz91, Kre93, LiM93, HHT96, MaW96], cf. also [DaP89] for
regulation concepts in string grammars) and the former by the use of graph class
expressions that specify subclasses of graphs. Moreover, in practical cases, one