Co-transformation of Graphs and Type Graphs
with Application to Model Co-evolution

Gabriele Taentzer	extsuperscript{1}, Florian Mantz	extsuperscript{2}, and Yngve Lamo	extsuperscript{2}

	extsuperscript{1} Philipps-Universität Marburg, Germany
taentzer@informatik.uni-marburg.de
	extsuperscript{2} Bergen University College, Norway
\{fma,yla\}@hib.no

Abstract. Meta-modeling has become the key technology to define domain-specific modeling languages in model-driven engineering. Since domain-specific modeling languages often change quite frequently, concepts are needed for the coordinated evolution of their meta-models as well as of their models, and possibly other related artifacts. In this paper, we present a new approach to the co-transformation of graphs and type graphs and show how it can be applied to model co-evolution. This means that models are specified as graphs while model relations, especially type-instance relations, are defined by graph morphisms specifying type conformance of models to their meta-models. Hence, meta-model evolution and accompanying model migrations are formally defined by co-transformations of instance and type graphs. In our approach, we clarify the type conformance of co-transformations, the completeness of instance graph transformations wrt. their type graph modifications, and the reflection of type graph transformations by instance graph transformations. Finally, we discuss strategies for automatically deducing instance graph transformation rules from given type graph transformations.

Keywords: meta-model evolution, model migration, graph transformation.

1 Introduction

Model-driven engineering (MDE) is a software engineering discipline that uses models as the primary artifacts throughout software development processes and adopt model transformation both for their optimization as well as for model and code generation. Models in MDE describe application-specific system design which is automatically translated into code. A commonly used technique to define modeling languages is meta-modeling. In contrast to traditional software development where programming languages rarely change, domain-specific modeling languages, and therefore meta-models, often change frequently: modeling language elements may be, e.g., renamed, extended by additional attributes,
or refined by a hierarchy of sub-elements. The evolution of a meta-model requires the consistent migration of its models (See Fig. 1) which is a considerable research challenge in MDE [18].

Since graphs and graph transformations are conceptually close to models and model transformations, we consider instance and type graph co-transformations as a suitable approach to the formalization of model co-evolution. In contrast to the traditional double pushout (DPO) approach in [4] where deletion of graph parts is performed before the creation of new graph elements, we use the dual approach where creation is done before deletion. The dual approach has been introduced as co-span DPO-approach in [5] which also shows equivalence of both approaches. We choose this variant of graph transformations because they allow better synchronization of deletion and creation actions than the usual DPO approach, since the intermediate graphs contain both elements to be deleted and those to be added. Since we do not want to restrict ourselves to a specific kind of graph and graph morphism in this paper, the theoretical concepts and results are formulated at the level of (weak) adhesive categories (see e.g. [9, 3]).

On this basis, we characterize co-transformations that lead to type conforming result graphs. Considering a given type graph transformation, a related instance graph transformation has to be complete, i.e., has to incorporate the whole instance graph, and has to reflect at least deletion actions of the type graph level. Furthermore, we present a strategy to deduce instance graph transformation rules from given type graph transformations.

The new graph transformation concepts and results presented are clearly motivated by our wish to develop an adequate formalization of model and meta-model co-evolution. Among existing approaches [17, 20, 28, 14, 16] only [16] contains fundamental results about the well-formedness of model and meta-model co-evolutions. However, the co-evolution approach presented in [16] is more restricted than ours.

The rest of the paper is organized as follows: in Section 2 we introduce a simple model co-evolution scenario. The definition of co-span transformation is recalled in Section 3 which prepares for the formalization of model and meta-model co-evolution presented in Section 4. We conclude this paper with a consideration of related work in Section 5 and final remarks in Section 6.