

Felix Klar: Efficient and Compatible Bidirectional Formal Language Translators based on Extended Triple Graph Grammars

Promotion: Technische Universität Darmstadt, Fachbereich Elektrotechnik und Informationstechnik

Erstgutachter: Prof. Dr. Andy Schürr, Technische Universität Darmstadt

Zweitgutachter: Prof. Dr. Albert Zündorf, Universität Kassel

Datum der Prüfung: 10. Februar 2012

Veröffentlichung: <http://tuprints.ulb.tu-darmstadt.de/2894/>

Kurzfassung:

In the context of model-driven engineering, models play an important role in everyday life. Models are used to abstract from certain subjects and to describe artifacts and procedures. In software engineering, a system under development is often modeled on different levels of abstraction and from multiple perspectives which results in plenty of models. Moreover, the resulting models depend on each other and the need for automatically translating between related models arises in order to reduce costs, errors, and laborious manual work and to speed-up development processes.

The model-driven engineering approach proposes model transformations as a key concept of model-based development which allows to automatically refine and transform models or translate between related models. Especially, bidirectional translators are often required which are able to automatically keep related models in a consistent state. The goal of bidirectional model transformations, which allow to execute transformations defined between a source and target model in both directions, is to assist in such situations. To be able to specify (bidirectional) model transformations the need for (bidirectional) model transformation languages arises.

Triple graph grammars (TGGs) are a formally founded bidirectional transformation language based on graph transformations with precisely defined semantics. A TGG specification describes correspondence relationships between two languages and consists of a set of productions that declaratively specify the simultaneous evolution of both related models. The main advantage of triple graph grammars is the possibility to automatically derive bidirectionally working forward and backward translators from a TGG specification that fulfill the fundamental properties efficiency and compatibility.

The grand challenge is to build translators that are efficient on the one hand and are compatible with respect to the TGG specification on the other hand. Compatibility means that translators are correct and complete with respect to the specification, i.e., pairs of models are in a consistent state after the translation operation and valid models are able to be translated into corresponding models. Moreover, the overall ex-

pressiveness of the triple graph grammar language has to be increased in order to create usable transformation specifications. But, it has to be ensured that derived translators still fulfill the fundamental properties.

In this thesis, the expressiveness of triple graph grammars is increased by supporting negative application conditions (NACs) that allow to restrict the applicability of transformation rules, which is required in certain cases. In addition, we accept the challenge of providing an efficiently working translation algorithm that still fulfills the properties correctness and completeness. We extend the expressiveness of triple graph grammars by a precisely defined class of NACs together with a new translation algorithm such that for the first time the fundamental properties of TGG-based translators are still satisfied. The resulting translators nevertheless have a polynomial runtime complexity and, therefore, can be considered efficient. Moreover, they are compatible with their triple graph grammar, which makes these translators usable in practice. In conclusion, the extended triple graph grammar formalism is applicable in real world scenarios, where model transformations are bidirectionally executed to keep related models in a consistent state.