Abstract

Model transformations play a key role in automated software development processes, where the modern trend is directed towards specification of software with an abstract model and its step-wise transformation into code. It is important that the code meets the initial specification. Therefore, the question for a model transformation is whether the transformed model fulfills the behavioural properties of the initial model.

In this thesis, a method is presented for proving that a specified model transformation is semantically correct in the sense that it preserves all behavioural properties of a source model. The method is introduced within the model-driven architecture approach, where syntax of a modelling language is defined by a meta-model. A meta-model is specified by means of object-oriented approaches, which based on the diagrams that are basically graphs, sometimes attributed with textual information. This allows a meta-model to be considered as a graph. Then, the model transformation is defined with the triple graph grammar technique.

We deal with languages, the behavioural semantics of which can be formally specified by means of graph transformation systems too. Therefore, the graph transformations are used twice in this thesis: for model transformation and for behavioural semantics specification. An application of a graph transformation system to a graph that stands for a model results a Labelled Transition System (LTS) generation. An LTS represents a set of all possible transition sequences, which is supposed to describe behaviour of the system.

A temporal logic called ACTL (an Action-based version of CTL*), is used to specify the behavioural properties of a model on an LTS. Full preservation of behavioural properties means that an LTS generated for *any* source model and the LTS generated for the corresponding target model cannot be distinguished by the truth value of ACTL formulas. When the same ACTL properties hold for two LTSs, then the LTSs are called ACTL equivalent.

The method introduced in this thesis is based on establishing an equivalence relation over LTSs called weak bisimulation, which is a standard concept from concurrency theory, normally used to compare executions and to decide whether they are indeed equivalent or not. In this thesis, it is shown that weak bisimulation on LTSs implies ACTL equivalence. Then, showing the behaviour preservation during model transformation requires establishing a weak bisimulation on the LTSs of source and target models.

As a summary, the method provides the guidelines how to specify the modelling languages and model transformation in order it could be possible to show that the specified model transformation is semantically correct. The method also explains how to define an equivalence relation over LTSs and to carry out the proof that the

defined equivalence relation is a weak bisimulation. Additionally, we explain how the behavioural properties of a source model can be interpreted for a target language by usage of the triple graph grammar transformation and the weak bisimulation.

In addition, the method is validated by a case study, where we define a model transformation between two modelling languages: Calculus of Communication Systems (CCS) and Petri nets. The main goal of the case study is to show how to proof that the defined model transformation is semantics preserving and to illustrate the specification of sample behavioural properties for the CCS and their interpretation for the Petri nets language.