Analysis Techniques for Graph Transformation Systems

Gabriele Taentzer

Philipps-Universität Marburg, Germany
taentzer@mathematik.uni-marburg.de

Abstract. In software and system development, verification and validation means investigating artifacts in order to check if they fulfill the intended purposes. Graph transformation systems can be considered as precise models of computation where states are represented by graphs and state changes by rule-based graph transformations.
This talk gives an overview of selected analysis techniques being available. For typed graph transformation systems, all graphs being transformation results automatically conform to the given type graph. Hence, they are type correct. Simple constraints such as multiplicities, are often taken into account as well. If a graph transformation system computes an operation, functional behavior may be intended. This means that, for each input graph, exactly one output graph is computed. Functional behavior is ensured if the following two properties hold: confluence (also called determinism) and termination. Confluence means that, given an initial graph, all transformations starting at this graph yield the same result graph. The check for confluence can be based on critical pairs specifying potential conflicts and dependencies. A graph transformation system terminates if finite transformation sequences are possible only. We discuss several termination criteria for graph transformation systems.
To check interesting properties of graph transformation systems, two techniques are considered: Invariants, i.e., graph-specific properties, may be formulated in first-order logic. A graph transformation system fulfills an invariant if its start graph(s) fulfill them and all rules preserve them. If a rule causes inconsistent result graphs, it is augmented by additional pre-conditions deduced from violated invariants. While invariants are formulated for graphs, temporal properties, e.g., safety or reachability properties, are checked for graph transformation systems. By model checking techniques, all possible transformations are systematically considered and counter examples are reported.