# Abstraction and Abstraction Refinement in the Verification of Graph Transformation Systems

**Vitaly Kozyura**

Graph transformation systems (GTSs) form a natural and convenient specification language which is used for modelling concurrent and distributed systems with dynamic topologies. These can be, for example, network and Internet protocols, mobile processes with dynamic behavior and dynamic pointer structures in programming languages. All this, together with the possibility to visualize and explain system behavior using graphical methods, makes GTSs a well-suited formalism for the specification of complex dynamic distributed systems.

Under these circumstances the problem of checking whether a certain property of GTSs holds – the verification problem – is considered to be a very important question. Unfortunately the verification of GTSs is in general undecidable because of the Turing-completeness of GTSs. In the last few years a technique for analysing GTSs based on approximation by Petri graphs has been developed. Petri graphs are Petri nets having additional graph structure.

In this work we focus on the verification techniques based on counterexample-guided abstraction refinement (CEGAR approach). It starts with a coarse initial over-approximation of a system and an obtained counterexample. If the counterexample is spurious then one starts a refinement procedure of the approximation, based on the structure of the counterexample. The CEGAR approach has proved to be very successful for the verification of systems based on their over-approximations.

This thesis investigates a counterexample-guided abstraction refinement approach for systems modelled with GTSs. Starting with a given spurious counterexample, we describe here how to construct a more exact approximation (by separating merged nodes) for which this counterexamples disappears. This procedure can be performed repeatedly for any number of spurious counterexamples. Furthermore, an incremental coverability approach for Petri nets is developed, which allows one to speed-up the construction of over-approximations of GTSs.

A well-known approach is to extend a modelling language with the possibility of describing attributes as values of some data types. The approximation-based verification technique, including a counterexample-guided abstraction refinement, is hence also generalized in this work to attributed GTSs (AGTSs), where the attributes are abstracted in the framework of abstract interpretation.

In the practical part, a verification tool Augur 2 is developed, which supports the whole verification process for GTSs and AGTSs. A number of case studies (both attributed and non-attributed GTSs) were successfully solved with Augur 2.