GrGen.NET: A Fast, Expressive, and General Purpose Graph Rewrite Tool

Rubino Geiß and Moritz Kroll

Universität Karlsruhe (TH), 76131 Karlsruhe, Germany
rubino@ipd.info.uni-karlsruhe.de
http://www.grgen.net

Introduction GrGen.NET is a graph rewrite tool enabling elegant and convenient development of graph transformation applications with comparable performance to manually developed ones. GrGen.NET compiles declarative specifications of graph meta models, patterns, and rewrite rules into .NET modules. The entire functionality (meta-model, matching, rewriting, elementary graph operations) is accessible through a convenient API (called libGr) enabling easy integration of GrGen.NET into custom applications. Meta-model and rule languages have formal semantics based on a new combination of category theory and denotational semantics [1].

The general purpose graph rewrite tool GrGen.NET is a descendant of GrGen [2], initially developed for transformations in compiler construction [3]. GrGen.NET is published under LGPL along with a user manual.

Meta Model Language GrGen.NET uses typed and directed multigraphs with multiple inheritance on node and edge types. These types can be equipped with typed attributes (primitive types, enums and C#-objects). The type hierarchies in GrGen.NET are similar to those in common OO-languages.

Pattern and Rewrite Language A set of rewrite rules can be specified referring to graph meta models. The pattern matcher is able to perform plain isomorphic subgraph matching (injective mapping) as well as homomorphic matching for a selectable set of nodes and edges. The language has special support for typical use cases like finding induced subgraphs and exact patterns (i.e., all the incident edges in the host graph are specified in the pattern) by pattern modifiers. Matches can further be restricted by arithmetic and logical conditions on the attributes and types (including powerful instanceof-like type expressions). Nested negative application conditions—i.e., subpatterns whose existence forbids the matching—are supported, too.

The task of rewriting is internally implemented as an extension to SPO semantics. However, the user is able to specify rules in well-known DPO semantics, too. The rewrite language offers an extensive set of useful graph operations, including recalculation of node and edge attributes and retyping (a more general version of type casts) of nodes and edges. In addition to rules in algebraic style, extended graph rewrite sequences (XGRS) and emit text can be applied in the rewrite part of a rule. This way we successfully performed MOF model transformation with automatic generation of XMI files[4].

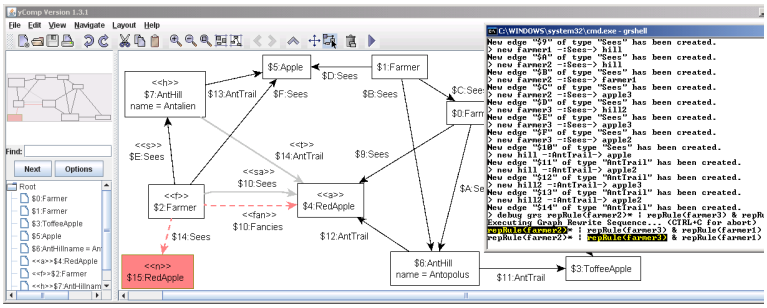

Fig. 1. Screenshot of GrShell and yComp in debug mode (colors optimised for print)

Usability Because GrGen.NET works on pure text files, handling of large meta models and rule sets as well as the integration with code generation tools is easy. A .NET shell application (GrShell) is shipped, capable of interactive and batched execution of the graph rewrite functionalities. This includes commands for creation, deletion, input, and output of graphs, nodes, and edges as well as application of rewrite rules. Several rules can be composed with logical and iterative sequence control to an XGRS, resulting in its Turing-completeness. Moreover, XGRS may contain nested transactions.

By accessing the match and rewrite facility through libGr, instead of using GrShell, custom algorithmic rule applications are possible. Graphical and stepwise debugging integrated into the GrShell complements the feature highlights of GrGen.NET (see Figure 1).

Performance According to all available benchmarks GrGen.NET is among the fastest graph rewrite tools, with a feature set being superior to most of the other tools. In contrast to Fujaba (competing in speed with our tool) GrGen.NET is fully automatic: The user does not have to (partly) specify how the tool should search for a pattern graph. To automatically optimize the performance for a given (class of) host graphs, GrGen.NET implements the search plan based approach to graph pattern matching [5] as the first tool.

References

1. Geiß, R.: Graphersetzung mit Anwendungen im Übersetzerbau. PhD thesis, Universität Karlsruhe (2007) approved, to appear.
2. Geiß, R., Batz, G.V., Grund, D., Hack, S., Szalkowski, A.: GrGen: A Fast SPO-Based Graph Rewriting Tool. In Corradini, A., Ehrig, H., Montanari, U., Ribeiro, L., Rozenberg, G., eds.: ICGT 2006. Volume 4178 of LNCS., Springer (2006) 383–397
3. Schösser, A., Geiß, R.: Graph Rewriting for Hardware Dependent Program Optimizations. In: in this volume. (2007)
4. Gelhausen, T., Derre, B., Geiß, R.: Customizing GrGen.NET for Model Transformation. GraMoT 2008, accepted (2008)
5. Batz, G.V., Kroll, M., Geiß, R.: A First Experimental Evaluation of Search Plan Driven Graph Pattern Matching. In: in this volume. (2007)