Model Transformation with Context-free Triple Graph Grammars and Application Conditions

Keywords: Triple Graph Grammars, NCE Graph Grammars, Model Transformation, Model-driven Development.

Abstract:

1 INTRODUCTION

2 RELATED WORKS

In this section, we offer a short literary review on the graph grammar and triple graph grammar approaches that are more relevant to our work. We focus, therefore, on the context-free node label replacement approach for graph grammars, although, there is a myriad of different alternatives to it, for example, the algebraic approach (Ehrig et al., 1999). We refer to context-free grammars, inspired by the use of such classification for string grammars, in a relaxed way without any compromise to any definition.

In the node label replacement context-free formalisms stand out the *node label controlled graph grammar* (NLC) and its successor *graph grammar with neighborhood-controlled embedding* (NCE). NLC is based on the replacement of one vertex by a graph, governed by embedding rules written in terms of the vertex's label (Rozenberg and Welzl, 1986). For various classes of these grammars, there exist polynomial-time top-down and bottom-up parsing algorithms (Flasiński, 1993; Flasiński and Flasińska, 2014; Rozenberg and Welzl, 1986; Wanke, 1991). The recognition complexity and generation power of such grammars have also been analyzed (Flasiński, 1998; Kim, 2012). NCE occurs in several formulations, including a context-sensitive one, but here we focus on the context-free formulation, where one vertex is replaced by a graph, and the embedding rules are written in terms of the vertex's neighbors (Janssens and Rozenberg, 1982; Skodinis and Wanke, 1998). For some classes of these grammars, polynomial-time bottom-up parsing algorithms and automaton formalisms were proposed and analyzed (Kim, 2001; Brandenburg and Skodinis, 2005). In special, one of these classes is the *boundary graph grammar with neighborhood-controlled embedding* (BNCE), that is used in our approach for model transformation.

Regarding TGG (Schürr, 1994), a 20 years review of the realm is put forward by Anjorin et al. (Anjorin et al., 2016). In special, advances are made in the direction of expressiveness with the introduction of application conditions (Klar et al., 2010) and of modularization (Anjorin et al., 2014). Furthermore, in the algebraic approach for graph grammars, we have found proposals that introduce inheritance (Bardohl et al., 2004; Hermann et al., 2008) and variables (Hoffmann, 2005) to the formalisms. Nevertheless, we do not know any approach that introduces non-terminal symbols to TGG with the purpose of gaining expressiveness or usability. In this sense, our proposal brings something new to the current state-of-the-art.

3 CONTEXT-FREE TGG

3.1 NCE Graph Grammars

3.2 NCE Triple Graph Grammars

4 CONTEXT-FREE TGG WITH APPLICATION CONDITIONS

5 MODEL TRANSFORMATION WITH NCE PAC TGG

6 EVALUATION

7 CONCLUSION

REFERENCES

Anjorin, A., Leblebici, E., and Schürr, A. (2016). 20 years of triple graph grammars: A roadmap for future research. *Electronic Communications of the EASST*, 73.

Anjorin, A., Saller, K., Lochau, M., and Schürr, A. (2014). Modularizing triple graph grammars using rule refinement. In *International Conference on Fundamental Approaches to Software Engineering*, pages 340–354. Springer.

Bardohl, R., Ehrig, H., De Lara, J., and Taentzer, G. (2004). Integrating meta-modelling aspects with graph transformation for efficient visual language definition and model manipulation. In *International Conference on Fundamental Approaches to Software Engineering*, pages 214–228. Springer.

Brandenburg, F.-J. and Skodinis, K. (2005). Finite graph automata for linear and boundary graph languages. *Theoretical Computer Science*, 332(1-3):199–232.

Ehrig, H., Rozenberg, G., Kreowski, H.-J., and Montanari, U. (1999). *Handbook of graph grammars and computing by graph transformation*, volume 3. World Scientific.

Flasiński, M. (1993). On the parsing of deterministic graph languages for syntactic pattern recognition. *Pattern Recognition*, 26(1):1–16.

Flasiński, M. (1998). Power properties of nlc graph grammars with a polynomial membership problem. *Theoretical Computer Science*, 201(1-2):189–231.

Flasiński, M. and Flasińska, Z. (2014). Characteristics of bottom-up parsable ednlc graph languages for syntactic pattern recognition. In *International Conference on Computer Vision and Graphics*, pages 195–202. Springer.

Hermann, F., Ehrig, H., and Taentzer, G. (2008). A typed attributed graph grammar with inheritance for the abstract syntax of uml class and sequence diagrams. *Electronic Notes in Theoretical Computer Science*, 211:261–269.

Hoffmann, B. (2005). Graph transformation with variables. In *Formal Methods in Software and Systems Modeling*, pages 101–115. Springer.

Janssens, D. and Rozenberg, G. (1982). Graph grammars with neighbourhood-controlled embedding. *Theoretical Computer Science*, 21(1):55–74.

Kim, C. (2001). Efficient recognition algorithms for boundary and linear ence graph languages. *Acta informatica*, 37(9):619–632.

Kim, C. (2012). On the structure of linear apex nlc graph grammars. *Theoretical Computer Science*, 438:28–33.

Klar, F., Lauder, M., Königs, A., and Schürr, A. (2010). Extended triple graph grammars with efficient and compatible graph translators. In *Graph transformations and model-driven engineering*, pages 141–174. Springer.

Rozenberg, G. and Welzl, E. (1986). Boundary nlc graph grammarsbasic definitions, normal forms, and complexity. *Information and Control*, 69(1-3):136–167.

Schürr, A. (1994). Specification of graph translators with triple graph grammars. In *International Workshop*

on Graph-Theoretic Concepts in Computer Science, pages 151–163. Springer.

Skodinis, K. and Wanke, E. (1998). Neighborhood-preserving node replacements. In *International Workshop on Theory and Application of Graph Transformations*, pages 45–58. Springer.

Wanke, E. (1991). Algorithms for graph problems on bnlc structured graphs. *Information and Computation*, 94(1):93–122.