Model Transformation with Triple Graph Grammars and Non-terminal Symbols

William da Silva, Max Bureck, Ina Schieferdecker, and Christian Hein

Fraunhofer Fokus, Berlin, Germany
william.bombardelli.da.silva@fokus.fraunhofer.de
Technische Universität Berlin, Berlin, Germany

November 2, 2018

Organization

1 Introduction

2 Triple Graph Grammars with Non-terminal Symbols

3 Evaluation

4 Conclusion

5 References

Introduction

1 Introduction

2 Triple Graph Grammars with Non-terminal Symbols

3 Evaluation

4 Conclusion

5 References

Background

- Model-driven software development as a technique to enhance quality of software

Background

- Model-driven software development as a technique to enhance quality of software
- Models as formal specifications of safety-critical systems

Background

- Model-driven software development as a technique to enhance quality of software
- Models as formal specifications of safety-critical systems
- Transformation between models (e.g. from a formal specification to high-level source-code and vice-versa)

Background

- Model-driven software development as a technique to enhance quality of software
- Models as formal specifications of safety-critical systems
- Transformation between models (e.g. from a formal specification to high-level source-code and vice-versa)
- **Goal:** Comprehensible and reliable transformations
  - Efficient representation of abstract concepts
  - Small size

The Model Transformation Problem

The Triple Graph Grammar Approach

- Models are graphs

The Triple Graph Grammar Approach

- Models are graphs
- Two correctly-transformed graphs G and T are in a triple graph $G \leftarrow C \rightarrow T$

The Triple Graph Grammar Approach

- Models are graphs
- Two correctly-transformed graphs G and T are in a triple graph $G \leftarrow C \rightarrow T$
- A triple graph grammar TGG is a generator of a set of triple graphs $L(TGG)$

The Triple Graph Grammar Approach

- Models are graphs
- Two correctly-transformed graphs G and T are in a triple graph $G \leftarrow C \rightarrow T$
- A triple graph grammar TGG is a generator of a set of triple graphs $L(TGG)$
- The correctly-transformed relation $\sim$ between graphs is described in terms of a triple graph grammar TGG
  - $G \sim T$ iff $(G \leftarrow C \rightarrow T) \in L(TGG)$

Triple Graph Grammars with Non-terminal Symbols

1 Introduction

2 Triple Graph Grammars with Non-terminal Symbols

3 Evaluation

4 Conclusion

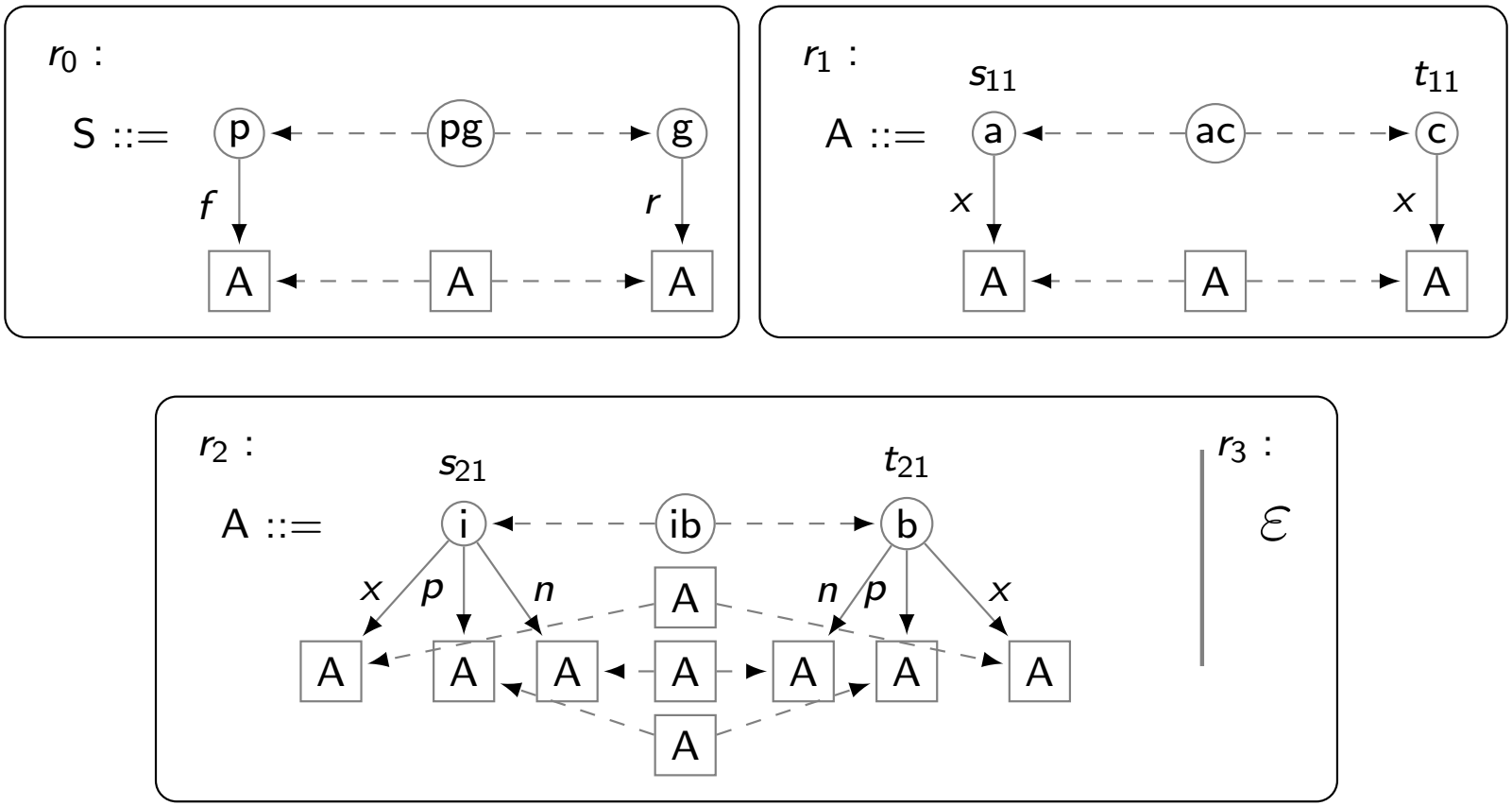
5 References

Our contribution – NCE TGG

- New formalism: NCE TGG
  - *Graph Grammar with Neighborhood-controlled Embedding* (NCE) [Janssens and Rozenberg(1982)]
  - *Triple Graph Grammar* (TGG) [Schürr(1994)]
- Non-terminal symbols
- Context-free

NCE TGG – An example

- *Pseudocode* to *Controlflow*



Evaluation

1 Introduction

2 Triple Graph Grammars with Non-terminal Symbols

3 Evaluation

4 Conclusion

5 References

Usability Evaluation

	Standard TGG		BNCE TGG	
Transformation	Rules	Elements	Rules	Elements
Pseudocode2Controlflow	45	1061	**7**	**185**
BTree2XBTree	**4**	**50**	5	80
Star2Wheel	-	-	**6**	**89**
Class2Database	**5**	**80**	-	-

Table: Results of the usability evaluation of the BNCE TGG formalism in comparison with the standard TGG for the model transformation problem

Conclusion

1 Introduction

2 Triple Graph Grammars with Non-terminal Symbols

3 Evaluation

4 Conclusion

5 References

Conclusion

- New context-free TGG formalism
  - Used to specify model transformations
  - Outperforms standard TGG in 2 evaluated cases
  - Special potential for code-generation
  - Cannot model important transformations (e.g. Class Diagrams)

Conclusion

- New context-free TGG formalism
  - Used to specify model transformations
  - Outperforms standard TGG in 2 evaluated cases
  - Special potential for code-generation
  - Cannot model important transformations (e.g. Class Diagrams)
- Future Work:
  - Application conditions: Positive experimental results
  - Broader evaluation including empirical assessment with engineers and performance reports
  - Model synchronization

References

1 Introduction

2 Triple Graph Grammars with Non-terminal Symbols

3 Evaluation

4 Conclusion

5 References

References

Dirk Janssens and Grzegorz Rozenberg.

Graph grammars with neighbourhood-controlled embedding.
Theoretical Computer Science, 21(1):55–74, 1982.

Andy Schürr.

Specification of graph translators with triple graph grammars.
In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 151–163. Springer, 1994.