

**Proposal for Master Thesis 01**

20 February 2018

**William Bombardelli da Silva****On the use of non-terminal symbols in graph grammars****Introduction**

With the objective to enhance software quality, much effort from academy and industry have been made to propose software modeling techniques and tools that guide software developers and engineers through more effective software engineering processes. These techniques utilize very often graphical modeling languages (e.g. UML) to specify computational systems. These modeling languages can be viewed as graph languages, which in turn can be described by graph grammars.

In this context, two common problems are the parsing of graphs in terms of graph grammars and the transformation of graphs from different languages. There is already some research that investigates such problems and proposes parsing and transformation algorithms based on graph grammars and specially on Triple Graph Grammars (TGGs). But as far as we know there is not much investigation on the use of non-terminal symbols in these grammars.

So the goal of this thesis is to study the concept of non-terminal symbols in graph grammars, including the syntactical and semantical implications of its use; propose a classification of graph grammars inspired by, but not limited to, the Chomsky hierarchy of formal languages; propose and analyze parsing algorithms; study non-terminal symbols in Triple Graph Grammars in special applied to the problem of bi-directional transformation; and, finally, measure empirically the advantages of the use of non-terminal symbols in these grammars, specially in terms of the amount of rules and the algorithms efficiency.

**Motivation.** The use of the concept of non-terminal symbols in (triple) graph grammars may increase significantly its usability in the parsing and transformation problems, as it reduces the minimum amount of rules necessary to describe a language. Moreover, the classification of graph grammars in terms of the use of non-terminal symbols in the production rules may serve as a strategy to restrict grammars to a treatable set for a specific context, like it is done in the compilation of programming languages with the LL languages, what in fact allows the creation of efficient parsing algorithms.

**Key-words.** Graph Grammars, Triple Graph Grammars, Parsing, Bi-directional Graph Transformation, Model-driven Engineering, Software Engineering.

**Methodology**

This research will be carried out for six months and will be divided into several phases, presented in Table 1. First, we intend to use graph theory, formal languages and abstract algebra concepts to define abstract syntax and semantics of graph grammars with non-terminal symbols. This includes, possibly, the translation of the concepts of symbols, strings, alphabets, concatenation and the Kleene-closure into graph theory.

Second, we plan to define classes of graph grammars in terms of the form of production rules and the occurrence of non-terminal symbols, create specialized parsing algorithms for some of these classes and analyze, if possible, their complexity as well as the classes' expressiveness and their relation with other current graph grammar techniques. After that, we expect to study the use of non-terminal symbols in TGGs and implement transformation algorithms based on the parsing algorithms.

Finally, we aim to evaluate the practical use of non-terminal symbols in some examples comparing the amount of rules used in the grammars and the efficiency of parsing and transformation algorithms in terms of run-time.

**Threats.** One possible obstacle for this work is the creation of a clear formalism for non-terminal symbols that is in conformity to the concepts of graph and formal language theories. The creation of parsing algorithms might be challenging as well, even though there exist well-known parsing algorithms for strings, translating their concepts into our definitions of graph grammar might be difficult. Moreover, we have the expectation that the problem of graph transformation with TGGs can be easily treated after having the parsing algorithm finished, but it can become troublesome, specially for the termination in the case of bi-directional incremental transformation with cyclic graphs.

Start and End Dates	Activities
12/02/2018 to 11/03/2018	Initial research; definition of theme; search of literature; proposal; problem concretization; approach definition and validation
12/03/2018 to 11/04/2018	Vacations
12/04/2018 to 11/05/2018	Development of the approach; theoretical definitions; sketch of parsing algorithms
12/05/2018 to 11/06/2018	Parsing algorithms and classification of grammars; 1st review
12/06/2018 to 11/07/2018	Work on TGGs and transformation; write introduction; 2nd review
12/07/2018 to 11/08/2018	Algorithms' analysis; evaluation of results; write the main and evaluation chapters
12/08/2018 to 11/09/2018	Finalize writing; 3rd review; prepare presentation/ defense

Table 1: Plan for the researching, developing and writing of the master thesis. This schedule is organized in months, where each month has its respective activities planned.

## Related Work

There exists an extensive literature on graph transformation and rewriting systems, specially on the algebraic approach to graph grammars [Ehrig *et al.*, 2015], which will be used as basic bibliography in this thesis. Nevertheless, there is also specific literature related to the problem proposed in this thesis. In [Gilroy *et al.*, 2017] [Rekers & Schürr, 1995] and [Rekers & Schürr, 1997], the problems of parsing graphs using graph grammars in regular languages and context-sensitive languages, as well as the classification of grammars were approached, but we believe that more investigation can be done in that direction.

## References

- [Ehrig *et al.*, 2015] Ehrig, Hartmut, Ermel, Claudia, Golas, Ulrike, & Hermann, Frank. 2015. Graph and Model Transformation.
- [Gilroy *et al.*, 2017] Gilroy, Sorch, Lopez, Adam, & Maneth, Sebastian. 2017. Parsing Graphs with Regular Graph Grammars. *Pages 199–208 of: Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (\* SEM 2017)*.
- [Rekers & Schürr, 1995] Rekers, Jan, & Schürr, A. 1995. *A parsing algorithm for context-sensitive graph grammars*. Rijksuniversiteit Leiden. Vakgroep Informatica.
- [Rekers & Schürr, 1997] Rekers, Jan, & Schürr, Andy. 1997. Defining and parsing visual languages with layered graph grammars. *Journal of Visual Languages & Computing*, 8(1), 27–55.