
— Bachelor's Thesis —

Designing and implementing a graph grammar to model the peano arithmetic

What is it all about?

Hyperedge-replacement grammars are used to define a language of hypergraphs. Such a presentation is used in order to abstract the semantics of a heap-manipulating program (implemented in a tool called ATTESTOR). This abstraction allows the analysis of invariant properties[Jan17]. However, in recent time we also could extend the analysis to also verify variant properties using graph grammar predicates, which we use to prove the termination of programs[Fes19].

ATTESTOR is currently only able to keep track of pointers (and some constants), thus we loose all information about integers. However, most programs use a combination of integer and pointer arithmetic. For this reason, the termination analysis is currently not able to prove any statements about integer. Modeling integers as a peano-like arithmetic using a list data structure may help us integrating integers into the current framework.

What is to be done?

1. Find a graph-grammar representation that can be used to model the integers
2. Implement the integer representation into ATTESTOR

Requirements

- Solid background in theoretical computer science.
- Capable of programming in Java.
- Background in static program analysis and (graph) grammars would be helpful.

Contact

- Ira Fesefeldt, fesefeldt@cs.rwth-aachen.de, Tel. 0241/80-21203.

References

- [Fes19] Ira Fesefeldt, [Proving termination of pointer programs on top of symbolic execution](#), Master's thesis, RWTH Aachen, 2019.
- [Jan17] Christina Jansen, [Static analysis of pointer programs - linking graph grammars and separation logic](#), Ph.D. thesis, RWTH Aachen University, Germany, 2017.