

# Master Thesis

## A Translator from BML annotated Java Bytecode to BoogiePL

Ovidio José Mallo

September 18, 2006

**Introduction** The Bytecode Modeling Language (BML) is a specification language designed to be the counterpart of the Java Modeling Language (JML) at bytecode level. As such, BML allows to specify the behavior of a Java bytecode program by annotating it using a subset of the JML specifications, including preconditions, postconditions, and class invariants, while introducing a new syntax which better meets the requirements of a bytecode environment.

Boogie is a modular, static program verifier part of the Spec# programming system. Its main purpose is to verify programs written in Spec#, an extension over C# offering support for specification constructs. For the actual verification, the program is first translated into an intermediate imperative language, called BoogiePL, from which verification conditions are then generated and passed to an automatic theorem prover. This permits to use Boogie for verifying programs written in other programming languages, provided an appropriate translation of the program's semantics to BoogiePL is given.

**The goal of this master's project** is to design and implement a compiler from BML annotated Java bytecode to BoogiePL, based on an already existing formalization of the actual translation for a limited subset of bytecode instructions. This mainly includes the design of adequate data structures for representing the bytecode together with the BML specifications as well as additional control flow information of the program. The translation will result in an abstract syntax tree representation of the BoogiePL program from which, in a second step, the actual program is generated.

**The main parts** of this project are:

1. The design of appropriate data structures for representing the BML annotated bytecode program and the BoogiePL program.
2. Parsing and checking the bytecode program along with the corresponding BML specifications.
3. The translation to BoogiePL.

Possible extensions include supporting further bytecode instructions and encoding various program properties enforced by the Java type system in BoogiePL for studying the system's practicability for verifying simple programs.

Supervised by: Prof. Peter Müller and Hermann Lehner