Automatic verification for the Bosque Programming Language

Mark Marron[1], Jose Abel Castellanos Joo[2]

[1]Microsoft Research
[2]University of New Mexico

Abstract. Automatic theorem proving tools has become extremely popular thanks to their application in software analysis/verification. It provides formal guarantees about software correctness and useful mechanisms for debugging highly non-trivial properties. However, these tools require a large amount of knowledge from their users to be able to use them correctly. Hence, it is complicated for many programmers to use verification in their projects.
Bosque is a new programming language designed for the future of Cloud and IoT application development with the intention to simplify the verification experience for application developers. This paper discusses the Bosque language features of that allow the automatic verification of its properties using a formal encoding in {FStar, SMTLib}.

Keywords: Formal Verification · Language Design · Software Engineering.

1 Introduction

2 Background

3 Encoding

Bosque is a strongly-typed functional programming language supporting nominal and structural typing. Bosque has a small core library consisting of booleans, integers, typed strings, tuples, records, unions, and structure-like composite data types called concepts (non-instantiable types) and entities (instantable types).

BEGIN META{Things to have in mind}

- High level verifiable language
  - Semantics
  - Encoding
- Collections / Inductive data types
  - Quantifiers

END META
The intermediate representation of Bosque