



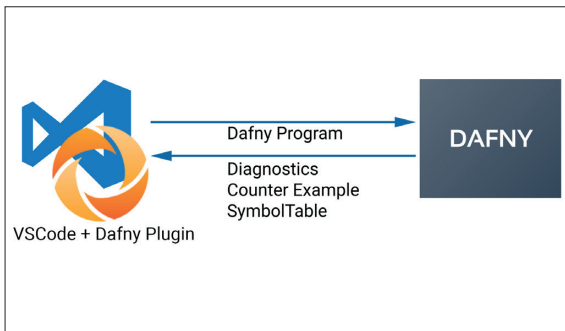
Rafael Krucker



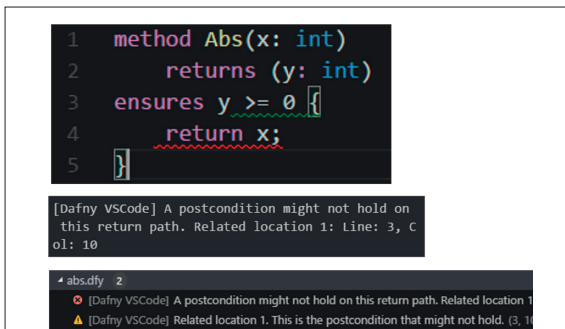
Markus Schaden

Graduate Candidates	Rafael Krucker, Markus Schaden
Examiner	Prof. Dr. Farhad D. Mehta
Co-Examiner	Dr. Valentin Wüstholz
Subject Area	Software Engineering – Core Systems
Project Partner	Microsoft Research, Redmond WA, USA

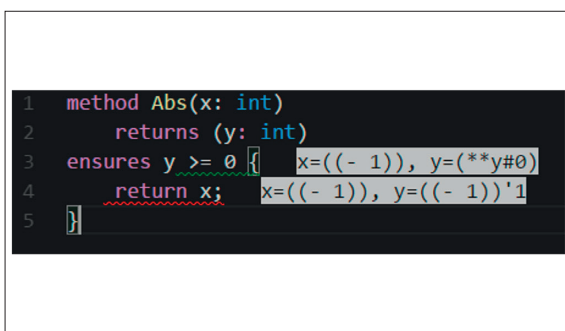
Visual Studio Code Integration for the Dafny Language and Program Verifier



Overview of the interaction between the plugin and Dafny



The plugin shows contract violations



Counter examples can be displayed

Introduction: The goal of this project was to integrate the Dafny programming language into Visual Studio Code. Emphasis was put into researching how Dafny programmers can be best supported during their work and how writing code can be made more productive. Since Dafny offers built-in specification constructs, novel work is to provide tooling that makes using them easier. The most beneficial feature would have been to implement a generic, context aware proof obligation generator that continuously suggests specification constructs to the programmer. This approach was eventually discarded because it was deemed unfeasible to implement after some research had been done. Instead, situations were identified that arise often during programming, for example bound checking, and specific aid with specification constructs was implemented for them. Another helpful feature is the displaying of counter examples where code written does not satisfy the corresponding specification constructs, allowing quick discovery of edge cases and refinement of specification constructs.

Approach/Technologies: Next to language specific features, standard IDE mechanisms allow for improvement regarding productivity. It was deemed paramount that the project implements the most common features such as go to definition and auto completion. This was achieved using the standard interfaces that Visual Studio Code provides, allowing programmers accessing these features in a well-established way. It is of concern that new users can get started quickly, so that the user base continues to grow. To support this, automatic installation on all major platforms was implemented. The installation resolves all dependencies such as the Dafny pipeline itself. To further maximize portability, the plugin implements the language server protocol. This allows for writing IDE agnostic language analysis platforms, making the plugin usable in Visual Studio Code, and also integrable into some other IDEs with only minor adjustments.

Result: This project concluded with the implementation of a production ready integration of Dafny into Visual Studio Code. Continuous integration allowed for a user base of about 300 people at the end of the project, proving that the plugin is robust and works across multiple environments. Dafny programmers are supported in their coding with standard IDE mechanisms and Dafny-specific features. Next to making the experience of programming more productive, this lays the foundation of a contentiously growing Dafny community.