Dafny, a programming language unlike any other

TU Delft Delft University of Technology

1. Introduction

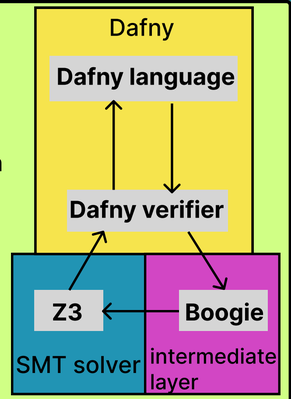
- Testing software can show the presence of bugs but not their absence.
- Formal verification is a stricter way of verifying correctness of software.
- This process is quite tedious to do on paper.
- SMT solvers exist to partially automate this process.
- Dafny is one such program that uses an SMT solver to verify software correctness.

2. Research Question

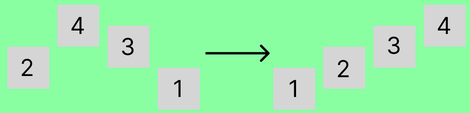
How can Dafny be used to formal verify a key-value store and sorting algorithm, which can then be compiled to C# code?

3. Background Dafny

- Created in 2009 by Microsoft Research under the lead of M. Leino.
- Dafny is a high level programming language with formal verification at its core.
- Dafny allows for compilation to other languages like C# and Python.
- While industry usage is low it has been used by Amazon and Microsoft for verification of security and validation systems.

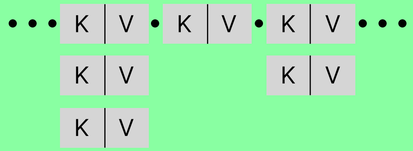


4. Formal problem description



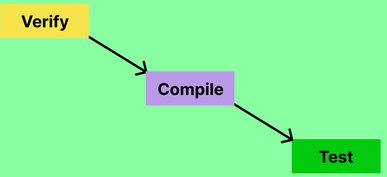
In-place selection sort

- The contents of the list stays the same.
- For every iteration of the algorithm, the list is sorted up until the pivot.



Key-value store

- all keys in the store are always unique.
- get(key) → value
- put(key, value)
- delete(key) → value



Compiling to C#

- after verifying compile to C#.
- Test functionality of output with a few simple tests.
- asses code readability.

5. Results & Discussion

- In-place selection sort was very straightforward.
- The built-in sequence types helped a lot here.

- Key-value store was less straightforward.
- Had to change a lot in order to verify it in Dafny.
- Beyond a few toy examples, a decent understanding of formal verification is needed.

- All basic tests passed.
- Over 1200 lines of code generated for what was ~400 lines of Dafny code.
- Code is barely readable.
- Less than ideal to work with.
- Very verbose to work with due to how the code is compiled.

6. Conclusion & Future work

- Dafny was able to verify both the selection sort and key-value store.
- Verifying programs becomes a lot harder the more complex it gets.
- Debugging via partially writing out the proof is often needed.
- The compiler is lacking.
- Future research regarding Dafny's ease of use compard to other tools could be conducted to see if Dafny's lacking compiler can outweigh having a native verifier specifically designed for a high-level programming language.

Jeroen Koelewijn
J.G.Koelewijn@student.tudelft.nl

Supervisors: Benedikt Ahrens, Kobe Wullaert
CSE3000 Research Project