

VDM2Dafny: An Automated Translation Tool for VDMSL to Dafny

Adam Winstanley

School of Computing Science, Newcastle University, UK

Abstract

This paper provides the rough skeleton for the writeup of my dissertation, there are short sentences throughout defining what the main focus of the section should be.

Keywords: Translation, VDMSL, Dafny, VDMJ plugin.

1 Introduction

2 Background

2.1 Program Proofs

Describe the information on how the programming patterns used in Dafny vary to those used in VDM. This will go on to subtly describe what possible issues there may be in the implementation of the translation strategies. Further details and mitigations will be discussed at a later point.

2.2 Why3

Why3 is another formal software verification language that was considered as a translation target for the project. Why3 is focused primarily on producing a

[1]

2.3 VDM2Isa

Existing tool that translated VDM to Isabelle. [2]

¹ Email: a.winstanley2@newcastle.ac.uk

3 What you did and how

3.1 *Possible problems and approaches to solve them*

On a preliminary look, why may each of these be a problem, how can they be resolved?

3.1.1 *Patterns*

3.1.2 *Cases*

3.1.3 *Types*

3.1.4 *Lemmas*

3.2 *Manual Translation*

Talks through the rationale in manually translating a Dafny example to translate to VDM. That being, identifying additional points of friction in the translation.

3.2.1 *Selecting an example*

Talks about why specific examples were chosen to translate. What makes these interesting.

3.2.2 *Thoughts on translation*

Talk on any of the previously identified points of friction and how to address them.

3.3 *Deriving string templates*

This should answer: why approach the project in this order? How would you go about deriving string templates for this task? what technology is used to produce strings.

3.4 *AST traversal*

How is the VDM AST traversed to gather all of information in each module?

3.5 *Producing a translated module*

Bringing this together, talk about how the AST is traversed to each leaf and how the translation produced as a result of this should arise.

4 Evaluation

4.1 *The subset of VDM that was tackled*

Evaluates the subset of VDM that can be translated into Dafny, against the subset of VDM that was actually translated — any difference in this is likely due to time constraints, or a mistaken belief that a certain part is untranslatable compared to the reality.

4.2 *Program Proofs Examples*

Show how the translation tool handles the examples translated earlier.

4.3 *VDM Toolkit Examples*

Show how the translation tool handles premade VDM models.

4.4 *Comparison of translation approaches*

How does the translation approach taken in VDM2ISA compare, merits and demerits of both.

5 Conclusions and further work

5.1 *Alternative approaches*

Talk on the more extensive approach that makes heavy use of the defined grammar rules in Dafny and VDM to produce the translation.

5.2 *Integrations with existing tools*

VDM2ISA is integrated with a vscode extension; on route forward to making this more usable would be to integrate this into the same one as VDM2ISA.

6 Acknowledgements

References

- [1] François Bobot, Jean-Christophe Filliâtre, Claude Marché, and Andrei Paskevich. Why3: Shepherd Your Herd of Provers. In *Boogie 2011: First International Workshop on Intermediate Verification Languages*, pages 53–64, Wrocław, Poland, 2011.
- [2] Leo Freitas. VDM Toolkit. https://github.com/leouk/VDM_Toolkit, 2024.