Testing functional agent-based simulations

A functional approach

Jonathan Thaler
Thorsten Altenkirch
jonathan.thaler@nottingham.ac.uk
thorsten.altenkirch@nottingham.ac.uk
University of Nottingham
Nottingham, United Kingdom

ABSTRACT

TODO: this would be ideal to submit to a conference where i can also discuss functional programming

TODO: write introduction TODO: implement case study 1: property-testing of SIR TODO: implement case study 2: property-testing of Sugarscape TODO: Show the use of Haskell Titan TODO: write discussion TODO: write background TODO: write related research TODO: write conclusion & further research

KEYWORDS

Agent-Based Simulation, Functional Reactive Programming, Property-Based Testing, Haskell

ACM Reference Format:
Jonathan Thaler and Thorsten Altenkirch. 2019. Testing functional agent-based simulations: A functional approach. In *Proceedings of International Symposium on Implementation and Application of Functional Languages (IFL'18).* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

We feel that code testing is still lacking in ABS and that there does not exist much research and work on it. We propose a new method, called property-based testing, from which we hypothesise that it is more expressive and allows testing ABS models more umfassend.

We use two models as case-studies : - SIR, explanatory model: formulate formal model-specifications in property-tests - Sugarscape, exploratory model: formulate the hypotheses in property tests

The aim of this paper is to investigate the potential of property-based testing which allows to directly express model-specifications in code and test them.

main message: testing of functional abs is easier due to composability and controlled side effects. also property-based testing is much more expressive, allowing a specification based testing. Contribution: first to look into the potential of property-based testing for agent-based simulation verification.

2 RELATED WORK

TODO: Test-driven agent-based simulation development [3] TODO: Back To the Future: Time Travel in FRP [4] TODO: Testing and Debugging Functional Reactive Programming [5]

IFL'18, August 2019, Lowell, MA, USA
2019. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

3 BACKGROUND

TODO: List of Common Bugs and Programming Practices to avoid them [6] TODO: QuickCheck: A Lightweight Tool for Random Testing of Haskell Programs [1] TODO: Testing Monadic Code with QuickCheck [2]

ACKNOWLEDGMENTS

The authors would like to thank

REFERENCES

[1] Koen Claessen and John Hughes. 2000. QuickCheck: A Lightweight Tool for Random Testing of Haskell Programs. In *Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming (ICFP '00).* ACM, New York, NY, USA, 268–279. https://doi.org/10.1145/351240.351266

[2] Koen Claessen and John Hughes. 2002. Testing Monadic Code with QuickCheck. *SIGPLAN Not.* 37, 12 (Dec. 2002), 47–59. https://doi.org/10.1145/636517.636527

[3] N. Collier and J. Ozik. 2013. Test-driven agent-based simulation development. In *2013 Winter Simulations Conference (WSC).* 1551–1559. https://doi.org/10.1109/WSC.2013.6721538

[4] Ivan Perez. 2017. Back to the Future: Time Travel in FRP. In *Proceedings of the 10th ACM SIGPLAN International Symposium on Haskell (Haskell 2017).* ACM, New York, NY, USA, 105–116. https://doi.org/10.1145/3122955.3122957

[5] Ivan Perez and Henrik Nilsson. 2017. Testing and Debugging Functional Reactive Programming. *Proc. ACM Program. Lang.* 1, ICFP (Aug. 2017), 2:1–2:27. https://doi.org/10.1145/3110246

[6] V. Vipindeep and Pankaj Jalote. 2005. *List of Common Bugs and Programming Practices to avoid them.* Technical Report. Indian Institute of Technology, Kanpur.

Received May 2018