1st Year Report

Jonathan THALER

March 13, 2017

Abstract

1 Part I: Basic Research

How can ABS/M be done in pure functional programming Haskell? Is it convenient? Are there special benefits / disadvantages to it?
- start with suitability of haskel / pure functional paradigm as implementation for ABS: take the programming paradigm paper for that. - extend section on Haskell with implementation details (no direct method/function call, parallelism, concurrency) - add special section on STM because its a killer-feature for concurrent ABS in Haskell - section on update-strategies following my submitted paper - EDSL for Functional Reactive ABS/M - model checking and reasoning by quickcheck: [6], [13]

2 Part II: Literature

2.1 Economics and Agent-Based Computational Economics

[19] gives a broad overview of agent-based computational economics (ACE), gives the four primary objectives of it and discusses advantages and disadvantages. She introduces a model called *ACE Trading World* in which she shows how an artificial economy can be implemented without the *Walrasian Auctioneer* but just by agents and their interactions. She gives a detailed mathematical specification in the appendix of the paper which should allow others to implement the simulation.
- Artificial agent-based economies: [19], [10], [11], [9], [3] - Artificial agent-based markets: [15], [7] - Agent-Based Market Design: [16], [5]
market-microstructure: [?], [2]
Basics of Economics [4], [14]

2.2 Social Simulation

The SugarScape model [8] is one of the most influential models of agent-based simulation in the social sciences. The book heavily promotes object-oriented programming (note that in 1996 oop was still in its infancy and not yet very well understood by the mainstream software-engineering industry). We ask how it can be done using pure functional programming paradigm and what the benefits and limits are. We hypothesize that our solution will be shorter (original reported 20.000 LOC), can make use of EDSL thus making it much more expressive, can utilize QuickCheck for a completely new dimension of model-checking and debugging and allows a very natural implementation of MetaABS (see Part III) due to its recursive and declarative nature.
TODO [12] TODO [17]

2.3 Functional Programming, Computation and Category Theory

all FRP, quickcheck, arrows, monads, wadler, hughes
Category-Theory [?] [18]

2.4 Agent-Based Modelling & Simulation and Actors

wooldridge, will clinger, hewitt

2.5 Complex Systems

TODO: baas: emergence, hierarchies and hyperstructures TODO: [1]

3 Part III: Phd Research

The central aspect of my PhD is centred around the main question of *How can Agent-Based Simulation be done using pure functional programming and what are the benefits and disadvantages of it?*. So far functional programming has not got much attention in the field of ABS and implementations always focus on the object-oriented approach. We claim, based upon the research of the first year that functional programming is very well suited for ABS and that it offers methods which are not directly possible and only very difficult to achieve with object-oriented programming.
We claim that to build large and complex agent-based simulations in functional programming is possible using the functional reactive programming (FRP) paradigm. We applied FRP to implementing ABS and developed a library in Haskell called FrABS. We implemented the quite complex model SugarScape from social simulation using FrABS and proofed by that. that applying FRP to ABS enables ABS to happen in pure functional programming.
Due to the declarative nature of pure functional programming it is an established method of implementing an EDSL to solve a given problem in a specific domain.

We followed this approach in FrABS and developed an EDSL for ABS in pure functional programming. Our intention was to develop an EDSL which can be used both as specification- and implementation-language. We show this by specifying all the rules of SugarScape in our EDSL.

Due to the lack of implicit side-effects and the recursive nature of pure functional programming we claim that it is natural to apply it to a novel method we came up with: MetaABS, which allows recursive simulation.

Finally having such an EDSL at hand this will allow us to reason about the programs. This will be applied to specify and reason about the dynamics and emergent properties of decentralized bilateral trading and bartering in agent-based computational economics (ACE) and social simulations like SugarScape. Disadvantages - although the lack of side-effects is also a benefit, it is also a weakness as all data needs to be passed in and out explicitly - indirection due to the lack of objects & method calls. - When not to use it: - if you are not familiar with functional programming - when you can solve your problem without programming in a Tool like NetLogo, AnyLogic,... - when you don't need to reason about your program

3.1 Main contributions to knowledge

1. Functional reactive ABS: ABS/M is possible in functional programming by development of FrABS

2. EDSL for ABS: An EDSL and its semantics for model specification in FrABS where spec = code, no need for explicit verification

3. MetaABS: Novel concept which is very natural enabled through FrABS: MetaABS, running locally-recursive simulations

4. Reasoning about ABS: dynamics and emergent properties in ABS based on FrABS EDSL and its semantics in the context of decentralized bilateral bartering

3.2 Planned/Published Papers

- Art of Iterating:

- FrABS: an EDSL and its semantics for ABS where the EDSL = spec = code

- MetaABS: recursive agent-based simulation

- reasoning about dynamics and emergent properties of ABS using FrABS in the context of decentralized bilateral trading / bartering

3.3 Research Questions

Define 5 general research questions for each Research-Context

- 2 related to FP

- 1 related to integration of FP to ABM/S

- 2 related to ABM/S

Main How can Agent-Based Simulation be done using pure functional programming? What are the benefits of doing so? What are the deficits of doing so?

Can we build an EDSL for ABS which is both specification- and implementation-language? - Can we define semantics for the EDSL to do reasoning about ABS in general?

MetaABS which allows to do recursive simulation

How can we reason about ABS in general in pure functional programming? - dynamics - emergent properties - deadlocks - silence (no messages/agent-agent communication and interaction) - define semantics of FrABS based on semantics of FRP and Actors - what is emergence in ABS and how can we reason about it? - identify emergent properties: equilibrium, behaviour on macroscale not defined on micro, chaos,... - can we anticipate emergent properties / dynamics just by looking at the code and reason about it? - can emergence in ABS be formalized? - hypothesis: it may be possible through functional programming because of its dual nature of declarative EDSL which awakens to a process during computation - what is the relation between emergence and computation? we need change over time (=computation) for emergence

Can we reason about the dynamics and equilibria of agent-based models of decentralized bilateral trading & bartering?

How we incorporate MetaABS in this direction?

References

[1] BAAS, N. A., AND EMMECHE, C. On Emergence and Explanation. Working Paper, Santa Fe Institute, Feb. 1997.

[2] BAKER, H. K., KIYMAZ, H., ALAN, N. S., BILDIK, R., AND SCHWARTZ, R. Market Microstructure in Emerging and Developed Markets. *Business Faculty Book Gallery* (Jan. 2013).

[3] BOTTA, N., MANDEL, A., IONESCU, C., HOFMANN, M., LINCKE, D., SCHUPP, S., AND JAEGER, C. A functional framework for agent-based models of exchange. *Applied Mathematics and Computation 218*, 8 (Dec. 2011), 4025–4040.

[4] BOWLES, S., EDWARDS, R., AND ROOSEVELT, F. *Understanding Capitalism: Competition, Command, and Change*, 3 edition ed. Oxford University Press, New York, Mar. 2005.

[5] BUDISH, E., CRAMTON, P., AND SHIM, J. Editor's Choice The High-Frequency Trading Arms Race: Frequent Batch Auctions as a Market Design Response. *The Quarterly Journal of Economics 130*, 4 (2015), 1547–1621.

[6] CLAESSEN, K., AND HUGHES, J. QuickCheck: A Lightweight Tool for Random Testing of Haskell Programs. In *Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming* (New York, NY, USA, 2000), ICFP '00, ACM, pp. 268–279.

[7] DARLEY, V., AND OUTKIN, A. V. *Nasdaq Market Simulation: Insights on a Major Market from the Science of Complex Adaptive Systems.* World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2007.

[8] EPSTEIN, J. M., AND AXTELL, R. *Growing Artificial Societies: Social Science from the Bottom Up.* The Brookings Institution, Washington, DC, USA, 1996.

[9] GAFFEO, E., GATTI, D. D., DESIDERIO, S., AND GALLEGATI, M. Adaptive microfoundations for emergent macroeconomics. Department of Economics Working Paper 0802, Department of Economics, University of Trento, Italia, 2008.

[10] GINTIS, H. The Emergence of a Price System from Decentralized Bilateral Exchange. *Contributions in Theoretical Economics 6*, 1 (2006), 1–15.

[11] GINTIS, H. The Dynamics of General Equilibrium. SSRN Scholarly Paper ID 1017117, Social Science Research Network, Rochester, NY, Sept. 2007.

[12] HUBERMAN, B. A., AND GLANCE, N. S. Evolutionary games and computer simulations. *Proceedings of the National Academy of Sciences 90*, 16 (Aug. 1993), 7716–7718.

[13] HUTTON, G. A Tutorial on the Universality and Expressiveness of Fold. *J. Funct. Program. 9*, 4 (July 1999), 355–372.

[14] KIRMAN, A. *Complex Economics: Individual and Collective Rationality.* Routledge, London ; New York, NY, July 2010.

[15] MACKIE-MASON, J. K., AND WELLMAN, M. P. Chapter 28 Automated Markets and Trading Agents. In *Handbook of Computational Economics*, L. T. a. K. L. Judd, Ed., vol. 2. Elsevier, 2006, pp. 1381–1431. DOI: 10.1016/S1574-0021(05)02028-9.

[16] MARKS, R. Chapter 27 Market Design Using Agent-Based Models. In *Handbook of Computational Economics*, L. T. a. K. L. Judd, Ed., vol. 2. Elsevier, 2006, pp. 1339–1380. DOI: 10.1016/S1574-0021(05)02027-7.

[17] Nowak, M. A., and May, R. M. Evolutionary games and spatial chaos. *Nature 359*, 6398 (Oct. 1992), 826–829.

[18] Spivak, D. I. *Category Theory for the Sciences*, 1 ed. Mit Press Ltd, Cambridge, Massachusetts, Nov. 2014.

[19] Tesfatsion, L. Agent-Based Computational Economics: A Constructive Approach to Economic Theory. Handbook of Computational Economics, Elsevier, 2006.