Research Diary
PhD Studies

Jonathan Thaler

November 5, 2016

Started: 8th August 2016

Abstract

In this diary I try to keep track of the process of my PhD studies by recording my thoughts, my reflections, my (self) doubts and my philosophical ideas in connection with my research output. I did some upfront literature research and reflecting ideas until the beginning of my Ph.D. in Nottingham. The notes are all put in the appendices where the real diary starts on the day of my first official supervision meeting which was on the 6th October 2016.

2016 October 6th

First official supervision-meeting

Had a nice and relaxed first official supervision meeting with Peer-Olaf today. We talked about two things: organisational and research topic related stuff.

Organisational stuff

- I should contact Nadine Holmes `https://www.nottingham.ac.uk/computerscience/people/nadine.holmes` for both my office key and for a website presence of me on the IMA page.
- Peer-Olaf does not care much about which kind of courses I attend for my Ph.D. but he suggests that I should go to "Thesiswriting" not in the last year.
- Regarding publishing papers: the computer-science Ph.D. here at Nottingham is organized in the way that officially no published papers are required and only the final thesis is sufficient to become a Ph.D. (if everything is OK). BUT it would be nice to have at least 1-2 journal papers (2 would be good and are suitable for my topic) because that is a very strong backup of the thesis against the neutral reviewer and also because

one should have the aim to publish something in a Ph.D. (I really want to publish something).

- Conference-Papers are mandatory: the first should conference (with paper of course) should be at the end of the first year, with more conferences and papers to come.
- Peer-Olaf suggests that I may attend courses/lecture in economics in the 2nd year when required (e.g. Market Microstructure or Equilibrium Theory).
- I should visit Simon Gächter for a quick chat and to build relationship to the economic guys.
- Sometimes there are interesting Summer-Schools and Workshops and I may attend some if the topics are interesting for me.
- I am now member of the IMA and will thus attend seminars held by IMA over the next 3 years.
- I will have 2 supervision-meetings a month with Peer-Olaf.

Research topic related stuff

We discussed intensely what I really want to do and what Peer-Olaf has in mind.

- His students go in the direction of developing some method and then showing that it can be applied to various fields e.g. agent-based computational economics (ACE), social simulation, epidemiology. This is also the way I will work: develop a framework which allows implementing and reasoning in one of the various fields - thus the framework should be domain-agnostic as much as possible.
- Peer-Olaf mentioned Akka to me, that I should take a look at it.
- I clearly stated (and Peer-Olaf told me to write that down) that: I don't want to force OO-concepts to pure functional programming but want to approach the agent-based modelling/simulation methodology from a pure functional way (which I guess will be then category theory).

First work-package

Peer-Olaf suggested to start with something very small but highly abstract and then extend it more and more. Thus my starting points will be

- How can an Agent be represented in a functional way?
- How can an Agent be implemented in a pure functional language?

I need to start with a general agent-model of interaction, concurrency and pro-activity and then map these to functional concepts and then translate these concepts to a pure functional language. Thus first I have to look at various agent-models like the Actor-Model, select the one which fits best (what are the criteria?). Then I have to find a pure functional representation and finally implement this in Haskell. Note that it is highly desirable that the mapping from functional specification to pure functional implementation should be straight forward without loosing much expressiveness thus category theory would be the first way to go for the functional specification. To test the ideas and implementations I should apply it to a simple epidemic model e.g. the SIR model as it is very well known and researched and the results can easily be compared. After all has been implemented then I should use Agda to implement the whole application in a pure functional language with dependent types and then do reasoning about this model e.g. termination checking.

1. Find agent-model of interaction, concurrency and pro-activity: *Actor-Model, Process Calculi.*
2. Create functional representation/specification of agent-model: *Category Theory, still open to research.*
3. Create functional representation/specification of application: SIR-model.
4. Implementation in pure functional language: *Haskell*
5. Implement in dependently typed pure functional language: *Agda*
6. Do formal verification and reasoning about the model.

2016 October 7th

Start of implementation

Maybe I still went too formal yesterday but should be more practical thus I should go right into implementation of the SIR model using agents in Haskell. As the underlying model for agents I will use the actor-model as it is very well defined, researched and proven to be useful (Erlang) and I am already familiar with it (Gul Aghas Book) and have experimented with it in Erlang. So the starting point will be to bring the actor-model to Haskell. Fortunately there are exist already a number of libraries which either directly implement the actor-model or provide mechanisms to implement them directly. Here is an overview, loosely based on `https://wiki.haskell.org/Applications_and_libraries/Concurrency_and_parallelism` and on searching for 'actor' on hackage - note that the ordering reflects the order of interest in the library:

1. **Hackage-Package 'hactor'. Conclusion: WINNER, it seems to be mature (version 1.2), minimalistic, not as outdated as the others (last commit 2 years ago), installing using cabal works, interface looks 'good'.**

2. Hackage-Package 'hactors. Conclusion: last commit 5 years ago, could install it using cabal.

3. Cloud Haskell is a full blown Erlang-style concurrent and distributed programming framework for Haskell. Conclusion: not simple enough, this is too much for now because I don't need distributed computation but Cloud Haskell would incur doing up a lot of infrastructure stuff.

4. Communicating Haskell Processes (CHP): is a Haskell library following the CSP (Communicating Sequential Processes). It may be an alternative to the Actor-Model but for now, follow the Actor-Model.

5. Hackage-Package 'simple-actors': last commit 4 years ago, couldn't install it using cabal, conflicts

6. Hackage-Package 'actor: Actors with multi-headed receive clauses by Martin Sulzmann et al. `http://sulzmann.blogspot.co.uk/2008/10/actors-with-multi-headed-receiv` `html`. Conclusion: couldn't install it using cabal, seems to have conflicting/missing dependencies (project is from 2008!)

7. Hackage-Package 'gore-and-ash-actor': is actually a game-engine extension of the (gore&Ash engine) to implement actor style of programming. Conclusion: too focused on game-engine, not minimal enough, didn't try it out.

Note that all of these libraries build upon the very low-level parallel & concurrency frameworks available in Haskell in the Concurrency package. Thus I also have to understand the basics of parallel (non-interacting, parallel computation) and concurrent (indeterministic-interacting, parallel computation) programming in Haskell. Using this low level actor-model an agent-model must be implemented. Then using the agent-model the actual simulation-model can be implemented: SIR.

2016 October 12th

Implementation so far

I rejected all the Actor-Model libraries as they either couldn't be installed using cabal/manual install or were tiresome to get into. The point is that I would have had to build on top of one of them my agent implementation which would have complicated things. So instead I chose to start from nothing and build up experience. The following points became clear

- My implementation technique so far is pretty straight-forward: all state is carried around, everything works deterministic.

- I tried experimenting with Concurrency-Mechanisms built into Haskell2010: forkIO, TChan, MVar but the problem is that they are inherently INDETERMINISTIC and thus one looses the ability to reason.

- Haskell also supports deterministic parallelism, which could be the solution to the dilemma of wanting to have parallel execution paths but still keeping the determinism.
- In the end the questions will be: 1. how can agents send messages in pure functional languages? 2. is deterministic parallelism possible?

Meeting with Paolo

In the morning I had an interesting meeting with Paolo, a post-doc, who is very experienced in Haskell & Agda. I showed him my current implementation and we discussed a bit about it, especially the problem of how state can be represented and handed around in the simulation using Haskell. He proposed *Coalgebras* and also explained it a bit to me but it was too complicated to understand it with the quick tutorial - I should contact Venanzio for this. Also he suggests to get into Category Theory, which will be a big task which will take months but will pay of in the long run. He also mentioned that a guy called Ivan is working on Functional Reactive Programming, a paradigm which can be used to program games in Haskell - this struck me as games can be understood as a continuous agent-based simulation, so maybe this is one way to go.

Sum-up

- Start learning *Category Theory*
- Start looking at *Coalgebras*
- Look at *Functional Reactive Programming (FRP)*
- Look at formal/functional definition of agent architecture in Wooldridges Book Chatper 2.6 Abstract Architectures for Intelligent Agents.
- How to do communication? Look at Wooldriges fundamental book chapter 7 communication.

Presentation of aim & impact

I had a 10 min. presentation of my aims & impact of my thesis to my PhD colleagues. It went very nice with good discussions and also the other presentations of my colleagues were very interesting - I hope to do that more often. One thing I noticed was not very well structured and transported in my presentation. What the others communicated very clearly was

1. Which kind of problem there is.
2. What the aim is e.g. solving the problem.
3. How the aim is achieved by enumerating VERY CLEAR objectives.

4. What the impact one expects (hypothesis) and what it is (after results).

Of course I am just at the start of my research so I couldn't provide all the given details. Formulating these things will be the major topic until summer.

2016 October 17th

On Friday I participated in the FP (functional programming) lunch where everyone buys sandwiches and then gathers in a presentation-room and one is holding a presentation for about 45 mins + 15 mins discussion. I met Nenrik Nilsson there and he said we should talk to each other as he is the main force behind YAMPA a FRP framework. This are good news as I thinkg FRP may be a very good approach to functional ABM/S.

Feedback on my presentation

Today Peer-Olaf gave me feedback on my presentation. He said that speaking was very good and confident but he mentioned a few things which I should focus on

- Use page-numbers, so that the audience will know where the presentation currently is (beginning, middle, end,...)
- Mention your name in the beginning, maybe not everyone knows you already.
- *the way to go* means that it is the future. This is not what I wanted to say when I said that OO is the way to go in ABM/S - I wanted to express, that it is the current state-of-the art, the current way how it is done.
- Know your audience! Always assume that you have a mixed audience who have very basic knowledge. 5 points is the most people can take from a presentation.
- The aim is something general, needs to be testable, so to be proofed that I have achieved it.
- Discussing my contribution: what is a framework?
- The research-question(s) should come before the aim. The aim is the statement, maybe just a statement of the research questions. Only phrase one of the two as both mean basically the same.

Peer-Olaf also said that I have to really proof that there is a gap in my field I am researching e.g. that there is a gap when using Akka, Erlang,... as one can't reason in these languages. He also told me to mention DEVS which I should look at and mentioned that there is a book by Wainer.
We also discussed the structure of how to build up a presentation:

1. Start with motivation & interest.
2. Give a background what is out there.
3. Develop this towards a gap and show what is missing.
4. This then leads to research question(s).
5. This in the end leads to an aim which is condensed research questions.
6. Refine the aim with objectives to be solved for the aim to be reached.

2016 October 18th

Meeting with Henrik Nilsson

I had a meeting with Henrik Nilsson, the inventor/maintainer of Yampa, a FRP framework for Haskell. We talked about FRP being suitable to be used for ABM/S and he was pretty confident that Yampa could handle the things I described (Agent-Model, agent-based economics). The main point when following this road will be developing a **EDSL for ACE** on top of Haskell and Yampa. This sound awesome as this would give a domain-expert a tool for developing his models and then reason about them. Henrik said I should play around with Yampa a few days and see if it is suitable for what I want to do - thats what I'm going to do: implement SIR in Yampa!

2016 October 20-21th

Experiences with *classic* SIR/SIRS model-implementation

I finally got the SIR implementation in classic Haskell running - with classic I refer to the kind of programming: no Monads, no FRP, no other fancy stuff just pure Haskell with maybe IO for text-output/files writing. I am pretty proud of it: it performs replications, detects when a replication has finished (no more infected people) and exports the dynamics as JSON to a file which can be then visualized using a Matlab/Octave script.

- After having implemented replications only the first replication actually did calculate, the remaining ones were just copies of the first one. My guess was that because the compiler / runtime-system infers due to referential transparency that the call to calculating one replication is the same for all. Initially I thought that to run multiple replications properly I would have to introduce pure use of RNG - which I did - but that was not the origin of the problem. The problem was that replications had *no input*, which of course led Haskell to believe that it is constant after having been calculated. So after having added an input - the index of the replication - all replications were calculated properly. This is also known as constant applicative form (CAF): `https:`

`//wiki.haskell.org/Constant_applicative_form`
It is important that one is careful detecting CAF throughout all the program in which RNG are used: they will lead to the same results when repeatedly called - this also applies to the main-function!!! This was a major point of confusion but works now: population-count and replication-count are now passed in from outside (command-line args/function arguments).

- It was very easy to extend the model to SIRS, which is more interesting.
- It was not so easy to debug due to confusing call-hierarchies. Maybe unit tests are a remedy for this.
- To a non-programmer / unfamiliar with Haskell this implementation is probably a horror to work on due to scattering of functions all over the place, unclear call-hierarchy, implementation details showing up.
- In my opinion this solution is too near to the OO-thinking: agents are a structure of data, have message-handler and in all functions the agent-instance is passed in. This is clearly oo-technique disguised as functional. Just because all is implemented in (pure) functions where all the state is threaded in and out through parameters/return values doesn't make it functional yet. The next approach should be one using Monads where the goal will be to encapsulate the whole thing and provide a small EDSL for this SIRS model.

2016 October 24th

I noticed that I have lot of hidden beliefs about science/computers. I need to detect them and write them down, so they don't take me by surprise
I removed all the todos, as they are terrible repetitive and I've put a todo/goals list in research proposal for 1st year.

STM with lambdas and data-closures

I tested STM with lambdas and data-closures. It is possible to send a lambda with data-closure using STM. This would allow to send a computation which encloses e.g. Agent a to another Agent b which then executes this computation.

What is my contribution?

In the PGR seminar today the lecturer made it VERY clear that when we defend our research and present it to others we have to show very clearly what our contribution is. This is what many PhD students fail to show, especially in the beginning. He made it clear that original work is not enough, there must be new knowledge in the research. Also I have still difficulties and to be honest at the moment I don't know it yet but my hope is that I can develop a *pure functional model of ABM/S* - this is what I am currently very deeply thinking

about, how such a pure functional model could look like and what it actually is. The problem is that I fall into the thinking about implementation details too fast, I must refrain from that and keep it purely in the abstract domain without going into Haskell details.
WHY FUNCTIONAL? "because its the ultimate approach to scientific computing": fewer bugs due to mutable state (why? is thos shown obkectively by someone?), shorter (again as above, productivity), more expressive and closer to math, EDSL, EDSL=model=simulation, better parallelising due to referental transparency, reasoning
scientific results need to be reproduced, especially when they have high impact. a more formal approach of specifying the model and the simulation (model=simulation) could lead to easier sharing and easier reporduction without ambigouites
pure functional agent-model & theory, EDSL framework in Haskell for ACE

A paper on Akka: using the actor-model in ABM

Unit-Tests

Updated Todos: Unit-Tests are written (very few), ignore data-parallelism and stuff for now - will be an implementation detail later, removed AKKA for now. What I want to look into is how to test functions which use RNG: the RNG must be made deterministic. unit tests: also test invariants: specific data of an agent hasnt changend

2016 October 26th

Supversision-Meeting with Peer-Olaf

Peer-Olaf told me what he thought about should the direction of my PhD be: I should think about 4 scenarios from economics (e.g. Auctions, Supply-Chain, Public-Goods Games,...) in which 1 is supposed to be VERY suitable for functional approaches, one is supposed to be absolutely not and the other 2 are somewhere in between. My contribution would be to show "What are the upsides and the limits using functional approach in ABM". Also I should apply 3-4 different technologies for implementing them: AnyLogic/Repast/Netlogo (very high level), Scala with Actors: Akka (concurrent object-oriented functional), Haskell (pure functional), Java (object-oriented, non-functional) and show what the differences are between them implementing the 4 scenarios. In the end I need to come up with criteria for deciding which is "better" suited than the other. Mainly it is all about robustness: making less errors. This is influenced by:

- Testability: how and how easily can it be tested, Unit-Tests?
- Size of Code: how many lines of code have to be written?

- How is state treated?
- Is it suited for scientific computing?
- Parallelism: how difficult is it?

2016 November 3rd

The entries become less frequent with less volume each. I think this is because I am settling more and more in, calming down, sinking into research, not wanting to solve *everything now*, but to work continuous and focus on specific things.

Wildfire with Akka

I implemented the Wildfire with Wind-Simulation of AnyLogic about 1 week ago (just finished it before the supervision meeting) using Akka & Scala. I was able to implement it in about 1 day in total. Scala is a nice language and the Akka-Actors are also very convenient to work with: it all feels like an updated and modern version of Erlang which I was experimenting with at the beginning of this year (and which taught a few basics of functional programming). This is definitely gonna be a path to follow on how ABM/S can be functionally done using Scala & Akka. In the end it all boils down to 1. the agent-model and 2. how the agent-model is implemented in the according language. Of course both points influence each other: functional languages will come up with a different agent-model (e.g. hybrid like yampa) than object-oriented ones (e.g. actors).

Wildfire with Haskell & Yampa

Things are progressing slowly but steadily with Haskell & Yampa where I try to implement the Wildfire simulation already done with Scala & Akka. I decided NOT to use a GUI-framework but to use OpenGL with GLFW to do the rendering - it turned out to work very easy and convenient (using OpenGL 3.0, so can submit vertices and their properties directly without the need for VBOs and shaders as in OpenGL 4.0).
Yampa was but another difficulty as it is a completely new paradigm to learn: advanced functional programming, Arrows (and their syntax) and reactive programming which needs to think a system in a different way.

2016 November 4th

Meeting with Alexander Possajenikov. In the meeting the following points came up

- Reproducability is a BIG issue and not easy - researches have to trust each other.

- Economic models are often qualitative ones: specify the quality of the results instead of quantiative concrete results - trying to give general results.
- Book: Computable Economics by Vellupilai
- I should look at a few agent-based market-models instead of the whole ACE because there is no established consencus what ACE is and how it should be approached. I should maybe look at a reduces subset of models (markets) and look into them, Alex said that there are about 5-6 models.

We agreed to meet again in January after I have looked into a few market-models.

The biggest conclusion of this meeting is that:

1. (Still) cant convincingly explain the benefits of functional programming to a non-computer scientist.
2. I noticed that it is pretty hard to convince an agent-based economics specialist who is not a computer scientist about a pure functional approach. My conjecture is that the implementation technique and method does not matter much to them because they have very little knowledge about programming and are almost always self-taught - they don't know about software-engineering, nothing about proper software-design and architecture, nothing about software-maintenance, nothing about unit-testing,... In the end they just "hack" the simulation in whatever language they are able to: C++, Visual Basic, Java or toolboxes like Netlogo. Thus I REALLY need to come up with convincing arguments why to use pure functional approaches in ACE THEY can understand, otherwise I will be lost and not heard (not published,...).

2016 August 8th

First entry of research diary is a retrospective how I got to study a PhD at Nottingham. Due to the length of it I placed it in the appendices as Appendix B. Note that the next entry (10th) will give a short overview of the topic as it was at that point (10th).

2016 August 10th

Where do I stand now

On 21st June I had the opportunity to give a Phd-Seminar presenting my research ideas within 45mins + 15min discussion. This gave me the opportunity to make up my mind about the ideas I want to follow and which ones I want to reject. Also to better shape my ideas and to give a clear overview of them I started to write down a research proposal about 3 months ago. Of course this proposal is always changing and will continue to do so during the 1st semester as according to my supervisor the 1st semester is reserved to get started and to really carve out the research ideas and most important of all the research questions. So this is a still ongoing process but to give an overview where I stand NOW regarding the research direction I have will cite here the abstract of my research proposal which I have recently reworked (again) and which pretty well sums up the overall ideas and directions I want to follow:

> Agent-Based Modeling and Simulation (ABM/S) is still a young discipline and the dominant approach to it is object oriented computation. This thesis goes into the opposite direction and asks how ABM/S can be mapped to and implemented using pure functional computation and what one gains from doing so. To the best knowledge of the author, so far no proper treatment of ABM/S in pure functional computation exists but only a few papers which only scratch the surface. The author argues that approaching ABM/S from a pure functional direction offers a wealth of new powerful tools and methods. The most obvious one is that when using pure functional computation (equational) resoning about the correctness and about total and partial correctness of the simulation becomes possible. The ultimate benefit is that Agda becomes applicable which is both a pure functional programming language and a proof assistant allowing both to compute the dynamics of the simulation and to look at meta-level properties of the simulation - termination, convergence, equilibria, domain specific properties - by constructing proofs utilizing computer aided verification.
> To map ABM/S to pure functional computation the idea is to apply both Robin Milner's PI-calculus and category theory. The PI-calculus will be used for a formal modelling of the problem and allows already a basic form of algebraic reasoning. Then the agents and the

process of the agent-simulation will be mapped to category theory because pure functional programming approaches complex problems from the direction of category theory in the form of monadic programming.
The application will be in the field of agent-based computational economics where the approach will be to take an established model/theory and then apply the above mentioned methods to it and to show that using them will lead to the same results.

Note that the goal is not to establish new economic theories but to provide methods and tools for deeper insight and verification in the context of agent-based computational economics - this is after all still a thesis purely rooted in computer science.

PhD Roadmap Semesters

1. Semester: Literature research & research questions. Topics:
   - Functional computation & programming: Agda, Haskell & monads, category theory, type-systems, computer aided formal verification
   - Formal ABM/S: pi-calculus, various formal agent-types
   - Finding application in agent-based computational economics: auction theory & auction types, market design
2. Semester: First publication: ? (Mapping auction types and ABM/S to category theory)
3. Semester: Second publication: ? (formalization of ABM/S and auction types in pi-calculus)
4. Semester: Simulation Framework, Finished Simulation framework implementation: basic framework with correctness & proofs
5. Semester: Third publication: ABM/S in Agda
6. Semester: Finalizing PhD - Writing final thesis combining all research & results.

Discussions with my Supervisors

- So far the proposed ideas are a huge amount for 3 years and I doubt it is realistic to do in 3 years. Also it is yet totally unclear whether it makes sense/is possible to map ZI-Agents and the auction-types to category theory AND write specifications in pi-calculus for both. Maybe only one of them is really necessary or maybe only part of each? This is to be discussed with my supervisors.

- Does it even make sense to study these auction-types with these types of agents? To look only at the dynamics? Is computational economics interested in these dynamics of the equilibrium processes? I really need to look into theoretical work on the various auction-types AND the computational economics approach to them.
- Do we really need the pi-calculus when we have category-theory and vice versa? Does not just one method suffice? Or is there a mapping between both methods / a connection?

August 2016 11th

Paper: Dependently-typed programming in scientific computing [?]

The authors give a short and nice introduction into the very basics of what economics is and what it tries to achieve: exchange of goods. They then go on and explain allocation and endowment and explain walrasian equilibrium and then use Agda for constructing fundamental economic proofs about Pareto Efficiency, Walrasian Equilibrium. Remarkable is their criticism of the equilibrium models:

> A general criticism of all these models is that they neglect the dynamical aspect of reaching the equilibrium situation. There is no known plausible mechanism which explains exactly how equilibrium prices can arise in practice. Walras' own proposal for such a price-formation mechanism involved an auctioneer. This is a central entity who can see all supply-demand imbalances and adjust prices accordingly, raising the prices of goods for which there is too great demand, and lowering those for which there is too little, in an iterative process. Even if one accepts that in some situations one could have an authority that might act as auctioneer, there is no general proof that the iterative process will eventually converge.

This is insofar remarkable as this is exactly the direction I want to head to and which was also the work of the Leverage-Cycle project in which I've written my Masterthesis: **compute (by simulation) and understand the dynamics of an equilibrium process**. One of the goals of this PhD thesis is to develop a tool in which such equilibrium processes can be modelled and simulated but it is still not clear in which economical context / which theory exactly - this is still an open question I have to deal with.

Another interesting point they make in the paper is that:

> The bad news is that most of these concepts are not constructive. Specifications of programs that take as input agents characterized

> by preference relations and initial endowments and return a Walrasian (or Nash, or correlated, ...) equilibrium can in general not be fulfilled. Even the so-called computable general equilibrium models are not, in fact, computable.

I can't yet tell the implication for my thesis but it will be interesting to see whether it has one or not.

Paper: Engineering Proof by Reflection in Agda [?]

The authors explain the reflection-mechanics of Agda and discuss how they can be put to use. In general reflection allows for a kind of meta-programming during both compile- and run-time by accessing and modifying the abstract syntax tree (AST) thus changing the structure and behaviour of the program during compile- and run-time. According to the authors, Proof by Reflection

> is the method of mechanically constructing a proof of a theorem by inspecting its shape.

Unfortunately this is paper comes too early as I don't know enough about Agda, intuitionistic logic, type theory and computer aided formal verification. I have to come back to this paper at a later point again.

Paper: Dependent Types at Work [?]

According to the authors

> The aim of these notes is to give a gentle introduction to dependently typed programming for a reader who is familiar with ordinary functional programming, and who has basic knowledge of logic and type systems.

A few important topics I will going to clarify deeper after having worked through the paper and learned more about Agda and the theory behind it:

- *Dependent Types* where the type depends on another one e.g. a Vector of length n or a tuple (a,b) where a is less than b are dependent types.
- *The Curry-Howard correspondence* says basically that programming and proving will be the same.
- *Totality of Agda programs and Type-normalisation* Types must be normal otherwise the type-checking algorithm may not terminate
- *The connection of Dependent Types to the Curry-Howard correspondence* Still a VERY IMPORTANT open question.

August 2016 16th

Been working on [**?**] (Dependent Types at Work) which includes Exercises at the end of each subsections which are of great help to better understand the content if one solves them.

Category Theory vs. Pi-Calculus?

I've been thinking again about the combined application of category theory and pi-calculus in my thesis. Initially I thought that either of them is going to make it into the thesis because just one of them will be useful as a tool for formally expressing the type of agents I want to implement but I think this is wrong.
I now think that the pi-calculus will serve as a tool to formally specify the agents and their interactions in a process-calculus way and that I will use category theory for the implementation of the Domain-Specific Language which runs these pi-calculus specifications. It is interesting so see that the more I know and the more I learn the more I can picture and imagine and the clearer the road to my PhD gets.

Foundations of computation

Since I got in touch with the term of computability and computation I wondered what exactly is meant with it. I would now say that *computation is the manipulation of symbols according to specific rules.* More generally speaking the symbols are letters over an alphabet of a formal language and could be any e.g. the binary system where the symbols are just 0 and 1.
What intrigued me was why did the Old Gods (Turing, Church, Gödel) make such a fuss about computing natural numbers? I asked myself why in the world didn't they just use the binary system in which we can represent numbers?
The problem was my approach to computable numbers: the Old Gods never thought about the natural numbers in a binary system but in a purely mathematical sense defined by peano: the peano numbers. Those are inductively defined as being successors of the initial number zero. Thus 1 is successor of zero, 2 is the successor of the successor of zero and so on. Now they asked how one can perform basic arithmetic operations on such numbers? They asked how one can *compute* the addition, subtraction, division, multiplication of two peano numbers. This led to the invention of Gödel System T, Churchs Lambda Calculus and the Turing Machine. All of them had a different approach to computing the results but were shown to be of equal computational power.
While working on [**?**] (Dependent Types at Work) where the natural numbers were introduced in the style of peano it became clear for me that I did look at the whole problem from the wrong point of view. Now I better understand the approach of the Old Gods and with what they were struggling with. Still I have to think about how the problem of undeciability relates to these things.
What is clear now is that the computation over the natural peano numbers has the alphabet of zero, succ where the operations are implemented in the systems

mentioned above: The Turing Machine performs the computations in imperative steps by changing a global (infinite) memory and Church and Gödel follow a functional mathematical style using so called primitive recursion.

August 2016 17th

As already mentioned in the entry of August 2016 11th, the authors of [**?**] tell bad news because "most of the concepts are not constructive". After thinking about this I have the feeling that this implies that I have to approach my agent-based simulation from a constructive direction: both the agents and the simulation itself have to be *constructive*. It is still not very clear to me what that means but I guess that one can draw parallels to the peano numbers: I think the peano numbers are an inductive *constructive* way of, well... *constructing* the natural numbers and not just postulating their existence - each number has to be computed! We will see whether my intuition is right and how this will apply to ABM/S. After having read the entry on Constructivism (Mathematics) on

Wiki it is now a bit clearer. To quote Wiki "Thus the proof of the existence of a mathematical object is tied to the possibility of its construction.". This is exactly what happens with the pano numbers: they are constructed and thus they exist - they are computable!
I am now in Section 3 of [**?**] (Dependent Types at Work) and have decided to

first read through some more introductions to Agda and dependent types before continuing as I first have to understand more basic things about this theory.
Found a very nice lecture about *Interactive Theorem Proving for Agda Users* at

`http://www.cs.swan.ac.uk/~csetzer/lectures/intertheo/07/interactiveTheoremProvingForAgdaUse`
`html`.

August 2016 22nd

I am currently on holidays with my lovely girlfriend Sarah. We started on the 20th towards Denmark and are now in Sweden (camping at Stocken) and I will return on the 7th of September. Due to lots of free time I brought a bit of things to read:

1. Paper by Per-Martin Löf
2. Yellow Paper of Ethereum (TODO: cite)
3. White Paper of Ethereum (TODO: cite)
4. Handbook of Market Design (TODO: cite)

.1 Ethereum

Based upon a hint by a good friend on (2016 August 17th) I started exploring the idea behind Ethereum which immediately fascinated me. This is a topic I will definitely pursue as a private interest during my PhD Studies. I hope to be able to participate in a project or have the idea for one myself and maybe it can be combined with my PhD Studies (agents running in Ethereum?).
After having read the White Paper (TODO: cite) and the Yellow Paper (TODO: cite) of Ethereum, I am even more determined to participate in this huge new thing of the internet. I am very excited to get my hands on it as I feel its a very interesting playground allowing for a huge variety of opportunities both in learning and earning money.

To sum up what fascinates me is that Ethereum

> ... provide a massive boost to other peer-to-peer protocols by adding for the first time an economic layer.

This allows

> ... anyone to build what is essentially a command line application run on a virtual machine that is executed by consensus across the entire network, allowing it to modify a globally accessible state as its "hard drive".

This immediately spawned the following ideas in my mind:

1. Implement an Ethereum client in Haskell. This is an amazing way to improve my Haskell and finally apply it in a real-world Application which touches on a huge range of computer-science topics. I've seen that attempts to implement an Ethereum client in Haskell have been already made, I will look at them when I return from my holidays but I want to attempt it nonetheless as it will hugely improve my Haskell skills. It will be a huge challenge because its a very complicated protocol including the implementation of a Virtual Machine (Ethereum Virtual Machine, EVM) but fortunately everything is described formally and very precise in the Yellow Paper (TODO: cite). Things to implement:
   - Ethereum Virtual Machine
   - Proof-Of-Work Hashing Algorithm - should be done on GPU otherwise too slow
   - Communication with Network
   - ?
2. Learn and play around with Serpent the programming language of the EVM to implement contracts. I want to try to implement various types of auctions (batch, continuous, simultaneous ascending, sealed second price,...) and voting-systems.

3. Set up a private account in the public Ethereum Network.
4. Set up a public Node for participating in Ethereum and mining Ether.
5. Thinking about an interesting DAO (Decentralized Autonomous Organization) to implement in Ethereum

Also Ethereum could be the perfect Use-Case for applying my ABM/S to computational Economics:

- Simulate Ethereum: dynamics of ether or gas? Ethereum is a very itnersting agent-based system (peer-2-peer with an ecoinomic layer
- Simulating dynamics of DAOs in Etherum?

August 2016 23rd

After meditation and after looking through "Handbook of Marketdesign" (TODO: cite) it struck me: Why don't do I just use the model and simulation of my Masterthesis instead of keep looking for an awesome model which I still coulnd't find? The vorteile are

- I am already very familiar with the model, simulation and results.
- I am already familiar with the theory behind it by Geanakoplos (TODO: cite Leverage-Cycle by Genakoplos)
- It is complicated but simple enough to serve as a starting point to apply and proof my methods.
- It includes a few interesting aspects which can be proofed using my methods e.g. under which circumstances does trading stop?
- It can then be extended to a full-fledged framework as originally intended.

TODOs

Starting from now on I will include TODO-subsections which include well... things to do so that I have a small roadmap ahead. When a bullet-point in the TODOs is done then I will write an entry with the updated TODO list.

- Read the following chapters of "Handbook of Market-Design" in the given order: 2,3,12,10,13,16,17
- Read my Masterthesis again.
- Rework my Reseach-Proposal and include the approach of my masterthesis and think about how I can apply the formal methods to it - this will then be the basics of the initial meeting with my supervisors unless I will find a kick-ass idea regarding ABM/S of Ethereum

- Get basic understanding in Agda
  - Work through "Dependent Types at Work"
  - Look at Thorsten Altenkirchs Lecture "Computer Aided Formal Verification
  - Read online Lectures about Dependent Types: TODO
- Haskell Learning: Implement the core of my Masterthesis in Haskell as I know it at the moment (no Monads): the core is the replicated auction-mechanism where every Agent knows all others.
- Haskell Learning: Start implementing Ethereum Client.
- Alternative Topic of my PhD: ABM/S of Ethereum as a decentralized marketplace. Maybe these questions help a little bit (TODO: explain, see Handbook of Market-Design Introduction to Chapter 1) but unfortunately I know not enough about it but hope that I can get deeper into it soon.
  - How well does it provide thickness?
  - How does it deal with congestion?
  - How does it make the market safe and simple to participate in?

August 2016 24th

Yesterday night I had some (self-)doubts about the direction of my PhD, more specific the application of ABM/S to agent-based computational economics.
Doubts came up that I can properly understand and get into this field but after thinking about it a while, it became clear that ABM/S is - unless one does fundamental theoretical work of ABM/S itself - always applied in the context of another domain e.g. Physics, Computer Science, Economics,... I plan to do fundamental theoretical work in ABM/S itself (formal methods) but I will apply it in the field of agent-basec computational economics thus I definitely have to learn the basics of this field and become an expert in the specific subfield I plan to work in (equilibrium in trading-processes).
I chose the field of agent-based computational economics as I have already a bit of experience in it due to my Masterthesis and because I think it is a fascinating field and is of very importance to Economists to better understand dynamics of static postulates (e.g. equilibria).
The main question is, what exactly I want to research in this field of agent-based computational economics (ABCE)? The answer

> There exist Equilibrium Theorems in trading but no description of processes reaching them. I want to look at simulating such processes which may or may reach some steady state which may or may not be an equilibrium.

Also while thinking about this fact I came up with the thought:

> The Economists focus completely on their equilibria. Maybe this is the wrong way to go and we should abandon to try verzweifelt reaching equilibria but should only look at the dynamics of the processes and how they unfold - I believe the dynamics are of most importance as they really affect the world and not equilibria.

Nonetheless to abandon equilibria in favour of dynamic processes I must start with the very basics of equilibria to understand them, to follow the road of the economists why they are so obsessed with them and to know to talk about them and to encounter economists in discussions - I REALLY have to know the equilibrium theory (to be found in Microeconomics, which I should study in a basic form like the book I own: TODO cite). Also I should really know Market Microstructure because it focuses more on the dynamics of prices and of trading and I already got a nice book for this (TODO: cite)

Thus the plan is to start from my masterthesis. Take the model and simulation (continuous double-auction with Zero-Intelligence Traders) as they are without any change and develop and apply the methods to it and also do formal verification and proofing of various properties like convergence (termination) and deadlock free trading and buyer optimism ¿= seller optimism. The used methods will be as already mentioned Category Theory, Pi-Calculus, Haskell and Agda.

Then after having developed the methods and seen them in action then I can work towards a more general application of them to research and study the dynamics of trading-processes and abandoning the focus on equilibria. Until then I have to have studied severely ABCE, equilibrium and auction theory, have an overall basic understanding of Microeconomics, have a very good understanding of Market-Microstructure (the key properties e.g. volatility,...) and have studied basics of Market-Design.

I know that this seems extremely ambitious but I think it is a very good point from which to start my PhD adventure - we will see what the outcome will be in the end.

TODOs

I think I should abandon the idea to do something with Ethereum as a decrentralized marketplace in my PhD, thus abandoning it from my TODOs. Also I think I should attempt to use Monads in reimplementing my Masterthesis.
Thus the updated todos are:

- Read my Masterthesis again.

- Rework my Reseach-Proposal and include the approach of my masterthesis and think about how I can apply the formal methods to it - this will then be the basics of the initial meeting with my supervisors. Also include the thoughts of the entry to the diary of 2016 August 24th.
- Read the following chapters of "Handbook of Market-Design" in the given order: 2,3,12,10,13,16,17
- Get basic understanding in Agda
  - Work through "Dependent Types at Work"
  - Look at Thorsten Altenkirchs Lecture "Computer Aided Formal Verification
  - Read online Lectures about Dependent Types: TODO
- Haskell Learning: Implement the core of my Masterthesis in Haskell as I know it at the moment and trying to incorporate Monads: the core is the replicated auction-mechanism where every Agent knows all others.
- Haskell Learning: Start implementing Ethereum Client.

August 2016 30th

Having enough time in my holidays to think and reflect about the direction of the computational economics part of my PhD I seem to go in circles but I think I finally agreed with myself about the right approach.

The economic theory & model behind my masterthesis are too complex to start with. The definitive approach is now to start with the classical paper of Gode & Sunder (in which they introduced zero-intelligence (ZI) traders within continuous double-auctions) to develop the basic methods and tools on an already researched topic and then to gradually broaden to a more general framework (CDA variants) with focus on dynamics instead of equilibrium.

TODOs

- Read Gode & Sunders paper again (TODO: cite).
- Read Everything what you wanted to know about continuous double-auction paper again (TODO: cite).
- Read my Masterthesis again.
- Read the following chapters of "Handbook of Market-Design" in the given order: 2,3,12,10,13,16,17

- Rework my Reseach-Proposal and reformulate it to cover the above idea. This will then be the basics of the initial meeting with my supervisors. Also include the thoughts of the entry to the diary of 2016 August 24th. And include essence of entry to diary of 2016 August 30th.
- Get basic understanding in Agda
  - Work through "Dependent Types at Work"
  - Look at Thorsten Altenkirchs Lecture "Computer Aided Formal Verification
  - Read online Lectures about Dependent Types: TODO
- Haskell Learning: Implement the core of my Masterthesis in Haskell as I know it at the moment and trying to incorporate Monads: the core is the replicated auction-mechanism where every Agent knows all others.
- Haskell Learning: Start implementing Ethereum Client.

I also had a contemplation with Samuel about the direction of my PhD and he told me that "Never stop following your feelings and intuition. If you follow that road it cant be wrong." - thus I am on the right path as I always follow the feelings and the intuition which comes up during reflection on the topics.

2016 September 2nd

Black Swan

I've been reading the book "Black Swan" (TODO: cite) since about one week and I'm thrilled by it.
He does a fundamental critic on economics as a "science" which he claims it is not because it can't predict due to the extreme complexity and uncertainty (and randomness) of the world. The question is then whether computational economics and simulation are also a dead end or not?
On position 4289 he writes that "If you hear a "prominent" economist using the word equilibrium, or normal distribution, do not argue with him; just ignore him, or try to put a rat down his shirt.". I thought right away from the beginning that the author would make fun of equilibria (and normal distribution) because the world is just not in an equilibrium and not normal distributed and will never be. What does this imply to my PhD studies? It should encourage me to go away from equilibrium research towards simulation of dynamics and to look force a different approach to computational economics. Fact is: I don't want to do something unrelated to reality and purely academic - but is it really within my reach to make an impact? It is not (one never can plan to make an impact on the world, as shown in the Black Swan Book) but I can select my focus accordingly to increase the possibility of an impact. Thus I feel that I should focus on alternative approaches to computational economics which abandon equilibrium theories altogether and look only on the dynamics instead of equilibria - a thing which I already mentioned in the entry of the 24th of August 2016.

The author also attacks linear regression and r square: I also never understood what one gains from it as it is only a linear fit and can blowup terribly when compared to the real-world. I remember an economic workshop in which I participated where all presenters gave statistics how well their model fits linear regression. I felt it was just dead numbers, giving you absolution and a stamp on your model which says: *accepted by the (conservative) economics community* - but it all seemed so far away from reality.

The author also praises a few authors - I noted 4 from which I should definitely read a bit of.

- Karl Popper
- Henri Poincaré
- J.M. Keynes
- Friedrich Hayek

Hayek is a major proponent of the Austrian School of Economics, which I should definitely look at because it seems to be in opposition to the neo-classical (cambridge school) which emphasises mathematical models of equilibrium where the Austrian School focuses on TODO?

Also I am aware that I know very very little of economics to be in the position to be a competent critic of it.

2016 September 5th

On the ferry from Bergen to Hirtshals on my way home.

Black Swan finished

I've finished the Black-Swan book 2 days ago. The quintessence of it is basically an attack on the established economic theories and the people behind it. It all boils down that to a massive critique of the Gaussian Normal Distribution. Although it has its applications in the distribution of n̈ormalq̈uantities like height, weight, age of persons it fails completely in case of predicting probabilities of unbounded quantities like they are common in economics where those parameters can totally explode. Also the author says that history is totally unpredictable due to it being an extremely complex process thus also implies that prediction for economics is impossible. What is interesting is that the author does never go explicitly in discussing equilibria and their theories but only mentions them twice from where it becomes imminent that he thinks that equilibria are bullshit as well as gaussian distribution - this is what I've already expected at the beginning of the book.
This book has thus quite an implication for my Ph.D. as I have the feeling this author is right I cannot ignore what he says and continue as if nothing happened. Thus I have to question the use of the gaussian distribution and more specific the use and the pursue of equilibria in my applications. But still I do not know enough - but I just stumbled across a book of Mandelbrot (TODO: cite) in which he goes into analyzing markets and price-formation from a fractal point of view (which he invented). But what I think the author of the Black

Swan misses or never mentioned is that we all want too much: newer cars, newer phones, newer computers. We want it every year and we want it cheap. In fact the problem is greed, we must return to a more simplistic life-style and not live above what is ours.

Mandelbrot book started

This book is not about predicting prices or future economic trends but to explain and better understand the overall structure of price formation and volatility. Also a critique of the gaussian distribution and a huge proponent of power laws (see my Masterthesis). On position 160 (kindle) Mandelbrot claims that

biologists do research on the healthy body, physicists collide particles, meteorologists look into hurricanes but economists seem to be curious incurious.

Thus my implication: economists should rely much much more on simulation and simulate their models using computer power (Mandelbrot says in the introduction that ¨financial economics as a discipline is where chemistry was in the 16th century", which pretty says all. Also the author of black swan says that there rarely happens a feedback from the real-world to the models). This could be a more specific direction of my research (as already mentioned in other diary notes): looking into the dynamics of markets but with non-gaussian, fractal models. The result will not be a predictive price model but a system in which dynamics which resemble those on markets can be analysed and better understood to be better prepared in case of crashes. Mandelbrot himself sums it best up: ¨basic research into the dynamics of pricing and volatility in a global marketplace gets short shrift." - this is what I want to do in my simulations.

A 1st semester Roadmap

I still need to read and learn much more about economics, but how much exactly and which kind of direction? I think it is enough to cover the basics and then go deep into the subjects of equilibrium theory, market micro-structure, the fractal models of markets and agent-based computational economics. Thus the very basic approach in the 1st semester is doing literature research and do prototyping of the new formal agent-based simulation methods. During literature research the goal is to find a paper to start with, which allows to apply the methods developed and then to do further research based upon that paper in which I can incorporate my ideas of dynamics instead of equilibria and fractal/creative randomness/complexity from simplicity instead of predictability, Gaussian distribution and complexity from complex models.

- Literature Research in Economics
  - Basic overview by reading the "Understanding Capitalism" Book (TODO: cite)
  - Basic understanding in equilibrium theory by studying specific chapters of microeconomics book (TODO: cite)
  - Basic understanding of market micro-structure by reading specific chapters of the Market Micro-structure book (TODO: cite)
  - Find and read papers in the fields of
    - Fractal Market Models inspired by Mandelbrot
    - agent-based computational economics
    - Combinations of both.
- Method development and experiments in Computer Science
  - Get better understanding of Agda
  - Get deeper into Haskell and monadic programming
  - Get basic understanding of Category Theory
  - Get introduction into Computer Aided Formal Verification

TODOs Update

Note that these TODOs should be done till the 26th of September because I expect to meet my supervisors then - maybe there is time until the 1st of October but better earlier than too late (Research Proposal!)
I think I will abandon the whole Ethereum stuff as I will need all the time for the preparations and the Ph.D. itself - and because I also want a little bit of free-time without comptuers.

- Read Mandelbrot Book (TODO: cite)
- Read Gode & Sunders paper again (TODO: cite).
- Read Everything what you wanted to know about continuous double-auction paper again (TODO: cite).
- Read my Masterthesis.
- Read the following chapters of "Handbook of Market-Design" (TODO: cite) in the given order: 2,3,12,10,13,16,17
- Rework my Reseach-Proposal and reformulate it to cover the above idea. This will then be the basics of the initial meeting with my supervisors. Also include the thoughts of the entry to the diary of 2016 August 24th. And include essence of entry to diary of 2016 August 30th.
- Get basic understanding in Agda
  - Work through "Dependent Types at Work"
  - Look at Thorsten Altenkirchs Lecture "Computer Aided Formal Verification
  - Read online Lectures about Dependent Types: TODO
- Haskell Learning: Implement the core of my Masterthesis in Haskell as I know it at the moment and trying to incorporate Monads: the core is the replicated auction-mechanism where every Agent knows all others.

2016 September 6th

Currently on the Wewelsburg on a stop on my way home. Of course I can't stop thinking about a potential application of my formal methods in agent-based computational economics.

My hypothesis is that a finite number of interacting agents following very simple rules in placing bids and asks lead to price dynamics that resemble power law distributions invented by Mandelbrot - price formation is unpredictable but the dynamics follow a rule and emerge out of the interaction of the agents which form a market. I would need to build a model of the agents and interactions among

them and then implement the simulation. I should also look at continuous double auction and continuous batch auction and compare their results. A big question is what the role of equilibrium (theory) in this approach is.

I want to abandon the idea of a simulation which will hunt for equilibrium as I became convinced by the books of black swan and Mandelbrot that this is infeasible arcane magic, something I already thought right away from the beginning of my masterthesis and which I also discussed with Prof. Vollbrecht. I MUST NOT blindly follow a scientific road in which I don't believe and I don't believe in equilibrium in economics as the world is totally obviously NOT going towards equilibrium.
Of course the process of reaching it must not be confused with equilibrium itself and no process has been given so far thus meaning we could be on the way to reaching it and will at an unknown time in the future because we don't know the process - but that lies no in the field of belief and I don't believe thus abandoning the search for it. Also the conditions of the system in which equilibrium may unfold are always a changing: traders enter and leave at various points thus shuffling all new as one knows regarding system dynamic.

But i have to be honest: I find the concept of equilibrium beautiful and it is still fascinating to look at the dynamics of how equilibria form in a process: how long does it take, how efficient is it, is it even reachable, ... maybe i can nonetheless incorporate it in SOME way in the new approach: see if and what Mandelbrot says about them and incorporate that then.

So the overall direction is clear and will be one of two ways (or a combination of both, as both topics are extremely interesting):

1. Research equilibrium processes: look at how and under which conditions equilibrium is established, what properties the agents have, how long it takes, if it is feasible,...

2. Research price dynamics and trading dynamcis: ignore equilibrium and only look at the dynamics of trading processes where tools developed of Mandelbrot should be used instead of neo-classical economics (gaussian distributions, equilibrium theory)

I've written an email to Martin Summer asking him a few questions about the Black Swand and Mandelbrot Books regarding Gaussian distribution and equilibrium theory - I am curious what he will answer. I will give a summary of the questions and his answers when he answered.

I just found "The future of agent-based modelling", Santa Fe Bulletin Winter 2006. It seems that what I wanted to do already exists in the form of a software package called Santa Fe Artificial Stock Market SF-ASM. According to this Bulletin this software package challenged the idea that financial markets are

in equilibrium. The paper also says that the SF-ASM is currently not under development - maybe I can take a look at it. Also this paper talks about the zero-intelligence traders invented by Gode & Sunder and how useful they are, as already shown in the classical paper by them - maybe a hint to go start again with this paper?

I also did a bit of paper research and found a few interesting papers I should read before reworking my research-proposal.

- Agent-based Computational Economics. A Short Introduction. MATTEO G. RICHIARDI (TODO: cite)
- Agent-based modeling and General Equilibrium, Lastis symposium, ETHZ, September 11 2012, Antoine Mandel (its a presentation)
- MANDELBROT AND THE STABLE PARETIAN HYPOTHESIS, Eugene Fama (TODO: cite)
- THE VARIATION OF CERTAIN SPECULATIVE PRICES, BENOIT MANDELBROT (TODO: cite)
- Out-of-Equilibrium Economics and Agent-Based Modeling, Brian Arthur (TODO: cite)
- The emergence of a price system from decentralized bilateral exchange, Herbert Gintis (TODO: cite)

The paper "The emergence of a price system from decentralized bilateral exchange" (TODO: cite) was mentioned in the paper about formal verification of economics (TODO: cite) as a paper which made lot of fuss but then an error was found - a perfect example for Computer Aided Formal Verification, maybe this is a nice paper to start from?

Starting Papers

I introduce hereby a new section which lists all papers which could act as a starting point for the application of the formal methods and agent-based simulation. So far we have the following papers:

1. Allocative efficiency of Markets with Zero-Intelligence Traders, Gode & sunder (TODO: cite)
2. The emergence of a price system from decentralized bilateral exchange, Herbert Gintis (TODO: cite)

Updated TODOs

Added SF-ASM entry and for reworking my research-proposal i should read through all research-diary entries and incorporate the essence of them. Added entry to read the enumerated papers.

- Read Mandelbrot Book (TODO: cite).
- Read the papers enumerated in entry of 2016 September 6th.
- Look into SF-ASM.
- Read Gode & Sunders paper again (TODO: cite).
- Read Everything what you wanted to know about continuous double-auction paper again (TODO: cite).
- Read my Masterthesis again.
- Read the following chapters of "Handbook of Market-Design" (TODO: cite) in the given order: 2,3,12,10,13,16,17
- Rework my Reseach-Proposal and reformulate it to cover the above idea. This will then be the basics of the initial meeting with my supervisors. Also include the thoughts of the entry to the diary of 2016 August 24th. And include essence of entry to diary of 2016 August 30th. Read through ALL the research-diary entries and incorporate the essence of them.
- Get basic understanding in Agda
  - Work through "Dependent Types at Work"
  - Look at Thorsten Altenkirchs Lecture "Computer Aided Formal Verification
  - Read online Lectures about Dependent Types: TODO
- Haskell Learning: Implement the core of my Masterthesis in Haskell as I know it at the moment and trying to incorporate Monads: the core is the replicated auction-mechanism where every Agent knows all others.

2016 September 7th

SF-ASM and equilibrium

I couldn't stop thinking about that the SF-ASM challenged the idea that financial markets are in equilibrium. Why did research stop there? Is equilibrium a too holy grail to put it down even if it is misguiding? Is there new research going on in this direction? Maybe I can start from this point?

An idea for a model & simulation

After thinking about the whole thing while driving home from Germany I came up with the simple idea that the real problem may be the one that every trader just wants to make a profit instead of really owning a share of a company. Also I asked myself electronic trading has changed how the markets behave, whether they exhibit now different dynamics in volatility as shown by Mandelbrot. When boiled down to the very basics of how profits are made then there should be no difference: one buys at low and sells at high. Thus the idea is the following:

> A considerable amount of the traders don't do long term trading, they are not interested in owning shares for a long period (say months, years) but are only interested in the quick profits when selling at a higher price than bought and also all trading algorithms (should) work the same. My hypothesis is that this quick trading creates those unpredictable dynamics and fat tail variance of the prices (does this also imply that markets are never in equilibrium?). Also because trading algorithms work basically the same way the same dynamics should be dominant also despite the massive use of electronic and automated trading.
> To validate the hypothesis the goal is to develop a suitable model and simulate it. The approach which comes to my mind is to model zero intelligence agents in continuous double auctions with time explicitly modelled. The agents are divide into long- and short-term traders where short-term traders jump on price-changes very quickly where long-term traders don't. Of interest are then the price-dynamics and equilibrium under different parametrisation: different distribution of short- and long-term traders (also changing over time), "quickness" of short-term traders,...

The approach would then be to 1. develop model, 2. transform the model to a formal representation (pi calcumus/category theory/actor model), 3. implement the simulation in Haskell & Agda and 4. do computer aided formal verification in Agda.

2016 September 18th

Currently in Glastonbury (lots of strange people). Here a sum up of the last days in which I was too busy to write down anything.

Martin has replied in email. He didn't read the Black Swan book but did note it as it caused some stir in the economics field but he does not like the style of it as he says it fishes for sensation - I partly agree but Taleb seems to bring in real evidence that economics models are just wrong (something Mandelbrot confirms in his book). Martin did send me a few interesting papers as I asked him for ideas:

- Hot and Cold Seasons in the Housing Market - L. Rachel Ngai Silvana Tenreyro
- GETTING AT SYSTEMIC RISK VIA AN AGENT-BASED MODEL OF THE HOUSING MARKET - John Geanakoplos et al
- LEVERAGE CAUSES FAT TAILS AND CLUSTERED VOLATILITY - Stefan Thurner, J. Doyne Farmer and John Geanakopolos
- Contagion in Financial Networks - Paul Glasserman and H. Peyton Young*
- When Bitcoin Grows Up - John Lanchester
- HIGHER-DIMENSIONAL MODELS OF NETWORKS - DAVID I. SPIVAK
- Housing and Macroeconomics - Monika Piazzesi, Martin Schneider

I will read through these papers as soon as possible and I think if not the Mandelbrot Book Idea (see below) will do, then something out of those papers will be interesting enough.

My former Professor Mr. Vollbrecht did provide me with an interesting hint, that I should look at the Journal of Artificial Societies and Social Simulation as they seem to have an "alternative" way of looking into things and have a very broad interest. It is to be found at `http://jasss.soc.surrey.ac.uk/JASSS.html`

I continued reading the Mandelbrot Book and got a few good quotations.

> The fastest way to simplify things is to spot the symmetries, or invariances - the fundamental properties that do not change from one object under study to another. (Position 1957)
>
> A lesson arises from this: never hurry and never publish any result based on a single tool. (Position 2830)
>
> Many a grand theory has died under the onslaught of real data. (Position 2840)

He also makes it very clear that economics is making wrong assumptions, which are:

1. Homo economicus is rational and self-interested - wrong, see the bubbles and bursts of the 90s.
2. Price variations follow the bell curve - wrong, see widely accepted by Mandelbrot and many others since the 1960s

3. Price variations are i.i.d. - evidence for short-term dependence has already been mounting and also evidence of long-term dependence.

Because the book is from 2004 I have a few questions regarding the up-to-date state of it:

- Is there research going on at the moment investigating fractal price changes today (opposed to changes mandelbrot looked at in the 60s)?
- How are fractal prices showing up today in stock markets and do they differ? is there a change detectable due to electronic trading? if no then why? what has stayed the same (buy low and sell high?)?
- Are the price changes also present on a millisecond scale?

Mandelbrot talks in his book about a model which he developed an which was tested by his students Laurent Calvet and Adlai Fisher in their doctoral thesis and proofed to be true. Thus my dream would an agent based simulation which is able to mimic the price dynamics of the model as an endogenous property of agent behaviour. The key would be to use use multifractal time to create busy and smooth market activity (see mandelbrot diagram starting from pos 3099). The question is: is it not too complicated? Is it possible? Hasn't it already been done?
After a quick googling I found out that the most recent model Calvet and Fisher have developed is called Markov switching multifractal `https://en.wikipedia.org/wiki/Markov_switching_multifractal`

2016 September 19th

Finally finished with the Mandelbrot Book and got a few more quotes

> Economics has no intrinsic time scales. in fractal analysis time is flexible - expanding and contracting.

Maybe this is interesting for modelling agent actions: many in dense times, few in stretched times.

> When examining price charts we should guard against jumping to conclusions that the invisible hand of Adam smith is somehow guiding them.
>
> Forecasting process may be perilous but you can estimate the odds of future volatility
>
> If there is one message I would wish to survive this book, it is this: finance must abandon its bad habits and adopt a scientific method (pos 3648)

Also Mandelbrot last chapter talks about a trader who looks at a heterogeneous market consisting of traders with different time scales coming together at various points in transactions which creates the multi-fractal behaviour of the market. This was also something i had in mind but of course not so articulated.

So the book was VERY interesting, and very fruitful and inspiring and giving a good hint at some workings in economics, finance and trading. Mandelbrot himself said that we are still very very far from REALLY understanding how markets work and that lot of research needs to go into that and that finance needs to change.
So what I take out of this book is the critic of the orthodox models of economics and finance and the inspiration that markets follow a multifractal nature. As already said in the last entry it would be interesting to have an agent-based simulation which generates prices which follow such a multi-fractal volatiliy - the question is how, if it is possible/or not too complicated (which I fear it is), if it hasn't been already done and what model to follow (e.g. Markov switching multifractal).

Thomas Schwarz (a former Student- and Working-colleague) told me about Zotero, which is an open-source reference management software to manage bibilographic data and related search materials (wiki). I definitely should manage all my papers in this one for a better search and find and management.

Updated TODOs

Finished Mandelbrot, combined all paper-entries into one, added Zotero entry.

- Read interesting papers in my folder - need to enumerate them in a following entry.
- Put ALL printed papers of my folders into Zotero.
- Read my Masterthesis again.
- Read the following chapters of "Handbook of Market-Design" (TODO: cite) in the given order: 2,3,12,10,13,16,17
- Rework my Reseach-Proposal and reformulate it to cover the above idea. This will then be the basics of the initial meeting with my supervisors. Read through ALL the research-diary entries and incorporate the essence of them and make a list of the research-ideas found so far in my diary.
- Get basic understanding in Agda
  - Work through "Dependent Types at Work"
  - Look at Thorsten Altenkirchs Lecture "Computer Aided Formal Verification
  - Read online Lectures about Dependent Types: TODO

- Haskell Learning: Implement the core of my Masterthesis in Haskell as I know it at the moment and trying to incorporate Monads: the core is the replicated auction-mechanism where every Agent knows all others.

2016 September 21st

A paper: Contagion in Financial Networks (TODO: Cite)

It is more or less an overview of current research and models in this field and reviews it and emphasises weaknesses of the models and proposes extensions. It gives also an overview of Open Problems where the two I would be interested in are 1. How the links between institution are formed (dynamic network formation) and 2. how to deal with network opacity:

> The conventional view is that institutions establish links with one another as a way of diversifying risk and facilitating intermediation. While this is certainly true, we would argue that a realistic model of network formation must include other factors. In particular, one must acknowledge that links between financial institutions are often created in a decentralized fashion within particular lines of business, such as commercial lending, foreign exchange, derivatives trading, repo desks,and the like. Building realistic models of the resulting dynamics will require a high degree of institutional knowledge and a clear under- standing of the incentives faced by the individuals who are forming (and severing) these links. We believe that this is one of the most important challenges for future research in this area.
>
> Much of the literature on financial network models presupposes complete information about the network. We have argued that network opacity is a first-order concern for agents within the network, for regulators monitoring the network, and therefore for researchers developing models. More work is needed on inference from partial observations of network data and on understanding how opacity itself may contribute to contagion.

The topic of contagion in financial networks is a very suitable one because

- I am already familiar with advanced network concepts through my Masterthesis.
- It should be very well suited for ABM/S due to nodes and connections resemble agents and their messages/communication channels.
- It is a real world problem.
- It is probably very well applicable to Pi-Calculus and category theory

- Surely I can get very good support by martin summer throughout the Ph.D. as he also has written a paper on this topic.
- It feels much better than the Mandelbrot topic

A paper: The emergence of a price system from decentralized bilateral exchange

I've been curious about this paper since a few weeks but haven't had time to read it yet. I came across it when reading the paper "Dependently typed programming in scientific computing". The authors of that paper mentioned that people trying to reproduce Gintis paper - which made a lot of fuss due to its claims of having found a mechanism leading to equilibrium without central authorities - found that the simulation had some fundamental programming error, leading to the (wrong) results (which could be prevented by using computer aided verification, but then the results would have been different and probably not being worth the publication). So I was curious about the initial idea of this Gintis guy and wanted to read the original paper.

After googling for Herbert Gintis (I didn't know anything about how that guy was) I found a very interesting customer review by him on amazon.com for the book "Emergent Macroeconomics: An Agent-Based Approach to Business Fluctuations" (TODO: cite). Here is what he says about it

> The theory of macroeconomic fluctuations has been a pathetic mess for a long time. Indeed, forever. The central model of the economy, the Walrasian general equilibrium model, is a purely equilibrium model, and no one has been able to derive an out-of-equilibrium mechanism of price and quantity adjustment that renders market equilibria dynamically stable, despite more than a half century of trying (see my paper, "The Dynamics of General Equilibrium", Economic Journal 117 (2007):1289-1309 for details and a proposed solution), and despite blistering attacks from within the economic establishment. See, for instance, Alan P. Kirman, "Whom or What does the Representative Individual Represent?" Journal of Economic Perspectives 6 (1992):117-136, and Franklin M. Fisher, Disequilibrium Foundations of Equilibrium Economics (Cambridge, UK: Cambridge University Press, 1983). In a recent paper (Herbert Gintis, "The Dynamics of General Equilibrium", Economic Journal 117 (2007):1289-1309) I showed using agent-based modeling techniques that the problem with the Walrasian model lies in the assumption that agents never interact, but rather that each makes decisions independently from a given system of prices. I also showed that the assumption that there exists a price structure accepted by all agents in the economy ("public prices") accounts for the chaotic nature of disequilibrium in the standard Walrasian disequilibrium models.

Instead, when each agent has his own set of "private prices," even though market competition leads to a very low standard deviation of private prices (I call this situation "quasi-public prices"), such prices tend strongly to equilibrate the market system in the long run. Finally, I showed that plausible learning processes in the economy lead the system sporadically to make large excursions from equilibrium even in the absence of any global stochastic shocks (I called these "local resonances").

Because of the lack of dynamics in the standard Walrasian model, macroeconomic theories that depend on this model must perform massive simplifications in order to investigate out-of-equilibrium behaviour. The reason these models are such a mess is that they take it for granted that public prices exist (they do not) and that we can analyse the market economy as if individuals never interact, but rather interaction only with private prices. This, of course, is completely incorrect, as I explain above. However, with this assumption, it is clearly permissible simply to aggregate all economic actors of the same type into a "representative agent" having the average characteristics of that type of agent. From this is born the Keynesian consumption, investment, and government sectors, from which the standard Keynesian models flow. For the rational expectations macro models, we have a similar aggregation, with the completely crazy assumption that an aggregate "representative agent" will satisfy the condition of "rational expectations" theory, as though the aggregation of "rational agents" is prima facie an aggregate rational agent. The intellectual value of these assumptions is rather meager.

This fine book, which was in preparation at the time of appearance of my Economic Journal paper, is quite in agreement with my findings, laying blame on the "representative agent" assumption, and using agent-based modeling (abm) to investigate macroeconomic dynamics. However, whereas I took individuals as the unit of analysis, the authors allow firms to fill this role. They use empirical data on within-industry firm heterogeneity to model the population of firms, and assume asymmetric information among firms. This leads them to a financial accelerator model of financial fragility with great similarity to a model proposed by Greenwald and Stiglitz in 1993 (Bruce Greenwald and Joseph E. Stiglitz, "Financial Market Imperfections and Business Cycles", Quarterly Journal of Economics (1993):77-114). Finance is central in their model because the absence of forward markets forces firms to rely on credit to finance investment that matures only across time periods.

Based on careful industry research, the authors' abm is populated with firms whose size distribution take the form of a power law density (Zipf's Law), and firm growth rates follow a Laplace (double exponential) rather than a normal distribution. Such a distribu-

> tion has 'fat tails' that imply more instability than in a system with normally distributed growth densities. Indeed, they show that normally distributed shocks give rise to power law distributions and a Pareto shaped firm size distribution. This is a quite nice finding, and surprising given the degree of aggregation of their agent-based economy (they assume only two sectors, firms and banks, and no individual agents). Clearly individual interactions underlie the power law assumptions concerning firm size and the Laplace distribution of growth rates.

So he is a quite critic of equilibrium theory and classical economics - this sounds intriguing.

2016 September 22nd

A new and very interesting idea

Finally read the Paper of Gintis (TODO: Cite). It is a very interesting paper as it doesn't provide a very complicated model but is a very simple mechanism with interesting outcomes. Probably the most interesting thing is the use of *mutation* of agents behaviour like in Genetic Algorithms (Gintis cites a paper on GA). Also it is an approach of explaining the dynamics of an equilibrium process, which I am very interested in.
Thus this could be a very interesting thing to follow as the mutation could be interpreted as the agents change and adaption of behaviour. Also I am familiar with basics of evolutionary algorithms and optimization so this would be another idea: to incorporate evolutionary algorithms and techniques into my work. So this paper could make up a very good starting point for a research-direction: "evolutionary strategies in agent-based simulation of dynamics of equilibrium processes in decentralized trading"

- Evolutionary Strategies - allow to introduce a kind of unpredictability in behaviour with a natural motivation of the use: agents adopting their trading strategies to those of other, more successful agents or just adopt them because the "think" it will be better (theoretically speaking: approaching and leaving local minima).
- Equilibrium processes - so far how an equilibrium is approach is not understood and no suitable theory has been presented. I think it is a very interesting topic to pursue and VERY well suited for agent-based simulation
- Decentralized trading - In earlier days trading was always a kind of barter without any central authority coordinating prices. Also when looking at a stock market the prices emerge out of a "bartering" process without a central authority calling out prices. Thus it seems suitable to drop the

assumption of a central authority and look for equilibrium dynamics in decentralized trading only.

I should finally read the book "Debt: the first 5.000 years" by David Graeber, I think it will be a good inspiration.

The principle direction of my research

It is now clear to me that when doing computational economics I could follow down one of two roads:

1. Orthodox Economics: complicated but unrealistic models, continuous functions nice behaving functions, nice-behaved mathematics, static equilibrium models with centralized institutions, Gaussian-distributions, individuals act rational but are egoists
2. Progressive Economics: sceptic about equilibrium, looking for equilibrium dynamics, fractal nature of reality instead of mediocrity, individuals act irrationals but are much more willing to cooperate

As I am very sceptic about the nice-behaved mathematics, complicated models, continuous nice behaving functions - the world does not work like that - I think that I should follow the road of progressive economics.

Updated TODOs

Added two books: "Debt: the first 5000 years" and "Understanding Capitalism"

- Read the book "Debt: the first 5000 years" by David Graeber (TODO: cite)
- Read the book "Understanding Capitalism" (TODO: cite)
- Read interesting papers in my folder - need to enumerate them in a following entry.
- Put ALL printed papers of my folders into Zotero.
- Read my Masterthesis again.
- Read the following chapters of "Handbook of Market-Design" (TODO: cite) in the given order: 2,3,12,10,13,16,17
- Rework my Reseach-Proposal and reformulate it to cover the above idea. This will then be the basics of the initial meeting with my supervisors. Read through ALL the research-diary entries and incorporate the essence of them and make a list of the research-ideas found so far in my diary.
- Get basic understanding in Agda

  - Work through "Dependent Types at Work"
  - Look at Thorsten Altenkirchs Lecture "Computer Aided Formal Verification
  - Read online Lectures about Dependent Types: TODO

- Haskell Learning: Implement the core of my Masterthesis in Haskell as I know it at the moment and trying to incorporate Monads: the core is the replicated auction-mechanism where every Agent knows all others.

Re-read Gode & Sunders classical paper

It is interesting to re-read a paper after about 1.5 years when one has learned so much since then. I remember struggling with the paper back then mainly because of the unfamiliar vocabulary but now it was a very nice read.

Still I find their results very interesting and convincing but I have to object...

> In the first version, ZI traders were subject to the budget constraint: if they generated a bid (to buy) above their redemption value or an offer (to sell) below their cost, such actions were considered invalid and were ignored by the market. In other words, the market forbade traders to buy or sell at a loss because then they would not have been able to settle their accounts.

Gode & Sunder showed that using ZI with constraints is enough to raise the allocative efficiency to nearly 100% and performing nearly as good as human traders. They argue then that imposing market-constraints on the agents which are irrational (zero-intelligent) is enough to let equilibrium be reached.
The problem is (as far as I understood markets) that to impose such constraints on traders is unrealistic in reality as neither the redemption value (utility) of an agent is publicly known nor does the central trading authority (stock exchange?) does not have access to account balances. Thus this result is theoretically very interesting but in my opinion too far from reality. We must find mechanics how agents adopt and change thus being non-zero intelligent. I believe that very simple adoption/mutation/changing rules are enough (buy low, sell high?) for equilibrium prices to emerge - thus following the direction of Gintis.

2016 September 27th

Moved in at Melton Hall in Nottingham and got accustomed to the University Campus. Yesterday I had a short and formal chat with my supervisor (Peer-Olaf and Thorsten). It was very nice seeing them finally in person - both are very relaxed and I think they will make very very good supervisors. Peer-Olaf told me to calm down, to relax, to take the first month easy, that I don't have to finish my PhD within one month. So I try to do that ;)

Today is the first day at the office, the AgentLab (C50) where I will be in office with 3 other guys from Russia, India and China.

Reflections on an extension of Gintis Model

I have been constantly thinking about an extension to the Gintis Model of decentralized barter. Here are the thoughts

- The approach should be more empirical: looking how agents trade and act in reality, in real markets. instead of relying on artificial unrealistic models
- I also want to compare continuous double auctions (CDA) and continuous batch auctions (CBA) because i think the latter one is more fair and a remedy to HFT
- Also look at how the system behaves when introducing disturbances or shocks to the stable prices. thus the prices are in some kind equilibrium prices but can change when the system us shocked
- Also add the notion of fractal time derived from mandelbrot, this could give rise to interesting phenomena
- No global adaption of behaviour but only from local information: if a seller trades with a buyer the will both adjust their future prices after the trade towards the half-way between the their matching prices. =¿ only infer information which is indeed available, no global prices and utilityinformation is available
- Also incorporate the case when supply is NOT equals demand
- Look at real trading strategies used by stock exchange traders
- Starting from ZI agents which mutate their selling-/buying utility limits
- Also of importance: cost of production

A possible working-title could be: "Price formation in decentralized trading in CDA and CBA" 1. design an empirical agent-based model using evolutionary strategies and real world assumptions and hypotheses (buy low, sell high). apply pi calculus and category theory for formal specification 2. develop hypotheses and test them using the formal specifications as far as possible 3. implement and verify the simulation using haskell and agda 4. test the hypotheses against the results of the executing simulation.
Possible papers could be: 1. the model itself and results 2. formal specification of models in ACE 3. pure functional ACE simulation implementation 4. ACE verification.

Another direction to look for: Cezar Ionescu

Thorsten Altenkirch told me about a researcher called Cezar Ionescu which does Functional programming, dependently-typed programming in the context of economics. He teaches at Oxford and is to be found at `https://www.conted.ox.ac.uk/profiles/cezar-ionescu`. I've already read a paper by him a few weeks ago: *Dependently-typed programming in scientific computing: Examples from economic modelling* [**?**] which mentioned Gintis error and from where I got interest in Gintis stuff [**?**]. This would be a perfect match to combine both: Gintis & Ionescu.

2016 September 28th

A A new paper building on Gintis: "A functional framework for agent-based models of exchange" [?]

So the threads seem to run together: Cezar Ionescu has released an updated and refined model of the one presented by Gintis in the paper [**?**].
This could be the starting point of my research as the paper seems to be exactly what I wanted to do.

This paper ([**?**]) corrects the original claim in [**?**] (or my misinterpretation) that Gintis Model in [**?**] was wrong - the authors which tried to reimplement the model made a mistake due to a poor interpretation of the original model. Thus it seems the model of [**?**] is a good way to go.

2016 September 30th

Having read the paper [**?**]. I did not understand all the details but the overall goal was clear: provide a *functional* framework for agent-based models of exchange exactly what I wanted to do/what I am interested in, where the most striking section was:

> In theoretical and computational economics, the relationship between the notion of equilibrium and models of exchange is a subject of ongoing research [19,20,36,40]. In this context, a natural question is whether it is possible to identify economically sound trading policies – agent-specific offer and demand policies and agent-independent trade-resolving policies – which realize equilibrium allocations through sequences of elementary bilateral trades. The question is interesting because attempts at explaining equilibrium allocations and prices as stationary states of sequences of bilateral interactions between agents have not, so far, been convincing. In

> particular, we do not know economically plausible bilateral trade mechanisms which provably lead (arbitrarily near) to equilibrium prices and allocations in a finite number of interactions and for all initial allocations and utility profiles that guarantee existence of Walrasian equilibria. The question of whether it is possible to identify economically sound trading policies which realize equilibrium allocations through sequences of elementary bilateral trades can only be studied in a well-defined context. This has to specify the offer and demand policies of the agents, the trade-resolving policy and the trading sequences that is, which agent pairs are interacting and the order of interaction. In the following sections we introduce the elementary notions which are needed to formulate such context.

Thus having now a framework AND a model, the best thing would go now straight into implementation, which hasn't yet been done and would be the perfect starting point for my PhD:

> Some of the specifications introduced in Section 3 can be directly translated into Haskell data types or type classes. A translation of the whole framework into an executable programming language, however, would require a dependently typed language and go well beyond the scope of this article.
> Some of the specifications introduced in Section 3 can be directly translated into Haskell data types or type classes. **A translation of the whole framework into an executable programming language, however, would require a dependently typed language and go well beyond the scope of this article.**

2016 October 1st

Aim of my PhD

After reflecting on the whole thing a bit I've started to rewrite my research-proposal to cover the topic from the direction of Gintis models [**?**] and [**?**].

2016 October 3rd

A paper: "The dynamics of general equilibrium" by Herbert Gintis [**?**]

This paper is a more complicated and comprehensive model of [**?**] which can be considered to be a kind of complete economy with firms producing goods, agents being employed by the firms,... According to Gintis:

> The article presents the first general, highly decentralized, agent-based model of the dynamics of general equilibrium.
>
> Moreover, for the first time, we know something substantive about the dynamic properties of the Walrasian system. **They are nothing like tatonnement.**

Gintis also shows that when using public prices called out by a central authority leads to highly chaotic prices with extreme volatility. He claims that

> Economic theory assumes public prices without justification. Public prices do not generally exist and equilibrium public prices cannot even be calculated in an economy of any appreciable size.

This is a fundamental critique of the equilibrium theory of modern economics and inherently of many models built upon this notion of Walrasian tatonnement using public prices.
I have the feeling that I have now arrived at the right direction, the right paper, the right model. This is what I was looking for - and thank god, it has already been done by highly competent men (e.g. Gintis) and I can start following that road now.

An interesting article of Gintis on Traditional Economics and new approaches to it

In his article, to be found at `http://evonomics.com/new-economics-with-tradtional-economics/`, Gintis argues for a new way of economics, where Behaviour & Evolutionary Sciences combine with Traditional Economics to form a new understanding of economics which tries to understand the dynamics of economic processes e.g. equilibrium.

A remarkable quote is the following:

> At least since the end of World War II economists have analyzed market failures in the terms described above. The recommended solution to market failure has always been the same: government intervention to replace markets or to regulate markets in the public interest. **But why would anyone seriously believe that the government would actually do what is in the best interest of efficiency and justice?** This belief, many years ago when I was a graduate student, was simply a precondition for admission to the inner sanctum of professional economics. I, on the other hand, found it about as reasonable as belief in reincarnation or flying saucers.

2016 October 5th

Tomorrow I have the first official supervision meeting with Peer-Olaf (and I guess Thorsten will also join us?). I've been programming in Haskell again and I found that I have forgotten many things and that I have to get used to Monads again. Most important is that I have to focus now on a good research-proposal so that I know what to talk about tomorrow with Peer-Olaf.

Updated TODOs

Focused TODOs to most important things

- Rework my Reseach-Proposal and reformulate it to cover the above idea. This will then be the basics of the initial meeting with my supervisors. Read through ALL the research-diary entries and incorporate the essence of them and make a list of the research-ideas found so far in my diary.
- Haskell learning by example
  - Implement Conways Game Of Life using Monads
  - Implement other nice games/simulations using Monads
- Get basic understanding in Agda
  - Work through "Dependent Types at Work"
  - Look at Thorsten Altenkirchs Lecture "Computer Aided Formal Verification
  - Read online Lectures about Dependent Types: TODO
- Read the book "Understanding Capitalism" (TODO: cite)
- Read the book "Debt: the first 5000 years" by David Graeber (TODO: cite)

B How to get a PhD-study at Nottingham

The idea to do a PhD at Nottingham came up during the visit of Nottingham for the Leverage-Cycle Project which was 21st - 23rd May 2015. I was part of a small team consisting of Thomas Breuer, Martin Summer, Hans-Joachim Vollbrecht and me and our goal was to demonstrate Simon Gächter our software for running a specific type of economic experiment with real agents. After 3 sessions we had him convinced and he agreed to join in a research-cooperation in which the goal was to get a grant to further study the equilibrium theory of Geanokoplos. The idea was that in the funding of the research-project a doctoral study position should be included which would be taylored towards me so I could come to Nottingham and do a PhD in computer-science but in the context of the Leverage-Cycle Project.
The problem of getting such research-project grants is that it is a quite long process and the outcome is very unsure and it is highly probable that one gets rejected. Thus when starting the preparations for a PhD study at Nottingham in October I had to search for a 'Plan B' covering the case of the rejection of the research-grant. Also I tried to make use of our connections to Nottingham so I first contacted Martin Summer who gave me contact details of Uwe Aickelin, the Head of the Computer School which has his research-interest in Agent-Based Modeling/Simulation and Data-Mining. I wrote him an e-mail about a potential supervision and he quickly replied tell me that he will leave Nottingham and move to the chinese campus of the University of Nottingham but referring me to Peer-Olaf Siebers who is also active in the ABM/S field.
Peer-Olaf quickly replied and we did an initial Skype Call to get to know each other. We both then agreed that we would like to work with each other, especially my interest to go in a (computational) economics direction was very well received by him as he tries to get foot into this area as well but hasn't managed to do so far.
Now I had to think about what I would like to do in my PhD as so far I had only a few vague "Visions" and brainstormed a few vague Ideas. The overall direction was clear: I wanted to do something in the field of ABM/S and economics but all approached from the direction of computer science as this is my very field of expertise and I would get rejected by economists because of complete different wording and approach. Prior to my application I already had the idea to head into the direction of functional programming in the field of ABM/S as the dominant method is Object-Orientation (OO) so I thought it would be a nice idea to try something very new.
The idea I came up with was to apply functional programming in Erlang to ABM/S to simulate "something in economics" where something was "trading in virtual economies", "High-frequency trading", "Electronic Trading Platforms"... which could be boiled down to "Market Design". It was an intense and interesting process as I had to read quite a few papers and had to "clean up" and clarify my ideas and my mind. So after about 2 weeks of intense reading, reflecting and meditating I decided to go into the direction of "applying functional programming with special regard of ERLANG to ABM/S with a case study

in simulating the influence of different auction- and market types in electronic trading platforms."
I presented this to Peer-Olaf in a skype-call on the 4th of january. He told me that he will get a co-supervisor because he is no specialist in functional programming and more important, that I should apply for one of the 10 studentships granted by the school of computer-science. He made it clear that he will accept me as a student only if I will come to Nottingham and only if get this Studentship.
So the next goal was to apply for the studentship which was also a quite time-consuming process as you have to REALLY nail it - it's gotta be perfect otherwise one gets rejected really fast as you are competing with lots of other applicants. I had to provide a document containing the following:

1. Half a page describing reasons for wishing to pursue a PhD at Nottingham.
2. Half a page describing the proposed research area and topic (including a few references).
3. A demonstration of my technical writing skills: my Masterthesis.
4. Contact details of a supervisor at Nottingham who has already agreed to supervise my PhD.
5. Contact details of 2 Academic Referees - I did include Thomas Breuer and Simon Gächter
6. A CV including my degree classes.
7. A scan of my Bachelors and Masters degrees of all courses and the diploma supplement of both.

The studentship would start on 1st October 2016 and would grant me 14,507 pounds a year AND tuition fees (which are about 9,000 pounds a year). I had time until the 29th of February but Peer-Olaf told me not to wait until the deadline but to submit maybe one month earlier to be able to react if something is rejected. With a little help of my Prof. Hans-Vollbrecht, Thomas Breuer and BIG help of Peer-Olaf I did write and compile all the necessary information and applied for the Studentship on the 18th January. Peer-Olaf also did find a second supervisor - Thorsten Altenkirch from the Functional Programming Group who has his interst in Computer Aided Formal Verification and Type Theory, which makes a perfect match.

After about 2 months on the 16th March I got news from the school of computer-science that I have been allocated on of the school PhD studentships. I nearly freaked out so happy was I but deep within me I always had the feeling that things would work out and that I'd go to Nottingham - the one way or the other.

B.1 Studentship received - what now?

Getting the studentship is one thing but organizing everything is another big thing. What had to be done was:

1. Officially apply with Admissions Office so a conditional offer can be made. Conditional means that a few things need to be provided in order to receive an unconditional offer which finally guarantees me that I will study there. The things to provide were an IELTS test (see below) and the referee's reports (see below).

2. Apply for the VC's Scholarship for Research Excellence (EU). The school of computer-science will guarantee a fully funded place but wants me to apply for this scholarship. If I receive it (which I did) then they just have to top up to 14.507.

3. Provide an Academic IELTS certificate with overall score not less than 6.5 and in no part less than 6.0. I easily passed getting an overall Band Score of 8.5 / CEFR Level of C2 with having achieved Listening 9.0, Reading 8.5, Writing 7.5, Speaking 8.0.

4. Provide a referee's report of both my academic referees (which they were happy to do).

5. Apply for an accomodation in one of the many student hostels on the campus. I opted for Melton Hall, non self-catered, En Suite (including a bath in my room) for 5,890.5 Pounds a Year.

6. And most important: prepare myself and refine my topic so that I come well-prepared to Nottingham.

I could provide all the details and got an unconditional offer which made me a future student starting my PhD Programm on 1st October 2016. But now I had to really focus and dig into the topic

B.2 Preparing for my PhD

Since January I did a lot of preparation and learning for my PhD (which was done parallel to 24hrs/Week Job)

- Learned Erlang – implemented basic version of masterthesis CDA
- Learned Haskell (Graham Hutton Book & Tutorials on wiki.haskell.org on Monadic programming) – also implemented basic version of masterthesis CDA
- Gained basic understanding of what Agda is and what is is capable of (Wikipedia of agda, Thorsten's computer aided formal reasoning lecture, learnyouanagda, ...)

- Reading and studying book "Category theory for the sciences"
- Reading book Actor Model by Agha Gul
- Reading book PI-calculus by Robin Millner's
- Reading papers on formal ABM/S
- Reading papers on PI-calculus
- Reading papers on ABS in computational economics
- Reading papers on market designs

Initially I focused too much on the method but couldn't really argue why I was using it and what I want to achieve in the end. Writing a Research Proposal, giving a Presentation of my PhD Topic at FHV and discussing my ideas with others was of a great help to focus more on these missing points.

The insights during the preparation-process were the following:

- Erlang is a dead-end as it is not a pure functional language and reasoning is not really possible due to parallel nature and language constructs. Haskell is the way to go.
- The Actor Model will most probably not be the direction I will head to because research seemed to move into different direction than that was I need, my agents are most probably of different nature. Actor Model is more about the concurrent and parallel interaction between agents (actors) which is an interesting aspect but most probably not the focus of my research (and would open up a whole new pandoras-box).
- I want too much, I need to focus on less - this is a big thing to discuss with both my supervisors and I am sure they can help me with narrowing down my research topic.

C The evolution of my computer science interests

This is a retrospective of how my interests in computer science evolved into the current state.

I was and am no genius. I was also not extremely smart, sometimes I was slow in understanding, I was good enough but what set me apart was: if I understood something, I've understood fundamental and I could transfer it to other areas as well. Also if I really wanted to do something I pursued it so long until I could do it (gameengine). And the most important
There was always a bit of playfulness in my approaches: it was the mere curiosity how things would unfold, the fun in exploring unknown territory - when I had the feeling I understood enough I moved on to new territories. Thus the major shift of interests from starting programming with age of 15 until now with age of 33 from basic text games to GUI programming to 3D graphics programming with OpenGL 4.0 to web 2.0 development to enterprise back-end applications to operating systems to embedded systems to system simulation and now to the very basics and mysteries of computation itself.