University of Nottingham
UK | CHINA | MALAYSIA

Examination for the degree of PhD - First Submission

Independent Report on the thesis

Each Examiner is required to submit an independent Report on the thesis **in advance of the** *viva voce* examination and to submit it to Student Services or Head of School/Department

School/Dept:	Computer Science
Candidate's name:	**14276122** THALER Jonathan
Title of thesis:	Investigating the use of pure Functional Programming for Agent-Based Simulation
Programme of Study:	**PhD Computer Science**
Name of External Examiner:	Prof. Dr. Cezar Ionescu
Name of Internal Examiner:	Dr Julie Greensmith

Please note a copy of this report will be given to the candidate when the examination outcome is notified.

Internal Examiner's report: Dr. J. Greensmith

This thesis concerns with a purely functional approach to creating agent based simulation. This is done to overcome issues in software engineering regarding reproducibility of code behaviour. To show that this approach can be used in parallel/concurrently and Functional Programming for testing the implementation of agent based models.

Part 1.

The thesis is well structured with correctly presented front matter. The abstract is too long. The contributions in Chapter 1 are very clear and concise. Achieves a good balance of technical clarity and brevity. On P. 18 would help to have 1/2 sentences explaining software transactional memory in this context. There is a challenge here in working across so many sub-fields of computer science, it is not always clear who is the intended audience of this thesis. I would be very interested to hear the External's perspective on this. If in this section you don't want to introduce terms, at least give reference to where in the thesis that they are relevant.

The Background is presented in Chapter 2, starting with a clear definition of what is an agent, and making a clear distinction between agent based simulation versus multi-agent systems. Both the sugar scape and SIR models are well used as exemplars. Network types are often implemented with the SIR model but are not mentioned here, why are networks excluded? The description of sugar scape's processes are very detailed. Could this be explained algorithmic ally or even expressed in UML as a starting point for understanding. I know the thesis features this in some detail in Ch 5.2 but some general clarification here would be helpful. A nice section follows on purely functional programming. Well justified choice of using Haskell, and the supporting example in 2. 2.2 is very helpful. Good definition came through P32 to 38 (the external is more experience to verify the technicalities. A thorough description of FRP/Yampa or arrowized FRP is given though I'm still confused by the word and what this means in this context.

A thorough sub chapter on ABS follows though I feel that this could be more focused on providing examples from related work. Section 2. 4 feels a bit terse, like some of these examples should have featured earlier in the chapter. By the end of this chapter I feel like I've had a very good tutorial in ABS and FP but unclear the gap in research? I see from later chapters that a lot of literature is introduced on a chapter by chapter basis – justify this choice. This I am not so sure about as there is only I-5 pages of related work, this

University of Nottingham
UK | CHINA | MALAYSIA

Examination for the degree of PhD - First Submission

Independent Report on the thesis

Each Examiner is required to submit an independent Report on the thesis **in advance of the** *viva voce* examination and to submit it to Student Services or Head of School/Department

School/Dept:	Computer Science
Candidate's name:	**14276122** THALER Jonathan
Title of thesis:	Investigating the use of pure Functional Programming for Agent-Based Simulation
Programme of Study:	**PhD Computer Science**
Name of External Examiner:	Prof. Dr. Cezar Ionescu
Name of Internal Examiner:	Dr Julie Greensmith

Please note a copy of this report will be given to the candidate when the examination outcome is notified.

General remarks

The thesis makes a valuable contribution to the fields of agent-based simulations (ABS) and functional programming, by showing how to implement ABS in a pure functional way, carefully analysing the problems encountered along the way and recording their solutions. Both the advantages and disadvantages of a functional programming approach are soberly enumerated. The former outweigh the latter and the approach can be expected to lead to progress in the area of reproducible research. An important addition to the thesis is the examination of the contrast between object-oriented approaches and functional programming ones, leading to the identification of coalgebras as a promising foundation for agent-based systems.

The quality of the thesis is confirmed by four papers presented and published in acknowledged conferences and journals (two others are currently under review).

A number of awkward expression mar the thesis, and there seem to be some issues with the references, as outlined below. The thesis can be accepted after their correction.

Before listing these minor problems, a brief summary of the thesis is given.

Part 1 Introduction

Chapter 1 Motivation sets up the hypotheses: ABS are difficult to be implemented correctly because OOP is based on unrestricted side-effects. Using pure functional programming would be better, because functional programs are easier to reason about, parallelise, and test. The thesis represents the first attempt to systematically test these hypotheses. The chapter concludes with a perspicuous outline of the thesis.

Date: 25/02/2020

Signature: (External Examiner)

Please continue on the second page if necessary and sign each page

SA/25/02/2020

Chapter 2 Background points out that Multi-Agent Systems and ABS are not the same thing. Three kinds of ABS implementation are described. The case studies, the SIR model (explanatory) and the Sugarscape model (exploratory) are introduced. The author explains the choice of programming language and provides a summary of monads, arrows, arrowized FRP, monadic stream functions. When discussing the implementation of ABS, the author makes clear that the descriptions of ABS are heavily state-based, which underlines the difficulty of implementing an ABS in a pure functional programming language.

Chapter 3 Methodology is a brief outline of the way in which the scientific method has been applied to test the research hypotheses.

Part 2 Research

Chapter 4 Pure Functional Time-Driven ABS presents a pure implementation of SRI. There is a very good discussion of under- and super-sampling. The first attempt at implementing SRI using Yampa is subject to the random number correlation problem. A better solution using monadic stream functions is then introduced. Environments are added to the implementation (without StateT, which would have violated the semantics of the model). A closing discussion summarises advantages and disadvantages of this kind of implementation: we have many static guarantees of correctness (but more could be obtained with dependent types), but the performance is poor.

Chapter 5 Pure Functional Event-Driven ABS presents an event-driven version of SIR. A very useful technical discussion provides a nice example of replacing a stack of monad transformers by a type class. There is an interesting discussion of the performance of the implementation.

Chapter 6 Parallel ABS discusses advantages and disadvantages of parallelism and concurrency. There is a clear explanation of the difference between parallelism and concurrency. The author hypothesises that parallelism and concurrent implementations of functional ABS will benefit performance, but, especially in the case of concurrency, will damage reproducibility.

Chapter 7 Parallelism in ABS explores evaluation parallelism and data-flow parallelism. Measurements indicate that performance improves only for CPU bound pure computations, and that monadic versions perform worse. The discussion shows that the best avenues to improve performance are parallelisation within agents (rather than at the level of the entire simulation), or of the environment, or parallelising multiple simultaneous runs.

Chapter 8 Concurrency in ABS presents an implementation of concurrency using STM. Many empirical tests are discussed (many more could be done). The results are mostly as expected, with a few surprises which are explained well. The author points out that there might be limits to the use STM in ABS (some fairly common behaviour results in deadlocks, lack of fairness in activating threads, etc.).

Signature: (External Examiner)

SA/25/02/2020

Chapter 9 Property-Based Testing in ABS examines PBT in the context of ABS. It is important to note that PBT for ABS has specific features (in addition to the standard ones). As an aside, we note that in one of the examples, QuickCheck validates a result that does not appear to hold (`reverse (reverse xs) == xs`).

```
Prelude> head (1 : undefined)
1
Prelude> head (reverse (reverse (1 : undefined)))
*** Exception: Prelude.undefined
CallStack (from HasCallStack):
  error, called at libraries/base/GHC/Err.hs:79:14 in base:GHC.Err
  undefined, called at <interactive>:7:29 in interactive:Ghci7
```

The author shows how using “cover” and “checkCoverage” allows for testing statistical properties, which is essential in the context of ABS.

Chapter 10 Testing Agent Specifications discusses the checking of agent-level invariants. The author shows how to encode statistical properties, and explains some limits of using “cover” and “checkCoverage” for statistical properties (e.g., bimodal distributions).

Chapter 11 Testing Model Invariants introduces tests for the simulation-level invariants (invariants of the dynamics of the simulation). The author shows how to test these invariants under random event sampling, in order to ensure that the results are not dependent on the precise output from the agents. PBT has uncovered a surprising (and subtle) bug in the implementation of the time-driven ABS, thereby proving its usefulness. The discussion of testing leads to an interesting analysis of the event-driven and time-driven implementations.

Part 3 Discussion and Conclusions

Chapter 12 General Discussion summarizes the results: the hypotheses have been confirmed. Drawbacks of the implementation are presented: loss of performance, space leaks. The author gathers a number of problems for future research and addresses the question of whether Haskell would have helped Gintis. Finally, the author discusses coalgebras as a foundation for both OOP and FP implementations of ABS.

Chapter 13 General Conclusion shows the fertility of the work accomplished here by identifying several future research directions.

Signature: (External Examiner)

SA/25/02/2020

Minor corrections

- "the non-deterministic IO Monad" is not conventional terminology ("the non-deterministic monad" is just the list monad)
- page 183: "which he should have" is judgemental and should be removed

Some references are "off":

- page 102: [84] does not seem to be the best reference for illustrating the advantages of doing parallelism in Haskell
- [91] should probably be marked "2nd Ed"
- [92] doesn't do the job required by the sentence "The same problem was reported in [92], which tried to reproduce the work of Gintis [63]." The article to be cited is probably Botta et al. 2013
- page 44 [179] seems to be given as a reference for SIR
- it is not necessary to repeat the same reference every time: e.g., Sugarscape [50]
- I haven't been able to find in [10] the report of difficulties in reproducing [8]. I only checked because the two papers have the same author. A page number would be helpful here.

Some stilted formulations:

- page 15: "considerably easy" –> "easier"
- page 41: "program with points" should probably be "program with explicit arguments"
- page 42: "what BearRiver puts on top" I'm afraid I didn't understand that
- page 43: "proactivity of both, agents and its environment" no comma
- page 48: "the notorious sampling." perhaps "…problem"
- page 50: "SugarScape" while everywhere else it's "Sugarscape"
- page 55: "pose solutions" –> "propose solutions"
- page 64: improper use of possesive 's' "BearRivers"
- page 71: "quantitatively equal"?
- page 78: "boiler plate" should be one word
- page 96: "purely functional"
- page 101: "spacial" is acceptable, but "spatial" would be better
- page 101: "a conservative approach and an optimistic approach exists" –> "exist"
- page 102: "PDEVS" what does "V" stand for?
- page 102: "the baseline" maybe "lesson" or "message"?
- page 102: "notorious" is overused
- title of Chapter 6 might have been "Parallel and Concurrent ABS"
- page 105: "referential transparent" –> "referentially transparent" (also on page 172)
- page 121: "by running each agent in a thread will guarantee" something seems to be missing
- page 127: "of very much importance" –> "of very high importance"
- page 140: "the authors [136]" –> "the authors of [136]"
- page 140: "proofing" –> "proving"
- page 161: "giving an excellent use case" –> "providing a use case that thoroughly exercises…"
- page 172: "if one programs careful" –> "carefully"
- page 173: "this seemed" why the past tense?
- page 173: "could be done purely functional" –> "could be implemented in a pure functional way" (this shows up in other places as well)
- page 180: "with much power comes much responsibility" –> "with great power comes great responsibility"

Signature: (External Examiner)

SA/25/02/2020

could be broadened. Also this is missing a summary, negatively impacting the narrative structure, and this should be carried forward to the start of Chapter 3 e. g. "In Ch 2 we have clearly identified the need for...". Ch3 is a bit odd, a collection of vague statements which does not quite belong anywhere in particular.

Part 2-

Chapter 4 at gives the implementation of the ABS of the SIR model using a FRP approach. Explicit FRP specifications are given and are well examined with features justified. In 4. I. 1. its had to see where the actual results are. Reduction in performance is mentored but where is the evidence of this? The issue surrounding randomness on p 63 needs more explanation in the Viva voce examination. Why is a Moore neighbourhood chosen in this example? How have you proven that there is a lower number of recovered agents? What are the averages and Standard Deviations here (you do use the Rand monad at this point right?). Why are the time semantics the most important advantage at this stage as stated on p 67? Good balanced discussion here. Aa ha, in 4.4.3 you start talking about runs in the results. Chapter goes on to describe a step further with Pure Functional Event-Driven ABS, also using the SR model. Thorough descriptions are given, clear differences are shown between the implementation here and in the previous chapter. Describe the link between tag less final, interfaces and different interpretations.

Its clear that a similar methodology has been applied in 5.2 with the ED sugar scape model, good use of shared mutable proactive environment. In general the found specification is appropriately detailed. Ask to explain circular event sending in context of send sync. Some of what is in 5.3 does it really belong in the discussion? Specification of the exact platform is needed to make claims about performance. 5-3.7 needs to act as a stronger primer over what is to come. Why do we travel between parallelism, to concurrency, to formal specifications to properly based testing? What is the story? What is the strategy? Why these subjects? Chapter 6: surely some of this should have been discussed in related work. Should 6 and 7 be merged? 6 seems to be the background for 7 and feels like another orphan chapter. Ch 7 is very interesting. P 106, "sparks" has a circular definition.

The case studies are interesting and well linked back to previous chapters. There is a surprise result on p110, why is this unexpected? As worse performance was reported, what does this imply? Can you be certain this is not a result of your specific implementation? Sensitive discussion of negative results provided on p 114. Good narrative at the end of this chapter and smooth intro into Ch 8. Its an interesting idea to use STM to remedy the problem: good description on p 116-119. I would like to talk through fine details of this. Chapter 8. 1 on p 121. Good choice to only apply this to SIR, clear choice of case studies. On p 129 why does this not work on Sugar scape? The SS case study comes afterwards… Fig 8.4 why is sequential a straight line? End of Chapter 8 should motivate Chapter 9, and state if the research hypothesis was satisfied or not.

In Chapters 9,10 + 11 the focus of the thesis changes again, to look at the correctness of the ABS, which is a vital area of study in simulation research. Again, an interesting choice to include a literature review in chapter 9, which like Chapter 6 does not have any experiments or results. This unconventional structure should be justified in the Viva. It is clear that if you are using FP then it follows that PBT is appropriate, though in this already long thesis, is 9.3. 1 necessary? This chapter really needs a conclusion. In Chapter 10 detailed specifications are given for the agents in the SIR model. Would like the external's feedback on the validity. Is there an edge case i. e. could you make it fail? How would you apply these tests to the environmental parameters?

Chapter 11 tests the dynamics of the SIR model, as a good exemplar is the application of Quick Check. Appropriate statistical tests are chosen in 11. 2, unclear where the results are. Also 11. 3.1I, system dyamics is suddenly introduced, what is going on here?

Part 3.

The thesis concludes a Part III. This research has presented a strong case for the use of functional programming, property based testing and concurrency in agent based simulations as techniques for a principled and verifiable simulation development methodology. In the bulleted list I would expect to see where in the thesis this evidence is presented to verify the contribution to knowledge. Section 12.3.13 very

interesting. These generalisations are more appropriate to Ch13. Could 12-1 13 be merged to give stronger narrative? The thesis concludes well, references are in good order, as are the appendices.

Note: Written supplemental notes have been provided in a hard copy of the thesis.

Date: 24/2/2020

Signature:  (Internal/External Examiner*) (*delete as applicable)

Please continue on the second page if necessary and sign each page

SA/31/03/2020

Examination for the Degree of PhD - First Submission

Joint Report Form

School/Dept:	Computer Science
Candidate's name: 14276122 THALER Jonathan	
Title of thesis: : Investigating the use of pure Functional Programming for Agent-Based Simulation	
Programme of Study: PhD Computer Science	
Name of External Examiner: Prof. Dr. Cezar Ionescu	
Name of Internal Examiner: Dr Julie Greensmith	

A brief report on the thesis examination should be included below.

If the examiners are recommending re-submission for PhD - but have concerns that it may not be possible for the candidate to achieve that standard in the re-submission period, they may inform the candidate that it is advisable to re-submit for MPhil, and if this is done this should be clearly stated here.

If the examiners require the candidate to incorporate revisions contained in their Independent Reports when completing corrections or submitting a thesis for re-examination this should be explicitly stated.

Report on the thesis examination:

The thesis is of the expected standard and generally well written. Here we specify a list of corrections resulting from discussions in the viva voce examination. NB. An addition list of typographic corrections was provided by the external examiner, attached in his independent report, and these typos should be corrected as part of the minor corrections required for the thesis resubmission.

- The abstract is excessively long and this should be shortened to be more concise (ideally one page maximum), and to stress the findings of the thesis.
- Individual contributions of the candidate to each of the listed papers should be added, to make it clear how these publications relate to what is written in the thesis.
- The JAAMAS paper has been revised and resubmitted and this information should be updated.

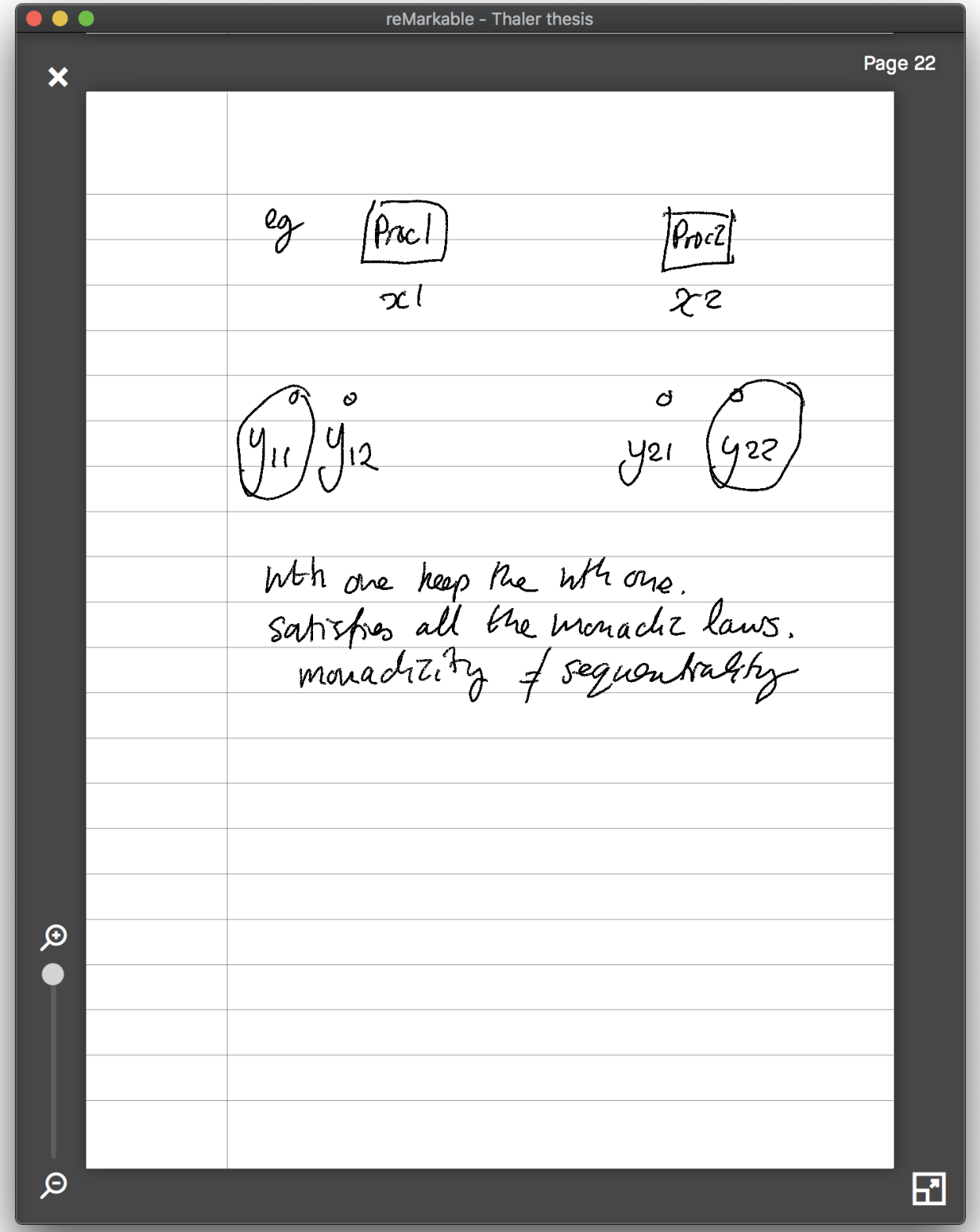
- p 15 add objectivity to random PBT statement by providing contrast with unit testing
- p 16 Why is FP 'notoriously slow'? Please clarify why this is the case notably show to clarify
- P16 Discussion on Internet of Things is detracting in motivation and should either be explained or removed
- P17 "In conclusion, this thesis claims that the ABS community needs functional programming because of its scientific computing nature, where results need to be reproducible and correct, while simulations can massively scale up as well. As pointed out above, the established object-oriented approach needs a considerably level of effort and might even fail to deliver these objectives due to its conceptually different approach to computing with unrestricted side effects. " – this needs to be disambiguated as it is unclear what you are trying to say here.
- P18 add Test Driven Development as part of a contribution
- In the introduction identify for which research community the chapters are aimed at
- P23 missing reference to Anylogic and Repast

- P24 ABS SIR and networked SIR, the differences in the two models need to be mentioned
- P24 The differences between Agent based simulation and ABMS needs to be clarified.
- p 27 add diagram to explained
- Provide better definitions of FRP, Yampa and arrowised systems.

- P63 discussion of the correlation and a random number generator around g.; susceptible g -> infected g,: is occasionally g being updated or not? Clarify.
- Are you creating the same deterministic random number stream? Clarify.
- Are the dynamic correlated versus not correlated? Clarify.
- Do you need to use monadic stream functions? Can this threading be done manually with Yampa? → Would you have to change the implementation of 'occasionally'. Clarify.
- Add an illustration to show the difference IF it makes a difference to the results.
- Does this dependency between agents on a determinate random number generator exist depending on the implementation of the random number generator? I.e. true vs pseudorandom number generators as pseudo numbers are not independent ⇒ would be worth checking and clarifying.
- To do these tests : Yampa with correlation, MSF without generator; will you see a significant difference? Clarify as this DOES have implcations for the effectiveness due to correlations. This should be added to the thesis IF you see a difference. If so this needs to be tied in to other chapters.
- Pg 99 Why performance difference between Java and FP versions? Clarify. Is this due to differences in:
- Array updates;
- Nested loops in Haskell to update the sleeping agents
- Fixing this with mutable arrays.
- Explain the reasons here as opposed to deferring any discussion until Chapters 7&8.
- THESIS WIDE CORRECTION: "purely functional" OR "pure functional" make this consistently correct REMOVE THIS INCONSISTENCY.
- P100 bridging sentence required
- P100 be dismantle big reference block and only use the references that you need.
- Generally, you don't need to repeat references multiple times unless there is a specific reason e. g. Sugarscape [50] [Global check]
- Some comments about references:
- some minor e.g. [91] is Graham's book including 2nd Edition.
- Some references which are out of context. e. g. [92] is not the point of this paper
- -> see EE report for these corrections for more appropriate suggestions for replacements including correct citing of Dr Botta.
- P102 citing [84] Marlow book for example would be more appropriate

- P44 on SIR model, you appear to have cited yourself as opposed to the original author
- Go through references again and check for appropriateness of references, sometimes they are a bit strange and not the standard references. Ask for advice on this from IE if needed.
- [10] and [8] has to be checked and if it is in there you have to add a page numbers
- P107 Discussion of ParT: this explanation needs strengthening by thinking about monads differently, {Bind vs unit and join}, 7.2.1 rewrite for clarity based on the corrected information in the next two screenshot panels of notes: See notes below.

```
data D a = D (P -> a)
             :f
             b

fmap :: (a -> b) -> Da -> Db

fmap f (Dd) = D (f ∘ d)        is a functor
                               also a monad

return :: a -> Da

return a = D (const a)

[[for each a get a distribution of bs.]]

(Dda) ▷ f = Ddb   where f :: a -> (P -> b)
 :: P -> a
where db :: P -> b

    db P = f (da P) P .
```



- P116 clarify the definition of reproducibility in terms of processes or results
- P121 Diagram is not described, fix to be concurrent or label as appropriate
- P148 Chapter needs a summary
- P144 prop_reverse_reverse you have written as ‘pass’; it fails! Fix please.
- E.g. head(1: undefined)=1
- Head(reverse(reverse(1:undefined)) -> undefined.

- P165 Justify your 90% and missing parentheses
- P176 12.1.3 refer to precisely in Chapter 8, p139.
- P143 Do not be so bold regarding your statements on PBT and similarly don't belittle the work described on p183.

Date of *viva voce* examination: 26/02/2020

Report on the *viva voce* examination:

We commenced the viva by asking the candidate how he arrived at the decision to study this particular research area. Jonathan specified that he was originally interested in games with specialisation in Embedded Systems, simulation and optimisation. He had been studying and using object oriented programming for a long time, but became inspired by the work of Eric Meyer on functional programming. He gave a good rationale as to how he wanted to combine elements of object oriented and functional programming, to research the intersection between the two. We asked Jonathan what kind of thesis he feels he has produced and stated that this is a software engineering thesis and the research evolved beyond what he originally anticipated.

We then examined how Jonathan had accrued knowledge regarding the subject area in particular how an object-oriented programmer learned Haskell. He described how he used the knowledge of this subject in the School of Computer Science to assist with this research. He was co-supervised across two research groups in the School, Intelligent Modelling and Analysis and Functional programming. He engaged actively with the Functional Programming group, using books, online resources, working through suggested examples and exercises and experimented with this new paradigm. He explained the process of how he accrued background information and he was able to outline his initial research strategy and how engagement with the Functional Programming seminar series was pivotal to this.

We asked Jonathan to describe for us an overview of the contributions in the thesis. He described three main contributions and significant findings:

1) The ability to combine functional programming with object oriented agents
2) Examination of concurrency in functional agent systems
3) The application of property based testing to agent based simulation

The descriptions and definitions given as an answer to this question were coherent, concise and well considered. This was followed up by a brief discussion on how Jonathan had been able to learn and apply quickcheck as a tool for property based testing.

Following the introductory questioning and discussion surrounding the motivation of the thesis, the examiners asked questions pertaining to specifics in the thesis itself. Questions regarding content were answered in a well-informed manner and constructive comments and corrections were agreed upon. Specifics of corrections to the thesis are given in the previous section of this report.

Wider discussions in the viva included to whom the research was aimed, and which communities may find it a useful contribution. Justifications for these answers were provided. The thesis was well defended and the candidate agreed to the requested minor corrections to the thesis document.

It is clear that an original contribution has been made by the candidate and that the thesis is clearly Jonathan's own work. Our recommendation as a result of the viva is 3 months minor corrections.

Candidate's Name: **14276122** THALER Jonathan

Summary of Examiners' Recommendations

Please tick the appropriate box (s)

Award of PhD

☐ without conditions
☐ subject to correction of typographical errors within 1 month[1]
X subject to minor amendments[2] to be completed within 3 months[3]

Referral, with

EITHER

a) resubmission of the thesis within 12 months for PhD:
☐ *viva voce* required
☐ *viva voce* not required

Please Complete this section if 'a)' is selected:
If resubmission for PhD is recommended but the student is unable to re-submit:
☐ MPhil to be awarded now
☐ MPhil to be awarded now subject to minor amendments[2] to be completed within 3 months[3]
☐ Thesis not currently at MPhil level

OR

b) another *viva voce* examination although thesis is accepted
☐ without conditions
☐ subject to minor amendments[2] to be completed within 3 months[3]

Failure at PhD standard

☐ No higher degree to be awarded at all[4]

[1] This option should be selected only in instances when the candidate is required to make minor corrections to the text, e.g. typographical errors, which are so minor as to be completed in one month. It is the responsibility of the Internal Examiner to verify that the corrections have been made to his/her satisfaction.

[2] Minor amendments are amendments to the thesis not requiring external academic re-assessment, e.g. extensive typographical errors, minor re-organisation of material, addition of supplementary material to clarify the content of the thesis, or removal of extraneous text and may include minor re-writing of material. Minor amendments are in excess of minor corrections but are not, in the opinion of the examiners, sufficient to require the student to be referred and to resubmit. It is the responsibility of the Internal Examiner to verify that amendments have been made to his/her satisfaction.

[3] Exceptionally, where a student has been previously registered as a part-time student and it has been demonstrated that circumstances exist such that it would be in the best interests of that student, the examiners may recommend that the degree be awarded subject to minor amendments being completed within six months. Where this option is chosen, please indicate by deleting '3' and replacing with '6'.

[4] The University recommends that, other than in exceptional circumstances, Examiners should not normally choose this option on a student's first examination.

It is confirmed that the *viva voce* examination has been conducted and that the student has been given informal feedback on the outcome of the examination.

Name of External Examiner: Prof. Dr. Cezar Ionescu

Date 30/03/2020. . . . .Signature

Name of Internal Examiner: Dr Julie Greensmith

Date 30/03/ 2020 Signature

I confirm that the examination process has been completed according to University regulations and procedures[1]. I endorse the Examiners' recommendations.

Signature of Head of School/Dept: Date . **31-3-2020**. . . . . .Signature **p.p.** .....

[1] For joint/dual awards it is confirmed that the Partner Institution(s) regulations have been satisfied.