The Australian National University
2600 ACT | Canberra | Australia

School of Computing

College of Engineering, Computing and Cybernetics (CECC)

Enriching a Verified Choreographic Language with a Simply Typed Lambda Calculus

— Honours project (S1/S2 2024)

A thesis submitted for the degree
Bachelor of Advanced Computing (Research and Development)

By:
Xin Lu

Supervisor:
Dr. Michael Norrish

September 2024

Declaration:

I declare that this work:

- upholds the principles of academic integrity, as defined in the University Academic Misconduct Rules;
- is original, except where collaboration (for example group work) has been authorised in writing by the course convener in the class summary and/or Wattle site;
- is produced for the purposes of this assessment task and has not been submitted for assessment in any other context, except where authorised in writing by the course convener;
- gives appropriate acknowledgement of the ideas, scholarship and intellectual property of others insofar as these have been used;
- in no part involves copying, cheating, collusion, fabrication, plagiarism or recycling.

September, Xin Lu

Acknowledgements

If you wish to do so, you can include some Acknowledgements here. If you don't want to, just comment out the line where this file is included.

There is absolutely no need to write an Acknowledgement section, so only do so when you *want* to – it's always important to stay sincere. One reason for including an acknowledgement could be to thank your supervisor for extraordinary supervision (or any other reason you deem noteworthy). Some supervisors sacrifice a lot, e.g., are always available, meet on weekends, provide multiple rounds of corrections for theses reports, or the like (keep in mind that writing a thesis is special for you, but not for them, so they do actually not have any reason to sacrifice their private time for this!). Seeing acknowledgements in this report can feel like a nice appreciation of this voluntary effort. For large works that form the end of some studies (like an Honours or Master thesis), it is also not uncommon to read acknowledgements to one's parents or partner. But again, completely optional!

Abstract

- choreography diagram; CC, Kalas - the meaning of deadlock freedom by design - mostly focus on interactions via message passing - contribution 1: richerLang: call by value, functional big step semantics with clock to be implemented in HOL4, richer data types, environment semantics; an environmental language model with type theory; and strong normalisation property - contribution 2: the enriched choreography, with a simple type theory, Kalas have the safety property of deadlock freedom

Table of Contents

Chapter 1

Introduction

Distributed systems consist of multiple endpoints that communicate by exchanging messages, operating with asynchrony and parallelism as these messages are sent and received between the various endpoints. But programming distributed system is notoriously error-prone as programmer has to implement the communication protocol by developing individual endpoint programs. Mismatched message sending and receiving can lead to errors such as *deadlock*, where the system is waiting forever for a message.

Choreography arises as a programming diagram to address this issue by providing a concrete global description of how the messages are exchanged between endpoints in a distributed system. A choreography program is written in a similar style to the "Alice and Bob" notation by Needham and Schroeder:

$$\begin{array}{ll} \texttt{1.} & \texttt{Alice} \to \texttt{Bob} : key \\ \texttt{2.} & \texttt{Bob} \to \texttt{Alice} : message \end{array}$$

Thus message mismatches are disallowed from the choreographic perspective. A property we refer to as *deadlock free by design*. The global choreography is then projected into process models for each endpoint via EndPoint Projection (EPP), with properties such as deadlock free by design preserved (Hallal et al.).

While most choreography languages focus on the message exchange behaviours, few pay attention to the local computation happening in the individual endpoint. It is shown by Hirsch and Garg that as long as the local language exhibits type preservation and progress, the choreography is deadlock free by design. Thus when studying message exchange behaviours for choreography such as multiparty sessions, asynchrony, and parallelism one can safely assume good behaviours of local computation (Montesi and Yoshida; Cruz-Filipe and Montesi, 2017; Carbone and Montesi). But when it comes to

Plan for the introduction: - what is a choreography, why is it desired (what brings to the idea of choreography): what is deadlock-free; also deadlock freedom by design

- CC: minimal design, not implementation-specific, foundamental model for studying choreography, elegant - What is Kalas and what it does well; what it can be impoved and why interesting - how different choreographies handle local computations and why a simplification is desired - research question

- Contribution - contreibution: simple typing system for choreography???

- Thesis Structure

Chapter 2

Background

2.1 Choreography as a Programming Diagram

- choreography semantics and basic results ?

2.2 Interactive Theorem Proving

- this work is done in HOL4 so maybe start from this and think how to introduce stuff
- Kalas is kind of both a choreography language and implemented in HOL

Chapter 3

Related Work

- (deadlock freedom/ type soundness) typing of choreography - CC, Kalas, CC M do not have typing system, deadlock freedom for closed choreography idirectly follows the semantics, where the local computation is always assumed terminated - When deadlock freedom does not directly follows from the choreography semantics, such as the compositional choreography proposed in (Yoshida), typing system is used to ensure progress - others, Channel Choreographies (ChC) in (Fabrizio Montesi thesis) uses typing system for checking protocal Correctness

- local procedure in choreography

- functional big-step semantics

- SN for STLC; strong normalisation for languages of environmental semantics

Chapter 4

The Environment Model

4.1 Syntax

- binary operators - unary operators (StrOf, NumOf) - value script?

4.2 Semantics

- functional big step semantics: intepreter style with clock, total function. for being implemented in HOL4 - properties: clock increment (cases on result) (and clock decrement) - closure: issues with dynamic environment, so we use lexical environment; do we explain restrciting to fv(e) and how?

4.3 Typing

4.3.1 Syntax

- rich data types: int, string, boolean, sum type, pair type, closure, ... - typecheck - uoptype, boptype - valuetype - envtype

4.3.2 Typing Rules

- typecheck - uoptype, boptype - valuetype: closure - no type uniqueness

4.3.3 Main Properties

- value invertability; envtype used for fnT case of it - operators: uoptype soundness and boptype soundness (use the invertability) - typecheck: reducing typing environment (for soundness fn case)

4.3.4 Type Soundness

Chapter 5

Strong Normalisation

Chapter 6

The Enriched Choreography

Chapter 7

Concluding Remarks

If you wish, you may also name that section *"Conclusion and Future Work"*, though it might not be a perfect choice to have a section named "A & B" if it has subsections "A" and "B". Also note that you don't necessarily have to use these subsections; that also depends on how much content you have in each. (E.g., having a section header might be odd if it contains just three lines.)

7.1 Conclusion

This section usually summarizes the entire paper including the conclusions drawn, e.g., did the developed techniques work? Maybe add why or why not. Also don't hold back on limitations of your work; it shows that you understood what you have done. And science isn't about claiming how great something is, but about objectively testing hypotheses. Also note that every single scientific paper has such a section, so you can check out many examples, preferably at top-tier venues, e.g., by your supervisor(s).

7.2 Future Work

- asynchronus messages; confluence property - Progress for EPP

Chapter 8

Test

We define what it is for a choreograph to be well-formed with the $G\,,Th{\vdash}c\Box$ relation.

This is a theorem:

$$\vdash \emptyset, \Theta \vdash c\Box \ \Rightarrow\ \exists\, \tau\ l\ s'\ c'.\ \emptyset \rhd c \ \xrightarrow[l]{\tau}\ s' \rhd c' \ \lor\ \neg\texttt{not\_finish}\ c$$

The transition relation looks like `eval_exp` *clk E exp*

new tests tex

Appendix A

Appendix: Explanation on Appendices

You may use appendices to provide additional information that is in principle relevant to your work, though you don't want *every reader* to look at the entire material, but only those interested.

There are many cases where an appendix may make sense. For example:

- You developed various variants of some algorithm, but you only describe one of them in the main body, since the different variants are not that different.
- You may have conducted an extensive empirical analysis, yet you don't want to provide *all* results. So you focus on the most relevant results in the main body of your work to get the message across. Yet you present the remaining and complete results here for the more interested reader.
- You developed a model of some sort. In your work, you explained an excerpt of the model. You also used mathematical syntax for this. Here, you can (if you wish) provide the actual model as you provided it in probably some textfile. Note that you don't have to do this, as artifacts can be submitted separately. Consult your supervisor in such a case.
- You could also provide a list of figures and/or list of tables in here (via the commands `\listoffigures` and `\listoftables`, respectively). Do this only if you think that this is beneficial for your work. If you want to include it, you can of course also provide it right after the table of contents. You might want to make this dependent on how many people you think are interested in this.

Appendix B

Appendix: Explanation on Page Borders

What you find here is an explanation of why the border width keeps flipping from left to right – which you might have spotted and wondered why that's the case.

Firstly, that is *intended* and thus correct, so there is no reason to worry about this. The reason is that this document is configured as a two-sided book, which means:

- We assume the document will be printed out,
- that this will be done in a two-sided mode (i.e., the document will be printed on both sides of each page), and
- that the bookbinding will be in the middle, just like in every book.

When you open the book, there are three borders of equal size n. This however requires that even pages have a border of n on their left and $\frac{n}{2}$ on their right, and odd pages have a border of $\frac{n}{2}$ on their left and n on their right. This is illustrated in Figure B.1.

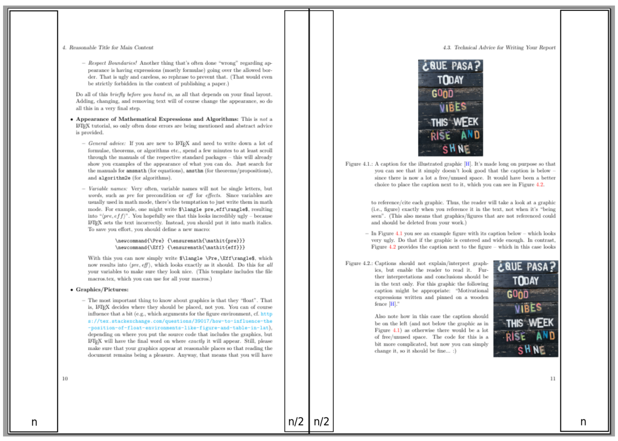

Figure B.1: Illustration showing why page borders flip.

Bibliography

CARBONE, M. AND MONTESI, F. Deadlock-freedom-by-design: multiparty asynchronous global programming. 48, 1 , 263–274. doi:10.1145/2480359.2429101. https://dl.acm.org/doi/10.1145/2480359.2429101. [Cited on page 1.]

CRUZ-FILIPE, L. AND MONTESI, F., 2017. A core model for choreographic programming. In *Formal Aspects of Component Software: 13th International Conference, FACS 2016, Besançon, France, October 19-21, 2016, Revised Selected Papers 13*, 17–35. Springer. [Cited on page 1.]

HALLAL, R.; JABER, M.; AND ABDALLAH, R. From global choreography to efficient distributed implementation. In *2018 International Conference on High Performance Computing & Simulation (HPCS)*, 756–763. doi:10.1109/HPCS.2018.00122. https://ieeexplore.ieee.org/abstract/document/8514427?casa_token=B9uMnW0mxFEAAAAA:DMmhwgQZJnHAX6o6p-EHBs4K9rct4pEKen9fdt2CXHC6N0WZxpHT5FSZZFAchGBDjiqmPeviH1U. [Cited on page 1.]

HIRSCH, A. K. AND GARG, D. Pirouette: higher-order typed functional choreographies. 6, 23:1–23:27. doi:10.1145/3498684. https://dl.acm.org/doi/10.1145/3498684. [Cited on page 1.]

MONTESI, F. AND YOSHIDA, N. Compositional choreographies. In *CONCUR 2013 – Concurrency Theory* (Berlin, Heidelberg, 425–439. Springer. doi:10.1007/978-3-642-40184-8_30. [Cited on page 1.]

NEEDHAM, R. M. AND SCHROEDER, M. D. Using encryption for authentication in large networks of computers. 21, 12 , 993–999. doi:10.1145/359657.359659. https://dl.acm.org/doi/10.1145/359657.359659. [Cited on page 1.]