My thesis and other animals

Jonathan Grattage
Outline

- Thesis Outline
- Contributions
- Future Directions
- Other Projects
- Summary
QML: A functional quantum programming language

Thesis Structure:

1. Introduction – history, motivation, and background

2. Reversible classical computing  
3. Reversible quantum computing  

4. **FCC**: Reversible and irreversible classical computation

5. **FQC**: Reversible and irreversible quantum computation 

*derived from reversible computation, inspired by physical models*

*first formalisation of the Quantum Circuit Model, using Category Theory*
6. **QML**: A functional quantum programming language
   *introduces syntax and typing rules + example programs*

7. Operational semantics of **QML**
   *with a denotational semantics using Superoperators
    all implemented in Haskell*

8. Further research
   *gives outline of completeness proof
    implementing infinite data structures and recursion
    dependent types, etc…*

9. Summary and conclusions
   *I passed!*

Appendix: Shor’s algorithm and the QFT
Contributions

**QML**

- functional language for quantum computations on finite types.
- quantum data *and* quantum control
- integrates *reversible* and *irreversible* quantum computation
- based on strict linear logic
  - *controlling measurement (weakening) rather than copying*
- Design guided by categorical semantics
  - *Programs are morphisms in FQC, giving quantum circuits*
  - *Gives first formal description of standard Quantum Circuit Model*
- Both operational and denotational semantics
  - *implemented in Haskell*
Future Directions

- Complete proof of *compositionality*
- Infinite data structures and recursion
- Full equational theory
- Dependent types and views for basis independence
- More research into **FxC** structure and *laws*
- Higher order programming (via Day’s construction?)
- Reintroduction of coproducts, and classical data …
Other recent projects
A thesis and beyond in 15 minutes (An overview)

Thanks for listening

Any questions?