Introduction to Quantum Programming and Semantics: summary

Chris Heunen
Quantum Programming

“Not yet one platform to rule them all”

- OpenQASM: open standard, very basic type system, have to define basic gates

- Qiskit: circuit description, simulation, hardware backend

- Quipper: Haskell, higher-order, data types, recursion, circuit-generating

- Q#: QRAM model, measurement for control flow, dot.net/Azure

- Silq: uncomputation, linear type system, qfree

- PyZX: Python, rewriting, optimisation
“Morphisms are more important than objects”

- **Set, Rel, FHilb**
- Universal properties: products
Monoidal categories

“Can compose morphisms in sequence and in parallel”

- Coherence theorem
- Graphical calculus: isotopy
- Braiding, symmetry
Scalars

“Monoidal categories replicate linear algebra features”

- Scalars commute
- Scalar multiplication
- Dagger categories
- Way of the dagger
“Dual objects model maximally entangled states”

- Definition: cup, cap, snake equation
- Names, conames
- Transpose morphisms
- Traces and dimension
- Teleportation, one-time pad encryption
Monoids and comonoids

“Comonoids model copying”

- Monoid: unit and multiplication
- Monoids embed into pair of pants
- No uniform cloning or deleting
- When tensor products are products
Frobenius structures

“Classical structures model classical data”

- Frobenius law: dagger and closure
- Spider theorem
- Phases
- In $\text{FHilb}$: matrix algebras and orthogonal bases
- in $\text{Rel}$: groupoids
Complementarity

“Complementary Frobenius structures are maximally incompatible”

- Mutually unbiased bases
- ZX calculus
- Oracles
- Deutsch-Jozsa algorithm
ZX calculus

“Axiomatise qubit computation graphically”
- Sound and complete
- Approximately universal
- Rewriting system may be automated
- Quantum circuit simplification
Exam

► Answer two out of three questions
► If can’t solve subquestion, assume answer and move on

Good luck!