Quantum Software

Aleks Kissinger, Stefano Gogioso

Oxford, Hillary Term 2024
Quantum software

1. := the code that runs on a quantum computer

   *factoring*

   *search*

   physical simulation
   optimisation problems
   linear systems & codes
   network flows
   natural language processing
   ...

2. := the code that makes that code (better)
   
   *compilers*
   *optimisation*
   *verification*
**Problem:** no quantum computers

(† Oxford, Vienna, Delft, Sussex, Maryland...)

**Rigetti Aspen**

**IBM Q Rochester**

**Google Sycamore**
Problem: no quantum computers

**Problem:** limited quantum computers

NISQ := “noisy intermediate-scale quantum”

NISQ devices have:

- short coherence times
- low numbers of qubits
- noisy operations
- limited connectivity
- ...

⇒ **small** advances in software give **big** gains on NISQ hardware!
Quantum circuits

• := the ‘assembly language’ of quantum computation, e.g.

INIT 5
CNOT 1 0
H 2
Z 3
H 0
H 1
CNOT 4 2

...
* Selinger 2015
* Amy, Chen, & Ross 2018
* Nam et al. 2018
ZX-diagram
ZX-diagrams

...are like circuits, but made from **spiders** instead of gates:

\[ Z_\alpha = \quad \alpha \quad \]

\[ X_\alpha = \quad \alpha \quad \]
ZX-diagrams are bendy
ZX-diagrams are bendy
ZX-diagrams are bendy
Why spiders?

- They generate all linear maps $\mathbb{C}^{2^m} \rightarrow \mathbb{C}^{2^n}$.
- Handy for building most common gates, e.g.

\[ S = \frac{\pi}{2} \quad T = \frac{\pi}{4} \]

\[ H = \frac{\pi}{2} \quad = \frac{\pi}{2} \cdot \frac{\pi}{2} \cdot \frac{\pi}{2} \]

- ....and....
ZX calculus

these 8 rules \implies everything before
Q: How do we scale up?

A: Automation.
In this course...

• We’ll look at **fundamental structures** underlying quantum computations:

\[ \text{ZX-diagrams} \quad \leftrightarrow \quad \{ \text{graph states} \quad \text{stabiliser groups} \quad \text{exponentiated Paulis} \quad \text{phase polynomials} \} \quad \ldots \]

• And apply them to...
Classical simulation

\[ \psi \ldots = C \ldots \]

\[ \text{Prob}(i \mid |\psi\rangle) = ??? \]
Circuit optimisation
Circuit routing

\[ \begin{align*}
0 & \rightarrow 1 \\
5 & \rightarrow 4 \\
3 & \rightarrow 2 \\
14 & \rightarrow 13 \\
12 & \rightarrow 11 \\
6 & \rightarrow 7 \\
16 & \rightarrow 17 \\
10 & \rightarrow 9 \\
\end{align*} \]
(NISQ) quantum algorithms

Dynamical simulations

Quantum chemistry
Condensed matter
Finding ground states

Variational quantum algorithms

Error correction

Compilation

Classifiers
Generative models

Machine learning

Mathematical applications

Systems of equations
Factoring
Principal components

Combinatorial optimization

New frontiers

Quantum information
Quantum metrology
Quantum foundations

3 https://www.nature.com/articles/s42254-021-00348-9
Aleks Kissinger Quantum Software HT 2022
Implementation


Aleks Kissinger
Quantum Software
HT 2022
Format of the course

• 8 weeks, 24 lectures, 6 problem sheets + classes (week 3, 4, 5, 6, 7, 8)
• 5 weeks theory (Aleks)
  • basics, ZX, classical simulation, quantum compilation, quantum error correction
• 3 weeks live coding (Stefano)
  • algorithms, implementation on IBMQ
• Materials:
  • handwritten lecture notes, matching Aleks’ lectures
  • textbook: preprint of Picturing Quantum Software. Kissinger & van de Wetering
  • Jupyter notebooks from live coding
  • Picturing Quantum Processes. Coecke & Kissinger. CUP (optional)
  • Quantum Computation and Quantum Information. Nielsen & Chuang. CUP (optional)
• Exam is by take-home miniproject
  • expect a theory and a (Python) coding component