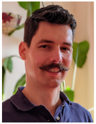
Introduction to Quantum Programming and Semantics

Week 1: Course organisation

Chris Heunen

THE UNIVERSITY *of* EDINBURGH
informatics

Practicalities

- Course team:  Chris Heunen

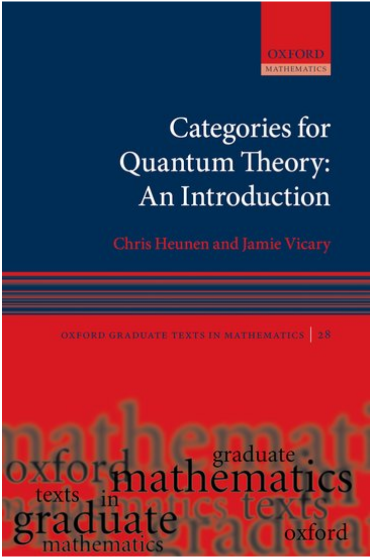
   Kengo Hirata

- Website: Learn, Piazza
- Lectures: Tuesdays and Thursdays 2-3pm
- Tutorials: Wednesdays 2-3pm, week 2-10
- Labs: Mondays 2-3pm, weeks 2 and 4

Assessment

- Labs (0%): practical
- Tutorials (0%): exercise sheets
- Coursework (30%): week 5
- Written exam (70%): April-May diet

Material



We can check the type of a vertex with:

```
print(diagram.type(2) == zx.VertexType.Z)
```

Similarly, we can add edges to the graph using:

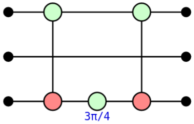
```
diagram.add_edge((0,1)) # Adds an edge between vertices 0 and 1
diagram.add_edge((1,2)) # Adds an edge between vertices 1 and 2
```

However, this isn't really the “standard” way to instantiate a ZX-diagram. It's much more common to obtain one from a quantum circuit of interest, and PyZX of course provides a method to do so. To convert our circuit to a diagram, we write:

```
circ_diagram = circ.to_graph()
```

We can then visualise the diagram with:

```
zx.draw(circ_diagram)
```



An interactive visualisation can be obtained using the PyZX graph editor. The editor can be used to manipulate vertices of the diagram, apply rewrite rules or even arbitrary edits to the diagram:

```
zx.editor.edit(circ_diagram)
```

Click “Save snapshot” to save the result back into `circ_diagram`. Within the PyZX editor, you will note that we can also apply some of the ZX-calculus rewrite rules to our diagram. Let's next see how

Introduction to Quantum Programming and Semantics

Week 1: ZX calculus

Chris Heunen

THE UNIVERSITY of EDINBURGH
informatics


Thank you
Your support powers our independent journalism

The Guardian
Newspaper of the year

News | Opinion | Sport | Culture | Lifestyle | More

UK World Climate crisis Ukraine Football Newsletters Business Environment UK politics Education Society Science Tech Global development Obituaries

The Observer
Physics

Sat 16 Dec 2023 14.00 GMT

70

Interview

Physicist Bob Coecke: 'It's easier to convince kids than adults about quantum mechanics'

Zeeya Merali

Your educational experiment involved 54 schoolchildren, aged 15-17, who were randomly selected from around 1,000 applicants, from 36 UK schools – mostly state schools. The teenagers spent two hours a week in online classes and after eight weeks were given a test using questions from an Oxford postgraduate quantum physics exam. More than 80% of the pupils passed and around half earned a distinction. Were you surprised by their success?
At one point, I was going to call off the whole thing because I thought it was going to be a complete disaster. We'd originally wanted the kids to interact with each other on social media or communicate online, but that wasn't allowed due to the ethical guidelines for the experiment. I thought, what sort of educational experience is it, if you can't talk to each other?

This is the Covid generation: none of them put their cameras on [for the online classes], so we were looking at a black screen. None of them asked questions using their voices, they just typed. It was a difficult teaching challenge by all standards. We also saw a self-esteem problem with the students. But the majority of kids liked that we had announced that you didn't need a complex maths background. The maths had been a barrier to kids who had wanted to access this knowledge.

And then we got back the numbers. They did significantly better than we see from university-level students. Exams were marked blind, so we don't know how many came in with the aim of pursuing Stem. We are processing that data now.

Overview

It's all about pictures:

- can manipulate like flow chart
- mathematically rigorous backend
- complete for quantum computation
- higher level than quantum circuits
- built up from basic elements: Z and X observables

Overview

ZX calculus:

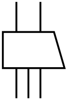
- Game
- Rules
- Interpretation
- Power
- Example

The game

Two-dimensional:

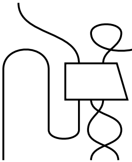
- Time goes up
- Space extends left and right
- Wires represent qubits
- Whole diagram represents process

Example process that takes 3 qubits and returns 2:



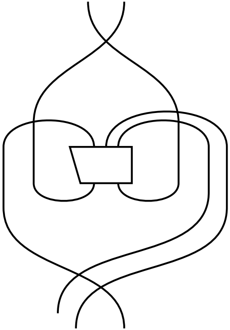
Only connectivity matters

Doesn't matter how exactly we draw wires:



Only connectivity matters

Doesn't even matter how boxes oriented:



Wedge breaks rotational symmetry

What goes inside the box?

Can build up from certain simple ones by placing side by side, or on top of each other and connecting input and output wires. Four basic ones:

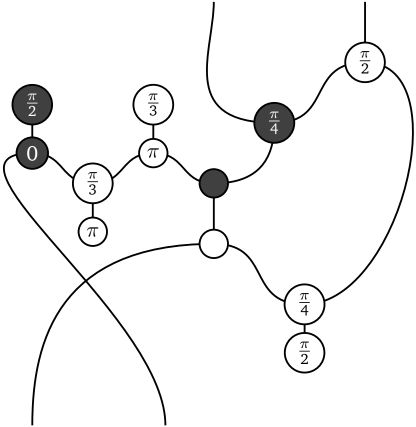
What goes inside the box?

Can build up from certain simple ones by placing side by side, or on top of each other and connecting input and output wires. Four basic ones:

$\alpha \in [0, 2\pi)$ can be any *phase*
Draw phase 0 as dot without label

Example

Process with 2 input qubits and 2 output qubits:

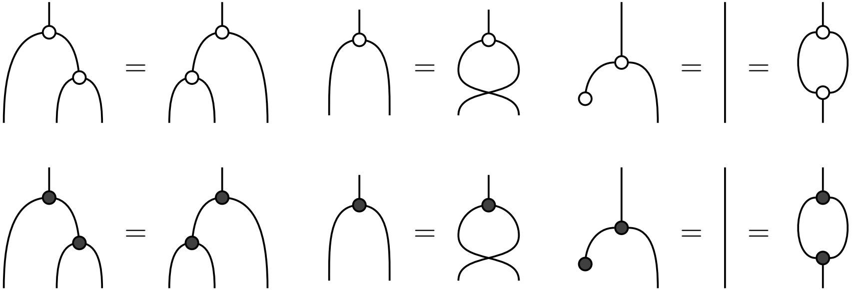


The rules

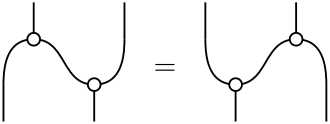
Two kinds:

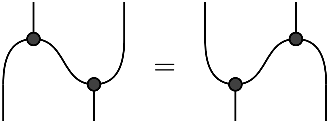
- Graphical rules: graph isotopy
- Axioms governing basic building blocks:
  9 groups making them behave as Z and X observables

Monoid rules

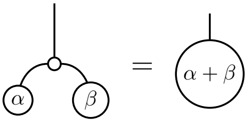


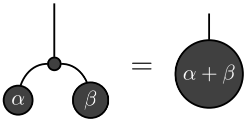
Frobenius rules





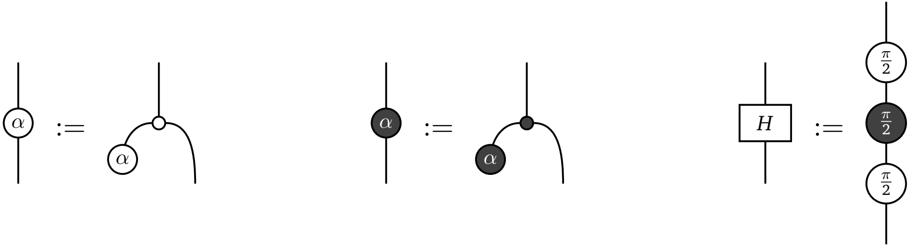
Fusion rules



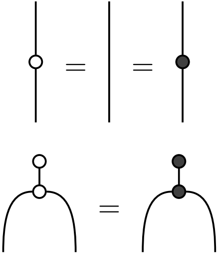


Identity rules

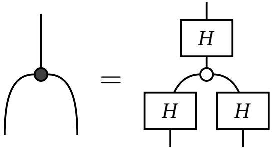
Shorthand notation:



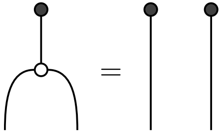
Identity rules



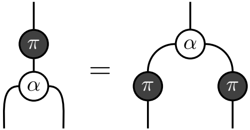
Colour change rule



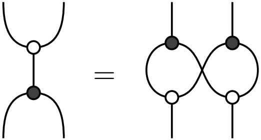
Copy rule



π-Copy rule

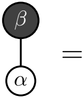


Bialgebra rule



Scalars rule

“Ignore global scalar factors”



The interpretation

What do all these rules mean?

- doesn't matter row dots of same colour connected, as long as phases add up (modulo 2π)
- think of wires as qubits/2-by-2 matrices:
  - white dots tell you how to multiply diagonal matrices (in Z basis)
  - black dots tell you how to multiply diagonal matrices (in X basis)
- what happens when changing between Z and X basis

Standard model

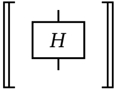
$$\left\llbracket \;\big|\; \right\rrbracket = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\left\llbracket \;\times\; \right\rrbracket = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

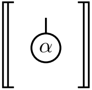
$$\left\llbracket \;\cup\; \right\rrbracket = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\left\llbracket \;\cap\; \right\rrbracket = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}$$

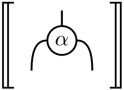
Standard model



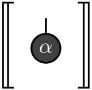
$$= \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$



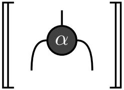
$$= \begin{pmatrix} 1 \\ e^{i\alpha} \end{pmatrix}$$



$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{i\alpha} \end{pmatrix}$$

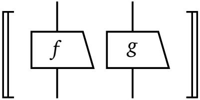


$$= \begin{pmatrix} 1+e^{i\alpha} \\ 1-e^{i\alpha} \end{pmatrix}$$

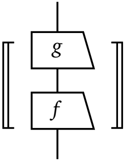


$$= \begin{pmatrix} 1+e^{i\alpha} & 1-e^{i\alpha} & 1-e^{i\alpha} & 1+e^{i\alpha} \\ 1-e^{i\alpha} & 1+e^{i\alpha} & 1+e^{i\alpha} & 1-e^{i\alpha} \end{pmatrix}$$

Standard model



$$= [\![f]\!] \otimes [\![g]\!]$$



$$= [\![g]\!] \circ [\![f]\!]$$

Game makes sense

Any graphical manipulations done with ZX diagrams yield valid equalities between matrices

Theorem (ZX calculus is sound): Let D_1, D_2 be diagrams in the ZX calculus. If D_1 equals D_2 under the axioms of the ZX calculus, then $[\![D_1]\!] = [\![D_2]\!]$.

Approximate universality

ZX calculus captures essence of quantum computation very efficiently

Any linear transformation from m qubits to n qubits can be approximated up to arbitrary precision.

Theorem (ZX calculus is approximately universal): For any 2^m-by-2^n matrix f, and any error margin $\varepsilon > 0$, there exists a diagram D in the ZX calculus, that only includes phases that are integer multiples of $\frac{\pi}{4}$, such that $\|[\![D]\!] - f\| < \varepsilon$.

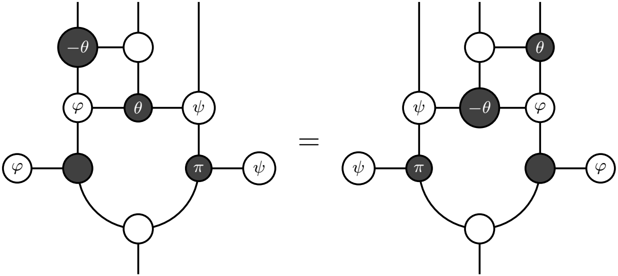
Completeness

If two matrices are equal, both given by ZX calculus diagrams, is there always a graphical proof of this using only the axioms of the ZX calculus?

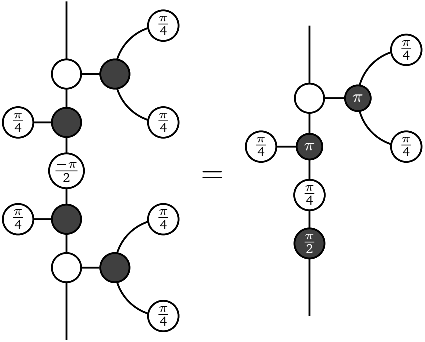
Completeness

If two matrices are equal, both given by ZX calculus diagrams, is there always a graphical proof of this using only the axioms of the ZX calculus?

Need two more axioms (that are sound):

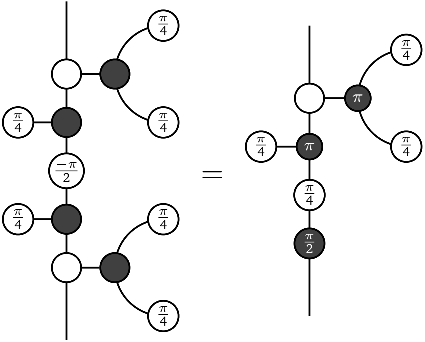


Completeness



for any phases φ, ψ, θ that are multiples of $\frac{\pi}{4}$. Let's call the ZX calculus with these two extra rules the $\frac{\pi}{4}$-*ZX calculus*.

Completeness



for any phases φ, ψ, θ that are multiples of $\frac{\pi}{4}$. Let's call the ZX calculus with these two extra rules the $\frac{\pi}{4}$-*ZX calculus*.

theorem (ZX calculus is complete): Let D_1, D_2 be diagrams in the $\frac{\pi}{4}$-ZX calculus. If $[\![D_1]\!] = [\![D_2]\!]$, then $D_1 = D_2$ under the axioms of the $\frac{\pi}{4}$-ZX calculus.

Automation

ZX calculus is very amenable to automation. All a ZX calculation really is, is a bunch of finite labelled graphs, and a sequence of one of finitely many rules. Computers can handle this very well, and in fact search for such proofs for us themselves.

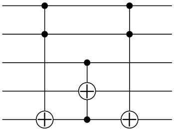
Quantum circuit optimisation

- Given quantum circuit may be very inefficient: many expensive (T) gates.
- Approximate universality: convert circuit into ZX diagram
- Manipulate ZX diagram using axioms: make it simpler
- Convert simplex ZX diagram back into circuit: fewer expensive gates.

Quantum circuit optimisation in practice

This optimisation strategy is in fact used in practice!
The currently best commercial state-of-the-art quantum circuit optimiser is based on this: t|ket⟩ by Cambridge Quantum Computing.

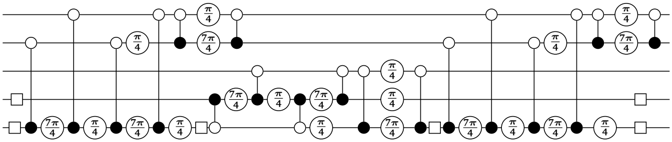
Original circuit:



Quantum circuit optimisation in practice

This optimisation strategy is in fact used in practice!
The currently best commercial state-of-the-art quantum circuit optimiser is based on this: t|ket⟩ by Cambridge Quantum Computing.

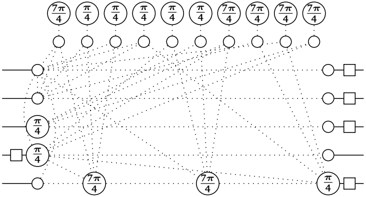
Circuit expanded as ZX diagram. Small boxes are Hadamard gates.
There are 21 T gates (dots with phases that are not multiples of $\pi/2$):



Quantum circuit optimisation in practice

This optimisation strategy is in fact used in practice!
The currently best commercial state-of-the-art quantum circuit optimiser is based on this: t|ket⟩ by Cambridge Quantum Computing.

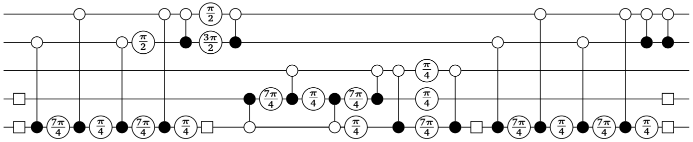
Simplified ZX diagram. Dotted lines are wires carrying a Hadamard gate.



Quantum circuit optimisation in practice

This optimisation strategy is in fact used in practice!
The currently best commercial state-of-the-art quantum circuit optimiser is based on this: t|ket⟩ by Cambridge Quantum Computing.

After optimisation 15 T gates remain:



Quantum circuit optimisation

Circuit	number of qubits	number of T gates	best previous method	ZX calculus
adder_8	24	399	213	173
Adder8	23	266	56	56
Adder16	47	602	120	120
Adder32	95	1274	248	248
Adder64	191	2618	504	504
barenco-tof3	5	28	16	16
barenco-tof4	7	56	28	28
barenco-tof5	9	84	40	40
barenco-tof10	19	224	100	100
tof_3	5	21	15	15
tof_4	7	35	23	23
tof_5	9	49	31	31
tof_{10}	19	119	71	71
csla-mux_3	15	70	58	62
csum-mux_9	30	196	76	84
cycle17_3	35	4739	1944	1797
$\text{gf}(2^4)\text{-mult}$	12	112	56	68
$\text{gf}(2^5)\text{-mult}$	15	175	90	115
$\text{gf}(2^6)\text{-mult}$	18	252	132	150
$\text{gf}(2^7)\text{-mult}$	21	343	185	217
$\text{gf}(2^8)\text{-mult}$	24	448	216	264
ham15-low	17	161	97	97
ham15-med	17	574	230	212
ham15-high	20	2457	1019	1019
hwb_6	7	105	75	75
hwb_8	12	5887	3531	3517
mod-mult-55	9	49	28	35
mod-red-21	11	119	73	73
mod5_4	5	28	16	8
nth-prime_6	9	567	400	279
nth-prime_8	12	6671	4045	4047
qcla-adder_{10}	36	589	162	162
qcla-com_7	24	203	94	95
qcla-mod_7	26	413	235	237
rc-adder_6	14	77	47	47
vbe-adder_3	10	70	24	24

Summary

ZX calculus:

- intuitive: all about pictures
- rigorous: mathematical backend
- sound: what you can do graphically you can do algebraically
- complete: what you can do algebraically you can do graphically
- approximately universal: can express any quantum computation
- higher level than quantum circuits
- automation: will do yourself in lab session
- circuit optimisation: state of the art