Introduction to Quantum Programming and Semantics: ZX calculus

Chris Heunen

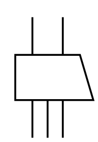
Spring 2024

This lecture gives a taste of the end product of the theory we will be building up in the rest of the course. It is a graphical calculus, called the ZX calculus. We can manipulate these pictures as if it were a flow chart. There are several special elements of these pictures that represent Z and X observables in quantum computing - hence the name - and several special rules about how to combine and equate such pictures.

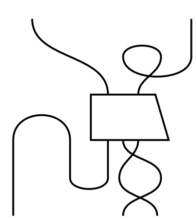
We will do this in three stages. First, *the game* itself: what kind of pictures are allowed in the first place? Second, *the rules*: which moves are allowed to change one picture into another? Third, *the interpretation*: what do the pictures mean, both intuitively, and in terms of matrices? Fourth, we give some examples of the *power* of this game: what kind of things can we do using these pictures that were hard to do without?

The game

We will draw a kind of flow chart diagrams on the two-dimensional page. Time goes up, space extends left and right. Wires represent qubits. The whole diagram represents a process, that takes a number of input qubits, does something to them, and eventually returns a number of output qubits. For example, here is a process that takes 3 qubits and returns 2:

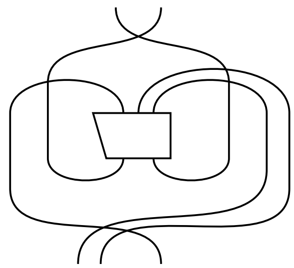


In such diagrams, it doesn't matter how exactly we draw the wires. All that matters is the *connectivity* of which wires connect which processes. For example, the above process is the exact same process as the following:



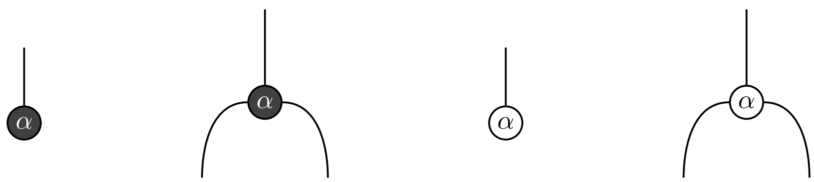
In fact, it doesn't even matter how we orient the box itself, as long as it stays connected in the same way.

We could also draw the process above as follows:



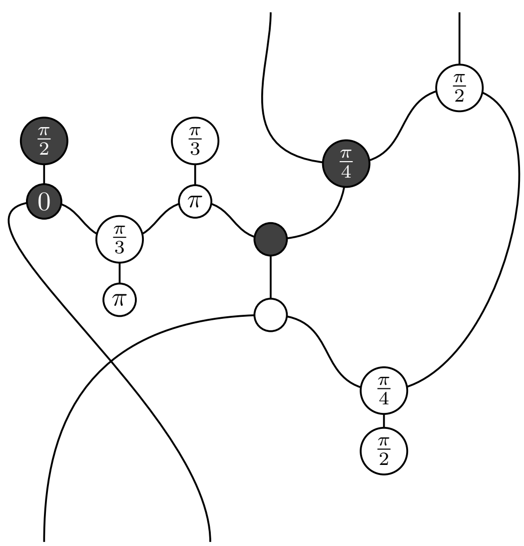
This invariance of the diagram under rotation is why we drew a wedge on the box; that way, we can remember the connectivity of what goes on inside the box.

But what goes inside the box? Well, we can build up such boxes from certain simple ones by placing them side by side, or on top of each other and connecting input and output wires. Here are the four simple processes from which we build up all others:



Here α can be any *phase*, i.e. any real number between 0 and 2π. So really, there aren't just four basic processes, but infinitely many of them, but there are still just four types of basic processes. For the phase 0, we will also draw a dot with no label in it.

For example, from these pieces we can build the following process with 2 input qubits and 2 output qubits:



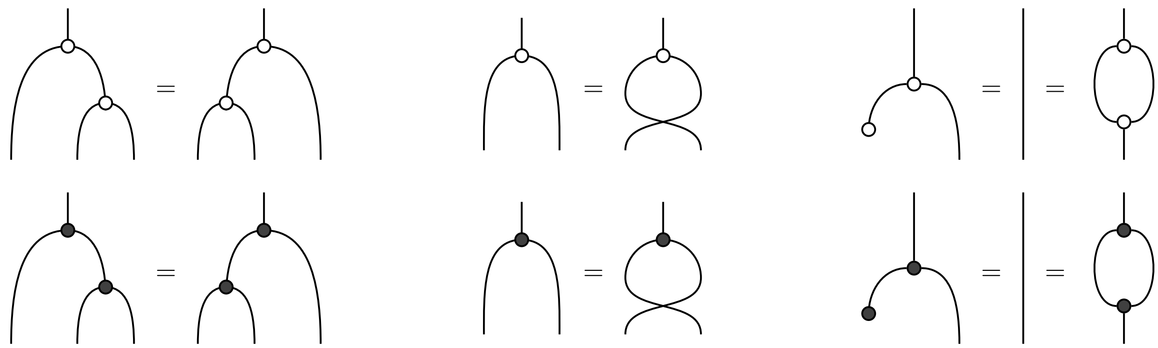
That's all. Those are the pieces of the game.

The rules

There are two types of rules following which we may manipulate diagrams. First, the *graphical rules* that we already talked about above. We may deform a diagram in any way we like, as long as the connectivity is respected. This is called a *graph isotopy* with a big word, and we will make this notion more precise in the course. For now, imagine a big frame around the whole diagram, so that the input wires are fixed to the lower edge of the frame, and the output wires are fixed to the upper edge of the frame. Similarly, the wires are fixed to the dots in a way that their order around the circle cannot change. Two diagrams are considered equal if there is a way to smoothly move the wires around, while keeping these endpoints fixed, turning the one diagram into the other. if you like you can think of such a graphical rewrite as a movie that morphs one diagram into another with the same connectivity. Try to imagine such a movie for the first three diagrams at the start of this lecture.

The second type of rule pertains to the basic building blocks. These rules govern how you may combine several building blocks. There are quite a few of them, and we will now simply list them.

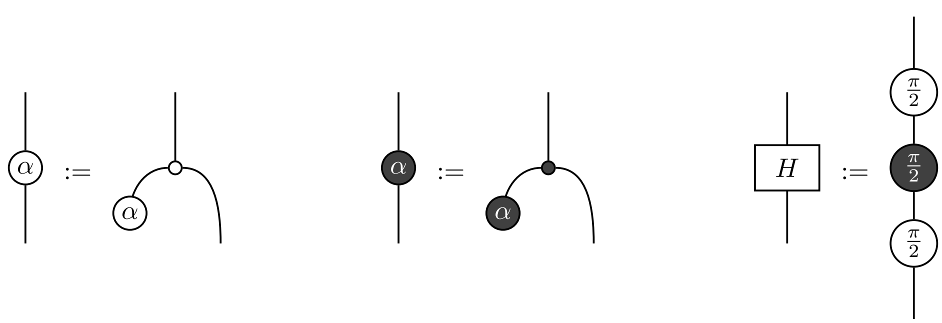
Monoid rules



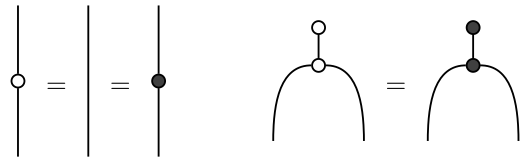
Frobenius rules

Fusion rules

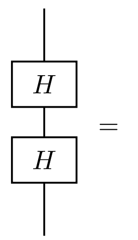
Identity rules For the following rules we first introduce a bit of shorthand notation:



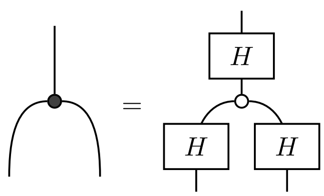
The identity rules now say:



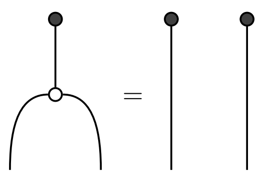
Hadamard rule



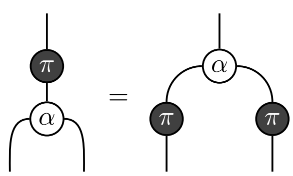
Colour change rule



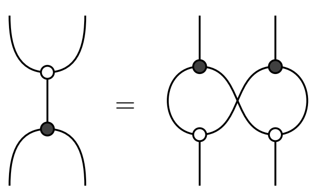
Copy rule



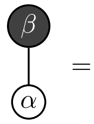
π-Copy rule



Bialgebra rule



Scalars rule

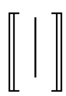


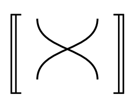
This rule says that we will ignore global scalar factors. It is more of a convenience, and we could have dropped this rule for the price of carefully inserting scalars into all the other rules.

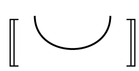
The interpretation

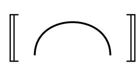
What do all these rules mean? Taken together, the monoid rules, Frobenius rules, fusion rules, and identity rules intuitively say that it doesn't matter how dots of the same colour are connected, as long as the phases inside the dots add up to the same number (modulo 2π). If you think of the wires as qubits, i.e. 2-by-2 matrices, you can read this as saying that the white dots tell you how to multiply matrices that are diagonal in the standard (Z) basis that have complex numbers of the form $e^{i\alpha}$ as eigenvalues. The same holds for black dots, but then in the (X) basis given by the basis vectors $|+\rangle = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$ and $|-\rangle = (\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$. The colour change rule tells us how to convert from one basis to the other. The bialgebra rule says that these two bases are at a maximal angle to one another, or complementary. The copy and π-copy rules are happy coincidences that hold for these two complementary bases.

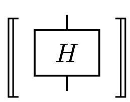
Indeed, there is a *standard model*. This interprets a diagram of the ZX-calculus as an actual matrix in a way that respects the rules:

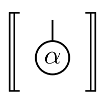
 $= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

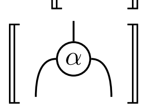
 $= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

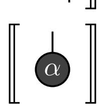
 $= \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$

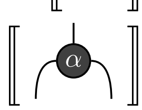
 $= \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}$

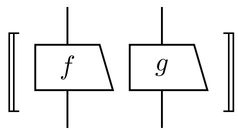
 $= \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

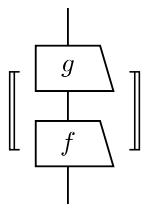
 $= \begin{pmatrix} 1 \\ e^{i\alpha} \end{pmatrix}$

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

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

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

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

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

This makes the intuitive interpretation very precise. All the rules remain valid under this standard interpretation. Try it out! In other words, this standard interpretation is *sound*: any graphical manipulations done with ZX diagrams yield valid equalities between matrices under the standard interpretation.

Theorem 0.1 (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 $[\![D_1]\!] = [\![D_2]\!]$.* □

In other words still: the rules of the game make sense.

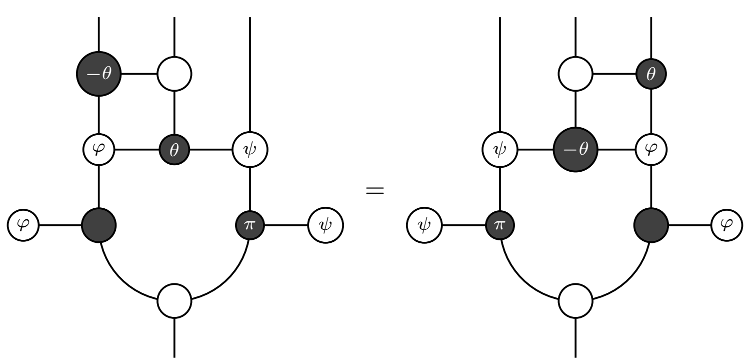
The power

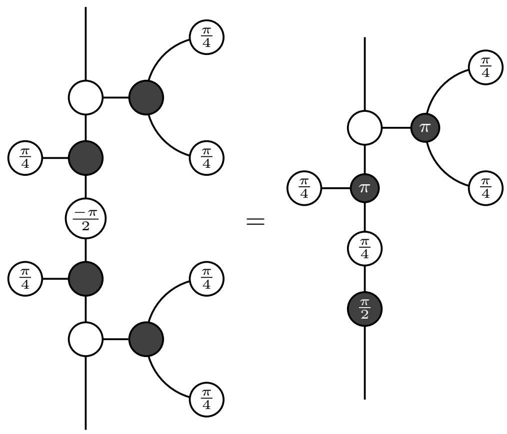
Finally, let us showcase without proof a few ways in which the ZX calculus captures the essence of quantum computation and is very efficient at it.

First, any possible linear transformation from m qubits to n qubits can be approximated up to arbitrary precision with ZX diagrams. In other words, the ZX calculus is *approximately universal*.

Theorem 0.2 (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 $\|[\![D]\!] - f\| < \varepsilon$.*

Second, the ZX calculus is *complete*. That is, if two matrices are equal, and are both given by some ZX calculus diagrams, is there always a graphical proof of this using only the axioms of the ZX calculus? This is the case if we assume the following further two axioms (that are sound under the standard interpretation):





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 0.3 (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.*

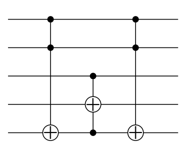
Third, the ZX calculus is very amenable to automation. All a ZX calculation really is, is a bunch of finite labelled graphs, and a sequence of one of finitely many rules. Computers can handle this very well, and in fact search for such proofs for us themselves.

One of the best examples of this is *quantum circuit optimisation*. Given a quantum algorithm in the form of a quantum circuit, it may be very inefficient, in the sense that it contains a lot of gates that are costly to implement in practice. The standard expensive gate is the T gate. Now, thanks to the approximate universality of the ZX calculus, you can convert any quantum circuit into a ZX diagram. Then you can manipulate that ZX diagram to your heart's content, fusing dots until it has become a lot simpler. Finally, you can, with some luck, convert the simpler ZX diagram back into a quantum circuit, that now has fewer of the expensive (e.g. T) gates.

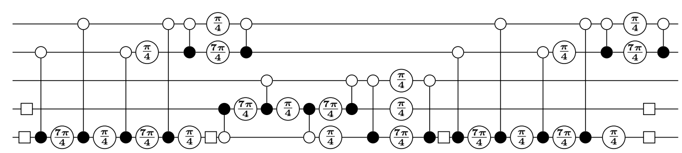
This strategy for quantum circuit optimisation is in fact used in practice. The currently best commercial state-of-the-art quantum circuit optimiser is based on this theory: t$|$ket$\rangle$ by Cambridge Quantum Computing. Below are some examples of what ZX calculus can do, taken from a research paper [arXiv:1903.10477].

In the lab session you will do some of this yourself!

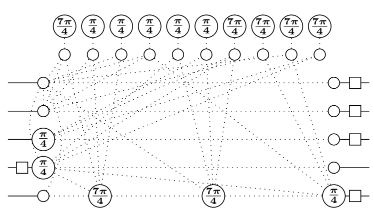
Circuit	number of qubits	number of T gates	best previous method	ZX calculus
adder_8	24	399	213	173
Adder8	23	266	56	56
Adder16	47	602	120	120
Adder32	95	1274	248	248
Adder64	191	2618	504	504
barenco-tof3	5	28	16	16
barenco-tof4	7	56	28	28
barenco-tof5	9	84	40	40
barenco-tof10	19	224	100	100
tof_3	5	21	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



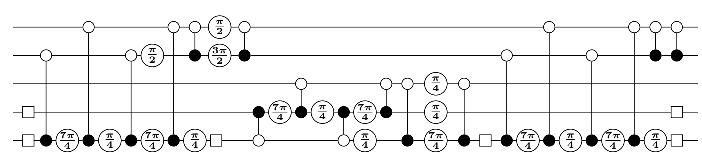
(a) Original circuit



(b) The circuit expanded as a ZX diagram. The small boxes are Hadamard gates. There are 21 T gates, i.e. dots with phases that are not multiples of $\pi/2$.



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



(d) 15 T gates remain after optimisation.

Figure 1: Quantum circuit optimisation using ZX calculus