THE ζ-CALCULUS

Nicklas Botö[1] Fabian Forslund[1]

[1]Chalmers University of Technology

We propose a language model ζ for describing rotational computation over a set of orthogonal axes, as **abstraction bases**. Terms of ζ are translated to categorical string diagrams, with semantics dependent on the chosen bases of abstraction. Using different numbers of bases yields languages capable of transforming values represented by the surfaces of hyperspheres of increasing dimension. In the case of 2 abstraction bases we model the **ZX-calculus** for quantum computation, while 0 bases gives the **λ-calculus**. The case for 4 bases is also explored, yielding a language over the **quaternionic Hopf fibration** as an extended Bloch sphere.

The ζ-calculus is then a rotational generalisation of the classical λ-calculus, fitting especially well for quantum computation in the ZX-calculus. We hope that the proposed model will serve as a stronger connection between diagrammatic calculi like the ZX-calculus, and functional programming languages.

Syntax

We define the language L_ζ of ζ-terms, compositions of abstractions and values parameterized by a phase, over a set of abstraction bases Ξ inductively. These abstraction bases are introduced by its own syntactic rules.

$$\frac{}{* \in L_\zeta} \qquad \frac{\beta \in \Xi \quad \alpha \in \arg\beta \quad n \in \mathbb{N}}{\beta^\alpha_n \in L_\zeta} \qquad \frac{\beta \in \Xi \quad \alpha \in \arg\beta \quad M \in L_\zeta}{\beta^\alpha \vec{x} \triangleright M[\vec{x}] \in L_\zeta}$$

$$\frac{M \in L_\zeta \quad \vec{x} = \mathrm{FV}(M)}{\lambda\vec{x} \triangleright M[\vec{x}] \in L_\zeta} \qquad \frac{M, N \in L_\zeta}{MN \in L_\zeta} \qquad \frac{M, N \in L_\zeta}{\langle M, N\rangle \in L_\zeta} \qquad \frac{M \in L_\zeta}{M^\dagger \in L_\zeta}$$

Fig. 1: Inductive syntactic rules.

The λ-abstraction is also treated as an implicit nullary abstraction basis among Ξ.

The ZX-calculus

Modelling the ZX-calculus is achieved by setting the abstraction bases to $\Xi = \{\mathbf{Z}, \mathbf{X}\}$ and including the abstraction $\mathbf{H}$. From the translation to categorical string diagrams, the semantics is given by the equational theory of ZX.

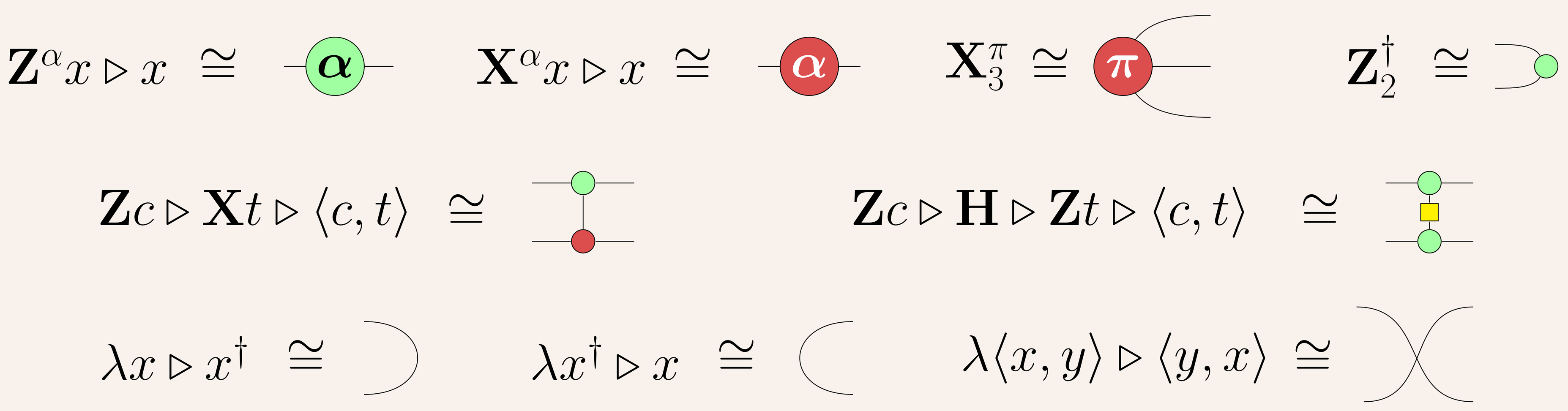


Fig. 2: Example ZX-progams as ζ-terms.

Figure 2 demonstrates some common diagrams in the ZX-calculus as ζ-terms.

The ZXPT-calculus

The third computational order is modelled four orthogonal abstraction bases $\Xi = \{\mathbf{Z}, \mathbf{X}, \mathbf{P}, \mathbf{T}\}$ over the 2-vector space of quaternions $\mathbb{H}^2$. These bases arise from the axes of the 4-sphere which is the projective space of the **quaternionic Hopf fibration** $S^3 \hookrightarrow S^7 \to S^4$, similar to the Bloch sphere as the complex projective space of the unit 3-sphere.

We define three other abstractions, denoted $\{\eta_X, \eta_Y, \eta_E\}$. In diagrams these abstractions are shown as boxes of their respective base colour (e.g. Hadamard is η_X, the red box). By extending the **colour change rule** to general bases and their η-abstractions as $\eta_j^\dagger \beta_1 \eta_j = \beta_2$. The η-abstractions relate the abstraction bases according to figure 3, where nodes are bases β_1, β_2, with the edge connecting them η_j.

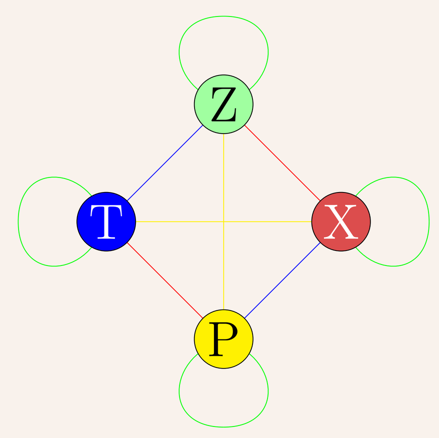


Fig. 3: Extended colour change rule.

Semantics

The semantics of a language in the model is given by an equational theory over the graph translation of terms. Terms in the language are translated by the following rules.

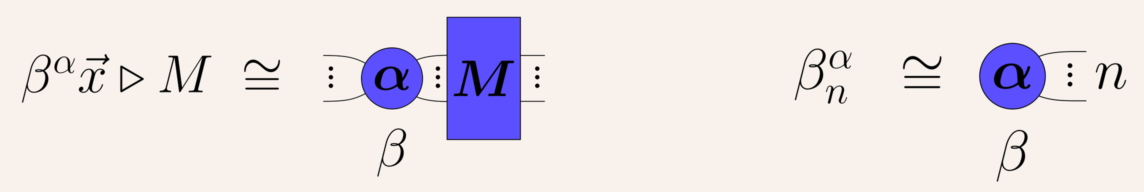


Fig. 4: Graph translations of abstractions and values.

Other connectives are translated to a tensor network of their term translations. The unit atom $*$ is translated to an empty graph with scalar 1.

Stacked abstractions connect their head nodes to simulate an interaction between them as in figure 5. This interaction can then be adjusted by injecting a basis into the term.

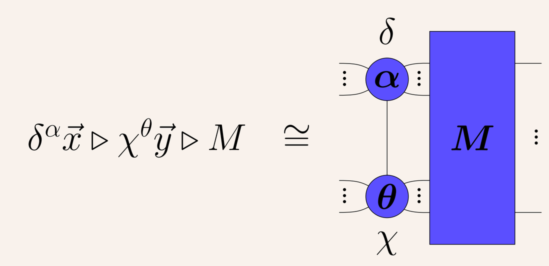
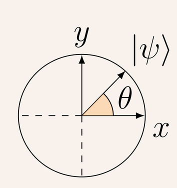
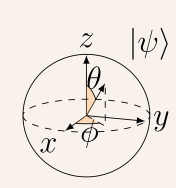


Fig. 5: Graph translations of stacked abstractions.

To control this for a term $\delta x \triangleright \delta y \triangleright M$, we can inject another basis η to prevent the head nodes from fusing, $\delta x \triangleright \eta \triangleright \delta y \triangleright M$.

Orders of computation

We denote a set of orthogonal abstraction bases over a vector space by a computation of order ᛟ$_n$ (*othal*). The vector spaces below are given in terms of 2-vectors corresponding to binary qubits of the given order, but any vector space dimension may be given; e.g dimension 3 for qutrits. For each order, a graphical calculus is also given by the ζ representation.

	ᛟ$_0$	ᛟ$_1$	ᛟ$_2$	ᛟ$_3$	...?
Bases	$\emptyset$	$\mathbf{Z}$	$\mathbf{Z}, \mathbf{X}$	$\mathbf{Z}, \mathbf{X}, \mathbf{P}, \mathbf{T}$	
Spaces	$\top^2$	$\mathbb{R}^2$	$\mathbb{C}^2$	$\mathbb{H}^2$	
Spiders	–				
n-Bloch	$\bullet$			S^4	

Tab. 1: Abstraction bases and the spaces they act on. The set $\mathbb{H}$ are the quaternions.

As the λ-basis is always implicit, the simplest order of computation ᛟ$_0$ models only a λ-calculus over a unit value $\bullet$.

Conclusion and Further Work

We've presented a language model for rotational computation in a given dimension. The power of this model ranges from a simple λ-calculus, to complex numbers and qubits, and beyond. This model is still in its very early stages of development but we hope to use it as a formalism for a functional hybrid programming language. We are currently working on proving that translations both from and to graphs are complete in the sense that every graph can be constructed as an equivalent ζ-term using some given bases Ξ.

We have demonstrated how to model the well known ZX-calculus in this language, resulting in a concise, functional representation. Going from the ZX-calculus we defined **orders of computation dependent on the dimensionality** of amplitudes in a given state space. Using the ZXPT-calculus of order ᛟ$_3$ it may be possible to describe a language over **pairs of entangled qubits**. Defining a subset of transformations in ᛟ$_3$ that can run efficiently on a quantum computer, like **Clifford computation** in ᛟ$_2$, is currently being worked on.

Author contact: Nicklas Botö (`boton@chalmers.se`) & Fabian Forslund (`fabfors@chalmers.se`). `https://zeta.nicbot.xyz` Chalmers University of Technology (Chalmers Tekniska Högskola), Gothenburg, Sweden