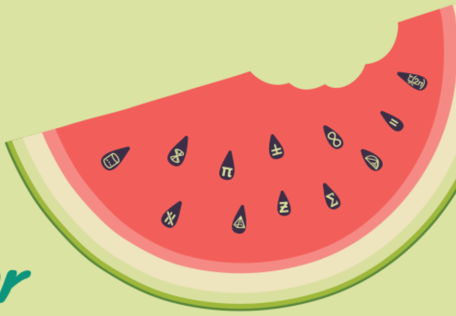
AMSI SUMMER RESEARCH SCHOLARSHIPS 2024–25

Get a taste for Research this Summer



Is a picture worth a thousand words? String diagrams for quantum game theory

Emily Sykes

Supervised by Dr Frank Valckenborgh and Shay Tobin

Macquarie University

Jane Street

AMSI AUSTRALIAN MATHEMATICAL SCIENCES INSTITUTE

Abstract

String diagrams are a mathematically sound and complete diagrammatic language that can be used to represent quantum processes. In particular, string diagrams show information flow and compositionality in categorical quantum mechanics. In this report, we develop the diagrammatic language using paradigms of monoidal categories with reference to the category of finite-dimensional Hilbert spaces. We also apply string diagrams to represent some processes in quantum game theory, including the quantum penny flip and the quantum prisoner's dilemma. It was found that string diagrams can effectively show information flow in quantum games.

1 Introduction

Categorical quantum mechanics is the study of quantum information using paradigms of monoidal category theory. A notable application of categorical quantum mechanics are string diagrams which accurately capture the compositional structure of quantum processes in a diagrammatic language. String diagrams can, arguably, represent some ideas, such as the quantum teleportation protocol, in more elegant ways than algebraic means (Coecke and Kissinger, 2017; Heunen and Vicary, 2019). In particular, string diagrams provide an intuitive visualisation of information flow which is just as mathematically sound as an algebraic counterpart. The first aim of this report is to synthesise and connect existing ideas in categorical quantum mechanics and diagrammatic reasoning in quantum processes.

Quantum game theory is a field of research that imagines multi-player strategy scenarios imbued with quantum properties including entanglement and superposition. The second aim of this report is to model various quantum games using string diagrams. While string diagrams have previously been used to visualise classical games (Hedges et al., 2016), no significant work has been done to model quantum games in this way. By examining game components including initial states, "moves", and measurements to determine the game's result, quantum games such as the penny flip and The prisoner's dilemma were successfully modelled using string diagrams. The benefits of modelling quantum games in this way are discussed. In particular, string diagrams for quantum games provide an intuitive way to visualise information flow while being mathematically sound and complete. Additionally, string diagrams are manipulable using conventions of the language when specific initial states and/or moves are considered. Some game conditions considered in this report are those which yield an equilibrium point of the game.

2 Statement of Authorship

Section 3 relies on work from Coecke and Kissinger (2017), Heunen and Vicary (2019) and Selinger (2011). The author summarises key ideas and connects relevant information. The game theoretic aspects of Section 4 are based on the work of Meyer (1999) and Eisert et al. (1999). All string diagrammatic interpretations of quantum games are the sole work of the author.

3 Categorical Quantum Mechanics

String diagrams are a formal diagrammatic language which represent morphisms and objects in monoidal categories. Throughout this section, this diagrammatic language is developed with interpretations for Hilbert spaces in quantum mechanics.

3.1 Monoidal Graphical Notation

Definition 3.1 (Category)**.** A *category* $\mathbf{C}$ consists of:

- a class $\mathrm{Ob}(\mathbf{C})$ of *objects*, denoted A, B, C,...;
- for each pair of objects A, B, a class $\mathbf{C}(A, B)$ of *morphisms*, with $f \in \mathbf{C}(A, B)$ written $A \xrightarrow{f} B$;

- for every pair of morphisms $A \xrightarrow{f} B$ and $B \xrightarrow{g} C$, with the codomain of f corresponding with the domain of g (i.e. a common intermediate object), a *composite* operation $A \xrightarrow{g\circ f} C$;
- for every object A, an *identity* morphism $A \xrightarrow{\mathrm{id}_A} A$.

For all objects A, B, C, D and morphisms $A \xrightarrow{f} B$, $B \xrightarrow{g} C$, $C \xrightarrow{h} D$, the following properties are satisfied:

- *associativity*:
$$h \circ (g \circ f) = (h \circ g) \circ f; \tag{3.1}$$
- *identity*:
$$\mathrm{id}_B \circ f = f = f \circ \mathrm{id}_A. \tag{3.2}$$

Definition 3.2. (Isomorphism) A morphism $A \xrightarrow{f} B$ is an *isomorphism* when it has an *inverse* morphism satisfying:
$$f^{-1} \circ f = \mathrm{id}_A \qquad\qquad f \circ f^{-1} = \mathrm{id}_B \tag{3.3}$$

Finite-dimensional Hilbert spaces (objects) and bounded linear maps (morphisms) form a category, **FHilb**, where composition is composition of linear maps and identity morphisms are identity linear maps.

A functor is a map between categories.

Definition 3.3. (Functor) Given categories **C** and **D**, a *functor* $F : \mathbf{C} \longrightarrow \mathbf{D}$ is defined by the following data:

- for each object $A \in \mathrm{Ob}(\mathbf{C})$, an object $F(A) \in \mathrm{Ob}(\mathbf{D})$;
- for each morphism $A \xrightarrow{f} B$ in **C**, a morphism $F(A) \xrightarrow{F(f)} F(B)$ in **D**.

These data must satisfy the properties:

- $F(g \circ f) = F(g) \circ F(f)$ for all morphisms with a common intermediate object in **C**;
- $F(\mathrm{id}_A)$=$\mathrm{id}_{F(A)}$ for every object A in **C**.

Functors preserve identity morphisms and composition of morphisms. Preservation of composition is shown in the following commutative diagrams for $A, B, C \in \mathrm{Ob}(\mathbf{C})$ and $F(A), F(B), F(C) \in \mathrm{Ob}(\mathbf{D})$:

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[dr, "g\circ f"'] & B \arrow[d, "g"] \\ & C \end{tikzcd} \qquad\qquad \begin{tikzcd} F(A) \arrow[r, "F(f)"] \arrow[dr, "F(g\circ f)"'] & F(B) \arrow[d, "F(g)"] \\ & F(C). \end{tikzcd}$$

A natural transformation is a map between functors preserving specified actions, symmetries, or other structures. Natural transformations can be thought of as "repackaging" functors.

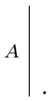
Definition 3.4. (Natural transformation, natural isomorphism) Given two functors $F : \mathbf{C} \longrightarrow \mathbf{D}$ and $G : \mathbf{C} \longrightarrow \mathbf{D}$, a *natural transformation* $\zeta : F \Longrightarrow G$ assigns every object A in **C**, a morphism $F(A) \xrightarrow{\zeta_A} G(A)$ in **D**, such that the following diagram commutes for every morphism $A \xrightarrow{f} B$ in **C**:

$$\begin{tikzcd} F(A) \arrow[r, "\zeta_A"] \arrow[d, "F(f)"'] & G(A) \arrow[d, "G(f)"] \\ F(B) \arrow[r, "\zeta_B"] & G(B). \end{tikzcd}$$

If, for every object A in **C**, ζ_A is an isomorphism, then ζ is a *natural isomorphism.*

While one can work with categories algebraically, it is useful to consider a graphical notation which can be used to visualise processes and is useful for exploring the compositional nature of physical quantum processes.

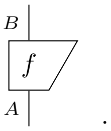
An object A can be represented as a vertically-oriented wire:



(3.4)

String diagrams are read from bottom to top. Object A is the "input" at the bottom of the diagram and the "output" at the top of the diagram. This wire can then be thought of as depicting the identity morphism $A \xrightarrow{\mathrm{id}_A} A$.

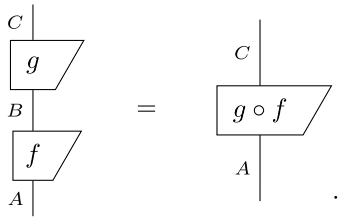
A morphism $A \xrightarrow{f} B$ can be represented as a node with an input object A and an output object B:



(3.5)

Morphism nodes are depicted as trapeziums with broken symmetry as rotating and flipping these nodes has additional meaning.

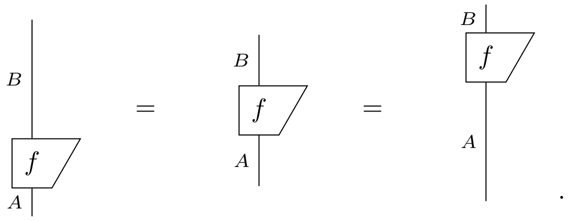
Composite operations $A \xrightarrow{g\circ f} C$ for morphisms $A \xrightarrow{f} B$ and $B \xrightarrow{g} C$, with a common intermediate object, can be depicted by connecting the output of the first node to the input of the second node:



(3.6)

This is known as *sequential composition.*

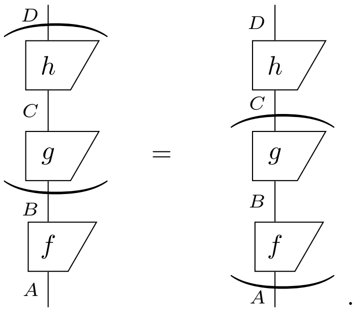
The identity law for categories, $\mathrm{id}_B \circ f = f = f \circ \mathrm{id}_A$, becomes:



(3.7)

So, morphism nodes may be slid along a straight wire without changing the diagram.

The associativity law, $h \circ (g \circ f) = (h \circ g) \circ f$, becomes:



(3.8)

We can, therefore, unambiguously write "chains" of sequentially composed morphisms without brackets.

In quantum mechanics, not only can processes occur sequentially, one after the other, but processes can also occur in parallel. Monoidal categories are equipped to deal with this additional structure.

Definition 3.5 (Monoidal Category)**.** A *monoidal category* is a category $\mathbf{C}$ with the following additional structure:

- a *tensor product* functor $\otimes : \mathbf{C} \times \mathbf{C} \longrightarrow \mathbf{C}$;
- a *unit object* $I \in \mathrm{Ob}(\mathbf{C})$;
- an *associator* natural isomorphism $(A \otimes B) \otimes C \xrightarrow{\alpha_{A,B,C}} A \otimes (B \otimes C)$;
- a *left unitor* natural isomorphism $I \otimes A \xrightarrow{\lambda_A} A$;
- a *right unitor* natural isomorphism $A \otimes I \xrightarrow{\rho_A} A$.

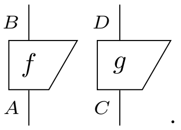
These data are subject to two coherence axioms; the "triangle axiom" and "pentagon axiom" (see Appendix A).

For the category **FHilb**, the functor $\otimes$ corresponds to the actual tensor product of finite-dimensional Hilbert spaces. Other structures can be chosen such as the direct sum of Hilbert spaces. However, the tensor product is chosen since it describes the state space of a composite system in quantum theory (Heunen and Vicary, 2019).

Two basic composition operations have now been introduced which can be summarised as follows:

$$f \circ g := \text{“process } f \text{ occurs after process } g \text{ occurs”;}$$
$$f \otimes g := \text{“process } f \text{ occurs while process } g \text{ occurs”.}$$

Graphically, $A \otimes C \xrightarrow{f \otimes g} B \otimes D$ for $A \xrightarrow{f} B$ and $C \xrightarrow{g} D$ can be depicted as follows:



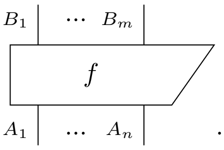
(3.9)

This is known as *parallel composition*. This notation which includes empty space between nodes f and g reinforces the fact that the morphisms are occurring independently.

Graphically, the unit object I is depicted as the empty diagram:

(3.10)

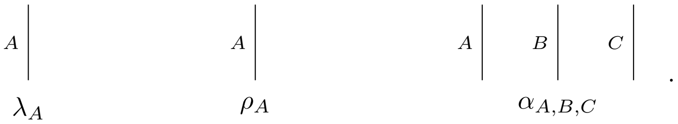
A morphism may also have multiple input and output wires. In general, the morphism $A_1 \otimes ... \otimes A_n \xrightarrow{f} B_1 \otimes ... \otimes B_m$ is depicted as:



(3.11)

Sequential composition requires only a one-dimensional (linear) graphical calculus. Since monoidal categories have an additional way to combine morphisms, the graphical calculus is now two-dimensional (planar). Not only is the graphical calculus a useful notation, it is also a sound and complete language for working with monoidal categories (Selinger, 2011).

The left unitor $I \otimes A \xrightarrow{\lambda_A} A$, the right unitor $A \otimes I \xrightarrow{\rho_A} A$ and the associator $(A \otimes B) \otimes C \xrightarrow{\alpha_{A,B,C}} A \otimes (B \otimes C)$ are simply not depicted in the graphical language:



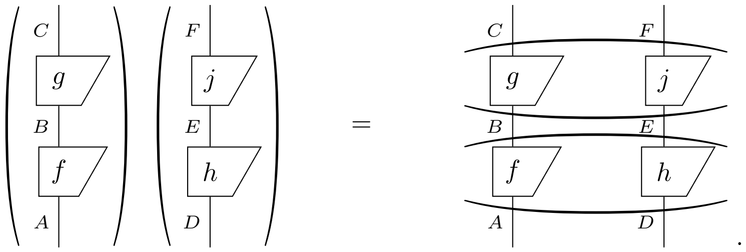
(3.12)

This makes the graphical language particularly powerful as the axioms of monoidal categories become a natural consequence of notation. Another example of this is shown in the *interchange law*. Any morphisms $A \xrightarrow{f} B$, $B \xrightarrow{g} C, D \xrightarrow{h} E$ and $E \xrightarrow{j} F$ in a monoidal category satisfy the interchange law:

$$(g \circ f) \otimes (j \circ h) = (g \otimes j) \circ (f \otimes h). \tag{3.13}$$

This law is a consequence of the fact that $\otimes$ is a functor which preserves composition between the categories $\mathbf{C} \times \mathbf{C}$ and $\mathbf{C}$. See Appendix B for an algebraic proof and definition of a product category.

Graphically, the interchange law becomes obvious:



(3.14)

The brackets indicate the order in which each diagram was formed. If the brackets are removed, we can see the two diagrams are exactly the same.

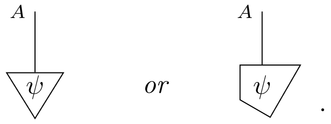
3.1.1 States and effects

Some processes have no inputs and some processes have no outputs.

A process with no inputs is a *state*. States can be thought of as a way to bring objects into being. Operationally, states are "preparation procedures" (Coecke and Kissinger, 2017). Formally:

Definition 3.6 (State)**.** A *state* of an object A is a morphism $I \xrightarrow{\psi} A$.

Graphically, states are drawn as triangles to explicitly show the lack of inputs. As with morphisms, often the symmetry of the nodes needs to be broken so states will also be drawn as funny looking quadrilaterals:



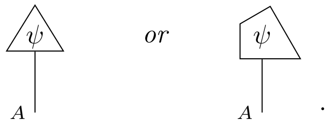
(3.15)

In the category **FHilb**, states of a Hilbert space H are linear functions $\mathbb{C} \to H$ which correspond to elements of H by considering the image of $1 \in \mathbb{C}$ (Heunen and Vicary, 2019).

A process with no outputs is an *effect*. Operationally, effects are used to model "tests" (Coecke and Kissinger, 2017). Formally:

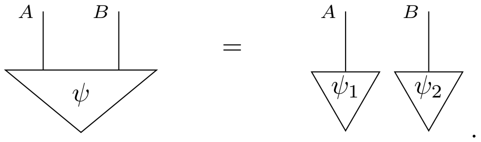
Definition 3.7 (Effect)**.** An *effect* for an object A is a morphism $A \xrightarrow{\psi} I$.

Graphically, effects are depicted as:



(3.16)

Definition 3.8 (Product state, entangled state)**.** A bipartite state $I \xrightarrow{\psi} A \otimes B$ is a joint state of objects A and B of a monoidal category. A bipartite state is a *product state* ($\otimes$-separable) when it is of the form $I \xrightarrow{\lambda^{-1}} I \otimes I \xrightarrow{\psi_1 \otimes \psi_2} A \otimes B$ for $I \xrightarrow{\psi_1} A$ and $I \xrightarrow{\psi_2} B$. i.e. a bipartite state is a product state if there exist states ψ_1 and ψ_2 such that:



(3.17)

When a bipartite state is *not* a product state, then it is an *entangled state*.

In the category **FHilb**:

- joint states of H and K are elements of $H \otimes K$;
- product states are factorisable states;
- entangled states are elements of $H \otimes K$ that cannot be factorised.

3.1.2 Dual Objects

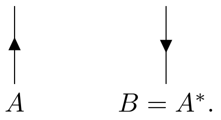
In a monoidal category, an exact pairing between two objects A and B is given by a unit (cup) morphism $\eta : I \to B \otimes A$ and a counit (cap) morhpism $\epsilon : A \otimes B \to I$, such that the following two diagrams commute:

$$\begin{tikzcd}
A \arrow[r, "\rho_A^{-1}"] \arrow[d, "\mathrm{id}_A"] & A \otimes I \arrow[r, "\mathrm{id}_A \otimes \eta"] & A \otimes (B \otimes A) \arrow[d, "\alpha^{-1}_{A,B,A}"] \\
A & I \otimes A \arrow[l, "\lambda_A"] & (A \otimes B) \otimes A \arrow[l, "\epsilon \otimes \mathrm{id}_A"]
\end{tikzcd}
\qquad
\begin{tikzcd}
B \arrow[r, "\lambda_B^{-1}"] \arrow[d, "\mathrm{id}_B"] & I \otimes B \arrow[r, "\eta \otimes \mathrm{id}_B"] & (B \otimes A) \otimes B \arrow[d, "\alpha_{A,B,A}"] \\
B & B \otimes I \arrow[l, "\rho_B"] & B \otimes (A \otimes B) \arrow[l, "\mathrm{id}_B \otimes \epsilon"]
\end{tikzcd}$$

(Heunen and Vicary, 2019).

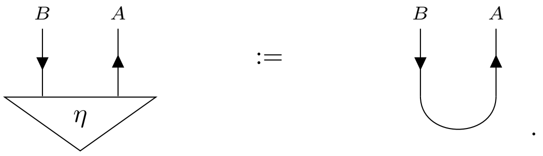
In such an exact pairing, B is called the *right dual* of A, and A is called the *left dual* of B, written $A \dashv B$. When A is both left and right dual to B, A is called a *dual* of B. Duals are unique up to isomorphism.

In the graphical language, left dual objects are depicted as an upward-pointing wire and right dual objects are depicted as a downward-pointing wire.



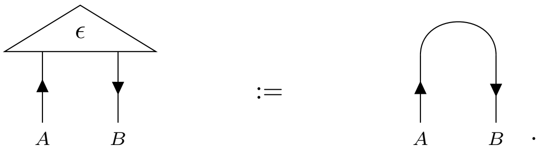
In the graphical language, the unit and counit morphisms have the following depictions:

Cup state $\eta : I \to B \otimes A$



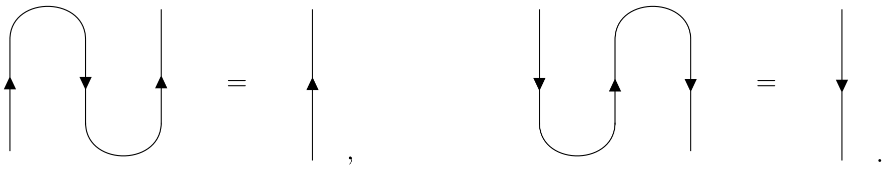
(3.18)

Cap effect $\epsilon : A \otimes B \to I$



(3.19)

Using the notation of 3.18 and 3.19, the dual object commutative diagrams can be rewritten in the graphical language as:

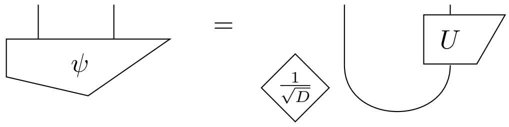


(3.20)

These are known as the yanking equations (see Coecke and Kissinger (2017) Theorem 4.8 for a proof). These equations demonstrate how cups and caps are inverses of each other.

For a two-dimensional Hilbert space, in the category **FHilb**, we can take $A = B = \mathbb{C}^2$. There is, then, no need to distinguish between dual objects with arrows.

Definition 3.9 (Maximally non-separable)**.** A state ψ is *maximally non-separable* (i.e. maximally entangled) if it can be written as:

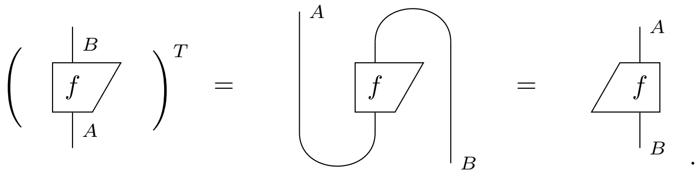


(3.21)

where U is some unitary (as defined in section 3.2.1) and D is the dimension of the input object of U (see equation 3.29 for graphical representations of numbers).

3.1.3 Graphical Manipulations

Definition 3.10 (Transpose)**.** The *transpose* f^T of a process f is another process:



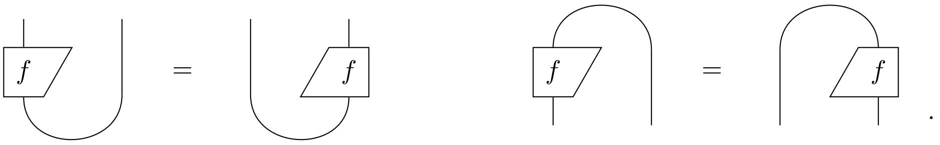
(3.22)

Algebraically, the transpose of f is:

$$f^T = (1_A \otimes \epsilon) \circ (1_A \otimes f \otimes 1_B) \circ (\eta \otimes 1_B). \tag{3.23}$$

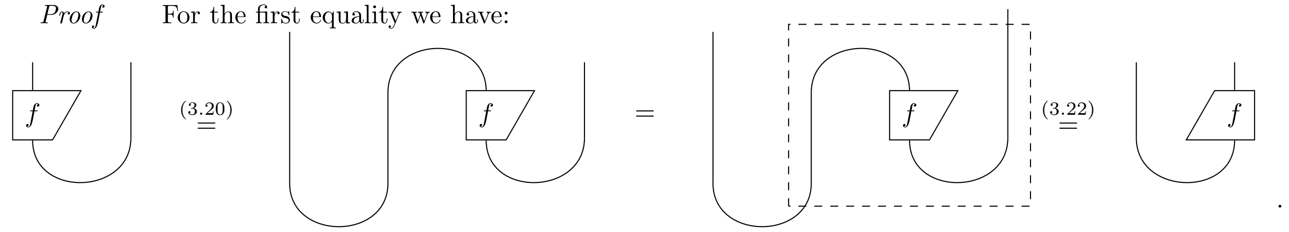
Graphically, the transpose is denoted by a 180° turn of the node. For linear operators, this corresponds to the usual matrix transpose. The transpose is also an involution so $(f^T)^T = f$. This is not surprising since two subsequent rotations of 180° will leave the original process unchanged. This property can be easily verified graphically.

In equation 3.22, it looks like the wires can be yanked to flip the box. Equivalently, the box could be threaded through the cap or cup, then the yanking equations could be applied to straighten the wire. This is correct! For any process f:



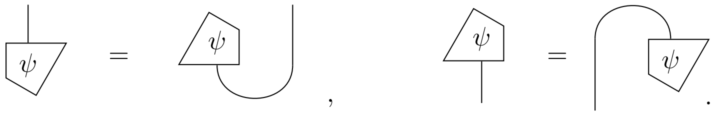
(3.24)

Proof For the first equality we have:



The proof for the second equality proceeds dually (Coecke and Kissinger, 2017).

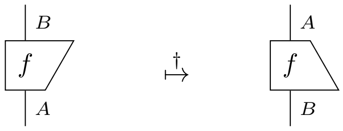
This is also true for states and effects:



(3.25)

Adjoints relate processes from A to B, to processes from B to A.

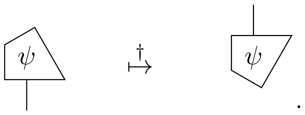
Definition 3.11 (Adjoint)**.** The *adjoint* $f^\dagger$ of a process f is another process:



(3.26)

corresponding to vertical reflection of boxes (Coecke and Kissinger, 2017).

Adjoints bijectively relate states and effects:

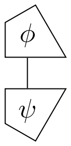


(3.27)

The adjoint is an involution. This is suggested by the graphical notation as applying two subsequent vertical reflections has no impact on a box.

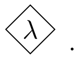
The adjoint preserves parallel composition and identities. The adjoint sends cups to caps and vice versa. The adjoint also reverses sequential composition.

For linear operators, the adjoint corresponds to the usual matrix adjoint. The adjoint of $|\psi\rangle$ is $\langle\psi|$ and vice versa. The following notation for the inner product $\langle\phi|\psi\rangle$ of two states $I \xrightarrow{\psi,\phi} H$ is then obtained:

$\langle\phi|\psi\rangle =$ 

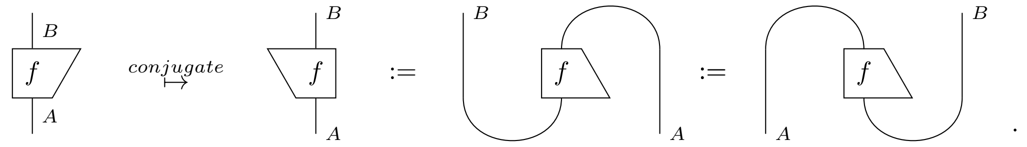
(3.28)

which produces a *number*. A number is a process with no inputs *and* no outputs i.e. a mapping $I \to I$ which can be represented graphically as:



(3.29)

Definition 3.12 (Conjugate)**.** The *conjugate* of a process f is the transpose of its adjoint or, equivalently, the conjugate of its transpose:



(3.30)

Graphically, the conjugate corresponds to a horizontal reflection (equivalent to rotating by 180° to conjugate and vertically reflecting to adjoint, in either order). Similarly to adjoints and transposes, the conjugate is an involution.

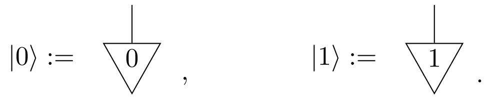
For processes with single input and output wires, the diagrammatic and algebraic notions of conjugates coincide. For processes with multiple input and/or outputs, horizontal reflection reverses the order of the wires. The wires can be twisted to retain the order of wires consistent with the algebraic conjugate.

3.2 Quantum Processes and Doubling

Qubits are the elementary units of information in quantum computing. A qubit is a quantum system with a state space $\mathbb{C}^2$. Qubits are described by two mutually orthogonal basis states defined as:

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, \quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}. \tag{3.31}$$

Or, in the graphical language:



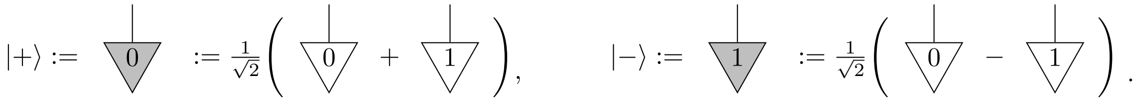
(3.32)

This is the *Z-basis* or *computational basis.*

The *X-basis* is another important basis for $\mathbb{C}^2$. The X-basis is described by the orthonormal states:

$$|+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right), \quad |-\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right). \tag{3.33}$$

In the graphical language, the X-basis is denoted by shaded nodes:



(3.34)

Qubits can occupy any state on the *Bloch sphere* as a normalised superposition of basis states. For example, in the Z-basis:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \qquad \text{with} \qquad |\alpha|^2 + |\beta|^2 = 1. \tag{3.35}$$

This is known as a *pure state* or simply a *state.*

Equivalently, $|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + \sin\left(\frac{\theta}{2}\right)e^{i\phi}|1\rangle$ (see Appendix C for derivation) which is visualised on the Bloch sphere in Figure 1.

From this point, all single-wire objects are assumed to be the Hilbert space $\mathbb{C}^2$ unless stated otherwise.

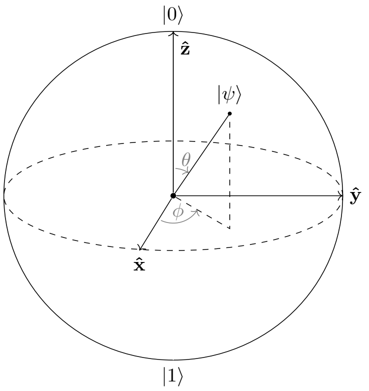


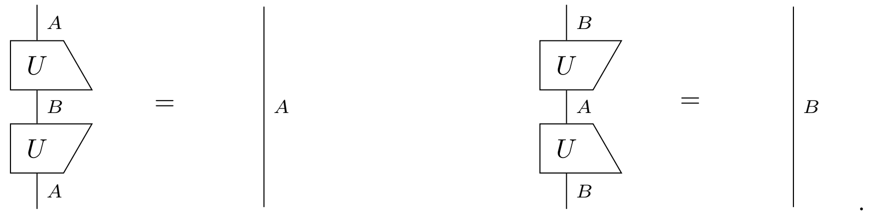
Figure 1: The Bloch sphere.

3.2.1 Unitary Matrices

Transformations of one state vector into another are performed by linear operators acting on the Hilbert space. These operators are represented by the action of matrices. Unitary matrices are particularly important as they act as rotations of states around the Bloch sphere, preserving normalisation and probability amplitudes. In quantum game theory, for example, any and all manipulations on qubits are performed by unitary matrices (Meyer, 1999). Unitary matrices, U, satisfy

$$U^\dagger U = UU^\dagger = I. \tag{3.36}$$

Graphically, equation 3.36 becomes:



(3.37)

That is, the adjoint of a unitary matrix is equal to its inverse i.e.

$$U^{-1} = U^\dagger. \tag{3.38}$$

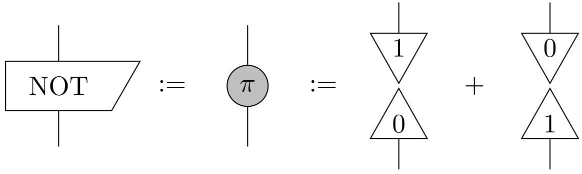
Unitary matrices also preserve normalisation when acting on a normalised state.

Unitary transformations function as gates do in classical computing. Not all logic gates yield unitary linear maps (e.g. the AND gate) so this section restricts to gates that do yield unitary linear maps.

The simplest example of a unitary operation is the identity matrix which preserves a given state:

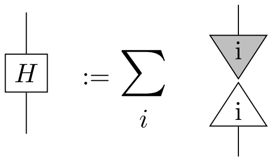
$$\text{i.e. } I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.39}$$

The NOT operation, which switches a pure state $|1\rangle$ to $|0\rangle$ and vice versa, is performed by the X Pauli matrix. When applied to a qubit, it rotates the state around the X-axis of the Bloch sphere by π radians.



$$\text{i.e. } \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{3.40}$$

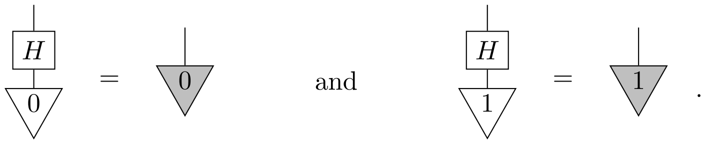
Definition 3.13 (Hadamard linear map)**.** The *Hadamard linear map* is:



$$\text{i.e. } H = \tfrac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ in 2-dimensional Hilbert space.} \tag{3.41}$$

The Hadamard map H is drawn as a symmetrical box in the graphical language as it is self-conjugate and self-adjoint, and hence also self-transposed.

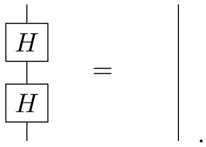
The Hadamard matrix is important as it transforms the qubit-basis states $|0\rangle$ and $|1\rangle$ into a superposition state with equal weight of the basis states $|0\rangle$ and $|1\rangle$. This can be seen in the effect of applying the Hadamard map to the basis states:



(3.42)

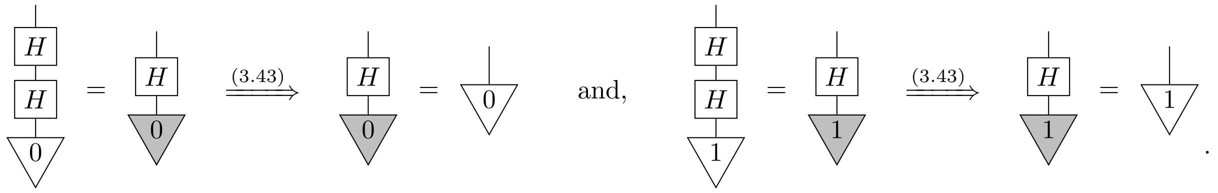
These results can be easily verified algebraically.

The Hadamard linear map is also self-inverse:



(3.43)

So, by applying H to equations 3.42a and 3.42b, respectively, we obtain:



(3.44)

The Hadamard map, then, has the property of changing the colour of state nodes.

3.2.2 Doubling

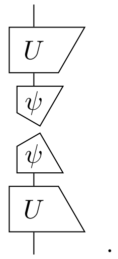
Perhaps one of the most syntactically appealing features of the string diagrams is action of unitaries on density matrices. A pure state $|\psi\rangle$ can be expressed as a density matrix:

$$\rho_{pure} = |\psi\rangle \langle\psi| \,. \tag{3.45}$$

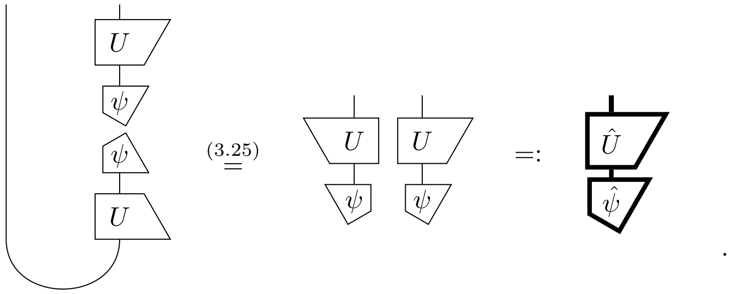
A density matrix, ρ, represents a pure state if and only if $\mathrm{Tr}[\rho^2] = 1$. Commonly, one would algebraically write

$$U |\psi\rangle \langle\psi| U^\dagger$$

to denote the action of a unitary transform on a state ψ described by a density matrix. Graphically, this would be written as:



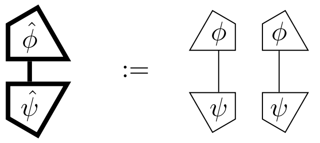
This does not make much sense from a diagrammatic point of view as it certainly cannot be read chronologically from bottom to top. The state ψ, represented by a density matrix, doesn't really "look" like a state anymore as the notation implies there should be some input object. To fix this, the effect $\langle\psi| U^\dagger$ can be transposed into its conjugate. Doing this retains the same data. We then obtain:



(3.46)

In other words, instead of a morphism $H \to H$, we can equivalently work with $I \to H^* \otimes H$ with $H \dashv H^*$. This notation is much more appropriate as this process can be read from bottom to top as first preparing the state $\hat{\psi}$, then acting with process $\hat{U}$. For conciseness, *doubling* notation is defined in equation 3.46. Each doubled component is defined to be the tensor product of each component with its conjugate (see Coecke and Kissinger (2017) Chapter 6 for a precise notion of doubling). Doubled wires represent the object $\mathbb{C}^2 \otimes \mathbb{C}^2$. Crucially, **doubling preserves all string diagrams** including notions of composition and transposition (see Coecke and Kissinger (2017) Corollary 6.11).

Doubling turns out to be extremely important for quantum process-theory. Firstly, a doubled state acted on by a doubled effect yields the familiar Born rule:



$$= (\langle\phi|\psi\rangle)^* \otimes \langle\phi|\psi\rangle = |\langle\phi|\psi\rangle|^2 \tag{3.47}$$

which is a real number $\in [0,1]$ indicating the probability of measuring state ϕ given the input state ψ. Note that equation 3.47 takes the tensor product of two numbers which is a product in the usual sense.

Doubled states also give rise to a diagrammatic representation for *mixed states.* Statistically mixed quantum states arise when there is limited information about a system. For a two-dimensional case, a system may be in the state ψ_1 with probability p and in the state ψ_2 with probability $1-p$.

A system in a mixed quantum state is described by the density matrix:

$$\rho = \sum_n p_n \left|\psi_n\right\rangle \left\langle\psi_n\right| \quad \text{where} \quad \sum_n p_n = 1 \tag{3.48}$$

where $\{\psi_n\}$ is some set of pure states as defined in (3.35), not necessarily orthogonal. A density matrix, ρ, represents a mixed state if and only if $\mathrm{Tr}[\rho^2] < 1$.

While pure quantum states are represented by points on the Bloch sphere, mixed states are visualised as a point *inside* the Bloch sphere (see Appendix D). The Bloch sphere then becomes the *Bloch ball.*

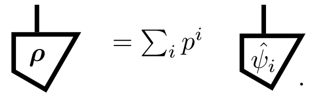
A mixed state prepared with 50% probability of $|0\rangle$ and 50% probability of $|1\rangle$, for example, is described by the density matrix

$$\rho = \frac{1}{2}\left|0\right\rangle\left\langle 0\right| + \frac{1}{2}\left|1\right\rangle\left\langle 1\right| = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}. \tag{3.49}$$

In the Bloch ball, this state would be positioned halfway between $|0\rangle$ and $|1\rangle$ in the centre of the ball. This is an example of a *maximally mixed state.* For a finite-dimensional Hilbert space H, the maximally mixed state is the density matrix:

$$\frac{1}{\dim(H)} \cdot \mathrm{id}_H. \tag{3.50}$$

Since string diagrams use doubled notation (tensor products of conjugates) to represent quantum states, rather than density matrices, mixed quantum states can, therefore, diagrammatically be written as:



(3.51)

Since the sum of the probabilities of each pure state sums to 1 (i.e. $\sum_i p^i = 1$), this known as a *causal* mixed state. The pure states that make up a causal quantum state can always be chosen to form an orthonormal basis (Coecke and Kissinger, 2017).

Table 1 summarises the graphical depictions of both classical and quantum, pure and mixed states. Table 4 (see Appendix D) depicts each of these states on the Bloch ball.

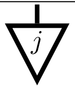
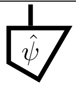
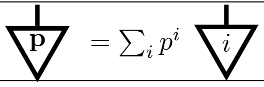
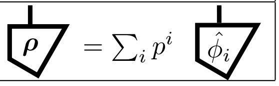
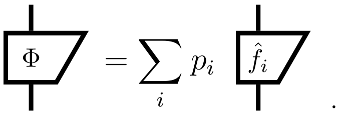
	Classical	Quantum
Pure		
(Causal) Mixed		

Table 1: Classical and quantum, pure and mixed states represented graphically.

3.2.3 Mixed Quantum Maps

Quantum processes can also be mixed. This represents a situation where different pure maps occur with different classical probabilities. There is a lack of knowledge about which pure process is happening, represented by a probability distribution. Any quantum map, Φ, can be written as a sum of pure quantum maps:



(3.52)

This is known as a *Kraus decomposition* (Coecke and Kissinger, 2017).

4 Applications in Quantum Game Theory

So far, quantum processes have been discussed in an abstract sense. Quantum games are an example of a quantum process and this section aims to investigate how string diagrams for different quantum games can be constructed and manipulated.

Quantum game theory aims to construct quantum analogues of classical strategy game models. Namely, quantum game theory imbues games with three main properties:

- Superposition of initial states,
- Entanglement of initial states,
- Superposition of strategies applied on the initial states.

In a given game, each player aims to use their allowed strategies to maximise their *payoff function*. Player i's payoff function, u_i, is a real-valued function which describes the reward given to the player at the outcome of the game and is a function of the game's outcome. Typically, a positive payoff value is associated with a win and a negative value is associated with a loss. A player's *expected payoff* $\bar{u}_i$ for a game can also be calculated. Expected payoff is the sum of all payoff possibilities weighted by the probability that each payoff will occur. Expected payoff indicates the fairness of a game. For two player games, if each player's expected payoff is equal, then the game is fair.

Definition 4.1 (Nash equilibrium)**.** Let S_i be the set of all possible strategies for player i, where $i = 1, 2, ..., N$ for an N player game. Let $s^* = (s_i^*, s_{-i}^*)$ be a strategy profile, a set consisting of one strategy for each player, where s_{-i}^* denotes the $N-1$ strategies of all the players except i. Let $u_i(s_i^*, s_{-i}^*)$ be player i's payoff as a function of the strategies. The strategy profile s^* is a Nash equilibrium if

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*) \quad \forall \quad s_i \in S_i. \tag{4.1}$$

The Nash equilibrium of a 2-player game occurs when player 1 does not gain anything from changing their initial strategy, assuming player 2 also leaves their initial strategy unchanged. Nash equilibrium points are strategically stable as it is in neither player's interest to deviate unilaterally from their strategy without decreasing their payoff.

Allowing quantum properties can drastically change the mechanics of a game such as the expected payoffs and Nash equilibrium points.

In any given game (classical or quantum), players may choose to play *pure strategies* (deterministic) or *mixed strategies* (probabilistic). Pure strategies correspond to playing one single move on a player's turn. Mixed strategies correspond to playing some convex linear combination of allowed moves. This describes playing different moves with different probabilities. Mixed strategies are useful when considering how a player might maximise their expected payoff over a number of games. This makes string diagrams an appropriate language in which to model quantum games as they are equipped to represent both pure and mixed quantum (and classical) states and maps.

4.1 Quantum Penny Flip

In 1999, Meyer created the first quantum game: the quantum penny flip. Classically, the penny flip game occurs as follows:

1. The referee places a penny heads-up in a box. The players, Alice and Bob, know that the penny is initially heads-up, but they cannot see the state of the penny again until it is revealed by the referee at the end of the game.
2. On Alice's turn, she can either flip the penny (X) or do nothing (I).
3. On Bob's turn, he can either flip the penny (X) or do nothing (I).
4. Alice has the final turn. She can either flip the penny (X) or do nothing (I).
5. The penny is revealed. If the penny is heads-up, Alice wins with payoff $u_A = 1$ and Bob loses with payoff $u_B = -1$. If the penny is tails-up, Bob wins with payoff $u_B = 1$ and Alice loses with payoff $u_A = -1$.

The classical penny flip is an example of a *zero-sum game.* A zero-sum game is where one player can only "win" when the other "loses" i.e. the sum for any given game is zero with $u_A + u_B = 0$. Classically, the penny flip is a fair game. There are 8 equally likely outcomes and each player wins in four of them as shown in the payoff matrix in Table 2.

		Alice			
		I, I	I, X	X, I	X, X
Bob	I	$(+1, -1)$	$(-1, +1)$	$(-1, +1)$	$(+1, -1)$
	X	$(-1, +1)$	$(+1, -1)$	$(+1, -1)$	$(-1, +1)$

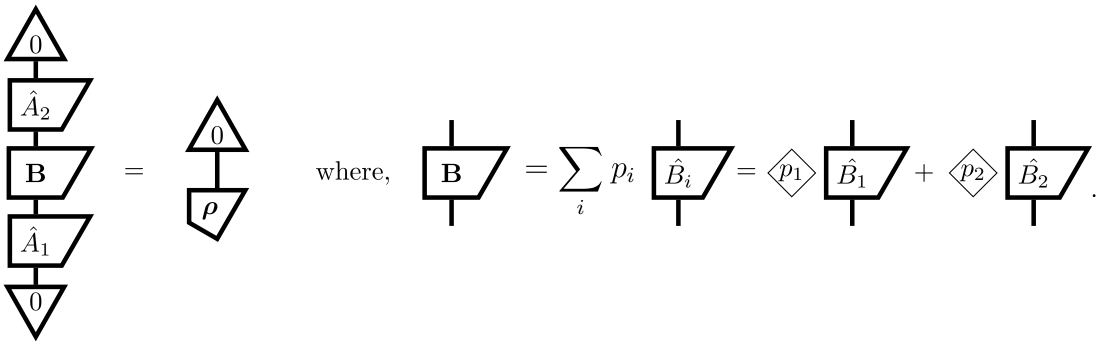
Table 2: Penny flip payoff matrix of classical strategy spaces with payoffs (u_A, u_B).

To quantise this game, Meyer considered what would happen if the penny could be placed into superposition states. Since heads and tails are mutually exclusive outcomes, we can assign $|0\rangle$ to represent heads and $|1\rangle$ to represent tails. The state of the game at any time can be described by:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{4.2}$$

with initial game state $|\psi_{initial}\rangle = |0\rangle$. Alice and Bob manipulate this initial state by applying unitary matrices. The (quantum) penny flip is an example of a *dynamic game*. A dynamic game is a game in which players move one after the other. This corresponds to sequential composition of unitary maps. The move of doing nothing to the penny corresponds to the identity unitary transform (3.39). The move of flipping the penny corresponds to the NOT gate, given by the X Pauli matrix (3.40). Finally, to determine the outcome of the game, the system can be tested for the state heads by applying the effect $\langle 0|$. From this measurement, the payoffs can be determined.

For the following example, we will allow Alice to play any pure quantum move which corresponds to any doubled unitary map. Bob will be restricted to *mixed* classical strategies. This means that Bob may only play the moves X and I, but may play the former with probability p and the latter with probability $1 - p$ for $p \in [0, 1]$. These probabilities are classical information which is encoded into the system by a general quantum map (see 3.52). In this scenario, the quantum penny flip can be modelled using the following string diagram:



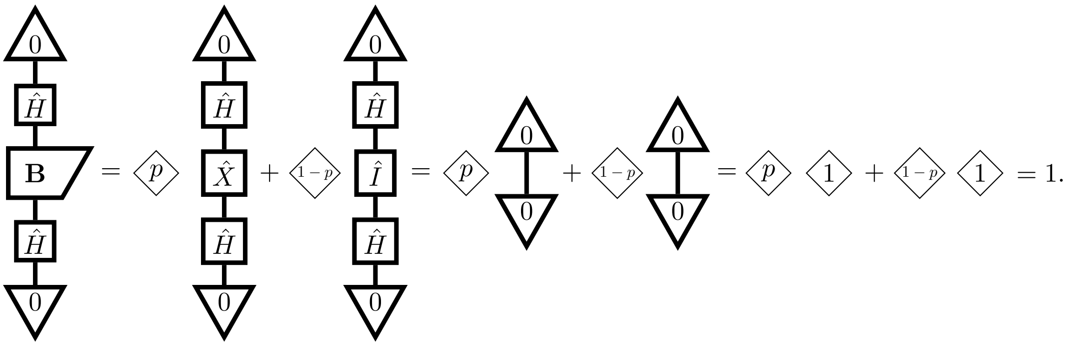
(4.3)

Here, A_1, A_2 are any unitary matrices, $B_1 = X$, $B_2 = I$ and $p_1 = p$, $p_2 = 1 - p$ for $p \in [0, 1]$. ρ is some mixed quantum state, resulting from the actions of Alice and Bob on the initial state, representing the state of the coin immediately prior to measurement. Since doubled states, maps and effects are being used, this string diagram will yield exactly the probability of measuring the coin in the state heads given the chosen moves of Alice and Bob. Using a string diagram to depict the game provides the added benefit of picturing the information flow in a chronological way which would not be possible algebraically.

It is a condition of quantum game theory that the classical counterpart of a given game is contained within the quantum game. This is easily verified as Alice still has the option of playing I and X, yielding the classical game.

Classically, this game does not have a *deterministic Nash equilibrium*. A deterministic Nash equilibrium is a Nash equilibrium arising from pure strategies from each player. The classical penny flip does, however, have a probabilistic Nash equilibrium arising from a pair of mixed strategies. This Nash equilibrium corresponds to Bob flipping the penny with probability 0.5, and Alice playing each of her four strategies with probability 0.25 (Meyer, 1999).

Under the quantum game conditions, a new Nash equilibrium arises (Meyer, 1999) which can be shown in the graphical language. Consider the scenario where Alice plays the Hadamard map H (3.41) on each of her turns. Equation 4.4 depicts this scenario and utilises established diagrammatic rules including the colour change and self-inverse properties of the Hadamard map, as well as invariance of the doubled X-basis states under application of the NOT gate, to evaluate the probability of the coin being measured in the state representing heads (see Appendix E).



(4.4)

In other words, if Alice plays the quantum move H on both of her turns, then the coin will be measured as heads 100% of the time, guaranteeing Alice a win, independent of the likelihood of Bob to flip the coin. The Hadamard map places the coin into an equal superposition of heads and tails, and neither of Bob's classical options can alter this superposition, allowing Alice to freely manipulate the game in her favour.

Suddenly, this game is extremely unfair and the strategy pair $(pX+(1-p)I, HH)$ corresponds to an equilibrium with payoff +1 to Alice and -1 to Bob.

4.2 Quantum Prisoner's Dilemma

The prisoner's dilemma is one of the most famous examples of game theory. It is a nonzero-sum game which means that players can benefit from mutual cooperation. In this game, the two players Alice and Bob, simultaneously choose whether to (C) cooperate with or (D) defect from their opponent. Alice and Bob are not allowed to confer with each other about their decision. Based on their choices, each player will receive a payoff shown in the payoff matrix in Table 3.

		Alice	
		Cooperate	Defect
Bob	Cooperate	$(3,3)$	$(5,0)$
	Defect	$(0,5)$	$(1,1)$

Table 3: Classical prisoner's dilemma payoff matrix with $(u_A,\, u_B)$.

As usual, each player aims to maximise their payoff. While mutual cooperation would benefit both players, cooperating at any point is a risky move as if the other player were to defect, the cooperating player would receive a payoff of 0. As a result, the only classical Nash equilibrium of the prisoner's dilemma is mutual defection (Eisert et al., 1999). Over multiple games, this leaves players worse off than if they were to cooperate with each other.

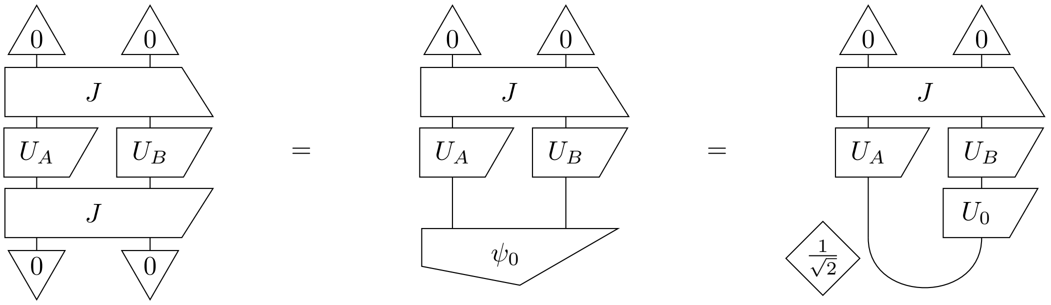
The prisoner's dilemma is an example of a *static* game. A static game is one in which players choose their actions simultaneously. This corresponds to parallel composition of unitary maps.

To formulate this game, cooperation is assigned $|0\rangle$ and defection is assigned $|1\rangle$. Since Alice and Bob play in parallel, the game state at any time is given by a vector in the Hilbert space $\mathbb{C}^2 \otimes \mathbb{C}^2$. The initial state of the game is $|\psi_{initial}\rangle = |0\rangle \otimes |0\rangle = |00\rangle$ which is the tensor product of Alice and Bob's qubits, respectively.

To quantise this game, Eisert et al. created a protocol with an entangling gate and a disentangling gate, as well as allowing Alice and Bob to act on the game state with unitary matrices which can alter the superposition of the game state. For initial separable states, Eisert et al. found that the quantum version of the game did not display any notable features beyond the classical game.

The initial game state is entangled by the operator J with $\mathbb{C}^2 \otimes \mathbb{C}^2 \xrightarrow{J} \mathbb{C}^2 \otimes \mathbb{C}^2$ and where J is symmetric for a fair game. J is known to both players. For notable features of the quantum game (i.e. different Nash equilibria compared to the classical version), J is chosen so that $|\psi_0\rangle = J \circ (|0\rangle \otimes |0\rangle)$ is a maximally entangled state.

Alice and Bob each act on their entangled qubits with some unitary matrix, U_A and U_B respectively. The disentanglement operator, $J^\dagger$, is then applied and measurements are performed on the final states to determine the outcome of the game. As usual, each of Alice's and Bob's states may be in a superposition at the end of the game but will probabilistically collapse into the state $|0\rangle$ or $|1\rangle$ when measured. We can choose to measure for the state $|0\rangle$ corresponding to cooperation. This protocol is shown in the following equivalent string diagrams:



(4.5)

for U_0 some unitary by 3.21. By utilising the properties of string diagrams, it is clear how separable initial states with an entangling gate, can equivalently be interpreted as beginning the game with some non-separable state $|\psi_0\rangle$. Since $|\psi_0\rangle$ is chosen to be maximally non-separable, it can also be written as a cup state acted on by some unitary U_0.

For the quantum prisoner's dilemma, string diagrams provide an intuitive and mathematically precise way to visualise information flow.

In this setup, a new Nash equilibrium does arise for a specific choice of J, U_A and U_B (see Eisert et al. (1999)). However, diagrammatically computing this result, as was done for the penny flip example, would employ further diagrammatic features such as phase gates (see Coecke and Kissinger (2017) Chapter 9). Further work could explore and apply the language rules needed to diagrammatically compute the result of the game under the Nash equilibrium conditions.

5 Discussion and Conclusion

The graphical string diagram language was developed with reference to the monoidal category structure it utilises. Interpretations for the category of finite-dimensional Hilbert spaces were discussed and key features of quantum process theory were established diagrammatically. In particular, the importance of doubling was discussed in order to chronologically describe processes when density matrices are used, and to yield the diagrammatic Born rule.

It was shown that string diagrams are a viable language in which to model and manipulate quantum games through some elementary examples. Features of quantum games such as composition, mixed strategies, initial entanglement and tests have established diagrammatic representations which make string diagrams a highly appropriate way to explore quantum games. Additionally, for doubled diagrams, string diagrams convey the chronological nature of games in a way that algebra does not.

Future directions may include modelling more complex quantum games using string diagrams. The capabilities of diagrammatic reasoning to compute game outcomes under specific choices of initial states and "moves" may also be explored through the use of phase gates and ZX-calculus (see Coecke and Kissinger, 2017).

6 Acknowledgements

I would like to thank Dr Frank Valckenborgh and Shay Tobin for their invaluable ideas and feedback throughout this project. This research project allowed me to explore many different areas of mathematics that interested me and, while a lot of fun for me, meant that my supervisors had to be very patient and adaptable. For this, they have my sincere gratitude.

I would like to thank my family for their continued support of and interest in my research. Their encouragement means the world to me.

References

Coecke, B. and Kissinger, A. (2017), *Picturing Quantum Processes*, Cambridge University Press.

Eisert, J., Wilkens, M. and Lewenstein, M. (1999), 'Quantum games and quantum strategies', *Physical Review Letters* **83**(15), 3077–3080.
URL: *http://dx.doi.org/10.1103/PhysRevLett.83.3077*

Hedges, J., Shprits, E., Winschel, V. and Zahn, P. (2016), 'Compositionality and string diagrams for game theory', *arXiv preprint, arXiv:1604.06061* .
URL: *http://arxiv.org/abs/1604.06061*

Heunen, C. and Vicary, J. (2019), *Categories for Quantum Theory: An Introduction*, Oxford University Press.

Meyer, D. A. (1999), 'Quantum strategies', *Physical Review Letters* **82**, 1052–1055.
URL: *https://link.aps.org/doi/10.1103/PhysRevLett.82.1052*

Selinger, P. (2011), A survey of graphical languages for monoidal categories, *in* B. Coecke, ed., 'New Structures for Physics', Springer Berlin, pp. 289–355.

A Triangle and Pentagon Axioms for Monoidal Categories

For all objects A, B, C, D, the triangle axiom and pentagon axiom are, respectively:

$$\begin{tikzcd}
(A \otimes I) \otimes B \arrow[rr, "\alpha_{A,B,C}"] \arrow[dr, "\rho_A \otimes \mathrm{id}_B"'] & & A \otimes (I \otimes B) \arrow[dl, "\mathrm{id}_B \otimes \lambda_B"] \\
& A \otimes B &
\end{tikzcd} \tag{A.1}$$

$$\begin{tikzcd}
& (A \otimes (B \otimes C)) \otimes D \arrow[rr, "\alpha_{A,B\otimes C,D}"] & & A \otimes ((B \otimes C) \otimes D) \arrow[dr, "\mathrm{id}_A \otimes \alpha_{B,C,D}"] & \\
((A \otimes B) \otimes C) \otimes D \arrow[ur, "\alpha_{A,B,C} \otimes \mathrm{id}_D"] \arrow[drr, "\alpha_{A\otimes B,C,D}"'] & & & & A \otimes (B \otimes (C \otimes D)) \\
& & (A \otimes B) \otimes (C \otimes D). \arrow[urr, "\alpha_{A,B,C\otimes D}"'] & &
\end{tikzcd} \tag{A.2}$$

B Interchange Law and Product Categories

The proof of the interchange law proceeds as follows:

$$\begin{aligned}(g\circ f)\otimes(j\circ h) &= \otimes(g\circ f, j\circ h)\\ &= \otimes((g,j)\circ(f,h)) && \text{(composition in } \mathbf{C}\times\mathbf{C}\text{)}\\ &= (\otimes(g,j))\circ(\otimes(f,h)) && \text{(functoriality of } \otimes\text{)}\\ &= (g\otimes j)\circ(f\otimes h).\end{aligned}$$

□

The category $\mathbf{C}\times\mathbf{C}$ used to define the bifunctor $\otimes$, is a *product category* of $\mathbf{C}$ and $\mathbf{C}$. For categories $\mathbf{C}$ and $\mathbf{D}$, their product category $\mathbf{C}\times\mathbf{D}$ has:

- objects as ordered pairs (A,B) where $A\in\mathrm{Ob}(\mathbf{C})$ and $B\in\mathrm{Ob}(\mathbf{D})$;
- morphisms as ordered pairs $(A,B)\xrightarrow{(f,g)}(C,D)$ with $A\xrightarrow{f}C$ and $B\xrightarrow{g}D$;
- composition as component-wise composition $(A,B)\circ(A',B')=(A\circ A',B\circ B')$ with $A,A'\in\mathrm{Ob}(\mathbf{C})$ and $B,B'\in\mathrm{Ob}(\mathbf{D})$.

C The Bloch Sphere

The Bloch sphere is a geometric representation of the pure states of a two-dimensional quantum system (qubit) on the surface of a unit sphere in $\mathbb{R}^3$. Any pure quantum state can be expressed as a superposition of the Z-basis states:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \qquad \text{with} \qquad |\alpha|^2 + |\beta|^2 = 1$$

where α and β are complex numbers. Expressing α and β in polar form gives:

$$\begin{aligned} |\psi\rangle &= r_\alpha e^{i\phi_\alpha}\,|0\rangle + r_\beta e^{i\phi_\beta}\,|1\rangle \\ &= e^{i\phi_\alpha}(r_\alpha\,|0\rangle + r_\beta e^{i(\phi_\beta - \phi_\alpha)}\,|1\rangle) \end{aligned}$$

for real parameters r_α, ϕ_α, r_β and ϕ_β. The global phase term $e^{i\phi_\alpha}$ has no physical significance as states that differ by a complex number multiple represent the same physical state. Thus, $|\psi\rangle$ can be expressed as:

$$|\psi\rangle = r_\alpha\,|0\rangle + r_\beta e^{i\phi}\,|1\rangle$$

where $\phi = \phi_\beta - \phi_\alpha$ is the relative phase with $0 \leq \phi < 2\pi$. Imposing the normalisation constraint gives:

$$r_\alpha^2 + r_\beta^2 = 1$$

This is the equation of a circle with radius 1. Written in polar coordinates, with $r_\alpha = r\cos\theta'$ and $r_\beta = r\sin\theta'$ this becomes:

$$(r\cos\theta')^2 + (r\sin\theta')^2 = 1 \Rightarrow r = 1 \text{ for } r > 0.$$

So, with $r_\alpha = \cos\theta'$ and $r_\beta = \sin\theta'$, the following expression for $|\psi\rangle$ is obtained:

$$|\psi(\theta', \phi)\rangle = \cos\theta'\,|0\rangle + \sin\theta' e^{i\phi}\,|1\rangle$$

So the state $|\psi\rangle$ is defined fully by two parameters, θ' and ϕ, defining a sphere. Finally, we have that:

$$|\psi(0, \phi)\rangle = |0\rangle$$

and,

$$\left|\psi(\frac{\pi}{2}, \phi)\right\rangle = e^{i\phi}\,|1\rangle$$

which suggests that $0 \leq \theta' \leq \frac{\pi}{2}$ generates the whole sphere. However, since the states $|0\rangle$ and $|1\rangle$ are antipodal in the Bloch sphere representation (i.e. the angle between $|0\rangle$ and $|1\rangle$ must be π), points in the lower hemisphere of the sphere are generated by the mapping:

$$\theta = 2\theta' \Rightarrow \theta' = \frac{\theta}{2}$$

where $0 \leq \theta \leq \pi$. Thus, any pure state ψ can be written as:

$$|\psi(\theta, \phi)\rangle = \cos\frac{\theta}{2}\,|0\rangle + \sin\frac{\theta}{2} e^{i\phi}\,|1\rangle$$

where $0 \leq \theta \leq \pi$, $0 \leq \phi < 2\pi$ are the coordinates of points on the Bloch sphere.

D Pure and Mixed States on the Bloch Ball

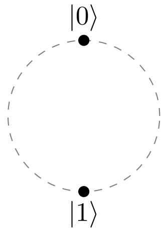
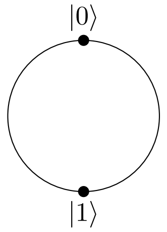
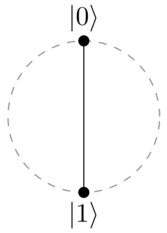
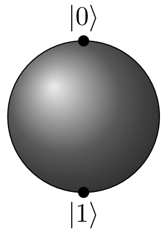
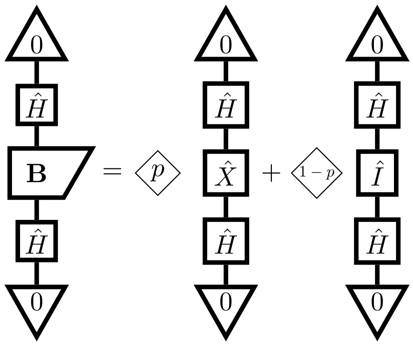
	Classical	Quantum
Pure	 Two points either side of the Bloch sphere	 All points on the surface of Bloch sphere
(Causal) Mixed	 Points on the line connecting the basis states	 All points on and within the Bloch ball

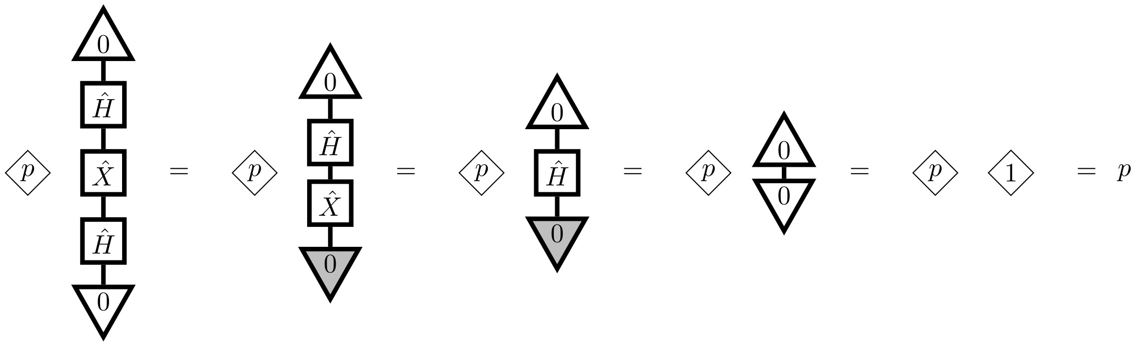
Table 4: 2-dimensional classical and quantum, pure and mixed states visualised on the Bloch ball.

E Quantum Penny Flip Nash Equilibrium

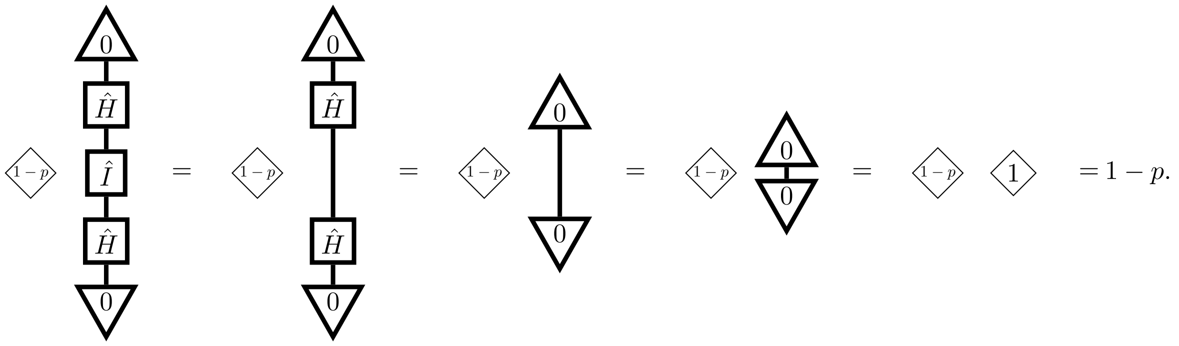
In the event that Alice plays H on both of her moves, the probability of measuring the coin in the state heads can be determined using the established diagrammatic rules:



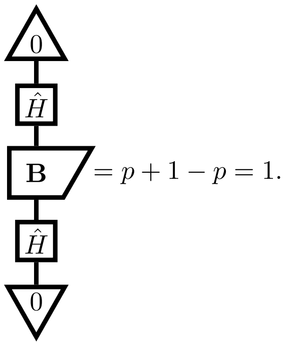
with,



and,



So,


0
$\hat{H}$
B
$\hat{H}$
0
$= p + 1 - p = 1.$