String diagrams and dagger categories

Summary.
(1) $\otimes$-separability and $\cdot$-separability. Duality between processes and states.
(2) String diagrams.
(3) Looking for old friends: transpose, trace, partial trace, adjoint, conjugate, and inner product.
(4) Unitary and positive processes. Projectors.
(5) Expressing quantum phenomena in string diagrams.
(6) Compact closed dagger categories.

Luís Soares Barbosa,
Universidade do Minho (Informatics Department) & INESC TEC

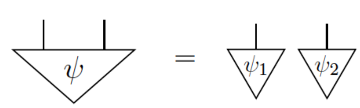
Introduction.

String diagrams are circuits in which inputs (outputs) can be connected to inputs (outputs). Such diagrams can be interpreted by process theories capturing typical characteristics of quantum theory, namely non-separability and the duality between processes and bipartite states. This lecture introduces the extended circuit language and the representation of a number of familiar concepts: transpose, trace, partial trace, adjoint, conjugate, inner product. Next it introduces unitary and positive processes and discusses how several phenomena in quantum theory can be expressed in string diagrams. The lecture ends with a brief introduction to dagger compact categories, the categorical framework which provides a suitable axiomatisation of string diagrams[1].

Separability.

The whole is more than the sum of parts:

A state is $\otimes$-separable if there exist two states into which it can be decomposed in parallel through $\otimes$:

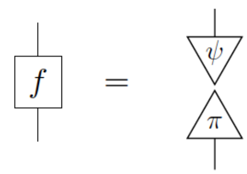


[1]Pictures are taken from Coecke and Kissinger book, *Picturing Quantum processes*, CUP, 2017.

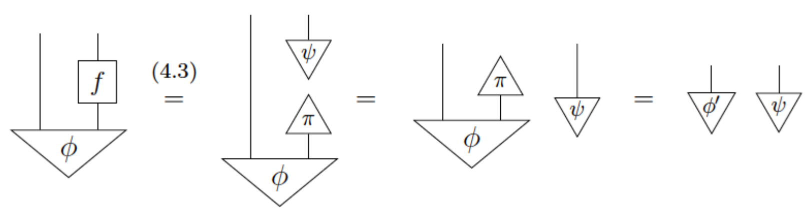
Exercise 1

Determine which states are ⊗-separable in the theory of functions and in the theory of relations.

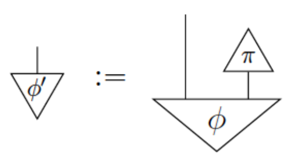
A state is •-separable if there exist two states into which it can be decomposed sequentially through •:



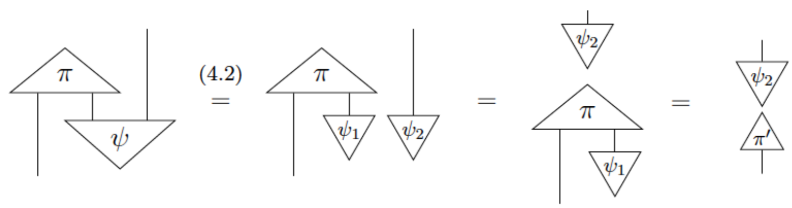
Both notions of separability are related as follows:



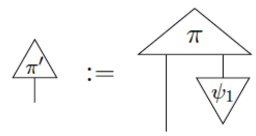
With



and



With

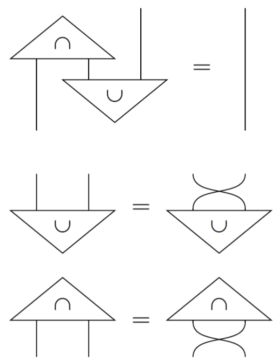


Exercise 2

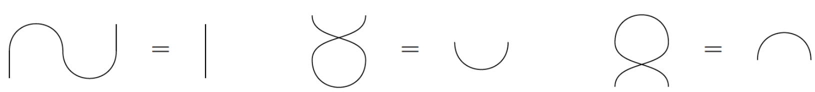
Determine which states are $\cdot$-separable in the theory of functions. Explain in what sense a process theory in which every process is $\cdot$-separable becomes trivial.

String diagrams.

String diagrams are circuits equipped with a special *cup* state $\cup_A$ and a special *cap* effect $\cap_A$, for every type A, such that

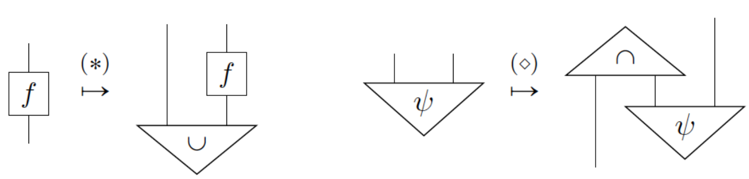


which can be written as



which form the so-called *yanking equations.*

If they hold, i.e. in any string diagram, the following maps, which convert processes into states and back, are mutually inverse



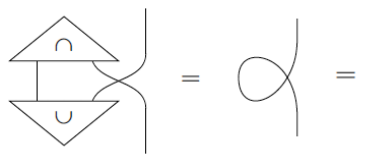
Exercise 3

Verify this statement and its converse.

This means that string diagrams form the language of process theories in which processes and bipartite states are in bijective correspondence. One may also say that a string diagram is a circuit whose inputs (outputs) can be connected to inputs (outputs).

Exercise 4

Show that



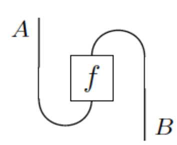
Exercise 5

Which relations correspond to $\cup_{\mathsf{A}}$ and $\cap_{\mathsf{A}}$ in the process theory of relations?

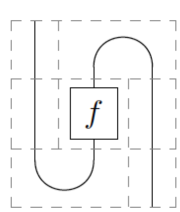
We will introduce now a number of basic notions one got used to in linear algebra and quantum theory formulated in terms of string diagrams. Later they will be suitably interpreted in concrete process theories.

Meeting old friends: transposition

The transpose of a process $\mathsf{f} : \mathsf{A} \longrightarrow \mathsf{B}$ is the process



Clearly (but differently from the *inverse* process) it can be realised by a string diagram involving f:



which corresponds to the following expression

$$\mathsf{f}^{\mathsf{T}} = (\mathtt{id}_{\mathsf{A}} \otimes \cap_{\mathsf{B}}) \cdot (\mathtt{id}_{\mathsf{A}} \otimes \mathsf{f} \otimes \mathtt{id}_{\mathsf{B}}) \cdot (\cup_{\mathsf{A}} \otimes \mathtt{id}_{\mathsf{B}})$$

Exercise 6

Show that

- Transposition is involutive.
- The transpose of a cap is a cup and vice-versa.

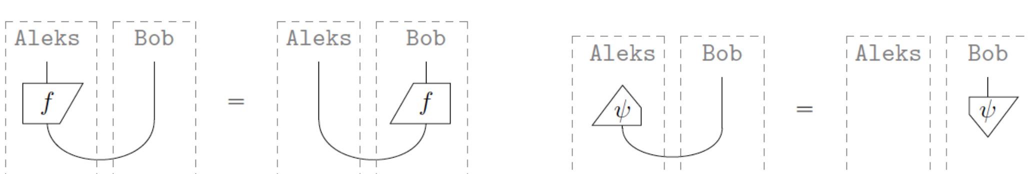
A transpose can be built in the diagram notation as follows, corresponding to a 180 rotation:

Exercise 7

Show that

i.e. boxes can slide along caps and cups.

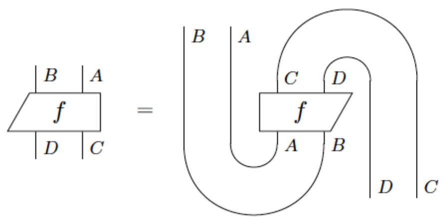
From an operational point of view, transposition can be regarded as a perfect correlation: as soon as a component of the system obtains an effect, the other component will be in the corresponding state:



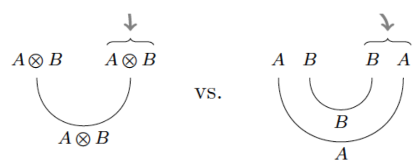
Exercise 8

What is the transpose of a scalar?

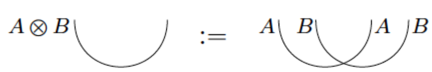
However, to transpose bipartite states requires a special care. Actually,

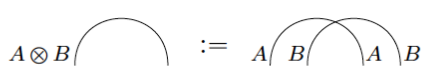


leads to a type mismatch:

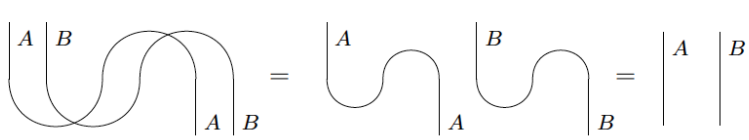


Alternative cross-cup/cap are defined as follows,

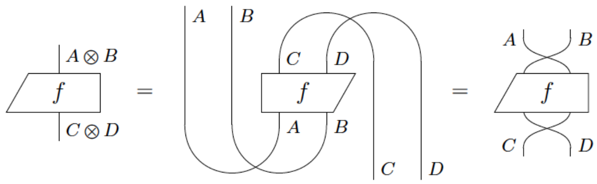


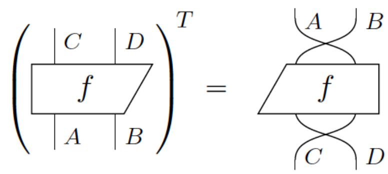


which are well-behaved wrt yanking

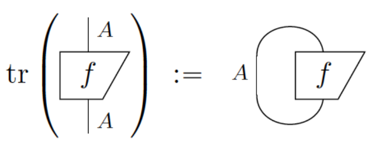


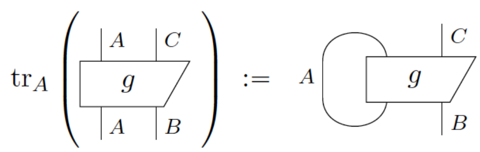
leading to an alternative notion of transposition, called the *algebraic transpose:*





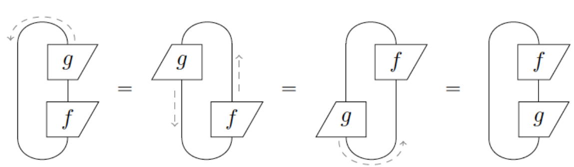
Meeting old friends: trace and partial trace





Exercise 9

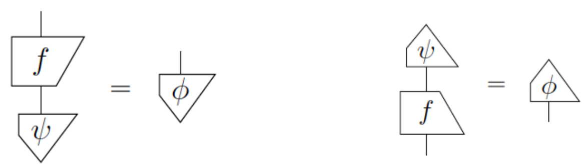
Formulate as an expression the statement of the theorem whose proof is as follows:



Meeting old friends: adjoints

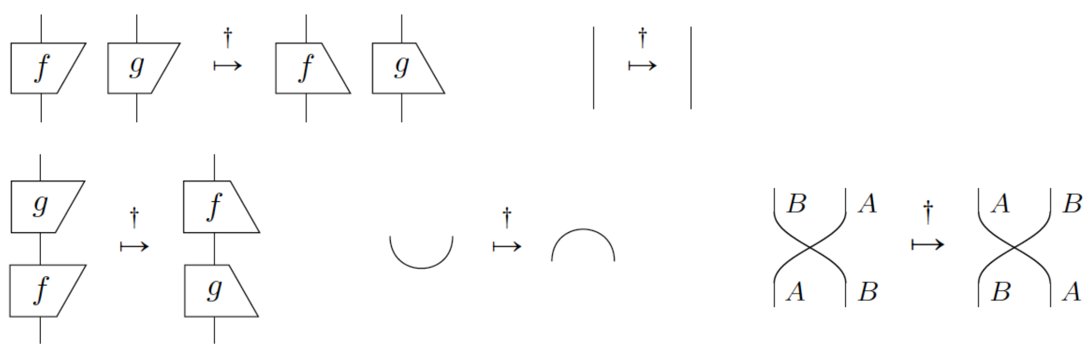
The adjoint of a state is the effect testing for it, which in a string diagram is represented by a vertical reflexion:

Thus the following are equivalent



i.e. if a process transforms a state into another, its adjoint transforms the corresponding effects.

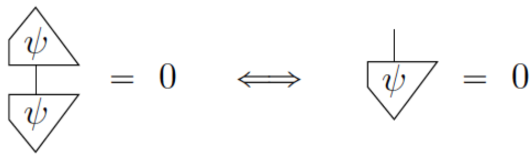
Properties:



Note that the concrete definition of an adjoint depends on the process theory at hands. Clearly, it is expected

- to be involutive
- and to reflect diagrams

It should also be compatible with the intuition that it sends a state to the effect that tests for that state. Formally, it should be definite, i.e.



which means that the only situation in which it is impossible to get an affirmative answer when testing a state for itself is when the original state is itself impossible.

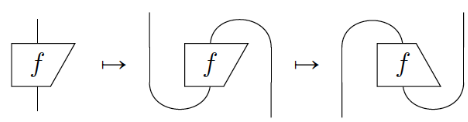
Exercise 10

What are adjoints in the theory of relations? Do we really need the concept of an adjunction in this theory?

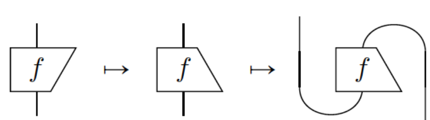
Meeting old friends: conjugates

Conjugates are combinations of adjoints with transposes (by any order), and therefore are expressed in string diagrams by an *horizontal reflection*.

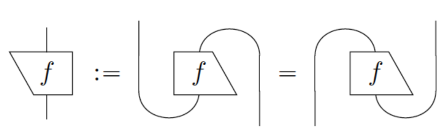
- transpose, then adjoin



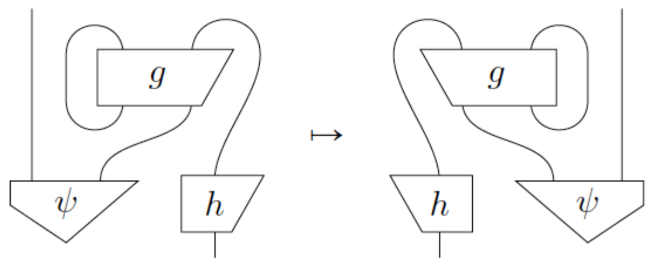
- adjoin, then tranpose



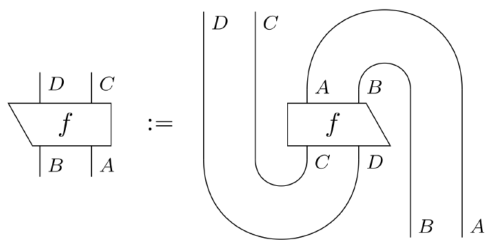
The conjugate of a process is the transpose of its adjoint (or the adjoint of its tranposition), depicted graphically as



As transposes and adjoints, conjugates mirrors entire diagrams in both vertical and horizontal directions:

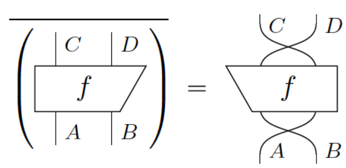


Note that conjugating a process with multiple inputs and outputs is order-reversing

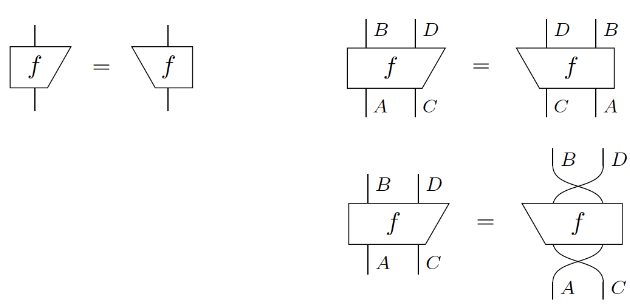


This can be avoided by replacing the transpose by the algebraic transpose, thus defining the *algebraic-conjugate* as follows

$$\overline{\mathsf{f}} := (\mathsf{f}^{\mathsf{T}})^{\dagger} = (\mathsf{f}^{\dagger})^{\mathsf{T}}$$



Processes that are equal to their own (algebraic) conjugates are said to be (algebraic) self-conjugates:



Exercise 11

Show that caps and cups are (algebraic) self-conjugates.

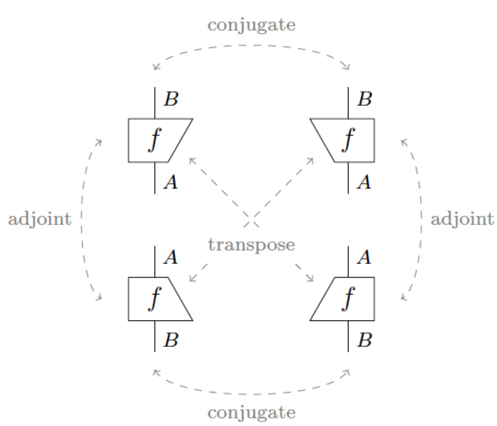
Exercise 12

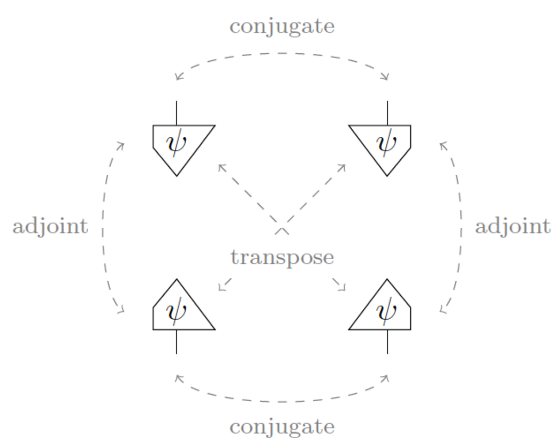
Discuss which relations are (algebraic) self-conjugate.

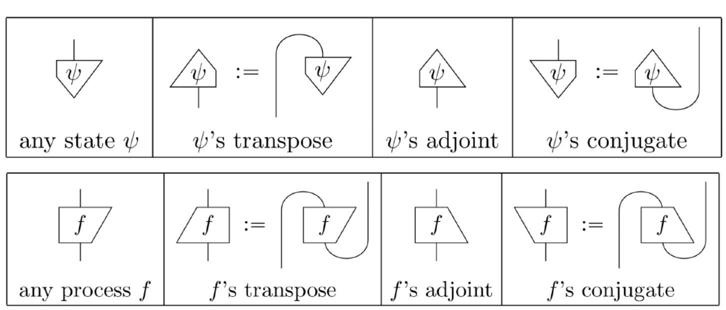
Exercise 13

What is the conjugate of a scalar?

Summing up

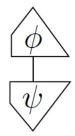






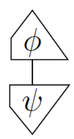
Meeting old friends: inner product

The inner product



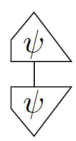
has a clear intuitive meaning: Since the adjoint of a state gives the effect that tests for that state, the inner product expresses *testing state* ψ *for being state* ϕ.

- Orthonormal states:

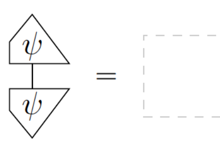


$$= \quad 0$$

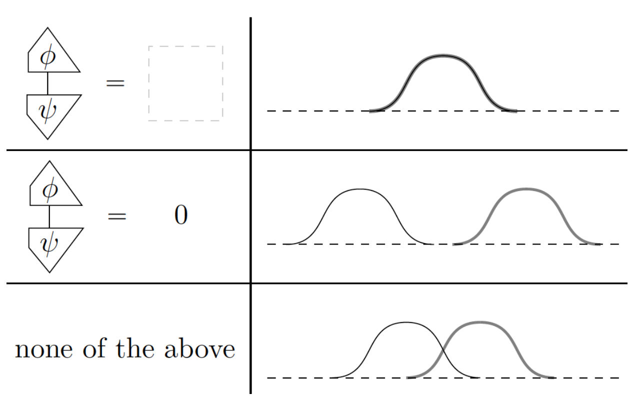
- The squared norm is the inner product with itself:



- Normalised state:



The inner product computes how much similar states are:



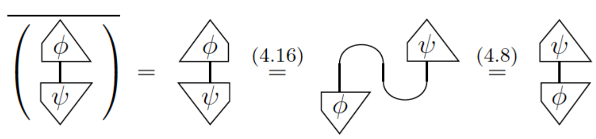
$\boxed{\textbf{Exercise } 14}$

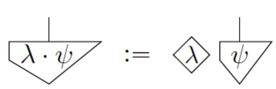
Discuss the following diagrams in the process theory of relations:

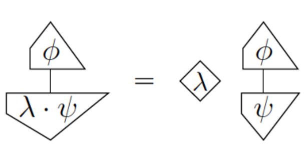
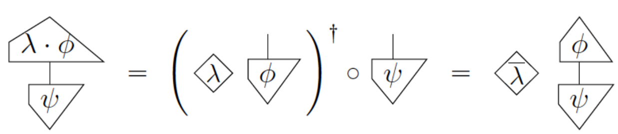
Properties

$$
\begin{aligned}
\overline{\langle\phi|\psi\rangle} &= \langle\psi|\phi\rangle && \text{conjugate symmetric}\\
\langle\phi|\alpha\cdot\psi\rangle &= \alpha\cdot\langle\psi|\phi\rangle && \text{linearity (preserves scalars on the second argument)}\\
\langle\alpha\cdot\phi|\psi\rangle &= \overline{\alpha}\cdot\langle\psi|\phi\rangle && \text{conjugate linearity (conjugates scalars on the first argument)}\\
\langle\phi|\phi\rangle = 0 \;&\Leftrightarrow\; |\phi\rangle = 0 && \text{positive definite}
\end{aligned}
$$

Diagrammatic proofs:

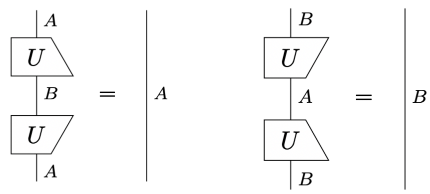


Making 

 and 

Unitary processes.

A process $\mathsf{U} : \mathsf{A} \longrightarrow \mathsf{B}$ is unitary if $\mathsf{U}^\dagger$ is its inverse, i.e. $\mathsf{U}^\dagger \cdot \mathsf{U} = \mathsf{id}_\mathsf{A}$ and $\mathsf{U} \cdot \mathsf{U}^\dagger = \mathsf{id}_\mathsf{B}$ Unitary processes are the ones that preserve the measure of commonality given by the inner product.



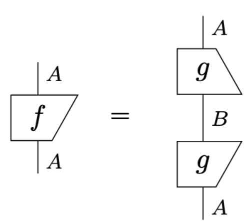
(isometry)

Exercise 15

Show that a unitary U preserves the inner product.

Positive processes.

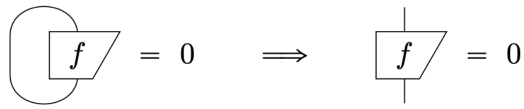
A process $\mathsf{f} : \mathsf{A} \longrightarrow \mathsf{A}$ is positive if there exists another process $\mathsf{g} : \mathsf{A} \longrightarrow \mathsf{B}$ such that $\mathsf{f} = \mathsf{g}^\dagger \cdot \mathsf{g}$, i.e.



The definition entails that positive processes are self-adjoint as they are invariant under vertical reflection. Note that the scalar representing the inner product of a state with itself is positive in this sense, which explains the qualifier *positive* when one requires the inner products to be *positive definite*, i.e. $\langle \phi | \phi \rangle \ = \ 0 \Leftrightarrow |\phi\rangle = 0$.

Exercise 16

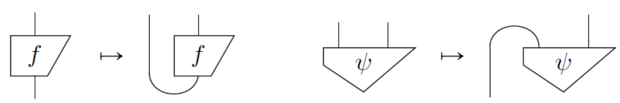
Show that if f is a positive process, $\mathsf{Tr}(\mathsf{f}) = 0 \Rightarrow \mathsf{f} = 0$, i.e.



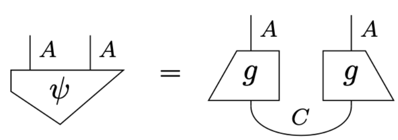
Exercise 17

In linear algebra f is positive if, for every ϕ, the number ⟨ϕ|f|ϕ⟩ is positive. Relate this formulation to the definition just given.

In the previous lecture we have noted that string diagrams express a duality (i.e. a bijective correspondence) between processes and bipartite states, cf.



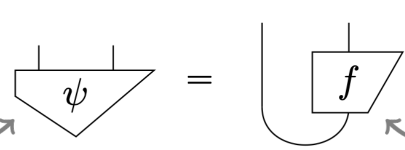
The state corresponding, under such a duality, to a positive process carried itself a positive structure in the horizontal dimension defined as follows: a bipartite state is ⊗-positive if there exists a process g such that



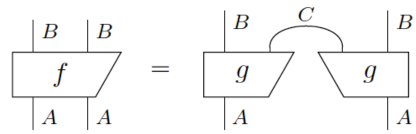
Thus,

Exercise 18

Verify the statement above, depicted as follows

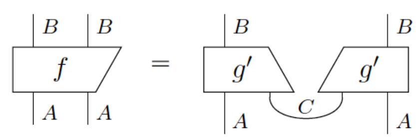


The definition extends to processes: f is $\otimes$-positive if there exists a process g such that



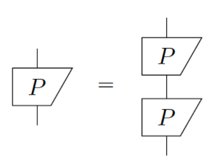
Exercise 19

Show this is equivalent to the existence of a process g' such that



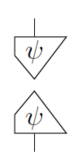
Projectors.

A process P positive and idempotent, i.e. such that



is called a *projector*.

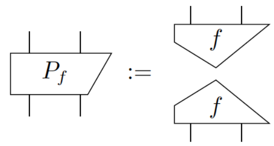
Any normalised state ψ yields a projector $|\psi\rangle\langle\psi|$ depicted as



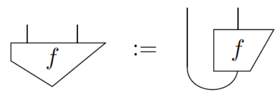
Exercise 20

Show this construction yields a positive and idempotent process.

In general, resorting to the duality between processes and bipartite states, one may define the notion of a *separable* projector as follows: A process $\mathsf{f} : \mathsf{A} \longrightarrow \mathsf{A}$ yields a separable projector via



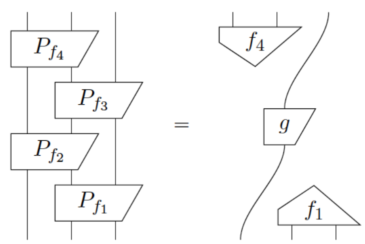
if state



is normalised. Note that a separable projector in linear algebra is exactly one that projects onto a one-dimensional vectorial space.

$\boxed{\textbf{Exercise } 21}$

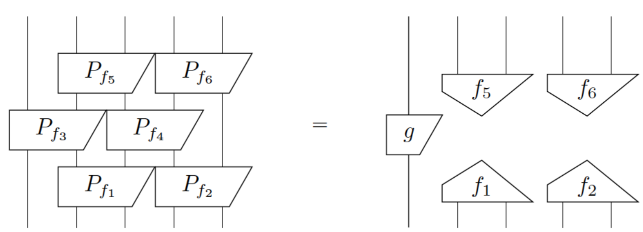
Show that



where $\mathsf{g} = \mathsf{f}_3 \cdot \overline{\mathsf{f}_4} \cdot \mathsf{f}_2^{\mathsf{T}} \cdot \mathsf{f}_3^{\dagger} \cdot \mathsf{f}_1 \cdot \mathsf{f}_1 \cdot \overline{\mathsf{f}_2}$

$\boxed{\textbf{Exercise } 22}$

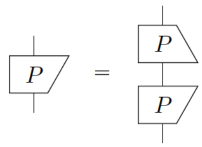
Show that



where $\mathsf{g} = \mathsf{f}_3^{\mathsf{T}} \cdot \mathsf{f}_5^{\dagger} \cdot \mathsf{f}_4^{\mathsf{T}} \cdot \mathsf{f}_6^{\dagger} \cdot \mathsf{f}_2 \cdot \overline{\mathsf{f}_4} \cdot \mathsf{f}_1 \cdot \overline{\mathsf{f}_3}$

$\boxed{\textbf{Exercise } 23}$

Show that one may define a projector, alternatively, as a self-adjoint idempotent or as a process P satisfying



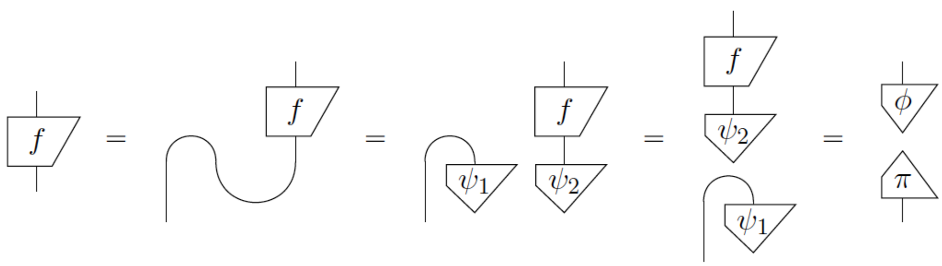
Expressing quantum phenomena in string diagrams.

---

1. Non-separable states exist.

In a theory described by string diagrams, if all bipartite states are $\otimes$-separable, then all processes will be $\cdot$-separable, therefore making the theory trivial.

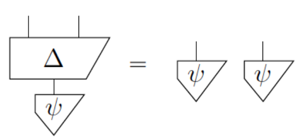
Proof.



for state $\phi = \mathsf{f} \cdot \psi_2$ and effect $\pi = (\psi_1)^{\mathsf{T}}$. The second step assumes that cup is $\otimes$-separable.

2. The non-cloning theorem.

Let us define a *cloning process* Δ as one that makes two copies of its input state[2]

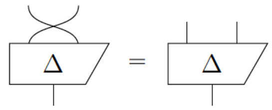


(1)

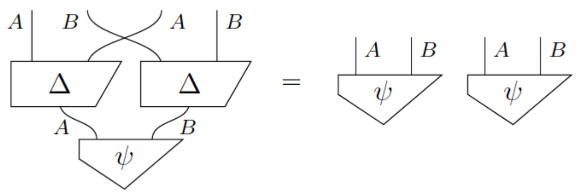
We formulate three reasonable assumptions on such a process:

[2]Note that in quantum information a *cloning process* is usually defined as a two inputs process whose second input gets overwritten by the first one. Our version captures the same phenomenon in a somehow less constrained way.

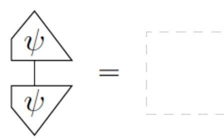
A (swapping does not affect cloning)



B (a composite is cloned by cloning each of its components)

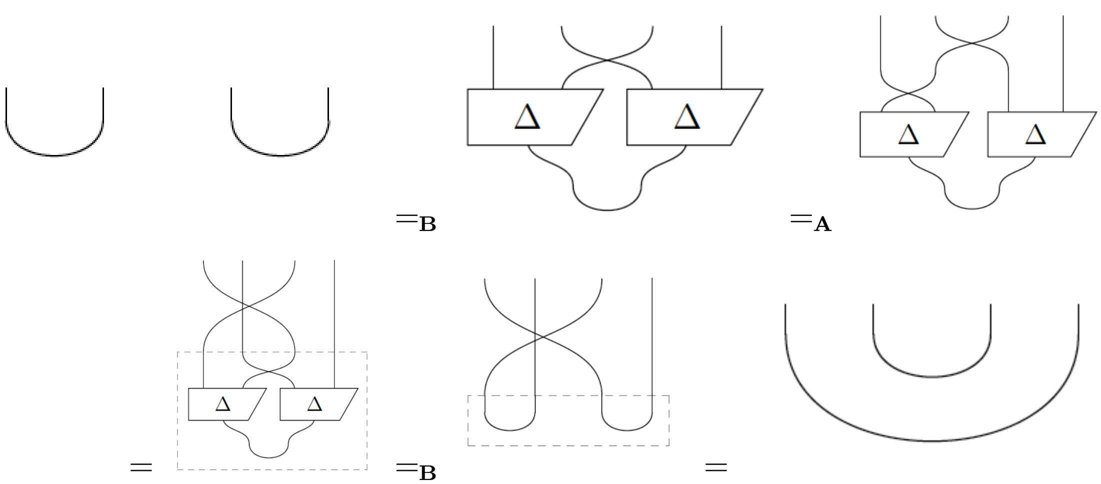


C (the process theory contains at least a normalised state)

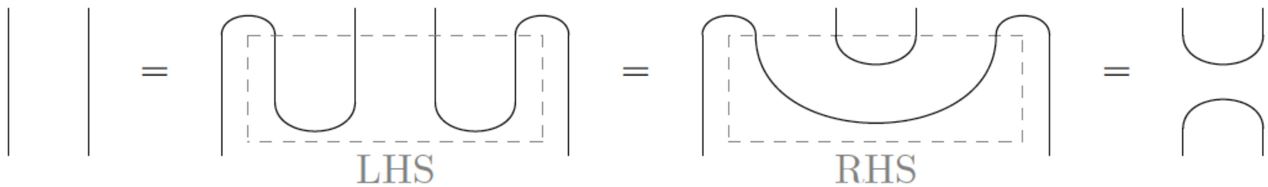


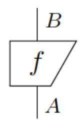
The *no-go* theorem is as follows: If a process theory described by string diagrams contains a cloning process for a type A, then every process with input A must be •-separable.

Proof.

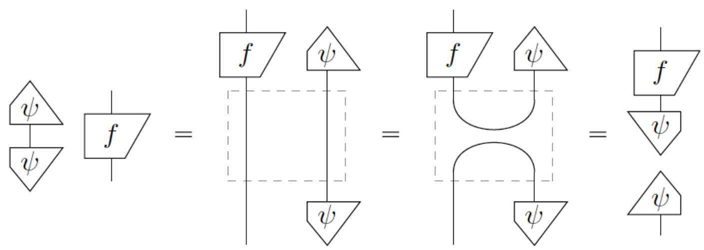


where all wires are of type A. Converting outputs into inputs in both sides of the equation above yields



Then, by assumption **C**, 

equals



The non-cloning theorem is folklore in quantum information. But what happens in the theory of relations? A cloning function is easily realised: $\Delta(a) = (a, a)$. Denoting by $\underline{x} : \mathbf{1} \longrightarrow A$ the constant function that always returns x, equation (1) defining a cloning process instantiates as follows:

$$\Delta(\underline{a}) \;=\; \underline{\Delta(a)} \;=\; (a, a) \;=\; \underline{a} \times \underline{a}$$

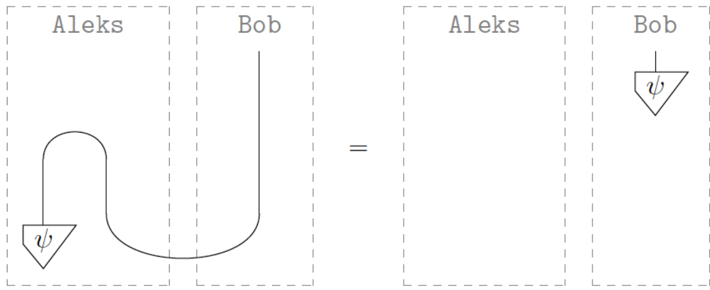
which is obviously true. Consider now a cloning relation $\Delta = \{(a, (a, a)) \mid a \in A\}$. Equation (1) now reads

$$\{(*, (a, a,)) \mid a \in A\} = \{(*, a) \mid a \in A\} \times \{(*, a) \mid a \in A\}$$

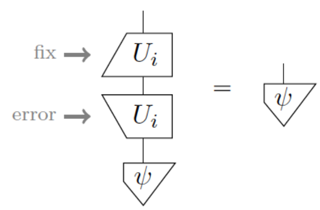
which is no longer true: the right hand side includes pairs $((*, a), (*, a))$ which are in bijective correspondence with pairs $(*, (a, a))$ in the left hand side, but also e.g. $((*, a), (*, b))$ for $a \neq b$. Note that in both process theories $\otimes$ is Cartesian product $\times$, but in the theory of relations this is not a categorical product.

3. A first version of teleportation.

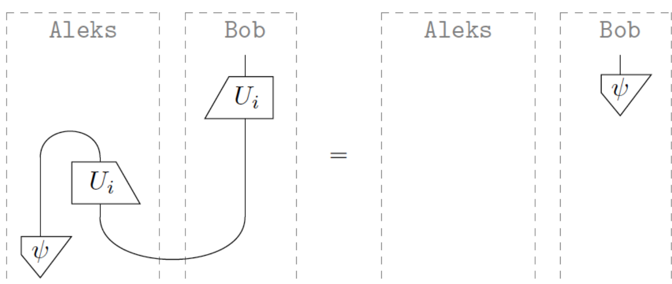
Assume Aleks possesses a state to be transmitted to Bob, with whom he shares a cup state. A solution may be



However, effects arise (to discuss later) as the result of a (quantum) *measurement*; thus Aleks might not get the cap itself, but the cap affected by some non-deterministic error from a given set of possible errors. Then Aleks needs to inform Bob of the error, i.e. to send a single index i so that Bob can choose the right error-corrector. Actually, assuming each U_i to be unitary, one has



leading to



Example: Teleportation in the theory of relations

The shared cup represents a pair of envelops, one for Aleks another for Bob, which inside have either a 0 or a 1. They do not know which bit is it, but they do know the bit is the same in both envelops. Formally, the shared cup represents this fact through the following relation

$$\cup = \{(*, (0,0)), (*, (1,1))\}$$

Aleks informs if the bit stored in his envelope is equal or different of his own bit ψ, which corresponds to the following effects, respectively:

$$\mathsf{M}_0 \;=\; \{((0,0),*),((1,1),*\} \qquad \mathsf{M}_1 \;=\; \{((0,1),*),((1,0),*)\}$$

From this information Bob may conclude if Alexs bit is the one in his own envelop or its complement. The correcting processes are, respectively,

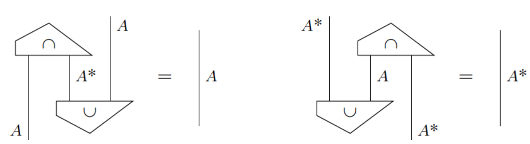
$$\mathsf{U}_0(x) = x \qquad \mathsf{U}_1(x) = 1 - x$$

Int the theory of relations this corresponds to what is known as a *one-time pad encryption*: Aleks sends public data — his bit encrypted by the parity measurement. Bob receives private data (after the right correction). A shared encryption key is used. In quantum teleportation Aleks sends classical data, Bob receives quantum data, using a shared quantum state.

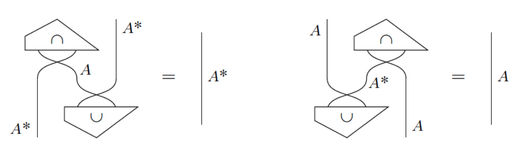
Dual objects.

String diagrams are sound and complete for *dagger compact closed categories*. These categories assume that each type A has a *dual*, A^* to which a cup state and a cap effect

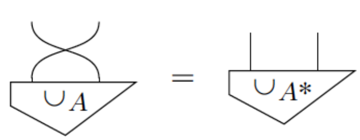
are associated and satisfy



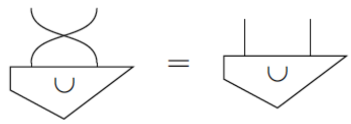
which, just by deformation, also yields



So, $(\mathsf{A}^*)^* \;=\; \mathsf{A}$ and, thus



When types are self-dual, i.e. $A = A^*$, as we have considered before, one gets two ways to define a cup for A, boiling down to the familiar equation



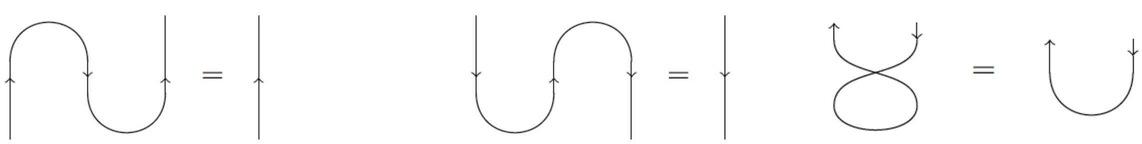
Note that from this more general perspective the typing problem with transposition of nested caps/cups vanishes by making

$$(A \otimes B)^* \ = \ B^* \otimes A^*$$

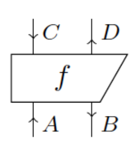
However, the analogy with wires becomes less obvious. The problem is (graphically) overcome through the introduction of a *direction* to the wires:

Thus, caps and cups are once again represented by wires, but directed wires:

And their axioms becomes



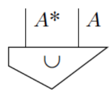
A process $f : A \otimes B^* \longrightarrow C^* \otimes D$ is depicted as

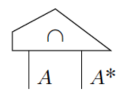


A *directed string diagram* allows any connection between two wires provided that both types and directions are compatible: types must coincide when connecting an input to an output, but should be dual when connecting ports of different polarity.

Example: The theory of linear maps

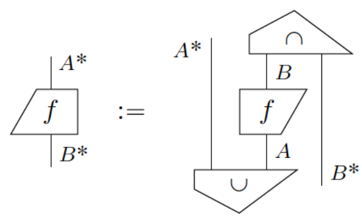
For each finite-dimensional vector space A, its dual A^* is the vector space of linear maps form A to $\mathbb{C}$, where sum and scalar multiplication are defined pointwise[3]. A basis for A^* is also obtained from the basis $\{u_i \mid i \in I\}$ of A as $\{\underline{u}_i \mid i \in I\}$ such that $\underline{u}_i u_j = \delta_{i,j}$. We now define a cap effect and cup state as follows:





$$\sum_i \underline{u}_i \otimes u_i \qquad\qquad t(\nu), \text{ for each } \nu \otimes t \in A \otimes A^*$$

Transposing a process $f : A \longrightarrow B$ with respect to these new caps and cups, yields



which corresponds to pre-composition with f, i.e.

$$f^*(t) \;=\; t \cdot f$$

Dagger compact closed categories.

A symmetric monoidal category C is *compact closed* if for each object A there is another object A^* and arrows

$$\epsilon_A : A \otimes A^* \longrightarrow I \qquad \text{and} \qquad \eta_A : I \longrightarrow A^* \otimes A$$

such that

$$(\epsilon_a \otimes id_A) \cdot (id_A \otimes \eta_A) \;=\; id_A$$
$$(id_{A^*} \otimes \epsilon_A) \cdot (\eta_A \otimes id_{A^*}) \;=\; id_{A^*}$$

A *dagger compact closed category* is a compact closed category C equipped with a *dagger functor* $\dagger : C \longrightarrow C$ such that

$$\epsilon_A^\dagger \;=\; \eta_{A^*}$$

where a *dagger functor* is defined by

$$A^\dagger \;=\; A \quad \text{and} \quad (f : A \longrightarrow B)^\dagger \;=\; f^\dagger : B \longrightarrow A$$

[3] i.e. $(t + s)(\nu) = t(\nu) + s(\nu)$ and $\alpha(t(\nu)) = \alpha t(\nu)$.

and, additionally, is involutive and respects the symmetric monoidal structure, i.e.

$$\begin{aligned} f &= (f^\dagger)^\dagger \\ (g \cdot f)^\dagger &= f^\dagger \cdot g^\dagger \\ (f \otimes g)^\dagger &= f^\dagger \otimes g^\dagger \\ \sigma_{A,B}^\dagger &= \sigma_{B,A} \end{aligned}$$