Lecture 6: String diagrams and dagger categories - I

Summary.
(1) ⊗-separability and ·-separability. Duality between processes and states.
(2) String diagrams.
(3) Looking for old friends: transpose, trace, partial trace, adjoint, conjugate, and inner product.

Luís Soares Barbosa,
Univ. Minho *(Informatics Dep.)* & Inl *(Quantum Software Engineering Group)*

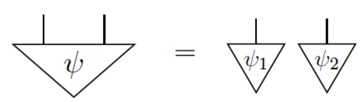
Introduction.

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 first lecture introduces the extended circuit language and the representation of a number of familiar concepts: transpose, trace, partial trace, adjoint, conjugate, inner product. The corresponding categorical framework to provide a suitable axiomatization will be discussed in an accompanying lecture[1].

Separability.

The whole is more than the sum of parts:

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

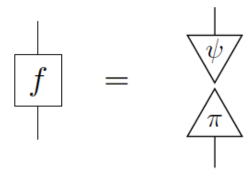


Exercise 1

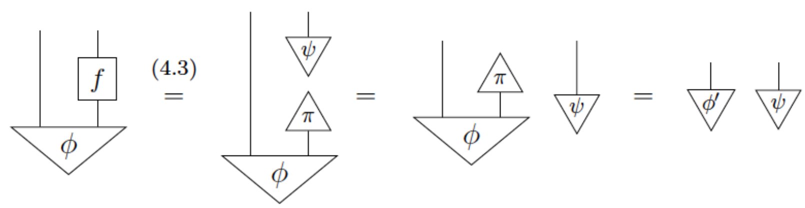
Determine which states are ⊗-separable in the theory of functions and in the theory of relations.

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

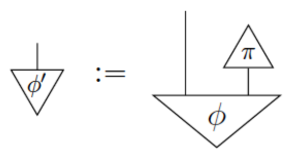
[1] Pictures are taken from Coecke and Kissinger book, *Picturing Quantum processes*, CUP, 2017.



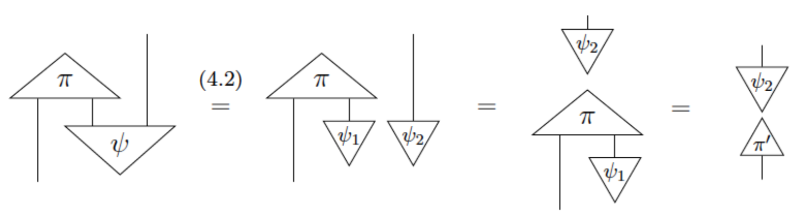
Both notions of separability are related as follows:



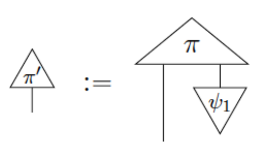
With



and



With

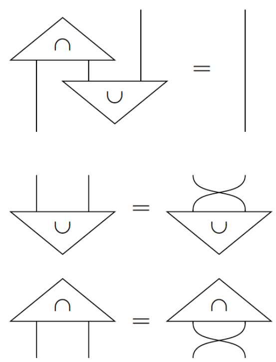


Exercise 2

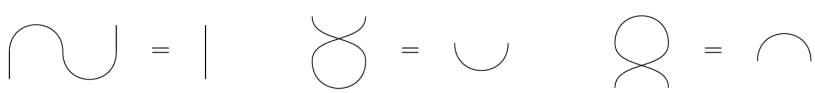
Determine which states are ⋅-separable in the theory of functions. Explain in what sense a process theory in which every process is ⋅-separable becomes trivial.

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

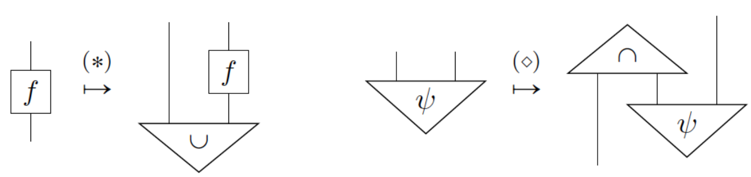


which can be written as



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



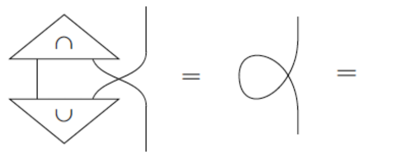
Exercise 3

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



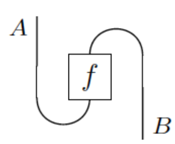
Exercise 5

Which relations correspond to $\cup_A$ and $\cap_A$ in the process theory of relations?

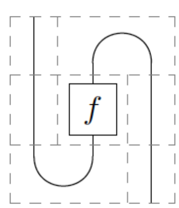
We will introduce now a number of basic notions one got used to in linear algebra and quantum theory formulated in terms of string diagrams. Later they will be suitably interpreted in concrete process theories.

Meeting old friends: transposition

The transpose of a process $f : A \longrightarrow B$ 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

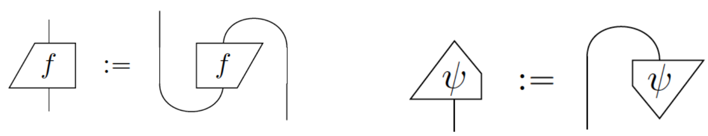
$$f^T = (id_A \otimes \cap_B) \cdot (id_A \otimes f \otimes id_B) \cdot (\cup_A \otimes id_B)$$

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:

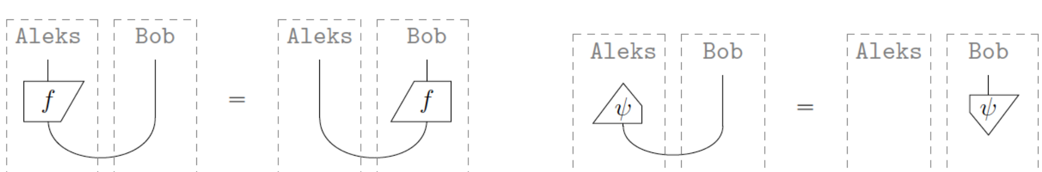


Exercise 7

Show that

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

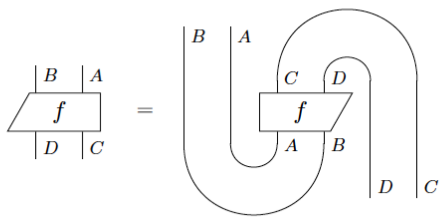
From an operational point of view, transposition can be regarded as a perfect correlation: as soon as a component of the system obtains an effect, the other component will be in the corresponding state:



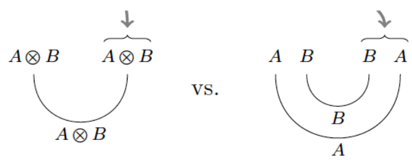
Exercise 8

What is the transpose of a scalar?

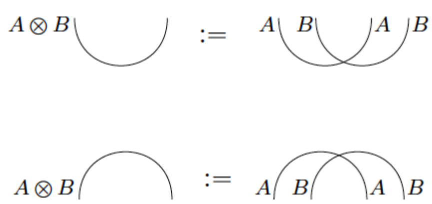
However, to transpose bipartite states requires a special care. Actually,



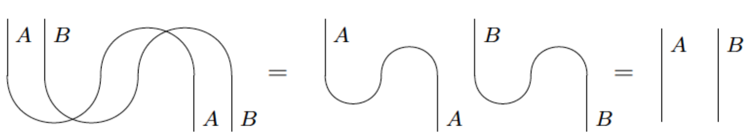
leads to a type mismatch:



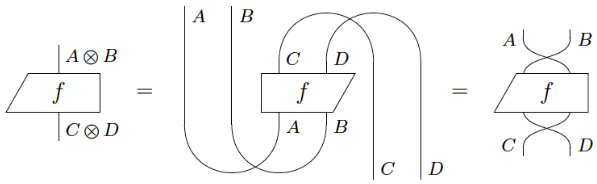
Alternative cross-cup/cap are defined as follows,

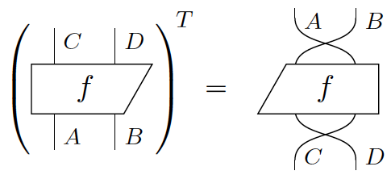


which are well-behaved wrt yanking

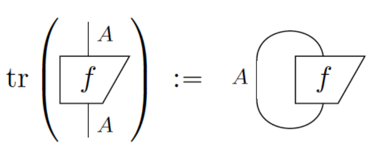


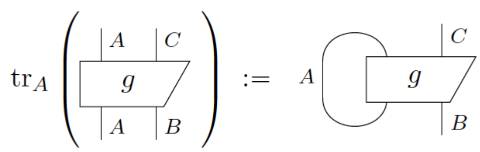
leading to an alternative notion of transposition, called the *algebraic transpose:*





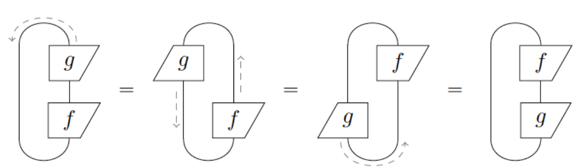
Meeting old friends: trace and partial trace





Exercise 9

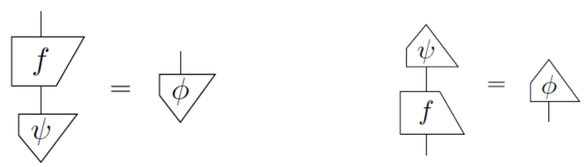
Formulate as an expression the statement of the theorem whose proof is as follows:



Meeting old friends: adjoints

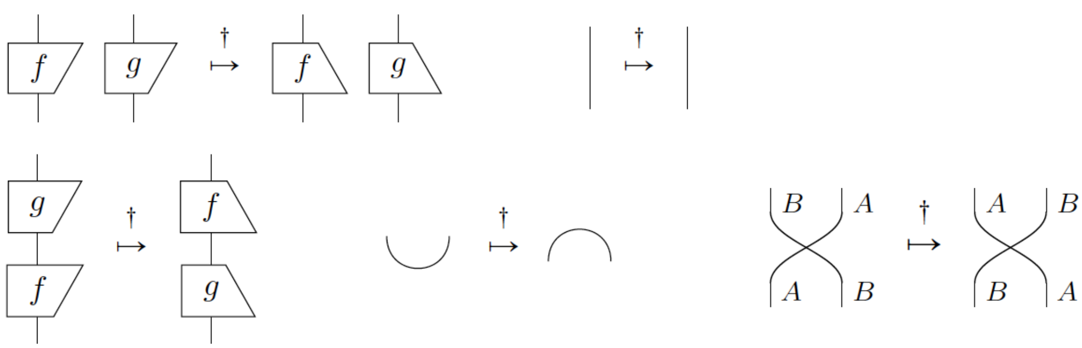
The adjoint of a state is the effect testing for it, which in a string diagram is represented by a vertical reflexion:

Thus the following are equivalent



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

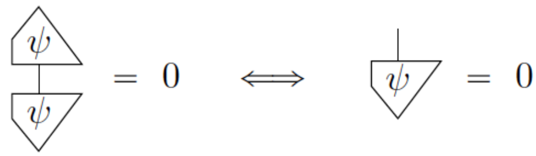
Properties:



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.

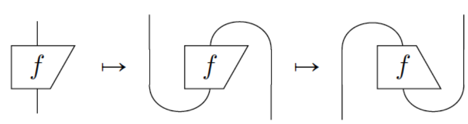
Exercise 10

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

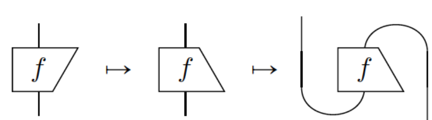
Meeting old friends: conjugates

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

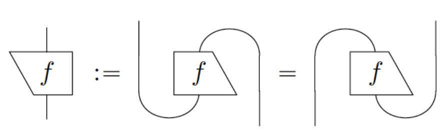
- transpose, then adjoin



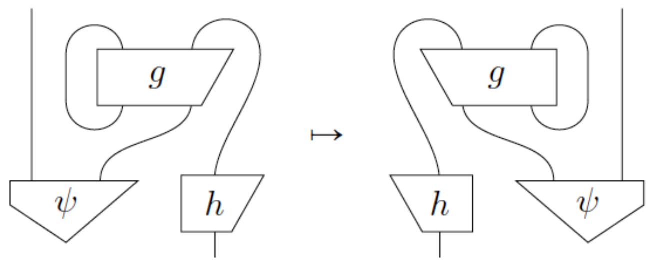
- adjoin, then tranpose



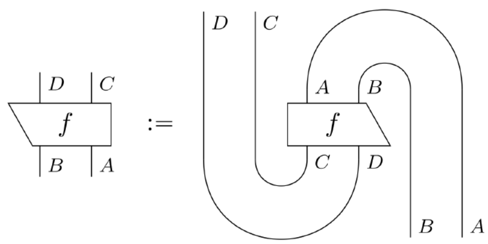
The conjugate of a process is the transpose of its adjoint (or the adjoint of its tranposition), depicted graphically as



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

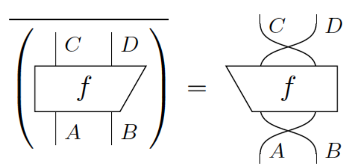


Note that conjugating a process with multiple inputs and outputs is order-reversing

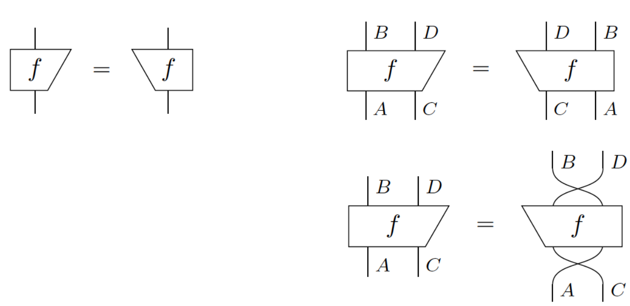


This can be avoided by replacing the transpose by the algebraic transpose, thus defining the *algebraic-conjugate* as follows

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



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.

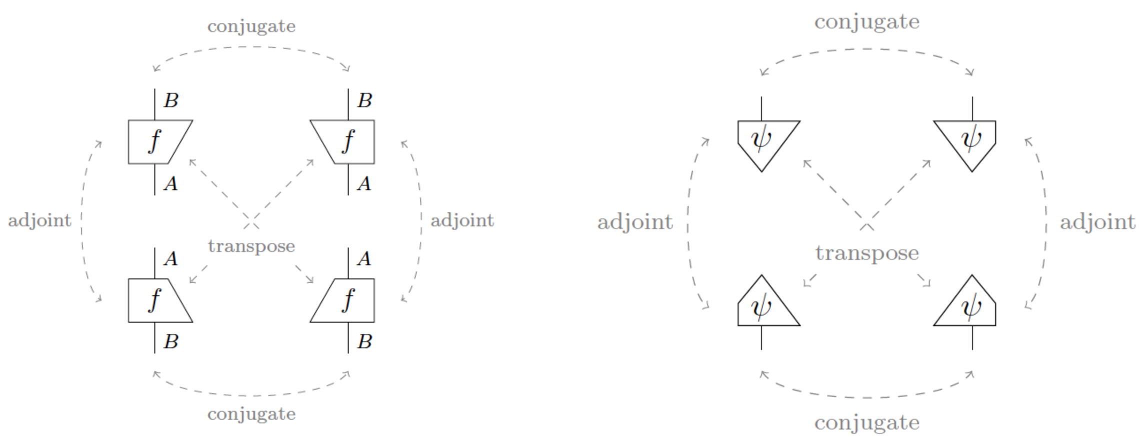
Exercise 12

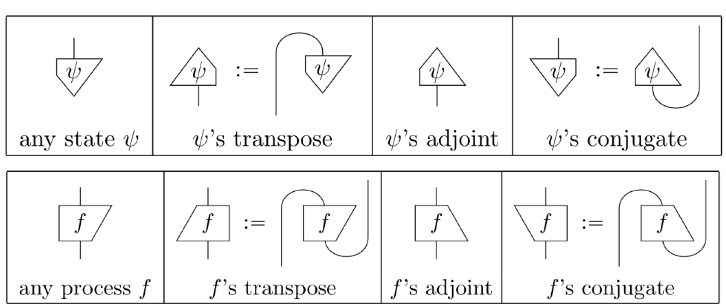
Discuss which relations are (algebraic) self-conjugate.

Exercise 13

What is the conjugate of a scalar?

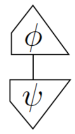
Summing up





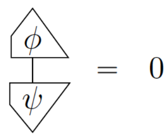
Meeting old friends: inner product

The inner product

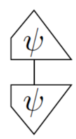


has a clear intuitive meaning: Since the adjoint of a state gives the effect that tests for that state, the inner product expresses *testing state* ψ *for being state* ϕ.

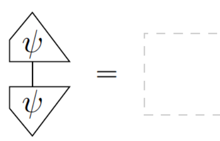
- Orthonormal states:



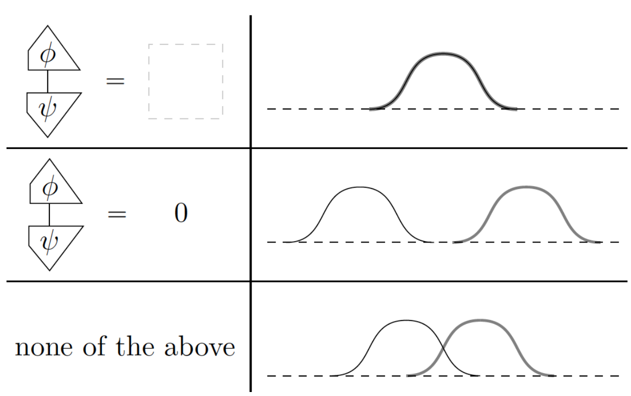
- The squared norm is the inner product with itself:



- Normalised state:



The inner product computes how much similar states are:



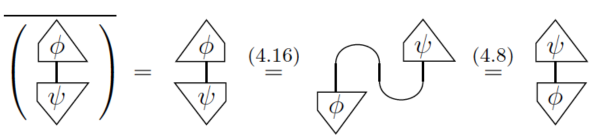
Exercise 14

Discuss the following diagrams in the process theory of relations:

Properties

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

Diagrammatic proofs:



Making