Game Semantics for Quantum Programming

PIERRE CLAIRAMBAULT, Univ Lyon, CNRS, ENS de Lyon, UCB Lyon 1, LIP, France
MARC DE VISME, Univ Lyon, ENS de Lyon, CNRS, UCB Lyon 1, LIP, France
GLYNN WINSKEL, University of Cambridge, United Kingdom

Quantum programming languages permit a hardware independent, high-level description of quantum algorithms. In particular, the quantum $\lambda$-calculus is a higher-order programming language with quantum primitives, mixing quantum data and classical control. Giving satisfactory denotational semantics to the quantum $\lambda$-calculus is a challenging problem that has attracted significant interest in the past few years. Several models have been proposed but for those that address the whole quantum $\lambda$-calculus, they either do not represent the dynamics of computation, or they lack the compositionality one often expects from denotational models.

In this paper, we give the first compositional and interactive model of the full quantum $\lambda$-calculus, based on game semantics. To achieve this we introduce a model of quantum games and strategies, combining quantum data with a representation of the dynamics of computation inspired from causal models of concurrent systems. In this model we first give a computationally adequate interpretation of the affine fragment. Then, we extend the model with a notion of symmetry, allowing us to deal with replication. In this refined setting, we interpret and prove adequacy for the full quantum $\lambda$-calculus. We do this both from a sequential and a parallel interpretation, the latter representing faithfully the causal independence between sub-computations.

CCS Concepts: • Theory of computation → Denotational semantics; • Computer systems organization → Quantum computing;

Additional Key Words and Phrases: Quantum $\lambda$-calculus, Denotational Semantics, Game Semantics, Concurrent Games

ACM Reference Format:

1 INTRODUCTION

Quantum computation, a paradigm that exploits the quantum physical aspects of reality, promises to have a huge impact in computing. Algorithms like Shor’s [Shor 1997] factoring integers in polynomial time and Grover’s [Grover 1996] permitting a database search of size $n$ in $O(\sqrt{n})$ challenge our traditional view of algorithmics and complexity. Meanwhile applications exploiting quantum features in cryptography [Gisin et al. 2002] are already deployed (notably based on Quantum Key Distribution). The field is moving fast, with large companies such as Google, IBM and Intel investing massively in the race for practical quantum hardware.

This activity around quantum computation prompts the need for programming languages, so as to give structured and high-level descriptions of quantum algorithms, with no reference to
the underlying hardware. And indeed, researchers have developed programming languages for quantum computing, see [Gay 2006] for a survey. In this work we are interested in particular in the quantum \(\lambda\)-calculus [Selinger and Valiron 2006], marrying quantum computation with classical control as offered by a higher-order, functional language with recursion and datatypes such as lists. From its inception, the quantum \(\lambda\)-calculus was equipped with a formal operational semantics. Yet a denotational account is also highly desirable: it brings with it compositional, syntax-independent reasoning and may guide the design and analysis of programming languages. Finding denotational semantics for the quantum \(\lambda\)-calculus has attracted a lot of attention. Selinger and Valiron first provided a fully abstract model for the \textit{linear} fragment [Selinger and Valiron 2008]. Delbecque [Delbecque 2011] gave a more dynamic model based on \textit{game semantics} [Abramsky et al. 2000; Hyland and Ong 2000], but for a restricted language: entanglement can happen within types \(\text{qbit}^\otimes n\), but \(\text{qbit}^\otimes (n+m)\) \textit{cannot} be decomposed as \(\text{qbit}^\otimes n \otimes \text{qbit}^\otimes m\). This means that the quantum state remains local, and can be propagated along with the execution flow. More recently, Hasuo and Hoshino proposed a model, based on Girard’s \textit{Geometry of Interaction} [Girard 1989], of a language with a similar restriction as in Delbecque’s language [Hasuo and Hoshino 2017]. In another direction, Malherbe [Malherbe 2013; Malherbe et al. 2013] gave a model based on presheaves, without this restriction on entanglement, albeit without recursion. An adequate denotational semantics for the full quantum \(\lambda\)-calculus with recursion was finally achieved five years ago [Pagani et al. 2014], mixing ideas from models of linear logic with the category \(\text{CPM}\) of \textit{completely positive maps}, a natural mathematical framework for representing (first-order) quantum computing.

While an important breakthrough, this model is not the end of the story. Indeed, it is \textit{static} (it is similar to the \textit{weighted relational models} [Laird et al. 2013]): it collects all completed executions of the classical layer of a quantum program, and annotates each with a quantum weight formalized as a morphism in \(\text{CPM}\). As a consequence, it carries no sequential information (nor does it claim to) and, fundamentally, cannot represent the evaluation order. For instance, it considers equal the two terms \(f\ (g \text{ skip})\) and \(g\ (f \text{ skip})\). This means that the model becomes inapplicable if one considers that a program of the quantum \(\lambda\)-calculus may interact with classical execution environments, in which such sequentiality may be observed (because of the presence of \textit{e.g.} exceptions). Besides this static aspect, the model also contains “junk”, \textit{i.e.} elements not corresponding to programs. In particular, the addition of infinite elements was required to obtain the closure properties used in the model construction of [Pagani et al. 2014]. The model has, for instance, an inhabitant of the interpretation of booleans that evaluates to true with “probability” 2, or even +\(\infty\).

For classical languages, \textit{static} semantics can be opposed to \textit{dynamic or interactive} semantics, that display information about the sequentiality and dynamics of execution – such semantics include \textit{game semantics} and the \textit{geometry of interaction} mentioned above. In this family, and besides the models of fragments mentioned above, a recent breakthrough was achieved in [Dal Lago et al. 2017]: yielding an adequate model of the full quantum \(\lambda\)-calculus based on an extension of the geometry of interaction. Compelling as it is, their approach however lacks the compositionality that one usually expects from a denotational semantics: it describes execution of a quantum program operationally as a \textit{token machine}, \textit{i.e.} an operational process where tokens walk through the syntax of the program, modifying the quantum store. Hence the problem of finding a \textit{compositional} interactive dynamic semantics of the full quantum \(\lambda\)-calculus has until now remained open.

In this paper, we address this, under the form of a computationally adequate \textit{game semantics} for the full quantum \(\lambda\)-calculus. The basic idea behind our approach is to annotate the dynamic description of execution offered by game semantics with annotations by quantum weights in \(\text{CPM}\). Whereas the model of [Pagani et al. 2014] associates quantum weights to completed executions,

\footnote{This holds if \(f, g : 1 \rightarrow 1\) are commands – it is of course not the case that application is commutative in this model.}
our model tracks them throughout computation. Through a condition constraining how quantum annotations may evolve locally, we manage to leverage this dynamic information to avoid the addition of infinite elements of [Pagani et al. 2014]. In fact, we show (Proposition 4.3) that all the quantum annotations we use are in fact superoperators; known to correspond to physically realisable operations on quantum states – ours is the first denotational model of the full quantum λ-calculus to achieve this.

We base our semantics on concurrent games, a framework for games based around ideas from concurrency theory. Initiated by [Abramsky and Melliës 1999; Melliès 2005], this family of game semantics has been actively developed recently, prompted by new foundations introduced in [Rideau and Winskel 2011] – milestones relevant to the present contribution include fully abstract models of higher-order [Castellan et al. 2015], concurrent [Castellan and Clairambault 2016] and probabilistic [Castellan et al. 2018; Winskel 2013] programs. A notion of quantum strategies was already proposed in this setting [Winskel 2013] but it was not broad enough for our purposes so we had to develop a new notion from scratch.

The choice to use concurrent games for a sequential language may seem surprising. In principle, our methodology to add quantum information on top of games could be deployed in sequential games. Elegance matters aside, there are two main reasons for this choice: (1) we want our model to be a good candidate for the challenging open question of full abstraction for the full quantum λ-calculus; already in the non-deterministic case, sequential games struggle with the branching information required to characterize “purely functional” non-deterministic behaviour [Harmer and McCusker 1999]. In contrast, concurrent games represent this information; (2) it is compelling to describe a parallel evaluation of quantum programs, in the hope of having connections with the geometry of interaction of [Dal Lago et al. 2017]; and reflecting the parallelism of quantum circuits. Indeed, besides our sequential interpretation, from our definitions we will get for free a parallel interpretation, reflecting the independence of sub-computations in the quantum λ-calculus.

Outline and contributions. In Section 2 we introduce the quantum λ-calculus and some quantum foundations. In Section 3, we recall the linear probabilistic games of [Winskel 2013]. Our contributions start in Section 4, where we construct the compact closed category QCG of quantum concurrent games. In Section 5, we build on QCG the further structure required to interpret the affine fragment, and prove adequacy. Finally, in Section 6 we mix quantum annotations with symmetry from [Castellan et al. 2015], and extend our adequacy result to the full calculus.

2 THE QUANTUM λ-CALCULUS

We start this section by introducing our programming language of study, the quantum λ-calculus. Our calculus is essentially that of [Pagani et al. 2014], with the proviso that it is affine rather than strictly linear: all variables, even quantum, can be left unused. We will come back to this later.

The design of the quantum λ-calculus has two main inspirations. On the one hand, it is a call-by-value higher-order programming language, not unlike ML, extended to handle quantum datatypes. On the other hand, its typing system includes the exponential or “bang” operator (written !) from Linear Logic [Girard 1987], used to annotate resources that are deemed safe to copy or duplicate. This is indeed required by the laws of physics: the no-cloning theorem implies that quantum states cannot be copied, and are therefore inherently linear. Therefore a programming language designed to manipulate quantum states must be able to explicitly handle linear resources.

In this section we will first give the syntax and typing rules for the quantum λ-calculus. Then we will review the basic elements of quantum mechanics used to represent pure quantum states, and finally use them to describe the operational semantics of our language.
2.1 Syntax and Typing

We first define the types of the quantum λ-calculus, generated by the grammar below.

\[
A, B ::= q\text{bit} | 1 | A \otimes B | A \oplus B | A' | A \rightarrow B | !(A \rightarrow B)
\]

The type q\text{bit} represents qubits, the quantum equivalent of bits and atomic pieces of quantum data. We also have a unit type 1 along with tensors (whose inhabitants are pairs), sums and finite lists (with, as a particular case, the type of integers nat = 1^\ell). We do not have an explicit primitive for classical bits, but those can be easily recovered as syntactic sugar via bit = 1 ⊗ 1. Accordingly, we introduce the syntactic sugar ff = \text{in}_r(\text{skip}) : \text{bit} and tt = \text{in}_l(\text{skip}) : \text{bit}. There are two function types: !(A \rightarrow B) represents functions that may be used more than once – in contrast with functions of type A → B, which are lost after one use. As in \cite{Pagani et al. 2014}, applications of the exponential modality ! are restricted to function types. This restriction forbids the unrealistic type !(q\text{bit}) of replicable qubits. Note however that !(1 → q\text{bit}) makes perfect sense: its elements are functions which may be called arbitrarily many times, and which at each call generates a new independent qubit. Types of the form !(A → B) are non-linear, while all the others are linear.

We now introduce the grammar of terms.

\[
t, u ::= x | λx^A.t | t u | skip | t | u | t \otimes u | \text{let } x^A \otimes y^B = t \\text{ in } u | \text{in}_l t | \text{in}_r t
\]

\[
| \text{match } t \text{ with } (x^A : u_1 | y^B : u_2) | \text{split } | \text{letrec } f^{A\rightarrow B} x^A = t \\text{ in } u
\]

\[
| \text{meas } | \text{new } | U
\]

The first two lines describe a simply-typed λ-calculus with unit, tensor, sums, lists, and recursive definitions. Hopefully any ambiguities concerning the syntax should be cleared up by the typing rules. In particular, though there are no constructors nil and cons for lists, those may be defined as syntactic sugar, respectively as nil = \text{in}_r skip and cons t u = \text{in}_l (t \otimes u).

The last line lists quantum primitives. The first, new : bit → q\text{bit}, prepares a new q\text{bit} based on a given bit. The second, meas : q\text{bit} → bit, performs a measurement. Finally, U : q\text{bit}^⊗n → q\text{bit}^⊗n (where q\text{bit}^⊗n is the iterated tensor q\text{bit} ⊗ \ldots ⊗ q\text{bit}) stands for any unitary map of arity n – the language includes a primitive for every unitary. The precise mathematical meaning of these primitives will be reviewed in Section 2.2, where we recall some elements of quantum computing.

Before we go on to typing rules, we give the grammar of values.

\[
v, w ::= x | λx^A.t | v \otimes w | \text{in}_l v | \text{in}_r v | \text{skip } | \text{split } | \text{meas } | \text{new } | U
\]

Typing judgements have the form Γ ⊢ t : A where Γ is a context, i.e. a list of declarations of distinct variables x_1 : A_1, . . . , x_n : A_n. We say that Γ is non-linear iff it has the form x_1 : !A_1, . . . , x_n : !A_n; we may then write Γ! to emphasize this fact. Most typing rules are displayed in Figure 1. To these we add an exchange rule allowing us to permute variable declarations in contexts – having an explicit exchange helps in writing a clean definition of the interpretation in game semantics.

Throughout this paper we assume that all terms are well-typed.

Example 2.1. The reader can find in \cite{Pagani et al. 2014} simple programs in this language. We mention one of their examples, illustrating the interactions of quantum effects and higher-order:

\[
\vdash \text{telep} : !(1 \rightarrow ((\text{qbit} → \text{bit} \otimes \text{bit}) \otimes (\text{bit} \otimes \text{bit} → \text{qbit})))
\]

Its definition relies on the quantum teleportation algorithm. Applied to skip, telep generates a pair f ⊗ g of entangled functions, respectively of type f : q\text{bit} → bit ⊗ bit and g : bit ⊗ bit → q\text{bit}. These functions are restricted to one use, but such pairs can be generated at will.

After generation, Alice takes f home and Bob takes g home. Alice can then teleport a qubit to Bob by performing the following operations: she applies f to it, obtaining two classical bits b_1 and b_2 that she can send Bob by classical means. Then Bob applies g to these, and doing so recovers
Alicè’s original qubit. Interestingly f and g can be used by Bob to send two classical bits to Alice via a single qubit (the dense coding algorithm). As stated in [Pagani et al. 2014], the entangled f and g form a “single-use isomorphism” between the otherwise non-isomorphic qbit, simply written as |b⟩ ⊗ |b⟩.

On weakening. We allow weakening on any variables, as opposed to only non-linear ones in [Pagani et al. 2014]. Even quantum variables can be left unused. This choice, already present in the original quantum λ-calculus [Selinger and Valiron 2006], is well-suited to game semantics, which form an inherently affine model. It is realistic: one can think of unused quantum variables as implicitly measured and the result thrown away. More importantly, we stress that this is an extension rather than a restriction: our adequacy result holds of course when applied to a term as implicitly measured and the result thrown away. More importantly, we stress that this is an extension rather than a restriction: our adequacy result holds of course when applied to a term satisfying the more restrictive typing discipline of [Pagani et al. 2014].

2.2 Pure Quantum States and Their Operations

Pure quantum states. In order to define the operational semantics of the quantum λ-calculus, we provide a reminder on some basic notions regarding the mathematical representation of quantum states. We first recall here the notions required for the representation of pure states; in Section 2.4 we will go more in depth and describe the representation of mixed states, i.e. probabilistic sums of pure states. The reader is directed to [Nielsen and Chuang 2002] for a more complete treatment.

The elementary unit of quantum information is the qubit, which is a normalized vector in the two-dimensional Hilbert space C^2. The vectors of the canonical basis of C^2 are usually written |0⟩ and |1⟩, thought of as corresponding to the booleans false and true respectively. The state of a qubit can therefore be described as a linear combination α|0⟩ + β|1⟩ (where α, β ∈ C such that |α|^2 + |β|^2 = 1), referred to as a quantum superposition. More generally, a state of a system of n qubits is a vector in the Hilbert space obtained as the n-fold tensor product C^2 ⊗ ⋯ ⊗ C^2, written (C^2)^⊗n, whose canonical basis vectors have the form |b⟩ ⊗ ⋯ ⊗ |b⟩, simply written as |b⟩ ⊗ ⋯ ⊗ |b⟩.

Basic quantum operations. Quantum algorithms on this data are defined via three basic operations: initializations, unitary maps and measurements, all reflected by primitives of the quantum λ-calculus.
With these elements of quantum computing in place, we can formally define the operational semantics of the quantum λ-calculus, again essentially imported from [Pagani et al. 2014].

Example 2.2. A common unitary operation is the Hadamard gate $H : \mathbb{C}^2 \to \mathbb{C}^2$, sending $|0\rangle$ to $\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$ and $|1\rangle$ to $\frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$. Preparing a qubit $|0\rangle$, applying the Hadamard gate and then measuring results in $\text{ff}$ with probability $\frac{1}{2}$ and $\text{tt}$ with probability $\frac{1}{2}$, yielding an unbiased coin toss.

Example 2.3. Another crucial unitary operation is the controlled not gate CNOT : $\mathbb{C}^2 \otimes \mathbb{C}^2 \to \mathbb{C}^2 \otimes \mathbb{C}^2$, defined as $|0\rangle \otimes |0\rangle \mapsto |0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle \mapsto |0\rangle \otimes |1\rangle, |1\rangle \otimes |1\rangle \mapsto |1\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle \mapsto |1\rangle \otimes |1\rangle$. It is a quantum extension of the classical gate preserving the first bit, and negating the second iff the first is $\text{tt}$. Preparing the state $|00\rangle$, applying $H$ on the first qubit followed by CNOT yields the maximally entangled state or "EPR pair" $\frac{1}{\sqrt{2}} |00\rangle + \frac{1}{\sqrt{2}} |11\rangle$, that is behind the quantum teleportation term of Example 2.1. Measuring either qubit yields $\text{ff}$ with probability $\frac{1}{2}$ and $\text{tt}$ with probability $\frac{1}{2}$. Subsequently measuring the other qubit then yields the same result, deterministically.

2.3 Operational Semantics

With these elements of quantum computing in place, we can formally define the operational semantics of the quantum λ-calculus, again essentially imported from [Pagani et al. 2014].

Reductions. Configurations are $[q, \ell, t]$ where $q \in (\mathbb{C}^2)^{\otimes n}$ is a quantum store; $\ell$ is a list of $n$ variables, written $|x_1 \ldots x_n\rangle$, assigning names to the qubits in the store; and $x_{\pi(1)} : q\text{bit}, \ldots, x_{\pi(n)} : q\text{bit} \vdash t : A$ (with $\pi$ a permutation of $(1, \ldots, n)$). Unlike in [Pagani et al. 2014] arbitrary weakening is admissible in our language, so the $x_i$s might not all appear free in $t$. Note that $\ell = |x_1 \ldots x_n\rangle$ acts as a binder for $x_1, \ldots, x_n$ in $[q, \ell, t]$. The notion of $\alpha$-equivalence is extended accordingly.

Reductions are probabilistic, and have the form \([q, \ell, t] \xrightarrow{p} [q', \ell', t']\), meaning that the former reduces to the latter with probabilistic weight \(p \in [0, 1]\). Rules without a probabilistic annotation are deterministic, and implicitly annotated with 1. In particular, all rules of Figure 2 are deterministic. In contrast, the two measurement rules of 3 are probabilistic. To understand them, observe that for any \(1 \leq k \leq n + 1\), a quantum state \(q \in (\mathbb{C}^2)^{\otimes(n+1)}\) can be decomposed uniquely as
\[
q = \alpha_q \rho_q + \beta_q = \alpha(\sum_i \alpha_i |\phi_i\rangle \otimes |0\rangle \otimes |\psi_i\rangle) + \beta(\sum_i \beta_i |\phi_i\rangle \otimes |1\rangle \otimes |\psi_i\rangle)
\]
where \(q_0, q_1\) are normalised, the isolated qubit in the tensor is the \(k\)-th, and \(i\) ranges over the canonical basis of \((\mathbb{C}^2)^{\otimes n}\). We can then define \(q'_0 = \sum_i \alpha_i |\phi_i\rangle \otimes |\psi_i\rangle\) and \(q'_1 = \sum_i \beta_i |\phi_i\rangle \otimes |\psi_i\rangle\) where the \(k\)-th qubit has been eliminated, reflecting the destructive aspect of quantum measurement.

The rule for unitary operations also requires disambiguation. If \(\ell\) has the form \([x_1 \ldots x_k x_{k+1} \ldots x_n]\), then \(q'\) is simply \((U \otimes \text{id})q\). In general though, we obtain \(q'\) as \((\pi^{-1} \circ (U \otimes \text{id}) \circ \pi)q\) where \(\pi\) is the action on the \(n\)-fold tensor of a permutation of \([1, \ldots, n]\) ensuring that \(\ell\) has the required form.

Finally, we add a congruence rule to close them under context. First we define
\[
E[\_] ::= [\_] | E[\_]; u | \text{let } x^A \otimes y^B = E[\_] \text{ in } u | \text{match } E[\_] \text{ with } (x^A : u_1 | y^B : u_2)
\]
the evaluation contexts, and set \([q, \ell, E[\_]] \rightarrow [q, \ell, E[\_']][t] \text{ whenever } [q, \ell, t] \rightarrow [q, \ell, t']\).

Convergence. Using the one-step reduction between configurations defined above, we can define a notion of convergence. First of all, a path \(\rho : [q_1, \ell_1, t_1] \rightarrow [q_n, \ell_n, t_n]\) is a reduction sequence \([q_1, \ell_1, t_1] \xrightarrow{p_1} \cdots \xrightarrow{p_n} [q_n, \ell_n, t_n]\). The weight of a path as above is \(w(\rho) = \prod_{1 \leq i \leq n-1} p_i\).

Finally, if \(t : 1\) is a closed term, then \(t\) converges with probability \(p\), written \(t \parallel_p\), iff
\[
\sum_{\rho : [1, \_] \rightarrow [q_\rho, \ell_\rho, \text{skip}]} w(\rho) = p.
\]

It is with respect to this notion of convergence that we will later on state our adequacy result.

### 2.4 Mixed Quantum States and Completely Positive Maps

In this section, we review some of the basic structures behind the denotational semantics of [Pagani et al. 2014] – these structures will also play a role in our game semantics.

From a program \(\vdash t : \text{qbit}^{\otimes n}\), the reductions terminate with values carrying pure quantum states. But since the reductions are probabilistic, they really yield a (sub-)probability distribution on pure states, also called a mixed state. In building a denotational semantics for quantum programs one must adopt a framework for quantum computation which treats mixed states and their transformations as first-class citizens: a natural choice for that is the category \(\text{CPM}\) of finite-dimensional Hilbert spaces and completely positive maps. Instead of the concrete presentation of \(\text{CPM}\) in [Pagani et al. 2014] we opt for a more abstract one, that will mix better with our game-theoretic construction.

**Hilbert spaces.** Let \(\text{Hilb}\) be the category of finite dimensional Hilbert spaces. It is well-known that \(\text{Hilb}\) is symmetric monoidal; write \(\otimes\) for its tensor product and \(I\) for its unit, which is simply \(\mathbb{C}\). It is further compact closed: any finite-dimensional Hilbert space \(H\) has a dual \(H^* = \text{Hilb}(H, I)\), with a unit \(\eta_H : I \rightarrow H^* \otimes H\) and a co-unit \(\epsilon_H : H^* \otimes H \rightarrow I\). Via this compact closed structure \(\text{Hilb}\) admits a partial trace (to form a traced monoidal category [Joyal et al. 1996]). Given \(f : H \otimes L \rightarrow K \otimes L\) in \(\text{Hilb}\), its partial trace is a map \(\text{Tr}_L(f) : H \rightarrow K\), obtained as in the string diagram in Figure 4. If \(f : L \rightarrow L\), its (complete) trace is \(\text{tr}(f) = \text{Tr}_L(I \otimes f) : I \rightarrow I\) so a scalar factor, matching the usual trace of the matrix of \(f\).
Indeed, \( \text{Hilb}(L, L) \) is isomorphic to \( L^* \otimes L \) whose vectors we can think of as matrices. With slightly more generality than in Section 2.2, a unitary map is \( f : H \to K \) in \( \text{Hilb} \) which is invertible with inverse \( f^{-1} = f^\dagger : K \to H \), given by its conjugate transpose.

Positive operators. An operator \( f : H \to H \) in \( \text{Hilb} \) is positive if it is hermitian, i.e \( f = f^\dagger \), and its eigenvalues are non-negative real numbers. Write \( \text{Op}(H) \), and \( \text{Pos}(H) \), for the set of operators, respectively positive operators, on \( H \). We equip \( \text{Op}(H) \) with an order, the Löwner order (see e.g. [Selinger 2004]), by \( f \leq_L g \) iff \( g - f \in \text{Pos}(H) \). Those \( \rho \in \text{Pos}(H) \) for which \( \text{tr}(\rho) \leq 1 \) are the subdensity operators, a standard for representing mixed quantum states.

Such operators on \( \mathbb{C}^2 \) represent mixed quantum states on one qubit: a pure qubit \( \alpha |0\rangle + \beta |1\rangle \) appears as \( \begin{pmatrix} |\alpha|^2 & a \\ a^* & |\beta|^2 \end{pmatrix} \). Here \( |\alpha|^2 \) and \( |\beta|^2 \) are reals and sum to 1, one may think of \( |\alpha|^2 \) as the probability of measuring ff, of \( |\beta|^2 \) as that of measuring tt, and the other coefficients as required to express the behaviour of the state under unitary transforms. More generally, a pure state expressed as a map \( f : I \to H \) in \( \text{Hilb} \) yields \( \tilde{f} = f f^\dagger \in \text{Pos}(H) \) a density operator that can be also represented as a density matrix. So, subdensity operators can represent pure states – but unlike pure states, they are also stable under convex (sub-probabilistic) sums.

Completely positive maps. Whereas positive operators can represent mixed states, completely positive maps can express transformations that take mixed states to mixed states. The category \( \text{CPM} \) again has finite-dimensional Hilbert spaces as objects, but now a map \( f : \text{Hilb} \to \text{Hilb} \) such that its correspondent \( \tilde{f} : \text{Hilb} \to \text{Hilb} \) is positive, is called positive. The 1-1 correspondence \( f \mapsto \tilde{f} \) between completely positive maps and positive operators is known as the Choi-Jamiołkowski isomorphism.

It is helpful conceptually and technically to regard \( f : \text{Hilb} \to \text{Hilb} \) as taking operators on \( H \) to operators on \( K \), as \( f : \text{Op}(H) \to \text{Op}(H) \) in \( \text{Hilb} \). A linear map \( f : \text{Op}(H) \to \text{Op}(K) \) is positive if it takes positive operators to positive operators. Those \( f : \text{Op}(H) \to \text{Op}(K) \) arising from completely positive maps are those for which \( f \otimes \text{id}_L \) is positive for any \( \text{id}_L : \text{Op}(L) \to \text{Op}(L) \). If a completely positive map \( f \) further satisfies \( \text{tr}(f(A)) \leq \text{tr}(A) \) it is called a superoperator. Superoperators represent the physically realisable operations on quantum states.

We can describe a map in \( \text{CPM} \), regarded as a map between operators, as mapping matrices to matrices linearly. For instance the measurement of a value 0 or 1 of a qubit in \( \mathbb{C}^2 \) is described, respectively, by the two superoperators \( \text{meas}_0, \text{meas}_1 \in \text{CPM}(\mathbb{C}^2, I) \) where

\[
\text{meas}_0 : \begin{pmatrix} a & b \\ c & d \end{pmatrix} \mapsto a \quad \text{and} \quad \text{meas}_1 : \begin{pmatrix} a & b \\ c & d \end{pmatrix} \mapsto d.
\]

Symmetrically, the two superoperators \( \text{new}_0, \text{new}_1 \in \text{CPM}(I, \mathbb{C}^2) \) represent initialization of a qubit:

\[
\text{new}_0 : a \mapsto \begin{pmatrix} a & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad \text{new}_1 : d \mapsto \begin{pmatrix} 0 & 0 \\ 0 & d \end{pmatrix}.
\]

Finally, for \( f : H \to K \) a unitary map, the superoperator \( \tilde{f} : \text{Hilb} \to \text{Hilb} \) takes \( g \in \text{Op}(H) \) to \( fgf^\dagger \).

Denotational semantics. Similarly to \( \text{Hilb} \), \( \text{CPM} \) is also compact closed, and therefore almost equipped to model higher-order programming. Freely adding biproducts to handle \( \oplus \), one obtains a fully abstract model for the linear fragment [Selinger and Valiron 2008]. Replication is more intricate: one needs to (1) enrich spaces with the action of a group of symmetries under which maps
should be invariant [Pagani et al. 2014], and (2) work on a formal completion of CPM to ensure convergence of infinite sums; yielding an adequate model of the quantum \( \lambda \)-calculus.

3 LINEAR PROBABILISTIC GAMES

In this section, we start introducing the required notions towards our quantum game semantics. From a pedagogical viewpoint, we find it simpler to first introduce the probabilistic model, on which we will build in Section 4 to present quantum games.

Game semantics present computation as a game between two players: Player and Opponent. Player plays for the program under study, whereas Opponent plays for its execution environment. They exchange moves obeying rules specified by a game interpreting the type. Finally, a program is represented as a strategy for Player, presenting its interactive behaviour with its environment.

Our strategies follow a game semantics paradigm focusing on positions and causality rather than (sequential) plays, initiated by [Abramsky and Melliès 1999; Melliès 2005], and developed further by [Faggian and Piccolo 2009; Melliès and Mimram 2007]. Our definitions build on the non-deterministic realization of this paradigm in the language of event structures, originating in [Rideau and Winskel 2011]. Recently, this formulation has been actively developed (e.g. [Castellan et al. 2015; Clairambault et al. 2012]) and proved to be a powerful and versatile basis for semantics; including with probabilities [Castellan et al. 2018]. See [Castellan et al. 2017] for an introduction.

3.1 Games and Non-Deterministic Strategies

We start by introducing games and strategies without probabilities, to be added in a second stage.

3.1.1 Games. The game corresponding to a type presents all computational events that may happen in a call-by-value computation on this type, along with their dependencies. As an example, Figure 6 shows the game corresponding to \((1 \otimes 1) \otimes (1 \otimes 1) \otimes 1\) (the details of this correspondence will be given later on in the paper). The diagram is read from top to bottom. Each move (or event) is annotated with a polarity that specifies whether it is due to Player (+) or Opponent (−). Before anything else, the term under study has to evaluate to a value (an abstraction), corresponding to the initial \(\lambda^+\). The next available move is \((\lambda, \lambda)^−\) corresponding to Opponent giving Player an argument – being on a tensor type, this argument is a pair of values, here abstractions. Upon being called, the function may return (yielding the rightmost \(sk^+\)), or may feed a value to one of the two argument functions, which in turn might return. In the diagram of Figure 6 and others to come, we attempt to place moves under the corresponding type component.

Formally, both games and strategies will be certain event structures [Winskel 1986].

Definition 3.1. An event structure (es) is \((E, \leq_E, \text{Con}_E)\) where \(E\) is a set of events partially ordered by \(\leq_E\) the causal dependency relation, and \(\text{Con}_E\) is a nonempty consistency relation consisting of finite subsets of \(E\). These are subject to the following additional axioms:

\[
[e]_E =_{def} \{e' \mid e' \leq_E e\} \text{ is finite for all } e \in E, \\
\{e\} \in \text{Con}_E \text{ for all } e \in E, \\
Y \subseteq X \in \text{Con}_E \text{ implies } Y \in \text{Con}_E, \text{ and} \\
X \in \text{Con}_E \& e \leq_E e' \in X \text{ implies } X \cup \{e\} \in \text{Con}_E.
\]

An event structure with polarities (esp) additionally has a function \(\text{pol}_E : E \to \{-, +\}\) assigning to each event a polarity.

We will often drop the \(E\) in \(\leq_E, \text{Con}_E\), \([e]_E\) whenever it is clear from the context. When introducing an event in the presence of polarities, we will often annotate it to specify its polarity, as in \(e^+, e^-\).

The relation \(e' \leq_E e\) expresses that \(e\) causally depends on the earlier occurrence of event \(e'\). That a finite subset of events is consistent conveys that its events can occur together by some stage in
the evolution of the process. Event structures come with a notion of state: a (finite) configuration is a finite \( x \subseteq E \) which is consistent, i.e. \( x \in \mathsf{Con}_E \), and down-closed, i.e. for all \( e \in x \), for all \( e' \leq_E e \) we have \( e' \in x \) as well. We write \( \mathcal{G}(E) \) for the set of all configurations of \( E \). Finally, we also write \( e \rightarrow_E e' \), called immediate causal dependency iff \( e <_E e' \) with no event strictly in between.

Figure 6 presents an event structure, with all subsets consistent and a tree-like causality drawn with dotted lines. Though the interpretation does not exploit the generality of this definition, we define a game to be an esp \( A \) that is race-free, i.e. if \( x, x' \in \{ a^+_1 \}, x, x' \in \{ a^+_2 \} \in \mathcal{G}(A) \), then \( x \cup \{ a_1, a_2 \} \in \mathcal{G}(A) \) as well. We introduce a few simple constructions on games: the dual \( A^\perp \) of \( A \) has the same components as \( A \), except for \( \mathsf{pol}_{A^\perp} = -\mathsf{pol}_A \). For \( A_1, A_2 \) two games, their parallel composition \( A_1 \parallel A_2 \) has events \( A_1 + A_2 = \{ 1 \} \times A_1 \cup \{ 2 \} \times A_2 \) the tagged disjoint union, and causality \( (i, a) \leq (j, a') \) iff \( i = j \) and \( a \leq_{A_i} a' \). Each subset of \( A_1 \parallel A_2 \) has the form \( X = X_1 \parallel X_2 \) – we set \( X \in \mathsf{Con}_{A_1 \parallel A_2} \) iff \( X_1 \in \mathsf{Con}_{A_1} \) and \( X_2 \in \mathsf{Con}_{A_2} \). Finally, polarity if \( \mathsf{pol}_{A_1 \parallel A_2}(i, a) = \mathsf{pol}_{A_2}(a) \).

3.1.2 Strategies. A strategy presents the computational events of the game that a program is prepared to make, along with enriched causal constraints. For instance, Figure 7 displays what will be the strategy for \( \vdash \lambda x. \text{let } f^{1 \rightarrow 1} \otimes g^{1 \rightarrow 1} = x \text{ in } g \text{ (f skip)} : (1 \rightarrow 1) \otimes (1 \rightarrow 1) \rightarrow 1 \). As the term is an abstraction, the strategy immediately plays the initial \( \lambda^+ \). When called with a pair \( (\lambda, \lambda)^- \), the strategy feeds \( \text{sk}^+ \) to \( f \). When \( f \) returns, the resulting value is copied to \( g \). Finally when \( g \) returns, the term returns at toplevel. This strategy is a total order, but in general strategies are partial orders and may be non-deterministic, formalized as event structures labelled by the game.

Definition 3.2. A strategy on game \( A \) is an es \( S \), with a labelling function \( \sigma : S \rightarrow A \), which is:

(1) Rule-abiding. For any \( x \in \mathcal{C}(S) \), \( \sigma x \in \mathcal{C}(A) \).
(2) Locally injective. For any \( s, s' \in \mathcal{C}(S) \), if \( \sigma s = \sigma s' \) then \( s = s' \).
(3) Receptive. If \( x \in \mathcal{C}(S) \) and \( \sigma x \) extends with negative \( a^- \in A \), i.e. \( a \notin \sigma x \) and \( \sigma x \cup \{ a \} \in \mathcal{C}(A) \),
then there exists a unique \( s \in S \) such that \( \sigma s = a \) and \( x \cup \{ s \} \in \mathcal{C}(S) \).
(4) Courteous. If \( s_1 \rightarrow_S s_2 \), then either \( \sigma s_1 \rightarrow_A \sigma s_2 \), or \( \mathsf{pol}_A(\sigma s_1) = - \) and \( \mathsf{pol}_A(\sigma s_2) = + \).

Besides conditions (1), (2) that express that \( \sigma \) is a sensible labeling of \( S \) by moves of the game (they amount to \( \sigma \) being a map of event structures [Winskel 1986]), receptivity expresses that Player cannot prevent Opponent from playing one of his moves, and courtesy that Player may only choose which positive events to play, and which negative events enable them.

The polarity \( A \) lifts to \( S \) through \( \sigma \); we may hence talk about the polarity of \( s \in S \) and write \( \mathsf{pol}_S(s) \) for \( \mathsf{pol}_A(\sigma s) \). When representing a strategy, we will as in Figure 7 show the event structure \( S \), with moves displayed as their image through \( \sigma \). We will also reflect through the dotted lines the immediate causal order in the underlying game, so as to avoid having to represent both the game and the strategy. The strategy of Figure 7 is deterministic (all finite subsets consistent), but some future diagrams (e.g. in Figure 9) will involve a relation \( \sim \) called immediate conflict and carrying the information on consistency: a finite subset of events is consistent iff its down-closure
does not contain two events in immediate conflict. Not all consistency relations can be represented in this way, but this representation will be sufficient for the purposes of this paper.

3.1.3 Composition. In denotational semantics programs are interpreted compositionally by induction on syntax, each syntactic construction being matched by a semantic one. Crucially, interpreting application and substitution involves composition of strategies. A strategy from game $A$ to game $B$ is simply a strategy $\sigma : S \to A^\perp \parallel B$ playing on a compound game $A^\perp \parallel B$. We will occasionally write $\sigma : A \to B$, keeping the $S$ anonymous. From $\sigma : A \to B$ and $\tau : B \to C$ we wish to define $\tau \circ \sigma : A \to C$ resulting from their interaction – this relies on the following definition.

To define composition, we rely on the following proposition.

**Proposition 3.4.** There is a strategy $\tau \circ \sigma : T \circ S \to A^\perp \parallel C$, unique up to isomorphism, such that there is an order-isomorphism between minimal causally compatible pairs $(x_S, x_T)$ and configurations $z \in \mathcal{G}(T \circ S)$ (we write $z = x_T \circ x_S$ to emphasize this correspondence), and such that writing $\sigma x_S = x_A \parallel x_B$ and $\tau x_T = x_B \parallel x_C$, we then have $(\tau \circ \sigma)(x_T \circ x_S) = x_A \parallel x_C$.

Here, isomorphism between strategies $\sigma : S \to A$ and $\sigma' : S' \to A$ means a bijection $\varphi : S \cong S'$ preserving and reflecting all structure, making the obvious triangle commute.

We show in Figure 8 an example of composition. Concretely, $\tau \circ \sigma : T \circ S \to A^\perp \parallel C$ is computed by replaying within event structures the paradigm that composition is parallel interaction plus hiding, familiar from traditional game semantics: first we construct the interaction $T \circ S$ where we let $S$
and $T$ freely synchronize on $B$; then we hide, keeping only the events that map to $A$ or $C$, i.e. that are not synchronized. In this paper we will not detail the concrete construction of composition, which is covered at length elsewhere [Castellan et al. 2017]. Instead we will only rely on the proposition above, which characterises composition uniquely.

3.1.4 Categorical structure. In order to get a category we need to define an identity, the copycat strategy. Again instead of repeating its concrete definition from [Castellan et al. 2017], we recall a useful characterization. In the proposition below, for $x, y$ configurations of a game $A$ we write $x \subseteq^y y$ iff $x \subseteq y$ and $\text{pol}(y \setminus x) \subseteq \{−\}$; and symmetrically for $x \subseteq^p y$.

**Proposition 3.5.** For any game $A$ there is a unique strategy $\varphi_A : \mathcal{C}_A \to A^\perp \parallel A$ with events $\mathcal{C}_A = A^\perp \parallel A$, $\varphi_A$ the identity function, and configurations those $x_A \parallel y_A$ such that $x_A \supseteq^+ x_A \cap y_A \subseteq^− y_A$.

One may think of $\varphi_A$ as an asynchronous forwarder: whenever it receives a negative event on one side, it propagates it to the other side – in an asynchronous manner. So a state $x_A \parallel y_A \in \mathcal{C}(\mathcal{C}_A)$ has a core $x_A \cap y_A$ of already propagated moves, present in both components, extended with negative not-yet-propagated moves to the right, and dually to the left.

As expected, copycat is neutral for composition up to iso [Rideau and Winskel 2011]. There is a monoidal structure, given on games by $A \parallel B$ and on strategies $\sigma_1 : S_1 \to A^+_1 \parallel B_1$ and $\sigma_2 : S_2 \to A^+_2 \parallel B_2$ by the obvious labeling map $\sigma_1 \parallel \sigma_2 : S_1 \parallel S_2 \to (A_1 \parallel A_2)^+_\parallel (B_1 \parallel B_2)$.

In [Castellan et al. 2017], it is proved that:

**Theorem 3.6.** There is a compact closed category $(\mathcal{CG}, \|, (−)^\perp)$ having games as objects, and as morphisms strategies up to isomorphism.

3.2 Probabilistic Strategies

Having reviewed the core category of concurrent games developed in [Castellan et al. 2017; Rideau and Winskel 2011], we now recall its enrichment with probabilities [Winskel 2013], applied to the semantics of a probabilistic programming language in [Castellan et al. 2018].

3.2.1 Definition. To motivate the definition of probabilistic strategies, let us look first at a simplified example: the interpretation of a $\text{coin} : 1 \to \text{bit}$ primitive, which upon call produces $\text{tt}$ with probability $\frac{1}{2}$ and $\text{ff}$ with probability $\frac{1}{2}$ (coin is actually definable in the quantum $\lambda$-calculus, following the quantum algorithm outlined in Example 2.2). We hope the reader will agree that the diagram of Figure 9 fairly represents the behaviour of $\text{coin}$; where each Player event is annotated with the probability of playing it once its conditions are met. Furthermore, in such a probabilistic choice, the sum of annotations for the different events in conflict should not exceed one.

However, simply annotating events with probabilities is too naive – in reality, probabilistic weights are not local. Indeed, consider the diagram of Figure 10. If Opponent plays only one $\text{sk}^−$, Player will play the corresponding $\text{sk}^+$ with probability $\frac{1}{2}$. But what happens if Opponent plays
both occurrences of $\text{sk}^-$, what is the probability of getting both occurrences of $\text{sk}^+$? If the choices are independent, then it should be $\frac{1}{4}$. On the other hand, maybe the diagram is intended to represent

\[
\text{if } \text{coin}() \text{ then } (\lambda y. \text{skip}) \otimes (\lambda y. \text{skip}) \text{ else } (\lambda y. \bot) \otimes (\lambda y. \bot) : (1 \to 1) \otimes (1 \to 1)
\]

(with the obvious syntactic sugar) in which case, the probability of getting both is $\frac{1}{2}$. So instead of simply annotating events with probabilities, we will annotate configurations.

More precisely, we will annotate a strategy with a probability valuation, a function $v : \mathcal{C}(S) \to [0, \infty)$ with the intuition that $v(x)$ records the probability of reaching $x$, conditionally to Opponent playing the negative events in $x$ (it will follow a posteriori that valuations are less than one).

However, not all such functions make sense as probability valuations. If $x$ can extend with $s_1^+$ and $s_2^+$ in conflict, then we expect that $v(x) \geq v(x \cup \{s_1\}) + v(x \cup \{s_2\})$ – an equality would mean that in state $x$ we make a probabilistic choice between $s_1$ and $s_2$, the inequality giving us a chance to get stuck at $x$. But if $s_1$ and $s_2$ are compatible, how should we adapt the constraint above? Reading $v(x)$ as the probability of ending up in $x$ or above, one observes that $v(x \cup \{s_1, s_2\})$ is accounted for twice in $v(x \cup \{s_1\}) + v(x \cup \{s_2\})$, hence the constraint becomes $v(x) \geq v(x \cup \{s_1\}) + v(x \cup \{s_2\}) - v(x \cup \{s_1, s_2\})$.

In general, formalizing this observation involves an inclusion-exclusion principle that is at the basis of our definition of probabilistic strategies [Winskel 2013].

Definition 3.7. A probabilistic strategy on game $A$ is a strategy $\sigma : S \to A$, together with a probability valuation, i.e., a function $v_\sigma : \mathcal{C}(S) \to [0, \infty)$ which is

- Normalised: $v_\sigma(\emptyset) = 1$;
- Oblivious: If $x \subseteq y$ then $v_\sigma(x) = v_\sigma(y)$, for $x, y \in \mathcal{C}(S)$; and
- Monotone: If $y \subseteq^+ x_1, \ldots, x_n$ then $d_v[y; x_1, \ldots, x_n] \geq 0$,

where the drop function is, for $y, x_1, \ldots, x_n \in \mathcal{C}(S)$,

\[
d_v[y; x_1, \ldots, x_n] = \text{def} \ v_\sigma(y) - \sum_{0 \neq I \subseteq \{1, \ldots, n\}} (-1)^{|I|+1} v_\sigma(x_I),
\]

where $x_I = \bigcup_{i \in I} x_i$ and $v_\sigma(x_I) = v_\sigma(\bigcup_{i \in I} x_i)$ when the union $x_I$ is a configuration and 0 otherwise.

Together these conditions ensure that the range of a probability valuation stays within $[0, 1]$.

3.2.2 Categorical Structure. Copycat, being deterministic [Winskel 2012], can be turned into a probabilistic strategy simply by giving to all configurations the value 1, i.e., $v_{\text{wa}}(x \parallel y) = 1$ for all $x \parallel y \in \mathcal{C}(\text{CC}_A)$. The crucial point is composition, which relies on the proposition below.

Proposition 3.8. If $\sigma : S \to A^\perp \parallel B$ and $\tau : T \to B^\perp \parallel C$ are probabilistic strategies, then $\tau \circ \sigma : T \circ S \to A^\perp \parallel C$, equipped with the function

\[
v_{\tau \circ \sigma} : \mathcal{C}(T \circ S) \to [0, \infty), \quad x_T \circ x_S \mapsto v_\tau(x_T) \times v_\sigma(x_S)
\]

is a probabilistic strategy.

The proof follows from an analysis of properties of drop functions, see [Winskel 2013] for details.

Likewise, the monoidal product of strategies is extended to probabilistic strategies by setting, for $\sigma_1 : S_1 \to A^\perp_1 \parallel B_1$ and $\sigma_2 : S_2 \to A^\perp_2 \parallel B_2$, $v_{\sigma_1 \parallel \sigma_2}(x_1 \parallel x_2) = v_{\sigma_1}(x_1) \times v_{\sigma_2}(x_2)$. Overall, we get:

Theorem 3.9. There is a compact closed category $(\text{PCG}, \parallel, (\cdot)^\perp)$ having games as objects, and as morphisms probabilistic strategies up to isomorphism.

In the statement above, isomorphism is to be understood as an isomorphism between the underlying strategies which additionally preserves the probability valuation. Though all categorical laws hold up to isomorphism of probabilistic strategies, it is too strict for our purposes. Indeed, on the game with one move $\text{sk}^+$, the probabilistic strategies $\text{sk}^+_{\left(\frac{1}{2}\right)} \sim \text{sk}^+_{\left(\frac{1}{2}\right)}$ and $\text{sk}^+_{\left(\frac{1}{2}\right)}$ are not isomorphic.
3.2.3 Simulation Preorder. In this paper, we will occasionally need a coarser equivalence merging these branches, considering probabilistic strategies up to the branching time information.

Definition 3.10. A simulation map from probabilistic strategy \( \sigma : S \to A \) to \( \sigma' : S' \to A \) is a map of event structures \( f : S \to S' \) (i.e. satisfying (1) and (2) of Definition 3.2) such that \( \sigma' \circ f = \sigma \), which is rigid i.e. for all \( s, s' \in S, s \leq_S s' \implies f s \leq_{S'} f s' \); and such that for all \( y \in \mathcal{E}(S') \),

\[
\sum_{x=y} v_{\sigma}(x) \leq v_{\sigma'}(y).
\]

We write \( \sigma \preceq \sigma' \) if there is such a map from \( \sigma \) to \( \sigma' \). This preorder is a congruence: it is preserved by all operations on strategies. In particular, for \( \sigma \preceq \sigma' \), then \( \tau \circ \sigma \preceq \tau \circ \sigma' \). Thus the corresponding equivalence relation \( \equiv \), defined as \( \sigma \equiv \sigma' \) iff \( \sigma \preceq \sigma' \) and \( \sigma' \preceq \sigma \), is also a congruence coarser than isomorphism. Throughout this paper we treat strategies as concrete representatives and make it explicit up to which equivalence equations hold.

Simulation equivalence merges branches, summing valuations: e.g. \( \text{sk}_{\frac{1}{2}}^+ \sim \text{sk}_{\frac{1}{2}}^+ \equiv \text{sk}_{\frac{1}{2}}^+ \).

4 LINEAR QUANTUM GAMES

We now come to the core technical contribution of this paper, quantum concurrent games. Our definitions are inspired by putting together ideas from the probabilistic games presented above, and the use of CPM for the interpretation of the quantum \( \lambda \)-calculus in [Pagani et al. 2014].

4.1 Quantum Games and Strategies

Before getting into the subject proper, let us start by presenting some of the guiding principles and objectives behind the technical definitions. First of all, we expect the interpretation of a closed term \( \vdash t : \text{qbit}^n \) to be a mixed quantum state on \( n \) qubits, i.e. a density matrix. Likewise, the interpretation of a term \( x : \text{qbit}^n \vdash t : \text{bit}^p \) should match the physically realizable quantum transformations from mixed states on \( n \) qubits to mixed states on \( p \) qubits, i.e. superoperators.

Another guiding principle is that the quantum state should only involve components of the type that are visited by the classical control flow. For instance, the term \( f : \text{qbit} \to 1 \vdash \text{skip} : 1 \) does not use its argument \( f \) and does not feed it a quantum state; accordingly its interpretation should be purely classical. More generally, the full quantum state is revealed lazily as the control flow visits quantum components. This is vital for the extension with replication in Section 6: for a term \( \vdash x^\text{qbit} : !(\text{qbit} \to \text{qbit}) \), to avoid the use of infinite dimensional spaces, the semantics will allocate new quantum resources lazily as Opponent calls new copies of the function. In particular, interpreting a term \( \vdash t : A \) where \( A \) is purely classical should yield a probabilistic strategy as in the previous section, even if \( t \) involves some quantum space that disappears in an application of one subterm to another. For instance, \( \vdash \text{meas}(H(\text{new}0)) : \text{bit} \) (see Example 2.2) will yield through the interpretation the probabilistic strategy \( tt_{\frac{1}{2}}^+ \sim ff_{\frac{1}{2}}^+ \) – the internal quantum state used in the course of computation does not appear in the strategy.

4.1.1 Quantum Games. With all these ideas in place, we give the notion of quantum games.

Definition 4.1. A quantum game \((A, \mathcal{H}_{A})\) comprises \( A \) a game, and \( \mathcal{H}_{A} : A \to \text{Hilb} \) associating to any event a finite-dimensional Hilbert space.

We extend \( \mathcal{H}_{A} \) to any \( x \subseteq A \) (not only configurations) by \( \mathcal{H}_{A}(x) = \bigotimes_{a \in x} \mathcal{H}_{A}(a) \) – in particular \( \mathcal{H}_{A}(\emptyset) = I \), i.e. complex numbers, the one-dimensional Hilbert space. The constructions on games of Section 3.1.1 are easily extended to quantum games, by stating \( \mathcal{H}_{A^{\perp}}(a) = \mathcal{H}_{A}(a)^* \) (the dual space), and \( \mathcal{H}_{A \| B}((1, a)) = \mathcal{H}_{A}(a) \text{ and } \mathcal{H}_{B}((2, b)) = \mathcal{H}_{B}(b) \), so that \( \mathcal{H}_{A \| B}(x_{A} \| x_{B}) = \mathcal{H}_{A}(x_{A}) \otimes \mathcal{H}_{B}(x_{B}) \).
we use the partial trace to bring back all terms of the sum to a common space. Alternative ways to
played the ambient Hilbert space grows, so the operators to be added act on different spaces. Hence,
configuration and to be 0, the zero operator, otherwise.

As an example, we give in Figure 11 the quantum game for the type (omitting the !) of our
program of Example 2.1 (we represent symbolically four events (tt, tt), (tt, ff), (ff, tt), and (ff, ff),
all conflicting, with two conflicting events (b₁, b₂) and (b₁', b₂')). This quantum game will indeed
arise as the interpretation of this type according to the definitions of Section 5, but for now we will
just argue that it captures the available computations on this type: first, a program of this type may
converge to an abstraction. When fed skip, it may return any pair of booleans. Finally if the right hand side function is fed a pair of booleans, it may return a qubit.

However, it is not the case that there is one event per (pure or mixed) quantum state. In fact, in
the example above an occurrence of qbit gives rise to exactly one move (though the one on the
right is duplicated to match its several possible causal histories). This reflects that the value of a
qubit cannot directly impact the control flow: it has to be measured first, giving rise to classical
data on which the program can then branch. Instead, the quantum nature of
qubit is reflected by the annotation with the Hilbert space $\mathbb{C}^2$ (or its dual ($\mathbb{C}^2$)*) , which is of course isomorphic to $\mathbb{C}^2$, serving as base for the forthcoming annotations of configurations of strategies with operators.

4.1.2 Quantum Strategies. Finally, we are in a position to introduce quantum strategies. Recall that
in the probabilistic case, for $\sigma : S \rightarrow A$, to any configuration $x \in \mathcal{C}(S)$ we associated a valuation $v_\sigma(x) \in [0, \infty)$ which a posteriori, by the other conditions appeared to be in $[0, 1]$. Likewise now, to any $x \in \mathcal{C}(S)$ we will associate a positive operator $Q_\sigma(x) \in \text{Pos}(\mathcal{H}_A(\sigma x))$, similarly restricted by
the other conditions (the extent of these restrictions is investigated in Section 4.1.3).

Definition 4.2. A quantum strategy on A is a strategy $\sigma : S \rightarrow A$, with a quantum valuation
for $\sigma$, an assignment which to each $x \in \mathcal{C}(S)$ associates $Q_\sigma(x) \in \text{Pos}(\mathcal{H}_A(\sigma x))$, which is

- Normalised: $Q_\sigma(\emptyset) = 1_I$, the identity operator on the unit space $I = \mathbb{C}$,
- Oblivious: If $x \subseteq^+ y$ then $Q_\sigma(x) \otimes \id_{\mathcal{H}_A(\sigma y \setminus \sigma x)} = Q_\sigma(y)$,
- Monotone: If $y \subseteq^+ x_1, \cdots, x_n$ then $d_Q[y; x_1, \cdots, x_n] \geq 0$, where:

$$d_Q[y; x_1, \cdots, x_n] \overset{\text{def}}{=} Q_\sigma(y) - \sum_{\emptyset \neq I \subseteq \{1, \cdots, n\}} (-1)^{|I|+1} \text{Tr}_{\mathcal{H}(\sigma x_I \setminus \sigma y)}(Q_\sigma(x_I)).$$

Analogously to the probabilistic case, we take $Q_\sigma(x_I) = Q_\sigma(\bigcup_{i \in I} x_i)$ when the union is a
configuration and to be 0, the zero operator, otherwise.

A difficulty in extending the "monotone" condition to the quantum case is that as events are
played the ambient Hilbert space grows, so the operators to be added act on different spaces. Hence,
we use the partial trace to bring back all terms of the sum to a common space. Alternative ways to
do so (e.g. completing, or taking the complete trace) suffer from various pathologies (typically, we lose stability of the monotone condition under composition of strategies).

4.1.3 Quantum Strategies and Superoperators. We mention a few properties of the definition.

Firstly, if $\sigma : S \rightarrow A$ is a quantum strategy where $A$ is classical (i.e. $H_A(a) = I$ for all $a \in A$), then for any $x \in C(S)$, $Q_\sigma(x)$ is a positive operator on $\mathbb{C}$, i.e. multiplication by a non-negative real. Through this correspondence, Definition 4.2 degenerates to Definition 3.7, so a quantum strategy $\sigma$:

$$\sigma \in CPM$$

corresponds through the Choi-Jamiolkowski isomorphism to a completely positive:

$$Q_\sigma : \mathcal{H}^+ \rightarrow \mathcal{H}^+$$

of quantum valuations, we have

$$Q_\sigma = \sum_{\lambda} \lambda \cdot \mathbb{C} \cdot 1$$

This proposition is an important consistency check for our notion of quantum strategy: it formalizes the fact that just as in the probabilistic case (where valuations are a posteriori observed to stay in $[0, 1]$), our conditions ensure that the quantum annotations we obtain make physical sense. Constructing a model sufficiently powerful to interpret the quantum $\lambda$-calculus but where quantum valuations remain physically realizable is not self-evident: for instance the interpretation of [Pagani et al. 2014] relies on completely positive maps which are not superoperators.

4.2 A Compact Closed Category of Linear Quantum Strategies

We now investigate the compositional and categorical structures for quantum strategies.

4.2.1 Composition. In preparation for composition, we use the Choi-Jamiolkowski isomorphism and the compact closed structure of $CPM$ to give a more composition-friendly specialization of quantum strategies in the case of strategies from a game $A$ to a game $B$.

Let $\sigma : S \rightarrow A^\perp \parallel B$ be a quantum strategy, and $x \in C(S)$ with $\sigma x = x_A \parallel x_B$. By definition of quantum valuations, we have $Q_\sigma(x) \in \text{Pos}(\mathcal{H}(x_A) \parallel \mathcal{H}(x_B))$. It corresponds through the Choi-Jamiolkowski isomorphism to a completely positive:

$$Q_\sigma^* : \mathcal{H}(x_A) \rightarrow \mathcal{H}(x_B)$$

Unlike $Q^+(x)$, $Q^-(x)$ is not in general a superoperator. However, its form makes it much easier to compose quantum valuations when composing quantum strategies. Indeed, for quantum strategies $\sigma : S \rightarrow A^\perp \parallel B$ and $\tau : T \rightarrow B^\perp \parallel C$, we can simply set the following strikingly simple

$$Q_{\tau \circ \sigma}^*(x_T \circ x_S) = Q_{\tau}^*(x_T) \circ Q_{\sigma}^*(x_S) : \mathcal{H}(x_A) \rightarrow \mathcal{H}(x_C)$$

where $\sigma x_S = x_A \parallel x_B$ and $\tau x_T = x_B \parallel x_C$.

Again through the Choi-Jamiolkowski isomorphism, this provides a quantum valuation $Q_{\tau \circ \sigma}$ as required by Definition 4.2. However, in order to prove that it satisfies the condition of a quantum
strategy, it is convenient to first reformulate these conditions based on $Q^-$ rather than $Q$; in particular the completion in condition "oblivious" and the partial trace in condition "monotone" need rephrasing. Consider $\sigma : S \to A^\perp \parallel B$, and $x, y \in \mathcal{C}(S)$ such that $x \subseteq y$. Furthermore, write $\sigma = \sigma_A \parallel x B$ and $\sigma y = y A \parallel y B$. Then, the **expansion** and **reduction**, with respective type

\[ \| y (Q^\sigma (x)) : \mathcal{H}(a_1, \ldots, a_n) \overset{\text{CPM}}{\to} \mathcal{H}(b_1, \ldots, b_m) \quad \| x (Q^\sigma (y)) : \mathcal{H}(a_1, \ldots, a_n) \overset{\text{CPM}}{\to} \mathcal{H}(b_1, \ldots, b_m) \]

are defined via the compact closed structure of $\text{Hilb}$, as described in Figures 12 and 13 (where $x_A = \{a_1, \ldots, a_n\}, x_B = \{b_1, \ldots, b_n\}, y_A \setminus x_A = \{a_1', \ldots, a_n'\}$ and $y_B \setminus x_B = \{b_1', \ldots, b_m'\}$).

With that, the conditions for a quantum strategy from $A$ to $B$ become:

**Proposition 4.4.** Let $\sigma : S \to A^\perp \parallel B$ be a classical strategy, and $Q^\sigma : (x \in \mathcal{C}(S)) \to \text{Pos}(\sigma x)$ a candidate quantum valuation. Then $Q^\sigma$ is a quantum valuation iff $Q^\sigma$ satisfies:

- Normalised: $Q^- (\emptyset) = 1 : I \overset{\text{CPM}}{\to} I$,
- Oblivious: If $x \subseteq y$ then $Q^- (y) = \| y (Q^- (x))$,
- Monotone: If $y \subseteq x_1, \ldots, x_n$ then $d_{Q^-} [y : x_1, \ldots, x_n] : \mathcal{H}(a_1, \ldots, a_n) \overset{\text{CPM}}{\to} \mathcal{H}(b_1, \ldots, b_m)$, where

\[ d_{Q^-} [y : x_1, \ldots, x_n] = Q^- (y) - \sum_{\emptyset \neq I \subseteq \{1, \ldots, n\}} (-1)^{|I| + 1} \| y (Q^- (x_I)) , \]

again with $Q^- (x_I) = Q^- (\cup_{i \in I} x_i)$ when the union is a configuration and the zero map otherwise.

The proof of this proposition follows directly from the compact closed structure of $\text{Hilb}$. Then, the fact that $Q^- \circ \sigma$ satisfies the condition of Proposition 4.4 mimics that of the probabilistic case [Winskel 2013], with the product of reals replaced by composition in $\text{CPM}$.

### 4.2.2 Categorical Structure

We need to equip the copycat strategy $\alpha_A : \mathcal{C}_A \to A^\perp \parallel A$ with a quantum valuation. As composition of quantum strategies relies on composition in $\text{CPM}$, we expect the quantum valuation of copycat to rely on the identities in $\text{CPM}$. Indeed, for balanced configurations of the form $x \parallel y$ in $\mathcal{C}(\mathcal{C}_A)$, we set $Q^-_{\alpha_A} (x \parallel y) = \text{id}_{\mathcal{H}(x)} : \mathcal{H}(x) \overset{\text{CPM}}{\to} \mathcal{H}(x)$.

In general, configurations of copycat are $x \parallel y$ where $x \supseteq x \cap y \subseteq y$; i.e. a balanced $x \cap y \parallel x \cap y \in \mathcal{C}(\mathcal{C}_A)$ with a negative extension to $x \parallel y$. The definition is forced by obliviousness to be:

\[ Q^-_{\alpha_A} (x \parallel y) = \| x \parallel y (\text{id}_{\mathcal{H}(x \cap y)}) : \mathcal{H}(x) \overset{\text{CPM}}{\to} \mathcal{H}(y) . \]

For the monoidal product, we again rely on that of $\text{CPM}$: given $\sigma_1 : S_1 \to A^\perp \parallel B_1$ and $\sigma_2 : S_2 \to A^\perp \parallel B_2$, with $x_1 \in \mathcal{C}(S_1)$ and $x_2 \in \mathcal{C}(S_2)$ such that $\sigma_1 x_1 = x_{A_1} \parallel x_{B_1}$ and $\sigma_2 x_2 = x_{A_2} \parallel x_{B_2}$, we set:

\[ Q^-_{\sigma_1 \circ \sigma_2} (x_1 \parallel x_2) = Q^-_{\sigma_1} (x_1) \otimes_{\text{CPM}} Q^-_{\sigma_2} (x_2) : \mathcal{H}(x_{A_1}) \otimes \mathcal{H}(x_{A_2}) \overset{\text{CPM}}{\to} \mathcal{H}(x_{B_1}) \otimes \mathcal{H}(x_{B_2}) . \]

Pairing the usual analysis of composition with algebraic manipulations in $\text{Hilb}$, we obtain:

**Theorem 4.5.** There is a compact closed category $(\mathcal{QCG}, \|, (-)^\perp)$ having quantum games as objects, and as morphisms quantum strategies up to isomorphisms preserving quantum valuations.
4.2.3 *Simulation Equivalence.* Just as in the probabilistic case, isomorphism of quantum strategies is too strict for some purposes. The definition of *simulation equivalence*, as stated in Section 3.2.3, extends transparently to the quantum case: for $\sigma : S \to A$ and $\sigma' : S' \to A$, we have $\sigma \lessdot \sigma'$ iff there is $f : S \to S'$ satisfying (1) and (2) of Definition 3.2, which is rigid (i.e. preserves the causal order) and such that for all $y \in \mathcal{C}(S')$, $\sum_{x=y} Q_{\sigma}(x) \leq L Q_{\sigma'}(y)$. Again we write $\approx$ for the corresponding equivalence, which is preserved by all operations on strategies. We immediately get:

**Proposition 4.6.** *The full subcategory of QCG (up to simulation equivalence) whose objects have one positive event is equivalent to the category CPM$_{\leq 1}$ of Hilbert spaces and superoperators.*

5 **ADEQUACY FOR THE AFFINE FRAGMENT**

Before introducing the constructions required for replication and $!$, we interpret the affine ($!$-free) fragment of the quantum $\lambda$-calculus: it is obtained by removing all types comprising $!$, along with letrec. As divergence is no longer definable we add a new constant $\bot$, with typing $\Gamma \vdash \bot : A$.

To construct the interpretation, we will first describe the interpretation of the ambient affine call-by-value $\lambda$-calculus, and then that of the quantum primitives. While we are not aware of a reference for a games model of a linear/affine call-by-value $\lambda$-calculus, our methodology is certainly not surprising, and is strongly related with Melliès’ account of tensorial logic [Melliès 2012].

From now on, by *game* we mean quantum game and by *strategy* we mean quantum strategy.

5.1 **Call-By-Value Primitives**

As is well-known in game semantics, the duality between call-by-name and call-by-value reflects the computational events in call-by-value. Indeed, strategies in QCG are positive, strategies need to be negative, but in fact both for call-by-name and call-by-value, strategies of Definition 3.2 are positive. Likewise, a strategy $\sigma : S \to A$ is negative iff the minimal events of $S$ are negative.

To apply it to call-by-value we restrict it to positive games. The full subcategory of QCG of positive games has objects the positive games, and morphisms the strategies. It might come as a surprise that though the games are positive, strategies need to be negative, but in fact both for call-by-name and call-by-value, evaluation is triggered by the evaluation environment. In the former case, it is done by requiring an output, while in the latter it is done via the context, by feeding values for the free variables. Indeed, strategies in QCG wait first for a move by Opponent, necessarily of the form of a minimal event, and then that of the quantum primitives. While we are not aware of a reference for a games model of a linear/affine call-by-value $\lambda$-calculus, our methodology is certainly not surprising, and is strongly related with Melliès’ account of tensorial logic [Melliès 2012].

From now on, by *game* we mean quantum game and by *strategy* we mean quantum strategy.

5.1.1 **Quantum Arenas.** We start by defining positive games and negative strategies.

**Definition 5.1.** A game $A$ is negative (resp. positive) iff all its minimal events are negative (resp. positive). Likewise, a strategy $\sigma : S \to A$ is negative iff the minimal events of $S$ are negative.

The category QCG$_+$ of positive games has objects the positive games, and morphisms the negative strategies $\sigma : A \to B$; it is a subcategory of QCG. It might come as a surprise that though the games are positive, strategies need to be negative, but in fact both for call-by-name and call-by-value, evaluation is triggered by the evaluation environment. In the former case, it is done by requiring an output, while in the latter it is done via the context, by feeding values for the free variables. Indeed, strategies in QCG$_+$ wait first for a move by Opponent, necessarily of the form of a minimal move in $A$ thought of as feeding an argument value to $\sigma$. The strategy may then perform further computation in $A$, or return a value in $B$; reflecting the computational events in call-by-value.

Accordingly, QCG$_+$ will be the target of our interpretation: a term $\Gamma \vdash t : A$ will yield a morphism $\llbracket t \rrbracket : \llbracket \Gamma \rrbracket \to \llbracket A \rrbracket$ in QCG$_+$, where $\llbracket \Gamma \rrbracket$ and $\llbracket A \rrbracket$ will both be certain positive games.

In fact, not all positive games will be reached by the interpretation. The following definition captures more precisely the games arising through the interpretation of types.

**Definition 5.2.** A *quantum arena* is a positive quantum game $A$ such that each $x \in \mathcal{C}(A)$ has at most one minimal event, and that is alternating: if $p_1 \rightarrow_A p_2$, then $\text{pol}_A(p_1) \neq \text{pol}_A(p_2)$.

The game of Figure 6 is an arena (with trivial quantum annotations), and so is Figure 11. In both cases, there is only one initial move – in general there may be many; corresponding to the values
A ⊥ B = ⊕ \downarrow_{H_{A,i} \otimes H_{B,j}} (N_{A,i} \parallel N_{B,j}).

This captures the fact that a value on A ⊗ B is a pair v ⊗ w of values on A and B respectively; accordingly a root of A ⊗ B corresponds to a pair of a root of A and a root of B. After this value v ⊗ w is played, the moves available are those enabled by v in A and w in B. We present in Figure 14 an example of application of the tensor construction. Anticipating on the interpretation of types, we also give the corresponding types to help build up intuition on the correspondence.

5.1.2 Premonoidal Structure. The tensor also has an effect on strategies. For σ : A → A′ and τ : B → B′ in QA we define σ ⊗ τ : A ⊗ B → A′ ⊗ B′, also in QA. Intuitively, σ ⊗ τ acts as follows: when initiated with (a, b)̸− on the left, it starts σ on a̸” and τ on b̸”, in parallel, as long as they both play on the left hand side. Whenever they are both ready to play a value on the right hand side, they synchronise and play the pair together; and then the remaining part of the two strategies is played in parallel (our sequential interpretation will only use the case where one side is the identity, and so remain sequential). We illustrate the construction in Figure 15 (omitting quantum annotations).

Fig. 14. Example for the tensor construction

Fig. 15. Tensor construction on strategies
To avoid an unwieldy concrete construction, we characterise the tensor of strategies uniquely via its action on configurations. Two configurations \( x = x_A \parallel x_A' \in \mathcal{C}(A^\perp \parallel A') \) and \( y = y_B \parallel y_B' \in \mathcal{C}(B^\perp \parallel B') \) are compatible if \( x_A \neq 0 \) iff \( y_B \neq 0 \), and \( x_A' \neq 0 \) iff \( y_B' \neq 0 \). If it is so, then by merging the minimal events of \( x_A \) and \( y_B \), and those of \( x_A' \) and \( y_B' \), we get \( x \sqcup y \in \mathcal{C}((A \otimes B)^\perp \parallel (A' \otimes B')) \).

**Proposition 5.3.** Let \( \sigma : S \to A^\perp \parallel A' \) and \( \tau : T \to B^\perp \parallel B' \). Then, there is

\[
\sigma \otimes \tau : S \sqcup T \to (A \otimes B)^\perp \parallel (A' \otimes B'),
\]

unique up to isomorphism, such that (1) \( \mathcal{C}(S \sqcup T) \) is order-isomorphic to the set of pairs \( (x_S, x_T) \in \mathcal{C}(S) \times \mathcal{C}(T) \) such that \( x_S \) and \( x_T \) are compatible, ordered by the product inclusion (we write \( x_S \sqcup x_T \in \mathcal{C}(S \sqcup T) \)) for the configuration corresponding to \((x_S, x_T)\); (2) we have \( \sigma \otimes \tau (x_S \sqcup x_T) = (\sigma x_S) \sqcup (\tau x_T) \); and (3) we have \( \mathcal{Q}^{-}_\sigma \sigma \otimes \tau (x_S \sqcup x_T) = \mathcal{Q}^-\sigma (x_S) \otimes \mathcal{Q}^-\tau (x_T) \).

The tensor is functorial in both components, i.e. \( (\sigma_2 \otimes B) \circ (\sigma_1 \otimes B) = (\sigma_2 \circ \sigma_1) \circ B \), and symmetrically; but not bifunctorial: for \( \sigma : A \to A' \) and \( \tau : B \to B' \), the three strategies \( \sigma \otimes_1 \tau = (\sigma \otimes \tau \circ (\sigma \otimes B), \sigma \otimes_r \tau = (\sigma \otimes B') \circ (A \otimes \tau) \), and \( \sigma \otimes \tau \), may all be distinct. In other words, \((\mathcal{QA}, \otimes, I)\) is a premonoidal category \((\text{Power and Robinson 1997})\), which is expected for call-by-value. The two \( \otimes_1 \) and \( \otimes_r \), already appearing in [Honda and Yoshida 1999] denote pairings with respectively a left-then-right and right-then-left evaluation strategy. In our concurrent setting, to these two is adjoined the symmetric \( \sigma \otimes \tau \), evaluating the two in parallel. Our sequential interpretation will use \( \otimes_1 \), matching the operational semantics; while our parallel interpretation will use \( \otimes \).

5.1.3 Thunkability and Values. The thunkable strategies will be semantic counterparts of values.

**Definition 5.4.** A strategy \( \sigma : S \to (A^\perp \parallel B) \) in \( \mathcal{QA} \) is thunkable iff for every minimal \( s_1^- \in S \), there is exactly one \( s_2^+ \in S \) such that \( s_1 \to_S s_2 \), mapping to \( B \). Furthermore, \( d_Q(\{s_1\}; \{s_1, s_2\}) = 0 \).

This definition corresponds to the instantaneous, in our particular case, of Führmann’s semantic notion of values as thunkable maps [Führmann 1999]. It will be crucial later that the interpretation of values will yield thunkable strategies, and that thunkable strategies enjoy the following property:

**Lemma 5.5.** If \( \sigma : A \to B \) in \( \mathcal{QA} \) is thunkable, then it is in the center of the premonoidal category \( \mathcal{QA} \), meaning that for all \( \tau : A' \to B' \), we have \( \sigma \otimes_1 \tau = \sigma \otimes_r \tau = (\sigma \otimes \tau) \).

Besides, thunkable strategies are stable under composition, so \( \mathcal{QA} \) has a subcategory \( \mathcal{QA}_t \), whose objects are arenas, and morphisms are thunkable strategies. From the lemma above, it follows that the premonoidal structure of \( \mathcal{QA} \) informs a monoidal structure on \( \mathcal{QA}_t \); the tensor becomes bifunctorial; in fact the three tensors of Section 5.1.2 coincide on thunkable strategies.

5.1.4 Closure. We introduce the structure matching the affine arrow type.

**Definition 5.6.** If \( A \) and \( B \) are two arenas, writing \( A = \bigoplus_{i \in I} \downarrow H_{A,i} N_{A,i} \), their arrow is

\[
A \to B = \downarrow I \left( \bigoplus_{i \in I} \downarrow H_{A,i} (N_{A,i} \parallel B^\perp) \right) \uparrow.
\]

This is reminiscent of the usual arrow construction for call-by-value games [Honda and Yoshida 1999]. It is helpful to see how it describes the computational events available in call-by-value over an arrow type. First, the initial move coming from the \( \downarrow I \) expresses evaluation to a lambda, with no Hilbert space. Then Opponent may feed a value on \( A \), corresponding to one of the incompatible \( \downarrow H_{A,i} \). In turn, Player may either keep playing on the component \( N_{A,i} \), or play in \( B \).

Let us write \( I : \mathcal{QA}_t \to \mathcal{QA} \) for the obvious inclusion functor. The constructions on strategies corresponding the arrow type along with the required equations are all summed up by the following.
\[ x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i \] \[ [\Gamma \vdash \lambda x^A, t : A \to B] = \lambda_{[A]}([t]) \] \[ [\Lambda, \Omega + t \otimes u : A \otimes B] = [I] \otimes [u] \] \[ [\Lambda, \Omega + \text{let } x^A \otimes y^B = t \text{ in } u : C] = [u] \circ ([\Omega] \otimes [\Omega]) \] \[ [\Gamma + \text{split} : A^\ell \otimes (A \otimes A^\ell)] = \lambda_{[A]}(\text{unfold}_{[A]}(A) \otimes [t]) \] \[ [\Lambda, \Omega + \text{match } t \text{ with } (x^{A_1} : u_1 | x^{A_2} : u_2) : C] = \left(\left([u_1] \circ ([\Omega] \otimes [\Omega])\right) \circ \left([I] \otimes [\Omega]\right)\right) \] \[ [\Gamma + \text{skip} : 1] = [w] \] \[ [\Gamma + \perp : A] = \perp_{[A]} \otimes [t] \] \[ [\Lambda, \Omega + t : B] = [I] \otimes [u] \] \[ [\Gamma + \text{in}_l(t) : A \otimes B] = [\iota] \circ [t] \] \[ [\Gamma + \text{in}_r(t) : A \otimes B] = [\iota] \circ [t] \]

**Proposition 5.7.** For any arena \( A \), we have an adjunction \( QA \xrightarrow{\perp} A \xleftarrow{\perp} QA \).

In other words, for all arenas \( A, B \) there is an evaluation \( ev_{A,B} : (A \to B) \otimes A \to B \); and for every \( \sigma : A \otimes B \to C \) there is a unique (up to iso) thunkable \( \lambda_B(\sigma) : A \to (B \to C) \) s.t. \( ev_{B,C} \circ (\lambda_B(\sigma) \otimes B) \equiv \sigma \). Altogether, we have a structure that could naturally be called a linear closed Freyd category, by extension of the usual usage [Power and Thielecke 1999].

### 5.1.5 Interpretation

We now describe the interpretation of the classical affine primitives. The unit type \( 1 \) is interpreted by the arena with only one move \( sk^+ \) (and trivial Hilbert space), also written \( 1 \). For lists, we simply set, for any arena \( A, A^\ell = \bigoplus_{n \in \mathbb{N}} A^{\otimes n} \). Up to isomorphism of arenas we have \( A^\ell \equiv 1 \oplus (A \otimes A^\ell) \). This isomorphism easily lifts to \( QA \), yielding strategies \( \text{fold}_A : 1 \oplus (A \otimes A^\ell) \to A^\ell \) and \( \text{unfold}_A : A^\ell \to 1 \oplus (A \otimes A^\ell) \), inverse to each other up to isomorphism of strategies.

The interpretation closely follows the structure introduced. Each type \( A \) yields an arena \([A]\) by matching classical type constructors with their corresponding arena constructions. **Contexts** \( \Gamma = x_1 : A_1, \ldots, x_n : A_n \) are interpreted as tensors \( \bigotimes_{1 \leq i \leq n} [A_i] \). **Typing judgements** \( \Gamma \vdash t : A \) are interpreted as strategies \( [t] : [\Gamma] \to [A] \) in \( QA \). **Values** \( \Gamma \vdash v : A \) are interpreted as thunkable \( [v] : [\Gamma] \to [A] \). We display in Figure 16 the interpretation of the classical affine fragment, writing \( w_A : A \vdash 1 \) for the affine projection which to any value in \( A \) reacts by playing \( sk^+ \) on the right, and \( \perp_A : 1 \to A \) for the diverging strategy with no positive moves. To aid readability, we omit structural isomorphisms for \( \otimes \) (associativity, symmetry, unit), and only display the term on the right hand side.

Finally, the case for matching relies on the distributivity \( A \otimes (B \otimes C) \equiv (A \otimes B) \otimes (A \otimes C) \). In other words, for all arenas \( A, B \) there is an evaluation \( ev_{A,B} : (A \to B) \otimes A \to B \); and for every \( \sigma : A \otimes B \to C \) there is a unique (up to iso) thunkable \( \lambda_B(\sigma) : A \to (B \to C) \) s.t. \( ev_{B,C} \circ (\lambda_B(\sigma) \otimes B) \equiv \sigma \). Altogether, we have a structure that could naturally be called a linear closed Freyd category, by extension of the usual usage [Power and Thielecke 1999].

**5.1.6 On Sequentiality.** In Figure 16, the clauses for \( t \otimes u \) and \( t \otimes u \) use \( \otimes_I \). It is familiar from the categorical models of call-by-value that the two bifunctors \( \otimes_I \) and \( \otimes_r \) offer two possibilities for the interpretation, notably of \( t \otimes u \) and \( t \otimes u \), matching respectively the left-then-right and right-then-left evaluation strategies for call-by-value. Our choice to use \( \otimes_I \) in Figure 16 permits a closer relationship with the operational semantics, which follows a left-then-right strategy. In semantic terms, this natural choice yields a *sequential* strategy. We will not use this sequentiality in the remainder of the development, but as it is an important aspect of our model, we recall from [Castellan et al. 2014]:
We now complete the interpretation of the affine fragment and prove its adequacy.

We show in Figure 19 the interpretation of a term, along with the two key steps for its computation.

Fig. 18. Quantum strategies for quantum primitives

Definition 5.8. A strategy \( \sigma : S \to A \) is sequential if (1) for each \( s \in S \), \([s]\) is a total order (i.e. \( \leq_S \) is forest-shaped), and (2) if \([s]\) extends with some distinct positive \( s_1, s_2 \), then \([s] \cup \{s_1, s_2\} \notin \text{Con}_S\).

Condition (2) expresses that Player does not spawn parallel threads (so all Player branchings are non-consistent), and (1) that he does not merge existing ones. All basic strategies used in the interpretation are sequential, and sequentiality is preserved by composition. Note that sequentiality does not mean that the strategy is totally ordered overall, but that as long as Opponent also remains sequential, only one thread will be live throughout computation. For instance, the strategy displayed in Figure 10 is sequential, although there is a non-conflicting Opponent branching.

The strategies in previously existing call-by-value games [Abramsky and McCusker 1997; Honda and Yoshida 1999] are inherently sequential (not in the formal sense above – sequentiality is hardwired into these models). In contrast, our setting allows strategies that are not sequential. This means that besides using \( \otimes_l \) and \( \otimes_r \), we have a third option: replacing \( \otimes \) with the parallel \( \parallel \) in the clauses for \( t u \) and \( t \parallel u \); we write \([\_\_\_]\) for the corresponding parallel interpretation. From a program, it yields a strategy implementing its parallel evaluation: in expressions \( t u \), the function and argument are converted to values in parallel before substitution is performed – and likewise for \( t \parallel u \). As an illustration we display in Figure 17 the parallel interpretation of \( \lambda x. \text{let } f^{1 \to 1} \otimes g^{1 \to 1} = x \text{ in let } y \otimes z = (f \text{ skip}) \otimes (g \text{ skip}) \text{ in } y; z \), to be compared with its sequential interpretation equal to that in Figure 7. From now on, \([\_\_]\) means either of the two – our constructions and proofs rely on laws satisfied by both and work either way.

5.2 Soundness and Adequacy for the Affine Quantum \( \lambda \)-Calculus

We now complete the interpretation of the affine fragment and prove its adequacy.

5.2.1 Interpretation of Quantum Primitives. As expected, we set \([q\text{bit}]\) to be the arena with one (positive) move \( q \), with \( \mathcal{H}(q) = \mathbb{C}^2 \); also referred to as \( q\text{bit} \). We introduce in Figure 18 quantum strategies \( \text{meas} : q\text{bit} \to \text{bit} \), \( \text{new} : \text{bit} \to q\text{bit} \), along with one \( U : q\text{bit} \otimes q\text{bit} \to q\text{bit} \otimes q\text{bit} \) for each unitary of arity \( n \). Using those, we complete the interpretation of the affine fragment with:

\[
\begin{align*}
[\Gamma \vdash \text{meas} : q\text{bit} \to \text{bit}] &= \lambda_{\text{qbit}(\text{meas})} \otimes w[\Gamma] \\
[\Gamma \vdash \text{new} : \text{bit} \to q\text{bit}] &= \lambda_{\text{bf}(\text{new})} \otimes w[\Gamma] \\
[\Gamma \vdash U : q\text{bit} \otimes q\text{bit} \to q\text{bit} \otimes q\text{bit}] &= \lambda_{q\text{bit}^{\otimes n}}(U) \otimes w[\Gamma]
\end{align*}
\]

Together with Figure 16, this concludes the interpretation of the affine fragment. To illustrate it we show in Figure 19 the interpretation of a term, along with the two key steps for its computation.

5.2.2 Convex Sums. In preparation for our proof of soundness and adequacy, we introduce one final operation on quantum strategies: sub-probabilistic sums (or convex sums).
Given a finite family of strategies \( \sigma_i : S_i \to A \) and positive real coefficients \( \{p_i\}_{i \in I} \) such that \( \sum_{i \in I} p_i \leq 1 \), we require \( \sum_{i \in I} \sigma_i : S \to A \), a strategy on \( A \), acting like \( \sigma_i \) with probability \( p_i \). Intuitively, \( S \) is the sum of all the \( S_i \)'s, as in the sum of arenas. However, this sum includes multiple copies of the possible minimal negative events of \( A \), failing receptivity. Therefore all these copies need to be merged. Rather than showing the corresponding concrete construction we state the following, where \( \mathcal{C}(S)^+ \) denotes the configurations of \( S \) with at least one positive event.

**Proposition 5.9.** Let \( \sigma_i : S_i \to A \) and \( \{p_i\}_{i \in I} \) be families as above.

Then, there is \( \sigma : S \to A \), unique up to iso, such that (1) \( \mathcal{C}(S)^+ \) is order-isomorphic to the disjoint union \( \bigcup_{i \in I} \mathcal{C}(S_i)^+ \) for \( x \in \mathcal{C}(S_i)^+ \) we write \( (i, x) \in \mathcal{C}(S)^+ \) to emphasize the correspondence; (2) for all \( (i, x) \in \mathcal{C}(S)^+ \), \( \sigma(i, x) = \sigma_i x \); and (3) for all \( (i, x) \in \mathcal{C}(S)^+ \), \( Q_\sigma(i, x) = p_i Q_{\sigma_i} x \).

We write \( \sum_{i \in I} p_i \sigma_i \) for this \( \sigma : S \to A \).

### 5.2.3 Computational Adequacy

In order to prove adequacy, we need to give an interpretation to the configurations used in the operational semantics (see Section 2.3). Recall that those have the form \([q, \ell, t] \) with \( q \in (\mathbb{C}^2)^{\otimes n}, \ell = [x_1 \ldots x_n] \) and \( x_\pi(1) : \text{qbit}, \ldots, x_\pi(n) : \text{qbit} \vdash t : A \) with \( \pi \) a permutation of \( \{1, \ldots, n\} \); informing a unitary \( \pi : (\mathbb{C}^2)^{\otimes n} \to (\mathbb{C}^2)^{\otimes n} \). Then, we set:

\[
[[q, \ell, t]] = [t] \odot \widehat{\pi} \circ \widehat{q} : 1 \to [A]
\]

silently coercing \( \widehat{\pi} : (\mathbb{C}^2)^{\otimes n} \xrightarrow{CPM} (\mathbb{C}^2)^{\otimes n} \) and \( \widehat{q} : I \xrightarrow{CPM} (\mathbb{C}^2)^{\otimes n} \) to strategies via Proposition 4.6.

**Invariance.** The core argument for correctness of the model is the following invariance lemma.

**Lemma 5.10.** For any \([q, \ell, t] \), with \([q, \ell, t] \xrightarrow{p_i} [q_i, \ell_i, t_i] \) \( i \in I \) all its possible reductions,

\[
[[q, \ell, t]] \equiv_r \sum_{i \in I} p_i [[q_i, \ell_i, t_i]]
\]

**Proof.** By induction on the operational semantics, relying directly either on the structures of Section 5.1 or on equations pertaining to the standard encoding of quantum structures in CPM. \( \square \)

Note that \( I \) is finite and \( \sum_{i \in I} p_i \leq 1 \): branching in the operational semantics, coming from measures, generates two branches with respective weights \( |\alpha|^2 \) and \( |\beta|^2 \) (where \( |\alpha|^2 + |\beta|^2 = 1 \)). So the tree of all reductions from a configuration is finitely branching. It has also finite depth: since the language under study is affine, it is easy to find a decreasing measure for all reductions.

**Adequacy.** As convergence is defined on closed terms \( \vdash t : 1 \), we have a closer look at their interpretation: they yield strategies \( [t] : 1 \to 1 \), which (up to simulation equivalence) correspond via Proposition 4.6 to superoperators from \( \mathbb{C} \) to \( \mathbb{C} \), i.e. linear maps \( x \mapsto \alpha x \) with \( \alpha \in [0, 1] \). We write \( [[t]]_{\alpha} \) accordingly. Relying on these observations and the invariance lemma, we easily deduce:
6 REPLICATION AND THE FULL QUANTUM λ-CALCULUS

As usual, replication can be obtained by creating copies of the replicated moves [Abramsky et al. 2000]. But one must then express that these copies are indistinguishable. In concurrent games, this was achieved by concurrent games with symmetry [Castellan et al. 2014, 2015], later generalized to probabilistic games [Castellan et al. 2018], simply requiring probability valuations to be invariant under symmetry. We now extend it to the quantum case, and complete our interpretation.

6.1 Quantum Games With Symmetry

Our setting puts together the thin concurrent games with symmetry of [Castellan et al. 2015] (see details in [Castellan et al. 2016]) and the quantum annotations of this paper, required to be invariant under properties. By lack of space we include few explanations and motivations on the definitions of symmetry, for that we direct the reader to [Castellan et al. 2016].

6.1.1 Symmetry on Event Structures and Games. First of all, we recall symmetry [Winskel 2007].

Definition 6.1. A symmetry on an event structure \( E \) is a set \( \equiv_E \) comprising bijections \( \theta : x \equiv y \) where \( x, y \in \mathcal{C}(E) \) are configurations (we write \( \theta : x \equiv y \) if \( \theta \in \equiv_E \)) satisfying:

- Groupoid. For any \( x \in \mathcal{C}(E) \), \( \text{id}_x \in \equiv_E \); and \( \equiv_E \) is closed under inverse and composition.
- Restriction. For any \( \theta : x \equiv y \) and \( x' \subseteq x \) such that \( x' \in \mathcal{C}(E) \), there exists a (necessarily unique) \( \theta' \subseteq \theta \) such that \( \theta' : x' \equiv_y y' \);
- Expansion. For any \( \theta : x \equiv y \) and \( x \subseteq x' \in \mathcal{C}(E) \), there exists a (not necessarily unique) \( \theta \subseteq \theta' \) such that \( \theta' : x' \equiv_y y' \).

One may regard \( \equiv_E \) as a sort of proof-relevant equivalence relation – we will write simply \( x \equiv_E y \) for the corresponding equivalence relation. The last two conditions amount to \( \equiv_E \) being a history-preserving bisimulation. We refer to elements of \( \equiv_E \) as symmetries. It follows from “restriction” that symmetries are order-isomorphisms (with configurations ordered by \( \leq_E \)). Two events \( e_1, e_2 \in E \) are symmetric (written \( e_1 \equiv_E e_2 \)) iff \( (e_1, e_2) \in \theta \in \equiv_E \) for some \( \theta \); or equivalently if \( [e_1] \equiv_E [e_2] \).

We can now define quantum games with symmetry, or \(~\)-games.

Definition 6.2. A \(~\)-game comprises \((A, \mathcal{H}_A, \equiv_A, \equiv^+_A, \equiv^-_A)\) where \((A, \mathcal{H}_A)\) is a quantum game, with three symmetries such that \( \equiv^+_A, \equiv^-_A \subseteq \equiv_A \), subject to additional conditions [Castellan et al. 2016].

Finally, we require that if \( a \equiv_A a' \), then \( \mathcal{H}_A(a) = \mathcal{H}_A(a') \).

Any \( \theta : x \equiv_A y \) induces a unitary in \( \text{Hilb} \) between \( \mathcal{H}(x) \) and \( \mathcal{H}(y) \) obtained by the action of \( \theta \) on the tensors \( \mathcal{H}(x) = \bigotimes_{a \in x} \mathcal{H}(a) \) and \( \mathcal{H}(y) = \bigotimes_{a \in y} \mathcal{H}(a) \); we write \( \mathcal{H}(\theta) : \mathcal{H}(x) \equiv \mathcal{H}(y) \).

Constructions on quantum games extend: we have have \( \equiv_{A^\perp} = \equiv_A, \equiv^+_{A^\perp} = \equiv^+_A \) and \( \equiv^-_{A^\perp} = \equiv^-_A \). Likewise, \( \equiv_{A\|B} \) comprises \( \theta_A \| \theta_B : x_A \| x_B \equiv y_A \| y_B \) such that \( \theta_A : x_A \equiv y_A \) and \( \theta_B : x_B \equiv y_B \).

We introduce now the key operation that introduces symmetry in games interpreting types.

Definition 6.3. Let \( N \) be a negative \(~\)-game, meaning that all minimal events in \( N \) are negative. Then, its \( \text{bang} \) \text{!} \( N \) has event structure the infinitary parallel composition \( \|_{\|} \) \( N \) (its events are pairs \((i, a)\) where \( a \in N \) and \( i \in \mathbb{N} \) is the copy index); and quantum valuation \( \mathcal{H}_N((i, a)) = \mathcal{H}_N(a) \). Finally, for a bijection between configurations \( \theta : \| \) \( x \equiv \| y \) \( j \), we have

- \( \theta \in \equiv_{\|} \) iff there is a permutation \( \pi \) on natural numbers, and for each \( i \in \mathbb{N} \) a symmetry \( \theta_i : x_i \equiv_{\|} y_{\pi(i)} \), such that for all \((i, a) \in \|\) \( x \) \( i \) we have \( \theta(i, a) = (\pi(i), \theta_i(a)) \).
where the quantum valuation is further required to be invariant under symmetry.

\[ \theta \in \Xi^+_N \text{ iff for all } i \in \mathbb{N} \text{ there is } \theta_i : x_i \cong^y y_{x(i)}, \text{ and such that for all } (i, a) \in || x_i \text{ we have } \theta(i, a) = (x(i), \theta_i(a)); \]

\[ \theta \in \Xi^-_N \text{ iff for all } i \in \mathbb{N} \text{ there is } \theta_i : x_i \cong^y y_{x(i)} \text{ s.t. for all } (i, a) \in || x_i \text{ we have } \theta(i, a) = (i, \theta_i(a)). \]

This illustrates the meaning of the three symmetries: besides the global \( \equiv_A, \cong^+_A, \text{ and } \equiv^-_A \) keep track of which player has changed their copy indices. As detailed in [Castellan et al. 2016], this information is crucial in proving that weak isomorphism (Definition 6.5) is preserved by composition.

6.1.3 Simulation Equivalence.

We can now define our quantum strategies with symmetry.

Definition 6.4. A \( \sim \)-strategy on \( A \) is a quantum strategy \( \sigma : S \to A \), together with a symmetry \( \equiv_S \) on \( S \), first subject to the conditions from [Castellan et al. 2016]:

- **Symmetry-preservation.** If \( \theta : x \cong_S y \), then \( \sigma \theta = \{ (\sigma s_1, \sigma s_2) \mid (s_1, s_2) \in \theta \} : \sigma x \cong_A \sigma y; \)

- **Strong-receptivity.** If \( \theta : x \cong_S y \), if \( \sigma \theta \cap \{ (a_1^-, a_2^+) \} : x \cup \{ a_1 \} \cong_A y \cup \{ a_2 \}, \) then \( \theta \cup \{ (s_1, s_2) \} : x \cup \{ s_1 \} \cong_A y \cup \{ s_2 \} \) where \( \sigma s_1 = a_1 \) and \( \sigma s_2 = a_2 \) come from receptivity;

- **Thin.** If \( x \in \mathcal{C}(S), \) if \( id_x \equiv^+ \theta \in \equiv_S, \) then \( \theta = id_y \) for some \( y \in \mathcal{C}(S); \)

where, additionally, for any \( \theta : x \cong_S y \) we have \( H(\sigma \theta)(Q_\sigma(x)) = Q_\sigma(y). \)

For \( \sim \)-strategies \( \sigma \) : \( S \to A \) \( B \), the last condition amounts to the fact that for all \( \theta : x \cong_S y \) with \( \sigma \theta = \theta_A \parallel \theta_B \) (writing \( \theta_A : x_A \equiv_A y_A \) and \( \theta_B : x_B \equiv_B y_B \)), the diagram on the right commutes in CPM. For \( \sim \)-strategies \( \sigma : S \to A \) \( B \) and \( \tau : T \to B \) \( C \), symmetries \( \theta : x_T \circ x_S \equiv_{T \circ S} y_T \circ y_S \) exactly correspond to pairs comprising \( \theta_T : x_T \equiv_T y_T \) and \( \theta_S : x_S \equiv_S y_S \) matching on \( B, \) i.e. such that \( \sigma \theta_S = \theta_A \parallel \theta_B \) and \( \tau \theta_T = \theta_B \parallel \theta_C; \) we write again \( \theta_T \circ \theta_S : x_T \circ x_S \equiv_{T \circ S} y_T \circ y_S \) to emphasize the correspondence. From the above, preservation of quantum state under symmetry is preserved by composition. Thanks to symmetry, more strategies can be considered equivalent: weak isomorphism, more permissive than than isomorphism, relates strategies via bijections intuitively allowing different choices of copy indices for Player moves.

Definition 6.5. Two \( \sim \)-strategies \( \sigma : S \to A \) and \( \sigma' : S' \to A \) are **weakly isomorphic** iff there is a bijection \( \varphi : S \equiv S', \) preserving and reflecting all structure (including symmetry and quantum valuations), and such that for all \( x \in \mathcal{C}(S), \) we have \( \{ (\sigma s, \sigma'(\varphi s)) \mid s \in x \} \in \equiv_A. \)

We obtain a compact closed category \( \sim \-QCG \) with objects \( \sim \-games, \) and morphisms \( \sim \)-strategies up to weak isomorphism. It admits two (dual) subcategories \( \sim \-QCG_+ \) and \( \sim \-QCG_-, \) with objects respectively positive and negative games, and morphisms negative strategies. As expected:

**Proposition 6.6.** The construction of Definition 6.3 extends to a linear exponential comonad [Hyland and Schalk 2003] on the symmetric monoidal category \( \sim \-QCG_-, \).

This also holds on the category \( \sim \-QCG^-_\text{alt} \) whose objects are additionally alternating.

For instance, the contraction \( \delta_A : !A \to !A \parallel !A \), following a bijection \( N \equiv N + N, \) should be associative: the two natural ways to obtain \( !A \to !A \parallel !A \parallel !A \) should coincide – but that only holds up to symmetry. The significant challenge of showing that composition preserves weak isomorphism is addressed in [Castellan et al. 2016] and undisturbed by quantum annotations.

6.1.4 Simulation Equivalence. For simulation equivalence, we cannot anymore ask maps to preserve labeling, since they need to relate strategies playing distinct but symmetric events.

**Definition 6.7.** A **simulation map** from \( \sim \)-strategy \( \sigma : S \to A \) to \( \sigma' : S' \to A \) is \( f : S \to S' \), satisfying axioms (1) and (2) of Definition 3.2, which is rigid and satisfies:

\[ f(x \cup \{ s \}) = f(x) \cup \{ f(s) \}, \]

\[ f(x \cup \{ s \}) = f(x) \cup \{ f(s) \}. \]

\[
\begin{align*}
\llbracket \Gamma, x : !A \vdash x : A \rrbracket & = d_{[A]} \otimes (w_{[\Gamma]} \otimes \! [A]) \\
\llbracket !\Gamma, A \vdash v : ! (A \rightarrow B) \rrbracket & = (\llbracket \Gamma \rrbracket) \otimes v \otimes (\llbracket \Gamma \rrbracket \otimes w_{[A]}) \\
\llbracket \Delta, \Gamma \vdash \text{letr} \ f^{A \rightarrow B} x^A = t \text{ in } u : C \rrbracket & = \llbracket u \rrbracket \otimes (\llbracket A \rrbracket \otimes \llbracket \Gamma \rrbracket \otimes Y(\lambda_{[A]}(\llbracket f \rrbracket)) \otimes (\llbracket A \rrbracket \otimes c_T))
\end{align*}
\]

Fig. 20. Interpretation of the non-linear rules

(a) For all \(x \in \mathcal{E}(S), \theta_x = (\sigma, \delta^\prime(f x)) \mid s \in x \in \mathcal{E}_A^+; \)
(b) For all \(y \in \mathcal{E}(S'), \sum_{f x = y} \theta_x(Q_\sigma(x)) \leq_\mathcal{L} Q_\sigma(y).\)

We write \(\sigma \preceq \sigma'\) if there is such a simulation map, and \(\sigma \approx \sigma'\) for the equivalence.

We do not require \(\sigma' \circ f = \sigma\), relaxing it to a commutation up to positive symmetry on \(A\) – this use of the positive symmetry is crucial to ensure that \(\preceq\) is stable under composition. The analogue of Proposition 4.6 still holds: the full subcategory of \(\sim\)-\(\text{QCG}\) having as objects those with one positive event and morphisms up to simulation equivalence is equivalent to \(\text{CPM}\)\(^{\text{\scriptsize \text{\textsuperscript{ex}}}}\).

6.2 Interpretation of the Full Quantum \(\lambda\)-Calculus

We call \(\sim\)-arenas those \(\sim\)-games whose underlying game is an arena and where symmetries act as the identity on minimal events – the subcategory \(\sim\)-\(\text{QA}\) has \(\sim\)-arenas as objects, and negative \(\sim\)-strategies as morphisms. The structure in Section 5 extends transparently with symmetry, and yields the same adjunction as in Proposition 5.7. The interpretation of the affine fragment works in \(\sim\)-\(\text{QA}\) in exactly the same way, the basic arenas and strategies involved having trivial symmetry.

6.2.1 Exponential on Positive Games. To transport it to positive games, we use the following.

**Lemma 6.8.** The SMC \((\sim\text{-\text{QCG}}^\text{alt}, [], !)\) is equivalent to \((\sim\text{-\text{QA}}^1, \otimes)\), the full subcategory of \(\sim\text{-\text{QA}}\) whose objects are positive \(\sim\)-arena with one minimal event \(\bullet^+\), with trivial Hilbert space \(\mathcal{H}(\bullet) = I\).

Indeed, \(\sigma : !A \rightarrow !B\) in \(\sim\text{-\text{QA}}^1\) waits for \(\bullet^+\), after which it plays \(\bullet^+\) with coefficient 1. The remaining behaviour yields \(\sigma' : A \rightarrow B\) negative. This correspondence yields the equivalence claimed. In particular, \(!\) transports to a linear exponential comonad on \((\sim\text{-\text{QA}}^1, \otimes)\), defined on objects as \(!((A) = ![A], whose components follow from Lemma 6.8.

6.2.2 Recursion-Free Fragment. At this point we can interpret the recursion-free fragment. The crucial observation is that \(!\) is restricted to types of the form \(A \rightarrow B\), having as required one unique minimal event; so we may define \([!(A \rightarrow B)] = ![A \rightarrow B].\) This provides us with morphisms \(\delta_{[A \rightarrow B]} : ![!(A \rightarrow B)] \rightarrow ![A \rightarrow B] \otimes ![A \rightarrow B]\) in \(\sim\text{-\text{QA}}^1\) satisfying naturality and comonoid laws up to symmetry, and allowing us to refine the clauses of Figures 16 to deal with sharing of the non-linear context. Furthermore, the first two lines of Figure 20 show the interpretation of the two new rules for non-linear variables and promotions, which make use respectively of \(v : ![\Gamma] \rightarrow ![\Gamma] \otimes ![\Gamma]\) obtained through the comultiplication \(\delta_A : ![A] \rightarrow ![A] \otimes ![A]\) and monoidality of \(!\); and the co-unit \(d_A : ![A] \rightarrow A\), natural in \(\sim\text{-\text{QA}}^1\) – note that the promotion rule is restricted to values, which permits the use of the functorial action of \(!\).

6.2.3 Recursion. Finally, we interpret recursion. The methodology is completely standard. First we define an order on \(\sim\)-strategies on a \(\sim\)-game \(A\) as follows.

**Definition 6.9.** Given \(\sigma : S \rightarrow A, \sigma' : S' \rightarrow A\) two \(\sim\)-strategies, we write \(\sigma \preceq \sigma'\) iff (1) \(S \subseteq S'\), (2) if \(x \equiv_S y\) and \(x \in \mathcal{E}(S)\) implies \(y \in \mathcal{E}(S)\), and (3) all components of \(\sigma\) and \(\sigma'\) coincide on \(S\).
We are grateful to Frank Roumen for numerous discussions on the mathematics of quantum computation. We believe that the present contribution is a significant step in this direction.

We define the typing rule as

$$\frac{}{\Gamma \vdash n : \tau} \text{letrec}_n \ f : A \rightarrow B \ x^A = t \ \text{in} \ u$$

where

$$\lambda^1_A(\bot)$$

and

$$\Gamma \vdash n : \tau$$

Furthermore, it follows from standard techniques that just as in the affine quantum λ-calculus, there is a bound to the possible reduction length in the bounded language. Hence the proof of Theorem 5.11 transports. It remains to extend this to the full calculus.

Relying essentially on the definition of recursion and the categorical properties of the exponential, it follows that Lemma 5.10 still holds for the interpretation of the full quantum λ-calculus.

6.2.4 Adequacy. We exploit the construction of recursion as a supremum of finite approximations. We define the bounded recursion operator letrec$_n$ $f : A \rightarrow B$ $x^A = t \ \text{in} \ u$ for all $n \in \mathbb{N}$ with same typing rule as letrec $f : A \rightarrow B$ $x^A = t \ \text{in} \ u$, but reductions (ignoring the quantum store):

$$\text{letrec}_0 \ f : A \rightarrow B \ x^A = t \ \text{in} \ u \rightarrow u[\lambda x^A. \bot / f]$$

$$\text{letrec}_{n+1} \ f : A \rightarrow B \ x^A = t \ \text{in} \ u \rightarrow u[\lambda x^A. \text{letrec}_n \ f : A \rightarrow B \ x^A = t \ \text{in} \ u / f]$$

The language where all recursion is bounded can be interpreted in ~$\mathbb{QA}$ easily, simply by setting

$$\left[\Lambda, x^A : C\right] = [u] \ominus ([\Lambda] \otimes [\pi]) \otimes Y_n(\lambda^1_A(\bot)) \ominus ([\Lambda] \otimes c_T)$$

which satisfies the invariance lemma. Furthermore, it follows from standard techniques that just as in the affine quantum $\lambda$-calculus, there is a bound to the possible reduction length in the bounded language. Hence the proof of Theorem 5.11 transports. It remains to extend this to the full calculus.

**Theorem 6.10.** Let $t : 1$ be a term. Then, $t \vdash_p$ iff $[t] \vdash_p$ for all $p \in [0, 1]$.

**Proof.** For all $n \in \mathbb{N}$, $t \uparrow n$ is $t$ where recursion is replaced by that bounded by $n$. For each $n \in \mathbb{N}$, we write $t \uparrow n$ $p_n$. Recall that computing $t \vdash_p$ involves a potentially infinite sum; each partial sum approximating it can be replicated in $t \uparrow n$ for sufficiently large $n$. It follows that $p = \sup_{n \in \mathbb{N}} p_n$.

On the other hand, by definition of $Y$ and continuity of operations on strategies, we have $[t] = \sup_{n \in \mathbb{N}} [t \uparrow n]$. Writing $[t] \vdash_{p'}$ and $[t \uparrow n] \vdash_{p'_n}$, it is direct to deduce that $p' = \sup_{n \in \mathbb{N}} p'_n$ as well. But by adequacy for the bounded calculus, $p_n = p'_n$ for all $n \in \mathbb{N}$ so $p = p'$ as well.

7 Conclusion

In contrast to the model of [Pagani et al. 2014], our monotone conditions ensure that quantum annotations remain bounded (see Proposition 4.3). Because of that and our being more intensional, our model also features less "junk" and is close to a definability result. However, the problem of building a fully abstract model for the quantum $\lambda$-calculus remains a challenging open problem. We believe that the present contribution is a significant step in this direction.

Another intriguing question is whether we can link our model with the recent operational account of the quantum $\lambda$-calculus, based on the Geometry of Interaction [Dal Lago et al. 2017]. Interestingly their Gol is parallel, like our parallel interpretation. A connection, if possible, would link the compositional aspects brought up by our model with their more operational description.

**Acknowledgments**

We are grateful to Frank Roumen for numerous discussions on the mathematics of quantum computation. We acknowledge support of the French LABEX MILYON (ANR-10-LABX-0070), the ERC Advanced Grant ECSYM, and the Collegium de Lyon.
REFERENCES


