Quantum Turing automata

Miklós Bartha
Department of Computer Science
Memorial University of Newfoundland
St. John's, NL, Canada, A1B 3X5
Email: bartha@mun.ca

***Abstract*—A denotational semantics of quantum Turing machines is defined in the strongly compact closed category of finite dimensional Hilbert spaces. Using the Moore-Penrose generalized inverse, a new additive trace is introduced on the restriction of this category to isometries, which trace is carried over to directed quantum Turing machines as monoidal automata. The resulting traced monoidal category is further transformed into the indexed monoidal algebra of undirected quantum Turing automata.**

I. Introduction

In recent years, following the endeavors of Abramsky and Coecke to express some of the basic quantum-mechanical concepts in an abstract axiomatic category theory setting, several models have been worked out to capture the semantics of quantum information protocols [1] and programming languages [11], [15], [23]. Concerning quantum hardware, an algebra of automata which include both classical and quantum entities has been studied in [12].

The objective of the present paper is to provide a denotational style semantics for quantum Turing machines as hardware devices. At the same time, the rigid topological layout of Turing machines as a linear array of tape cells is replaced by a flexible graph structure, giving rise to the concept of Turing automata and graph machines as introduced in [6]. By denotational semantics we mean that the changing of the tape contents caused by the entire computation process is specified directly as a linear operator, rather than just one step of this process.

Our presentation will use the language of [1], [16], but it will be specific to the concrete strongly compact closed category $(\mathbf{FdHilb}, \otimes)$ of finite dimensional Hilbert spaces at this time. One can actually read Section 4 separately as an interesting study in linear algebra, introducing a novel application of the Moore-Penrose generalized inverse of range-Hermitian operators by taking their Schur complement in certain block matrix operators. This is the main technical contribution of the paper. We believe, however, that the category theory contributions are much more interesting and relevant. All of these results are around the well-known Geometry of Interaction (GoI) concept introduced originally by Girard [13] in the late 1980's as an interpretation of linear logic. The ideas, however, originate from and are directly related to a yet earlier work [2] by the author on the axiomatization of flowchart schemes, where the traced monoidal category axioms first appeared in an algebraic context. Our category theory contributions are as follows.

(i) We introduce a total trace on the monoidal subcategory of $(\mathbf{FdHilb}, \oplus)$ defined by isometries, which has previously been sought by others [14], [21].

(ii) We explain the role of the Int construction for traced monoidal categories [16] in turning a computation process bidirectional or reversible.

(iii) We capture the phenomenon in (ii) above by our own concept "indexed monoidal algebra" [7], an equivalent formalism for compact closed categories.

Due to space limitations we have to assume familiarity with some advanced concepts in category theory, namely traced monoidal categories [16], compact closed categories [18], and the Int construction that links these two types of symmetric monoidal categories [20] to each other. For brevity, by a monoidal category we shall mean a symmetric monoidal one throughout the paper.

II. Monoidal categories and indexed monoidal algebras

The following definition of (strict) traced monoidal categories uses the terminology of [16]. Trace (called feedback in [2]) in a monoidal category $\mathcal{C}$ with unit object I, tensor $\otimes$, and symmetries $c_{A,B} : A \otimes B \to B \otimes A$ is introduced as a left trace, i.e., an operation $\mathcal{C}(U \otimes A, U \otimes B) \to \mathcal{C}(A, B)$.

Definition 2.1. A *trace* for a monoidal category $\mathcal{C}$ is a natural family of functions

$$Tr^U_{A,B} : \mathcal{C}(U \otimes A, U \otimes B) \to \mathcal{C}(A, B)$$

satisfying the following three axioms:
vanishing:

$$Tr^I_{A,B}(f) = f \ , \quad Tr^{U \otimes V}_{A,B}(g) = Tr^V_{A,B}(Tr^U_{V \otimes A, V \otimes B}(g));$$

superposing:

$$Tr^U_{A,B}(f) \otimes g = Tr^U_{A \otimes C, B \otimes D}(f \otimes g), \text{ where } g : C \to D;$$

yanking:

$$Tr^U_{U,U}(c_{U,U}) = 1_U.$$

Naturality of trace is meant in all three variables A, B, U. The word *sliding* is used as a synonym for (di-)naturality in U. When using the term *feedback* for trace, the notation Tr changes to $\uparrow$ or $\Uparrow$, and we simply write Tr^U ($\uparrow^U$, $\Uparrow^U$) for $Tr^U_{A,B}$ whenever A and B are understood from the context. One further axiom will be of interest for us in Section 4.

derived composition:

$$f \circ g = Tr^B_{A,C}(c_{B,A} \circ (f \otimes g)) \text{ for } f : A \to B,\ g : B \to C.$$

It is known, cf. [2, Axiom X3], that this identity is a consequence of the traced monoidal category axioms. Moreover, in the presence of derived composition, it is sufficient to impose/check naturality with respect to permutations only.

Notice that we write composition of morphisms ($\circ$) in a left-to-right manner, avoiding the use of ";", which some may find more appropriate. Accordingly, when working in a Hilbert space $\mathcal{H}$, we shall think of a vector $v \in \mathcal{H}$ as a "row vector", that is, a morphism $v : \mathbb{C} \to \mathcal{H}$. Consequently, we apply an operator (matrix) T on v as vT, and *not* Tv. A "column vector" ($v\downarrow$) is a morphism $\mathcal{H} \to \mathbb{C}$, which is naturally isomorphic to a (row) vector in the dual space $\mathcal{H}^*$. We shall use the symbols I and 0 as "generic" identity (respectively, zero) operators, provided that the underlying Hilbert space is understood from the context. As a further technical simplification we shall be working with the strict monoidal formalism, even though the monoidal category of Hilbert spaces with the usual tensor product is not strict.

Definition 2.2. A monoidal category $\mathcal{C}$ is *compact closed* (CC, for short) if every object A has a left adjoint A^* in the sense that there exist morphisms $d_A : I \to A \otimes A^*$ (the unit map) and $e_A : A^* \otimes A \to I$ (the counit map) for which the two composites below result in the identity morphisms 1_A and 1_{A^*}, respectively.

$$\begin{aligned}
A &= I \otimes A \to_{d_A \otimes 1_A} (A \otimes A^*) \otimes A \\
&= A \otimes (A^* \otimes A) \to_{1_A \otimes e_A} A \otimes I = A, \\
A^* &= A^* \otimes I \to_{1_{A^*} \otimes d_A} A^* \otimes (A \otimes A^*) \\
&= (A^* \otimes A) \otimes A^* \to_{e_A \otimes 1_{A^*}} I \otimes A^* = A^*.
\end{aligned}$$

Category $\mathcal{C}$ is *self-dual compact closed* (SDCC, for short) if $A = A^*$ for each object A. The category **SDCC** has as objects all SDCC categories, and as morphisms monoidal functors preserving the given self-adjunctions. As it is well-known, every CC category admits a so called *canonical trace* [16] defined by the formula

$$Tr^U_{A,B} f = (d_{U^*} \otimes 1_A) \circ (1_{U^*} \otimes f) \circ (e_U \otimes 1_B).$$

See Fig. 1.

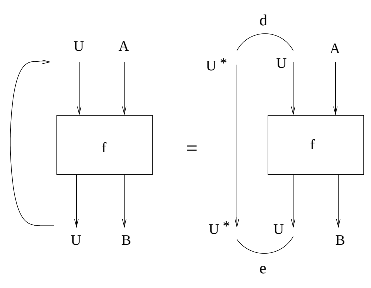

Fig. 1. Canonical trace

On the analogy of enriched categories [19] indexed monoidal algebras have been introduced recently in [6]. An *indexed monoidal algebra* (or half-category) $\mathcal{M}$ consists of objects $A, B, \ldots$, morphisms $f, g, \ldots$, and an operation *rank*, which assigns to each morphism f an object A. We write $f : A$ to indicate the rank of f. There is an associative binary operation $\otimes$ (tensor) on objects and a unit object I, defining a monoid structure M. On morphisms, the following operations are defined.

– A binary operation, also called *tensor*, which assigns to each pair of morphisms $f : A$ and $g : B$ a morphism $f \otimes g : A \otimes B$.
– A unary operation *trace*, by which every morphism $f : A \otimes A \otimes B$ is assigned a morphism $\updownarrow_{A,B} f : B$. We shall write $\updownarrow_A f$ if B is understood.
– For each object A, an *identity* morphism $\mathbf{1}_A : A \otimes A$.

Intuitively, a ("half"-)morphism $f : A$ stands for a real morphism $f : I \to A$ in a corresponding hypothetical SDCC category $\mathcal{C}$. Tensor in $\mathcal{M}$ is essentially $\otimes$ in $\mathcal{C}$, and $\updownarrow_A f$ for $f : A \otimes A \otimes B$ captures the canonical trace of the morphism $f_A : A \to A \otimes B$ in $\mathcal{C}$ that corresponds to f by compact closure. There is also an indexing mechanism in $\mathcal{M}$, which employs *permutation symbols* as a key instrument. Using Mac Lane's coherence theorem for monoidal categories [20], a permutation symbol $\rho : A \Rightarrow B$ is a free symbolic representation of a permutation $A \to B$ in $\mathcal{C}$, independently of any concrete $\mathcal{C}$ sharing the given object structure M with $\mathcal{M}$.

Permutation symbols do not form a category over the objects of $\mathcal{M}$, though. They do form a monoidal category in which the objects are *object terms* (words) over $\mathcal{M}$'s objects as *object variables*. Composition and tensor in this free category will be denoted by $\bullet$ and $\oslash$ in the axioms I1-I9 below. Since each object term evaluates to an object according to the given monoid M, every permutation symbol $\rho : A \Rightarrow B$ has a unique canonical interpretation as a permutation $A \to B$ in any monoidal category having M as its object structure. In our algebraic language, each permutation symbol $\rho : A \Rightarrow B$ serves as a unary operation in $\mathcal{M}$, which takes a morphism $f : A$ to a morphism $f \cdot \rho : B$. Two permutation symbols $\rho, \rho' : A \Rightarrow B$ are said to be *equivalent*, $\rho \equiv \rho'$, if they denote the same permutation in every monoidal category having the object structure M.

Composition ($\circ$) is the following derived operation in $\mathcal{M}$.

– For $f : A \otimes B$ and $g : B \otimes C$,

$$f \circ g = \updownarrow_B ((f \otimes g) \cdot (c_{A,BB} \oslash 1_C)).$$

See Fig. 2.

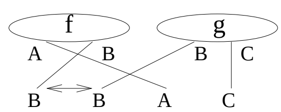

Fig. 2. Composition as a derived operation

Operations in $\mathcal{M}$ are subject to the following nine equational axioms, which postulate that the resulting indexed monoidal algebra (IMA, for short) be indeed equivalent to an SDCC category.

I1. *Functoriality of indexing*

$$f \cdot (\rho_1 \bullet \rho_2) = (f \cdot \rho_1) \cdot \rho_2$$

for $f : A$ and composable $\rho_1 : A \Rightarrow B$, $\rho_2 : B \Rightarrow C$;

$$f \cdot 1_A = f \quad \text{for } f : A.$$

I2. *Naturality of indexing*

$$(f \otimes g) \cdot (\rho_1 \oslash \rho_2) = f \cdot \rho_1 \otimes g \cdot \rho_2$$

for $f : A$, $g : B$, $\rho_1 : A \Rightarrow C$, $\rho_2 : B \Rightarrow D$;

$$(\updownarrow_A f) \cdot \rho = \updownarrow_A (f \cdot (1_{AA} \oslash \rho))$$

for $f : A \otimes A \otimes B$, $\rho : B \Rightarrow C$.

I3. *Coherence*

$$f \cdot \rho_1 = f \cdot \rho_2 \quad \text{for } f : A,\ \rho_i : A \Rightarrow B,\ \text{whenever } \rho_1 \equiv \rho_2.$$

I4. *Associativity and symmetry of tensor*

$$(f \otimes g) \otimes h = f \otimes (g \otimes h) \text{ for } f : A, g : B, h : C;$$

$$f \otimes g = (g \otimes f) \cdot c_{A,B} \text{ for } f : A, g : B.$$

I5. *Right identity*

$$f \circ \mathbf{1}_B = f \text{ and } f \otimes \mathbf{1}_I = f \text{ for } f : A \to B.$$

I6. *Symmetry of identity*

$$\mathbf{1}_A \cdot c_{A,A} = \mathbf{1}_A.$$

I7. *Vanishing*

$$\updownarrow_I f = f \text{ for } f : A;$$

$$\updownarrow_{A \otimes B} f = \updownarrow_B (\updownarrow_A f \cdot (1_A \oslash c_{B,A} \oslash 1_{BC}))$$

for $f : A \otimes B \otimes A \otimes B \otimes C$.

I8. *Superposing*

$$\updownarrow_A (f \otimes g) = \updownarrow_A f \otimes g \text{ for } f : A \otimes A \otimes B, g : C.$$

I9. *Trace swapping*

$$\updownarrow_B (\updownarrow_A f) = \updownarrow_A (\updownarrow_B (f \cdot (c_{AA,BB} \oslash 1_C)))$$

for $f : A \otimes A \otimes B \otimes B \otimes C$.

Analogously to a functor, an *indexed monoidal homomorphism* $h : \mathcal{M} \to \mathcal{M}'$ between IMA's $\mathcal{M}$ and $\mathcal{M}'$ consists of a pair of functions. The object function assigns to each object A in $\mathcal{M}$ an object hA, so that h preserves the monoid structure, and the morphism function assigns to each morphism $f : A$ a morphism $hf : hA$ in such a way that h defines a homomorphism in the algebraic sense.

Let **IMA** denote the category of indexed monoidal algebras with indexed algebraic homomorphisms between them. The following result was proved in [6].

Theorem 2.1. *The categories* **SDCC** *and* **IMA** *are equivalent.*

Given an arbitrary traced monoidal category $\mathcal{C}$, one can turn it functorially into an IMA $Alg(\mathcal{C})$ by the help of the Int construction [16]. The trick is to simply restrict the CC category $Int(\mathcal{C})$ to its self-dual objects (A, A), and then use Theorem 2.1 to obtain an equivalent IMA. See Theorem 5.1 below for details.

III. Monoidal vs Turing automata

Circuits and automata over an arbitrary monoidal category M have been studied in [3], [4], [5], [17]. It was shown that the collection of such machines has the structure of a monoidal category equipped with a natural feedback operation, which satisfies the traced monoidal axioms, except for yanking. Moreover, sliding holds in a weak sense, for isomorphisms only.

Let A and B be objects in M. An M*-automaton* (circuit) $A \to B$ is a pair (U, α), where U is a further object and $\alpha : U \otimes A \to U \otimes B$ is a morphism in M. If, for example, $M = (\mathbf{Set}, \times)$, then the pair (U, α) represents a deterministic Mealy automaton with states U, input A, and output B. The structure of M-automata/circuits has been described as a monoidal category $Circ(M)$ with feedback in [17]. This category was also shown to be freely generated by M.

In this paper we take a different approach to the study of monoidal automata. We follow the method of [6] with the aim of constructing a *traced* monoidal category as an adequate semantical structure for these automata. One must not confuse this type of semantics with the meaning normally associated with the category $Circ(M)$ above, as they have seemingly very little in common. A traced monoidal category indicates a *delay-free* semantics, as opposed to the step-by-step *delayed* semantics suggested by $Circ(M)$. Arguing at an intuitive level, the difference is the following. In a delayed model, the "combinational logic" $\alpha : U \otimes A \to U \otimes B$ describes one primitive step of the automaton, and the stepwise behavior is derived naturally as a kind of operational semantics in terms of sequences. In contrast, a delay-free model is like an asynchronous automaton (e.g. a flip-flop constructed from two NAND gates), which must first stabilize on a given input, keeping it steady over an indefinite number of steps, before the next input can even be considered.

Even though the analogy above is quite appropriate, the category that we are going to construct is not meant to be the quotient of $Circ(M)$ by the yanking identity, so as to turn it into a traced monoidal category in the straightforward manner. Rather, we define a brand new tensor and feedback (trace) on our M-automata, which are analogous to the basic operations in iteration theories [10]. Regarding the base category M, we shall assume an additional, so called additive tensor $\oplus$, so that $\otimes$ distributes over $\oplus$. These two tensors will then be "mixed and matched" in the definition of tensor ($\boxtimes$) for M-automata, providing them with an intrinsic Turing machine behavior.

The "prototype" of this construction, resulting in the indexed monoidal algebra of conventional Turing automata, has been elaborated in [7] using $M = (\mathbf{Rel}, \times, +)$ as the base category. This category was ideal as a template for the kind of construction we have in mind, since it is biproduct by $+$ and self-dual compact closed according to $\times$. Below we present the quantum counterpart of this construction, working in the biproduct strongly compact closed category [1] of finite dimensional Hilbert spaces ($\mathbf{FdHilb}, \otimes, \oplus$). More precisely, the category M above will be the restriction of **FdHilb** to isometries as morphisms, which subcategory is no longer compact closed or biproduct. We shall only use the inner product feature of **FdHilb**, completeness of the metric space induced is irrelevant.

IV. Directed quantum Turing automata

In this section we present the construction outlined above, to obtain a strange asymmetric model which does not yet qualify as a recognizable quantum computing device in its own right. The model represents a Turing machine in which cells are interconnected in a directed way, so that the control (tape head) always moves along interconnections in the given fixed direction, should it be left or right. In other words, direction is incorporated in the scheme-like graphical syntax, rather than the semantics. We use this model only as a stepping stone towards our real objective, the (undirected) quantum Turing automaton described in Section 5.

Definition 4.1. A *directed quantum Turing automaton* is a quadruple
$$T = (\mathcal{H}, \mathcal{K}, \mathcal{L}, \tau),$$
where $\mathcal{H}$, $\mathcal{K}$, and $\mathcal{L}$ are finite dimensional Hilbert spaces over the complex field $\mathbb{C}$, and $\tau : \mathcal{H} \otimes \mathcal{K} \to \mathcal{H} \otimes \mathcal{L}$ is an isometry in **FdHilb**.

Recall that an *isometry* between Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$ is a linear map $\sigma : \mathcal{H}_1 \to \mathcal{H}_2$ such that $\sigma \circ \sigma^\dagger = I$, where $\sigma^\dagger$ is the (Hilbert space) *adjoint* of σ. Following the notation of general monoidal automata we write $T : \mathcal{K} \to \mathcal{L}$, and call the isometry τ the *transition operator* of T. Thus, T is the monoidal automaton $(\mathcal{H}, \tau) : \mathcal{K} \to \mathcal{L}$. Sometimes we simply identify T with τ, provided that the other parameters of T are understood from the context.

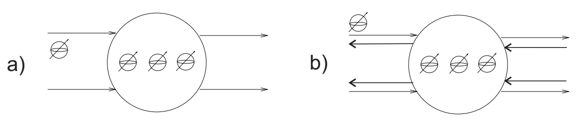

Fig. 3. Two simple DQTA

The reader can obtain an intuitive understanding of the automaton T from Fig. 3a. The state space $\mathcal{H}$ is represented by a finite number of qubits, while the control is a moving particle that moves from one of the input interfaces (space $\mathcal{K}$) to one of the output ones (space $\mathcal{L}$). It can only move in the input $\to$ output direction, as specified by the operator τ. The number of input and output interfaces is finite. The control itself does not carry any information, it is just moving around and changes the state of T. In comparison with conventional Turing machines, the state of T is the tape contents of the corresponding Turing machine, and the current state of the Turing machine is just an interface identifier for T. For example, one can consider the DQTA in Fig. 3b as one tape cell of a Turing machine TM having 2^3 symbols in its tape alphabet and only 2 states (2 left-moving and 2 right-moving interfaces, both input and output). Correspondingly, $\mathcal{H}$ is 8-dimensional, while the dimension of both $\mathcal{K}$ and $\mathcal{L}$ is 4. In motion, if the control particle of T resides on the input interface labeled (L, i) ((R, i)), then TM is in state i moving to the left (respectively, right). The point is, however, that the automaton T need not represent just one cell, it could stand for any finite segment of a Turing machine, in fact a Turing graph machine in the sense of [6]. In our concrete example, a segment of TM with n tape cells would have $3n$ qubits inside the circle of Fig. 3b, but still the same $4 + 4$ interfaces.

An *isometric isomorphism* $\sigma : \mathcal{H}_1 \to \mathcal{H}_2$ (*unitary map*, if $\mathcal{H}_1 = \mathcal{H}_2$) is a linear operator such that both σ and $\sigma^\dagger$ are isometries. Two automata $T_i : (\mathcal{H}_i, \tau_i) : \mathcal{K} \to \mathcal{L}$, $i = 1, 2$, are *isomorphic*, notation $T_1 \cong T_2$, if there exists an isometric isomorphism $\sigma : \mathcal{H}_1 \to \mathcal{H}_2$ for which
$$\tau_2 = (\sigma^\dagger \otimes I_{\mathcal{K}}) \circ \tau_1 \circ (\sigma \otimes I_{\mathcal{L}}).$$
For simplicity, though, we shall work with representatives, rather than equivalence classes of automata.

Turing automata can be composed by the standard *cascade product* of monoidal automata, cf. [4], [5], [17]. If $T_1 = (\mathcal{H}_1, \tau_1) : \mathcal{L} \to \mathcal{M}$ and $T_2 = (\mathcal{H}_2, \tau_2) : \mathcal{M} \to \mathcal{N}$ are directed quantum Turing automata (DQTA, for short), then
$$T_1 \circ T_2 = (\mathcal{H}_1 \otimes \mathcal{H}_2, \mathcal{L}, \mathcal{N}, \tau)$$
is the automaton whose transition operator τ is
$$(\pi_{\mathcal{H}_1,\mathcal{H}_2} \otimes I_{\mathcal{L}}) \circ (I_{\mathcal{H}_2} \otimes \tau_1) \circ (\pi_{\mathcal{H}_2,\mathcal{H}_1} \otimes I_{\mathcal{M}}) \circ (I_{\mathcal{H}_1} \otimes \tau_2),$$
where $\pi_{\mathcal{H},\mathcal{K}}$ is the symmetry $\mathcal{H} \otimes \mathcal{K} \to \mathcal{K} \otimes \mathcal{H}$ in ($\mathbf{FdHilb}, \otimes$). As known from [17], the cascade product of automata is compatible with isomorphism, so that it is well-defined on isomorphism classes of DQTA. The identity Turing automaton $1_{\mathcal{K}} : \mathcal{K} \to \mathcal{K}$ has the unit space $\mathbb{C}$ as its state space, and its transition operator is simply $I_{\mathcal{K}}$. The results in [17] imply that these data define a category **DQT** over finite dimensional Hilbert spaces as objects, in which the morphisms are isomorphism classes of DQTA.

Now let
$$T_1 = (\mathcal{H}_1, \tau_1) : \mathcal{K}_1 \to \mathcal{L}_1 \text{ and } T_2 = (\mathcal{H}_2, \tau_2) : \mathcal{K}_2 \to \mathcal{L}_2$$
be DQTA, and define $T_1 \boxtimes T_2$ to be the automaton over the state space $\mathcal{H}_1 \otimes \mathcal{H}_2$ whose transition operator
$$\tau = \tau_1 \boxtimes \tau_2 : (\mathcal{H}_1 \otimes \mathcal{H}_2) \otimes (\mathcal{K}_1 \oplus \mathcal{K}_2) \to (\mathcal{H}_1 \otimes \mathcal{H}_2) \otimes (\mathcal{L}_1 \oplus \mathcal{L}_2)$$
acts as follows: $\tau \simeq \sigma_1 \oplus \sigma_2$, where the morphisms
$$\sigma_i : (\mathcal{H}_1 \otimes \mathcal{H}_2) \otimes \mathcal{K}_i \to (\mathcal{H}_1 \otimes \mathcal{H}_2) \otimes \mathcal{L}_i, \quad i = 1, 2 \text{ are:}$$
$\sigma_1 = (\pi_{\mathcal{H}_1,\mathcal{H}_2} \otimes I_{\mathcal{K}_1}) \circ (I_{\mathcal{H}_2} \otimes \tau_1) \circ (\pi_{\mathcal{H}_2,\mathcal{H}_1} \otimes I_{\mathcal{L}_1})$, and $\sigma_2 = I_{\mathcal{H}_1} \otimes \tau_2$.

In the above equations, $\oplus$ denotes the orthogonal sum of Hilbert spaces. Intuitively, τ is the selective performance of *either* τ_1 *or* τ_2 on the tensor space $\mathcal{H}_1 \otimes \mathcal{H}_2$. The natural isomorphism $\simeq$ is *distributivity* in the sense of [1, Proposition 5.3], which is meaningful in all biproduct compact closed categories. It is clear that the operator $\tau_1 \boxtimes \tau_2$ is an isometry, so that the operation $\boxtimes$ is well-defined. We call this operation the *Turing tensor*. The Turing tensor is also associative, up to natural isomorphism, of course.

The symmetries $\mathcal{K} \boxtimes \mathcal{L} \to \mathcal{L} \boxtimes \mathcal{K}$ associated with $\boxtimes$ are the "single-state" Turing automata whose transition operator is the permutation

$$\kappa_{\mathcal{K},\mathcal{L}} = \begin{array}{c c} & \begin{array}{cc} \mathcal{L} & \mathcal{K} \end{array} \\ \begin{array}{c} \mathcal{K} \\ \mathcal{L} \end{array} & \left(\begin{array}{cc} 0 & I \\ I & 0 \end{array} \right) \end{array} : (\mathbb{C}\otimes)(\mathcal{K} \oplus \mathcal{L}) \to (\mathbb{C}\otimes)(\mathcal{L} \oplus \mathcal{K}).$$

Along the lines of [17] it is routine to check that $\boxtimes$ is also compatible with isomorphism of automata, and $(\mathbf{DQT}, \boxtimes)$ becomes a monoidal category in this way.

Our third basic operation on DQTA is feedback. Feedback follows the scheme of iteration in Conway matrix theories [10], using an appropriate star operation. Let $T : \mathcal{U} \oplus \mathcal{K} \to \mathcal{U} \oplus \mathcal{L}$ be a DQTA having

$$\tau : \mathcal{H} \otimes (\mathcal{U} \oplus \mathcal{K}) \to \mathcal{H} \otimes (\mathcal{U} \oplus \mathcal{L})$$

as its transition operator. Then $\uparrow^{\mathcal{U}} T : \mathcal{K} \to \mathcal{L}$ is the automaton over (the *same* space) $\mathcal{H}$ specified as follows. Consider the matrix of τ:

$$\begin{array}{c c} & \begin{array}{cc} \mathcal{H} \otimes \mathcal{U} & \mathcal{H} \otimes \mathcal{L} \end{array} \\ \begin{array}{c} \mathcal{H} \otimes \mathcal{U} \\ \mathcal{H} \otimes \mathcal{K} \end{array} & \left(\begin{array}{cc} \tau_A & \tau_B \\ \tau_C & \tau_D \end{array} \right) \end{array}$$

according to the biproduct decomposition

$$\tau = \langle [\tau_A, \tau_C], [\tau_B, \tau_D] \rangle,$$

where $[\_,\_]$ stands for coproduct and $\langle \_,\_ \rangle$ for product. The transition operator of $\uparrow^{\mathcal{U}} T$ is defined by the *Kleene formula*:

$$\uparrow^{\mathcal{U}} \tau = \lim_{n \to \infty} (\tau_D + \tau_C \circ \tau_A^{*n} \circ \tau_B). \tag{1}$$

In the Kleene formula, $\tau_A^{*n} = \sum_{i=0}^{n} \tau_A^i$, where $\tau_A^0 = I$ and $\tau_A^{i+1} = \tau_A^i \circ \tau_A$. In other words, τ_A^{*n} is the n-th approximation of τ_A's *Neumann series* well-known in operator theory. The correctness of the above definition is contingent upon the existence of the limit and also on the resulting operator being an isometry. For these two conditions we need to make a short digression, which will also clarify the linear algebraic background.

Let **Iso** denote the subcategory of **FdHilb** having only isometries as its morphisms. Notice that $(\mathbf{Iso}, \otimes)$ is no longer compact closed, even though the multiplicative tensor $\otimes$ is still intact in it. (The duals are gone.) This tensor, however, does not concern us at the moment. Consider $\oplus$ as an additive tensor in **Iso**:

$$\tau_1 \oplus \tau_2 = \langle [\tau_1, 0], [0, \tau_2] \rangle$$

for all isometries $\tau_i : \mathcal{H}_i \to \mathcal{K}_i$, $i = 1, 2$. Clearly, $\tau_1 \oplus \tau_2$ is an isometry. The new additive unit (zero) object is the zero space $\mathcal{Z}$. With the additive symmetries $\kappa_{\mathcal{H},\mathcal{K}} : \mathcal{H} \oplus \mathcal{K} \to \mathcal{K} \oplus \mathcal{H}$, $(\mathbf{Iso}, \oplus)$ again qualifies as a monoidal category. The biproduct property of $\oplus$ is lost, however. Nevertheless, one may attempt to define a trace operation $\uparrow^{U} \tau$ in **Iso** by the Kleene formula (1), where $\tau : \mathcal{U} \oplus \mathcal{K} \to \mathcal{U} \oplus \mathcal{L}$. (Cut $\mathcal{H}\otimes$ in the matrix of τ.)

Since the Kleene formula does not appear to be manageable, we first redefine $\Uparrow^{\mathcal{U}} \tau$ and prove the equivalence of the two definitions later. Let

$$\Uparrow^{\mathcal{U}} \tau = \tau_D + \tau_C \circ (I - \tau_A)^{+} \circ \tau_B, \tag{2}$$

where $(\,)^{+}$ denotes the *Moore-Penrose generalized inverse* of linear operators. Recall e.g. from [8] that the Moore-Penrose inverse (MP inverse, for short) of an arbitrary operator $\sigma : \mathcal{H} \to \mathcal{K}$ is the unique operator $\sigma^{+} : \mathcal{K} \to \mathcal{H}$ satisfying the following two conditions:

(i) $\sigma \circ \sigma^{+} \circ \sigma = \sigma$, and $\sigma^{+} \circ \sigma \circ \sigma^{+} = \sigma^{+}$;
(ii) $\sigma \circ \sigma^{+}$ and $\sigma^{+} \circ \sigma$ are Hermitian.

The connection between formulas (1) and (2) is the following. If the Neumann series τ_A^{*} converges, then $(I - \tau_A)$ is invertible and

$$\tau_A^{*} = (I - \tau_A)^{-1} = (I - \tau_A)^{+}.$$

We know that $\|\tau_A\| \leq 1$, where $\|\,\|$ denotes the operator norm. (Remember that τ is an isometry.) Therefore the Kleene formula needs an explanation only if $\|\tau_A\| = 1$. In that case, even if $(I - \tau_A)$ is invertible, τ_A^{*} may not converge.

Just as the Kleene formula in computer science, the expression on the right-hand side of equation (2) is well-known and frequently used in linear algebra. For a block matrix

$$M = \left(\begin{array}{cc} A & B \\ C & D \end{array} \right),$$

where A is square, the matrix $D - CA^{+}B$ is called the *Schur complement* of A on M, denoted A/M. See e.g. [8]. Observe that, under the assumption $\mathcal{K} = \mathcal{L}$,

$$\Uparrow^{\mathcal{U}} \tau = I - (I - \tau_A)/(I - \tau).$$

For this reason we call $\Uparrow^{\mathcal{U}} \tau$ the *Schur I-complement* of τ_A on τ, and write $\Uparrow^{\mathcal{U}} \tau = \tau_A \backslash \tau$.

Theorem 4.1. *The operator $\tau_A \backslash \tau$ is an isometry.*

Proof. Isolate the kernel $\mathcal{N}$ of $(I - \tau_A)$, and let $\mathcal{U}_0$ be the orthogonal complement [22] of $\mathcal{N}$ on $\mathcal{U}$. The matrix of $(I - \tau_A)$ in this breakdown is

$$I - \tau_A = \begin{array}{c c} & \begin{array}{cc} \mathcal{N} & \mathcal{U}_0 \end{array} \\ \begin{array}{c} \mathcal{N} \\ \mathcal{U}_0 \end{array} & \left(\begin{array}{cc} 0 & 0 \\ -\tau_A^{\mathcal{N}} & I - \tau_A^0 \end{array} \right) \end{array}. \tag{3}$$

Put this matrix in the top left corner of τ:

$$\begin{array}{c c} & \begin{array}{ccc} \mathcal{N} & \mathcal{U}_0 & \mathcal{L} \end{array} \\ \begin{array}{c} \mathcal{N} \\ \mathcal{U}_0 \\ \mathcal{K} \end{array} & \left(\begin{array}{ccc} I & 0 & \tau_B^{\mathcal{N}} \\ \tau_A^{\mathcal{N}} & \tau_A^0 & \tau_B^0 \\ \tau_C^{\mathcal{N}} & \tau_C^0 & \tau_D \end{array} \right) \end{array}.$$

Since τ is an isometry (regardless of its concrete orthogonal representation as a matrix operator), all entries in the above block matrix with superscript $\mathcal{N}$ must be 0. Consequently, $(I-\tau_A^0)$ is invertible and $\tau_A\backslash\tau = \tau_A^0\backslash\tau_0$, where

$$\tau_0 : \mathcal{U}_0 \oplus \mathcal{K} \to \mathcal{U}_0 \oplus \mathcal{L}$$

is the restriction of τ to the bottom right 2×2 corner. Indeed,

$$\begin{pmatrix} 0 & 0 \\ 0 & I-\tau_A^0 \end{pmatrix}^+ = \begin{pmatrix} 0 & 0 \\ 0 & (I-\tau_A^0)^{-1} \end{pmatrix},$$

so that

$$\tau_C \circ (I-\tau_A)^+ \circ \tau_B = \tau_C^0 \circ (I-\tau_A^0)^{-1} \circ \tau_B^0.$$

It turns out from the above discussion that $(I-\tau_A)$ is *group invertible* and *range-Hermitian*, cf. [8], [9]. Therefore the MP inverse of $(I-\tau_A)$ coincides with its Drazin inverse, which is the group generalized inverse of this operator. See again [8], [9]. It follows that we can assume, without loss of generality, that $(I-\tau_A)$ is invertible. Note that (3) is only a unitary similarity, therefore the sliding axiom is needed to make this argument correct. See Theorem 4.4 below. For better readability, replace the symbols τ_A, τ_B, τ_C, and τ_D by A, B, C, and D, respectively. Furthermore, ignore the composition symbol $\circ$ as if we were dealing with ordinary matrix product. Then we have:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}\begin{pmatrix} A^\dagger & C^\dagger \\ B^\dagger & D^\dagger \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}.$$

The following four matrix equations are derived:

$$AA^\dagger + BB^\dagger = I, \tag{4}$$
$$AC^\dagger + BD^\dagger = 0, \tag{5}$$
$$CA^\dagger + DB^\dagger = 0, \tag{6}$$
$$CC^\dagger + DD^\dagger = I. \tag{7}$$

We need to show that

$$(D + C(I-A)^{-1}B)(D^\dagger + B^\dagger(I-A^\dagger)^{-1}C^\dagger) = I.$$

The product on the left-hand side yields:

$$\begin{aligned} DD^\dagger &+ DB^\dagger(I-A^\dagger)^{-1}C^\dagger + C(I-A)^{-1}BD^\dagger \\ &+ C(I-A)^{-1}BB^\dagger(I-A^\dagger)^{-1}C^\dagger. \end{aligned}$$

By (5) and (6) this is equal to:

$$\begin{aligned} DD^\dagger &- CA^\dagger(I-A^\dagger)^{-1}C^\dagger - C(I-A)^{-1}AC^\dagger \\ &+ C(I-A)^{-1}BB^\dagger(I-A^\dagger)^{-1}C^\dagger, \end{aligned}$$

which is further equal to $DD^\dagger + CQC^\dagger$, where

$$Q = (I-A)^{-1}BB^\dagger(I-A^\dagger)^{-1} - A^\dagger(I-A^\dagger)^{-1} - (I-A)^{-1}A.$$

According to (7) it is sufficient to prove that $Q = I$. A couple of equivalent transformations follow.

1. Multiply both sides of $Q = I$ by $(I-A)$ from the left:

$$\begin{aligned} BB^\dagger(I-A^\dagger)^{-1} - (I-A)A^\dagger(I-A^\dagger)^{-1} - A &= I - A, \\ BB^\dagger(I-A^\dagger)^{-1} - (I-A)A^\dagger(I-A^\dagger)^{-1} &= I. \end{aligned}$$

2. Multiply by $(I-A^\dagger)$ from the right:

$$\begin{aligned} BB^\dagger - (I-A)A^\dagger &= I - A^\dagger, \\ BB^\dagger + AA^\dagger &= I. \end{aligned}$$

The result is equation (4), which is given. The proof is now complete. □

Lemma 4.2. *Let $\tau : \mathcal{U}\oplus\mathcal{V}\oplus\mathcal{K} \to \mathcal{U}\oplus\mathcal{V}\oplus\mathcal{L}$ be an isometry defined by the matrix*

$$\begin{pmatrix} \begin{matrix} & M & \\ \end{matrix} & \begin{matrix} B_1 \\ B_2 \end{matrix} \\ \begin{matrix} C_1 & C_2 \end{matrix} & D \end{pmatrix}, \text{ where } M = \begin{pmatrix} P & Q \\ R & S \end{pmatrix}.$$

If $I - (P\backslash M) = I - (S + R(I-P)^+Q)$ is invertible, then

$$\Uparrow_{\mathcal{V}} (\Uparrow_{\mathcal{U}} \tau) = \Uparrow_{\mathcal{U}\oplus\mathcal{V}} \tau.$$

Proof. Using the kernel-on-top representation of operators as explained under Theorem 4.1, we can assume (without loss of generality) that $I-P$ is also invertible. Then the statement follows from the Banachiewicz block inverse formula [9, Proposition 2.8.7]:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} =$$

$$\begin{pmatrix} A^{-1}+A^{-1}B(D-CA^{-1}B)^{-1} & -A^{-1}B(D-CA^{-1}B)^{-1} \\ -(D-CA^{-1}B)^{-1}CA^{-1} & (D-CA^{-1}B)^{-1} \end{pmatrix},$$

using $A = I-P$, $B = -Q$, $C = -R$, and $D = I-S$. Computations are left to the reader. □

Note that the Banachiewicz formula does not hold true for the MP or the Drazin inverse of the given block matrix when A^{-1} and $(D-CA^{-1}B)^{-1}$ are replaced on the right-hand side by A^+ and $(D-CA^+B)^+$, respectively, even if one of these square matrices is invertible. There are appropriate block inverse formulas for generalized inverses, cf. [9], but these formulas are extremely complicated and are of no use for us.

Lemma 4.3. *Let $\tau : \mathcal{U}\oplus\mathcal{V}\oplus\mathcal{K} \to \mathcal{U}\oplus\mathcal{V}\oplus\mathcal{L}$ be an isometry as in Lemma 4.2. If $P\backslash M = I$, then*

$$\Uparrow_{\mathcal{V}} (\Uparrow_{\mathcal{U}} \tau) = \Uparrow_{\mathcal{U}\oplus\mathcal{V}} \tau.$$

Proof. Again, we can assume that $I-P$ is invertible. To keep the computation simple, let $\mathcal{U}$ and $\mathcal{V}$ both be 1-dimensional. This, too, can in fact be assumed without loss of generality, if one uses an appropriate induction argument. The induction, however, can be avoided at the expense of a more advanced matrix computation. Thus,

$$\tau = \begin{pmatrix} p & q & u_1 \\ r & s & u_2 \\ v_1\downarrow & v_2\downarrow & D \end{pmatrix},$$

where u_i and $(v_i\downarrow)$, $i = 1,2$ are row and column vectors, respectively. To simplify the computation even further, let the numbers p,q,r,s be real. The 2×2 matrix $I-M$ is singular and range-Hermitian, therefore it is Hermitian (only because the numbers are real, see [9, Corollary 5.4.4]), so that it must be of the form

$$I - M = \begin{pmatrix} a & b \\ b & b^2/a \end{pmatrix}$$

for some real numbers a, b with $a = 1 - p \neq 0$. Then

$$\Uparrow_{\mathcal{U}} \tau = \begin{pmatrix} c & u \\ v\!\downarrow & D' \end{pmatrix},$$

where $c = (1 - b^2/a) + b^2/a = 1$,

$$\begin{array}{rcl} u & = & u_2 - (b/a) \cdot u_1, \\ (v\!\downarrow) & = & (v_2\!\downarrow) - (b/a) \cdot (v_1\!\downarrow), \quad \text{and} \\ D' & = & D + (1/a) \cdot (v_1\!\downarrow) u_1. \end{array}$$

Since $c = 1$, u and $(v\!\downarrow)$ must be 0. Consequently,

$$a \cdot u_2 = b \cdot u_1 \text{ and } a \cdot (v_2\!\downarrow) = b \cdot (v_1\!\downarrow). \tag{8}$$

In order to calculate $(I - M)^+$, let $M' = S(I - M)S^{-1}$, where $S = S^{-1}$ is the unitary matrix

$$S = \frac{1}{d} \cdot \begin{pmatrix} -b & a \\ a & b \end{pmatrix}, \quad d^2 = a^2 + b^2.$$

After a short computation,

$$M' = \begin{pmatrix} 0 & 0 \\ 0 & d^2/a \end{pmatrix}.$$

It follows that:

$$(I - M)^+ = S \begin{pmatrix} 0 & 0 \\ 0 & a/d^2 \end{pmatrix} S, \quad \text{and}$$

$$\Uparrow_{\mathcal{U}\oplus\mathcal{V}} \tau = D + (v_1\!\downarrow, v_2\!\downarrow) S \begin{pmatrix} 0 & 0 \\ 0 & a/d^2 \end{pmatrix} S \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}.$$

Comparing this expression with

$$\Uparrow_{\mathcal{V}} (\Uparrow_{\mathcal{U}} \tau) = D' = D + (1/a) \cdot (v_1\!\downarrow) u_1,$$

we need to prove that

$$(v_1\!\downarrow, v_2\!\downarrow) S \begin{pmatrix} 0 & 0 \\ 0 & a/d^2 \end{pmatrix} S \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \frac{1}{a} \cdot (v_1\!\downarrow) u_1.$$

On the left-hand side we have:

$$\begin{aligned} & \frac{a}{d^4} \cdot (a \cdot v_1\!\downarrow + b \cdot v_2\!\downarrow)(a \cdot u_1 + b \cdot u_2) \\ & = \frac{a}{d^4} \cdot (a \cdot v_1\!\downarrow + \frac{b^2}{a} \cdot v_1\!\downarrow)(a \cdot u_1 + \frac{b^2}{a} \cdot u_1) \quad \text{(by (8))} \\ & = \frac{a}{d^4} \cdot (\frac{a^2 + b^2}{a} \cdot v_1\!\downarrow)(\frac{a^2 + b^2}{a} \cdot u_1) \\ & = \frac{1}{a} \cdot (v_1\!\downarrow) u_1. \end{aligned}$$

The proof is complete.

Theorem 4.4. *The operation $\Uparrow_{\mathcal{U}}$ defines a trace for the monoidal category* $(\mathbf{Iso}, \oplus)$.

Proof. Naturality of trace with respect to permutations is easy to see. For example, the sliding axiom (dinaturality in $\mathcal{U}$) can be shown for an arbitrary permutation $\sigma : \mathcal{V} \to \mathcal{U}$ as follows. Let $\tau : \mathcal{U} \oplus \mathcal{K} \to \mathcal{U} \oplus \mathcal{L}$ be an isometry with $\langle [A, B], [C, D] \rangle$ being the biproduct decomposition (matrix) of τ. Then, for the "matrix" S of σ:

$$\begin{array}{l} \Uparrow_{\mathcal{V}} ((\sigma \oplus I) \circ \tau \circ (\sigma^{-1} \oplus I)) \\ \quad = \quad D + CS^{-1}(I - SAS^{-1})^+ SB \\ \quad = \quad D + CS^{-1}(SS^{-1} - SAS^{-1})^+ SB \\ \quad = \quad D + CS^{-1}(S(I - A)S^{-1})^+ SB \\ \quad = \quad D + CS^{-1}S(I - A)^+ S^{-1}SB \\ \quad = \quad D + C(I - A)^+ B = \Uparrow_{\mathcal{U}} \tau. \end{array}$$

In the above derivation we have used the obvious property

$$(SMS^{-1})^+ = SM^+S^{-1}$$

of the MP inverse. Remember that σ is a permutation, so that $\sigma^{-1} = \sigma^\dagger$. Superposing, yanking, and the derived composition axiom are trivial. Therefore the only challenging axiom is vanishing.

Let $\tau : \mathcal{U} \oplus \mathcal{V} \oplus \mathcal{K} \to \mathcal{U} \oplus \mathcal{V} \oplus \mathcal{L}$ be an isometry given by the matrix

$$\begin{pmatrix} M & B \\ C & D \end{pmatrix}, \text{ where } M = \begin{pmatrix} P & Q \\ R & S \end{pmatrix}.$$

We need to prove that $\Uparrow_{\mathcal{V}} (\Uparrow_{\mathcal{U}} \tau) = \Uparrow_{\mathcal{U}\oplus\mathcal{V}} \tau$. Again, without loss of generality, we can assume that $(I - P)$ is invertible and

$$I - P \backslash M = \begin{pmatrix} 0 & 0 \\ 0 & S_0 \end{pmatrix},$$

where $\mathcal{V} = \mathcal{N} \oplus \mathcal{V}_0$ and $S_0 : \mathcal{V}_0 \to \mathcal{V}_0$ is invertible. If $\mathcal{N}$ is the zero space, so that $I - P \backslash M$ itself is invertible, then the statement follows from Lemma 4.2. Otherwise

$$\Uparrow_{\mathcal{V}} (\Uparrow_{\mathcal{U}} \tau) = \Uparrow_{\mathcal{V}_0} (\Uparrow_{\mathcal{N}} (\Uparrow_{\mathcal{U}} \tau)).$$

By Lemma 4.3, $\Uparrow_{\mathcal{N}} (\Uparrow_{\mathcal{U}} \tau) = \Uparrow_{\mathcal{U}\oplus\mathcal{N}} \tau$, and by Lemma 4.1,

$$\Uparrow_{\mathcal{V}_0} (\Uparrow_{\mathcal{U}\oplus\mathcal{N}} \tau) = \Uparrow_{\mathcal{U}\oplus\mathcal{N}\oplus\mathcal{V}_0} \tau = \Uparrow_{\mathcal{U}\oplus\mathcal{V}} \tau.$$

The proof is now complete. □

At this point the reader may want to check the validity of the Conway semiring axioms

$$(ab)^* = a(ba)^*b + 1, \quad (a + b)^* = (a^*b)^*a$$

for all $a, b \in \mathbb{C}$, where

$$c^* = (1 - c)^+ = \begin{cases} (1 - c)^{-1} & \text{if } c \neq 1 \\ 0 & \text{if } c = 1. \end{cases}$$

See [10]. Obviously, they do not hold, but they come very close. It may also occur to the reader that the Schur I-complement defines a trace in the whole category $(\mathbf{FdHilb}, \oplus)$. Of course this is not true either, because the Banachiewicz formula does not work for the MP inverse.

In a recent paper [21], Malherbe et al. introduced the so called kernel-image trace as a partial trace [14] on any additive category $\mathcal{C}$. Given a morphism $\tau : \mathcal{U} \oplus \mathcal{K} \to \mathcal{U} \oplus \mathcal{L}$ in $\mathcal{C}$ with a block matrix $\tau = \langle [\tau_A, \tau_C], [\tau_B, \tau_D] \rangle$ as above, the *kernel-image trace* $\uparrow^{\mathcal{U}}_{k-i} \tau$ is defined if both τ_B and τ_C factor through $(I - \tau_A)$, that is, there exist morphisms $i : \mathcal{K} \to \mathcal{U}$ and $k : \mathcal{U} \to \mathcal{L}$ such that

$$\tau_C = i \circ (I - \tau_A) \ \text{ and } \ \tau_B = (I - \tau_A) \circ k.$$

See Fig. 4.

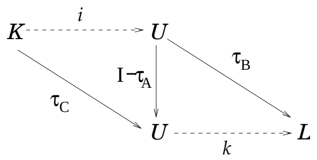

Fig. 4. The kernel-image trace

In this case

$$\uparrow^{\mathcal{U}}_{k-i} \tau = \tau_D + \tau_C \circ k = \tau_D + i \circ \tau_B.$$

It is easy to see that $\uparrow^{\mathcal{U}}_{k-i} \tau$ is always defined if τ is an isometry, and $\uparrow^{\mathcal{U}}_{k-i} \tau = \Uparrow^{\mathcal{U}} \tau$. (Use the kernel-on-top transformation of $(I - \tau_A)$ as in Theorem 4.1.) Therefore $\uparrow^{\mathcal{U}}_{k-i}$ is totally defined on $(\mathbf{Iso}, \oplus)$ and it coincides with $\Uparrow^{\mathcal{U}}$. Using [21, Remark 3.3] we thus have an alternative proof of our Theorem 4.4 above.

Now we turn back to the original definition of trace in $(\mathbf{Iso}, \oplus)$ by (1).

Theorem 4.5. *For every isometry $\tau : \mathcal{U} \oplus \mathcal{K} \to \mathcal{U} \oplus \mathcal{L}$, $\uparrow^{\mathcal{U}} \tau$ is well defined as an isometry $\mathcal{K} \to \mathcal{L}$. Moreover,*

$$\uparrow^{\mathcal{U}} \tau = \Uparrow^{\mathcal{U}} \tau.$$

Proof. This is in fact a simple formal language theory exercise. Take a concrete representation of τ as an $(n+k) \times (n+l)$ complex matrix (a_{ij}), where n, k, and l are the dimensions of $\mathcal{U}$, $\mathcal{K}$, and $\mathcal{L}$, respectively. For a corresponding set of variables $X = \{x_{ij}\}$, consider the matrix iteration theory $\mathbf{Mat}_{\ L(X^*)}$ determined by the iteration semiring of all *formal power series* over the ω-complete Boolean semiring $\mathbf{B}$ with variables X as described in Chapter 9 of [10]. The fundamental observation is that $\uparrow^n (a_{ij})$ is the evaluation of the series matrix $\uparrow^n (x_{ij})$ under the assignment $x_{ij} = a_{ij}$, provided that each entry in this matrix is convergent. In our case, since $|a_{11}| \leq 1$, this matrix is definitely convergent if $n = 1$, and $\uparrow^1 (a_{ij}) = \Uparrow^1 (a_{ij})$. A straightforward induction on the basis of Theorem 4.4 then yields $\uparrow^n (a_{ij}) = \Uparrow^n (a_{ij})$, knowing that every iteration theory is a traced monoidal category. $\square$

Corollary 4.6. *The monoidal category $(\mathbf{DQT}, \boxtimes)$ is traced by the feedback $\uparrow$.*

Proof. Now the key observation is that, for every isometry $\tau : \mathcal{U} \oplus \mathcal{K} \to \mathcal{U} \oplus \mathcal{L}$ and object $\mathcal{M}$,

$$(\Uparrow^{\mathcal{U}} \tau) \otimes I_{\mathcal{M}} = \Uparrow^{\mathcal{U} \otimes \mathcal{M}} (\tau \otimes I_{\mathcal{M}}).$$

This equation is an immediate consequence of

$$(\sigma \otimes I)^+ = \sigma^+ \otimes I,$$

which is an obvious property of the MP inverse. (See the defining equations (i)-(ii) of σ^+.) In the light of this observation, each traced monoidal category axiom is essentially the same in $(\mathbf{DQT}, \boxtimes)$ as it is in $(\mathbf{Iso}, \oplus)$. Thus, the statement follows from Theorems 4.4 and 4.5. $\square$

V. Making Turing automata bidirectional

Now we are ready to introduce the model of quantum Turing automata as a real quantum computing device.

Definition 5.1. A *quantum Turing automaton* of *rank* $\mathcal{K}$ is a triple $T = (\mathcal{H}, \mathcal{K}, \tau)$, where $\mathcal{H}$ and $\mathcal{K}$ are finite dimensional Hilbert spaces and $\tau : \mathcal{H} \otimes \mathcal{K} \to \mathcal{H} \otimes \mathcal{K}$ is a *unitary* morphism in **FdHilb**.

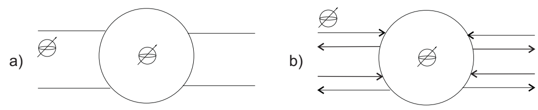

Fig. 5. One cell of a Turing machine as a QTA

Again, two automata $T_i : (\mathcal{H}_i, \mathcal{K}, \tau_i)$, $i = 1, 2$ are called *isomorphic* if there exists an isometric isomorphism $\sigma : \mathcal{H}_1 \to \mathcal{H}_2$ for which $\tau_2 = (\sigma^\dagger \otimes I_{\mathcal{K}}) \circ \tau_1 \circ (\sigma \otimes I_{\mathcal{K}})$.

Example In Fig. 5a, consider the abstract representation of one tape cell drawn from a hypothetical Turing machine having two states: 1 and 2. The tape alphabet $\{0, 1\}$ is also binary, which means that there is a single qubit sitting in the cell. Thus, $\mathcal{H}$ is 2-dimensional. The control particle c can reside on any of the given four interfaces. For example, if c is on the top left interface, then control is coming from the left in state 1. After one move, c can again be on any of these four interfaces, so that the dimension of $\mathcal{K}$ is 4. Notice the undirected nature of one move, as opposed to the rigid input→output orientation forced on DQTA. The situation is, however, analogous to having a separate input and dual output interface for each undirected one in a corresponding DQTA. See Fig. 5b. Let T_0 denote the quantum Turing automaton (QTA, for short) so obtained, equipped with an appropriate transition operator τ as an 8×8 unitary matrix.

We are going to describe the structure of QTA directly as an indexed monoidal algebra $\mathcal{QT}$. The object monoid for $\mathcal{QT}$ is the monoid of objects in $(\mathbf{Iso}, \oplus)$. Morphisms $\mathcal{QT}$ of rank $\mathcal{K}$ are all isomorphism classes of QTA of rank $\mathcal{K}$. Tensor of morphisms will be denoted by $\boxplus$, for the symbol $\otimes$ is heavily overloaded.

With a slight ambiguity we identify each permutation symbol ρ with its interpretation in $(\mathbf{Iso}, \oplus)$ as a permutation isometry. Using the algebraic language, by an automaton $T : \mathcal{K}$ (transition operator $\tau : \mathcal{H} \otimes \mathcal{K}$) we mean one with state space $\mathcal{H}$ and interfaces $\mathcal{K}$. Thus, our example automaton is $T_0 : 4$ with $\tau_0 : 2 \times 4$. The algebra $\mathcal{QT}$ is defined as follows.

— For $T = (\mathcal{H}, \mathcal{K}, \tau)$ and $\rho : \mathcal{K} \Rightarrow \mathcal{L}$, $T \cdot \rho$ comes with the transition operator

$$(I_{\mathcal{H}} \otimes \rho^{-1}) \circ \tau \circ (I_{\mathcal{H}} \otimes \rho). \tag{9}$$

— The identity automaton $\mathbf{1}_{\mathcal{K}} : \mathcal{K} \oplus \mathcal{K}$ is the single-state QTA having the transition operator:

$$\kappa_{\mathcal{K},\mathcal{K}} = \langle [0, I], [I, 0] \rangle \tag{10}$$

— For automata

$$T_1 = (\mathcal{H}_1, \mathcal{K}_1, \tau_1) \text{ and } T_2 = (\mathcal{H}_2, \mathcal{K}_2, \tau_2),$$

$$T_1 \boxplus T_2 = (\mathcal{H}, \mathcal{K}_1 \oplus \mathcal{K}_2, \tau_1 \boxplus \tau_2),$$

where $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2$ and

$$\begin{gathered} \tau_1 \boxplus \tau_2 = \tau_1 \boxtimes \tau_2 : \\ \mathcal{H} \otimes (\mathcal{K}_1 \oplus \mathcal{K}_2) \to \mathcal{H} \otimes (\mathcal{K}_1 \oplus \mathcal{K}_2) \end{gathered} \tag{11}$$

— For $T = (\mathcal{H}, \mathcal{U} \oplus \mathcal{U} \oplus \mathcal{K}, \tau)$,

$$\updownarrow_{\mathcal{U}} T = (\mathcal{H}, \mathcal{K}, \updownarrow_{\mathcal{U}} \tau),$$

where

$$\updownarrow_{\mathcal{U}} \tau = \uparrow^{\mathcal{U}\oplus\mathcal{U}} (\tau \circ (I_{\mathcal{H}} \otimes (\kappa_{\mathcal{U},\mathcal{U}} \oplus I_{\mathcal{K}}))). \tag{12}$$

Notice the "alternating twist" κ in the definition of **1** and $\updownarrow$, which is characteristic of the Int construction. For a better intuitive understanding, see also the corresponding analogous definitions in [7] with respect to conventional Turing automata.

Recall from Section 2 that the functor Alg takes an arbitrary traced monoidal category and turns it into an indexed monoidal algebra through the Int construction.

Theorem 5.1. *$\mathcal{QT} \cong Alg(\mathbf{DQT})$ is an indexed monoidal algebra.*

Proof. (Sketch) First we review the definition of the functor $\mathcal{I} : \mathbf{SDCC} \to \mathbf{IMA}$ from [6], which turns an SDCC category into an equivalent IMA. Let $\mathcal{C}$ be an SDCC category over a monoid M as objects. Then, for each object A in M, the morphisms of $\mathcal{M} = \mathcal{IC}$ with rank A are all morphisms $I \to A$ in $\mathcal{C}$. Indexing in $\mathcal{M}$ is essentially the restriction of the covariant hom functor $\mathcal{C}(I, \_)$ to permutations, and

$$\mathbf{1}_A = d_A : I \to A \otimes A.$$

For $f : A$ and $g : B$,

$$f \otimes_{\mathcal{M}} g = f \otimes_{\mathcal{C}} g : I \to A \otimes B,$$

and for $f : A \otimes A \otimes B$, $\updownarrow_A f$ is defined as the canonical trace of the morphism $f_A : A \to A \otimes B$ in $\mathcal{C}$ that corresponds to f according to compact closure.

Now let $\mathcal{C}$ be the restriction of $Int(\mathbf{DQT})$ to its self-dual objects $(\mathcal{K}, \mathcal{K})$. That is, a morphism $\mathcal{K} \to \mathcal{L}$ in $\mathcal{C}$ is a morphism $(\mathcal{K}, \mathcal{K}) \to (\mathcal{L}, \mathcal{L})$ in $Int(\mathbf{DQT})$. The morphisms of rank $\mathcal{K}$ in $Alg(\mathbf{DQT})$, being the morphisms $(\mathcal{Z}, \mathcal{Z}) \to (\mathcal{K}, \mathcal{K})$ in $Int(\mathbf{DQT})$, are therefore isometries $\mathcal{H} \otimes \mathcal{K} \to \mathcal{H} \otimes \mathcal{K}$ for some $\mathcal{H}$. In **FdHilb**, these are exactly the unitary maps $\mathcal{H} \otimes \mathcal{K} \to \mathcal{H} \otimes \mathcal{K}$. Consequently, the correspondence between the morphisms of $Alg(\mathbf{DQT})$ and $\mathcal{QT}$ is one-to-one and onto.

It is easy to verify that the unit map $d_{\mathcal{K}}$ in $\mathcal{C}$ is $\kappa_{\mathcal{K},\mathcal{K}}$, so that the definition (10) of $\mathbf{1}_{\mathcal{K}}$ is correct. Also, indexing becomes the combination of the covariant and contravariant hom functors $\mathbf{DQT}(I, \_)$ and $\mathbf{DQT}(\_, I)$ as specified in (9). Concerning (11), let

$$T_1 : (\mathcal{Z}, \mathcal{Z}) \to (\mathcal{K}_1, \mathcal{K}_1) \text{ and } T_2 : (\mathcal{Z}, \mathcal{Z}) \to (\mathcal{K}_2, \mathcal{K}_2)$$

be morphisms in $Int(\mathbf{DQT})$. By definition, $T_1 = (\mathcal{H}_1, \tau_1) : \mathcal{K}_1 \to \mathcal{K}_1$ and $T_2 = (\mathcal{H}_2, \tau_2) : \mathcal{K}_2 \to \mathcal{K}_2$ in **DQT**. Again, by the very definition of $\otimes$ in $Int(\mathbf{DQT})$,

$$T_1 \otimes T_2 = T_1 \boxtimes T_2 : \mathcal{K}_1 \oplus \mathcal{K}_2 \to \mathcal{K}_1 \oplus \mathcal{K}_2.$$

Similarly, as to (12), if

$$T : (\mathcal{Z}, \mathcal{Z}) \to (\mathcal{U} \oplus \mathcal{U} \oplus \mathcal{K}, \mathcal{U} \oplus \mathcal{U} \oplus \mathcal{K}), \text{ that is}$$

$$T = (\mathcal{H}, \tau) : \mathcal{U} \oplus \mathcal{U} \oplus \mathcal{K} \to \mathcal{U} \oplus \mathcal{U} \oplus \mathcal{K},$$

then the morphism $T_{\mathcal{U}} : \mathcal{U} \to \mathcal{U} \oplus \mathcal{K}$ in $\mathcal{C}$ (that is,

$$T_{(\mathcal{U},\mathcal{U})} : (\mathcal{U}, \mathcal{U}) \to (\mathcal{U} \oplus \mathcal{K}, \mathcal{U} \oplus \mathcal{K})$$

in $Int(\mathbf{DQT})$) that corresponds to T by compact closure is the automaton

$$(\mathcal{H}, \tau_{\mathcal{U}}) : \mathcal{U} \oplus \mathcal{U} \oplus \mathcal{K} \to \mathcal{U} \oplus \mathcal{K} \oplus \mathcal{U},$$

where $\tau_{\mathcal{U}} = \tau \circ (I_{\mathcal{H}} \otimes \kappa_{\mathcal{U},\mathcal{U}\oplus\mathcal{K}})$. Therefore

$$Tr^{\mathcal{U}} T_{\mathcal{U}} = Tr^{(\mathcal{U},\mathcal{U})} T_{(\mathcal{U},\mathcal{U})} = (\mathcal{H}, \sigma),$$

where

$$\begin{aligned} \sigma &= \uparrow^{\mathcal{U}\oplus\mathcal{U}} (\tau_{\mathcal{U}} \circ (I_{\mathcal{H}} \otimes (I_{\mathcal{U}} \oplus \kappa_{\mathcal{K},\mathcal{U}}))) \\ &= \uparrow^{\mathcal{U}\oplus\mathcal{U}} (\tau \circ (I_{\mathcal{H}} \otimes (\kappa_{\mathcal{U},\mathcal{U}} \oplus I_{\mathcal{K}}))) = \updownarrow_{\mathcal{U}} \tau. \end{aligned}$$

The proof is complete. □

Example (Continued) In Fig. 6a, consider a segment of our hypothetical Turing machine consisting of n cells. As shown in Fig. 6b, the semantics of this segment as a QTA is:

$$\updownarrow_{2(n-1)} ((\boxplus_{i=1}^{n} T_0) \cdot \rho) : 4,$$

where $\rho : 4n \to 4n$ is the permutation that sends $k = 1, 2$ and $l = 4n-1, 4n$ to $4(n-1)+k$ and l, respectively, and

$$\rho(4i+k) = \begin{cases} 2i + (k-2) & \text{if } 0 \le i < n-1,\ k = 3, 4 \\ 2(n-1) + 2(i-1) + k & \text{if } 1 \le i < n,\ k = 1, 2. \end{cases}$$

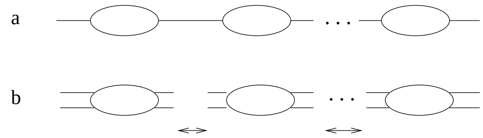

Fig. 6. A segment of a Turing machine as a QTA

VI. Conclusion

We have provided a theoretical foundation for the study of quantum Turing machines. The biproduct strongly compact closed category **FdHilb** of finite dimensional Hilbert spaces served as the underlying structure for this foundation. We narrowed down the scope of this category to isometries, switched from multiplicative to additive tensor, and defined a new additive trace operation by the help of the Moore-Penrose generalized inverse. This trace was then carried over to the monoidal category of directed quantum Turing automata. Finally, we applied the Int construction to obtain a compact closed category, which we further transformed into the indexed monoidal algebra of undirected quantum Turing automata.

REFERENCES

[1] S. Abramsky and B. Coecke, A categorical semantics of quantum protocols, In *Proceedings, 19th Annual Symposium on Logic in Computer Science, LICS 2004*, pp. 415–425. IEEE Computer Society Press, 2004.

[2] M. Bartha, A finite axiomatization of flowchart schemes, *Acta Cybernet.* **8** (1987) 203–217. Available at www.cs.mun.ca/~bartha/linked/flow.pdf

[3] M. Bartha, An equational axiomatization of systolic systems. *Theoret. Comput. Sci.* **55** (1987), 265–289.

[4] M. Bartha, An algebraic model of synchronous systems. *Inform. and Comput.* **97** (1992), 97–131.

[5] M. Bartha, Simulation equivalence of automata and circuits, 12th International Conference on Automata and Formal Languages, Balatonfüred, Hungary, May, 2008, Local Proceedings, pp. 86–99. Available at www.cs.mun.ca/~bartha/linked/afl08.pdf

[6] M. Bartha, Turing automata and graph machines, 6th Workshop on Developments in Computational Models, Edinburgh, Scotland, 2010, Electronic Proceedings in Theoretical Computer Science **26**, pp.19–31.

[7] M. Bartha, The monoidal structure of Turing machines, *Mathematical Structures in Computer Science*, to appear. Available at www.cs.mun.ca/~bartha/linked/mono.pdf

[8] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, Springer-Verlag, 2003.

[9] D. S. Bernstein, *Matrix Mathematics*, Princeton University Press, 2005.

[10] S. L. Bloom and Z. Ésik, *Iteration Theories: The Equational Logic of Iterative Processes*, Springer-Verlag, 1993.

[11] E. D'Hondt, and P. Panangaden, Quantum weakest preconditions, *Mathematical Structures in Computer Science* **16** (2006), 429–451.

[12] L. de Francesco Albasini, N. Sabadini, and R.F.C. Walters, An algebra of automata which includes both classical and quantum entities, *ENTCS* **270** (2011), 263–272.

[13] J.-Y. Girard, Geometry of Interaction I: Interpretation of system F. In *Logic Colloquium '88*, Studies in Logic, North Holland, 1989, pp. 221–260.

[14] E. Haghverdi, P. J. Scott, Towards a typed geometry of interaction, *Mathematical Structures in Computer Science* **20** (2010), 1–49.

[15] I. Hasuo, N. Hoshino, Semantics of higher order quantum computation via Geometry of Interaction, In *Proceedings, 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011*, pp. 237–246. IEEE Computer Society Press, 2011.

[16] A. Joyal, R. Street, and D. Verity, Traced monoidal categories, *Math. Proc. Camb. Phil. Soc.* **119** (1996), 447–468.

[17] P. Katis, N. Sabadini, and R.F.C. Walters, Feedback, trace, and fixed-point semantics, *Theoret. Informatics Appl.* **36** (2002), 181–194.

[18] G. M. Kelly and M. L. Laplaza, Coherence for compact closed categories, *J. Pure Appl. Algebra* **19** (1980), 193–213.

[19] G. M. Kelly, *Basic concepts of enriched category theory*, London Math. Soc. Lecture Notes Series No. 64, Cambridge University Press, 1982.

[20] S. Mac Lane, *Categories for the Working Mathematician*, Springer-Verlag, 1997.

[21] O. Malherbe, P. J. Scott, P. Selinger, Partially traced categories, arXiv[math.CT] 1107.3608, 2011.

[22] S. Roman, *Advanced Linear Algebra*, Springer-Verlag, 2005.

[23] P. Selinger, Towards a quantum programming language, *Mathematical Structures in Computer Science* **14** (2004), 527–586.