TRANSFER OF STATUS REPORT: An abstract perspective on Entanglement and Entropy with applications to Quantum Information Theory

Jose Hugo Max Nava Kopp
Oxford University. Computer Science Department

May , 2012

Contents

1 Introduction

This PRS Transfer report looks at recently proposed categorical frameworks that characterize entropy as well as accommodate both classical and quantum probability theory [23] [24]. We identify common categorical components and propose further directions this approach could take.

Entangled quantum states work as a means of communication. Entropy is a measurement of information content [12]. Entropy works as a measure of information, although there are other measures [26]. In this report, it is proposed to relate quantum entanglement and entropy in an abstract manner within the categorical quantum mechanics framework. Following this objective, we try to incorporate compression as a quintessential *information-theoretical* trait that ought to be captured categorically when studying entropy, since entropy is precisely a quantification of compression.

More in the software engineering direction, a work plan for the documentation of the Quantomatic software is proposed.

This report uses [23] and [24] as seminal papers. Regarding the requirements of the PRS Transfer Report, section 2 and 3 conform the bulk of the progress report and literature review. Sections 4 and 5 conform the thesis proposal.

2 Overview of CQM

2.1 Symmetric Monoidal Compact Closed Categories

In [23] a diagrammatic framework is stated in the graphical language of *symmetric monoidal categories with compact structures and Frobenius structures*. A description of what each of these structures adds to the general idea of category is given next.

Monoidal

A monoidal structure on a category provides an internal (bi-)functor and an identity object, as well as associativity for this bi-functor. The idea of having a bi-functor with same domain and codomain suggests an abstraction for an algebraic operation.

More precisely, a monoidal category is a structure $(\mathcal{C}, \otimes, I, a, l, r)$ where $\mathcal{C}$ is a category, $\otimes : \mathcal{C} \times \mathcal{C}$ is the internal functor, I is the identity object and a, l, r are natural isomorphisms defined as so:

$$a_{A,B,C} : A \otimes (B \otimes C) \xrightarrow{\cong} (A \otimes B) \otimes C \tag{1}$$

$$l_A : I \otimes A \xrightarrow{\cong} A \qquad r_A : A \otimes I \xrightarrow{\cong} A \tag{2}$$

such that $l_I = r_I$ and :

$$\begin{array}{ccc} A \otimes (I \otimes B) & \xrightarrow{a} & (A \otimes I) \otimes B \\ & {\scriptstyle id\otimes l} \searrow \quad \swarrow {\scriptstyle r\otimes id} & \\ & A \otimes B & \end{array} \tag{3}$$

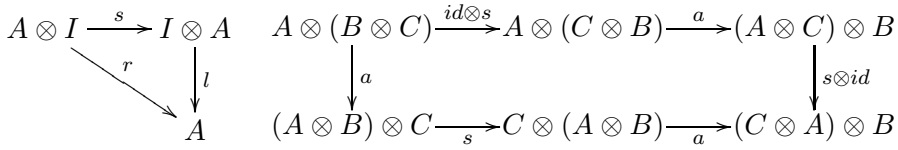
$$\begin{array}{ccccc} & & (A \otimes B) \otimes (C \otimes D) & & \\ & {\scriptstyle a} \nearrow & & \searrow {\scriptstyle a} & \\ A \otimes (B \otimes (C \otimes D)) & & & & ((A \otimes B) \otimes C) \otimes D \\ {\scriptstyle id\otimes a}\downarrow & & & & \uparrow{\scriptstyle a\otimes id} \\ A \otimes ((B \otimes C) \otimes D) & & \xrightarrow{\quad a \quad} & & (A \otimes (B \otimes C)) \otimes D \end{array} \tag{4}$$

Symmetric

The symmetric structure provides commutativity to the monoidal bi-functor, namely a natural isomorphism

$$s_{A,B} : A \otimes B \xrightarrow{\cong} B \otimes A \tag{5}$$

such that $s_{B,A} = s_{A,B}^{-1}$ and the following two diagrams commute:



(6)

Closed

A closed structure identifies the Symmetric Monoidal Categories $\mathcal{C}$ whose hom-sets $\mathcal{C}(A, B)$ have the same structure as the objects of the category. Formally this means that for every object the Hom-functor

$$F := \_\_ \otimes A : \mathcal{C} \to \mathcal{C}$$

lands on a newly defined object

$$X^A := F(X) \in |\mathcal{C}|, \quad \forall X \in |\mathcal{C}|$$

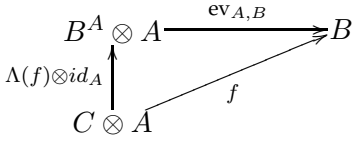
provided by the closed structure. In other words, there is an object B^A and a morphism

$$\text{ev}_{A,B} : B^A \otimes A \longrightarrow B$$

for every pair of objects A, B, such that for every morphism $f : C \otimes A \longrightarrow B$ there is a unique morphism $\Lambda(f) : C \to B^A$ that satisfies

$$\text{ev}_{A,B} \circ (\Lambda(f) \otimes id_A) = f$$

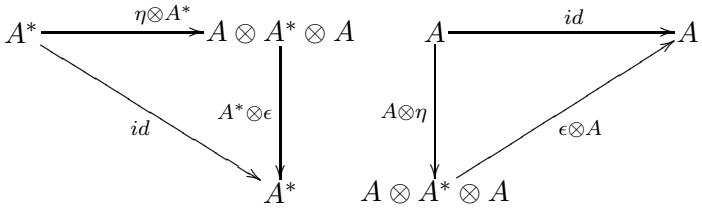
graphically:



(7)

Compact

A compact structure provides the *dual* object, a generalization of dual of a vector space. In more detail, a compact structure in any Symmetric Monoidal Closed Category $\mathcal{C}$ is a quadruple (A, A^*, ϵ, η) where the pairing $\epsilon : A \otimes A^* \to I$ and co-pairing $\epsilon : I \to A \otimes A^*$ make the following diagrams commute:



(8)

Dagger

A dagger category is a category equipped with an involutive functor which inverts the direction of each morphism and leaves the objects intact.

Dagger compact categories introduced in [1] express quantum mechanical concepts like unitary maps, inner products and projectors [7]. They are also complete with respect to finite dimensional Hilbert spaces [8].

That said, what physical trait is being captured by the dagger structure can only be conjectured. Yet it is possible to identify at least two traits that makes **Rel** a dagger category and hence a quantum-like category as opposed to **Set**, a non-quantum or classic-like category [11].

A particle in Newtonian physics has a definite and unique trajectory, whereas photons and subatomic particles behave wave-like, traversing simultaneously through multiple trajectories. With this approach, classical trajectories behave like morphisms from the **Set** category, and quantum particles behave like morphisms in **Rel**. See fig.1.

1 Remark. In this setting, the *preimage* of a mapping in $Mor(\mathbf{Set})$ in general doesn't classify as a valid morphism in **Set** yet it does as a valid morphism in **Rel**. See fig.2.

In this same tenor, classical information processing is inherently *irreversible*, whereas quantum information processing is inherently *reversible*. Again surjective maps in $Mor(\mathbf{Set})$ are non-invertible, whereas all relations

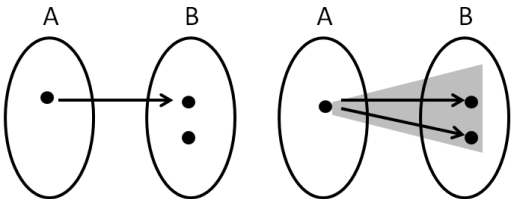


Figure 1: Example of a morphism in **Set** (left) and a morphism in **Rel** (right). The latter can cover all possible set-like maps simultaneously

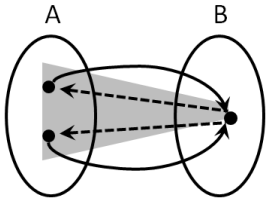


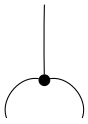
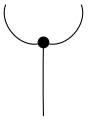
Figure 2: A preimage (dotted line) of a surjective mapping in $Mor(\mathbf{Set})$ (solid line) is a multi-valued function, a valid relation in $Mor(\mathbf{Rel})$.

in $Mor(\mathbf{Rel})$ are invertible by definition. Therefore it is possible to assert that the dagger structure must relate to the preservation of information, as stated in [10].

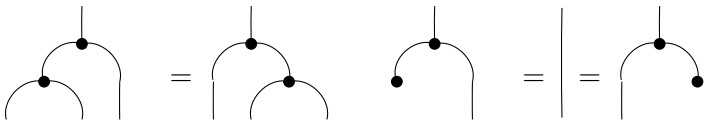
Frobenius structure

Frobenius algebras show up in a variety of contexts including computer science: in circuit diagrams and proof nets. Frobenius structure is an essentially *topological structure*, meaning that the axioms for a Frobenius algebra can be given completely in terms of graphs [5].

A Frobenius structure on *SMC* consists of a multiplication $\mu : A \otimes A \to A$, a unit $\eta : I \to A$, a comultiplication $\delta : A \to A \otimes A$ and a counit $\varepsilon : A \to I$:

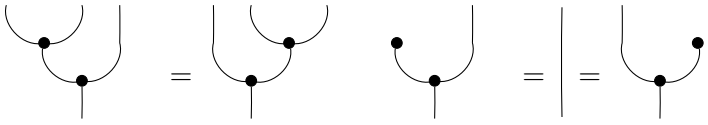
$\mu :=$  $\delta :=$  $\eta :=$  $\epsilon :=$  (9)

which satisfy the associative and unit laws for multiplication:



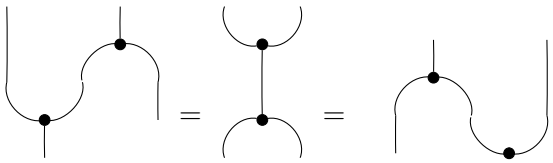
(10)

similarly for co-multiplication:



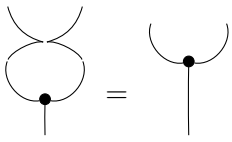
(11)

The Frobenius law is depicted in the following diagram:



(12)

Lastly, the commutativity condition is given by the following diagram:



(13)

3 Probability and Entropy

3.1 Probability theory

Probability is used as an extension of logic to cases where deductions cannot be made [20]. Deductive logic can only deal with certainty, whereas the rules of probability extend logic to include cases where there is uncertainty [21]. *Inductive reasoning* makes use of probability theory as its primary tool for reasoning, much in the same way that deductive reasoning makes use of mathematical logic. Inductive reasoning deals with uncertainty as a common trait.

2 Remark. If the uncertainty contained within a probability is thought of a multi-valued function in the sense that it refers to multiple possibilities, then dagger categories like **Rel** can accommodate probabilistic formalisms like inductive reasoning, whereas categories like **Set** can't, due to the morphisms being mappings and not multi-mappings.

It is possible to perceive the logic behind probability theory with the help of Venn Diagrams. Parting from $A \supset B$ — which is represented in the first Venn diagram in 3 — what can be deduced from A, $\neg B$ and $\neg A$ in logic is shown in Eq. 14. If one parts from the second Venn diagram in fig.3, nothing can be deduced.

$$\begin{array}{ccc} A \supset B & A \supset B & A \supset B \\ A & \neg B & \neg A \\ \hline B & \neg A & B \vee \neg B \end{array} \tag{14}$$

Probability theory has the following axioms.

1. $P(A) \geq 0$ for any event A.
2. $P(U) = 1$. Probability of Universe = 1 *i.e.* some outcome occurs every time you conduct an experiment.
3. If A and B are mutually exclusive events, then $P(A \cup B) = P(A) + P(B)$. In other words, probability is additive over disjoint events.

The rest of the rules of probability can be proved from these axioms. In order to understand the nature of function P, the mathematical concept of *measure* is needed.

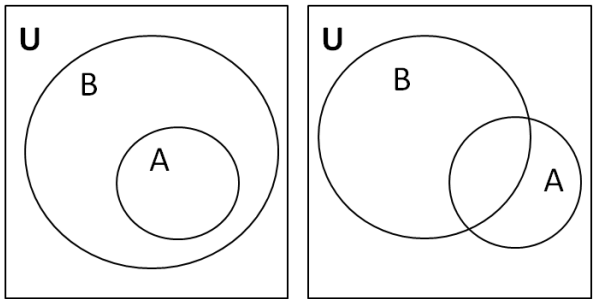


Figure 3: The Venn diagram in the left represents "If A is true then B is true", hence deduction is possible. The one in the right represents "Sometimes A implies B, sometimes B implies A, sometimes neither", making deduction impossible.

Measure

The measure aspect of probability theory can be understood with only a few concepts, the main one being that of σ-algebra.

3 Definition. Let X be a set. A *σ-algebra* of subsets of X is a family Σ of subsets of X such that:

1. $\emptyset \in \Sigma$
2. for every event $E \in \Sigma$ its complement $X \backslash E$ in X belongs to Σ.
3. for every sequence of events E_n, $n \in \mathbb{N}$, its union $\bigcup_{n \in \mathbb{N}} E_n$ belongs to Σ.

Noticeably, X is always a member of Σ due to the $\emptyset \in \Sigma$ condition. For examples, take set $X = \{1, 2, 3, 4\}$. Two possible σ-algebras are:

$$\Sigma = \{\emptyset, X, \{1, 2, 3\}, \{4\}\} \quad , \quad \Sigma = \{\emptyset, X, \{1, 2\}, \{3, 4\}\}$$

A *sample space* X is the set of all possible *outcomes*. *Events* $\{A, B, C, \ldots\}$ are sets of outcomes that conform the σ-algebra Σ. The *probability measure* function

$$P : \Sigma \longrightarrow [0, 1]$$

specifies the likelihood of each event happening. The triple (X, Σ, P) define a *probability space*, which is a specific case of the more general concept of *measure space*.

4 Definition. A *measure space* is a triple (X, Σ, μ) where X is a sample space, Σ is a σ-algebra over X and μ is a *measure*, *i.e.* a function such that:

1. $\mu : \Sigma \longrightarrow [0, \infty]$
2. $\mu(\emptyset) = 0$
3. if the sequence of events E_n is disjoint, then $\mu(\bigcup_{n\in\mathbb{N}} E_n) = \sum_{n=0}^{\infty} \mu(E_n)$

Hence a *probability* measure space is a measure space where the codomain is restricted to the real line between 0 and 1 and $\mu(X) = 1$.

Joint Probability

The joint probability of events A and B is the probability that both events occur simultaneously, on the same repetition of the random experiment, *i.e.* the set of outcomes that are both in event A and event B, the *intersection* $A \cap B$.

If event A and B are *independent*, then $P(A \cap B) = P(A) \times P(B)$, the product of the individual probabilities. If this does not hold then they are *dependent* events.

Marginal probability

The probability of only one of the events in the joint probability setting is called the *marginal* probability. It is found by summing $P(A \cap B)$ and $P(A \cap \neg B)$, *i.e.* by summing its disjoint parts:

$$P(A) := P(A \cap B) + P(A \cap \neg B)$$

Conditional probability

The probability of event B given event A is given by:

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \tag{15}$$

The scaling quality of $\frac{1}{P(A)}$ can be better understood if one considers the *reduced universe*. Given that a certain event A has occurred, all the other outcomes outside A are no longer possible, therefore the reduced universe $U_r = A$ must have probability $P(U_r) = 1$ (see fig. 4). The observation of event A in conditional probability $P(B|A)$ can be thought of as a probability-preserving mapping in the following sense:

$$\text{observe}(A) := U \mapsto U_r, \text{ s.t. } P(U) = P(U_r) = 1$$

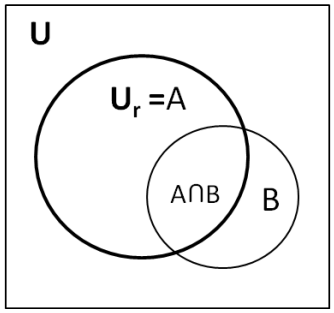


Figure 4: The reduced universe in conditional probability. $P(U_r) = 1$.

Bayesian inference

Bayesian inference [2] makes use of Bayes rule:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{16}$$

which describes *Bayesian inversion*:

$$P(A|B) \mapsto P(B|A)$$

and viceversa. In Eq.(16) $P(A|B)$ is called the *posterior* density; $P(B|A)$ is the sampling density or *likelihood*; $P(A)$ is the *prior* density and $P(B)$ is the *normalizing constant*.

As opposed to a frequentist approach to statistics, Bayesian statistics makes use of subjective information e.g. to determine the prior. Bayes' theorem is a way of updating our uncertainty in the light of new evidence.

3.2 Shannon Entropy

Entropy is a measure of information, although there are other measures [26]. The concept of entropy in information theory was born from the idea of information compression [27][12]. Shannon entropy H of a random variable X is calculated making use of probabilities $p_i \in \mathbb{Q}$:

$$H(X) = -\sum_{i=1}^{n} p_i(x_i) \log_b p_i(x_i) \tag{17}$$

where x_i are distinct outcomes for random variable X, $p_i \in [0, 1]$ are the probabilities for each outcome and b is some basis, normally taken to be 2.

Given a string s, one does not know the probabilities for each symbol s_i in the string, one must calculate (infer) them from the string. Given an amount $H(X)$ of information, there are many strings that represent the exact same information.

3.3 Kolmogorov complexity and Entropy

Shannon's entropy measures the amount of information of a generating source of strings [12]. More precisely, given a random variable X, $S(X)$ measures the amount of information producible by X (Eq.17). Information theory is about processing strings, hence concatenation is an essential structure. Yet Eq.(17) doesn't make use of concatenation.

Kolmogorov complexity [3] measures the information in a string rather than of the string-source. It is defined as the length of the shortest effective (binary) description of X, or generating program. Yet it also provides an absolute and objective quantification of the amount of information of X [4].

Shannon's entropy uses probabilities to quantify the information of a string, hence uncertainty is involved in the conceptualization of this entropy. Kolmogorov complexity is deterministic in the sense that it doesn't loose information: it compresses the string to its minimal syntactic expression $K(x)$, hence *inferring* a source (program). It then makes sense for these two measures to relate in some way, and they do: the *expected* Kolmogorov Complexity equals Shannon's entropy [4].

3.4 Measure preserving functions

Given a measure space (X, Σ, μ) one can represent the σ-family of measurements of each event $E_i \in \Sigma$ as column vectors p with components $p_i := \mu(E_i)$:

$$p = \begin{pmatrix} p_1 \\ p_2 \\ \dots \\ p_n \end{pmatrix} \quad , \quad |p| = n \tag{18}$$

It is possible to define a measure preserving function (*mpf*) between these type of vectors $f : p \to q$. A *mpf* between vectors of the same dimension are given by permutations, which preserve μ. When going from a smaller

to a bigger dimension:

$$f_< : p \longrightarrow q \qquad \text{such that} \quad |p| < |q|$$

one can define $f_<$ as the mapping that copies the values of the domain and fills the rest with $|q| - |p|$ zeros:

$$f_<(p) := \begin{pmatrix} p_1 \\ p_2 \\ \dots \\ p_n \end{pmatrix} \mapsto \begin{pmatrix} q_1 \\ q_2 \\ \dots \\ q_{m-n-1} \\ q_{m-n} \\ q_{m-n+1} \\ \dots \\ q_m \end{pmatrix} := \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \\ p_1 \\ p_2 \\ \dots \\ p_n \end{pmatrix} \quad , \; n < m \tag{19}$$

As mentioned before permutation of the components of the vectors preserves the measure, hence the zeros need not inserted in an orderly manner. Although $|\mu(\Sigma)|$ changed:

$$f_< : |\mu(\Sigma_p)| \mapsto |\mu(\Sigma_q)| \quad f_< : n \mapsto m \tag{20}$$

the measure μ remained the same, as can be confirmed by summing over the components:

$$\sum_i p_i = \sum_j q_j \tag{21}$$

which shows satisfaction for requirement 2 of the measure definition. Requirement 3 is satisfied because *jointness and disjointness* of events $E_i \in \Sigma$ is left untouched by the addition with zero.

One can define the measure preserving function going downwards on the dimension scale as so:

$$f_> : p \longrightarrow q \qquad \text{such that} \quad |p| > |q|$$

$$f_>(p) := \begin{pmatrix} q_1 \\ q_2 \\ \dots \\ q_m \end{pmatrix} \quad \text{where } q_j := \sum_{i \in f_>^{-1}(j)} p_i \tag{22}$$

Since the domain is already a valid measure space and addition of the measures preserves the jointness and disjointness relations, then this definition of *mpf* $f_>$ is a valid one.

In conclusion, a *mpf* $f : p \to q$ between measurements of events $\mu(E_i)$ is defined as:

- a permutation if p and q are of the same dimension
- $f_<$ as defined in Eq.(19) if p is smaller than q
- $f_>$ as defined in Eq.(22) if p is larger than q

3.5 Categories FinProb and FinMeas

Baez et al. [24] define categories **FinProb** and **FinMeas** to characterize entropy where objects are the vectors described in section 3.4 and morphisms are measure preserving functions defined here, but only between dimensions of the same or smaller size. **FinProb** uses probability measures and **FinMeas** uses arbitrary measures.

Compoundness on objects is given by the weighted direct sum:

$$p \otimes q := \lambda p \oplus (1 - \lambda)q$$

with $\lambda \in [0, 1]$. Compoundness on morphisms $f : p \to p'$ and $g : q \to q'$ is defined as:

$$f \otimes g := \lambda f \oplus (1 - \lambda)g : \lambda p \oplus (1 - \lambda)q \longrightarrow \lambda p' \oplus (1 - \lambda)q'$$

Baez gives a way to characterize the entropy of a single probability measure as the *change in entropy of the unique measure-preserving function onto the one-point space*.

3.6 Classical Bayesian Graphical Calculus

Coecke et al. [23] define a category to accommodate classical Bayesian inference and entropy, where objects are natural numbers and $m \times n$ positive valued matrices, composition is matrix product and tensor product is the matrix tensor product.

Representations of Frobenius product and coproduct are given analogous to the *mpf* f defined in this section, yet constrained to *probability*

(rather than *arbitrary*) measures, and only for cases where $|f_<(p)| = |p|^2$ and $|p| = |f_>(p)|^2$.

In this same work an abstract characterization of Shannon entropy is given, namely as inner products between two alternate representations of the diagrammatical calculus. This same product gives the joint, conditional and marginal entropies. Besides the aforemention representation in terms of column vectors p with components measurements of events $p_i := \mu(E_i)$, a second representation is given in terms of column vectors s with components negative logarithms of probabilities $s_i := -ln(p_i)$, hence the Shannon entropy S as an inner product is :

$$S(X) := (p_1, p_2, \ldots, p_n) \begin{pmatrix} s_1 \\ s_2 \\ \cdots \\ s_n \end{pmatrix} = \tag{23}$$

$$(s_1, s_2, \ldots, s_n) \begin{pmatrix} p_1 \\ p_2 \\ \cdots \\ p_n \end{pmatrix} = \sum_i^n p_i \, s_i = \sum_i^n p_i \ln p_i \tag{24}$$

4 Suggested directions of work

4.1 Entropy related category

5 Suggestion. Although both Baez's and Coecke's categories in [24] and [23] are apparently different, it is conjectured in this report that *there is a more general category that includes both formalisms* which should fall within the categorical quantum mechanics framework.

Baez et al. do not reference the symmetric monoidal categories formalism, nor the corresponding diagrammatic calculus. Since the work there focuses only on mappings on to the one-point space, an increasing dimension map like $f_<$ is not identified. Since graphical languages are sound with respect to categories [9], it should be possible to define a graphical calculus for the category conjectured.

4.2 The functorial nature of entropy

In **Set** the category of sets, the cardinality $|-| : X \longrightarrow \mathbb{R}$ of a hom-set $\mathcal{C}(A, B)$ is given by :

$$|\mathcal{C}(A, B)| = |B|^{|A|} \tag{25}$$

That said, the logarithm operation is a way of *inferring* the cardinality of the domain given the cardinalities of both the codomain and the hom-set. In other words, the logarithm calculates the *preimage* cardinality.

6 Suggestion. It is then possible to propose that whatever information — if any — the logarithm infers about the domain of the hom-set, should be contained within a hom-set going in the opposite direction (from codomain to domain) in some category. Considering remark 1 one can conclude that since **Set** and **Rel** are related by a functor— namely the powerset functor— the categorical interpretation of logarithm should be dependant of this powerset functor. Moreover *the categorical interpretation of entropy* should depend on the powerset functor, since the definition of entropy makes use of the logarithm operator.

$$\log_{|B|}|\mathcal{C}(A, B)| = |A| \tag{26}$$

7 Suggestion. As per remark 2, if the morphisms of a category abstract probabilities as part of their definition, then they share the dagger structure

with quantum-like categories like **Rel** just as the preimage of mappings do. Then singling out of classical categorical structures from quantum one should be done within a dagger category.

4.3 A topology for classical compression

Classical information theory [12], or more precisely, lossless compression of strings in classical information has a topological structure if one draws a graph of the information flow for copying/uncopying (fig. 5) and swapping (fig. 6) of bits. An arbitrary string can be created an *co-created* (*i.e.* traced back to its alphabet $\Sigma = \{0, 1\}$) in this way. See fig. 7.

If one considers entropy as a measure of how *mixed-up* a string is —as it actually is— it is possible to assert that co-creation codifies entropy in some way, just as lossless data compression is an estimate of entropy of a block of data [13].

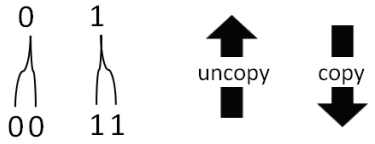


Figure 5: Bit copying from the binary *alphabet* $\Sigma = \{0, 1\}$ to a binary *string*.

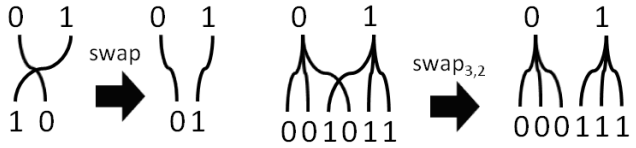


Figure 6: Bit swapping

As an example, run-length encoding (RLE) —used mainly for compressing images— is a lossless data compression method that uses the logic described in fig. 7, namely going from step 2 to step 3′. This method works well when the alphabet of the string to be compressed is small and well known *a priori*, which is reflected graphically by the connecting wires between alphabet an string in *pre-processing* step 0.

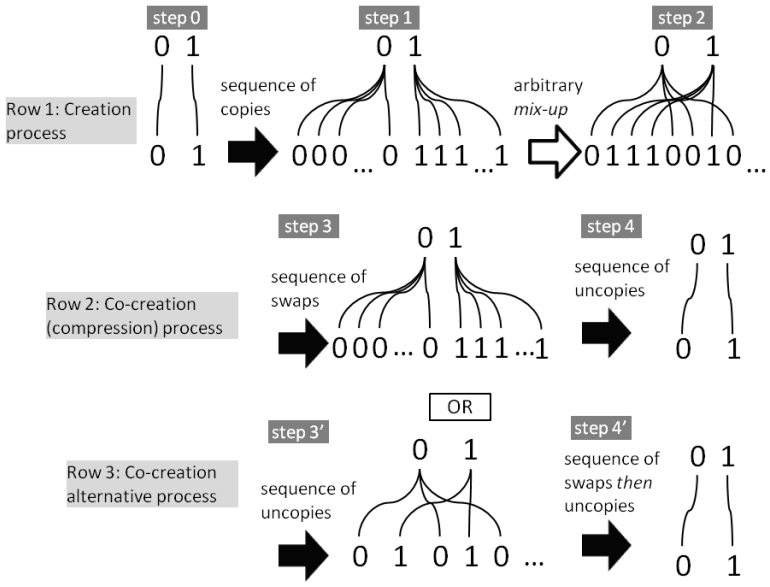


Figure 7: Creation and co-creation of an arbitrary bit string through copying, uncopying and swapping. The white arrow indicates an unknown source of data manipulation that involves a sequence of unknown swappings.

Next is an example of RLE compression on a string representing a row of *b*lack and *w*hite in a fax machine:

$$\text{RLE}(\text{"WWWWWBBBBBBWWWBBBWBWWWWWBBBBB"}) = \texttt{5W6B3W3B1W1B5W5B} \qquad (27)$$

It is possible to use the graphical setting to include arbitrary (finite) alphabets Σ e.g. $\Sigma = \{a, b, c, \ldots\}$. Binary strings would then generalize to *sentences* in *formal languages* $L = \Sigma^*$. The theory of formal languages [14] possesses its own structures — automata and generating grammars— that hierarchize classical information processing including the computability limit, and identifies where the strings of symbols begin to be understandable only by a human being, *i.e.* strings that belong to *natural language* [15]. Formal languages and classical information theory have natural language as a common research topic [16].

Continuing in this same venue, non-statistical methods like RLE — the so-called *dictionary based methods* [17][18]— require a complete knowledge of the tokens in the string, which conform the *dictionary i.e.* the alphabet Σ of tokens. Dictionaries can be static or non-static, the former meaning is fixed *a priori* and continues fixed throughout the compression process.

The compression depicted in fig.7 has a static dictionary. The mere *need* of a dictionary in dictionary-based methods reflects the *connecting* topology needed at all times in the graphical representation of compressing and co-compressing.

Shannon himself makes use of the need of a connecting topology when defining his compression method in his seminal paper [12], now known as the Shannon–Fano Method. The first step of this compression algorithm is to obtain the list of symbols in the string and calculate the frequency of each, that is, to determine the dictionary or alphabet.

8 Suggestion. Lossless information compression can be thought of as a surjective map. By definition, the compressed string can re-generate (co-compress) the original string. This re-generation is a multi-map. A dagger category should be able to accommodate compression and co-compression as morphisms and hence relatable categorically to quantum phenomena (cf. remarks 1 and 2). In this same spirit, any categorical structure — e.g. a functor— that relates lossy and lossless compression should relate to comparable lossy (e.g. quantum measurement) and lossless (e.g. quantum gate) quantum processes.

5 Documentation for Quantomatic software

Quantomatics is a 2-tier system for reasoning about graphical languages within the formalism of monoidal categories. In the "engine" or core — the logic tier — the functional language ML is used. In the graphical user interface — the presentation tier — the object-oriented language Java is used.

Currently Quantomatic software has a limited amount of documentation, mainly program source code listings. The documents that are partially or totally included already in Quantomatic are indicated with an asterisk $*$. As it grows, a system needs documentation so that users, developers and architects know how to operate, update/mantain and re-design it, respectively.

It is proposed in this report to add the full set of standard user an system documentation [28]. Each of these is described next.

5.1 User documentation

For each of the following five types of user document there is a corresponding user:

1. *Functional description* *.- This document describes the overall functionality of the system. The so called *system evaluator* user should be able to decide whether to use the software or not from this document.

2. *Installation document* *.- This is intended for *system administrators*. It should provide details of how to install the system and any additional software or hardware, as well as how to start the system.

3. *Introductory manual*.- This is intended for *novice users*. It should provide an informal introduction to the system, how to get started as well as describe its normal usage.

4. *Reference manual*.- This is intended for *experienced users*. It should describe the system facilities and their usage, as well as how to recover from detected errors. Its description should be complete and formal.

5. *System administrators guide*.- Beside the installation document, system administrators need to know any configurations not modifiable by normal users. These configurations could involve, for instance, network-related permissions or permissions on other systems which

the current system interacts with. Also it should describe how to recover from less-well known exceptions.

5.2 System documentation

This documentation includes all the documents that describe the system itself, at many different levels:

1. *Requirements*.- Here the requirements made by the end user are explained. The language in this document involves mainly the *business logic* rather than any logic from within the system. It describes the *status quo ante* the system existed, and also where and how the system is to be implemented.

2. *Architecture and design*.- This document can be seen as an intermediate step between the requirements and the technical details of the system. It describes *how* it has been decided (usually by the *architect* of the system) that the system will fulfill each of the requirements, emphasizing the motivation behind the decisions. This document is low on technicalities and high on explanation.

3. *Technical documentation*.- Here are contained the details of the code, namely the description of what each module does, the types of their inputs and outputs and how each of these modules interact. There are three levels for technical documentation:

   (a) *Program*.- Both an overall description and a description of the architecture of each program. In Quantomatic, this amounts to a description of — for instance — what each folder and each file in the *core* folder is for. See fig. 8.

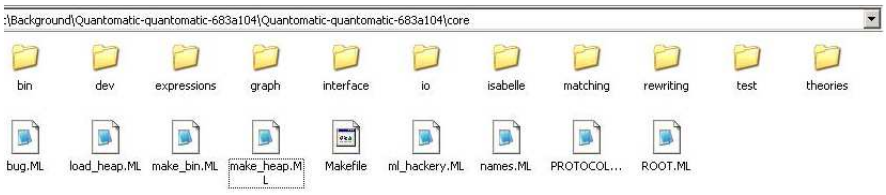
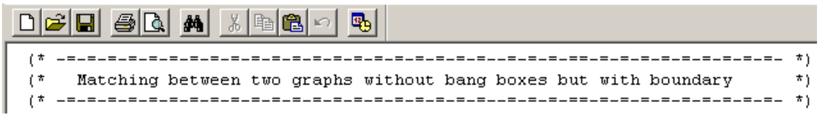


Figure 8: Folder structure of the *core* folder

   (b) *Component* *.- Both an overall description and a description of the architecture of each component. In Quantomatic this amounts

to a description on the header of each of the *.ml* files. As indicated by the asterisk, there is currently some documentation of this sort. See fig. 9.



```
(* -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=- *)
(*   Matching between two graphs without bang boxes but with boundary       *)
(* -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=- *)
```

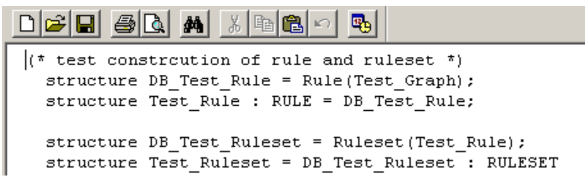
Figure 9: A piece of existing component code documentation in file *core\matching\match.ML*

(c) *Source code* *.- Descriptions of each function and sub-procedure, as well as important variables and complex sections of code. For instance, in fig. 10 an example of current source code documentation (between the "(∗" and the "∗)" symbols) is shown.

```
(* instantiate rule with a match *)
fun instantiate_rule m r =
    Rule.mk (Match.inst_pat_graph m (Rule.get_lhs r),
             Match.inst_replacement_graph m (Rule.get_rhs r));
```

Figure 10: A piece of existing source code documentation in file *core\rewriting\ruleset_rewriting.ML*

4. *Validation documents* *.- These describe how each program is validated —*e.g.* by giving the values used for testing— and how these tests match the requirements. Currently there are some tests in Quantomatic, an example of which can be seen in fig. 11.



```
(* test constrcution of rule and ruleset *)
  structure DB_Test_Rule = Rule(Test_Graph);
  structure Test_Rule : RULE = DB_Test_Rule;

  structure DB_Test_Ruleset = Ruleset(Test_Rule);
  structure Test_Ruleset = DB_Test_Ruleset : RULESET
```

Figure 11: An existing test in file *core\theories\ruleset-test.ML*

5. *System maintenance guide*.- Here known problems are described, as well as pending solutions (the so called *to-do's*) to be implemented. Fig. 12 shows an existing current example of this in Quantomatic.

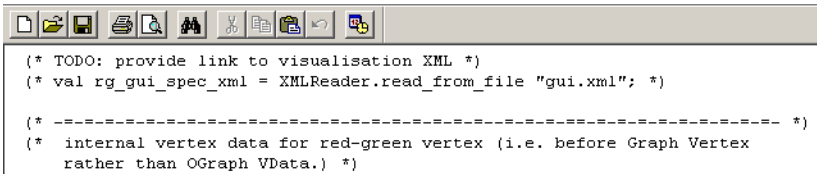


Figure 12: A pending implementation in file *core\theories\vertex-test.ML*

References

[1] S. Abramsky and B. Coecke (2004) *A categorical semantics of quantum protocols*. In: Proceedings of 19th IEEE conference on Logic in Computer Science, pages 415-425. IEEE Press. arXiv:quant-ph/0402130.

[2] Bayes, Thomas, and Price, Richard (1763). *An Essay towards solving a Problem in the Doctrine of Chance. By the late Rev. Mr. Bayes, communicated by Mr. Price, in a letter to John Canton, M. A. and F. R. S.*. Philosophical Transactions of the Royal Society of London 53 (0): 370418. doi:10.1098/rstl.1763.0053

[3] A. N. Kolmogorov (1968). *Three approaches to the quantitative definition of information*, International Journal of Computer Mathematics, 2:1-4, 157-168.

[4] Peter Grüunwald and Paul Vitányi (2010). *Shannon Information and Kolmogorov Complexity*. Submitted to IEEE Trans Information Theory. arXiv:cs/0410002v1

[5] J. Kock (2004)*Frobenius algebras and 2D topological quantum Field theories*. Cambridge University Press.

[6] M. S. Leifer (2006)*Quantum dynamics as an analog of conditional probability*. Physical Review A 74, 042310. arXiv:quant-ph/0606022

[7] P. Selinger (2007) *Dagger compact categories and completely positive maps*. Electronic Notes in Theoretical Computer Science **170**, 139163.

[8] P. Selinger (2008) *Finite dimensional Hilbert spaces are complete for dagger compact closed categories*. Electronic Notes in Theoretical Computer Science **270**, 113119.

[9] P. Selinger (2009) *A survey of graphical languages for monoidal categories*. arxiv.org/pdf/0908.3347.pdf

[10] C. Heunen (2009). *Categorical Quantum Models and Logics*. D.Phil. Thesis, Amsterdam University.

[11] C. Heunen and J.Vicary (2012). *Lectures on Categorical Quantum Mechanics*. Computer Science Department. Oxford University.

[12] C.E.Shannon (1948) *A Mathematical Theory of Communication*.Mobile Computing and Communications Review, Volume 5, Number I.

[13] T. Schürmann and P. Grassberger (2002).*Entropy estimation of symbol sequences*. CHAOS Vol.6, No. 3(1996) 414-427. arXiv:cond-mat/0203436v1.

[14] Brookshear, Glenn (1989) *Theory of Computation: Formal Languages, Automata, and Complexity*

[15] Chomsky, Noam (1956) *Three models for the description of language*. IRE Transactions on Information Theory (2): 113124.

[16] F. Pereira (2000) *Formal grammar and information theory: Together again?* in Philosophical Transactions of the Royal Society **358**, 1239-1253.

[17] D. Salomon (2000). *Data Compression. The Complete Reference.* Second edition. Springer Verlag.

[18] K. Sayood (2000). *Introduction to Data Compression*. Second edition. Morgan Kaufmann publishers.

[19] S. Abramsky and N. Tzevelekos (2011). *Introduction to Categories and Categorical Logic* in New Structures for Physics, B. Coecke (ed). Lecture Notes in Physics Vol. 813, pages 3–94, Springer-Verlag 2011.

[20] E.T. Jaynes (2003). *Probability theory. The logic of science.* Cambridge University Press.

[21] W.M. Bolstad. *Introduction to Bayesian Statistics.* Second edition. John Wiley and Sons, Inc.

[22] A. Joyal and R. Street (1991) *The Geometry of tensor calculus I*. Advances in Mathematics **88**, 55112.

[23] Bob Coecke and Robert W. Spekkens (2011) *Picturing classical and quantum Bayesian inference*. arXiv:1102.2368v1

[24] John C. Baez, Tobias Fritz, and Tom Leinster (2011) *A Characterization of Entropy in Terms of Information Loss*. arXiv:1106.1791v3

[25] Samson Abramsky and Nikos Tzevelekos (2009) *Introduction to Categories and Categorical Logic*

[26] Martin B. Plenio and Shashank Virmani (2006) *An introduction to entanglement measures*. arXiv:quant-ph/0504163v3

[27] R.V. Hartley (1927) *Transmission of Information*. International Congress of Telegraphy and Telephony. Lake Como, Italy. September 1927.

[28] Ian Sommerville (2001). *Software Documentation*. Lancaster University, UK.