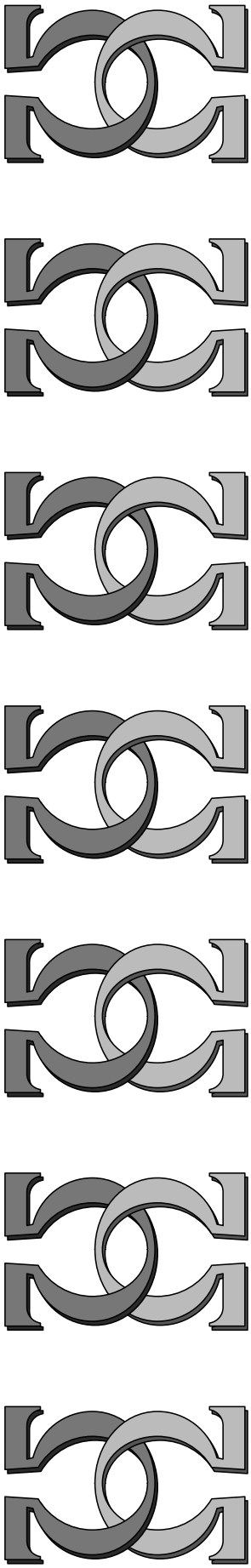


CDMTCS
Research
Report
Series

Bicategorical Semantics for Nondeterministic Computation

M. Stay[1], **J. Vicary**[2]
[1]University of Auckland, NZ
[2]National University Singapore, Singapore

CDMTCS-439
May 2013

Centre for Discrete Mathematics and
Theoretical Computer Science

Bicategorical Semantics for Nondeterministic Computation

Mike Stay[1]

Department of Computer Science, University of Auckland, New Zealand
Biosimilarity LLC, Seattle, USA

Jamie Vicary[2]

Centre for Quantum Technologies, National University of Singapore, Singapore
Department of Computer Science, University of Oxford, UK

Abstract

We present a topological bicategorical syntax for the interaction between public and private information in classical information theory. This allows high-level graphical definitions of encrypted communication and secret sharing, including a characterization of their security properties. This analysis shows that these protocols have an identical abstract form to the quantum teleportation and dense coding procedures, giving a concrete mathematical analogy between quantum and classical computing. Specific implementations of these protocols as nondeterministic classical procedures are recovered by applying our formalism in a symmetric monoidal bicategory of matrices of relations. `http://www.elsevier.nl/locate/entcs`.

Keywords: Categories, security, geometry

1 Introduction

1.1 Overview

This paper describes a bicategorical language for reasoning about cryptographic processes in classical computation. Bicategories can be thought of as generalizations of monoidal categories, mathematical structures which have already found significant application to quantum computation [1,13]. In this work, we describe a monoidal bicategory **2Rel**, and describe how its different layers of structure can be used to describe public information, private information and nondeterministic classical processes in a natural way.

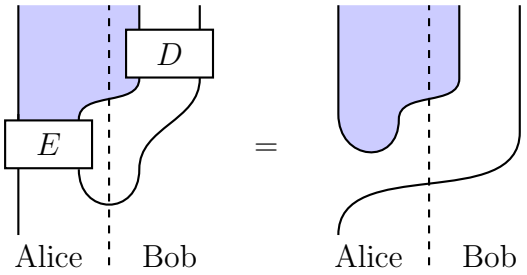
Bicategories have a well-known sound and complete graphical calculus [7,12] involving points, lines and regions, which we make use of almost exclusively for presenting our formalism and proving our results. The reason is that the basic axioms we impose have a direct topological interpretation which are cumbersome from an algebraic perspective, but which the graphical calculus naturally absorbs and makes trivial. Why this should be is far from clear; it provides evidence of a

[1] Email: `msta039@aucklanduni.ac.nz`
[2] Email: `jamie.vicary@cs.ox.ac.uk`

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: `www.elsevier.nl/locate/entcs`

deep relationship between topological structures and the theory of information [**?**] which deserves to be explored further.

A diagram in our calculus can he interpreted as a history of computational events, in which time flows from bottom to top. To use the terminology of physics, they are 'spacetime diagrams' for our computation. As an example of our notation, the following diagram represents an encrypted communication protocol making use of a one-time pad:



(1)

A full description of the components of these diagrams must wait for Section 3, but we can summarize the basic features here. The shaded regions represent public information, and the vertices represent computational processes. Lines represent computational systems which carry information: if the line borders a shaded region then the associated system carries a copy of the associated public information, but otherwise it carries private information. The dashed vertical lines, which are not a part of the mathematical formalism, imply a separation of the components involved between Alice and Bob which is convenient for our interpretation.

In the left-hand diagram, the bottom-left line indicates a system held by Alice which stores some private information. This information is the plaintext that we will encrypt. The bowl-shaped curve represents the first nontrivial process—the nondeterministic creation of a one time-pad—which is shared between the two parties. The next vertex E represents encryption, a process which takes as input the plaintext and a copy of the one-time pad, and produces public information. A copy of that public information is then transferred to Bob, and is fed into the decryption process D along with a copy of the one-time pad. The line emerging at the top-right represents the decrypted plaintext. The result of this entire composite procedure is given by the right-hand diagram: the original plaintext is simply transmitted unaltered, and the public information is disconnected, meaning it is uncorrelated with the plaintext. The equality between these two sides says that the encryption-decryption process is successful.

In this description, and throughout this paper, we freely interpret the underlying mathematical structures in a way which is intended to make our formalism easier to understand at an intuitive level. However, this interpretation it is secondary to the basic mathematical content of our theory, which is crisp and unambiguous. The motivating result is Theorem **??**, which states that solutions to equation (1) in **2Rel** such that E is kernel-free correspond exactly to implementations of classical encrypted communication via a one-time pad. A variety of security properties of this procedure are also provable using our techniques.

The form of equation (1) corresponds exactly to the equation for *quantum teleportation*, as described in the bicategorical approach to quantum information [13]. One of the most important procedures in quantum theory, and yet uncovered only relatively recently [2], quantum teleportation is a procedure whereby two parties who share pre-existing quantum entanglement can transmit quantum information between them, by communicating only classical information. A strong analogy to classical encrypted communication can be drawn: two parties who share a pre-agreed secret key can transmit secret information between them, by communicating only public information. This analogy has already been recognized by several authors [4,8], and our work provides a new formal mathematical basis for it.

The monoidal bicategory **2Rel** which forms the basis for our constructions is described in Section 2. In Section 3 we gives the details of our graphical formalism, and Section 4 contains an application of our techniques to encrypted communication and secret sharing procedures, including an analysis of their security properties.

2 The Bicategory of Matrices of Relations

2.1 Introduction

Bicategories, also known as weak 2-categories, are algebraic structures akin to higher-dimensional directed graphs, and play an important role in modern mathematics. They are built from vertices, edges going between vertices, and surfaces going between edges, which are called 0-cells, 1-cells and 2-cells respectively. They also carry algebraic structure, allowing edges to be composed along a common vertex, and surfaces to be composed along a common edge. These composition operations are required to be unital and associative in a suitable fashion. Though elegant, the full definition is lengthy and we omit it here; see [3] for a good introduction.

In this section we describe the bicategory **2Rel** which will be the target for our constructions. It can be presented quite simply in terms of finite sets and partitions: 0-cells are finite sets, 1-cells are finite sets partitioned by their source and target sets, and 2-cells are relations getting along with the partitioning. All the structure of a bicategory can be defined quite naturally on these structures. We give a careful definition below, although for must purposes an intuitive understanding of the structure is quite adequate.

2.2 Construction

The n-cells of **2Rel** are defined in the following way. **0-cells** are finite sets, denoted $S, T, \ldots$. A **1-cell** $A : S \to T$ is a family of finite sets $A_{t,s}$ indexed by $s \in S$ and $t \in T$. For 1-cells $A, B : S \to T$, a **2-cell** $\rho : A \Rightarrow B$ is a family of relations $\rho_{t,s} : A_{t,s} \to B_{t,s}$ indexed by $s \in S$ and $t \in T$.

To demonstrate that these form a bicategory we first observe that for each pair of 0-cells S and T, the 1-cells $S \to T$ and the 2-cells between them form a category in a straightforward way, under ordinary relational composition of 2-cells. Identity 1-cells $\mathrm{id}_S : S \to S$ are chosen as the family $\delta_{s,s'}$, which is defined as the 1-element

set if $s = s'$ and the 0-element set otherwise. Horizontal composition is a family of functors

$$\circ : \mathrm{Hom}(S,T) \times \mathrm{Hom}(T,U) \to \mathrm{Hom}(S,U) \tag{2}$$

for each ordered triple S, T, U of 0-cells. On 1-cells $A : S \to T$ and $B : T \to U$, we define this as

$$(B \circ A)_{u,s} = \coprod_{t \in T} B_{u,t} \times A_{t,s}. \tag{3}$$

This extends to 2-cells in a natural way.

The final pieces of structure are the structural 2-cells of the bicategory. For each family of composable 1-cells $A : S \to T$, $B : T \to U$ and $C : U \to V$ we require an invertible 2-cell

$$\phi_{A,B,C} : (C \circ B) \circ A \Rightarrow C \circ (B \circ A). \tag{4}$$

Writing out the source and target using definition (3), we see that ϕ is built from a family of isomorphisms $\coprod_t \left((\coprod_u C_{v,u} \times B_{u,t}) \times A_{t,s} \right) \simeq \coprod_u \left(C_{v,u} \times (\coprod_t B_{u,t} \times A_{t,s}) \right)$, each of which can be constructed canonically. Unit 2-cells can be straightforwardly defined, and it is then straightforward to show that the required pentagon and triangle equations commute.

In fact, **2Rel** can be given the structure of a symmetric monoidal bicategory, for which the tensor product of two 0-cells is their cartesian product as sets. For full details see [11], in which an equivalent bicategory **Mat**(**Rel**) is described. According to this structure, the monoidal unit 0-cell is the 1-element set.

Endomorphisms in **2Rel** have the following property, which will useful later.

Lemma 2.1 *In* **2Rel***, if 2-cells* σ *and* τ *are endomorphisms, then* $\sigma \circ \tau = \mathrm{id}$ *implies* $\tau \circ \sigma = \mathrm{id}$*.*

Proof. Suppose at first that σ and τ are relations on a finite set S. Then if $\sigma \circ \tau = \mathrm{id}_S$, there must be at least one $y \in S$ such that $(x, y) \in \sigma$ and $(y, x) \in \tau$. But then there must be exactly one such y, otherwise we could not ensure that $x \neq z \in S$ implies $\nexists y \in S$ with $(x, y) \in \sigma$ and $(y, x) \in \tau$. It follows that σ and τ are graphs of mutually inverse bijections, and so in particular $\tau \circ \sigma = \mathrm{id}_S$ also. The hypothesis follows immediately. □

3 Private and Public Information

3.1 Private information

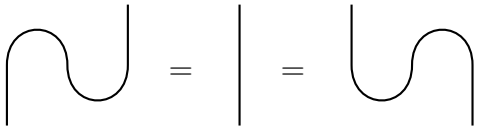
We assume that a single, isolated computational system is located at any moment at a single point in space, so that over time its history traces out a line in spacetime. In the absence of shaded regions, our diagrams are simple representations of such a scenario, with vertices representing points at which multiple systems interact. This part of our graphical formalism is the standard notation for morphisms in symmetric monoidal category [10]. Our string diagrams are valued in **Rel**, the symmetric monoidal category of finite sets and relations. This forms the endomorphisms of

the 1-element set considered as a 0-cell of **2Rel**. In this way the ordinary string diagram calculus for **Rel** embeds into our surface diagram calculus for **2Rel** in a natural fashion. We will interpret an object of **Rel** as representing a classical computational system, with a particular finite set of internal states. Morphisms are arbitrary nondeterministic computational dynamics, transforming states of the domain into states of the codomain.

Using this formalism, we call a system *self-dualizable* if it can be equipped with a unit morphism and a counit morphism

(5)

satisfying the following equations, called the snake equations:



(6)

The unit morphism represents a way to create two systems together, and the counit morphism represents a way to eliminate two systems together. In essence, the snake equations say that we can choose these structures in a way that represents the topology of a string.

We say that the unit and counit morphisms *witness* the self-duality. In **Rel** every object A is self-dualizable, with the unit morphism $\eta : 1 \to A \times A$ given canonically by $\eta = \bigcup_{a \in A} \big(\bullet, (a, a)\big)$, and with the counit given by the converse of this relation. Every unit and counit map is of this form, up to isomorphism. The unit morphism η represents a nondeterministic processes whereby a pair of systems are prepared, each in the same state $a \in A$, such that any pair (a, a) can arise in this way. We can interpret this computationally as a *one-time pad distribution procedure.* It is deeply interesting that this should arise solely from the requirements of the snake equations (8).

For a set A there is a unique relation of type $A \to 1$ which is *total*, meaning that every element of $A \times 1$ is in the relation. It can be characterized abstractly as the unique morphism of this type with zero kernel [**?**], and is interpreted as eliminating the system A without halting the computation. It has a converse relation, which represents the process of creating the system A in an arbitrary state. We denote these morphisms graphically in the following way:

(7)

These are related by the unit and counit morphisms (7) witnessing self-dualizability

via the following equations:

(8)

Each of these has a natural interpretation in **Rel** terms of nondeterministic classical computation: the first set of equalities (10) say that if you nondeterministically create shared keys and then delete one of the keys, the remaining key is uniformly random; while the second set say that if you have a given key, it is always possible that another key produced nondeterministically might match it.

3.2 Public information

Public information is represented in our formalism by regions. The intuitive idea is that public information is stored by a family of systems, each strongly correlated with their neighbours. Each individual system sweeps out a line through time, so the family sweeps out an entire region:

(9)

We shade this region in blue to indicate its special interpretation. The interpretation as public information derives entirely from the fact that the information is now available at many locations, each of which store an identical copy. So public information is more accessible than private information, but as a consequence less mutable, since to change its value every representative would have to be modified. This mathematical model gives a reasonable abstraction for real-world public data services, such as the Domain Name Service, which stores public information redundantly on many independent computers.

As mentioned in the introduction, we are making use here of the standard graphical calculus for monoidal bicategories. Shaded regions correspond to 0-cells of the bicategory, and unshaded regions correspond to the monoidal unit 0-cell.

Manipulations of our public data are described by a small number of basic components. Copying and comparing public data are represented as follows:

(10)

In the first of these one region splits into two, each carrying a copy of the original public information. In the second two regions fuse to become one, which carries the same information as the initial regions in the event that the data in both initial

regions is the same. Otherwise, the computation halts; in this sense, this second vertex can be interpreted as the *assertion* that two data values compare successfully.

We can also represent deletion and uniform creation of public information:

(11)

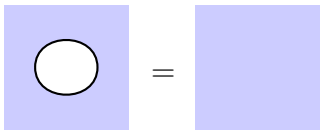
In the first of these, a single region is eliminated, deleting the information it stores. In the second, a single region is created, which we interpret as holding any possible value of the public information in a nondeterministic sense.

As with the bicategorical syntax for quantum information [13], in order to support their interpretations, we require these copying, deleting, comparison and uniform creation components to satisfy certain equations. They are topological, in that they amount to saying that any composite diagram built from them is determined only by its connectivity.

(12)

(13)

(14)



(15)

Each of these equations is consistent with the interpretation we give to the basic components (12)–(13). For example, the first equality labelled (14) asserts that copying public information and then deleting the new copy results in the identity; the first equality labelled (16) represents the fact that exchanging public information and then comparing gives the same result as simply comparing; and equation (17) states that copying public information and then immediately comparing yields the identity. In terms of higher category theory, equations (14)–(15) state that the region boundaries are ambidextrous adjoints [7], and equations (16)–(17) state that the associated Frobenius algebra is special and commutative.

The following theorem demonstrates that these structures are easy to work with in **2Rel**.

Theorem 3.1 *Every 0-cell in* **2Rel** *carries structures* (12)–(13) *satisfying equations* (14)–(17) *in an essentially unique way.*

Proof sketch. A 1-cell $A : 1 \to S$ is determined by an S-indexed family of finite sets A_s, and its isomorphism class is determined by the cardinalities of those sets. Every such 1-cell has an ambidextrous adjoint, meaning precisely that essentially unique values can be given for structures (12)–(13) that satisfy equations (14)–(15). The result is a Frobenius algebra structure [7], which will be commutative exactly when each of the finite sets A_s has cardinality 1, which satisfies the equations labelled (16). The resulting structures automatically satisfy equation (17). □

3.3 Interacting private and public data

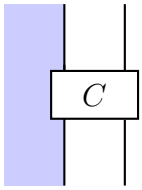
Interesting phenomena arise when we study interactions between public and private information. There are three basic forms that such an interaction can take: converting private data to public data; converting public data to private data; and using public data to modify private data.

Conversion processes between public and private data take the following forms:

(16)

Here P is a publication process converting private data into public data, and S is a sampling process converting public data into private data. Their interpretations rests entirely on their types; there are no equations which we require them to satisfy. These processes need not be deterministic, or invertible, in general. However, it will later be convenient to require them to be kernel-free, meaning that they do not halt on any input.

The final type of process we introduce is the controlled computation, which performs an operation on private data in a way which depends on the value of some public data:



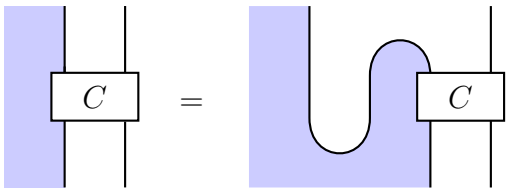
(17)

Such an operation can modify the private data, but not the public data.

Lemma 3.2 *A controlled computation cannot modify public data.*

Proof. We can use the topological behaviour of public information to rewrite our

controlled computation vertex C in the following way:



(18)

In this form it is clear that the public data is not modified, since it is explicitly copied before C is implemented. □

This result fits well with our intuition about public data as a being carried by a large, correlated family of systems. To change the value of the public data would require modifying all of these systems, but the process C only has access to a restricted subset, as made explicit by the open boundary on the left-hand side of the diagram. If the proof given here seems too slick to have any real content, that is because this is really a lemma about our *interpretation* of these mathematical structures, rather than about those structures themselves.

4 Modelling Cryptographic Procedures

4.1 Encrypted communication

Suppose Alice is sending an encrypted message to Bob. We use a 2-cell E to represent Alice's encryption process, which relates the private plaintext P and the private key K to the public ciphertext C. Bob's decryption process is a 2-cell D that relates the public ciphertext and private key to the same ciphertext and a private plaintext. We represent these 2-cells graphically in the following way:

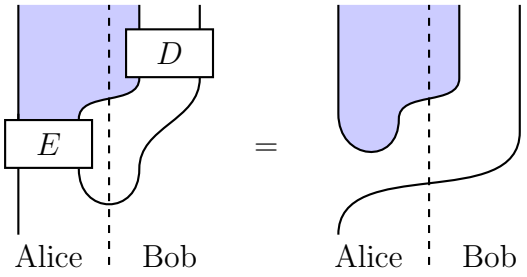
(19)

While encryption and decryption are deterministic, key generation is not. We represent key generation as a special 2-cell, the curried identity relation on the set of keys K:

This is the unit morphism for a self-duality on K, as described in Section 3.

Using our topological language, we can express correctness of encrypted

communication in the following way:



(20)

This is the same 2-dimensional equation as that used in [13] to describe quantum teleportation. The encryption step takes the place of the measurement operation, and the decryption step takes the place of the controlled unitary correction. The ciphertext takes the place of the classical bits transmitted from Alice to Bob. This provides an intuition for why no faster-than-light communication is possible with entangled particles: Alice and Bob merely share a quantum variant of a one-time pad, and the actual encoded message must still be sent at some finite speed.

The following theorem is the motivation for our entire theory.

Theorem 4.1 *Solutions to* (**??**) *in* **2Rel** *for which E is kernel-free correspond exactly to implementations of classical possibilistic encrypted communication by a one-time pad.*

Proof sketch. The result is established by construction. From a description of a space of messages, a family of one-time pads, and encryption and decryption procedures, a solution to (**??**) can be directly constructed, and the inverse procedure is also possible. □

We illustrate our proof sketch with the simplest nontrivial implementation of the protocol: the encrypted communication of a single bit. We can describe concretely the values of E, D and the key creation step η as 2-cells in **2Rel** which correspond to this scenario. We choose $C = P = K$ to be the 2-element set, and we define the 2-cells as follows:

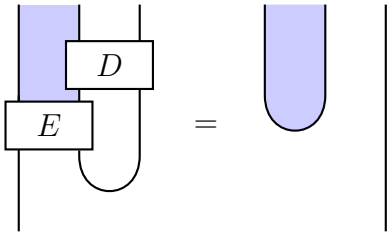
$$E = \left(\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \right) \qquad D = \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \right) \qquad \eta = \left(\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right) \tag{21}$$

Here E is a matrix containing a single relation from a 4-element set to a 2-element set, which is exactly the multiplication operation for the group $\mathbb{Z}_2$; D is matrix of invertible single-bit operations to apply depending on which bit is published at the encryption step; and η is a matrix with a single entry, the relation representing nondeterministic creation of the pair of keys $(0, 0)$ or $(1, 1)$. Using the definition of the bicategory **2Rel**, it can be checked that these values satisfy equation (**??**).

4.2 Security properties

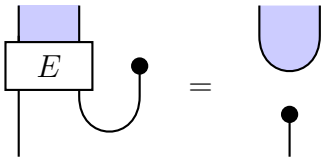
Our formalism allows us to prove results about the protocol based on only its abstract form, and hence draw conclusions which will apply for any implementation. Many of these results can be naturally interpreted as describing security properties. The generality of our results means that we can presume an attacker with arbitrary computational abilities, as long as their actions are constrained to those that can be described using our formalism (i.e. arbitrary nondeterministic processes.)

To focus on its algebraic properties, we simplify equation (**??**) topologically in the following way:



(22)

The first property we will examine can be considered the primary security property for encrypted communication:



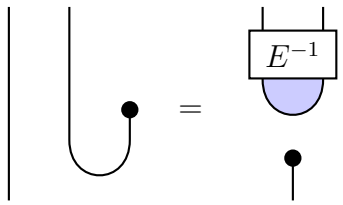
(23)

This says that if we encrypt a message using one copy of a one-time pad, and then delete the other copy of the one-time pad, this is equivalent to deleting our original message and producing a random ciphertext. So the ciphertext carries no information about the plaintext in the absence of the private key.

We can use our formalism to derive from this security property a strong constraint on the encryption operation E.

Theorem 4.2 *If the encryption step in classical encrypted communication satisfies property* (24)*, then encryption is not invertible unless the space of messages is trivial.*

Proof. Suppose encryption is invertible. Then composing both sides of (24) with E^{-1} gives the following graphical expression:



(24)

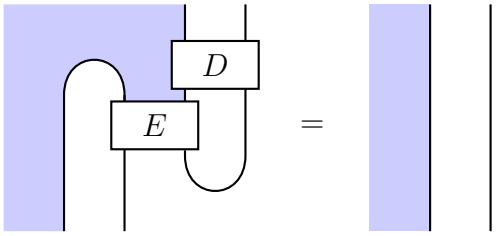
Hence the identity process on the set of messages factors through the one-element set. □

This is a desirable property: if encryption were invertible, then both the plaintext and the secret key would be derivable in principle from the ciphertext.

We can draw a very different conclusion for the decryption process D.

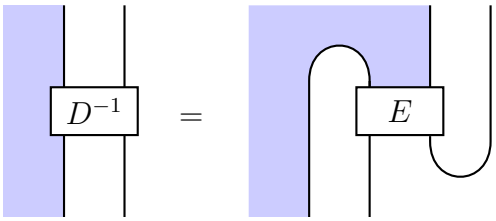
Theorem 4.3 *In classical encrypted communication, the decryption step is invertible.*

Proof. From equation (23) representing correctness of encrypted communication, we apply the topological properties of public information to obtain the following equivalent equation:



(25)

This says that D has a right inverse given by E with its top-left and bottom-right legs twisted in the manner indicated. However, by Theorem 2.1, if an endomorphism is a left inverse then it must also be a right inverse, and hence our theorem follows, with the following expression for D^{-1}:

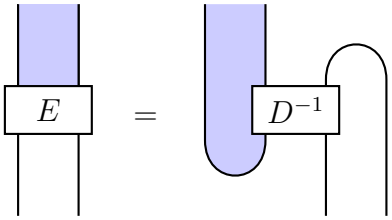


(26)

□

It follows that we can reconstruct E from the knowledge of D and its inverse.

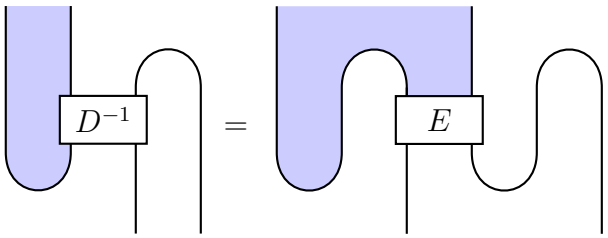
Theorem 4.4 *For an implementation of classical encrypted communication, we have*



(27)

Proof. We apply the topological properties of public information to expression (27)

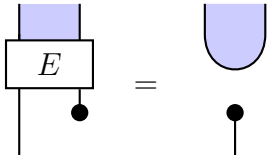
to obtain the following:



(28)

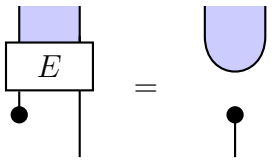
The right-hand side of this expression evaluates to E, by the topological properties (14) of 2-dimensional regions and the snake equations (8). □

While property (24) is primary, there are other security properties of the encryption process that we could consider. The first states that if we encode with a random key, this is equivalent to deleting the original message and producing random ciphertext:



(29)

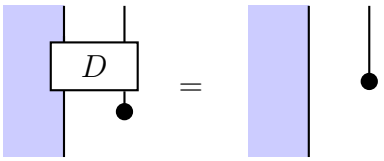
Secondly, we could encode a random message with a specified key:



(30)

This property says that this is the same as deleting the key, and producing a random ciphertext.

We can also consider security properties for the decryption process.



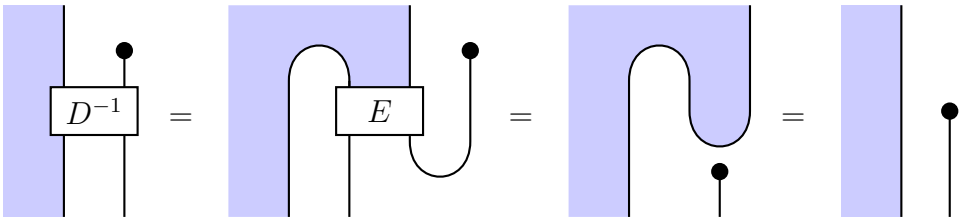
(31)

This says that if an attacker chooses nondeterministically from the space of all possible keys, every possible message can be produced, regardless of the ciphertext. So if an attacker has no knowledge of the key, they cannot extract information from the ciphertext.

In fact, we can use our formalism to show that all of these security properties follow from the primary security property (24).

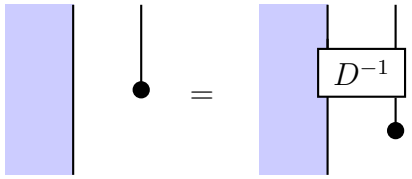
Theorem 4.5 *In classical encrypted communication,* (24) *implies* (30), (31) *and* (32).

Proof. The implication (24) ⇒ (30) follows from the topological property (10) of the deletion map. For the other implications, we compose expression (27) for D^{-1} with the deletion map at the top-right leg, obtaining the following:



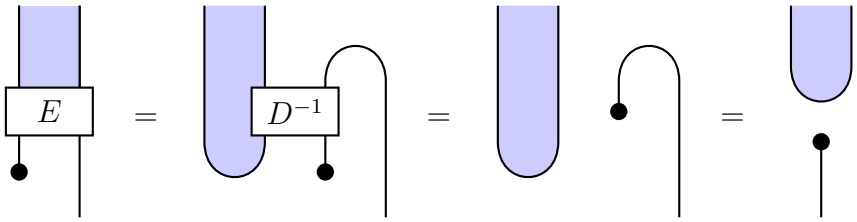
(32)

Every invertible 2-cell in **Rel** is a family of bijections, and hence its converse is its inverse. Taking the converse is a functorial operation, and so taking the converse of of the first and last diagram here, we obtain property (32). For the final property (31), we postcompose this expression with the 2-cell D^{-1}, obtaining the following expression:



(33)

We can use this to prove security property (31), where we also make use of expression (28) giving E in terms of D^{-1}:

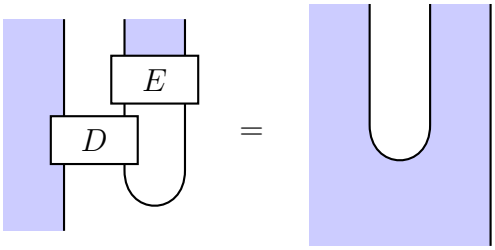


(34)

This completes the proof. □

4.3 Secret sharing

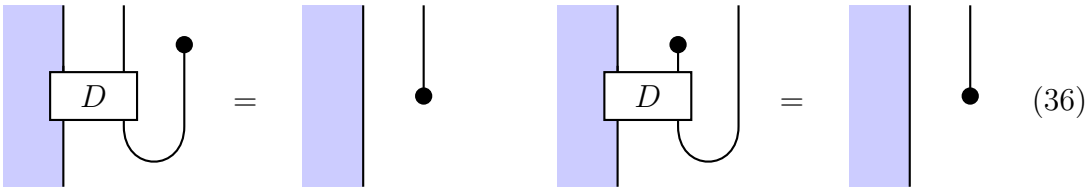
We can represent correctness of a secret sharing procedure in the following way:



(35)

On the left, we begin with some pre-existing public information. This is the information to be communicated by the procedure. We prepare two correlated systems using a one-time pad, and then manipulate the first copy by a procedure D that depends on the value of the classical data. The result is a pair of ciphertexts. Both are then consumed together by a process E, producing public information. This is successful when the result is to copy the original public information.

The important security property of a secret sharing procedure is that if only one ciphertext is available, then no information about the original message can be regained. A strong, constructive way to phrase this is to say that if one of the ciphertexts is erased, the other becomes uniformly random, and independent of the original message. This gives two conditions, with the following graphical representations:



(36)

Equation (36) has an identical structure to the quantum dense coding equation given in [13], and the security properties (37) are equivalent to properties (32) and (34) of the encrypted communication protocol.