Information Leakage in Resource Theories

Niels Voorneveld
Cybernetica AS
Tartu, Estonia

Alessandro Di Giorgio
Tallinn University of Technology
Tallinn, Estonia

Paweł Sobociński
Tallinn University of Technology
Tallinn, Estonia

When considering information leaks, one is not necessarily concerned with the exact data which was leaked, but more so with the content and context of the data. For instance, leaking the same piece of information twice should not be different from leaking it once, since both situations leak the same information. We introduce leakage categories, a categorical framework for reasoning about information leakage. Starting from a symmetric monoidal category $\mathbb{C}$, we construct the leakage category $\mathsf{L}(\mathbb{C})$ as a quotient of the copara construction $\mathsf{coPara}(\mathbb{C})$. Moreover, we give an equivalent characterisation of $\mathsf{L}(\mathbb{C})$ as a poset-enriched symmetric monoidal category with extra structure. This allows to derive properties of information leaking, e.g. idempotency and propagation through deterministic morphisms in the setting of copy-discard categories. We use the framework to analyse data leaking in cryptographic protocols.

1 Introduction

We examine the concept of information leaks from a categorical viewpoint. Monoidal categories can be used to describe resource theories [8], and information can be considered as a resource: what, then, is an information leak? In considering this, we are not concerned about the exact data which was leaked, but more about its content, as might be considered in cryptography [17]. Leaking something already publicly known or derivable from public information should not be considered a true leak.

Understanding what is *truly* leaked can be complicated. Approaches such as *differential privacy* [12] determine whether small leaks together leak sensitive information. The general framework of privacy properties is as follows: we describe some process with a particular utility, such as sending a message, and then designate what an external adversary learns. We then reason whether this adversary can learn anything that was supposed to be kept private. Considering the nuance of these concepts, we aim to explore them from first principles.

Given any symmetric monoidal category, we may consider a *leakage morphism* to be a morphism with an additional output signifying what was leaked. We then say that one leakage morphism *leaks more* than another, if one can simulate the other using a morphism on the leaked data. This corresponds to morphisms in the coPara construction [5, 4], with simulations given by the 2-cells. We quotient this category over the equivalence relation induced by this "leaks more" relation, obtaining what we call a *leakage category*. The principle of simulation we use here is akin to the notion of adversary simulation used in *Universal Composability* [3] used in cryptography.

Moreover, we study an algebraic characterisation of leakage categories as a free construction. When this construction is applied to a copy-discard category, we can derive the following conceptual properties:

- Leaking a piece of immutable data once is the same as leaking it many times.
- If all inputs of a deterministic function are leaked, then the output of the function is leaked.
- Leaking data from isolated processes, without inputs or outputs or any auxiliary observable effects, has no consequence.

Submitted to:
ACT 2026

Note that whenever we evaluate the informational content of a leak, we can always use known details of the process. Knowing that some part of the process is predictable, for instance deterministic, allows us to derive information of the output based on information regarding the input. In other words, we see the process itself as leaked by default; and the adversary knows which process was executed but not necessarily the input, output and auxiliary details of a particular execution of the process.

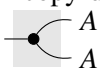
Concretely, we apply the construction to `FinStoch`, the category of finite sets and stochastic maps, to explore examples of encryption-decryption and secure multiparty computation. This illustrates how we can reason about: 1) leaks propagating through a process in order to derive further leaks, and 2) leaks being isolated in order to render them inconsequential. Both these ideas together allow us to prove whether something was truly leaked or not. Indeed, despite its simplicity, the framework of leakage categories allows us to state precisely whether a cryptographic protocol is correct in the presence of plain-text (leaked) information. Our headline examples include leakages of message encryption and the Dining Cryptographers Problem [6].

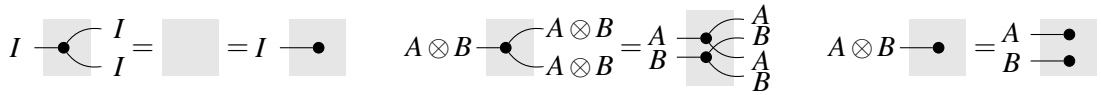
Synopsis: After establishing the necessary preliminaries in §2 we give the main definitions in §3 and establish properties of leaking in §4. In §5 we illustrate the framework on several examples. Appendix A contains the omitted proofs and Appendix B contains further examples and connections to related fields.

2 Resource Theories and Symmetric Monoidal Categories

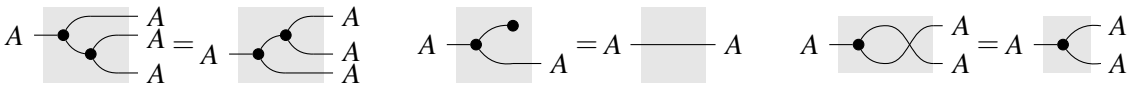
We think of (strict) symmetric monoidal categories (smcs) as resource theories, and consider their associated graphical calculus [16, 19, 18], following [8]. Objects $A, B, C, \ldots$ are types of resources, and morphisms $f\colon A \to B$ are processes that consume a resource of type A and produce one of type B. Composition $f;g$ applies processes in sequence, with the identity being the process that does nothing. The monoidal product $\otimes$ on objects combines resources, and on morphisms it is the parallel composition of processes. The monoidal unit I is the void resource. The symmetry χ swaps the order of resources.

In resource theories, it is typically the case that resources can be shared and discarded by processes. Categorically, this amounts to requiring additional structure on the objects.

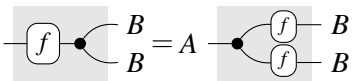
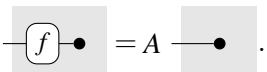
Definition 1. *A* copy-discard *category [9, 7] is a smc where each object A has a* copy $\delta_A : A \to A \otimes A$, *depicted as*  *, and a* discard $\varepsilon_A : A \to I$, *depicted as* *, which are compatible with* $\otimes$*:*



and form a cocommutative comonoid:



Definition 2. *In a copy-discard category, a morphism* $f : A \to B$ *is*

deterministic *if*  *and* total *if*  .

Definition 3. *A* Markov category *[13] is a copy-discard category in which all morphisms are total.*

`FinStoch` is our main example. It has finite sets as objects. Morphisms $f : A \multimap B$ are functions from A to distributions over B. We write a distribution over A as a formal sum $p_1|a_1\rangle + \cdots + p_n|a_n\rangle$ where each $a_i \in A$ and $p_i \in [0,1]$ such that $\sum_{1\leq i\leq n} p_i = 1$. The identity, copy and discard morphisms send a to $1|a\rangle$, $1|(a,a)\rangle$, and $1|()\rangle$ respectively, and the swap sends (a,b) to $1|(b,a)\rangle$.

3 Leakage Categories

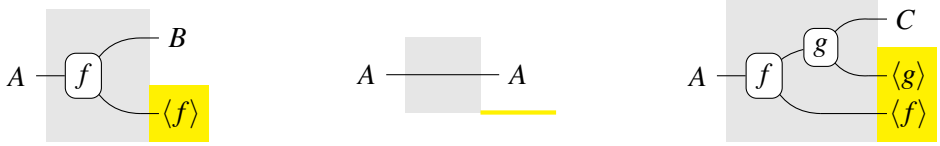
In this section we introduce *leakage categories*. These arise from a construction on a symmetric monoidal category $\mathbb{C}$ that designates part of the output of a process to explicitly tracked leaked data. Moreover, they come equipped with an order that compares processes on the amount of *information* they leak.

Our construction builds on a 1-categorical adaptation of the coPara construction [5, 4], where we refer to the *coparameters* as *leaks*.

Definition 4. *Let $\mathbb{C}$ be a symmetric monoidal category.* $\mathsf{coPara}(\mathbb{C})$ *is the category whose*

- *objects are those of $\mathbb{C}$;*
- *morphisms $f : A \nrightarrow B$ are given by an object $\langle f\rangle$ in $\mathbb{C}$ and a morphism $\widehat{f} : A \to B \otimes \langle f\rangle$ in $\mathbb{C}$;*
- *identities $i_A : A \nrightarrow A$ are given by $\langle i_A\rangle := I$ and $\widehat{i_A} := id_A : A \to A = A \otimes 1$;*
- *composition of $f : A \nrightarrow B$ and $g : B \nrightarrow C$ is given by $f \rhd g : A \nrightarrow C$, where $\langle f \rhd g\rangle := \langle g\rangle \otimes \langle f\rangle$ and $\widehat{f \rhd g} := f;(g \otimes id_{\langle f\rangle})$.*

In string diagrams, we designate a special part of the output to represent the type of leaks. For instance, a morphism $f : A \nrightarrow B$, the identity i_A, and the composition $f \rhd g$ are rendered as follows:



Morphisms of $\mathsf{coPara}(\mathbb{C})$ make the leakage of *data* explicit. However, none of the desired properties regarding information content of what is leaked is incorporated. In order to do so, we define an equivalence relation using the 2-cells from the coPara construction, stating when the same *information* has been leaked.

Definition 5. *Given morphisms $f, g : A \nrightarrow B$ of* $\mathsf{coPara}(\mathbb{C})$ *of the same type, we say f* leaks more than *g, denoted $f \sqsupseteq g$, if there is a morphism $a : \langle f\rangle \to \langle g\rangle$ in $\mathbb{C}$ such that $f;(id_B \otimes a) = g$. We say f and g* leak equally as much*, denoted as $f \equiv g$, if $f \sqsupseteq g$ and $g \sqsupseteq f$.*

Lemma 1. *$\sqsupseteq$ forms a preorder, and $\equiv$ forms an equivalence relation on morphisms.*

Proof. For any $f : A \nrightarrow B$, $f;(id_B \otimes id_{\langle f\rangle}) = f$ and thus $f \sqsupseteq f$. For any $f, g, h : A \nrightarrow B$ such that $f \sqsupseteq g$ and $g \sqsupseteq h$ via $a : \langle f\rangle \to \langle g\rangle$ and $b : \langle g\rangle \to \langle h\rangle$, we have that $f;(id_B \otimes (a;b)) = f;(id_B \otimes a);(id_B \otimes b) = g;(id_B \otimes b) = h$, and thus $f \sqsupseteq h$. By definition of $\equiv$, for any $f, g : A \nrightarrow B$, $f \equiv g$ if and only if $g \equiv f$. □

Definition 6. *Given a symmetric monoidal category $\mathbb{C}$, the* leakage category $\mathsf{L}(\mathbb{C})$ *is given by* $\mathsf{coPara}(\mathbb{C})$ *quotiented by the equivalence relation $\equiv$.*

Because $\mathsf{coPara}(\mathbb{C})$ is a category, so is $\mathsf{L}(\mathbb{C})$. Note that $\mathsf{coPara}(\mathbb{C})$ is typically *not* a monoidal category because $\otimes$ fails to be a bifunctor – the two sides of the middle four interchange produce a different ordering of leaks. However, if there is an underlying symmetric monoidal structure, then this ordering is immaterial because of $\equiv$, then indeed:

Lemma 2. *If $\mathbb{C}$ is a symmetric monoidal category, then $\mathsf{L}(\mathbb{C})$ is a symmetric monoidal category.*

Here we define $f \otimes g \colon A \otimes C \nrightarrow B \otimes D$ as $\langle f \otimes g \rangle := \langle f \rangle \otimes \langle g \rangle$ and $\widehat{f \otimes g} := (\widehat{f} \otimes \widehat{g}); (id_B \otimes \chi_{\langle f \rangle, D} \otimes id_{\langle g \rangle})$.

Lemma 3. *If $\mathbb{C}$ is a symmetric monoidal category, then $\mathsf{L}(\mathbb{C})$ is a poset-enriched symmetric monoidal category.*

Proof. Given $f_1, f_2 \colon A \nrightarrow B$ and $g_1, g_2 \colon B \nrightarrow C$ such that $f_1 \sqsupseteq f_2$ and $g_1 \sqsupseteq g_2$ via $a \colon \langle f_1 \rangle \to \langle f_2 \rangle$ and $b \colon \langle g_1 \rangle \to \langle g_2 \rangle$, we have that $f_1; g_1 \sqsupseteq f_2; g_2$ via $b \otimes a$. Similarly, given $h_1, h_2 \colon C \nrightarrow D$ such that $h_1 \sqsupseteq h_2$ via $c \colon \langle h_1 \rangle \to \langle h_2 \rangle$, we have that $f_1 \otimes h_1 \sqsupseteq f_2 \otimes h_2$ via $a \otimes c$. □

Proposition 1. *There is a functor $F \colon \mathbb{C} \to \mathsf{L}(\mathbb{C})$ giving the* non-leaking *morphisms in $\mathsf{L}(\mathbb{C})$.*

Proof. F is defined as the identity on objects and, for any $f \colon A \to B$ in $\mathbb{C}$, $F(f) \colon A \to B$ is given by $\langle F(f) \rangle := I$ and $\widehat{F(f)} := f \colon A \to B = B \otimes I$. □

3.1 Free Leakage Categories

In order to reason *algebraically* about information leakage, we now introduce a *free* construction $\mathsf{FL}(\mathbb{C})$ that equips a symmetric monoidal category $\mathbb{C}$ with leakage as additional structure. We then show that this construction coincides with $\mathsf{L}(\mathbb{C})$, in the sense that the two are isomorphic.

Definition 7. *Given an smc $\mathbb{C}$, the* free leakage category $\mathsf{FL}(\mathbb{C})$ *is constructed as follows:*

- *objects are those of $\mathbb{C}$;*
- *morphisms are inductively generated by:*

$$\frac{f : A \to_{\mathbb{C}} B}{\{f\} : A \nrightarrow B} \qquad \frac{f : A \nrightarrow B \quad g : B \nrightarrow C}{f;g : A \nrightarrow C} \qquad \frac{f : A \nrightarrow B \quad g : C \nrightarrow D}{f \otimes g : A \otimes C \nrightarrow B \otimes D} \qquad \frac{A \in \mathbb{C}_0}{\dashv_A \colon A \nrightarrow I}$$

  subject to

$$\overline{\{f\};\{g\} = \{f;g\}} \qquad \overline{\{f\} \otimes \{g\} = \{f \otimes g\}} \qquad \overline{\dashv_A \otimes \dashv_B = \dashv_{A \otimes B}} \qquad \overline{\dashv_I = \{id_I\}}$$

  and closed under the axioms of symmetric monoidal categories;

- *homsets are equipped with an inductively generated partial order:*

$$\frac{}{f \sqsupseteq f} \qquad \frac{f \sqsupseteq g \quad g \sqsupseteq h}{f \sqsupseteq h} \qquad \frac{f \sqsupseteq g \quad g \sqsupseteq f}{f = g} \qquad \frac{f_1 \sqsupseteq f_2 \quad g_1 \sqsupseteq g_2}{f_1; g_1 \sqsupseteq f_2; g_2} \qquad \frac{f_1 \sqsupseteq f_2 \quad g_1 \sqsupseteq g_2}{f_1 \otimes g_1 \sqsupseteq f_2 \otimes g_2} \qquad \frac{f : A \nrightarrow B}{\dashv_A \sqsupseteq f; \dashv_B}$$

$\mathsf{FL}(\mathbb{C})$ adds to $\mathbb{C}$ a morphism $\dashv_A \colon A \nrightarrow I$ for every object A that represents a leakage of data of type A. This structure is coherent w.r.t. $\otimes$ and is the maximally leaking process of type $A \nrightarrow I$. Diagrammatically:

We can establish that the leakage and free leakage constructions are isomorphic.

Theorem 1. *There is an isomorphism of poset-enriched symmetric monoidal categories* $\mathsf{L}(\mathbb{C}) \cong \mathsf{FL}(\mathbb{C})$.

The two definitions for leakage categories are equivalent, and each is useful in their own right. The free leakage construction is more flexible for proving *equalities* between morphisms as series of rewrites. The direct construction, on the other hand, helps with proving *inequalities*, e.g. non-triviality statements in Subsection 4.2, since we know that any equality must be facilitated by two concrete morphisms equating the two. Given the isomorphism, throughout the rest of the paper we will use the two constructions interchangeably. We shall refer to $f \sqsubseteq g$ as an *inclusion* of f in g.

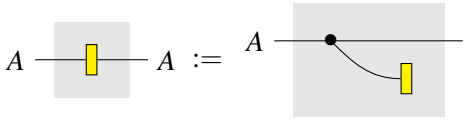
4 Information versus data

Recall that the additional structure of copy-discard categories allows to model resource sharing. Leakage categories preserve this structure, allowing us to investigate further algebraic properties of leakage.

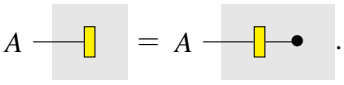
Lemma 4. *If* $\mathbb{C}$ *is a copy-discard category, then* $\mathsf{FL}(\mathbb{C})$ *is a copy-discard category.*

Proof. The copy-discard structure of $\mathsf{FL}(\mathbb{C})$ is inherited from $\mathbb{C}$, i.e. $\varepsilon_A := \{\varepsilon_A\}$ and $\delta_A := \{\delta_A\}$. □

Definition 8. *The* leakage identity $l_A : A \multimap A$ *in* $\mathsf{FL}(\mathbb{C})$ *is defined as* $\delta_A;(id_A \otimes \dashv_A)$*:*



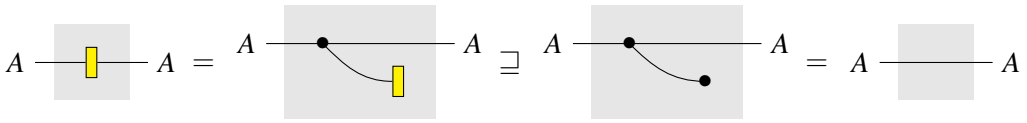
Note that, we could have also defined $\dashv_A$ in terms of the leakage identity:



With the leakage identity, we can investigate intuitive algebraic properties of leaking. The first of which we have already encountered in our proof that leakage categories are monoidal: the order of leaks does not matter. We go through some further properties:

Lemma 5. *A leakage identity leaks more than an identity:* $l_A \sqsupseteq id_A$.

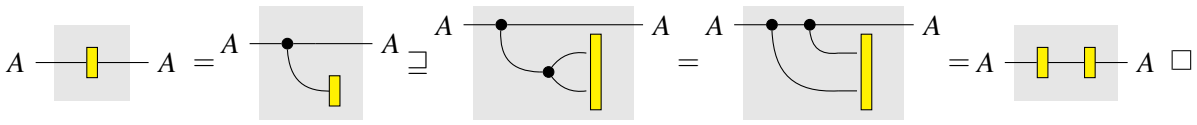
Proof.



□

Lemma 6. *Leaking something twice is no different from leaking it once:* $l_A;l_A = l_A$.

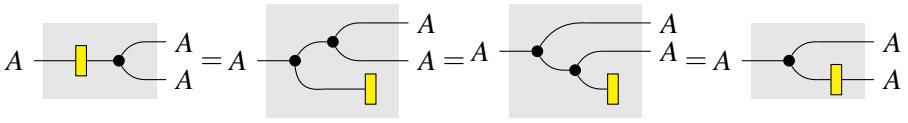
Proof. By the previous lemma, $l_A;l_A \sqsupseteq l_A$. For the other inclusion the following holds:



□

Lemma 7. *A copy of leaked data is also leaked:* $l_A;\delta_A = \delta_A;(id_A \otimes l_A) = \delta_A;(l_A \otimes id_A)$.

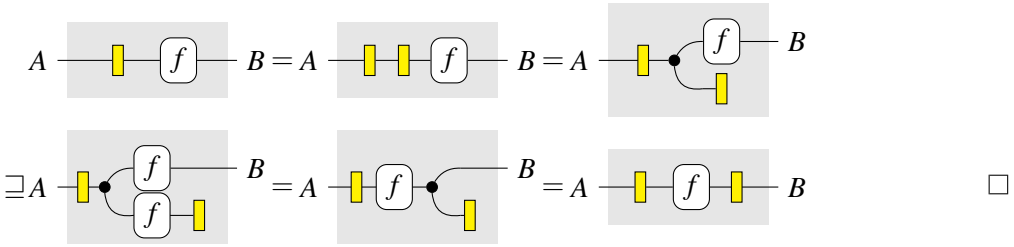
Proof. We prove one equation; the other equation is proved analogously.



□

Lemma 8. *If all inputs of a* deterministic *morphism $f\colon A \to B$ are leaked, then all outputs are leaked as well: $l_A;\{f\} \sqsupseteq l_A;\{f\};l_B$.*

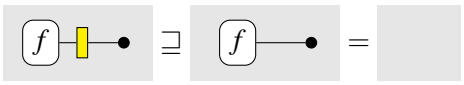
Proof.



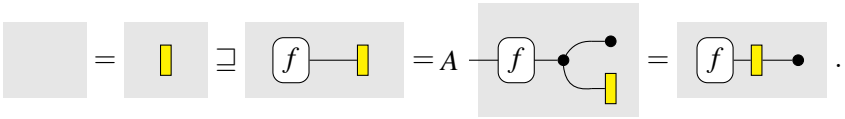
□

Lemma 9. *Discarding the leaked outputs of a* total *morphism $f\colon I \to A$ is equivalent to doing nothing: $\{f\};l_A;\{\varepsilon_A\} = id_I$.*

Proof. One inclusion is proved using Lemma 5 and the fact that f is total:



For the other inclusion, the following holds:

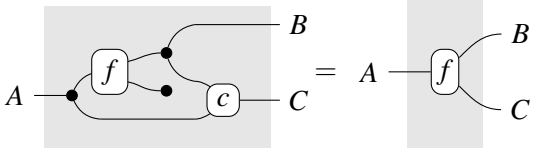


□

4.1 Conditionals

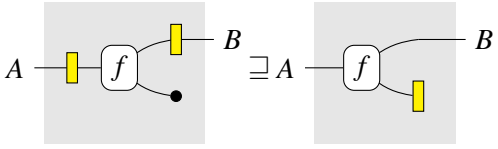
We investigate further properties of leaking when the underlying copy-discard category has *conditionals*: a factorisation property often found in categorical probability [13].

Definition 9. *A copy-discard category* $\mathbb{C}$ has conditionals *if for every morphism $f : A \to B \otimes C$, there is a morphism $c : B \otimes A \to C$ such that:*

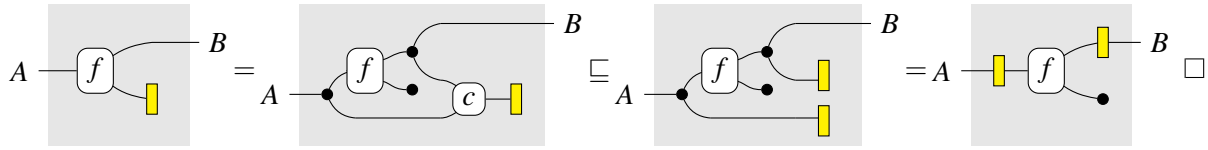


If $\mathbb{C}$ is a copy-discard category with conditionals, then in $\mathsf{FL}(\mathbb{C})$ we have the following properties:

Lemma 10. *Any leak can be retrieved from leaking the input and the output:*



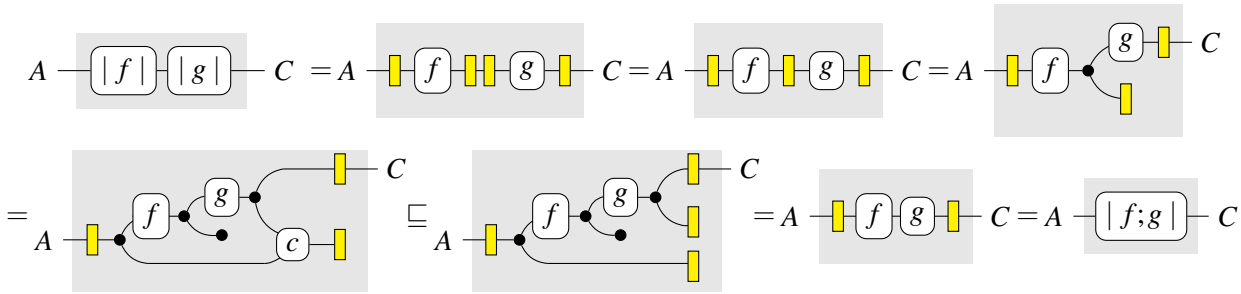
Proof.



□

Lemma 11. *Leaking input and output forms a semi-functor* $|\cdot|\colon \mathsf{FL}(\mathbb{C}) \to \mathsf{FL}(\mathbb{C})$.

Proof. $|\cdot|$ is the identity on objects, and for any morphism $f\colon A \nrightarrow B$, $|f| := l_A; f; l_B$. To show that $|\cdot|$ preserves composition, observe that one inclusion holds by means of Lemma 6, while the other inclusion is proved with conditionals:



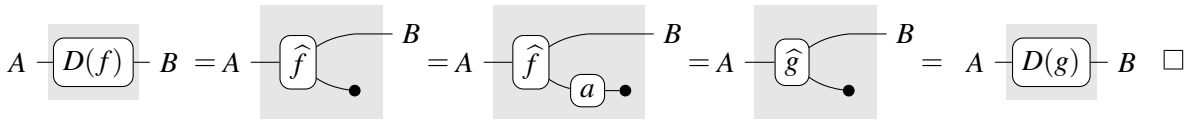
□

In a copy-discard category, we say that two leakage morphisms have the same *underlying* process, if replacing any $\dashv_A$ with ε_A gives the same morphism.

Lemma 12. *If* $\mathbb{C}$ *is a Markov category, then there is a functor* $D\colon \mathsf{L}(\mathbb{C}) \to \mathbb{C}$ *given by replacing leaks with discard operations.*

Proof. D is the identity on objects, and for any $f\colon A \nrightarrow B$, $D(f) := \widehat{f}; (id_B \otimes \varepsilon_{\langle f \rangle})$. By the isomorphism in Theorem 1, D can be seen as the functor that replaces any occurrence of $\dashv_X$ in f with ε_X.

The functor is well-defined: suppose $f \sqsupseteq g$ in $\mathsf{L}(\mathbb{C})$, then we have a morphism $a\colon \langle f \rangle \to \langle g \rangle$ such that $\widehat{f}; (id_B \otimes a) = \widehat{g}$ and hence:



□

In a leakage category over a Markov category with conditionals, every morphism f lies between its leak-proof and leak-full version in the order of leaks. That is, $F(D(f)) \sqsubseteq f \sqsubseteq |D(f)|$.

4.2 Non-triviality

Since the leakage category definition adds equations, there is the risk that the category $\mathsf{L}(\mathbb{C})$ is *trivial*. That is, any two morphisms f, g from the same homset may be equivalent. We can however show that in a few useful examples, including Markov categories, this is not the case.

Firstly, we can show that the functor $F : \mathbb{C} \to \mathsf{L}(\mathbb{C})$ is faithful given certain conditions. That is, ensuring that $F(f) \equiv F(g)$ implies $f = g$, and therefore, if $\mathbb{C}$ is non-trivial then so is $\mathsf{L}(\mathbb{C})$.

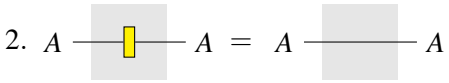
Lemma 13. *If $\mathbb{C}$ is a poset-enriched symmetric monoidal category with a discard ε_A, for every object A, such that, for every morphism $f : A \to B$, $f;\varepsilon_B \leq \varepsilon_A$, then $F : \mathbb{C} \to \mathsf{L}(\mathbb{C})$ is faithful.*

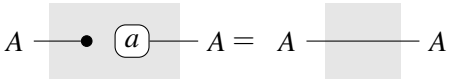
Examples which meet this requirement are the category of sets and relations `Rel`, `FinStoch` and any other Markov or partial Markov category [11].

Secondly, the following result shows that leaks are significant in copy-discard categories; if leaking a certain resource is the same as doing nothing, then the associated resource type has no substance.

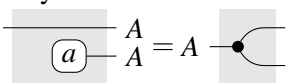
Lemma 14. *If $\mathbb{C}$ is a copy-discard category, then the following are equivalent in $\mathsf{FL}(\mathbb{C})$:*

1. 

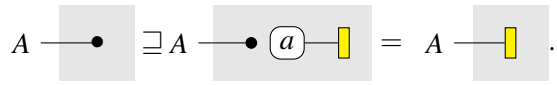
2. 

3. *there is an $a : I \to A$ in $\mathbb{C}$ such that* 

Proof. From the definition of the leakage identity, it is easy to see that 1. if and only if 2. holds.

Assume 2., then there is an $a : I \to A$ such that 

, and thus 3. holds by postcomposing with $\varepsilon_A \otimes id_A$.

To conclude, assume 3., and observe that 

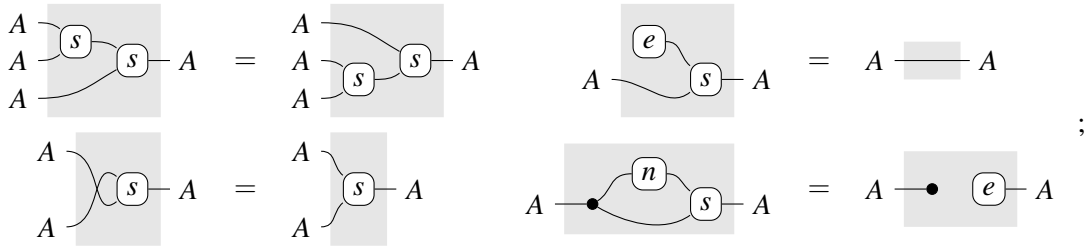
. □

5 Encryption examples

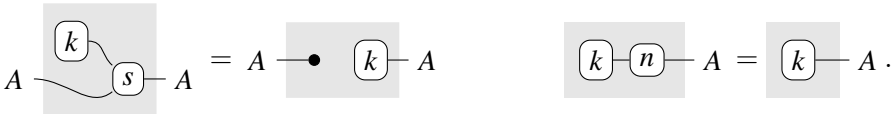
We consider as primary example *symmetric encryption* in a Markov category. Within this category, we consider the following structure:

Definition 10. *A* generating group algebra *(or gg-algebra) in a copy-discard category is a tuple (A, e, s, n, k) where A is some object and $e, k : I \to A$, $s : A \otimes A \to A$ and $n : A \to A$ are morphisms such that:*

- *e, s, k and n are total;*
- *e, s, and n are deterministic;*
- *(e, s, n) forms a commutative group, with identity e, group operation s and inverse n:*



- *k is absorbing for s and it is inverse to itself:*

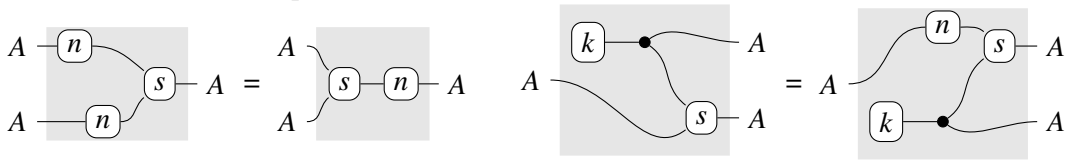


Here we see A as the space for representing keys, messages and cipher texts, with messages often encoded by some injection. e represents the neutral element of the space, s as the symmetric encryption, and $(n \otimes id_A); s$ the decryption operation, and k as the primary way to generate keys randomly.

Lemma 15. *If k is deterministic then A is isomorphic to I.*

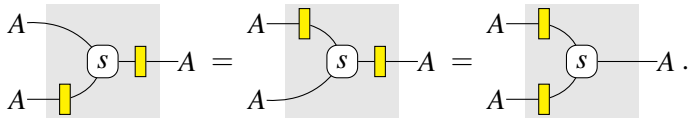
An example of such a gg-algebra in `FinStoch` can be constructed by taking a natural number $m > 1$ and let $A = \mathbb{Z}_m = \{0, 1, 2, \ldots, m-1\}$, with $e() := |0\rangle$, $s(a,b) := |a+b \mod m\rangle$, $n(a) = |(-a) \mod m\rangle$ and $k() := \sum_{a \in \mathbb{Z}_m} \frac{1}{m} |a\rangle$. Commonly, m is taken to be a prime for additional multiplicative properties we do not concern ourselves with in this example. By taking $m = 2$, we retrieve the example of Booleans, false constant, xor-gate and coin-toss (inverse is identity function). Lastly, given two such algebras, their monoidal product gives such an algebra as well, allowing for encryption over multiple bits.

Lemma 16. *The following equations hold in a gg-algebra:*



Suppose $\mathbb{C}$ is a Markov category with a gg-algebra (A, e, s, n, k). We consider how leaks in $\mathsf{FL}(\mathbb{C})$ are propagated across the encryption operation s.

Lemma 17.

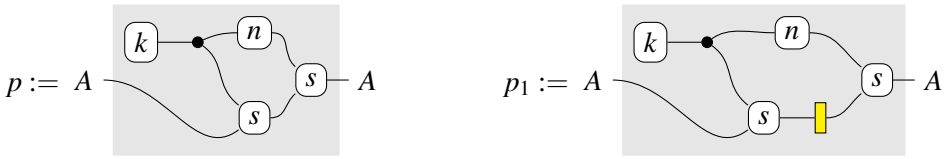


Leaking a randomly generated key, if not used anywhere else, is of no consequence. Moreover, any deterministic unit can be considered leaked by default.

Lemma 18.

5.1 One time pad with leaks

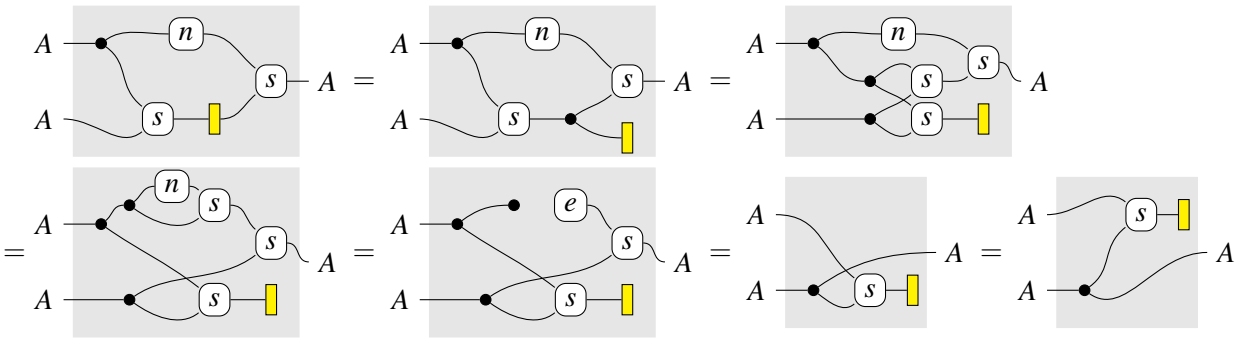
We consider the one-time pad protocol: a private key is randomly generated using k and shared with someone else; then a message is encrypted with the key using s and sent to the same person; this person then decrypts the message using the same key, retrieving the original message. Following [2], we can describe this process in a copy-discard category $\mathbb{C}$ with a gg-algebra, as the diagram below on the left.



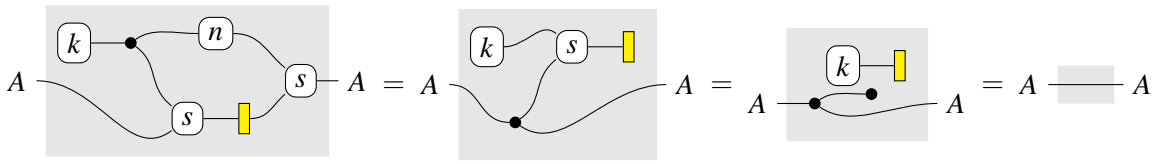
Here, the message is considered as an input, and the process describes the generation of the key, and both the encryption and decryption of the message. It can be easily proved that $p = id_A$, that is: the protocol is correct.

Now suppose the encrypted message was intercepted, which we consider as a leak. This is the morphism p_1 in $\mathsf{FL}(\mathbb{C})$, given above on the right. Note that $D(p_1) = p$.

Disregarding the key generation for a moment, we can prove that despite the leak, the recipient will get the correct message. That is, we can prove that the process can be rewritten as simply independently sending the unencrypted message to the recipient whilst leaking the encryption.



Using this, we can prove that the process $p_1 = id_A$:

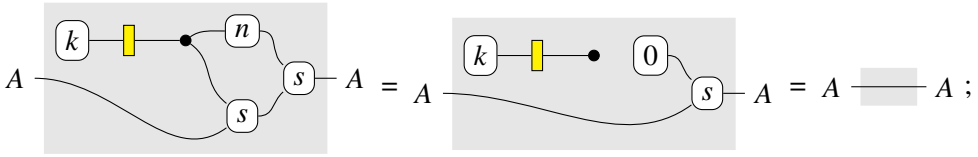


So we can say that the message was communicated correctly, and there was no leak. By this we mean that even though there was a physical *data* leak, that leak had no *informational* content.

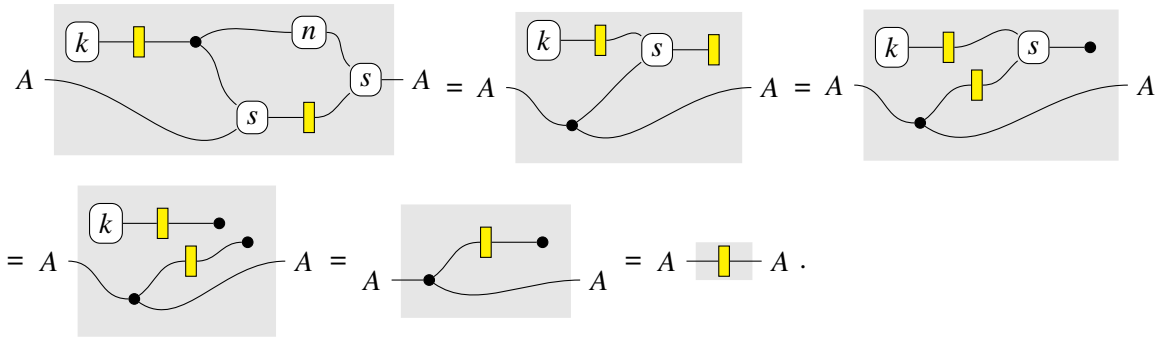
Moreover, even though the process could be used in some wider context, in which the message, or part of it may be leaked, this does not change the fact that the particular leak in this process itself *is of no consequence*. It cannot help an adversary in any way, even in a wider context.

Lastly, we consider two further scenarios of this process:

- The key is leaked, e.g. it was not shared securely, but the encrypted message is not leaked, then nothing got leaked.



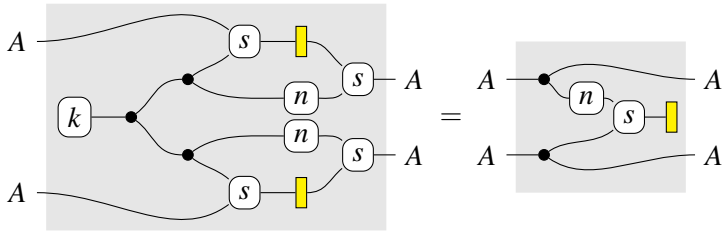
- If both the key and the encrypted message are leaked, then the message is leaked as well.



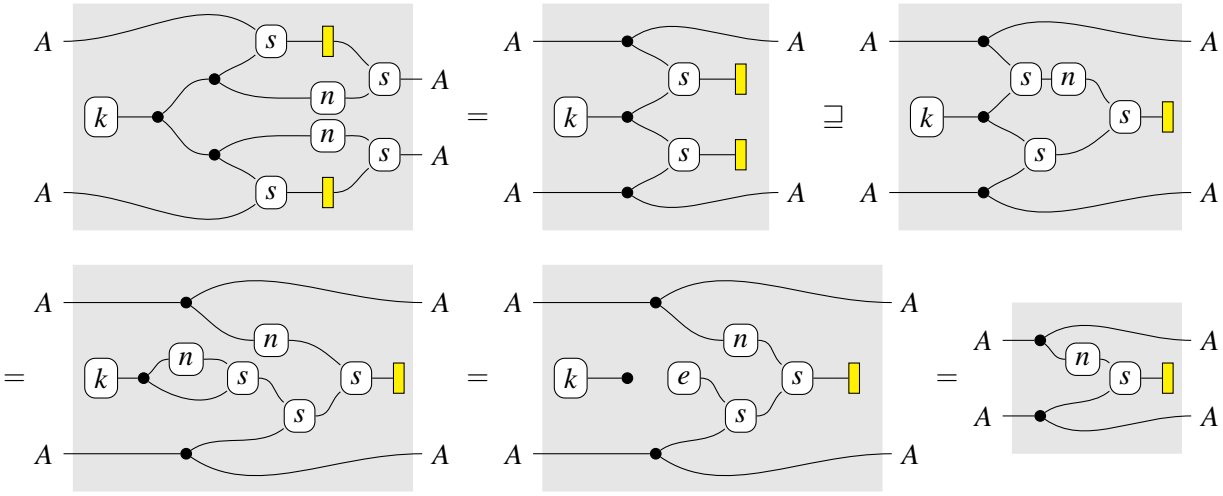
5.2 Reusing the one-time pad key

Suppose now that the key is reused in two encryption-decryption procedures, and both the encrypted messages are leaked. This should not be done in practise, as the name of the protocol suggests.

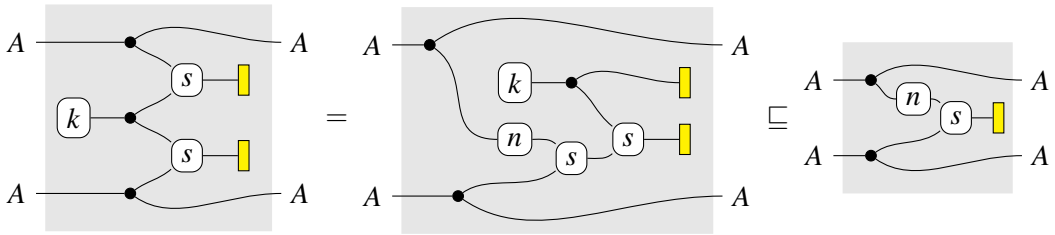
We can show that some combination of the unencrypted messages is also leaked. Moreover, we can show that in terms of information, this is the *exact* thing that got leaked, and establish an equivalence:



We prove the two inclusions separately. One inclusion is proved as follows:



and for the other inclusion, the following holds:



As a consequence, if the morphism is composed with another which leaks part of one of the outputs, this translates to potential information leak regarding the other output. For example, if furthermore one of the messages got fully leaked, the other gets fully leaked as well.

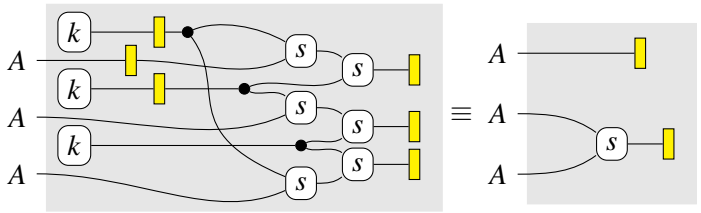
5.3 The dining cryptographers problem

As final example, let us consider the following classical cryptographic protocol [6]. Suppose there are three people, at most one of which satisfies some property. All three people want to figure out whether

someone actually has this property, or whether none of them have, without learning who specifically has it. The classic problem illustrates this scenario as three cryptographers sharing a meal in a restaurant, figuring out whether it was one of them who paid the bill anonymously.

The solution to the problem supposes that all three people toss a coin in private and share the result with the person on their right, but not with the person on their left. All three will then publicly share whether the two coins they can see match or not, speaking honestly if they have the property and lying if they do not. It can then be derived that each person learns, on top of knowing whether they themselves have the property, what the parity of the number of people satisfying the property is. In case the number of people satisfying the property is one or zero, as proposed in the problem statement, then they learn whether or not someone satisfies the property.

We can model this example using `FinStoch`'s gg-algebra $(A, e, s, n, k) := (\mathbb{B}, \texttt{false}, \texttt{xor}, id_{\mathbb{B}}, \frac{1}{2})$, where $\frac{1}{2}$ is a coin-toss. The scenario is modelled by marking all information known to one participant in the protocol using leaks:



The participant will know whether they hold the property and what the results of the coin tosses they can see are. Moreover, they would know what everyone publicly proclaimed. Using the reasoning principles established in this paper, we can then show that this is the same as the participant knowing whether they hold the property, and whether exactly one of the others hold the property.

6 Conclusions

We have seen how a relatively simple definition allows us to investigate the notion of information leaked in a resource theory, abstracting away from data to informational content of leaks. This relates to *Blackwell informativeness theorem* [1], where different equivalent formulations of an order compares the informational content of *signals*.

Another connection exists to non-interference [14], where the notion of sensitive data influencing insecure data is studied. We can consider a *leakage category variation* of non-interference, expressing whether leaks of insecure data propagate to leaks of sensitive data.

One interesting avenue of future research is to consider the dual of leakage, *corruption*. Here we can think of an *active adversary* as influencing a protocol, as might happen in a man-in-the-middle attack. This could be formulated by using the Para construction instead. Like with leakages, we can think of corruption as spreading through a process, resulting in a similar algebraic theory.

We plan to develop the theory further in the direction of actual privacy results in security protocols, taking into consideration both distributed interaction protocols and adversarial capabilities. One principal consideration is to limit to a category with only polynomial time algorithms. To this end, we would look at the combination of the leakage construction and polynomial iteration formulated in [10].

References

[1] David Blackwell (1953): *Equivalent Comparisons of Experiments*. The Annals of Mathematical Statistics 24(2), pp. 265–272, doi:10.1214/aoms/1177729032.

[2] Anne Broadbent & Martti Karvonen (2024): *Categorical composable cryptography: extended version*. Logical Methods in Computer Science 19.

[3] Ran Canetti (2001): *Universally composable security: A new paradigm for cryptographic protocols*. In: Proceedings 42nd IEEE Symposium on Foundations of Computer Science, IEEE, pp. 136–145.

[4] Matteo Capucci & Bruno Gavranović (2022): *Actegories for the Working Amthematician*. ArXiv:2203.16351.

[5] Matteo Capucci, Bruno Gavranović, Jules Hedges & Eigil Fjeldgren Rischel (2022): *Towards Foundations of Categorical Cybernetics*. In: Proceedings of Applied Category Theory 2021, Electronic Proceedings in Theoretical Computer Science 372, pp. 235–248. ArXiv:2105.06332.

[6] David Chaum (1988): *The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability*. Journal of Cryptology 1(1), pp. 65–75, doi:10.1007/BF00206326.

[7] Kenta Cho & Bart Jacobs (2019): *Disintegration and Bayesian inversion via string diagrams*. Mathematical Structures in Computer Science 29(7), pp. 938–971.

[8] Bob Coecke, Tobias Fritz & Robert W Spekkens (2016): *A mathematical theory of resources*. Information and Computation 250, pp. 59–86.

[9] Andrea Corradini & Fabio Gadducci (1999): *An algebraic presentation of term graphs, via gs-monoidal categories*. Applied Categorical Structures 7(4), pp. 299–331.

[10] Alessandro Di Giorgio, Paweł Sobociński & Niels F. W. Voorneveld (2026): *Parametric Iteration in Resource Theories*. In Stefano Guerrini & Barbara König, editors: 34th EACSL Annual Conference on Computer Science Logic (CSL 2026), Leibniz International Proceedings in Informatics (LIPIcs) 363, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 29:1–29:23, doi:10.4230/LIPIcs.CSL.2026.29.

[11] Elena Di Lavore & Mario Román (2023): *Evidential decision theory via partial Markov categories*. In: 2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), IEEE, pp. 1–14.

[12] Cynthia Dwork (2008): *Differential Privacy: A Survey of Results*. In: Theory and Applications of Models of Computation—TAMC, Lecture Notes in Computer Science 4978, pp. 1–19. Available at `https://www.microsoft.com/en-us/research/publication/differential-privacy-a-survey-of-results/`.

[13] Tobias Fritz (2020): *A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics*. Advances in Mathematics 370, p. 107239, doi:10.1016/j.aim.2020.107239. ArXiv:1908.07021.

[14] Joseph A. Goguen & José Meseguer (1982): *Security Policies and Security Models*. In: 1982 IEEE Symposium on Security and Privacy, pp. 11–20, doi:10.1109/SP.1982.10014.

[15] Bart Jacobs, Aleks Kissinger & Fabio Zanasi (2021): *Causal Inference via String Diagram Surgery*. Mathematical Structures in Computer Science 31(5), pp. 553–574, doi:10.1017/S0960129521000098. ArXiv:1811.08338.

[16] Andre Joyal & Ross Street (1991): *The Geometry of Tensor Calculus,* I. Advances in Mathematics 88(1), pp. 55–112.

[17] Jonathan Katz & Yehuda Lindell (2014): *Introduction to Modern Cryptography, Second Edition*, 2nd edition. Chapman & Hall/CRC.

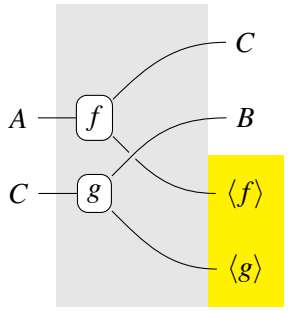
[18] Robin Piedeleu & Fabio Zanasi (2025): *An Introduction to String Diagrams for Computer Scientists*. Elements in Applied Category Theory, Cambridge University Press.

[19] P. Selinger (2010): *A Survey of Graphical Languages for Monoidal Categories* 813, pp. 289–355. doi:10.1007/978-3-642-12821-9_4.

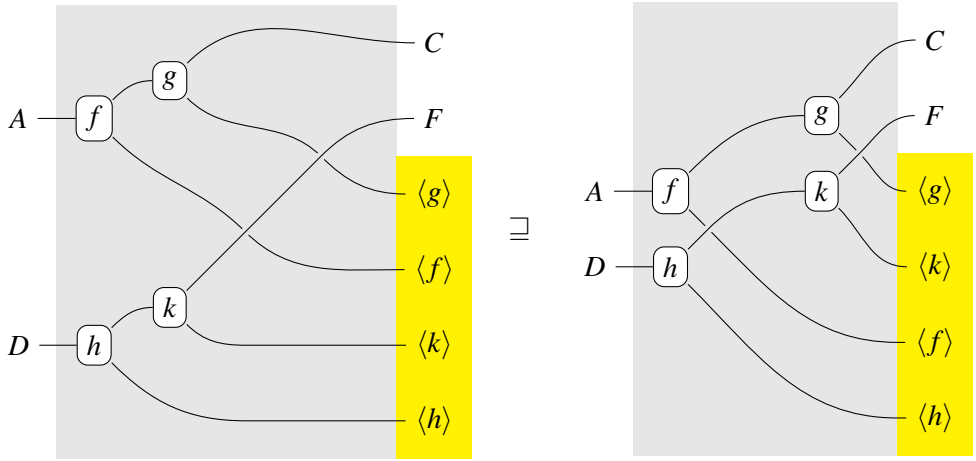
A Additional Proofs

Proof of Lemma 2

Proof. We define a monoidal product $\otimes$ on $\mathsf{L}(\mathbb{C})$ that on objects coincides with that of $\mathbb{C}$, and on morphisms $f\colon A \nrightarrow B$ and $g\colon C \nrightarrow D$ is defined as $f \otimes g\colon A \otimes C \nrightarrow B \otimes D$, consisting of $\langle f \otimes g\rangle := \langle f\rangle \otimes \langle g\rangle$ and $\widehat{f \otimes g} := (\widehat{f} \otimes \widehat{g}); (id_B \otimes \chi_{\langle f\rangle, D} \otimes id_{\langle g\rangle})$. The monoidal unit is inherited from $\mathbb{C}$. In diagrams, $f \otimes g$ is rendered as follows:



Associativity and unitality can be shown directly. To prove the interchange law we rely on $\equiv$. Given $f\colon A \nrightarrow B, g\colon B \nrightarrow C, h\colon D \nrightarrow E, k\colon E \nrightarrow F$, we use $id_{\langle g\rangle} \otimes \chi_{\langle f\rangle,\langle k\rangle} \otimes id_{\langle h\rangle}$ to prove that $(f \rhd g) \otimes (h \rhd k) \sqsupseteq (f \otimes h) \rhd (g \otimes k)$:

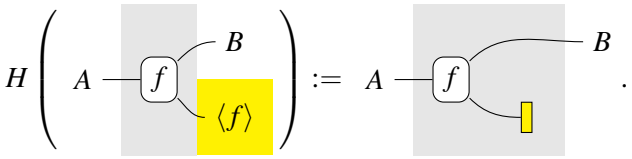


The other inclusion is proved analogously, since the morphism witnessing the inclusion is invertible.

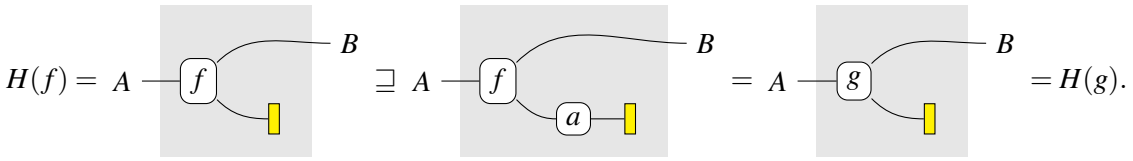
The symmetry $\chi_{A\otimes B}\colon A \otimes B \nrightarrow B \otimes A$ is given by $\langle \chi_{A,B}\rangle := I$ and $\widehat{\chi_{A,B}} := \chi_{A,B}\colon A \otimes B \to B \otimes A = B \otimes A \otimes I$ and we can easily see that $\chi_{A\otimes B} \rhd \chi_{B\otimes A} = i_{A\otimes B}$ and for any $f\colon A \nrightarrow B$, $(f \otimes i_C) \rhd \chi_{B\otimes C} \equiv \chi_{A\otimes C} \rhd (i_A \otimes f)$. □

Proof of Theorem 1

Proof. The functor $H\colon \mathsf{L}(\mathbb{C}) \to \mathsf{FL}(\mathbb{C})$ is the identity on objects, and on morphisms it sends $f\colon A \nrightarrow B$, given by $\langle f\rangle$ and $\widehat{f}\colon A \to B \otimes \langle f\rangle$, to $H(f) := \{\widehat{f}\}; (\{id_B\} \otimes \dashv_{\langle f\rangle})$. In diagrams:



H preserves the order: for any $f,g\colon A \nrightarrow B$ in $\mathsf{L}(\mathbb{C})$ such that $f \sqsupseteq g$, we know that there is some $a\colon \langle f\rangle \to \langle g\rangle$ such that $f;(id_B \otimes a) = g$ and

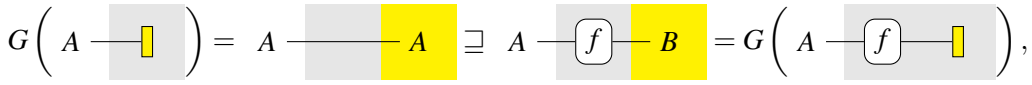


The functor $G\colon \mathsf{FL}(\mathbb{C}) \to \mathsf{L}(\mathbb{C})$ is the identity on objects and on morphisms is defined inductively as follows:

- $\langle G(\{f\})\rangle := I$, $\widehat{G(\{f\})} := f$;
- $\langle G(f;g)\rangle := \langle G(g)\rangle \otimes \langle G_L(f)\rangle$ and $\widehat{G(f;g)} := \widehat{G(f)};(\widehat{G(g)} \otimes id_{\langle G(f)\rangle})$;
- $\langle G(f \otimes g)\rangle := \langle G(f)\rangle \otimes \langle G(g)\rangle$ and $\widehat{G(f \otimes g)} := (\widehat{G(f)} \otimes \widehat{G(g)});(id_B \otimes \chi_{\langle G(f)\rangle,D} \otimes id_{\langle G(g)\rangle})$;
- $\langle G(\dashv_A)\rangle := A$ and $\widehat{G(\dashv_A)} := id_A\colon A \to A = I \otimes A$.

In diagrams:

G preserves the order. To show this, it is sufficient to see that it preserves the following inclusion:



where the inclusion in the middle is given by the fact that $id_A;(id_I \otimes f) = f$.

It readily follows from their definitions that H and G are mutually inverse, and hence $\mathsf{L}(\mathbb{C}) \cong \mathsf{FL}(\mathbb{C})$. □

Proof of Lemma 13

Let $f,g\colon A \to B$ be two morphisms in $\mathbb{C}$ and suppose $F(f) \equiv F(g)$. Since $\langle F(f)\rangle = \langle F(g)\rangle = I$, there are morphisms $a,b\colon I \to I$ such that $f;(id_B \otimes a) = g$ and $g;(id_B \otimes b) = f$.

Observe that $a = a;id_I = a;\varepsilon_I \leq \varepsilon_I = id_I$, and similarly $b \leq id_I$. Thus, in $\mathbb{C}$, $f = g;(id_B \otimes b) \leq g;(id_B \otimes id_I) = g$ and, similarly, $g = f;(id_B \otimes a) \leq f$, hence $f = g$.

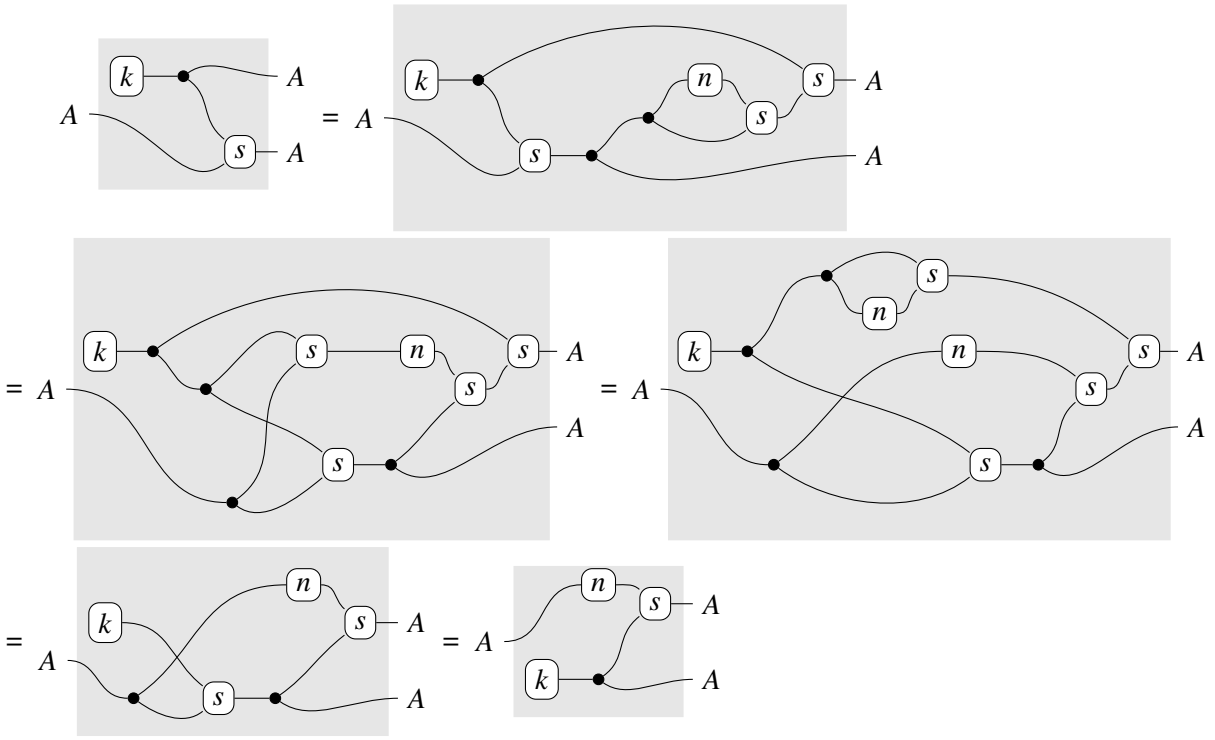
Proof of Lemma 15

If k is deterministic, then $e = k;\varepsilon;e = k;\delta;(n \otimes id_A)s = ((k;n) \otimes k);s = k;\varepsilon;k = k$, and hence $id_A = (e \otimes id_A);s = (k \otimes id_A);s = \varepsilon_A;k$, giving an isomorphism between A and I.

Proof of Lemma 16

Most properties hold due to standard commutative group theoretic results. E.g. the first equation corresponds to $(a \cdot b)^{-1} = b^{-1} \cdot a^{-1} = a^{-1} \cdot b^{-1}$.

We prove the second equation:



Proof of Lemma 17

The first equivalence holds due to commutativity of s.

By determinism, Lemma 8 gives us that $s; l_A \sqsubseteq (l_A \otimes l_A); s$, and hence $(id_A \otimes l_A); s; l_A \sqsubseteq (l_A \otimes (l_A; l_A)); s = (l_A \otimes l_A); s$.

$(id_A \otimes l_A); s; l_A = (id_A \otimes \delta_A); ((s; \delta_A) \otimes id_A); (id_A \otimes \dashv_A \otimes \dashv_A) \sqsupseteq (id_A \otimes \delta_A); ((s; \delta_A) \otimes id_A); (id_A \otimes (((n \otimes id_A); s); \dashv_A))$ which using determinism of s can be reduced to $(id_A \otimes l_A); s$.

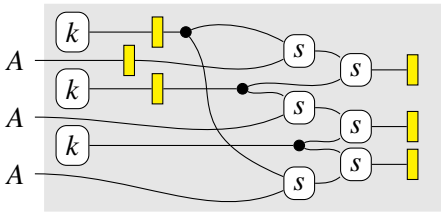
Hence $(l_A \otimes id_A); s; l_A = ((l_A; l_A) \otimes id_A); s; l_A \sqsupseteq (l_A \otimes l_A); s$.

These two inequalities together prove: $(l_A \otimes id_A); s; l_A \equiv (l_A \otimes l_A); s$.

Proof of Lemma 18

- $k; \dashv_A \sqsupseteq k; \varepsilon_A; \dashv_I = k; \varepsilon_A = id_I$.
- $k; \dashv_A \sqsupseteq \dashv_I = id_I$
- $e = e; id_A \sqsupseteq e; l_A$.
- $e = e \otimes \dashv_I \sqsupseteq e \otimes (e; \dashv_A) = e; \delta; (id_A \otimes \dashv_A) = e; l_A$.

A.1 Dining Cryptographers Problem Proof Sketch

Let: $f =$ 

We first prove that f is equivalent to $g = (l_A \otimes s); s; \dashv_A$.

- To prove that f leaks more than g, we use the order $(\dashv_A \otimes \dashv_A \otimes \dashv_A) \sqsupseteq (s \otimes id_A); s; \dashv_A$ first. Then by simplifying the statement, we can derive this to be equivalent to g.
- To prove that f leaks less than g, use Lemmas 17 and 7 to move the leakage identities around such that all the inputs are connected without any leaks between them (except for a leak at the top input). Then use Lemma 16 to rewrite the morphism to $(l_A \otimes s); s; h; (\dashv_A \otimes \dashv_A \otimes \dashv_A)$ for some morphism $h : A \to A^3$. That is, the morphism consists of combining all inputs with s, followed by some other morphism h, followed by leakage co-units. We then use $h; (\dashv_A \otimes \dashv_A \otimes \dashv_A) \sqsubseteq \dashv_A$ to rewrite the morphism to f.

So we have shown that $f \equiv g$. One final reasoning step allows us to derive that knowing one bit, then from parity of the sum of all three bits you can derive the parity of the sum of the other two bits, and vice versa. So we can derive: $f \equiv g \equiv \dashv_A \otimes (s; \dashv_A)$.

B Connections to related fields

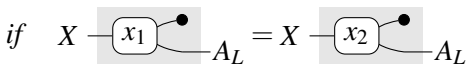
B.1 Non-interference

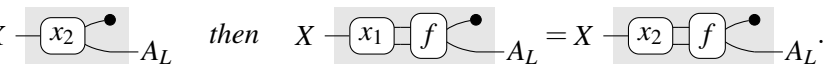
Non-interference is a multilevel security policy, introduced in [14], where the inputs and outputs of a process are classified as *lowly sensitive* or *highly sensitive*. The highly sensitive data should not influence the lowly sensitive data, otherwise some information may be leaked from the high to the low.

Supposing we have some process $f : A_H \otimes A_L \to B_H \otimes B_L$ with input and output partitioned into high and low sensitivity, non-interference requires that, for any two inputs of f that agree on A_L, the output of f also agrees on B_L.

The notion of input and what it means to have the same input and output can be expressed using the structure of copy-discard categories, leading us to the following formalisation of non-interference:

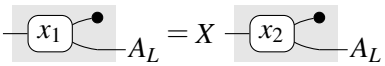
Definition 11. *A morphism $f : A_H \otimes A_L \to B_H \otimes B_L$ in $\mathbb{C}$ has the* non-interference *property if for any object X and morphisms $x_1, x_2 : X \to A_H \otimes A_L$,*

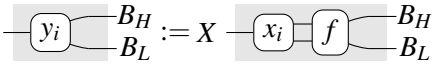
if 

then 

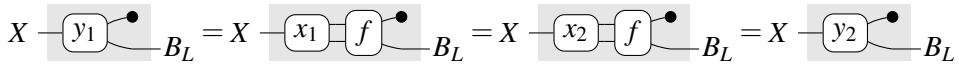
For instance, in `FinStoch` this boils down to producing the same distribution of outputs given any particular input. It is worth showing that non-interference is compositional.

Lemma 19. *If $f : A_H \otimes A_L \to B_H \otimes B_L$ and $g : B_H \otimes B_L \to C_H \otimes C_L$ in $\mathbb{C}$ have non-interference, then their composition $f; g : A_H \otimes A_L \to C_H \otimes C_L$ has non-interference.*

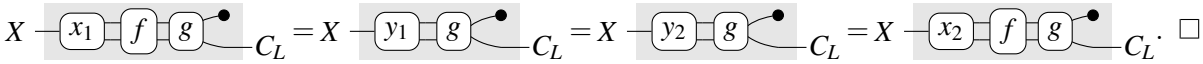
Proof. Let $x_1, x_2 \colon X \to A_H \otimes A_L$ be such that  and take



By non-interference of f, it holds that

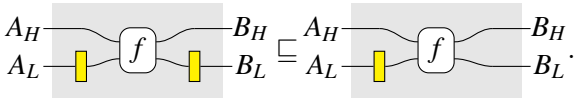


and hence, by non-interference of g, we can conclude that

 □

We can translate this property in terms of *leakage categories*, by observing that non-interference is essentially talking about whether any highly sensitive data is leaked. By default, we consider the lowly sensitive data as leaked, and ask ourselves whether the highly sensitive data is leaked. This leads us to the following definition of non-interference in $L(\mathbb{C})$, which requires that a morphism with leaked low inputs and outputs does not leak more than the same morphism where the output is not leaked.

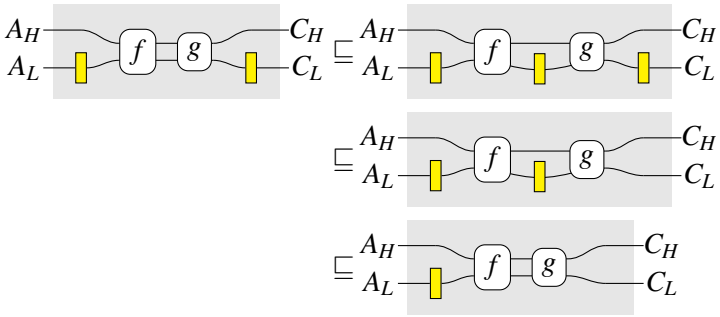
Definition 12. *A morphism $f : A_H \otimes A_L \to B_H \otimes B_L$ in $\mathbb{C}$ has the* leakage non-interference *property if in $L(\mathbb{C})$ it holds that*



It should be noted that, by Lemma 5, the inclusion above can be strengthened to an $\equiv$. As for the traditional definition, leakage non-interference is compositional:

Lemma 20. *If $f : A_H \otimes A_L \to B_H \otimes B_L$ and $g : B_H \otimes B_L \to C_H \otimes C_L$ in $\mathbb{C}$ have leakage non-interference, then their composition $f;g : A_H \otimes A_L \to C_H \otimes C_L$ has leakage non-interference.*

Proof.



(Lemma 5)

(*g* has leakage non-interference)

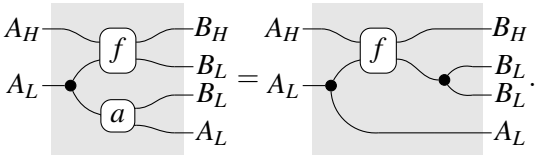
(*f* has leakage non-interference)

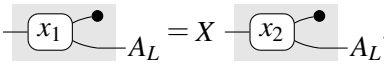
□

We now compare the two definitions of non-interference. In particular, when the morphism is total, we have the following implication:

Lemma 21. *If a total morphism $f : A_H \otimes A_L \to B_H \otimes B_L$ in $\mathbb{C}$ has leakage non-interference, then it has non-interference.*

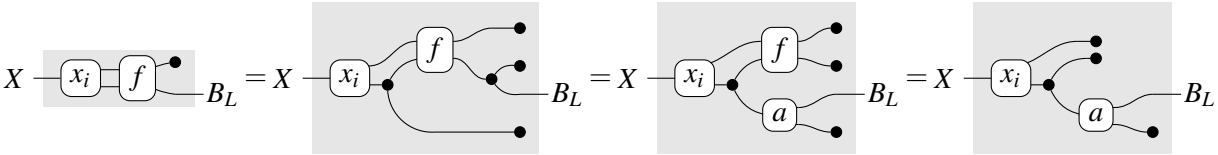
Proof. Suppose f has leakage non-interference, then, by definition of $\sqsubseteq$, there is a morphism $a\colon A_L \to B_L \otimes A_L$ in $\mathbb{C}$ such that

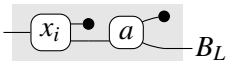


Given inputs $x_1, x_2\colon X \to A_H \otimes A_L$ such that 

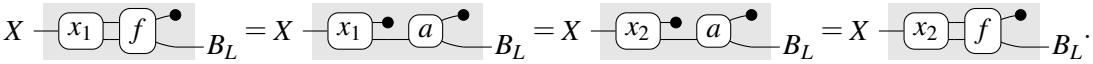
$= X$

, then for any $i \in \{1,2\}$





and hence



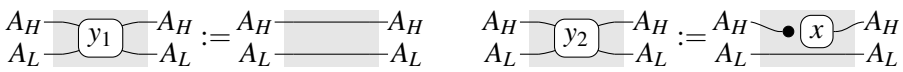
Therefore, we conclude that f has the non-interference property. □

The other direction is not true in general, since the traditional definition of non-interference does not consider to what extent the output B_H is leaked, whereas the leakage non-interference guarantees B_H is not leaked. The two definitions coincide when f is total and deterministic and the category is supplied with a total morphism $x\colon I \to A_H$.

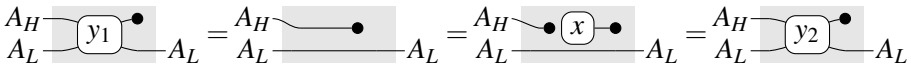
Remark 1. *The existence of a total morphism $x\colon I \to A_H$ is not an arbitrary requirement. Copy-discard categories with this additional structure already appear in the literature as copy-discard-uniform categories [15]. A common example is* `FinStoch`*, where x represents a uniform distribution.*

Theorem 2. *If there is a total morphism $x\colon I \to A_H$ in $\mathbb{C}$, then a total and deterministic morphism $f\colon A_H \otimes A_L \to B_H \otimes B_L$ in $\mathbb{C}$ has non-interference if and only if it has leakage non-interference.*

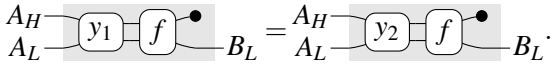
Proof. One direction follows from Lemma 21. For the other direction, take



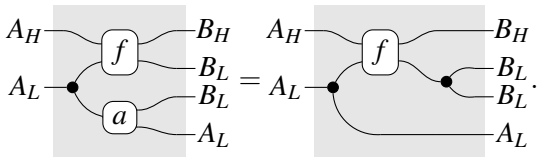
and observe that

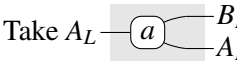


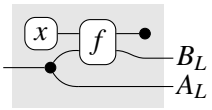
and, since f has non-interference, it holds that



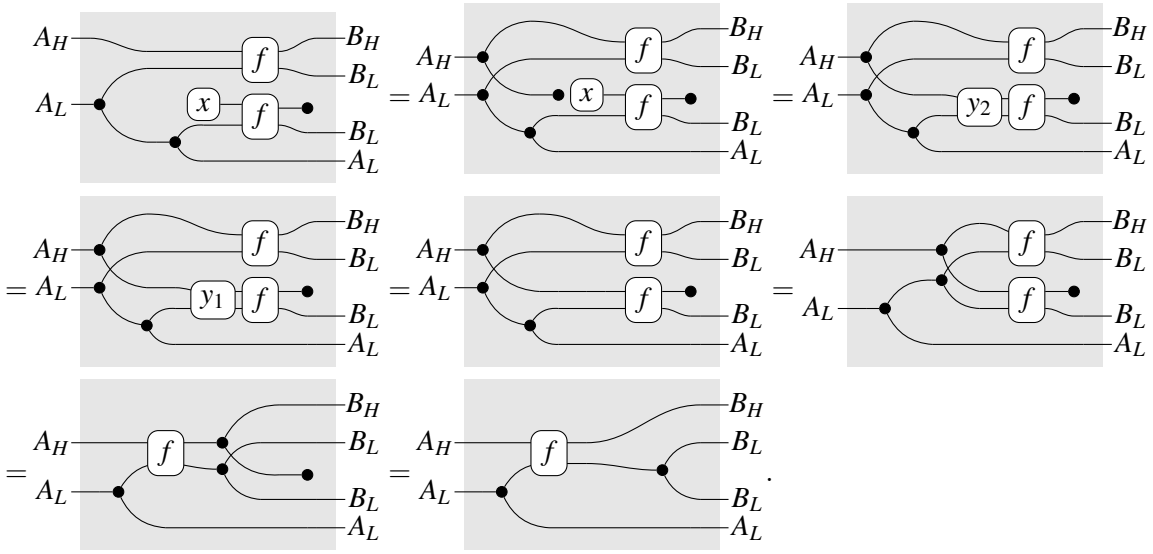
To prove that f has leakage non-interference we need to exhibit a morphism $a\colon A_L \to B_L \otimes A_L$ such that



Take 

$:=$


and observe that the following holds:



□

B.2 Blackwell Informativeness Theorem

The *Blackwell informativeness theorem* [1] establishes equivalent definitions for the order measuring informational content of *signals*. Specifically, supposing we work in `FinStoch`, let Ω be a set of possible worlds or states, then a morphism $\sigma : \Omega \multimap S$ in the category signifies a signal, with S some set of signals. For any possible world state ω, the distribution $\sigma(\omega)$ tells us which signal an *observer* sees, and wonders how useful this signal is in a wider context.

First, we consider the *utility* interpretation of σ. Suppose given some prior distribution of worlds $p : I \multimap \Omega$, and some utility function $u : \Omega \times A \multimap \mathbb{B}$ which accepts actions from some set A and produces a Boolean[1]. An observer receiving some signal $s \in S$ can then choose an action $a \in A$ to maximize the utility. This boils down to choosing some morphism $a : S \multimap A$ maximizing the probability that the following morphism produces $\texttt{true} \in \mathbb{B}$:

$$p \triangleright \delta_\Omega \triangleright (id_\Omega \otimes (\sigma \triangleright a)) \triangleright u$$

We write $W(\sigma, p, u) \in [0,1]$ for this maximal probability, given σ, p and u. Given two signals on the same set of worlds, $\sigma : \Omega \to S$, and $\sigma : \Omega \to S'$, then we write $\sigma \leq \sigma'$ if for any prior p and utility function

[1] Utility functions more commonly use some set of grades, e.g. the real numbers. We simplify the situation here, using distributions over $\mathbb{B}$ to mimic the interval $[0,1]$. Any utility using $\mathbb{R}$ over a finite set can be rescaled to fit into $[0,1]$.

u on Ω, $W(\sigma, p, u) \leq W(\sigma', p, u)$. The Blackwell informativeness theorem says that $\sigma \leq \sigma'$ holds if and only if there is some simulation function $f : S' \to S$ such that $\sigma' \triangleright f = \sigma$. One direction of the proof is rather trivial, since we immediately see that any optimal action $a : S \to A$ made by the observer seeing σ can be immediately transformed to an action $f \triangleright a : S' \to A$ for the observer seeing σ', meaning that the latter has actions at least as good as the former. The other direction usually requires some in-depth vector space analysis, out of the scope of the paper.

The equivalence between relations establishes a kind of *universal* notion of informational content hierarchy among signals. We observe that the second definition of the relation coincides with our notion of the *leaks more* relation, in the sense that $\sigma \leq \sigma'$ if and only if $\{\sigma\}\triangleright \dashv_S \sqsubseteq \{\sigma'\}\triangleright \dashv_{S'}$. Perhaps more appropriately to the utility interpretation, we can also observe that this is equivalent to $\delta_\Omega \triangleright (id_\Omega \otimes (\{\sigma\}\triangleright \dashv_S)) \sqsubseteq \delta_\Omega \triangleright (id_\Omega \otimes (\{\sigma'\}\triangleright \dashv_{S'}))$.

Taking a slightly wider perspective, consider two morphisms $f, f' : \Psi \to \Omega$ in $L(\texttt{FinStoch})$ represented by morphisms $\widehat{f} : \Psi \multimap \Omega \otimes S$ and $\widehat{f'} : \Psi \multimap \Omega \otimes S'$ in $\texttt{FinStoch}$ such that $\widehat{f} \triangleright (id_\Omega \otimes \varepsilon_S) = \widehat{f'} \triangleright (id_\Omega \otimes \varepsilon_{S'})$ which we denote as $p : \Psi \multimap \Omega$. Since we have conditionals, there is a $\sigma : \Omega \otimes \Psi \multimap S$ such that $f = \delta_\Psi \triangleright ((p \triangleright \delta_\Omega) \otimes id_\Psi) \triangleright (id_\Omega \otimes \sigma)$. Similarly, there is a $\sigma' : \Omega \otimes \Psi \multimap S'$ with the same relation to f'. We can think of f and f' as representing the same process p on sets of worlds, both leaking different amounts according to the specified signals. We indeed have that if $\sigma \leq \sigma'$, then σ' is better than σ in any utility context. One such context is exemplified by the measurement of how much is leaked from the process p, and indeed: $\sigma \leq \sigma'$ implies $f \sqsubseteq f'$. In the case that $\Psi = I$, we get back the scenario considered in the Blackwell informativeness theorem, with $p : I \multimap \Omega$ functioning as the prior. So we can think of leakage categories as a kind of generalisation in which the distribution of worlds is dependent on some input parameter space, and the signals may also carry information from this input parameter. If this input is given a prior distribution as well, we return to the setting of the theorem.