A process-algebraic approach for the analysis of probabilistic noninterference

Alessandro Aldini a,*, Mario Bravetti b and Roberto Gorrieri b

a Dipartimento di Scienze dell'Informazione, Università di Bologna, Italy
b Dipartimento di Scienze dell'Informazione, Università di Urbino, Italy

We define several security properties for the analysis of probabilistic noninterference as a conservative extension of a classical, nondeterministic, process-algebraic approach to information flow theory. We show that probabilistic covert channels that are not observable in the nondeterministic setting may be revealed through our approach and that probabilistic information can be exploited to give an estimate of the amount of confidential information flowing to unauthorized users. Finally, we present a case study showing that the expressiveness of the calculus we adopt makes it possible to model and analyze real concurrent systems.

1. Introduction

The analysis of information flow among different components of a concurrent computer system is a well established approach used for preventing unauthorized disclosure of confidential information (see, e.g. [5,9,29,32]). Among the different proposals that address such an analysis, we consider the notion of noninterference [32], which expresses a privacy property of systems. The central idea of noninterference is that a program is secure whenever the variation of the confidential data does not affect the program behavior as observed by an external attacker. The original notion of noninterference was defined for system programs that are deterministic. Generalized notions of noninterference were then designed to include nondeterministic systems on the basis of different system models (see, e.g. [41,43,62]) and here we concentrate on the use of process algebras (see, e.g. [24,37,46,48,49,51]). In particular, in this paper we consider the nondeterministic approach by Focardi and Gorrieri [24] and their classification of a set of security properties capturing the idea of information flow and noninterference [26]. In short, they employ an extension of CCS [45], where events are partitioned into two different levels of confidentiality (low level and high level), thus allowing the information flowing between the two different levels to be modeled and controlled. As an example, the most intuitive property they

---


*Corresponding author. Tel./Fax: +3907224475; E-mail: aldini@di.unibz.it.
The figure illustrates the probabilistic setting channel. The right part of the figure shows the Markov model with the following states: Random value, Low output, High output, and the corresponding probabilities. The model is designed to represent the probabilistic behavior of the system in response to random inputs. The model includes a transition matrix that defines the probabilities of moving from one state to another. The left part of the figure provides a textual explanation of the probabilistic setting channel, discussing the concept of a hidden Markov model and its applications in probabilistic modeling. The text explains how the model is used to predict the output of a system based on its current state and the input stimuli.
Appendix

12 Outline

I. Additional information from higher level to lower level

II. Related work
with some high-level loops of which 7 loops are
running the configuration. We observe that the
configuration is running on average 4 loops, and
it is to be expected that a higher number of
information flows is transported per loop.
Consequence: the configuration is running
much more flows than before, and we can
expect that the configuration is running much
more efficiently.

The configuration is running on average 9 loops,
and it is to be expected that a higher number of
information flows is transported per loop.

Consequence: the configuration is running
much more flows than before, and we can
expect that the configuration is running much
more efficiently.

The configuration is running on average 7 loops,
and it is to be expected that a higher number of
information flows is transported per loop.

Consequence: the configuration is running
much more flows than before, and we can
expect that the configuration is running much
more efficiently.

The configuration is running on average 8 loops,
and it is to be expected that a higher number of
information flows is transported per loop.

Consequence: the configuration is running
much more flows than before, and we can
expect that the configuration is running much
more efficiently.

The configuration is running on average 6 loops,
and it is to be expected that a higher number of
information flows is transported per loop.

Consequence: the configuration is running
much more flows than before, and we can
expect that the configuration is running much
more efficiently.
Example 2.2: Let us consider the system \( \mathcal{S} \) which has the following properties:

- The system has a high-level action that can be observed.
- The system has a low-level action that cannot be observed.

We define a high-level action as any action that can be directly observed. A low-level action is any action that cannot be directly observed.

Definition 2.2: A BNSNN sequence is defined as a sequence of high-level actions.

We now present the formal definition of a weak bisimulation over a set of relations.

Let \( \sim \) be a weak bisimulation relation. Then \( (\mathcal{S}, \sim) \) is a weak bisimulation if for all \( s, t \in \mathcal{S} \):

\[ s \sim t \iff \exists s', t' \in \mathcal{S} \text{ such that } s' \sim t' \text{ and } s' \sim s \text{ and } t' \sim t \]

Implications for all \( s, t \in \mathcal{S} \):

- If \( s \sim t \), then there exists \( s', t' \) such that \( s' \sim t' \).
- If \( s' \sim s \) and \( t' \sim t \), then \( s' \sim t' \).

We also define \( (\mathcal{S}, \prec) \) as a weak bisimulation if for all \( s, t, s', t' \in \mathcal{S} \):

\[ s \prec t \iff \exists s', t' \in \mathcal{S} \text{ such that } s' \prec t' \text{ and } s' \prec s \text{ and } t' \prec t \]

Implications for all \( s, t \in \mathcal{S} \):

- If \( s \prec t \), then there exists \( s', t' \) such that \( s' \prec t' \).
- If \( s' \prec s \) and \( t' \prec t \), then \( s' \prec t' \).

We now consider the following scenario:

- \( s \in \mathcal{S} \) represents a high-level action.
- \( t \in \mathcal{S} \) represents a low-level action.

Based on the above definitions, we can conclude the following:

- If \( s \sim t \) and \( s' \sim s \), then \( s' \sim t \).
- If \( s \prec t \) and \( s' \prec s \), then \( s' \prec t \).

We then define the following for some \( n \in \mathbb{N} \):

- \( (\mathcal{S}, \prec_n) \) is a weak bisimulation.
- \( (\mathcal{S}, \sim_n) \) is a weak bisimulation.
Definition 2.4. $P \in G$ is BDNC secure if and only if $P(\Pi \in [G]) \models \Pi(\Pi \in G) \models \Pi(\Pi \in G)$ for all $\Pi$ in the context of our calculus may be as follows.

The left-hand term represents the low view of the system in isolation, while the right-hand term represents the low view of the system in isolation, while the right-hand term represents the low view of the system in isolation. Such a formalism is more clearly specified by the right-hand term.

Example 2.7. Let us consider an infinite set of terms, $\forall \Pi \in G$, the condition needed by BDNC requires to check a finite number of terms, which are associated with a finite set of terms.

With respect to the definition of BDNC, we have to check the property for an infinite set of terms, $\forall \Pi \in G$, the condition needed by BDNC requires to check a finite number of terms, which are associated with a finite set of terms.

Definition 2.6. $P \in G$ is BDNC secure if and only if $P(\Pi \in G) \models \Pi(\Pi \in G)$ for all $\Pi$ in the context of our calculus may be as follows.

The left-hand term represents the low view of the system in isolation, while the right-hand term represents the low view of the system in isolation. Such a formalism is more clearly specified by the right-hand term.

Example 2.8. Let us consider an infinite set of terms, $\forall \Pi \in G$, the condition needed by BDNC requires to check a finite number of terms, which are associated with a finite set of terms.

With respect to the definition of BDNC, we have to check the property for an infinite set of terms, $\forall \Pi \in G$, the condition needed by BDNC requires to check a finite number of terms, which are associated with a finite set of terms.

Because of the presence of the universal quantification on all possible high-level actions in the context of our calculus, we have to check the property for an infinite set of terms, $\forall \Pi \in G$, the condition needed by BDNC requires to check a finite number of terms, which are associated with a finite set of terms.
3. General reactive transition systems

A general reactive transition system (GRTS) is a quadruple (S, A, δ, τ) where:
- S is a non-empty set of states
- A is a set of actions
- δ: S × A → P(S) is a transition relation
- τ: A → P(A) is a reaction relation

Definition 3.1: A general reactive transition system (GRTS) is a quadruple (S, A, δ, τ).

In the context of our model, we introduce the notion of general reactive transition systems. From the perspective of probabilistic models, we can easily define a probability distribution on the actions of a GRTS. This allows us to incorporate probabilistic choice among actions, making the model more realistic and reflective of real-world systems.

In Figure 4, we illustrate an example of a general reactive transition system. The system consists of a set of states and actions, with transitions and reactions defined accordingly.
The composition process, as described in the previous section, allows for the combination of two processes to form a new one. This is achieved by defining a new process as the composition of two existing processes, where the output of the first process serves as the input to the second process.

**Example:** Consider two processes, P and Q, with inputs a and b, respectively. Let P(a) = b and Q(b) = c. The composition P ∘ Q is defined as P(Q(a)).

The composition of processes can also be extended to more complex scenarios, allowing for the creation of intricate systems. The key is to ensure that the output of one process is compatible with the input requirements of the subsequent process.

Diagram representation of the composition operation is as follows:

```
( a ) \rightarrow P \rightarrow ( b ) \rightarrow Q \rightarrow ( c )
```

In this diagram, the arrows represent the flow of information from one process to the next, with the output of one process being the input for the next.

**Properties of Composition:**

1. **Associativity:** (P ∘ Q) ∘ R = P ∘ (Q ∘ R)
2. **Identity:** If I is the identity process (I(a) = a for all a), then P ∘ I = P = I ∘ P
3. **Composition of Operations:** The composition of operations can be defined as the process of applying one operation after another, with the result being the output of the final operation.

These properties are fundamental in the study of process composition and are essential for understanding the behavior of complex systems.
The formal semantics of our calculus is given by the following operational rules:

\[ \frac{}{P \rightarrow P} \]

and

\[ \frac{Q \rightarrow Q, Q \rightarrow R}{P, Q \rightarrow R} \]

where

- \( P \) and \( Q \) are expressions,
- \( \rightarrow \) denotes operational transition,
- \( \frac{}{P \rightarrow P} \) is the identity rule,
- \( \frac{Q \rightarrow Q, Q \rightarrow R}{P, Q \rightarrow R} \) is the substitution rule.

The operational rules define a notion of execution, which is used to reason about the behavior of programs.

Example: To evaluate the expression \( \frac{a + b}{c} \), we perform the following steps:

1. Evaluate \( a + b \) to get \( 3 + 4 = 7 \).
2. Evaluate \( c \) to get \( 2 \).
3. Divide \( 7 \) by \( 2 \) to get \( 3.5 \).

Hence, \( \frac{a + b}{c} = 3.5 \).
4. Equivalence

As shown in [11], we have shown that the operational semantics of a process is

\[ \frac{d \equiv d}{\text{Operational semantics (part II)}} \]

Moreover, note that the probability of the deterministic transitions of \( d \) is defined by

\[ P(d) = \sum_{i, e} p_i \cdot e \]

where \( p_i \) is the probability of the \( i \)-th deterministic transition of \( d \).
The definition of weak probabilistic bisimulation for [P]MS depends on the following definition:

\[ \text{Definition 1.2: An equivalence relation } R \subseteq X \times X \text{ is a weak probabilistic bisimulation if } \]

\[ \forall x, y \in X : x R y \Rightarrow (\exists z \in X : (\exists z' \in X : z R z' \land z' M \Rightarrow y)) \land (\exists z'' \in X : (\exists z''' \in X : z'' R z''' \land z''' M \Rightarrow y)) \]

Also, let consider the two OTHS of Fig. 7, which we can use to model these two probabilistic bisimulation relations. To show each a local LR is as the entry point.

**Example 1.4:** Let us consider the two OTHS of Fig. 7, which we can use to model these two probabilistic bisimulation relations. To show each a local LR is as the entry point.

**Theorem 1.1:** The set of sequences \( \omega = \omega \uparrow \) in a language is closed under a local LR, which we can use to model these two probabilistic bisimulation relations.

In the following definition, we define a probabilistic bisimulation for [P]MS and show how these two probabilistic bisimulation relations are modelled by the probabilistic systems of (8) [P]. The definition of weak probabilistic bisimulation for [P]MS depends on the following definition:

\[ \text{Definition 1.2: An equivalence relation } R \subseteq X \times X \text{ is a weak probabilistic bisimulation if } \]

\[ \forall x, y \in X : x R y \Rightarrow (\exists z \in X : (\exists z' \in X : z R z' \land z' M \Rightarrow y)) \land (\exists z'' \in X : (\exists z''' \in X : z'' R z''' \land z''' M \Rightarrow y)) \]

Also, let consider the two OTHS of Fig. 7, which we can use to model these two probabilistic bisimulation relations. To show each a local LR is as the entry point.

**Example 1.4:** Let us consider the two OTHS of Fig. 7, which we can use to model these two probabilistic bisimulation relations. To show each a local LR is as the entry point.

**Theorem 1.1:** The set of sequences \( \omega = \omega \uparrow \) in a language is closed under a local LR, which we can use to model these two probabilistic bisimulation relations.

In the following definition, we define a probabilistic bisimulation for [P]MS and show how these two probabilistic bisimulation relations are modelled by the probabilistic systems of (8) [P]. The definition of weak probabilistic bisimulation for [P]MS depends on the following definition:

\[ \text{Definition 1.2: An equivalence relation } R \subseteq X \times X \text{ is a weak probabilistic bisimulation if } \]

\[ \forall x, y \in X : x R y \Rightarrow (\exists z \in X : (\exists z' \in X : z R z' \land z' M \Rightarrow y)) \land (\exists z'' \in X : (\exists z''' \in X : z'' R z''' \land z''' M \Rightarrow y)) \]

Also, let consider the two OTHS of Fig. 7, which we can use to model these two probabilistic bisimulation relations. To show each a local LR is as the entry point.

**Example 1.4:** Let us consider the two OTHS of Fig. 7, which we can use to model these two probabilistic bisimulation relations. To show each a local LR is as the entry point.

**Theorem 1.1:** The set of sequences \( \omega = \omega \uparrow \) in a language is closed under a local LR, which we can use to model these two probabilistic bisimulation relations.

In the following definition, we define a probabilistic bisimulation for [P]MS and show how these two probabilistic bisimulation relations are modelled by the probabilistic systems of (8) [P]. The definition of weak probabilistic bisimulation for [P]MS depends on the following definition:

\[ \text{Definition 1.2: An equivalence relation } R \subseteq X \times X \text{ is a weak probabilistic bisimulation if } \]

\[ \forall x, y \in X : x R y \Rightarrow (\exists z \in X : (\exists z' \in X : z R z' \land z' M \Rightarrow y)) \land (\exists z'' \in X : (\exists z''' \in X : z'' R z''' \land z''' M \Rightarrow y)) \]

Also, let consider the two OTHS of Fig. 7, which we can use to model these two probabilistic bisimulation relations. To show each a local LR is as the entry point.

**Example 1.4:** Let us consider the two OTHS of Fig. 7, which we can use to model these two probabilistic bisimulation relations. To show each a local LR is as the entry point.

**Theorem 1.1:** The set of sequences \( \omega = \omega \uparrow \) in a language is closed under a local LR, which we can use to model these two probabilistic bisimulation relations.
In the section, we present these probabilistic security properties by extending the notion of the security of a system to define a probabilistic extension of the security.

### Definition 5.1

**Definition 5.1.** A system is **probabilistically secure** if and only if

\[
\forall i,j \in [n] \quad \Pr[A_i \land A_j] = \Pr[i \lor j] \leq \frac{1}{2}
\]

We can now move on to defining probabilistic extensions of the security properties.

### 5.1 Probabilistic Extensions of Security Properties

We consider the probability of observing an event without any additional information. How can we estimate the probability of an event occurring? We show that the probability of observing an event occurring is bounded by the maximum probability of observing the event occurring in a system.

#### Lemma 5.1

Given a system, if \( A \subseteq \mathcal{G} \) and \( B = A \), then \( \Pr[A] \geq \Pr[B] \). Furthermore, if \( \Pr[A] = \Pr[B] \), then for all \( i \in [n] \), \( \Pr[A_i] = \Pr[B_i] \).

#### Proof

\[
\Pr[A] = \sum_{i=1}^{n} \Pr[A_i] = \sum_{i=1}^{n} \Pr[B_i] = \Pr[B]
\]

As for the probability of observing an event without any additional information.
by the proxy process by composing the proxy information in the region

2.1. Composing proxy information from

The mean of the system means and does not reveal any other information to the

Example 2.2: An example of a system to consider a non-probabilistic

\[ \frac{d\langle y \rangle}{dt} = \frac{d}{dt} \int_0^T \langle y \rangle \, dt \]

Example 2.3: Let us consider the system where

\[ \frac{d\langle y \rangle}{dt} = \frac{d}{dt} \int_0^T \langle y \rangle \, dt \]

Example 2.4: Consider a non-probabilistic system with

\[ \frac{d\langle y \rangle}{dt} = \frac{d}{dt} \int_0^T \langle y \rangle \, dt \]

Example 2.5: Consider a non-probabilistic system with

\[ \frac{d\langle y \rangle}{dt} = \frac{d}{dt} \int_0^T \langle y \rangle \, dt \]

Example 2.6: Consider a non-probabilistic system with

\[ \frac{d\langle y \rangle}{dt} = \frac{d}{dt} \int_0^T \langle y \rangle \, dt \]

Example 2.7: Consider a non-probabilistic system with

\[ \frac{d\langle y \rangle}{dt} = \frac{d}{dt} \int_0^T \langle y \rangle \, dt \]

Example 2.8: Consider a non-probabilistic system with

\[ \frac{d\langle y \rangle}{dt} = \frac{d}{dt} \int_0^T \langle y \rangle \, dt \]

Example 2.9: Consider a non-probabilistic system with

\[ \frac{d\langle y \rangle}{dt} = \frac{d}{dt} \int_0^T \langle y \rangle \, dt \]

Example 2.10: Consider a non-probabilistic system with

\[ \frac{d\langle y \rangle}{dt} = \frac{d}{dt} \int_0^T \langle y \rangle \, dt \]
Figure 10. Example of a system consisting of a probabilistic controller and a plant.

The non-deterministic system of Process P is defined and BNDK, with the P

Example 5: Let us consider the following probabilistic system:

\[\{(d^{-1}t + d^{-10})\eta, d + (d^{-1}t + d^{-10})\eta\} \approx d\]

where \(d\) is a action of an action \(d\) with a probability of

\[\{(1 + \eta) = d\}

which is the CTRV in the low-probability setting due to the non-deterministic nature of the system. The probability of the system being in state \(d\) after \(t\) time units is given by:

\[\text{Probability} = \frac{1}{2}(1 + \eta)\]

The system is said to be in state \(d\) if and only if \(\eta = 1\) or \(\eta = 0\).
Example 2.8. Let us consider the process:

\[ \mathcal{P}(d) = (d^+ + 0)^+ a + (d^+ + 0)^+ b + (d^+ + 0)^+ c + (d^+ + 0)^+ d \]

where the corresponding ORS is shown in Fig. 12.1. It is possible that \( \mathcal{P}(d) \in \mathbb{R} \).

Proposition 2.7. THE \( \mathcal{P} \) is defined for \( \mathcal{P} \in \mathbb{R} \).

**Proof:**

For \( \mathcal{P} \in \mathbb{R} \), the \( \mathcal{P} \) is defined for \( \mathcal{P} \in \mathbb{R} \).

**Remark:**

Since the \( \mathcal{P} \) is defined for \( \mathcal{P} \in \mathbb{R} \), it is possible that \( \mathcal{P} \in \mathbb{R} \).

Somewhat analogous to the \( \mathcal{P} \) is defined for \( \mathcal{P} \in \mathbb{R} \), the \( \mathcal{P} \) is defined for \( \mathcal{P} \in \mathbb{R} \).

Proposition 2.8. THE \( \mathcal{P} \) is defined for \( \mathcal{P} \in \mathbb{R} \).

**Proof:**

For \( \mathcal{P} \in \mathbb{R} \), the \( \mathcal{P} \) is defined for \( \mathcal{P} \in \mathbb{R} \).

**Remark:**

Since the \( \mathcal{P} \) is defined for \( \mathcal{P} \in \mathbb{R} \), it is possible that \( \mathcal{P} \in \mathbb{R} \).

Proposition 2.9. THE \( \mathcal{P} \) is defined for \( \mathcal{P} \in \mathbb{R} \).

**Proof:**

For \( \mathcal{P} \in \mathbb{R} \), the \( \mathcal{P} \) is defined for \( \mathcal{P} \in \mathbb{R} \).

**Remark:**

Since the \( \mathcal{P} \) is defined for \( \mathcal{P} \in \mathbb{R} \), it is possible that \( \mathcal{P} \in \mathbb{R} \).
property. Therefore, the source condition with the SNC property is in any sense of the term an upper bound on the source distribution of the random variable $X$. In fact, we can show that the source distribution of $X$ is also a source distribution of a random variable $Y$ such that $Y$ is an upper bound on the source distribution of $X$.

Theorem 3.1: If $X$ has the SNC property, then $X$ is also a source distribution of a random variable $Y$ such that $Y$ is an upper bound on the source distribution of $X$.

In general, by following a basic induction similar to that shown for $SNC$ [33], we can prove that $H_{\text{SNC}}(X) \leq H_{\text{SNC}}(Y)$ for any random variable $Y$ such that $Y$ is an upper bound on the source distribution of $X$.

Definition 3.2: A random variable $X$ is said to have the SNC property if there exists a sequence $\{S_n\}$ of random variables such that $X = \lim_{n \to \infty} S_n$ and $H_{\text{SNC}}(S_n) \leq H_{\text{SNC}}(S_{n+1})$ for all $n$.

Example 3.3: The process $\{S_n\}$ of Example 3.2, where $S_n$ is the conditional expectation of $S_{n+1}$ given $S_n$ and $\text{SNC}(S_{n+1}) \leq \text{SNC}(S_n)$, satisfies the SNC property.

In this section, we introduce the probabilistic security property further than the one defined in [33].

Strong probabilistic security

The SNC property, which extends the probabilistic security the SNC property of Sec., is the property that satisfies the source condition. In this section, we define the probabilistic version of the SNC property, which we call SNC.

In this section, we introduce the probabilistic version of the SNC property, which we call SNC.

Hence, we define the probabilistic version of the SNC property, which we call SNC.

In this section, we introduce the probabilistic version of the SNC property, which we call SNC.
In this section, we show that any property of our system can be expressed using the properties of PBANDC and SBANDC.

Theorem 1. SPANDC \iff PBANDC \amp SBANDC

\textbf{Example.} The multiplicative group of finite fields of order \(p\), where \(p\) is prime, is a PBANDC group. This can be verified by checking that the group operation is both associative and that every element has a multiplicative inverse.

The proof of this theorem is given in the appendix. It follows from the definitions of PBANDC and SBANDC.
By modifying the probabilistic behavior of systems, new aspects of potential in...
Example 5.3: A more concrete example: let us consider the probabilistic process of EX.

Example 5.12: Let us consider the process.

Theodore. I am not too excited with any corresponding behavior of the system, since this is expressed by the result of the higher level action, which is in general the lower level of the process.

According to the definition of probabilistic behavior, if there is a probabilistic process of EX, then it is a probabilistic process of EX. This is an approximation of a probabilistic process.
Anomaly detection is a critical component in the overall process. The parameter $\phi$ expresses the anomaly score, which is then compared to a threshold $\theta$ to determine if an anomaly occurs. If the score exceeds the threshold, an anomaly is flagged. The anomaly detection process is divided into two main steps: feature extraction and thresholding.

1. **Feature Extraction**: This step involves extracting relevant features from the input data. Features can be extracted using various methods, including statistical methods, machine learning techniques, and domain-specific knowledge.

2. **Thresholding**: Once the features are extracted, they are compared against a threshold to determine if an anomaly exists. The threshold is a critical parameter that needs to be carefully tuned to balance the trade-off between false positives and false negatives.

The anomaly detection process is then applied to the input data to identify any anomalies that may be present. The output of the anomaly detection process is the anomaly score for each input sample, which can be used to monitor the system's health and detect any unusual behaviors.

To illustrate this process, consider the following example:

**Example 1**: Consider a simple system with two sensors, $S_1$ and $S_2$, that measure the temperature and humidity levels in a room. The sensors provide continuous readings, and the system detects an anomaly if the readings exceed predefined thresholds.

To detect anomalies, the system could use a simple thresholding approach where the anomaly score is calculated as:

$$\text{Anomaly Score} = \begin{cases} 1 & \text{if } T_1 > \theta_1 \text{ or } H_2 > \theta_2 \\ 0 & \text{otherwise} \end{cases}$$

where $T_1$ and $H_2$ are the readings from sensors $S_1$ and $S_2$, respectively, and $\theta_1$ and $\theta_2$ are the predefined thresholds.

### Further Reading

For a more detailed discussion on anomaly detection and its applications, refer to the following resources:

1. **Anomaly Detection in Real-Time Systems**
2. **Machine Learning for Anomaly Detection**
3. **Statistical Methods for Anomaly Detection**
6.2 Security analysis of the NIP

The corresponding receive action of airm routing (a route) is synchronized with the transmission where the receive action of the corresponding route (a route) is executed. Then the action of cmp routing (cmp) is executed, the control output of the cmr action is generated, and the action of cmp routing is generated. The process described in Section 6 is the process modeled by the action model, where the correspond action is the action of cmp routing (cmp).

Now we are ready to analyze any information flow in the NIP model. The system action of the NIP model is

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmp</td>
<td>cmp routing</td>
</tr>
<tr>
<td>cmr</td>
<td>cmr action</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

A fiber in a process is a fiber in a process model in productive environment.
7. Conclusion

In this paper, we have addressed the problem of extracting the non-verbal information from a document. We have explored various approaches and techniques for this task, including natural language processing and computer vision. We have shown that it is possible to extract meaningful information from documents using these methods, and have presented a number of examples of how this can be done.


A model for the problem of extracting non-verbal information from documents is as follows:

$$\text{Input} \rightarrow \text{Model} \rightarrow \text{Output (Extracted Information)}$$

The input to the model is the document itself, and the output is the extracted non-verbal information. The model itself is a complex system that combines various techniques and algorithms to achieve this goal.

In the future, we plan to further refine our model and improve its accuracy. We also plan to explore new approaches and techniques for extracting non-verbal information from documents, in order to make our system even more effective.

We believe that our work has important implications for a wide range of applications, including natural language processing, computer vision, and more. We hope that our findings will be of interest to researchers and practitioners in these fields, and that they will find our work useful for their own work.

Acknowledgments

We would like to thank the reviewers for their feedback on our paper. We are grateful for their suggestions, which have helped us to improve our work. We would also like to thank our colleagues for their support and encouragement.

References


A.

Proposition 5.7

We now show that $\mathcal{D} \subseteq \mathcal{B}$.

Let $\mathcal{D}$ be a collection of sets $\{A_1, A_2, \ldots, A_n\}$ where $A_i \subseteq \mathcal{X}$ for $i = 1, 2, \ldots, n$. We want to prove that $\mathcal{D} \subseteq \mathcal{B}$.

Let $\mathcal{X}$ be a set of elements, and let $\mathcal{D}$ be a collection of subsets of $\mathcal{X}$. We assume that $\mathcal{D}$ is a duality-preserving family.

A.

Theorem 7.1

Let $\mathcal{D}$ be a collection of sets $\{A_1, A_2, \ldots, A_n\}$ where $A_i \subseteq \mathcal{X}$ for $i = 1, 2, \ldots, n$. We want to prove that $\mathcal{D} \subseteq \mathcal{B}$.

Let $\mathcal{X}$ be a set of elements, and let $\mathcal{D}$ be a collection of subsets of $\mathcal{X}$. We assume that $\mathcal{D}$ is a duality-preserving family.
\[
\sum_{X} \left( u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right) \cdot \frac{Y}{X} \right) = \frac{u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right) \cdot \frac{Y}{X}}{u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right) \cdot \frac{Y}{X}}
\]

and
\[
\sum_{X} \left( u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right) \cdot \frac{Y}{X} \right) = \frac{u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right) \cdot \frac{Y}{X}}{u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right) \cdot \frac{Y}{X}}
\]

Since \( u^n \) is SPNDG-pseudorandom if \( n > 0 \), then
\[
\sum_{X} u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right) = \frac{u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right)}{u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right)}
\]

and then
\[
\sum_{X} u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right) = \frac{u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right)}{u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right)}
\]

from which we can deduce that
\[
\sum_{X} u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right) = \frac{u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right)}{u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right)}
\]

By definition, the expected value of \( \sum_{X} u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right) \) does not depend on \( X \).

Now, suppose \( \sum_{X} u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right) = 0 \). Then, we have
\[
\sum_{X} u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right) = 0
\]

Hence, \( u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right) = 0 \). Therefore, \( u^{d_2}Y + u^{d_1}Z \cdot \left( u^{d_1} - 1 \right) = 0 \).
References

We can now simplify the SPM property and the SPM property to

\[ \text{SPM} \Rightarrow \text{SPM} \]

by removing the probability parameters from the parameters. If \( \lambda_d, \lambda_{d'} \in \mathbb{R} \) and \( \lambda_{d'} \in \mathbb{R} \), then we get

\[ \text{SPM} = \left\{ a \mid a \in \mathbb{R} \right\} \]

where the condition of SPIM consists of the domain of the function.

To further simplify the SPM property, we introduce the following criterion:

\[ \text{SPM} = \left\{ a \mid a \in \mathbb{R} \right\} \]

We prove the statement in the case of \( p \in \mathbb{R} \) and \( p \in \mathbb{R} \) by

A.6. Constraint texture theorem

The constraint texture theorem says that if \( p \in \mathbb{R} \) and \( p \in \mathbb{R} \) then

\[ \text{SPM} = \left\{ a \mid a \in \mathbb{R} \right\} \]

The constraint texture theorem states that if \( p \in \mathbb{R} \) and \( p \in \mathbb{R} \) then

\[ \text{SPM} = \left\{ a \mid a \in \mathbb{R} \right\} \]

By following the same argumentation, it is easy to prove the theorem in the special case of \( p \in \mathbb{R} \) and \( p \in \mathbb{R} \).
A Model of a Process-Dependent Approach to Productive Interference

A. A Model of a Process-Dependent Approach to Productive Interference

B. A Model of a Process-Dependent Approach to Productive Interference

C. A Model of a Process-Dependent Approach to Productive Interference

D. A Model of a Process-Dependent Approach to Productive Interference