1 Introduction

Formulation of information flow security model requires at least the following ingredients : an abstract model to represent the system under observation and to describe its behaviour, and security properties or constraints which are concerned with and regulate the flow of information between different parts of the system.

Several information flow security models have been proposed over the recent years which include Non-Interference (NI, [GM82]), Non-Deducibility on Input (NDI, [Sut86]), Non-Deducibility on Strategy (NDS, [WJ90]), Restrictiveness (RES, [McC88]), and Forward Correctability (FC, [JT88]), Non-Deducibility on Strategy (NDS, [WJ90]) and Information Flow Secure Nets (IFSN, [Var90]).

An important characteristic of an information flow security property is whether or not it is preserved under composition. This feature is extremely important as systems are often connected together to form composite or networked systems thereby sharing information and resources. Moreover, since the design and analysis of a complex system is often carried out by dividing it into smaller cooperating subsystems, it is very important for a designer to be provided with a tool which enables him to design a secure system step by step, in a divide and conquer fashion.

Non-Interference was one of the first proposed information flow security properties. In this approach, a deterministic automaton is used to model systems. It takes an input stream from the users and generates deterministically, for each user, an output, that is a particular view of the system for that user. The presence or absence of information flow is determined using the concept of non interference between users whereby a user is said to be interfering with another user if he can, by changing his inputs to the system, modify the view of the other. A nice feature of the Non-Interference approach is its simplicity, while its main limitation is the choice of a deterministic model.

The search for a security model able to treat non-deterministic systems led to the definition of Non-Deducibility on Inputs. NDI still has a major problem in that it is not composable since an insecure system may be obtained as a result of connecting two secure systems.

Steps towards a suitable definition of security enjoying the composability property led to the definitions of RES and FC. Though they achieve composability, they introduce other restrictions which are somewhat unrealistic. RES assumes that systems are *input total*, that is, systems are always ready to take inputs delivered to them or, equivalently, systems are infinitely fast. FC assumes that systems are *input extensible*, a property equivalent to the input total one; a system can collect *any number* of inputs from the users before delivering the first output.

The Non-Deducibility on Strategies approach extends the NDI approach by taking into account in the definition, the possible behaviour of the system when a composition operation is executed. The model is sufficiently general to treat non-deterministic systems. However, it is synchronous in that the users send their inputs and receive their outputs at the same time.

The Information Flow Security Net model describes a security property based on the actual flow of information. The model is able to deal with non-deterministic systems and can handle asynchronous behaviour. Work is being carried out in defining composition operators which preserve the information flow security property when two such nets are connected together. A paper describing this work is being presented at the 1991 Computer Security Foundations Workshop [Var91].

From above, we can see that there has been a large amount of work done in the area of information flow security; however, it seems that not much effort has been dedicated to the consideration of the underlying model, a matter the present paper is devoted to. We believe that the underlying system model plays an important role in the formulation of an information flow security model.

Different notions of information flow security that have been presented are often based on different underlying system models. This disallows the possibility of a true comparison between the different information flow definitions, and a proper understanding of their *mutual relationships*. Moreover, it obstructs the comprehension of the *true power* of an information flow security property definition. Such observations constitute the basis for our belief that there is a need for a unifying underlying framework and the need for a model without many unrealistic assumptions.

The model we present in this paper is based on Petri nets and uses the work described in [Var89], [Var90] and [Rou86]. We will refer to this model as *FIFO Information Flow Nets* (FIFN). Petri nets provide a deeply

mathematically founded approach to system modeling, and are capable of grasping the essence of concurrency. Furthermore, they have a straightforward interpretation in terms of processes and channels.

One of our intentions is to show that the FIFO Information Flow Nets are a good candidate as a unifying framework capable of representing several of the above mentioned information flow security properties. To support this claim, we formulate the definitions of Non-Interference and Non-Deducibility using this model. As this model is different from the ones on which the original definitions were based on, we have had to make some fundamental choices in interpreting them within the context of net theory. However, we believe that we have remained completely faithful to the original concepts and that our definitions of Non-Interference between Places (NIP) and Non-Deducibilty on Views (NDV) correspond directly to NI and NDI.

Having considered a single underlying model in which both NIP and NDV can be represented, the next natural step was to consider compositionality. We consider a general composition operation and study the behaviour of NIP and NDV with respect to this operation. In doing this, in fact, we follow Millen's proposal [Mln90] in decomposing the composition operation into two primitive operations, namely *Parallel Composition* and *Feedback*. It will be seen that neither NIP nor NDV are preserved under the feedback operation.

Analysis of non-composability of NIP and NDV in the FIFN context, led to a new definition of information flow, which we will refer to as the *Feedback Non-Deducibility on Views (FNDV)*. We will then show that FNDV is a composable security property. It is interesting to note that this definition shares some characteristics with Non-Deducibility on Strategies.

The paper is organized as follows:

- In Section 2 we define the FIFN model and give definitions of NIP and NDV. We also prove that neither NDV implies NIP nor NIP implies NDV.
- In Section 3 we define the composition operation and show that neither NIP nor NDV are preserved under composition.
- In Section 4 we define FNDV and prove that it is preserved under composition.
- Section 5 summarizes the main results achieved in this paper and considers some implications of this work.
- Appendix A recalls some basic definitions about Petri Nets.

2 FIFO Information Flow Nets (FIFN) Security Model

This section describes the FIFN security model. First, we give the definition of a FIFN; then we define NIP and NDV. Finally, we show that NIP does not imply NDV and give a condition under which this is true. In this subsection we build on the classical definition of Petri net (see appendix A) and FIFO nets [Rou86] to obtain a model which is able to represent the security aspects of a system; note that the new model is still able to handle the relevant aspects of concurrent and distributed systems design.

The approach lies somewhere between *Information Flow Nets (IFN)* proposed by the second author [Var89], [Var90] and *FIFO Nets* [Rou86]. It shares with *IFN* mainly the interpretation of places and transitions and the use of individual tokens manipulated by transitions, while it chooses to look at the contents of places as ordered lists of tokens as in FIFO Nets. Hence we refer to it as *FIFO Information Flow Nets (FIFN)*.

The general scenario that we consider consists of a collection of processes communicating with each other over connecting channels, manipulating the information received and transfering it to another process. The transitions represent processes and places represent channels between them. A place can belong to the input set and output set of more than one transition. This can be used to model different types of communication, e.g., buses, broadcasting. In our case, we interpret the places to be FIFO channels. We feel that this is adequate for modeling the required security properties; furthermore the added benefit is that this results in a simpler model.

We start the formal description of our model by introducing some well-known concepts and fixing appropriate notations for them.

As usual, given a set T, we denote by T^* the free monoid of strings on T. Composition of strings is indicated by juxtaposition. The length of a string $\tau \in T^*$, i.e. the number of characters which it is composed by, is written

as $|\tau|$. The empty string is the (unique) string of length zero and is denoted by ϵ. In the following, since no confusion is possible, we will also use $|\cdot|$ to indicate the cardinality of sets.

Given a finite set P, we define a *numbered subset* of P to be a pair (S, ℓ), where $S \subseteq P$ and ℓ is an injective function from S to the subset of natural numbers $\{1, \ldots, |S|\}$. This results in associating a unique number to each element of S. Therefore, we rarely give a numbered subset by defining both S and ℓ: we rather write $S = \langle p_1, \ldots, p_n \rangle$ with the intended meaning that $S = \{p_1, \ldots, p_n\}$ and $\ell(p_i) = i$. The cardinality of (S, ℓ) is the cardinality of S and will be indicated again as $|(S, \ell)|$. Moreover, we will write $p \in (S, \ell)$ to mean that b is a element of S and $(S, \ell)_i$, $i = 1, \ldots, |S|$, to indicate the (unique) element $p \in S$ such that $\ell(i) = i$. In other words, we treat (S, ℓ) both as a set and as a tuple. The set of numbered subsets of P will be denoted by $\mathcal{T}(P)$.

In order to simplify notation, we sometimes indicate the set of functions $f : P \to T$ as $[P \to T]$. Moreover, the notation for the cartesian product of n copies of a set T will be denoted as usual by T^n.

In the following, we will assume a fixed universe of tokens, i.e. in our setting the universe of messages that can be exchanged between. We will refer to it as $Tokens$.

Definition 2.1 *(FIFN)*
A FIFO Information Flow Net *is a 7-tuple given by* $\mathcal{N} = \langle P, T, I, O, B, Transop, S, \mathcal{M} \rangle$ *where:*

- P *is a finite set of* places*, each of which represents a channel through which information is exchanged between system entities.*
- T *is a finite set of* transitions*, each of which corresponds to a system process.*
- $I : T \to \mathcal{T}(P)$ *is the* input function*.*
- $O : T \to \mathcal{T}(P)$ *is the* output function*.*
- $B : P \to I\!N$ *is the* bound function*, which associates a natural number with each place. The bound function describes the size of the channel buffers.*
- $Transop : T \to \bigcup_{n,m \in I\!N} \Big[Tokens^n \to \wp(Tokens^m) \Big]$ *is a function associating an operation with each transition in* T *in such a way that*

$$Transop(t) : Tokens^{|I(t)|} \to \wp\Big(Tokens^{|O(t)|}\Big),$$

  where $I\!N$ *is the set of natural numbers and* $\wp(\cdot)$ *denotes the power set construction. The function* $Transop(t)$ *is used to describe the manipulation of the input tokens by a transition.*

- $S \in [P_{int} \to Tokens^*]$ *is the* initial state *of the (internal places of the) net, where*

$$P_{inp} = \Big\{ p \in P \;\Big|\; \nexists t \in T \text{ s.t. } p \in O(t) \Big\}$$

  is the set of the input places of the net, and $P_{int} = P \setminus P_{inp}$ *where* $\setminus$ *is the set difference operator.*
  S *is such that* $\forall p \in P_{int},\ |M(p)| \leq B(p)$*.*

- $\mathcal{M}$ *is the set of* initial assignments *for the input places of the net. Each element* $M \in \mathcal{M}$ *is an assignment of admissible sequences of tokens to the input places, i.e. a function* $M \in [P_{inp} \to Tokens^*]$ *such that* $\forall p \in P_{inp},\ |M(p)| \leq B(p)$*.*
  The initial assignment set is used to simulate the inputs a process can receive through a channel.

Moreover, FIFO Information Flow Nets have no isolated places*, i.e.*

$$\forall p \in P\ \exists t \in T \text{ such that } p \in I(t) \cup O(t).$$

□

In addition to the set of input places of a net, we also need to describe the set of output places. These are defined in a similar manner to P_{inp} as

$$P_{out} = \Big\{ p \in P \;\Big|\; \nexists t \in T \text{ s.t. } p \in I(t) \Big\}.$$

Observe that P_{int}, P_{inp} and P_{out}, thanks to the condition in the previous definition, are disjoint. Although such a condition is not strictly necessary, it is quite natural in our interpretation for places.

For the rest of the paper, we establish the convention that while talking about a net $\mathcal{N}$, the symbols P, T, I, O, $Transop$, S and $\mathcal{M}$ will denote the corresponding entities in Definition 2.1, even if not explicitly stated. In case any possible confusion, we will use appropriate and self-explaining subscripts.

Few comments on the above definition are in order. We consider *numbered subsets*—essentially sets—of places to be the input places of a transition instead of the classical *bag*. At a time, a process can remove one input token from each input channel and deposit one output token to each output channel. This modification of the general definition of net (see Definition A.1), is not an heavy modelling restriction as any number of channels is possible between two given processes. We also require a notation to distinguish the order of places in the input and output sets of a transition. This is achieved by having I and O taking values in $\mathcal{T}(P)$.

Generally speaking, the formal structure of processes is still not completely clear. However, for the purpose of this work, it is fair to describe a process by its *input-output* behaviour, i.e. by a function. This is the aim of $Transop$. In order to capture the typical non-deterministic behaviour of processes, the domains of such functions are power sets. Two kinds of channels can be recognized in a system: input and output channels, i.e. channels through which the system exchanges information with the external environment, and internal channels, which are not observable by external users. These are represented using P_{inp}, P_{out} and P_{int}.

Following this intuition about internal and external channels, a FIFN specifies the initial state for its internal places, S, and a set of "admissible" (or expected) behaviours for the external environment, $\mathcal{M}$.

At each time, the state of the net is described by the sequences of tokens located at its places, i.e. by the set of local state of each of its channels.

Definition 2.2 *(Markings)*
A marking *of a FIFN $\mathcal{N}$ is a function $M : P \to Tokens^*$ such that $\forall p \in P,\ |M(p)| \leq B(p)$.* □

Notice therefore that S is not a marking of $\mathcal{N}$ because it does not define the contents of the input places of the net. The set of initial markings associated with a FIFN is obtained by merging an initial assignment and the initial state. The formal definition is as follows.

Let P_0 and P_1 be two disjoint sets. A pair of function $f_0 : P_0 \to X$ and $f_1 : P_1 \to X$, where X is any set, define a "*sum*" function $f_0 \oplus f_1 : P_0 \cup P_1 \to X$ in the obvious following way:

$$(f_0 \oplus f_1)(p) = \begin{cases} f_0(p) & \text{if } p \in P_0 \\ f_1(p) & \text{if } p \in P_1. \end{cases}$$

Definition 2.3 *(Initial Markings)*
Given an FIFN $\mathcal{N}$ the set of initial markings *is the set*

$$\mathcal{MS} = \mathcal{M} \oplus S = \Big\{ M \oplus S \ \Big|\ M \in \mathcal{M} \Big\}.$$

□

Observe that, thanks to the conditions on S and $\mathcal{M}$ in Definition 2.1, the functions in $\mathcal{MS}$ respect the bound function and therefore are markings of $\mathcal{N}$.

The restriction of a marking M to a subset P' of P will be denoted by $M_{P'}$. Moreover, given a set of markings $\mathcal{X}$ we will use $\mathcal{X}/P'$ to denote the set $\{M_{P'} \,|\, M \in \mathcal{X}\}$.

To complete the definition of the model, we need to define the enabling and the firing of a transition and the computations of the net.

Definition 2.4 *(Transition Enabled)*
Given a FIFN $\mathcal{N}$, a transition $t \in T$ is said to be enabled *at a marking M if*

$$\forall p \in P,\ p \in I(t) \Rightarrow |M(p)| \geq 1 \text{ and } p \in O(t) \Rightarrow |M(p)| < B(p).$$

This is denoted by $M[t\rangle$. □

Therefore, a transition may fire only if its firing does not cause the bound function to be exceeded by the actual length of the string of tokens in any of its output places. In other words, a transition may fire only if there is enough space in the FIFO queues to deliver the output.

Definition 2.5 *(Firing Rule)*
A transition t in a FIFN may fire *at a marking M if it is enabled. Firing results in a new marking M'' defined as follows.*

First the transition collects its inputs $(\alpha_1, \ldots, \alpha_n)$, where $n = |I(t)|$ and α_j is from place $I(t)_j$, creating a transitory marking M':

$$M'(p) = \begin{cases} \sigma & \text{if } p = I(t)_j \text{ and } M(p) = \alpha_j\sigma \\ M(p) & \text{if } p \notin I(t) \end{cases} \qquad \text{where } \alpha_j \in Tokens \text{ and } \sigma \in Tokens^*.$$

Let the output, non-deterministically produced from the collected input, be $(\beta_1, \ldots, \beta_m) \in Transop(t)(\alpha_1, \ldots, \alpha_n)$. Then the transition delivers the outputs to the places in $O(t)$, each β_j to the corresponding $O(t)_j$, and the marking M'' is reached:

$$M''(p) = \begin{cases} M'(p)\beta_j & \text{if } p = O(t)_j \\ M'(p) & \text{if } p \notin O(t). \end{cases}$$

The firing of the transition t at M is denoted by $M[t\rangle M''$. □

The classical definitions of reachability and firing sequence are unchanged.

Definition 2.6 *(Firing Sequences)*
A firing sequence *of a FIFN $\mathcal{N}$ is a finite sequence of transition firings $M_0[t_1\rangle M_1[t_2\rangle, \ldots, [t_n\rangle M_n$, also denoted by $M_0[t_1 t_2 \ldots t_n\rangle M_n$, where $M_0 \in \mathcal{MS}$.* □

Definition 2.7 *(Reachability Set)*
Given a FIFN $\mathcal{N}$ and one of its initial markings M_0, the reachability set *for M_0 is defined to be the set*

$$\mathcal{R}(\mathcal{N}, M_0) = \left\{ M' \in [P \to Tokens^*] \;\middle|\; \exists \sigma \in T^* \text{ such that } M_0[\sigma\rangle M' \right\}.$$ □

2.1 Non-Interference between Places

In this subsection we define *Non-Interference between Places* (NIP) which corresponds to the concept of Non-Interference introduced by Goguen and Meseguer in [GM82].

In the context of FIFO Information Flow Nets, we look at the input and the output places of a net as channels through which the system represented by the net will communicate with other systems and users. Note that users may be explicitly represented using nets themselves or implicitly—as we do most of the times—represented using set of initial assignments $\mathcal{M}$. Therefore it is natural to associate the users with the input and output places of the net. Hence we will talk about users linked with a set of input and/or output places.

In the definition of Non-Interference, there is a clear notion of *action performed* by a group of users through which they interfere with another group. In our approach we will associate the users that perform the actions with the input places and the users that receive the effects of those actions with the output places of the system net. Hence the definition of NIP only makes sense between the input and the output places of the net.

Let us start with the definition of *perturbation* of a marking with respect to a set of places.

Definition 2.8 *(Perturbation)*
Let P' be a subset of P, the set of places of a FIFN $\mathcal{N}$, and let M be a marking. Then we define perturbation *of M with respect to P' as*

$$\mathcal{P}_{P'}(M) = \left\{ M' \in [P \to Tokens^*] \;\middle|\; M'_{(P \setminus P')} = M_{(P \setminus P')} \right\}.$$ □

Definition 2.9 *(Non-Interference between Places (NIP))*
Given a FIFN $\mathcal{N}$ and $P_i \subseteq P_{inp}$, $P_o \subseteq P_{out}$, we say that P_i is non-interfering with P_o, $P_i \,|\!:\, P_o$, if and only if

$$\forall M_0 \in \mathcal{MS} \text{ and } \forall M_0' \in \mathcal{P}_{P_i}(M_0) \cap \mathcal{MS} \text{ we have that } \mathcal{R}(\mathcal{N}, M_0)/P_o = \mathcal{R}(\mathcal{N}, M_0')/P_o.$$

The fact that P_i does interfere with P_o will be denoted as $P_i \not|\!:\, P_o$. □

NIP enjoys the following properties:

Proposition 2.10 *(NIP on Groups, part I)*
For all $P_i \subseteq P_{inp}$ and $P_1, P_2 \subseteq P_{out}$, we have that $P_i \,|\!:\, (P_1 \cup P_2) \Rightarrow P_i \,|\!:\, P_1$ and $P_i \,|\!:\, P_2$.

Proof. $P_i \,|\!:\, (P_1 \cup P_2)$ implies that $\forall M_0 \in \mathcal{MS},\ \forall M_0' \in \mathcal{P}_{P_i}(M_0) \cap \mathcal{MS}$,

$$\mathcal{R}(\mathcal{N}, M_0)/(P_1 \cup P_2) = \mathcal{R}(\mathcal{N}, M_0')/(P_1 \cup P_2).$$

This in turn implies that $\mathcal{R}(\mathcal{N}, M_0)/P_1 = \mathcal{R}(\mathcal{N}, M_0')/P_1$ and $\mathcal{R}(\mathcal{N}, M_0)/P_2 = \mathcal{R}(\mathcal{N}, M_0')/P_2$.
Hence $P_i \,|\!:\, (P_1 \cup P_2) \Rightarrow P_i \,|\!:\, P_1$ and $P_i \,|\!:\, P_2$. □

The above result has a clear interpretation. In fact, it can be written as: if $P_i \not|\!:\, P_1$ or $P_i \not|\!:\, P_2$ then $P_i \not|\!:\, (P_1 \cup P_2)$, implying that if a user interferes with another user, so does he with any group containing that user.

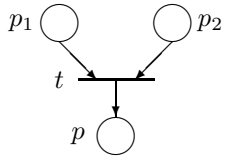
Proposition 2.11 *(NIP on Groups, part II)*
For all $P_1, P_2 \subseteq P_{inp}$ and $P_o \subseteq P_{out}$, we have that $(P_1 \cup P_2) \,|\!:\, P_o \Rightarrow P_1 \,|\!:\, P_o$ and $P_2 \,|\!:\, P_o$.

Proof. For any $M_0 \in \mathcal{MS},\ \forall M_0' \in \mathcal{P}_{P_1}(M_0) \cap \mathcal{MS}$, we have $M_0' \in \mathcal{P}_{(P_1 \cup P_2)}(M_0)$. Using the definition of $(P_1 \cup P_2) \,|\!:\, P_o$, we have $\mathcal{R}(\mathcal{N}, M_0)/P_o = \mathcal{R}(\mathcal{N}, M_0')/P_o$. Hence $P_1 \,|\!:\, P_o$.
Similarly we can show that $P_2 \,|\!:\, P_o$. □

The above result can be written as: if $P_1 \not|\!:\, P_o$ or $P_2 \not|\!:\, P_o$ then $(P_1 \cup P_2) \not|\!:\, P_1$. This implies that if a user does interfere with another user, then any group containing the former user also interferes with latter user. Note that the reverse arrow does not exist either in Proposition 2.10 or in Proposition 2.11. The following examples illustrate these points.

In the examples given below, we define markings as set of pairs $(p, M(p))$. Moreover, input, output and bound functions are not specified. The first two are encoded in the pictures by assuming the convention that places in the input or output set of a transition are drawn from left to right according to the crescent verse of their indices; while the bound function, when not differently specified, must be considered the function wich always yields 1. In the examples, S will always be the empty marking.

Example 2.12 *(NIP on Groups, part III)*
Refer to the figure shown alongside and consider the reverse arrow in Proposition 2.11. Let $\mathcal{MS}$ be $\Big\{\{(p_1, \alpha), (p_2, \alpha_,)(p, \epsilon)\}, \{(p_1, \epsilon), (p_2, \epsilon), (p, \epsilon)\}\Big\}$, with $\alpha \in Tokens$, ϵ the empty string and $Transop(t)$ is the function which produces as output the input taken from p_1 and forgets that from p_2.
We have $p_1 \,|\!:\, p$ and $p_2 \,|\!:\, p$, but $\{p_1, p_2\} \not|\!:\, p$.
This shows that $P_1 \,|\!:\, P_o$ and $P_2 \,|\!:\, P_o \not\Rightarrow (P_1 \cup P_2) \,|\!:\, P_o$.



□

Example 2.13 *(NIP on Groups, part IV)*

Let $\mathcal{N}$ be the net in the figure shown alongside, with
$\mathcal{MS} = \Big\{\{(p,\alpha)\},\{(p,\beta)\}\Big\},\ \alpha,\beta \in Tokens$. *Suppose that*

$Transop(t)(\alpha) = \Big\{\langle\alpha,\beta\rangle,\langle\beta,\alpha\rangle\Big\}$

$Transop(t)(\beta) = \Big\{\langle\alpha,\alpha\rangle,\langle\beta,\beta\rangle\Big\}$ *and*

$Transop(t_1) = identity = Transop(t_2)$.

We have

$$\mathcal{R}\Big(\mathcal{N},\{(p,\alpha)\}\Big)/\{p_1\} \;=\; \Big\{\{(p_1,\alpha)\},\{(p_1,\beta)\}\Big\} \;=\; \mathcal{R}\Big(\mathcal{N},\{(p,\beta)\}\Big)/\{p_1\}$$

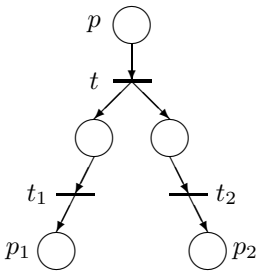
and

$$\mathcal{R}\Big(\mathcal{N},\{(p,\alpha)\}\Big)/\{p_2\} = \Big\{\{(p_2,\alpha)\},\{(p_2,\beta)\}\Big\} = \mathcal{R}\Big(\mathcal{N},\{(p,\beta)\}\Big)/\{p_2\}$$

But

$$\mathcal{R}\Big(\mathcal{N},\{(p,\alpha)\}\Big)/\{p_1,p_2\} = \Big\{\{(p_1,\alpha),(p_2,\beta)\},\{(p_1,\beta),(p_2,\alpha)\}\Big\}\neq$$
$$\Big\{\{(p_1,\alpha),(p_2,\alpha)\},\{(p_1,\beta),(p_2,\beta)\}\Big\} = \mathcal{R}\Big(\mathcal{N},\{(p,\beta)\}\ \Big)\ /\{p_1,p_2\}.$$

Hence $P_i \,|\!:\! P_1$ and $P_i \,|\!:\! P_2 \not\Rightarrow P_i \,|\!:\! (P_1 \cup P_2)$.



□

A comment is in order regarding the result of Example 2.13. At a first glance, the absence of this property could seem somewhat unnatural; however if we think about composition of systems and consider the situation whereby a single user can be linked to the channels represented by $P_1 \cup P_2$, then we can see that this result is a valid one.

On the other hand, the result shown by Example 2.12 is an intuitive one; even if no one in P_1 or in P_2 interferes with P_i on its own, there could still be a subset of users in $P_1 \cup P_2$ who are jointly interfering with users in P_i.

It is hence seen that the problem arises due to possible cooperation between P_1 and P_2. To avoid this situation, we can add a new condition and obtain the following:

Proposition 2.14
If $\forall M_0 \in \mathcal{MS},\ \forall M_0' \in \mathcal{P}_{(P_1\cup P_2)}(M_0) \cap \mathcal{MS}$

$$\exists M' \in (\mathcal{P}_{P_1}(M_0) \cup \mathcal{P}_{P_2}(M_0)) \cap \mathcal{MS}\ \textit{s.t.}\ M \in (\mathcal{P}_{P_1}(M') \cup \mathcal{P}_{P_2}(M')) \tag{1}$$

then $P_i \,|\!:\! P_1$ and $P_i \,|\!:\! P_2 \Rightarrow P_i \,|\!:\! (P_1 \cup P_2)$.

Proof. $\forall M_0 \in \mathcal{MS}, \forall M_0' \in \mathcal{P}_{(P_1\cup P_2)}(M_0) \cap \mathcal{MS}$, let us choose M' as in the condition (1). Without loss of generality, let us suppose that $M' \in \mathcal{P}_{P_1}(M_0)$ and $M \in \mathcal{P}_{P_2}(M')$.
So, $\mathcal{R}(\mathcal{N},M_0)/P_i =$ (since $P_i \,|\!:\! P_1$) $= \mathcal{R}(\mathcal{N},M')/P_i =$ (since $P_i \,|\!:\! P_2$) $= \mathcal{R}(\mathcal{N},M)/P_i$, which is the required result. □

It is worth pointing out that the condition (1) essentially asks for independent behaviours of users linked to P_1 and P_2. This condition seems to be a strong one. (In fact, it is not a *necessary* condition). However it does illustrate what we want to achieve.

2.2 Non-Deducibility on Views

This subsection introduces *Non-Deducibility on Views* (NDV) which corresponds to Sutherland's notion of Non-Deducibility on Inputs [Sut86].

In order to simplify notations, let us establish the following conventions: given a net $\mathcal{N}$,

P_{ext} denotes $P_{inp} \cup P_{out}$, i.e. $P \setminus P_{int}$;
$\forall Q \subseteq P_{ext}$, $\Im(Q)$ denotes $Q \cap P_{inp}$ and $\mathcal{O}(Q)$ denotes $Q \cap P_{out}$;
$\forall \{p_1,\ldots,p_n\} = Q \subseteq P$, $M(Q)$ denotes $\{M(p_1),\ldots,M(p_n)\}$.

Observe that the last notational convention impies that $M(\emptyset) = \emptyset$. Recall that, given a marking M and a subset of places Q, M_Q denotes the restriction of M to Q.

In order to model Sutherland's notion in the FIFO Information Flow Nets framework, we first need to give a definition of a view.

Definition 2.15 *(Views)*
Given a FIFN $\mathcal{N}$, let $Q \subseteq P_{ext}$. Then for each $M_0 \in \mathcal{MS}$ define $\mathcal{V}iew_{M_0}(Q)$ to be the set

$$\left\{(v^i, v^o) \;\middle|\; v^i \in [\Im(Q) \to Tokens^*],\ v^o \in [\mathcal{O}(Q) \to Tokens^*],\ \exists M \in \mathcal{R}(\mathcal{N}, M_0) \text{ s.t. } M_{0\Im(Q)} = v^i,\ M_{\mathcal{O}(Q)} = v^o\right\}$$

The set of views of Q *is the union of the views on the set of initial markings, i.e.*

$$\mathcal{V}iew(Q) = \bigcup_{M_0 \in \mathcal{MS}} \mathcal{V}iew_{M_0}(Q).$$

□

Hence essentially a view is a pair of *partial* markings. The above definition basically means that a view of a set of users is, as expected, all that they can see *contemporarily*, i.e., in the same computation. Two views of two different sets of users are *compatible* if they can be observed in the same computation. The formal definition is given below.

Definition 2.16 *(Compatibility)*
Given a FIFN $\mathcal{N}$, let P_1 and P_2 be subsets of P. Views $(v^i, v^o) \in \mathcal{V}iew(P_1)$ and $(w^i, w^o) \in \mathcal{V}iew(P_2)$ are compatible*, in symbols $(v^i, v^o) \uparrow (w^i, w^o)$, if and only if $\exists M_0 \in \mathcal{MS}$ and $M \in \mathcal{R}(\mathcal{N}, M_0)$ such that*

$$M_{0\Im(P_1)} = v^i,\ M_{0\Im(P_2)} = w^i,\ M_{\mathcal{O}(P_1)} = v^o \text{ and } M_{\mathcal{O}(P_2)} = w^o.$$

□

Recall that in the case of Non-Interference, there exists the notion of *action* performed on the system to interfere with someone. In the case of Non-Deducibility, there is the more general notion of *deduction* perfomed by a user. In the context of FIFN, the latter results in a broader definition for NDV than for NIP; in fact, the NDV definition is valid for every pair of subset of input and output places.

Definition 2.17 *(Non-Deducibility on Views)*
Let $P_1, P_2 \subseteq P_{ext}$. We say that P_1 and P_2 are non-deducible *(with respect to each other), $P_1 \not\longleftrightarrow P_2$, if and only if*

$$\forall (v^i, v^o) \in \mathcal{V}iew(P_1) \text{ and } \forall (w^i, w^o) \in \mathcal{V}iew(P_2) \text{ we have that } (v^i, v^o) \uparrow (w^i, w^o).$$

□

In other words, the above definition states that there is no flow of information between two groups of users (linked to sets of channels P_1 and P_2 respectively) if each pair of views is *compatible*, i.e., each pair of views can be observed in the same computation. This means that whatever the users in one group can see, they cannot deduce anything about what the users in the other group observe.

It is worth pointing out that the relation $\cdot \not\longleftrightarrow \cdot$ is symmetric, i.e., $P_1 \not\longleftrightarrow P_2 \Rightarrow P_2 \not\longleftrightarrow P_1$.

NDV has the following property:

Proposition 2.18 *(NDV on Groups)*
$(P_1 \cup P_2) \not\longleftrightarrow P_3 \Rightarrow P_1 \not\longleftrightarrow P_3$ *and* $P_2 \not\longleftrightarrow P_3$

Proof. For all $(v_1^i, v_1^o) \in \mathcal{V}iew(P_1)$, there exists $(v_2^i, v_2^o) \in \mathcal{V}iew(P_1 \cup P_2)$ s.t. $v^i_{2\Im(P_1)} = v_1^i$ and $v^o_{2\mathcal{O}(P_1)} = v_1^o$. Then, by hypothesis, for all $(w^i, w^o) \in \mathcal{V}iew(P_3)$, we have that $(v_2^i, v_2^o) \uparrow (w^i, w^o)$ which implies that for all $(w^i, w^o) \in \mathcal{V}iew(P_3)$, $(v_1^i, v_1^o) \uparrow (w^i, w^o)$. This shows that $P_1 \not\longleftrightarrow P_3$.
Similarly one can show that $P_2 \not\longleftrightarrow P_3$. □

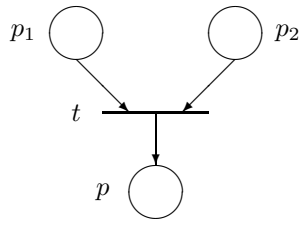
This is quite a natural result; it can also be written in the following ways:

(i) $(P_1 \longleftrightarrow P_3$ or $P_2 \longleftrightarrow P_3) \Rightarrow (P_1 \cup P_2) \longleftrightarrow P_3$;
(ii) $(P_3 \longleftrightarrow P_1$ or $P_3 \longleftrightarrow P_2) \Rightarrow P_3 \longleftrightarrow (P_1 \cup P_2)$,

where $\cdot \longleftrightarrow \cdot$ stands for the negation of $\cdot \not\longleftrightarrow \cdot$.

Once again the reverse arrow in the Proposition 2.18 does not hold. This is illustrated by the following example.

Example 2.19
Let $\mathcal{N}$ be the net in the figure shown alongside, with
$\mathcal{MS} = \Big\{\{(p_1,\alpha),(p_2,\alpha),(p,\epsilon)\},\{(p_1,\alpha),(p_2,\beta),(p,\epsilon)\},$
$\{(p_1,\beta),(p_2,\alpha),(p,\epsilon)\},\{(p_1,\beta),(p_2,\beta),(p,\epsilon)\}\Big\}$ $\alpha,\beta \in Tokens.$
Suppose that
$Transop(t)(\alpha,\beta) = \{\alpha\},\ Transop(t)(\beta,\alpha) = \{\alpha\}$
$Transop(t)(\alpha,\alpha) = \{\beta\},\ Transop(t)(\beta,\beta) = \{\beta\}.$
We have $\{p_1\} \not\longleftrightarrow \{p\}$ *and* $\{p_2\} \not\longleftrightarrow \{p\}$ *but* $\{p_1,p_2\} \longleftrightarrow \{p\}$.
This shows that $P_1 \not\longleftrightarrow P_3$ *and* $P_2 \not\longleftrightarrow P_3 \not\Rightarrow (P_1 \cup P_2) \not\longleftrightarrow P_3$.



□

The result in Example 2.19 means that even if it is possible to deduce something about P_1 and P_2 as a pair, it can still be the case you may not be able to deduce anything about either P_1 or P_2.

2.3 Relation between NIP and NDV

Having defined both Non-Interference and Non-Deducibility using a single underlying model, let us now consider the relation between them.

We start by showing that NDV does not imply NIP. Consider the FIFN whose structure is shown in the figure of Example 2.19 above. Moreover, suppose that

$$\begin{array}{ll} (i) & \mathcal{MS} = \Big[\{p_1,p_2\} \rightarrow \{\alpha,\beta\}\Big],\ \alpha,\beta \in Tokens; \\ (ii) & Transop(t)(\alpha,\alpha) = \beta; \\ & Transop(t)(\alpha,\beta) = \alpha; \\ & Transop(t)(\beta,\alpha) = \beta; \\ & Transop(t)(\beta,\beta) = \alpha. \end{array}$$

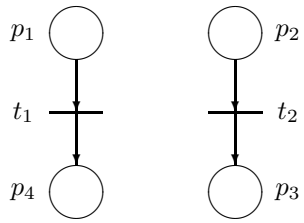
Let M_0 be $\{(p_1,\alpha),(p_2,\alpha)\} \in \mathcal{MS}$, and M_0' be $\{(p_1,\beta),(p_2,\alpha)\} \in \mathcal{P}_{\{p_1\}}(M_0) \cap \mathcal{MS}$.

Then we have $\mathcal{R}\Big(\mathcal{N},M_0\Big)/\{p\} = \Big\{\{(p,\alpha)\}\Big\} \neq \Big\{\{(p,\beta)\}\Big\} = \mathcal{R}\Big(\mathcal{N},M_0'\Big)/\{p\}$. Therefore, $\{p_1\} \not|\!:\{p\}$.

On the other hand, it is easy to see that $\{p_1\} \not\longleftrightarrow \{p\}$, since $\mathcal{V}iew(\{p_1,p\}) = \Big[\{p_1,p\} \rightarrow \{alpha,\beta\}\Big]$.

Now, consider the following example.

Example 2.20
Consider the net in the figure shown alongside, where
$\mathcal{MS} = \Big\{\{(p_1,\alpha),(p_2,\beta),(p_3,\epsilon),(p_4,\epsilon)\},\{(p_1,\beta),(p_2,\gamma),(p_3,\epsilon),(p_4,\epsilon)\}\Big\}$
with $\alpha,\beta,\gamma \in Tokens.$
Suppose that
$Transop(t_1)(\alpha) = \{\alpha'\},\ Transop(t_1)(\beta) = \{\beta'\}$
$Transop(t_2)(\gamma) = \{\beta'\},\ Transop(t_2)(\beta) = \{\alpha'\}$, *with* $\alpha',\beta' \in Tokens.$
We have $\{p_1\}\,|\!:\{p_3\}$ *but* $\{p_1\} \longleftrightarrow \{p_3\}$.
This shows that $P_i\,|\!:P_o \not\Rightarrow P_i \not\longleftrightarrow P_o$.



□

The above example shows that NIP does not imply NDV. This is an interesting result, contrary to our original expectation.[1] It depends on the choice to model the inputs of the system by means of $\mathcal{MS}$. We believe that such a choice is a fair one in the case of real systems; in fact, in general, not all combinations of a single channel input streams will turn out to be admissible system input streams. $\mathcal{MS}$ enables us to distinguish this aspect. Example 2.20 stresses this point.

In reality, a user could use two channels to interfere with another user whereas he may not be interfering when considering each channel on its own (see Example 2.12). Non-Interference does not capture such a behaviour whereas the finer Non-Deducibility can.

[1] Previous result [Sut86] stating that the two definitions coincide when they are defined is due to the choice of the formalism used.

If we add a condition ensuring that there is independence between the input streams observed in the channels, then we get the following:

Proposition 2.21 *(NIP ⇒ NDV)*
Let $P_i \subseteq P_{inp}$ and $P_o \subseteq P_{out}$. If $\forall M_0,\ M'_0 \in \mathcal{MS}$,

$$\exists M''_0 \in \mathcal{P}_{P_i}(M_0) \cap \mathcal{MS} \text{ s.t. } M'_{0P_i} = M''_{0P_i} \tag{2}$$

then $P_i \mid\!: P_o \Rightarrow P_i \not\leftrightarrow P_o$.

Proof. If $P_i = \emptyset$ or $P_o = \emptyset$, there is nothing to prove. Let[2] $(v^i, \emptyset) \in \mathcal{V}iew(P_i)$ and $(\emptyset, w^o) \in \mathcal{V}iew(P_o)$. Using the definition of $\mathcal{V}iew(\cdot)$, the former implies that there exists $M'_0 \in \mathcal{MS}$ such that $M'_{0P_i} = v^i$, while the latter implies that there exists $M''_0 \in \mathcal{MS}$ and $M'' \in \mathcal{R}(\mathcal{N}, M''_0)$ such that $M''_{P_o} = w^o$.
Let us consider $M_0 \in \mathcal{P}_{P_i}(M''_0)$ such that $M''_{0P_o} = v^i$, i.e., $(v^i, \emptyset) \in \mathcal{V}iew_{M_0}(P_i)$.
By condition (2), $M_0 \in \mathcal{MS}$ and using NIP definition, we have $\mathcal{R}(\mathcal{N}, M''_0)/P_o = \mathcal{R}(\mathcal{N}, M_0)/P_o$. Therefore $(v^i, w^i) \in \mathcal{V}iew_{M_0}(P_i \cup P_o)$. This concludes the proof. □

Note that the condition (2) requires Non-Deducibility between P_i and $P_{inp} \setminus P_i$, that is, $P_i \not\leftrightarrow (P_{inp} \setminus P_i)$. We have written it in such a way to reflect the fact that it only depends on $\mathcal{MS}$. Once again, condition (2) is not a necessary condition, but it is sufficient to illustrate the point we want to make.

3 System Composition and Security

In this section, we first define the composition operation for FIFNs and then study the effect of this operation on the security definitions given above in Section 2.

We consider a general form of the composition operation. The basic idea behind composition is that systems are composed by sharing channels. In the net context, this means that nets are composed by sharing places—input places of one net with the output places of another. This approach is inspired by the way in which processes are composed in the process algebra approach to the semantics of concurrency

We use Millen's proposal in [Mln90] and consider the composition operation to consist of two primitive operations—*parallel* and *feedback*. We will first consider the parallel operation and then look at the feedback.

3.1 Parallel Operation

In the case of the *parallel* operation, the two nets are just juxtaposed to make them run in parallel, whereas the feedback operation merges the input and the output places of a net.

Definition 3.1 *(Parallel Operation)*
Let $\mathcal{N}_1 = \langle P_1, T_1, I_1, O_1, B_1, Transop_1, S_1, \mathcal{M}_1\rangle$ and $\mathcal{N}_2 = \langle P_2, T_2, I_2, O_2, B_2, Transop_2, S_2, \mathcal{M}_2\rangle$ be two FIFNs such that—up to an appropriate renaming—the respective sets of places and transitions are disjoint. The parallel composition *of $\mathcal{N}_1$ and $\mathcal{N}_2$ is a net $\mathcal{N}$, also denoted as $\mathcal{N}_1 \oplus \mathcal{N}_2$, defined as follows:*

$$\mathcal{N} = \langle P_1 \cup P_2, T_1 \cup T_2, I_1 \oplus I_2, O_1 \oplus O_2, B_1 \oplus B_2, Transop_1 \oplus Transop_2, S_1 \oplus S_2, \mathcal{M}\rangle,$$

where $\mathcal{M} = \mathcal{M}_1 \oplus \mathcal{M}_2 = \{M_1 \oplus M_2 \mid M_1 \in \mathcal{M}_1 \text{ and } M_2 \in \mathcal{M}_2\}$. □

Let us now consider the parallel composition operation with respect to NIP and NDV. The two nets which are to be composed will be denoted using $\mathcal{N}_1$ and $\mathcal{N}_2$, defined as follows: $\mathcal{N}_1 = \langle P_1, T_1, \ldots, \mathcal{M}_1\rangle$ and $\mathcal{N}_2 = \langle P_2, T_2, \ldots, \mathcal{M}_2\rangle$. The composed net is denoted by $\mathcal{N} = \langle P, T, \ldots, \mathcal{M}\rangle$. Moreover, NIP and NDV on the composing nets are denoted as $\mid\!:_1$, $\mid\!:_2$, $\not\leftrightarrow_1$ and $\not\leftrightarrow_2$ respectively. Clearly the parallel composition is a secure operation; it preserves all the relations existing in the two composing nets and creates new relations in the composed net. These properties are given as follows:

[2] Here $\emptyset$ is the unique function with empty domain, i.e. the function with the empty graph.

Proposition 3.2 *(Parallel Composition and NIP, part I)*
For any $j = 1, 2$, $I \subseteq (P_j)_{inp}$ and $O \subseteq (P_j)_{out}$, we have $I \mid:_j O \Rightarrow I \mid: O$. □

Proposition 3.3 *(Parallel Composition and NIP, part II)*
For any $j = 1, 2$, $I \subseteq (P_j)_{inp}$ and $O \subseteq (P_{3-j})_{out}$, we have $I \mid: O$. □

Proposition 3.4 *(Parallel Composition and NDV, part I)*
For any $j = 1, 2$ and $Q_1, Q_2 \subseteq (P_j)_{ext}$, we have $Q_1 \not\leftrightarrow_j Q_2 \Rightarrow Q_1 \not\leftrightarrow Q_2$. □

Proposition 3.5 *(Parallel Composition and NDV, part II)*
For any $j = 1, 2$, $Q_1 \subseteq (P_j)_{ext}$ and $Q_2 \subseteq P_{3-j}$, we have $Q_1 \not\leftrightarrow Q_2$. □

3.2 Feedback Operation

Let us now consider the *feedback* operation. In general, feedback is not a secure operation. Some earlier work has demonstrated this. Within the net framework, we believe that the problems caused by the feedback operation can be better understood and treated.

The feedback operation in a FIFN is defined by giving

(i) a set of input places and a set of output places which are to be merged pairwise,
(ii) a function which specifies the correspondence between the input and the output places and
(iii) an initial assignment which will become part of the state of the net with feedback.

The new set of initial assignments will be part of the old set, compatible with the assignment chosen to build the state of the new net. Moreover, there must be compatibility between the input and the output places which have been chosen to be merged, i.e., the operation associated with each transition must remain defined for all possible markings of the new net. This is ensured by the definition of *feedbackable* given below. Before describing this, we need to give another definition which will help us to simplify the notation.

Definition 3.6
Let X, Y and Z be sets and $f : Y \to X$ be a one-to-one function. Let us define a concatenation *operation with respect to f ($;_f$) of functions from X to Z^* and from Y to Z^* as follows:*

$$\forall M' : Y \to Z^*,\ \forall M : X \to Z^*,\ M;_f M' : X \to Z^* \text{ and}$$

$$\forall x \in X,\ (M;_f M')(x) = M(p)M'(f^{-1}(x)),$$

where the juxtaposition of values of M and M' is the concatenation of strings. □

Definition 3.7 *(Feedbackable)*
Let $\mathcal{N}$ be a FIFN $P_i \subseteq P_{inp}$ $P_o \subseteq P_{out}$ and $f : P_o \to P_i$ be a one-to-one function. Given $M^s \in \mathcal{M}$, we say that $\mathcal{N}$ is feedbackable *via $\langle f, M^s \rangle$ if and only if*

$$\forall M_0 \in \mathcal{M} \text{ s.t. } M_{0P_i} = M^s_{P_i},\ \forall M \in \mathcal{R}(\mathcal{N}, M_0 \oplus S)/P_o,\ M_{0(P_{inp} \setminus P_i)} \oplus (M^s_{P_i};_f M) \in \mathcal{M}.$$ □

So, as anticipated, the condition under which a net is feedbackable is just a condition of soundness: feeding the net with strings of tokens generated from the net itself in its output places, must lead to admissible markings, including for the bounds on the size of the strings.

Note that in Definitions 3.7, we have $|P_i| = |P_o|$.

Now, we are ready to give the definition of the feedback operation.

It consists of merging together the input and the output places as specified by f. The merged places become internal places not anymore visible from outside the system and therefore not anymore usable for feedback operations. Observe that even if f is a one-to-one function, we can still model compositions in which more places are merged in one place (or viceversa) by simply adding a transition which collects the inputs and delivers them (as a string) to the designed place.

Definition 3.8 *(Feedback)*
Given $\mathcal{N} = \langle P, T, I, O, B, Transop, S, \mathcal{M}\rangle$, $P_i \subseteq P_{inp}$, $P_o \subseteq P_{out}$, $f : P_o \to P_i$ and $M^s \in \mathcal{M}$ such that $\mathcal{N}$ is feedbackable via $\langle f, M^s\rangle$, feedback of $\mathcal{N}$ via $\langle f, M^s\rangle$ is given by the net

$$\overline{\mathcal{N}} = \langle P \setminus P_o, T, I, \overline{O}, B_{(P\setminus P_o)}, Transop, \overline{S}, \overline{\mathcal{M}}\rangle,$$

where:

$$\begin{array}{rcl} \overline{O}(t) & = & O(t)[f(p)/p\ \forall p \in P_o], \text{ that is, the numbered set } O(t) \text{ modified by replacing all} \\ & & \text{the places in } P_o \text{ with the corresponding places in } P_i; \\ \overline{S} & = & S \oplus M^s_{P_i}; \\ \overline{\mathcal{M}} & = & \left\{ \overline{M}_{(P_{inp}\setminus P_i)} \;\middle|\; \overline{M} \in \mathcal{M} \text{ and } \overline{M}_{P_i} = M^s_{P_i} \right\}. \end{array}$$

□

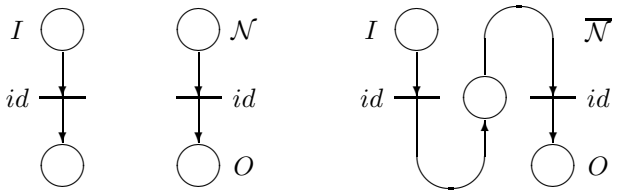
In case of any possible confusion, NIP and NDV on a feedbacked net will be distinguished from the respective relations on the original net by an appropriate subscript.

3.2.1 Feedback and NIP

Let us now consider the behaviour of NIP under the feedback operation. The first point to note is that NIP is not preserved by feedback. In the rest of the paper, we will use P_i and P_o to denote the input and output places respectively, which are connected via the feedback operation.

Proposition 3.9 *(Feedback and NIP, part I)*
Let $I \subseteq P_{inp} \setminus P_i$ and $O \subseteq P_{out} \setminus P_o$.
Then $I \,|\!:_{\mathcal{N}} O \not\Rightarrow I \,|\!:_{\overline{\mathcal{N}}} O$

Proof. Consider the figure shown alongside where $\overline{\mathcal{N}}$ is the net obtained from $\mathcal{N}$ with a feedback operation, I and O are the singletons indicated in figure and the operation corresponding to both transitions of $\mathcal{N}$ is the identity. We have $I \,|\!:_{\mathcal{N}} O$ but $I \not|\!:_{\overline{\mathcal{N}}} O$. This shows that NIP is not preserved by feedback.



□

In the example above the problem is as follows: with feedback, we can create a link between a user connected to I and a trojan horse hidden in the system in such a way that the user can now interfere with the user connected to O. In order to forbid this, we need to ensure that no links capable of making I interfere with O due to feedback are allowed. If we do this, we get the following result.

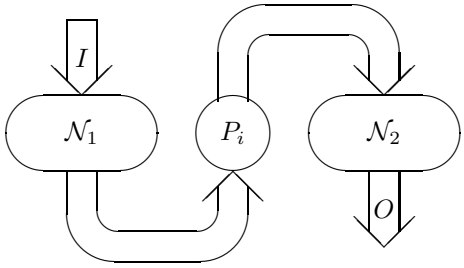
Proposition 3.10 *(Feedback and NIP, part II)*
Let $I \subseteq P_{inp} \setminus P_i$ and $O \subseteq P_{out} \setminus P_o$. Then $(I \cup P_i) \,|\!:_{\mathcal{N}} O \Rightarrow I \,|\!:_{\overline{\mathcal{N}}} O$.

Proof. Let $\overline{M}_0 \in \overline{\mathcal{MS}}$ and $\overline{M}'_0 \in \mathcal{P}_I(\overline{M}_0) \cap \overline{\mathcal{MS}}$.
We will show that $\forall \overline{M} \in \mathcal{R}(\mathcal{N}, \overline{M}_0)$, $\exists \overline{M}' \in \mathcal{R}(\mathcal{N}, \overline{M}'_0)$ such that $\overline{M}(O) = \overline{M}'(O)$. This shows that $\mathcal{R}(\overline{\mathcal{N}}, \overline{M}_0) \subseteq \mathcal{R}(\overline{\mathcal{N}}, \overline{M}'_0)$. The other side of the inclusion can be shown in the same manner.
Taking such an $\overline{M}$, let us consider the firing sequence $\overline{M}_0[\overline{\sigma}\rangle\overline{M}$ which generates it.
Let $M_0 = (\overline{M}_{0(P\setminus P_i)} \oplus M^s_{P_i};_f \delta)$ where $\delta : P_o \to Tokens^*$ represents the sequences of tokens generated in P_i and used by some transition in $\overline{\sigma}$. Then, in the net $\mathcal{N}$, there exists a computation $M_0[\sigma\rangle M$ with $M(O) = \overline{M}(O)$. Now let $M'_0 \in \mathcal{P}_{(I\cup P_i)}(M_0) \cap \mathcal{MS}$ such that $M'_{0P_i} = M^s_{P_i}$, $M'_{0I} = \overline{M}'_{0I}$. By hypothesis, there exists a $M'_0[\sigma'\rangle M'$ such that $M'(O) = M(O) = \overline{M}(O)$. Therefore, in $\overline{\mathcal{N}}$, there exists $\overline{M}'_0[\overline{\sigma}'\rangle\overline{M}'$ such that $\overline{M}'(O) = M'(O) = M(O) = \overline{M}(O)$.
This concludes the proof. □

There is another interesting case in the behaviour of NIP under feedback: this is the sequential composition case.

Proposition 3.11 *(Feedback and NIP, part III)*

Given two FIFNs, $\mathcal{N}_1$ and $\mathcal{N}_2$, let us suppose that there exists $P_o \subseteq (P_1)_{out}$, $P_i \subseteq (P_2)_{inp}$, $f : P_o \to P_i$ and $M^s \in (\mathcal{M}_1 \oplus \mathcal{M}_2)$ such that $\mathcal{N}_1 \oplus \mathcal{N}_2$ is feedbackable via $\langle f, M^s \rangle$ (see figure alongside).
Let $I \subseteq (P_1)_{inp}$ and $O \subseteq (P_2)_{out}$.
Then $(I \,|\!:_{\mathcal{N}_1} P_o$ or $P_i \,|\!:_{\mathcal{N}_2} O) \Rightarrow I \,|\!:_{\mathcal{N}} O$.



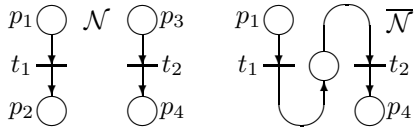
Proof. Standard. □

As we expect, in the case of sequential composition, if all the possible covert channels between two users are removed in just one of the two systems being composed, we obtain a secure system.

The final point that we would like to emphasize about feedback and NIP is that the reverse arrow in Proposition 3.10 does not exist. This is stated by the next proposition.

Proposition 3.12 *(Feedback and NIP, part IV)*
$I \,|\!:_{\overline{\mathcal{N}}} O \not\Rightarrow (I \cup P_i) \,|\!:_{\mathcal{N}} O$.

Proof. Consider $\mathcal{N}$ and $\overline{\mathcal{N}}$ in the figure shown beside, where $\mathcal{M} = [\{p_1, p_3\} \to \{\alpha, \beta, \gamma\}]$, $\alpha, \beta, \gamma \in Tokens$, and $Transop(t_1)(\alpha) = \{\alpha\}$, $Transop(t_1)(\beta) = \{\alpha\}$, $Transop(t_1)(\gamma) = \{\beta\}$ and $Transop(t_2) = Transop(t_1)$.
We then have
$\{p_1\} \not|\!:_{\mathcal{N}} \{p_2\}$, $\{p_3\} \not|\!:_{\mathcal{N}} \{p_4\}$ but $\{p_1\} \,|\!:_{\overline{\mathcal{N}}} \{p_4\}$.



□

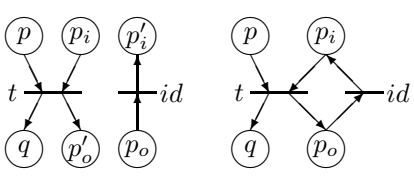
The above proposition shows that it is possible to obtain a secure system by composing two non-secure systems!

3.2.2 Feedback and NDV

We now study the behaviour of NDV under the feedback operation. We will see that there is a strong similarity between the two properties (NDV and NIP) with respect to feedback. Let us begin by showing that NDV is not preserved by feedback.

Proposition 3.13 *(Feedback and NDV, part I)*
Let $Q_1, Q_2 \subseteq P_{ext} \setminus (P_i \cup P_o)$. Then $Q_1 \not\leftrightarrow_{\mathcal{N}} Q_2 \not\Rightarrow Q_1 \not\leftrightarrow_{\overline{\mathcal{N}}} Q_2$.

Proof. Consider the nets in the figure shown alongside, where $\mathcal{M}$ is the set of initial assignments M such that $M(p) \in \{\alpha, \beta\}^2$ and $M(p_i), M(p_o) \in \{\epsilon, \alpha, \beta\} \cup \{\alpha, \beta\}^2$, with $\alpha, \beta \in Tokens$ and where A^2 denotes the set of strings on A of length two. Suppose that $Transop(t)$ is the function which delivers in q the token from p_i and delivers in p'_o the token from p. We have $\{p\} \not\leftrightarrow_{\mathcal{N}} \{q\}$.
Include feedback by choosing $M^s = \left\{(p_i, \alpha), (p_o, \epsilon)\right\}$ and $f : \{p'_i, p'_o\} \to \{p_i, p_o\}$ such that $f(p'_i) = p_i$ and $f(p'_o) = p_o$.
We have $\overline{\mathcal{M}} = \left\{\{p\} \times \{\alpha, \beta\}^2\right\}$ and $\{p\} \longleftrightarrow_{\overline{\mathcal{N}}} \{q\}$.



□

The problem is the same as the one illustrated in the previous section on NIP. Solution is once again the same. This proposition for NDV corresponds to Proposition 3.9 for NIP.

Proposition 3.14 *(Feedback and NDV II)*
For any $Q_1, Q_2 \subseteq P_{ext} \setminus (P_i \cup P_o)$*, we have* $(Q_1 \cup P_i) \not\leftrightarrow_{\mathcal{N}} Q_2 \Rightarrow Q_1 \not\leftrightarrow_{\overline{\mathcal{N}}} Q_2$.

Proof. $\forall (v^i, v^o) \in \mathcal{V}iew_{\overline{\mathcal{N}}}(Q_1)$, $\forall (w^i, w^o) \in \mathcal{V}iew_{\overline{\mathcal{N}}}(Q_2)$, we have a computation $M_0''[\sigma''\rangle M''$ such that $M''_{0\Im(Q_2)} = w^i$ and $M''_{\mathcal{O}(Q_1)} = w^o$. As $(Q_1 \cup P_i) \not\leftrightarrow_{\mathcal{N}} Q_2$, we have a computation $M_0'[\sigma'\rangle M'$ such that $M'_{0\Im(Q_1)} = v^i$, $M'_{0\Im(Q_2)} = w^i$, $M'_{0P_i} = M^s_{P_i}$, $M'_{\mathcal{O}(Q_1)} = v^o$ and $M'_{\mathcal{O}(Q_2)} = w^o$. Hence there exists a computation $M_0[\sigma\rangle M$ in $\overline{\mathcal{N}}$ such that $M_{0\Im(Q_1)} = v^i$, $M_{0\Im(Q_2)} = w^i$, $M'_{\mathcal{O}(Q_1)} = v^o$ and $M'Q_2^{out} = w^o$. This concludes the proof. □

Once again the reverse arrow in Proposition 3.14 is false (cf. Proposition 3.12 for NIP).

Finally, there is no problem with sequential composition.

Proposition 3.15 *(Feedback and NDV III)*
Given FIFNs $\mathcal{N}_1$ *and* $\mathcal{N}_2$*, let us suppose that there exists* $P_o \subseteq (P_1)_{out}$*,* $P_i \subseteq (P_2)_{inp}$*,* $f : P_o \to P_i$ *and* $M^s \in (\mathcal{M}_1 \oplus \mathcal{M}_2)$ *such that* $\mathcal{N}_1 \oplus \mathcal{N}_2$ *is feedbackable via* $\langle f, M^s\rangle$ *(see figure in Proposition 3.11).*
Let $Q_1 \subseteq (P_1)_{ext} \setminus P_o$ *and* $Q_2 \subseteq (P_2)_{ext} \setminus P_i$*. Then* $(Q_1 \not\leftrightarrow_{\mathcal{N}_1} P_o$ *or* $P_i \not\leftrightarrow_{\mathcal{N}_2} Q_2) \Rightarrow Q_1 \not\leftrightarrow_{\mathcal{N}} Q_2$.

Proof. Standard. □

Proposition 3.16 *(Feedback and NDV, part IV)*
$Q_1 \not\leftrightarrow_{\mathcal{N}} Q_2 \not\Rightarrow (Q_1 \cup P_i) \not\leftrightarrow_{\overline{\mathcal{N}}} Q_2$.

Proof. Easy to show using an example similar to the one considered in Proposition 3.12. □

4 Feedback Non-Deducibility on Views - A Composable Property

The need for the having a realistic composable security property is well-known. In this section, we introduce a new definition of information flow which has the useful property of being preserved by both parallel and feedback operations. That is, this information flow security property is *composable*.

Reflecting back on the results of the previous section, the reader can notice a strange asymmetry in the roles played by the input and output places involved in the feedback operation. "Control" of input places—by which we mean having no information flow between them and the places under observation—gives us the necessary power to maintain security of a system with feedback, whereas the "control" of output places does not (see Propositions 3.10 and 3.14).

Two views that are compatible in the original net can be incompatible in the feedbacked net. This is due to the fact that the net is not able to produce in the feedbacked output places the tokens it needs in the feedbacked input places. Hence it is not able to perform any of the computations which would make the views compatible. Now, by controlling the input places involved in the feedback, we can ensure that each of those computations has a corresponding one in the feedbacked net—having the same behaviour with respect to the places under observation—by simply imposing that the computation makes no use of inputs taken from the feedback mechanism.

The above observation is the starting point for defining a new security property referred to as *Feedback Non-Deducibility on Views (FNDV)*.

Before defining FNDV, we first need to define the concepts of *Feedback Simulating Net* and *Output Guided Computation*. We will not only require that each pair of views of two groups of places under observation be compatible, but also need it to be compatible with at least one system Output Guided Computation (OGC), for each Feedback Simulating Net (FSN).

Definition 4.1 *(Feedback Simulating Nets (FSN))*
Let $\mathcal{N}$ *be a FIFN with* $I \subseteq P_{inp}$ *and* $O \subseteq P_{out}$*. A* Feedback Simulating Net *on* (I, O) *is defined as a 5-tuple:*

$$\langle \pi, \pi_i, \pi_o, M^\pi, M^s\rangle,$$

where $M^s \in \mathcal{M}$*,* $\pi = \langle P^\pi, T^\pi, I^\pi, O^\pi, B^\pi, Transop^\pi, S^\pi, \mathcal{M}^\pi\rangle$ *is a FIFN,* $M^\pi \in \mathcal{M}^\pi$*,* $\pi_i : P^\pi_o \to I$ *and* $\pi_o : O \to P^\pi_i$ *are one-to-one functions, for* $P^\pi_i \subseteq P^\pi_{inp}$ *and* $P^\pi_o \subseteq P^\pi_{out}$*, and and* $\mathcal{N} \oplus \pi$ *is feedbackable via* $\langle \pi_i \oplus \pi_o, M^s \oplus M^\pi\rangle$. □

Therefore, a Feedback Simulating Net models the possible behaviours of a user which is aware of the system history, of the system output in O and feeds it with tokens in I. Sometimes we will use just π to refer to a Feedback Simulating Net. The choice to use π to denote a FSN is a deliberate one. The intention is to later show some similarity with the Non-Deducibility of Strategies definition.

Now, an Output Guided Computation is a sequence of computations of the net which is guided by a Feedback Simulating Net in the precise sense that it is the latter which decides what inputs in I the net will see at each step.

Definition 4.2 *(Output Guided Computations (OGC))*
Given a FIFN $\mathcal{N}$, $I \subseteq P_{inp}$, $O \subseteq P_{out}$ and given an FSN on (I,O), $\langle \pi, \pi_i, \pi_o, M^\pi \rangle$, an Output Guided Computation *on π is a sequence of firing sequences of $\mathcal{N}$,*

$$M_0^0[\sigma_0\rangle M^0, M_0^1[\sigma_1\rangle M^1, \ldots, M_0^n[\sigma_n\rangle M^n$$

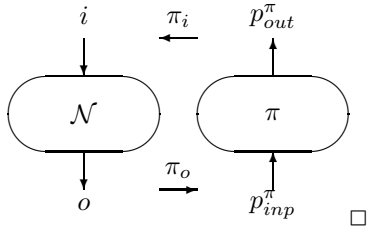
for which there exists a sequence of firing sequences of π, $\overline{M}_0^0[\overline{\sigma}_0\rangle \overline{M}^0, \overline{M}_0^1[\overline{\sigma}_1\rangle \overline{M}^1, \ldots, \overline{M}_0^n[\overline{\sigma}_n\rangle \overline{M}^n$ such that $M_0^0 \in \mathcal{MS}$ and $M_{0I}^0 = M_I^s$, $\overline{M}_0^0 \in \mathcal{MS}^\pi$ and $\overline{M}_{0P_i^\pi}^0 = M_{P_i^\pi}^\pi$ and for $i = 0, \ldots, n$,

$$\begin{array}{lll}
(i) & M_{0P_{inp}}^i \in \mathcal{M}; & \overline{M}_{0P_{imp}^\pi}^i \in \mathcal{M}^\pi; \\
(ii) & M_{0O}^i = O \times \{\epsilon\}; & \overline{M}_{0P_o^\pi}^i = P_o^\pi \times \{\epsilon\}; \\
(iii) & M_{0I}^i = M_I^{i-1};_{\pi_i} \overline{M}_{P_o^\pi}^{i-1}; & \overline{M}_{0P_i^\pi}^i = \overline{M}_{P_i^\pi}^{i-1};_{\pi_o} M_O^{i-1}; \\
(iv) & M_{0P\setminus(I\cup O)}^i = M_{P\setminus(I\cup O)}^{i-1}; & \overline{M}_{0P^\pi\setminus(P_i^\pi \cup P_o^\pi)}^i = \overline{M}_{P^\pi\setminus(P_i^\pi \cup P_o^\pi)}^{i-1}.
\end{array}$$

The set of OGCs on π will be denoted by $\sum_\pi$. □

It is worth pointing out that an output guided computation is nothing but a computation of the net $\mathcal{N} \oplus \pi$ feedbacked via $\langle \pi_i \oplus \pi_o, M^s \oplus M^\pi \rangle$ and indeed we could have given such a definition instead of Definition 4.2. However, the motivation for our choice resides in the fact that we wanted to point out the analogy between Output Guided Computations, Feedback Simulating Net on the one hand, and Strategies in the sense of Wittbold and Johnson [WJ90, see also appendix B] on the other. Let us illustrate this using the following example.

Let us consider the nets $\mathcal{N}$ and π shown in figure below. Now, consider the input and output places $\{i\} \cup \{p_{inp}^\pi\}$ and $\{o\} \cup \{p_{out}^\pi\}$ (see figure), such that $\pi_i(p_{out}^\pi) = i$ and $\pi_o(o) = p_{inp}^\pi$. Now an output guided computation, $M_0^0[\sigma_0\rangle M^0, M_0^1[\sigma_1\rangle M^1, \ldots, M_0^n[\sigma_n\rangle M^n$, together with $\overline{M}_0^0[\overline{\sigma}_0\rangle \overline{M}^0, \overline{M}_0^1[\overline{\sigma}_1\rangle \overline{M}^1, \ldots, \overline{M}_0^n[\overline{\sigma}_n\rangle \overline{M}^n$, works in the following way:



$$\begin{array}{lll}
\textsf{step 0} & M_0^0(i) = M^s(i) \xrightarrow{\sigma_0} M^0(o) = \alpha_0 \\
& \overline{M}_0^0(p_{inp}^\pi) = M^\pi(p_{inp}^\pi) \xrightarrow{\overline{\sigma}_0} \overline{M}^0(p_{out}^\pi) = \beta_0 \\
\textsf{step 1} & M_0^1(i) = \beta_0 \xrightarrow{\sigma_1} M^1(o) = \alpha_1 \\
& \overline{M}_0^1(p_{inp}^\pi) = \alpha_0 \xrightarrow{\overline{\sigma}_1} \overline{M}^1(p_{out}^\pi) = \beta_1 \quad . \\
\vdots & \vdots \quad\quad \vdots \\
\textsf{step n} & M_0^n(i) = \beta_{n-1} \xrightarrow{\sigma_n} M^n(o) = \alpha_n \\
& \overline{M}_0^n(p_{inp}^\pi) = \alpha_{n-1} \xrightarrow{\overline{\sigma}_n} \overline{M}^n(p_{out}^\pi) = \beta_n
\end{array}$$

□

Hence a FSN is a particular strategy, while an OGC is a computation of the system in the presence of a strategy on (I,O). By varying the function calculated by the net π, all the strategies can be obtained. Furthermore, Feedback Non-Deducibility on Views which we define now corresponds to Non-Deducibility on Strategies.

Definition 4.3 *(Feedback Non-Deducibility on Views)*
Let $\mathcal{N}$ be a FIFN, $I \subseteq P_{inp}$, $O \subseteq P_{out}$. $Q_1, Q_2 \subseteq P_{ext} \setminus (I \cup O)$ are said to be Feedback Non-Deducible on Views *on (I,O), $Q_1 \not\leftrightarrow_{I,O} Q_2$, if and only if*

$$\begin{array}{l}
\forall (v^i, v^o) \in \mathcal{V}iew(Q_1),\ \forall (w^i, w^o) \in \mathcal{V}iew(Q_2) \text{ and } \forall \pi \text{ FSN on } (I,O), \\
\quad \exists M_0^0[\sigma_0\rangle M^0, \ldots, M_0^n[\sigma_n\rangle M^n \in \sum_\pi \text{ such that } M_{0\Im(Q_1)}^0 = v^i; \quad M_{0\Im(Q_2)}^0 = w^i; \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad M_{\mathcal{O}(Q_1)}^n = v^o; \quad M_{\mathcal{O}(Q_2)}^n = w^o.
\end{array}$$

□

Hence we ask each view of Q_1 and each view of Q_2 to be compatible in the sense that they can be observed in the same computation. This is same as the condition required in the case of NDV. In addition, we require that, for each FSN, at least one of those computations is actually an OGC.

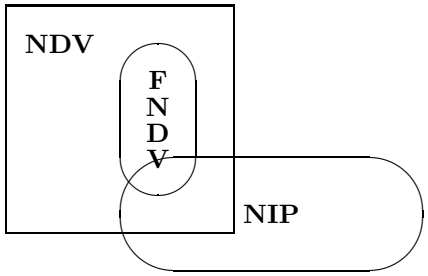
Note that $\not\leftrightarrow_{\emptyset,\emptyset} = \not\leftrightarrow$, since in this case the notion of FSN is trivial and, therefore, OGCs are nothing but computations of the net.

The next proposition shows that FDNV is effectively a restriction of NDV.

Proposition 4.4 *(FNDV ⇒ NDV)*
For any (I,O), $Q_1 \not\leftrightarrow_{I,O} Q_2 \Rightarrow Q_1 \not\leftrightarrow Q_2$.

Proof. Obvious □

Hence in the light of the above proposition and the ones given in earlier sections, we can depict the relationship between NIP, NDV and FNDV as follows:

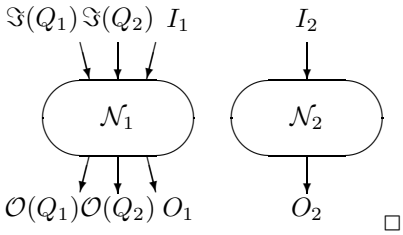


4.1 Composition and Feedback Non-Deducibility on Views

In this subsection, we present the fundamental results regarding the behaviour of FNDV under composition.

Proposition 4.5 *(Parallel Composition and FNDV, part I)*

Let $\mathcal{N}_1$ *be a FIFN,* $I_1 \subseteq (P_1)_{inp}$, $O_1 \subseteq (P_1)_{out}$ *and* $Q_1, Q_2 \subseteq (P_1)_{ext} \setminus (I_1 \cup O_1)$ *such that* $Q_1 \not\leftrightarrow_{I_1,O_1} Q_2$.
Let us consider another FIFN, $\mathcal{N}_2$, *and let* $\mathcal{N}$ *be the net obtained by the parallel composition of* $\mathcal{N}_1$ *and* $\mathcal{N}_2$.
Then, for each $I_2 \subseteq (P_2)_{inp}$, $O_2 \subseteq (P_2)_{out}$, *said* $I = I_1 \cup I_2$ *and* $O = O_1 \cup O_2$, Q_1 *and* Q_2 *are Feedback Non-Deducible on Views on* (I,O) *in* $\mathcal{N}$, *that is,* $Q_1 \not\leftrightarrow_{I,O} Q_2$.

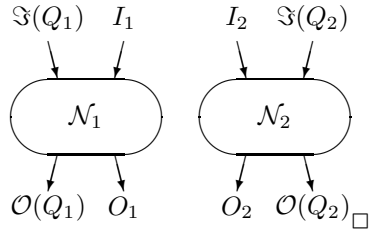


Proof. Trivial. □

Hence parallel composition preserves all the FNDV relations which are true in the original nets. Let us now consider the new relations induced by the composition. In general not all choices of Q_1 and Q_2 are acceptable.

Proposition 4.6 *(Parallel Composition and FNDV, part II)*

Let $\mathcal{N}_1$ *and* $\mathcal{N}_2$ *be FIFNs,* $I_1 \subseteq (P_1)_{inp}$, $O_1 \subseteq (P_1)_{out}$, $I_2 \subseteq (P_2)_{inp}$, $O_2 \subseteq (P_2)_{out}$ *and* $Q_1 \subseteq (P_1)_{ext} \setminus (I_1 \cup O_1)$, $Q_2 \subseteq (P_2)_{ext} \setminus (I_2 \cup O_2)$.
Let $\mathcal{N}$ *be the net obtained by the parallel composition of* $\mathcal{N}_1$ *and* $\mathcal{N}_2$, *and let* $I = I_1 \cup I_2$ *and* $O = O_1 \cup O_2$.
$Q_1 \not\leftrightarrow_{I,O} Q_2$ *in* $\mathcal{N}$ *does not hold.*



Proof. Easy to find an example. □

The next proposition considers the conditions under which the Feedback Non-Deducibility of Q_1 and Q_2 is preserved.

First we need the following definition. In words, it states that a group of users is compatible Feedback Simulating Nets on (I,O) if their set of view is not reduced under the action of any FSN playing on (I,O), i.e. under any strategy involving the channels $I \cup O$.

Definition 4.7 *(Compatibility with FSNs)*
Given a FIFN $\mathcal{N}$, $I \subseteq P_{inp}$, $O \subseteq P_{out}$ and $Q \subseteq P_{ext} \setminus (I \cup O)$, we say that Q is compatible *with FSNs on (I,O), denoted as $Q \not\leftarrow_{I,O}$, if*

$$\forall (v^i, v^o) \in \mathcal{V}iew(Q) \text{ and } \forall \pi \text{ FSN on } (I,O)$$
$$\exists M_0^0[\sigma_0\rangle M^0, \ldots, M_0^n[\sigma_n\rangle M^n \in \textstyle\sum_\pi \text{ such that } M^0_{0\Im(Q)} = v^i \text{ and } M^n_{\mathcal{O}(Q)} = v^o. \qquad \square$$

Proposition 4.8 *(Parallel Composition and FNDV, part III)*
Taken $\mathcal{N}_1$, $\mathcal{N}_2$, I_1, O_1, I_2, O_2, Q_1 and Q_2 as in the previous proposition, let us suppose that $Q_1 \not\leftarrow_{I_1,O_1}$ in $\mathcal{N}_1$ and $Q_2 \not\leftarrow_{I_2,O_2}$ in $\mathcal{N}_2$. Then $Q_1 \not\leftarrow_{I,O} Q_2$ in $\mathcal{N}$.

Proof. Let $(v^i, v^o) \in \mathcal{V}iew(Q_1)$, $(w^i, w^o) \in \mathcal{V}iew(Q_2)$ and $\pi = \langle \pi, pi_i, \pi_o, M^\pi, M^s\rangle$ be an FSN on (I,O), with $\pi = \langle P^\pi, T^\pi, I^\pi, O^\pi, B^\pi, Transop^\pi, S^\pi, \mathcal{M}^\pi\rangle$. Let $P_1^\pi = \pi_o^{-1}(I_1)$ and $P_2^\pi = \pi_o^{-1}(I_2)$.

Clearly, $\mathcal{N}_1^\pi = \langle \pi, \pi_{iP_1^\pi}, \pi_{oO_1}, M^\pi, M^s_{(P_1)_{inp}}\rangle$ is an FSN on (I_1, O_1). Therefore, by hypothesis, there exists an OGC on $\mathcal{N}_1^\pi$

$$M_0^0[\sigma_0\rangle M^0, M_0^1[\sigma_1\rangle M^1, \ldots, M_0^n[\sigma_n\rangle M^n$$

and the corresponding $\overline{M}_0^0[\overline{\sigma}_0\rangle \overline{M}^0, \overline{M}_0^1[\overline{\sigma}_1\rangle \overline{M}^1, \ldots, \overline{M}_0^n[\overline{\sigma}_n\rangle \overline{M}^n$, such that $M^0_{0\Im(Q_1)} = v^i$ and $M^n_{\mathcal{O}(Q_1)} = v^o$.

Now consider the net π in the final marking of the OGC, namely $\overline{M}^n$. By simply moving the strings of tokens in places P_2^π of π to the correspondent (via π_i) places of $\mathcal{N}_2$, we have a new FSN $\mathcal{N}_2^\pi$ on (I_2, O_2). Formally, we have $\pi' = \langle P^\pi, T^\pi, I^\pi, O^\pi, B^\pi, Transop^\pi, S'^\pi, \mathcal{M}^\pi\rangle$, where $S'^\pi = \overline{M}^n_{P^\pi_{int}}$ and

$$\mathcal{N}_2^\pi = \langle \pi', \pi_{iP_2^\pi}, \pi_{oO_2}, M'^\pi, M'^s\rangle,$$

where $M'^\pi = \overline{M}^n_{P^\pi_{inp}}$ and $\left(M'^s = M^s_{I_2};_{\pi_{oO_2}} M^n_{O_2}\right) \oplus M^s_{(P_2)_{inp}\setminus I_2}$.

Now, by hypothesis, there exists an OGC on $\mathcal{N}_2^\pi$

$$M_0^{n+1}[\sigma_{n+1}\rangle M^{n+1}, M_0^{n+2}[\sigma_{n+2}\rangle M^{n+2}, \ldots, M_0^{n+m}[\sigma_{n+m}\rangle M^{n+m}$$

and the corresponding $\overline{M}_0^{n+1}[\overline{\sigma}_{n+1}\rangle \overline{M}^{n+1}, \overline{M}_0^{n+2}[\overline{\sigma}_{n+2}\rangle \overline{M}^{n+2}, \ldots\ldots, \overline{M}_0^{n+m}[\overline{\sigma}_{n+m}\rangle \overline{M}^{n+m}$, such that $M^{n+1}_{0\Im(Q_2)} = w^i$ and $M^{n+m}_{\mathcal{O}(Q_2)} = w^o$.

Let $M = M^s_{I_2} \oplus M^{n+1}_{(P_1)_{inp}\setminus I_2} \oplus S_2$, where S_2 is the initial state of the internal places of $\mathcal{N}_2$, and consider the sequences of firing sequences

$$(M_0^0 \oplus M)[\sigma_0\rangle(M^0 \oplus M), \ldots, (M_0^n \oplus M)[\sigma_n\rangle(M^n \oplus M),$$
$$(M^n_{P_1} \oplus M_0^{n+1})[\sigma_{n+1}\rangle(M^n_{P_1} \oplus M^{n+1}), \ldots, (M^n_{P_1} \oplus M_0^{n+m})[\sigma_{n+m}\rangle(M^n_{P_1} \oplus M^{n+m}),$$

and

$$\overline{M}_0^0[\overline{\sigma}_0\rangle \overline{M}^0, \ldots, \overline{M}_0^n[\overline{\sigma}_n\rangle \overline{M}^n, \overline{M}_0^{n+1}[\overline{\sigma}_{n+1}\rangle \overline{M}^{n+1}, \ldots, \overline{M}_0^{n+m}[\overline{\sigma}_{n+m}\rangle \overline{M}^{n+m}.$$

Clearly we have that

$$(M_0^0 \oplus M)_{\Im(Q_1)} = v^i; \qquad (M^n \oplus M)_{\mathcal{O}(Q_1)} = v^o;$$
$$(M^n_{P_1} \oplus M_0^{n+1})_{\Im(Q_2)} = w^i; \qquad (M^n_{P_1} \oplus M^{n+m})_{\mathcal{O}(Q_2)} = w^o.$$

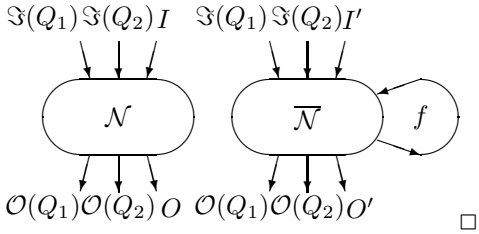
Observe that the sequences above do not form an OGC on π because conditions *(ii)* and *(iii)* in Definition 4.2 are not satisfied. In particular in the $\overline{M}_0^i$, $i = 1, \ldots, n$, the output places in P_2^π are not necessarily empty and their content is not moved to the correspondent places on $\mathcal{N}_2$. The symmetric case happens in the rest of the sequences, for the $\overline{M}_0^{n+i}$ and the places corresponding to P_1^π in $\mathcal{N}_1$. However, since π is a FSN on (I,O) and, therefore, $\mathcal{N}_1 \oplus \pi \oplus \mathcal{N}_2$ is feedbackable via $\langle \pi_i \oplus \pi_o, M^s \oplus M^\pi\rangle$, it is now a trivial task to see that this problems can be eliminated in the obvious way, so getting an OGC on π in which (v^i, v^o) and (w^i, w^o) are observed.

This concludes the proof. $\square$

Finally, we can show that FNDV is preserved by feedback.

Proposition 4.9 *(Feedback and FNDV)*
Let $\mathcal{N}$ be a FIFN with $I \subseteq P_{inp}$, $O \subseteq P_{out}$ and $Q_1, Q_2 \subseteq P_{ext} \setminus (I \cup O)$ such that $Q_1 \overleftarrow{\not\rightarrow}_{I,O} Q_2$.
Let $P_o \subseteq O$, $P_i \subseteq I$, $f : P_o \to P_i$ and $M^s \in \mathcal{N}$ such that $\mathcal{N}$ is feedbackable via $\langle f, M^s \rangle$ and let $\overline{\mathcal{N}}$ be the feedbacked net.
Then, for each $I' \subseteq I \setminus P_i$ and $O' \subseteq O \setminus P_o$ we have $Q_1 \overleftarrow{\not\rightarrow}_{I',O'} Q_2$ in $\overline{\mathcal{N}}$.

Proof. Trivial: each FSN π on (I', O') for the net $\overline{\mathcal{N}}$ can be thought of as a FSN $\pi \oplus \pi'$ on (I, O) for $\mathcal{N}$ where π' is the net which takes the tokens in P_o and delivers them to the corresponding (via f) places P_i without modifying them and which takes the tokens in $O \setminus (O' \cup P_o)$ and loses them.



□

5 Conclusions and Further Work

First we briefly summarise the main results described in this paper and then consider some of the implications of this work.

Summary

We started by introducing *FIFO Information Flow Nets (FIFN)* model. FIFN is based on Petri nets and has been derived from the work described in [Var89], [Var90] and [Rou86]. Using this new model, we then presented the information flow security properties *Non-Interference on Places (NIP)* and *Non-Deducibility on Views (NDV)* which correspond to Non-Interference and Non-Deducibility on Inputs. Then we went on to show that in the FIFN context, NIP does not always imply NDV and developed conditions under which this is true. We then considered a general composition operation and showed that neither NIP nor NDV is preserved under this operation. This led to the definition of a new information flow security property, referred to as the Feedback Non-Deducibility on Views (FNDV), which is shown to be preserved under the composition operation. We then showed the similarities between FNDV and Non-Deducibility on Strategies recently proposed by [WJ90].

Remarks

This paper has reported several new ideas in the area of information flow security. The proposed FIFN model is mathematically based and avoids unrealistic hypotheses such as infinite buffers, total inputs, non-blocking communication or infinitely fast systems.

We believe that the FIFNs provide a uniform framework for specifying several information flow security properties. This is supported by the fact that we are able to model the major information flow security definitions using FIFNs (without bringing any conceptual change to the original definitions). The significance of this is that one is able to compare the different notions of information flow security using a single underlying computational model. This we believe was one of the major deficiencies of the earlier work. The assumption we made about the channels was their *FIFO* organization. This is not a strong hypothesis as it reflects the way in which computer systems are usually organized. Furthermore, in an expressive enough formalism, any buffer policy can be implemented using FIFO buffers as a primitive tool by means of processes which read the messages in the FIFO order and organize them in their internal memory in the order specified by the desired policy. We believe that this is also the case for the FIFN formalism.

It may seem at times that the model presented in this paper is a bit heavy to handle. This is partly due to notations and partly due to the level of generality with which we wanted to treat the problems. Part of the complexity of the notations is due to the fact that we have given net based formulations of previously defined concepts. We feel that new definitions of information flow (e.g. in [Var90]) using typical characteristics of nets will be simpler.

Acknowledgements

We wish to thank Phillip Allen at Hewlett-Packard Labs. Bristol for many interesting and useful discussions, and the anonymous referees of the Journal of Computer Security whose suggestions led to improvements of this paper.

References

[**BLP74**] D.E. Bell and L.J. LaPadula, *Secure Computer System*, Technical Report M74-244, The MITRE Corporation, Bedford, MA, 1974.

[**Den83**] D.E. Denning, *Cryptography and Data Security*, Addison Wesley, 1983.

[**GM82**] J. Goguen and J. Meseguer, Security Policies and Security Models, in *Proceedings of the 1982 IEEE Symposium on Security and Privacy*, 1982.

[**Hoa85**] C.A.R. Hoare, *Communicating Sequential Processes*, Prentice-Hall, 1985.

[**JT88**] D.M. Johnson and F.J. Thayer, Security and Composition of Machines, in *Proceedings of the Computer Security Foudation Workshop*, 1988.

[**Lan81**] C.E. Landwehr, Formal Models of Computer Security, in *Computing Surveys, 13, 3*, 1981. 1

[**McC88**] D. McCullough, Non-Interference and Composability of Security Policies, in *1988 IEEE Symposium on Security and Privacy*, 1988.

[**Mln90**] J.K. Millen, Hookup Security for Synchronous Machines, 1990.

[**Mil80**] R. Milner, A Calculus of Communicating Systems, *Lecture Notes in Computer Science 92*, Springer Verlag, 1980.

[**Pet62**] C.A. Petri, *Kommunikation mit Automaten*, Ph.D. Thesis, Institut für Instrumentelle Mathematik, Bonn, FRG, 1962.

[**Pets81**] J.L. Peterson, *Petri Net Theory and the Modelling of Systems*, Prentice-Hall, 1981.

[**Rei85**] W. Reisig, *Petri nets : An Introduction*, EATCS Monographs on Theoretical Computer Science, Springer-verlag, 1985.

[**Rou86**] G. Roucairol, FIFO - NETS, in *Advanced Course on PETRI NETS*, GMD, Bad Honnef 8–19 Sept. 1986.

[**Sut86**] D. Sutherland, A Model of Information, in *Proceedings of the 9th National Computer Security Conference*, 1986.

[**Var89**] V. Varadharajan, Petri Net based System Design and Refinement, *Journal of Systems and Software*, Vol.15, No.3, July 1991.

[**Var90**] V. Varadharajan, Petri Net based Modelling of Information Flow Security Requirements, in *Proceedings of the 1990 Foundations of Computer Security Workshop*, Franconia, 1990.

[**Var91**] V. Varadharajan, Hook-Up Property for Information Flow Secure Nets, *Proceedings of the 1991 Foundations of Computer Security Workshop*, Franconia, 1991.

[**WJ90**] J.T. Wittbold and D.M. Johnson, Information Flow in Nondeterministic Systems, in *Proceedings 1990 IEE Computer Society Symposium on Research in Security and Privacy*, 1990.

A Petri Nets

In this appendix, we give a very brief introduction to some of the fundamental principles of Petri nets which are relevant to this paper. For detailed treatment of Petri nets, the reader is referred to [Pets81], [Rei85].

A Petri net is a particular kind of a directed graph with two types of nodes, namely the *places* (graphically depicted as circles) and the *transitions* (graphically depicted as bars). The basic structure of a Petri net consists of a set of places, a set of transitions and a set of directed arcs which connect the transitions and the places.

An arc directed from a place p_i to a transition t_j defines that place to be an *input place* of transition t_j. Similarly, an arc connecting transition t_j to a place p_k implies that the place p_k is an output place of the transition t_j. In particular, a place can be a multiple input or output place of a certain transition and this is represented using multiple arcs.

The *state* of a Petri net is described by the distribution of markers, called *tokens* in the places of the net. Tokens are represented by dots drawn inside the places which have them. A particular assignment of tokens is referred to as the *marking* of the Petri net. Formally, markings can be described as mappings from the set of places of the net to $\mathbb{N}$, the set of natural numbers, which associate with a place the number of tokens it contains.

A Petri net can be formally define as follows [Pets81]:

Definition A.1 *(Petri Nets)*
A Petri Net N *is defined as a five tuple,* $N = \langle P, T, I, O, M_0 \rangle$*, given by:*

$P = \{p_1, p_2, \ldots, p_n\}$ *is a finite set of Places*

$T = \{t_1, t_2, \ldots, t_m\}$ *is a finite set of Transitions*

$I : T \to [P \to \mathbb{N}]$ *is the input function which maps a transition to a bag of places.*

$O : T \to [P \to \mathbb{N}]$ *is the output function which maps a transition to a bag of places.*

$M_0 \in [P \to \mathbb{N}]$ *is a bag representing the* initial marking *of the net.* □

Note that a *bag* is similar to a *set*, in that it is a collection of items, except that multiple occurrence of items is allowed.

A place p_i is an input place to a transition t_j if $p_i \in I(t_j)$, and p_i is an output place of the transition if $p_i \in O(t_j)$. The input and output functions can be extended to map places to bags of transitions, so that we can refer to t_j as an input or output transition of the place p_i.

Definition A.2 *(Transitions Enabled)*
A transition t_j *in a marked Petri net* $N = \langle P, T, I, O, M_0 \rangle$ *with marking* M *is* enabled *if*

$$\forall p_i \in P,\ M(p_i) \geq mult(p_i, I(t_j)),$$

where $mult(x, Y)$ *is the multiplicity of item* x *in bag* Y*.* □

Definition A.3 *(Firing Rule)*
A transition t_j *in a marked Petri net with marking* M *may* fire *whenever it is enabled. Firing results in a new marking* M' *defined by:*

$$M'(p_i) = M(p_i) - mult(p_i, I(t_j)) + mult(p_i, O(t_j)),\ \textit{for each}\ p_i \in P.$$

□

So we see that two sequences result from the execution of a Petri net, namely the *marking sequence* and the *firing sequence*. The sequence of markings correspond to the sequence of states reached by the net and the sequence of transitions reflect the transitions that were fired during the execution of the Petri net.

Definition A.4 *(Immediate Reachability)*
A marking M' *is* immediately reachable *from a marking* M *if the firing of some transition* t *in* M *will yield* M'*. We will write this as* $M[t\rangle M'$*.* □

Definition A.5 *(Reachability)*
A marking M' is reachable *from M if it is immediately reachable from M or is reachable from any marking which is immediately reachable from M or if it is M itself. Thus M' is reachable from M if there exists a firing sequence $t_1, \ldots, t_n$ starting from M that results in M', that is, $M[t_1, \ldots, t_n\rangle M'$.* □

We can now define the set of reachable markings from some initial marking M to be $\mathcal{R}(N, M)$.

Definition A.6 *(Reachability Set and Firing Sequences)*
Given a Petri net N, with initial marking M_0, the reachability set *for M is defined to be*

$$\mathcal{R}(N, M_0) = \Big\{ M1 \in [P \to I\!N] \;\Big|\; \exists \sigma \in T^* \text{ and } M_0[\sigma\rangle M \Big\}.$$

The set of firing sequences *of $\mathcal{N}$ is defined to be*

$$\mathcal{FS}(N, M_0) = \Big\{ \sigma \in T^* \,|\, \exists M \in \mathcal{R}(N, M_0) \text{ and } M_0[\sigma\rangle M \Big\}.$$

□

B Security

In this appendix, we will give some basic results about security, which constitute the starting point for our work and are necessary to understand the present paper. Actually, we will just recall the main concepts of Non-Interference [GM82], Non-Deducibility [Sut86] and Non-Deducibility on Strategy [WJ90], simplifing and purging the exposition as much as possible. The reader who needs a wider introduction or wants to know the other approaches which have been developed is referred to the bibliography [BLP74, Lan81, Den83, McC88, JT88, Mln90]

Non-Interference

The results in this subsection are due to Goguen and Meseguer and can be found in [GM82].

Goguen and Meseguer base their approach to security on an abstract machine used to describe systems and a basic mechanism, the Non-Interference assertion, used to specify security policies.

Informally, a Non-Interference assertion between two groups of users is a statement of the kind:

" what a group of users does using the system has
no effect on what the other group of users sees".

A security policy can be expressed by means of Non-Interference assertions ensuring that information does not flow where it should not.

Let us give the previous concepts in a formal way.

Definition B.1 *(State Machine)*
A state machine $\mathcal{M}$ *consists of the following:*

- *a set* $\mathcal{U}$ *whose elements are called* users*;*
- *a set* $\mathcal{S}$ *whose elements are called* states*;*
- *a set* $\mathcal{C}$ *whose elements are calles* commands*;*
- *a set* $\mathcal{O}ut$ *whose elements are called* outputs*;*
- *a function* $out : \mathcal{S} \times \mathcal{U} \to \mathcal{O}ut$ *which represents the users view, called* output function*;*
- *a function* $do : \mathcal{S} \times \mathcal{U} \times \mathcal{C} \to \mathcal{S}$ *which represents the updating of the state, called* state transition function*;*
- *a constant* $s_0 \in \mathcal{S}$*, called* initial machine state. □

We can extend the state transition function in the classical way to describe computations, i.e. the effects of strings of inputs on the machine state, and define what an user view of a computation is.

Definition B.2 *(Computations and Views)*
Given a state machine $\mathcal{M}$ *we define* $sdo : \mathcal{S} \times (\mathcal{U} \times \mathcal{C})^* \to \mathcal{S}$ *by the equations*

$$sdo(s, \epsilon) = s \text{ and}$$
$$sdo\Big(s, w \cdot (u, c)\Big) = do\Big(sdo(s, w), u, c\Big) \text{ with } s \in \mathcal{S},\ u \in \mathcal{U},\ c \in \mathcal{C} \text{ and } w \in (\mathcal{U} \times \mathcal{C})^*$$

where ϵ *is the empty string and "·" denotes strings concatenation.*
Moreover, $\forall u \in \mathcal{U}$ *we define* $\| \cdot \|_u : (\mathcal{U} \times \mathcal{C})^* \to \mathcal{O}ut$*, the view of the user* u*, by the equation*

$$\|w\|_u = out\Big(sdo(s_0, w)\Big) \qquad \forall w \in (\mathcal{U} \times \mathcal{C})^*$$ □

Definition B.3 *(Purged Inputs)*
Given $G \subseteq \mathcal{U}$ *and* $w \in (\mathcal{U} \times \mathcal{C})^*$ *we denote by* $\mathcal{P}_G(w)$ *the subsequence obtained from* w *purging those pairs* (u, c) *with* $u \in G$*.* □

Finally, we are ready for the definition of Non-Interference.

Definition B.4 *(Non-Interference)*
Given a state machine $\mathcal{M}$ and sets $G, G' \in \mathcal{U}$, G is non-interfering *with G', written $G \,|\!:\! G'$, if and only if*

$$\|w\|_u = \|\mathcal{P}_G(w)\|_u \qquad \forall w \in (\mathcal{U} \times \mathcal{C})^* \text{ and } \forall u \in G' \qquad \square$$

It has been noticed that the model chosen by Goguen and Meseguer is not completely general, because, by assuming that the output and the state are functions of the input, they restrict their approach to deterministic systems.

Non-Deducibility

Starting from the final consideration of the previous subsection, Sutherland has developed his definition of Non-Deducibility [Sut86].

In this approach a system is represented as a set of possible *worlds*, corresponding to the set of possible execution sequences of an automata, users views are represented by *information functions* which extract a particular *view of the system* from a *world*, and the basic mechanism to express security policies is the *information flow*, whose meaning is explained by the following:

Definition B.5 *(Information Flow I)*
Given the set of possible worlds $\mathcal{W}$, a set $\mathcal{V}$ of views and $f, g : \mathcal{W} \to \mathcal{V}$ information functions, information does not flow from f to g if and only if the function $f \times g : \mathcal{W} \times \mathcal{W} \to \mathrm{Img}(f) \times \mathrm{Img}(g)$ is onto. $\square$

The previous definition can be given in an equivalent way, which is, perhaps, more intuitive [WJ90].

Definition B.6 *(Information Flow II)*
Given $\mathcal{W}, f$ and g as above information flows *from f to g if and only if there exist $w \in \mathrm{Img}(g)$ and $v \in \mathrm{Img}(f)$ such that for each world (trace) $t \in \mathcal{W}$ $f(t) = v \Rightarrow g(t) \neq w$.* $\square$

The interpretation of the latter is very easy: when the user represented by f observes v, he will know something about what the user represented by g sees, exactly that it is not w.

A security policy can be expressed by means of a predicate *legal-to-get* over pairs of information functions which specifies from where to where information is supposed to flow legally in the system under observation.

Definition B.7 *(Secure System)*
Given a set of possible worlds $\mathcal{W}$, a set of information functions $\Im$ and a predicate legal-to-get $\subseteq \Im \times \Im$, we say that the system $(\mathcal{W}, \Im)$ is secure *if and only if whenever information flows from f to g, both belonging to $\Im$, then legal-to-get(f, g).* $\square$

Sutherland's definition solves the problem of treating nondeterminism generalizing the Non-Interference, having been proved that in case of deterministic systems the two definitions do coincide, but it is not composable, in the sense that composing two systems for which the definition holds a non-secure system can be obtained.

Non-Deducibility on Strategy

An answer to the problem of composability has been given by Wittbold and Johnson [WJ90] defining a restriction of Non-Deducibility preserved under composition. In this subsection we will recall their definition.

The model consist of a synchronized nondeterministic state machine, controlled by two users/processes, T (*high transmitter*) and R (*low receiver*), which send their inputs to the machine contemporarily. From the inputs and the state the machine produces deterministically, the outputs, which are delivered contemporarily to the users, and nondeterministically its new state.

The formal definitions follows.

Definition B.8 *(Synchronised State Machine)*
A synchronized state machine $\mathcal{M}$ *consists of the following:*

- $\mathcal{S}$*, the set of states with* $S_0 \in \mathcal{S}$ *the set of initial states;*
- $\mathcal{I}_R$*, the set of inputs from receiver* R*;*
- $\mathcal{I}_T$*, the set of inputs from trasmitter* T*;*
- $\mathcal{O}_R$*, the set of outputs to receiver* R*;*
- $\mathcal{O}_T$*, the set of outputs to transmitter* T*;*
- $\mathcal{N} : \mathcal{S} \times \mathcal{I}_R \times \mathcal{I}_T \to \wp(\mathcal{S}) \setminus \emptyset$*, the next-state function;*
- $\mathcal{O}ut_R : \mathcal{S} \times \mathcal{I}_R \times \mathcal{I}_T \to \mathcal{O}_R$*, the output function for* R*;*
- $\mathcal{O}ut_T : \mathcal{S} \times \mathcal{I}_R \times \mathcal{I}_T \to \mathcal{O}_T$*, the output function for* T*.* □

Definition B.9 *(Moves and Traces)*
A move *of a synchronized state machine* $\mathcal{M}$ *has the form* $s_{m-1}(i_m, j_m, k_m, l_m)s_m$*, where* $s_{m-1}, s_m \in \mathcal{S}$*,* $i_m \in \mathcal{I}_R$*,* $j_m \in \mathcal{I}_T$*,* $k_m \in \mathcal{O}_R$*,* $l_m \in \mathcal{O}_T$ *such that*

$$s_m \in \mathcal{N}(s_{m-1}, i_m, j_m);$$
$$\mathcal{O}_R(s_{m-1}, i_m, j_m) = k_m;$$
$$\mathcal{O}_T(s_{m-1}, i_m, j_m) = l_m.$$

A trace or execution of the state machine is a finite sequence of moves starting from an initial state $s_0 \in S_0$*.* □

The basic security mechanism is constituted by strategies, low views (the projection on low inputs and outputs of traces) and the concept of *consistence* between them. A strategy for a user/process links its input with its past input/output history, that is it can determinate the next input from the process looking at its previous inputs and outputs. Formally, a strategy is a function from sequences of pairs (input,output) of a user/process to the its input, as stated by the following.

Definition B.10 *(High Transmitter Strategies)*
A strategy *of length* n *is a sequence of* n *functions* $\pi = (\pi^1, \ldots, \pi^n)$ *where for each* $1 \leq i \leq n$*,*

$$\pi^i : (\mathcal{I}_T \times \mathcal{O}_T)^{(i-1)} \to \mathcal{I}_T.$$ □

Definition B.11 *(Compatibility and Consistence)*
Let $t = s_0(i_1, j_1, k_1, l_1)s_1, \ldots, s_{n-1}(i_n, j_n, k_n, l_n)s_n$*, be a trace of length* n*,* $\pi = (\pi^1, \ldots, \pi^n)$ *be a strategy of length* n *and* $\lambda = (\overline{\imath}_1, \overline{k}_1, \ldots, \overline{\imath}_n, \overline{k}_n)$ *be a low view of length* n*. We say that:*

i. λ *is* compatible *with* t *if and only if* $\overline{\imath}_r = i_r$ *and* $\overline{k}_r = k_r$ *for each* $1 \leq r \leq n$*, i.e. the view is the low projection of the trace.*

ii. π *is* compatible *with* t *if and only if* $\pi^r(j_1, l_1, \ldots, j_{r-1}, l_{r-1}) = j_r)$ *for each* $1 \leq r \leq n$*, i.e. the strategy is contained in the trace.*

iii. λ *is* consistent *with* π *if and only if there exists a trace* t *such that* λ *and* π *are compatible with* t*, i.e. they can be observed in the same trace.* □

Finally, we can give the definition of security for system, saying that a system is non-deducible on strategy if whatever view the *low receiver* has, every strategy can have been used by the *high transmitter*.

Definition B.12 *(Non-Deducibility on Strategy)*
A synchronized state machine is Non-Deducible on Strategy *if and only if for any* n*, any low view of length* n *is consistent with any strategy of the same length.* □