Strategy-based and knowledge-based models of information flow: equivalence and decidability

Cătălin Dima
LACL, Université Paris 12,
61 av. du Général de Gaulle,
94010 Créteil Cedex, France

Constantin Enea*
University "Al. I. Cuza",
Str. General Berthlot 16,
740083, Iaşi, Romania

Radu Gramatovici
University of Bucharest,
str. Academiei 14,
010014 Bucureşti, Romania

Alexandru Sofronia
University of Bucharest,
str. Academiei 14,
010014 Bucureşti, Romania

Abstract

We show that two notions of information flow, one which is strategy-based and the other knowledge-based, are equivalent. The first notion models the possibilities that a high-level user has to send information to the low-level user by distinctly observable runs, whereas the second notion models information leak as variations in the knowledge of the low-level user during his observation of the system behavior. We also give an algorithm for deciding whether a finite-state system has no information flow in our setting, by reducing the problem to a reachability problem in finite automata.

1 Introduction

Information flow analysis is one of the main techniques used when checking whether confidentiality is ensured in some system. There are several intuitions behind information flow, and they have been informally synthesized in statements about the (non-)dependence of high-level ("confidential") activity and low-level ("public") observation and/or deduction. These statements have produced a number of non-equivalent notions, which resulted in a span of trace-based possibilistic models [15, 9, 12] which are able to detect a wide variety of types of information flow, including information flow through timing [5].

In this paper, we are interested in investigating the relationship between a high-level, strategy based view of information leak, and a low-level, deducibility based view, in general systems represented as sets of traces. The strategy based view is inspired by Nondeducibility on Strategies [14] or Nondeducibility on Compositions [4], whereas the deducibility view is similar to the semantic framework of the Temporal Logic of Knowledge of [7]. The first is centered around the idea that, in a system, *High* can unintentionnaly leak information to *Low* if there exists a low-level projection of a run that can only come from one type of *High*-strategies, and not from others. In other words, *High* can transmit a 0 by choosing a strategy that can be observed by *Low* as a run ρ, and transmit a bit of 1 by choosing another strategy that cannot be observed as ρ. For the deducibility notion, we consider that, at each moment during a system run ρ, *Low* can make deductions of what *High* could have been doing to provoke ρ. Then, *Low* gains unwanted information about *High*'s activity when, at a moment t, *Low* rules out some *High* behavior that he considered possible at $t-1$.

We show that the ambiguity of the trace model about what to consider *High*'s choice and what is the system choice induces some problems when trying to relate the two models. We then show that the two views can be made equivalent if we solve this ambiguity by "making system decisions visible" – or, in other words, we consider a worst case scenario, in which the system choices provoke themselves information leak.

We investigate the relationship between our approach and other existing approaches, like Generalized Noninterference [11], Separability [12], Forward Correctability [8, 10] and the Perfect Security Property [15]. To this end, we give a slight generalization of our model, constructed around the idea that *Low* is interested in information regarding only a *part* of *High*'s choices. We finally show that, for systems specified by finite automata, the two notions of information flow are decidable. This result is obtained by

*Work partly done while visiting LACL at Université Paris 12 means of reducing the problem to a reachability problem in finite automata.

The paper goes on as follows: in the next section we give the two notions of information flow, together with some simple examples supporting the intuition. We then discuss, in the third section, the problem posed by handling system's choices between permitting or disallowing *High*'s actions within the deducibility-based model, and prove the equivalence between the two models of information flow. In the fourth section we generalize our definition to the case where some of the high-level actions are "not interesting" for *Low*, and prove that this case reduces to the particular case of section 2. The fifth section serves for comparing our notions of information flow with existing notions in the litterature. In the sixth section we give the decidability result, and end with a section with conclusions.

2 The trace model

A **trace** is a finite sequence of events in E, i.e., a *word* over E^*. By $|t|$ we denote the length of a trace t, i.e. the number of events in t. $\preceq$ denotes the prefix ordering of E^*, hence $w \preceq w'$ if there exists $z \in E^*$ such that $wz = w'$. The set of prefixes of a trace w is denoted $\mathsf{pref}\{w\}$ and the set of prefixes of a set of traces T is denoted $\mathsf{pref}T$. For each trace $t = a_1 \dots a_n \in E^*$ and $0 \leq i \leq n$, $t[1..i] = a_1 \dots a_i$ denotes the prefix of t composed of the first i letters. The i-th symbol in t is denoted $t[i]$.

The set of high-level actions, which are invisible to the low-level user, *Low*, are modeled as a subset $\mathsf{Inv} \subseteq E$, while the set of observations that *Low* can make are modeled as a subset $\mathsf{Vis} \subseteq E$. We suppose, in this section, that $E = \mathsf{Vis} \cup \mathsf{Inv}$ and $\mathsf{Vis} \cap \mathsf{Inv} = \emptyset$. The observation that *Low* can make of a trace $t \in E^*$ is its *projection* over the set of visible events, which is obtained by removing the high-level events. We denote here $o : E^* \to \mathsf{Vis}^*$ the observation mapping, hence $o(h_1 l_1 l_2 h_2) = l_1 l_2$, if $h_1, h_2 \in \mathsf{Inv}$ and $l_1, l_2 \in \mathsf{Vis}$.

We identify a system with its set of traces, $\mathrm{Tr} \subseteq E^*$. We assume that Tr is **prefix-closed**, that is, for each $w \in \mathrm{Tr}$ and $w' \preceq w$, $w' \in \mathrm{Tr}$.

Definition 1 *A **strategy** for* High *for interacting with a system* Tr *is a mapping* $s : E^* \to \mathsf{Inv} \cup \{\perp\}$ *satisfying the following properties:*

- *For each* $w \in E^*$, *if* $s(w) = e \in \mathsf{Inv}$ *then* $we \in \mathrm{Tr}$.
- *For each* $w \in E^*$, *if* $s(w) = \perp$ *then there exist some* $e \in \mathsf{Vis}$ *with* $we \in \mathrm{Tr}$.

*The two requirements are called the **consistency requirements** for the strategy* s.

Hence, for $w \in E^*$, $s(w) \in E$ denotes the fact that the system lets *High* choose an invisible move, whereas $s(w) = \perp$ denotes the fact that the system forbids *High* to choose any move.

The *set of behaviors* spawned by a strategy is the set of behaviors that are produced according to *High*'s decisions encoded in the strategy:

$$\begin{aligned}\mathsf{Beh}(s) = \mathsf{pref}\big\{ & t \in \mathrm{Tr} \mid |t| = n, \forall i \leq n, \text{if } s(t[1..i]) = \perp \\ & \text{then } t[i+1] \in \mathsf{Vis} \text{ or } i = n \text{ and if } s(t[1..i]) \in \mathsf{Inv} \\ & \text{then } n > i \text{ and } t[i+1] = s(t[1..i]) \big\}\end{aligned}$$

Example 2 *Consider the system* $\mathrm{Tr} = \mathsf{pref}\{h_1 l_1, h_2 l_2\}$, *in which* $\mathsf{Inv} = \{h_1, h_2\}$ *and* $\mathsf{Vis} = \{l_1, l_2\}$. *The two "essentially different" strategies for* High *in* Tr *are* $s_1(\varepsilon) = h_1$, $s_1(h_1) = \perp$, *resp.* $s_2(\varepsilon) = h_2$, $s_2(h_2) = \perp$. *The sets of behaviors are respectively* $\mathsf{Beh}(s_i) = \mathsf{pref}\{h_i l_i\}$ *(*$i = 1, 2$*).*

Example 3 *Consider now the system* $\mathrm{Tr}' = \mathsf{pref}\{h_1 l_1, h_1 l_2, h_2 l_1, h_2 l_2\}$ *with the same sets of visible and invisible events. Here again* High *has only two "essentially different" strategies:* $s_1(\varepsilon) = h_1$, $s_1(h_1) = \perp$ *and* $s_2(\varepsilon) = h_2$, $s_1(h_2) = \perp$. *The sets of behaviors are respectively* $\mathsf{Beh}(s_i) = \mathsf{pref}\{h_i l_1, h_i l_2\}$ *(*$i = 1, 2$*).*

Example 4 *Consider now the system* $\mathrm{Tr}_{zeno} = \mathsf{pref}\{h^n l h^m l \mid m, n \geq 0\}$, *with* $E = \{h, l\}$, $\mathsf{Vis} = \{l\}$. *As a first example,* $h^n \perp h^m \perp$ *denotes the following strategy:*

$$s_1(h^i) = \begin{cases} h & \text{if } i < n \\ \perp & \text{if } i = n \end{cases} \quad s_1(h^n l h^i) = \begin{cases} h & \text{if } i < m \\ \perp & \text{if } i = m \end{cases}$$

The set of behaviors is $\mathsf{Beh}(s_1) = \{h^k, h^n l h^p l \mid k \leq n, 0 \leq p \leq m\}$.

Example 5 *In the framework of Example 4,* $s_{zeno} = h^n \perp h^\omega$ *denotes the following strategy for* High*:*

$$s_{zeno}(h^i) = \begin{cases} h & \text{if } i < n \\ \perp & \text{if } i = n \end{cases} \quad s_{zeno}(h^n l h^k) = h, \forall k \geq 0$$

The problem with this strategy is that the system permits *High* to decide to run forever. This is a strategy that we will disallow in the following, for two reasons:

1. Infinite behaviors cannot be properly observed.
2. In any realistic model, unboundedly many actions should eventually make time progress with an unboundedly amount of time – in order to avoid the so-called *Zeno phenomenon*.

In our framework, the only object that could have been "generated" by the strategy s_{zeno} is the *infinite word* $w = h^n l h^\omega$. But, by definition, $\mathrm{Tr} \subseteq E^*$, hence w is not a member of Tr, and $\mathsf{Beh}(s_{zeno})$ cannot contain prefixes of an infinite word.

We consider here systems accepting only finite behaviors, in order to avoid a phenomenon similar to the *Zeno phenomenon* [2]: the fact that infinitely many invisible actions take place but *Low* doesn't observe that. A more proper treatment of this phenomenon would require considering a dense time semantics for systems, which is out of the scope of our paper.

Definition 6 *A strategy s is* ***non-Zeno*** *if there exists no infinite sequence of words $\left(t_n\right)_{n\geq 0}$ such that $s(t_n) \in \mathsf{Inv}$ and $t_{n+1} = t_n s(t_n)$.*

A strategy is ***Zeno for*** *t_0 if it satisfies the negation of the above property, with the sequence of words starting in t_0.*

Remark 7 *Note that for each strategy which is Zeno for some trace t_0, no trace that extends t_0 can be in $\mathsf{Beh}(s)$, that is, for all $t \succeq t_0$, $t \notin \mathsf{Beh}(s)$.*

With strategies meant to model high-level choices, the lack of information flow should be restated as the fact that *distinct strategies cannot be distinguished by observation*. For example, the system Tr in Example 2 above should not satisfy noninterference since the trace $h_1 l_1 \in \mathsf{Beh}(s_1)$ gives the observation l_1 and there is no trace in $\mathsf{Beh}(s_2)$ that gives the same observation. On the other hand, Tr' from Example 3 should satisfy noninterference since for both $i = 1, 2$, there exists a trace that gives the observation l_i in both $\mathsf{Beh}(s_1)$ and $\mathsf{Beh}(s_2)$. The formalization is the following:

Definition 8 *A system Tr* ***has nondeterministic noninterference*** *if for any two strategies s_1, s_2, the following property holds*

$$\textbf{(NNI)} \quad o\big(\mathsf{Beh}(s_1)\big) = o\big(\mathsf{Beh}(s_2)\big)$$

Deducible information flow. So far, we have seen *High*'s definition of "information flow". The next step is to define *Low*'s view on it. To this end we will make use of the *low-level equivalent set* from [15], denoted $\mathsf{LLES}(w, \mathrm{Tr})$, which stands for the set of traces in the system that produce the same observation as w (from *Low*'s point of view):

$$\mathsf{LLES}(w, \mathrm{Tr}) = \big\{ w' \in \mathrm{Tr} \mid o(w) = o(w') \big\}$$

This is generally considered to belong to the knowledge of *Low* after each occurrence of the trace w.

Since the set of traces of the system is prefix closed, we consider that *Low*'s knowledge after the occurrence of w is composed of all the prefixes of the traces that give the same observation. We then denote $\mathsf{knl}(w,\mathrm{Tr})$ *Low*'s *knowledge* after the behavior w has passed in the system Tr:

$$\begin{aligned}\mathsf{knl}(w,\mathrm{Tr}) &= \big\{ z \mid z \preceq w' \text{ for some } w' \in \mathsf{LLES}(w, \mathrm{Tr}) \big\} \\ &= \mathsf{pref}\big(\mathsf{LLES}(w, \mathrm{Tr})\big)\end{aligned}$$

Definition 9 *The system represented by Tr* ***has no (deducible) information flow*** *if for all $w, w' \in \mathrm{Tr}$ such that $w \preceq w'$, we have that:*

$$\mathsf{knl}(w,\mathrm{Tr}) \subseteq \mathsf{knl}(w',\mathrm{Tr}).$$

Tr ***has information flow*** *if there exists $w, w' \in \mathrm{Tr}$ with $w \preceq w'$ for which the inclusion is not satisfied.*

In other words, in a system having information flow, *Low* can pass from a state in which he cannot rule out some traces, to a state in which he deduces that *certain traces couldn't have previously occurred*.

Example 10 *Consider again the following system $\mathrm{Tr} = \mathsf{pref}\big\{h_1 l_1, h_2 l_2\big\}$ from Example 2. In the system Tr we have information flow because*

$$\mathsf{knl}(h_1,\mathrm{Tr}) = \mathsf{pref}\big\{h_1, h_2\big\} \not\subseteq \mathsf{knl}(h_1 l_1,\mathrm{Tr}) = \mathsf{pref}\big\{h_1 l_1\big\}.$$

This corresponds to the following intuition. Suppose *Low* had observed ε. In this moment he knows that the set of possible traces is $\{\varepsilon, h_1, h_2\}$. When he sees l_1, he deduces that h_2 could not have been possible at his previous observation.

Example 11 *Consider now the system $\mathrm{Tr}' = \mathsf{pref}\big\{h_1 l_1, h_1 l_2, h_2 l_1, h_2 l_2\big\}$, within the same framework as above. In the system Tr' we* do not have *information flow because for all $i, j = 1, 2$,*

$$\begin{aligned}\mathsf{knl}(h_i, \mathrm{Tr}') = \mathsf{pref}\big\{h_1, h_2\big\} &\subseteq \mathsf{knl}(h_i l_j, \mathrm{Tr}') \\ &= \mathsf{pref}\big\{h_1 l_j, h_2 l_j\big\}.\end{aligned}$$

Example 12 *Finally, consider $\mathrm{Tr}'' = \mathsf{pref}\big\{h_1 l, h_2 l\big\}$, with $\mathsf{Inv} = \{h_1, h_2\}$ and $\mathsf{Vis} = \{l\}$. The system Tr'' has no information flow either, since for all $i = 1, 2$,*

$$\begin{aligned}\mathsf{knl}(h_i, \mathrm{Tr}'') = \mathsf{pref}\big\{h_1, h_2\big\} &\subseteq \mathsf{knl}(h_i l, \mathrm{Tr}'') \\ &= \mathsf{pref}\big\{h_1 l, h_2 l\big\}.\end{aligned}$$

Example 13 *We can show that information flow through termination is also properly handled in our framework: consider $\mathrm{Tr}_{term} = \mathsf{pref}\big\{h_1 l, h_2\big\}$, within the same framework as in the previous example. Then*

$$\begin{aligned}\mathsf{knl}(h_1, \mathrm{Tr}_{term}) = \mathsf{pref}\big\{h_1, h_2\big\} &\not\subseteq \mathsf{knl}(h_1 l, \mathrm{Tr}_{term}) \\ &= \mathsf{pref}\big\{h_1 l\big\}\end{aligned}$$

This is in accordance with the intuition that the impossibility to continue h_2 with l must be information leak.

An alternative for Definition 9 is stated in the following:

Proposition 14 *The system represented by Tr has no information flow if and only if the following property holds:*

(DIF) *For all* $t, t' \in \mathrm{Tr}$ *and* $u \in E^*$ *with* $o(t) = o(t')$ *and* $o(u) \in \mathsf{Vis}$*, if* $tu \in \mathrm{Tr}$ *then there exists* $u' \in E^*$ *such that* $o(u') = o(u)$ *and* $t'u' \in \mathrm{Tr}$.

Proof: The fact that Definition 9 implies property (DIF) is rather straightforward, since, under the assumption that $t, tu \in \mathrm{Tr}$, the inclusion $\mathsf{knl}(t,\mathrm{Tr}) \subseteq \mathsf{knl}(tu,\mathrm{Tr})$ implies that for any $t' \in \mathrm{Tr}$ with $o(t) = o(t')$, $t' \in \mathsf{knl}(tu,\mathrm{Tr})$, which in turn is equivalent to the existence of u' such that $o(u) = o(u')$ and $t'u' \in \mathrm{Tr}$.

For the reverse implication, note that it suffices to prove that for all $w, w' \in \mathrm{Tr}$, with $w \preceq w'$, $\mathsf{LLES}(w, \mathrm{Tr}) \subseteq \mathsf{knl}(w',\mathrm{Tr})$. The case $o(w) = o(w')$ is trivial since it implies $\mathsf{knl}(w,\mathrm{Tr}) = \mathsf{knl}(w',\mathrm{Tr})$.

Suppose that $o(w') = o(w)z_1 \dots z_n$, for some $n \geq 1$ and $z_1, \dots, z_n \in \mathsf{Vis}$. Then there exist $v_1, \dots, v_n \in E^*$ such that $w' = wv_1 \dots v_n$ and $o(v_i) = z_i$, for all $1 \leq i \leq n$. Take also $w'' \in \mathsf{LLES}(w, \mathrm{Tr})$.

We may first apply the hypothesis (DIF) for $t = w$, $t' = w''$, $u = v_1$ to obtain that there exists $v'_1 \in E^*$ such that $o(v_1) = o(v'_1)$ and $w''v'_1 \in \mathrm{Tr}$. Now, for $t = wv_1$, $t' = w''v'_1$ and $u = v_2$, we will obtain that there exists $v'_2 \in E^*$ such that $o(v_2) = o(v'_2)$ and $w''v'_1v'_2 \in \mathrm{Tr}$. By induction on the remaining v_is, we get $v'_1, \dots, v'_n \in E^*$ such that $w''v'_1 \dots v'_n \in \mathrm{Tr}$ and $o(v_i) = o(v'_i)$ for all $1 \leq i \leq n$. The latter identities imply that $o(w''v'_1 \dots v'_n) = o(w')$, and hence $w''v'_1 \dots v'_n \in \mathsf{LLES}(w', \mathrm{Tr})$. As a consequence, $w'' \in \mathsf{knl}(w',\mathrm{Tr})$.

3 Relating NNI and DIF

We want to show here that the two notions are equivalent. Unfortunately, the equivalence does not come for free, by means of the inability of the model to distinguish between system choices and *High*'s choices.

To see this, consider the system $\mathrm{Tr}_{limit} = \mathsf{pref}\{hl_1, hl_2, l_2\}$, where $\mathsf{Inv} = \{h\}$ and $\mathsf{Vis} = \{l_1, l_2\}$. According to Definition 9, this system has no information flow, since $\mathsf{knl}(\varepsilon,\mathrm{Tr}) = \mathsf{pref}\{h\} \subseteq \mathsf{knl}(hl_1,\mathrm{Tr}) = \mathsf{pref}\{hl_1\}$ and $\mathsf{knl}(\varepsilon,\mathrm{Tr}) \subseteq \mathsf{knl}(l_2,\mathrm{Tr}) = \mathsf{pref}\{hl_2, l_2\}$.

On the other hand, according to Definition 8, system Tr_{limit} does not satisfy nondeterministic noninterference: consider the following two strategies: $h\bot$, resp. $\bot$, that is, $s_1(\varepsilon) = h$ and $s_1(h) = s_2(\varepsilon) = \bot$. We then have $\mathsf{Beh}(s_1) = \mathsf{pref}\{hl_1, hl_2\}$ and $\mathsf{Beh}(s_2) = \mathsf{pref}\{l_2\}$, and hence the trace $t = hl_1$ cannot be observed in the second strategy.

The point about this difference is the inability of our deduction model to separate what is *High*'s choice from what is system's choice. This separation is made possible in the definition of nondeterministic noninterference by means of specifying jointly a strategy for *High* combined with a strategy for the system.

We show here that, by slightly modifying our setting, we are able to capture such information flow. The basic idea is simple and rather straightforward: we just need to *make system choices visible*. That is, whenever the system disallows *High* to make any choice, this fact should be signaled to *Low*.

Formally, this visibility of system choices can be done by inserting $\top$ events right before the occurrence of each visible event: consider the mapping $\mathcal{T} : E \to \left(E \cup \{\top\}\right)^*$ defined by $\mathcal{T}(e) = \top e$ iff $e \in \mathsf{Vis}$ and $\mathcal{T}(e) = e$ otherwise. As usual, we may extend $\mathcal{T}$ to traces, and hence we get the transformed system $\mathcal{T}(\mathrm{Tr}) = \left\{\mathcal{T}(t) \mid t \in \mathrm{Tr}\right\}$. We will consider $\top$ an observable event, hence work with $\mathsf{Vis}' = \mathsf{Vis} \cup \{\top\}$.

We may alternatively characterize "making system choices visible" as follows:

Proposition 15 *Given system* Tr*,* $\mathcal{T}(\mathrm{Tr})$ *has no information flow if and only if* Tr *has no deducible information flow and for each* $t \in \mathrm{Tr}$ *and* $z \in \mathsf{Vis}$ *with* $tz \in \mathrm{Tr}$*, the following inclusion holds:*

$$\left\{t'z \mid t' \in \mathsf{LLES}(t, \mathrm{Tr}), \exists z' \in \mathsf{Vis} \textit{ such that } t'z' \in \mathrm{Tr}\right\} \subseteq \mathrm{Tr}$$

The first main result of the paper is then the following:

Theorem 16 Tr *satisfies nondeterministic noninterference if and only if* $\mathcal{T}(\mathrm{Tr})$ *has no deducible information flow.*

The proof relies on the following:

Lemma 17 Tr *satisfies nondeterministic noninterference if and only if* $\mathcal{T}(\mathrm{Tr})$ *satisfies nondeterministic noninterference as well.*

Proof: By straightforward modification of strategies of Tr into strategies of $\mathcal{T}(\mathrm{Tr})$ – each strategy s in Tr can be transformed into a strategy $s^\top$ of $\mathcal{T}(\mathrm{Tr})$ defined as follows:

$$\forall w \in E^*, \begin{cases} s^\top(\mathcal{T}(w)) & = s(w) \\ s^\top(\mathcal{T}(w)\top) & = \bot \end{cases}$$

Proof:[of Theorem 16] Thanks to the previous lemma, we will actually prove that $\mathcal{T}(\mathrm{Tr})$ satisfies nondeterministic noninterference if and only if $\mathcal{T}(\mathrm{Tr})$ has no deducible information flow.

Suppose $\mathcal{T}(\mathrm{Tr})$ has deducible information flow. According to Proposition 14, there exist $t_1, t_2 \in \mathcal{T}(\mathrm{Tr})$ and $u \in E^*$ with $o(t_1) = o(t_2)$, $o(u) \in \mathsf{Vis}$ and $t_1u \in \mathrm{Tr}$ such that for all $u' \in E^*$ with $o(u') = o(u)$ we have $t_2u' \notin \mathrm{Tr}$.

We will construct then two strategies s_1 and s_2 from t_1u, resp. t_2, which will contradict nondeterministic noninterference. To this end, we enumerate the events in t_1, t_2 and u, say $t_j = t_j^1 \dots t_j^{n_i}$ for some $n_1, n_2 \geq 0$ and $u = u_1 \dots u_n$. Then s_1 and s_2 are defined as follows:

$$\forall j = 1,2, \forall 1 \leq i < n_j, s_j(t_j^1 \dots t_j^i) = \begin{cases} t_j^{i+1} & \text{if } t_j^{i+1} \in \mathsf{Inv} \\ \perp & \text{otherwise} \end{cases}$$

$$\forall 0 \leq i \leq n, s_1(t_1^1 \dots t_1^{n_1} u_1 \dots u_i) = \begin{cases} u_{i+1} & \text{if } u_{i+1} \in \mathsf{Inv} \\ \perp & \text{otherwise} \end{cases}$$

The above definitions give the form for s_1 and s_2 only on their "interesting" part – on arguments that are not of the specified form, s_1 and s_2 may give any result satisfying the requirements from Definition 8.

We claim that $o(tu) \notin o(\mathsf{Beh}(s_2))$. To see this, note first that the only trace t_2' in $\mathsf{Beh}(s_2)$ that has the same observation as t_1 is t_2, by uniqueness in the definition of s_2. Therefore, if we had some $t_2' \in \mathsf{Beh}(s_2)$ with $o(t_2') = o(tu)$, then t_2' should have t_2 as a prefix.

But, by hypothesis on t_1, u, t_2, we cannot have some $t_2 u' \in \mathrm{Tr}$ with $o(t_2 u') = o(t_1 u)$, which means that no t_2' with $o(t_2') = o(t_1 u)$ may belong to $\mathsf{Beh}(s_2)$, which proves the direct implication.

For the reverse implication, consider two strategies s_1 and s_2 satisfying the negation of the property (NNI), i.e. there exists $t_1 \in \mathsf{Beh}(s_1)$ such that for any trace $t_2 \in \mathsf{Beh}(s_2)$ we have $o(t_1) \neq o(t_2)$. Without loss of generality, we will take as t_1 the least trace, w.r.t. prefix ordering, that satisfies this property. Hence, if $t_1 = ta$ for some $a \in E$, t can be observed in s_2, that is, there exists some $t' \in \mathsf{Beh}(s_2)$ such that $o(t') = o(t)$.

Remark 18 *t, a, t' is a triple that does not satisfy (DIF), and a must be visible, or otherwise $o(ta) = o(t')$ which would contradict the fact that $t_1 = ta$ has no observable equivalent in $\mathsf{Beh}(s_2)$.*

Hence, we may choose $t' \in \mathsf{Beh}(s_2)$ to be the first trace, w.r.t. prefix ordering, which satisfies $o(t') = o(t)$.

Suppose first that $a \neq \top$. Then, $t = w\top$ by construction of $\mathcal{T}(\mathrm{Tr})$. Consequently, $t' = w'\top$. Furthermore, by construction of $\mathcal{T}(\mathrm{Tr})$, for any $t'u' \in \mathcal{T}(\mathrm{Tr})$, u' must start with a visible event, that is, whenever $t'u' \in \mathcal{T}(\mathrm{Tr})$ and $o(u') = b \in \mathsf{Vis}$ we must have also that $t'b \in \mathcal{T}(Tr)$.

In particular, if $\mathcal{T}(\mathrm{Tr})$ were to satisfy (DIF), then $t'a \in \mathcal{T}(\mathrm{Tr})$. But, by rewriting the defining property (NNI) for t_1, we get that $t'a \notin \mathsf{Beh}(s_2)$. We then need to show that this also implies the stronger property that $t'a \notin \mathcal{T}(\mathrm{Tr})$.

To this end, observe that, since t' ends with $\top$, t' is always followed by a visible event, that is, there is no $h \in \mathsf{Inv}$ such that $t'h \in \mathcal{T}(\mathrm{Tr})$. Therefore we must have $s_2(t') = \perp$. This means that after t', the system forbids *High* for choosing, and makes his own choice, which must be a visible event. But the construction of $\mathsf{Beh}(s_2)$ will always include such a continuation, that is, if $t' \in \mathsf{Beh}(s_2)$, $s_2(t') = \perp$ and $t'a \in \mathcal{T}(\mathrm{Tr})$ then we should also have $t'a \in \mathsf{Beh}(s_2)$.

This means that we cannot have $t'a \in \mathcal{T}(\mathrm{Tr})$, which means that in fact the triple t, a, t' satisfies the negation of property (DIF) from Proposition 14, that is, $\mathcal{T}(\mathrm{Tr})$ has deducible information flow.

The proof will end if we show that a cannot be $\top$, a result which relies on the Remark 7:

Suppose, for the sake of contradiction, that $t_1 = t\top$, $t' \in \mathsf{Beh}(s_2)$ with $o(t') = o(t)$ and for any $u' \in E^*$ with $o(u) = \top$ we have $t'u' \notin \mathsf{Beh}(s_2)$. Recall also that we may choose t' to be the first, w.r.t. prefix ordering, that satisfies this property.

We will first prove that $s_2(t') \neq \perp$. For, assuming the contrary, we would obtain that $t'b \in \mathcal{T}(\mathrm{Tr})$ for some $b \in \mathsf{Vis}$, and, since $t'\top \notin \mathcal{T}(\mathrm{Tr})$ by assumption, we obtain that $b \neq \top$ either. But this would imply that $t' = \tau'\top$, which, on its turn, would imply that $t_1 = \tau\top\top$ for some τ with $o(\tau) = o(\tau')$, which is forbidden by construction in $\mathcal{T}(\mathrm{Tr})$. Hence, $s_2(t') \in \mathsf{Inv}$.

We construct then an infinte sequence of events $\overline{e} = \left(e_n\right)_{n \geq 1}$ which represents the decisions coded in s_2 that extend t'. That is, $e_1 = s_2(t')$ and $e_{n+1} = s_2(t'e_1 \dots e_n)$. And we may easily observe, by induction, that $e_i \in \mathsf{Inv}$, for, otherwise, considering e_i the first event equal to $\perp$, and $e_j \in \mathsf{Inv}$ for all $j < i$, we would conclude that $o(t'e_1 \dots e_{i-1}e_i) = o(t\top)$, which contradicts the assumption that $t\top$ has no equivalent observable in $\mathsf{Beh}(s_2)$.

But this means that s_2 is Zeno for t', which, by Remark 7, implies that $t' \notin \mathsf{Beh}(s_2)$, in contradiction with our choice. This ends our proof of the Theorem 16.

4 Generalized deducible information flow

In this section we generalize our notion of deducible information flow to the case in which *Low* is interested in information regarding only a *part* of *High*'s choices. We show here that information flow analysis in this generalized setting can be reduced to the formalism from Section 2.

The framework of our generalization is the following: we consider, as before, that the system under analysis is represented by a set of traces $\mathrm{Tr} \subseteq E^*$, where $E = \mathsf{Vis} \cup \mathsf{Inv}$ is the set of possible events. *Low*, the low-level user, will only be interested in making *deductions* about the high-level events from a subset $\mathsf{Conf} \subseteq \mathsf{Inv}$. We will define a *deduction mapping* $d : E \to \mathsf{Conf} \cup \mathsf{Vis} \cup \{\epsilon\}$ by $d(e) = e$ if $e \in \mathsf{Conf} \cup \mathsf{Vis}$ and $d(e) = \epsilon$, otherwise. We extend d to sequences of events, as usual, by concatenating the result of d applied to each event in the sequence. The fact that $d(e) = \varepsilon$ for some $e \in E$ signals that the occurrence the event e is of no importance to *Low*.

Remind that $\mathsf{LLES}(w, \mathrm{Tr})$ was supposed to model the deductions *Low* can make once he observes $o(w)$. In the current setting *Low* is no longer interested in deducing *all* information about Inv, but rather in getting information about activities in Conf. Once the system issues trace w, *Low* makes the following deductions:

$$\mathsf{DED}(w, \mathrm{Tr}) = \{d(w') \in \mathrm{Tr} \mid o(w) = o(w')\}.$$

Continuing our parallel to Section 2.2, we define *Low*'s *generalized knowledge after seeing* w, denoted by $\mathsf{genknl}(w, \mathrm{Tr})$, as the set of all prefixes of the deductions he can make after the system issues the trace w:

$$\begin{aligned}\mathsf{genknl}(w, \mathrm{Tr}) &= \{z \mid z \preceq w' \text{ for some } w' \in \mathsf{DED}(w, \mathrm{Tr})\} \\ &= \mathsf{pref}\big(\mathsf{DED}(w, \mathrm{Tr})\big)\end{aligned}$$

The definition of deducible information flow in this generalized setting is obtained by replacing in Definition 9, $\mathsf{knl}(w,\mathrm{Tr})$ with $\mathsf{genknl}(w, \mathrm{Tr})$:

Definition 19 *The system represented by* Tr ***has no generalized deducible information flow w.r.t. the deduction mapping d*** *if for all* $w, w' \in \mathrm{Tr}$ *such that* $w \preceq w'$*, we have:*

$$\mathsf{genknl}(w, \mathrm{Tr}) \subseteq \mathsf{genknl}(w', \mathrm{Tr}).$$

If there exist $w, w' \in \mathrm{Tr}$ *with* $w \preceq w'$ *such that the inclusion is not satisfied, we say that* Tr ***has generalized deducible information flow w.r.t. d****.*

Proposition 20 Tr *has no generalized deducible information flow w.r.t.* d *if and only if* $d(\mathrm{Tr}) = \{d(t) | t \in \mathrm{Tr}\}$ *has no deducible information flow.*

Proof: Straightforward, since in $d(\mathrm{Tr})$, *Low*'s knowledge for $\alpha = d(t)$, $t \in \mathrm{Tr}$, is

$$\begin{aligned}\mathsf{knl}(\alpha, d(\mathrm{Tr})) &= \{\beta \in d(\mathrm{Tr}) \mid o(\alpha) = o(\beta)\} \\ &= \{d(t') \mid t' \in \mathrm{Tr} \text{ and } o(d(t)) = o(d(t'))\} \\ &= \mathsf{genknl}(t, \mathrm{Tr})\end{aligned}$$

Proposition 14 can then be generalized as follows:

Proposition 21 Tr *has no generalized deducible information flow if and only if the following property holds:*

(GDIF) *For all* $t, t' \in \mathrm{Tr}$ *and* $u \in E^*$ *with* $o(t) = o(t')$*,* $o(u) \in \mathsf{Vis}$*, if* $tu \in \mathrm{Tr}$ *there exists* $t'' \in \mathrm{Tr}$ *and* $u' \in E^*$ *such that* $o(u') = o(u)$*,* $d(t') = d(t'')$ *and* $t''u' \in \mathrm{Tr}$*.*

5 Relationship with other definitions of information flow

In this section we show how generalized deducible information flow can be related with other well-known information flow properties from the literature. Throughout this section we suppose, as in the previous section, that $E = \mathsf{Vis} \cup \mathsf{Inv}$, $\mathsf{Conf} \subseteq \mathsf{Inv}$ and d is the deduction mapping.

Relationship with Perfect Security Property (PSP) and Separability.

The *Perfect Security Property* from [15] is defined for $\mathsf{Conf} = \mathsf{Inv}$. As we noticed earlier, in this context, generalized deducible information flow is exactly deducible information flow. We will prove that PSP implies deducible information flow.

We define $c : E \to \mathsf{Conf} \cup \{\varepsilon\}$ by $c(e) = e$ if $e \in \mathsf{Conf}$ and $c(e) = \epsilon$, otherwise. We extend c to handle sequences of events in the usual way.

Definition 22 ([15]) *We say that a system* Tr *satisfies the* ***Perfect Security Property*** *(PSP) if*

1. *For all* $\tau \in \mathrm{Tr}$*,* $o(\tau) \in \mathrm{Tr}$*.*
2. *For all* $t_1,\ t_2 \in \mathrm{Tr}$ *and all* $h \in \mathsf{Conf}$*, if* $t_1t_2 \in \mathrm{Tr}$*,* $c(t_2) = \epsilon$ *and* $t_1h \in \mathrm{Tr}$ *then* $t_1ht_2 \in \mathrm{Tr}$*.*

Example 23 *Consider again the system* $\mathrm{Tr}'' = \mathsf{pref}\{h_1l, h_2l\}$ *from Example 12, with* $\mathsf{Vis} = \{l\}$ *and* $\mathsf{Conf} = \mathsf{Inv} = \{h_1, h_2\}$*. Remind that the system* Tr'' *has no deducible information flow. But clearly* Tr'' *does not satisfy PSP since* $o(h_1l) \notin \mathrm{Tr}$*.*

The next result, combined with the above example, shows that our notion of deducible information flow is strictly weaker than PSP.

Proposition 24 *A system that satisfies PSP has no deducible information flow.*

Remark 25 *Since* separability *([12]) implies PSP, we also obtain that a system satisfying separability has no deducible information flow.*

Relationship with Generalized Noninterference (GNI) and Forward Correctability.

GNI [11] was originally defined in terms of high-level and low-level inputs and outputs but we can also define it in terms of confidential events, invisible and non-confidential events, and visible events, as in the MAKS framework [10].

Definition 26 *We say that a system defined by a set of traces* Tr ***satisfies Generalized Noninterference*** *(GNI), if for all* $t_v \in \mathrm{Tr}$*,* $t_c \in \mathsf{Conf}^*$ *and* $\tau \in E^*$*:*

$$\tau \in interleaving(t_c, o(t_v)) \Rightarrow \exists t \in \mathrm{Tr} \text{ such that } d(t) = \tau.$$

Example 27 *The system from Example 12 satisfies generalized deducible information flow but does not satisfy generalized noninterference. The system under discussion is* $\mathrm{Tr}'' = \mathsf{pref}\{h_1l, h_2l\}$*, with* $\mathsf{Vis} = \{l\}$ *and* $\mathsf{Conf} = \mathsf{Inv} =$

$\{h_1, h_2\}$. *Remind that the system* Tr'' *has no generalized deducible information flow w.r.t.* d *(since it has no deducible information flow). But* Tr'' *does not satisfy GNI since we may put* $t_v = h_1 l$, $t_c = \varepsilon$ *and for the only element of* $interleaving(t_c, o(t_v)) = \{l\}$ $d^{-1}(l) = \{l\}$ *but* $l \notin \mathrm{Tr}$.

Proposition 28 *If the system* Tr *satisfies GNI then, it has no generalized deducible information flow w.r.t.* d.

Remark 29 *Since* forward correctability *([8, 10]) implies GNI, we obtain that it also implies generalized deducible information flow w.r.t.* d *as above.*

6 A finite state model and its decidablility

Throughout this section we assume that the system is represented by a *finite automaton* $\mathcal{A} = (Q, E, \delta, q_0)$. There is no final state since the language is prefix-closed, hence $\mathrm{Tr} = L(\mathcal{A})$ is the set of traces which do not block the automaton. We also assume that all the states of the automaton are *reachable*, that is, for each $q \in Q$ there exists a trace $t \in E^*$ (in fact, $t \in \mathrm{Tr}$!) that labels a path from q_0 to q.

In this setting, we are interested in solving the following decision problem:

Problem[Finite-state information flow problem] Given an automaton $\mathcal{A}$, does $L(\mathcal{A})$ have information flow?

Theorem 30 *The finite state information flow problem is decidable.*

Proof: We will consider here *deterministic* automata, that is $\delta : Q \times E \to Q \cup \{\uparrow\}$. The uparrow denotes that $\delta(q, a)$ is undefined - i.e. if the automaton receives the event a while in state q, it is blocked forever.

We extend the transition function to words, that is, we build $\bar{\delta} : (Q \cup \{\uparrow\}) \times E^* \to Q \cup \{\uparrow\}$ by structural induction: for all $q \in Q$, $w \in E^*$, $a \in E$ $\bar{\delta}(\uparrow, w) = \uparrow$, $\bar{\delta}(q, \epsilon) = q$ and $\bar{\delta}(q, wa) = \delta(\bar{\delta}(q, w), a)$.

We define the *observable transition relation* $\rho \subseteq (Q \cup \{\uparrow\}) \times Vis \times (Q \cup \{\uparrow\})$ and the *invisible transition relation* $\psi \subseteq Q \times Q$ as follows:

$$\rho = \big\{(q, a, r) \mid q, r \in Q \cup \{\uparrow\}, a \in \mathsf{Vis}, \exists t \in E^*, o(t) = a \text{ and } \bar{\delta}(q, t) = r\big\}$$

$$\psi = \Big\{(q, r) \mid q, r \in Q, \exists t \in E^*, o(t) = \epsilon \text{ and } \bar{\delta}(q, t) = r\Big\}$$

Note that both ρ and ψ can be effectively constructed.

To complete the proof, we construct the nondeterminstic automaton $\mathcal{B} = (Q \times Q \cup \{f\}, Vis, (q_0, q_0), \theta, \{f\})$ where $f \notin Q$ is the only final state and

$$\theta = \big\{(q_1, q_2, a, r_1, r_2) \mid (q_1, a, r_1), (q_2, a, r_2) \in \rho\big\} \cup \big\{(q_1, q_2, a, f) \mid (q_1, a, \uparrow) \notin \rho, (q_2, a, \uparrow) \in \rho\big\}$$

Then, $L(\mathcal{A})$ does not have information flow iff $L(\mathcal{B})$ is empty. Furthermore, if each word $w \in L(\mathcal{B})$ corresponds to a sequence of observations that give deducible information flow i.e. there exist $t, t' \in L(\mathcal{A})$ and $u, u' \in E^*$ such that $o(u) = o(u') = a$, $o(t) = o(t') = w$, $tu \in L(\mathcal{A})$ and $t'u' \notin L(\mathcal{A})$.

Note that the size of $\mathcal{B}$ is polynomial in the size of the original automaton.

7 Conclusions

We have presented a framework for trace-based possibilistic information flow which starts from the principle that *high-level choices should not be deduced by low-level users*. Our framework labels less systems as having information flow, than other properties like Perfect Security Property, Separability, Generalized Noninterference or Forward Correctability. The above enumerated properties are sometimes too strong, as they do not allow auditing, which is a "legal" type of information flow from low to high-level [15, 9]. Our framework avoids this by emphasizing on strategies as intensional behaviors of agents, rather than traces.

We have shown that our model is sufficiently powerful to express only "partial information flow", that is, the situation in which the low-level user is not interested in achieving information about all the high-level activity, but only in a part of it. And we have also designed, in the finite-state case, an exponential-space algorithm for deciding whether a system has information flow.

An interesting question is whether the decidability result holds also for infinite-state models of systems, like pushdown automata.

References

[1] D. Beauquier, M. Duflot, and M. Minea. A probabilistic property-specific approach to information flow. In *MMM-ACNS*, volume 3685 of *Lecture Notes in Computer Science*, pages 206–220, 2005.

[2] B. Bérard and C. Picaronny. Accepting zeno words: a way toward timed refinements. *Acta Informatica*, 37:45–81, 2000.

[3] J.-Fr. Bergeretti and B. Carré. Information-flow and data-flow analysis of while-programs. *ACM Transactions on Programming Language Systems*, 7:37–61, 1985.

[4] R. Focardi and R. Gorrieri. A taxonomy of security properties for process algebras. *Journal of Computer Security*, 3(1):5–34, 1995.

[5] R. Focardi, R. Gorrieri, and F. Martinelli. Real-time information flow analysi. *IEEE Journal on Selected Areas in Communications*, 21:20–35, 2003.

[6] J.A. Goguen and J. Meseguer. Security policies and security models. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 11–20, Oakland, CA, USA, 1982.

[7] J. Halpern. *Reasoning about knowledge*. MIT Press, 2003.

[8] D.M. Johnson and F.J. Thayer. Security and the composition of machines. In *Proceedings of the Computer Security Foundations Workshop*, pages 72–89, Franconia, NH, USA, 1988.

[9] H. Mantel. Possibilistic definitions of security - an assembly kit. In *Proceedings of the IEEE Computer Security Foundations Workshop*, pages 185–199, Cambridge, UK, 2000.

[10] H. Mantel. *A Uniform Framework for the Formal Specification and Verification of Information Flow Security*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 2003.

[11] D. McCullough. Specifications for multi-level security and a hook-up property. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 161–166, Oakland, CA, USA, 1987.

[12] J. McLean. A general theory of composition for trace sets closed under selective interleaving functions. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 79–93, Oakland, CA, USA, 1994.

[13] A. Sabelfeld and A. C. Myers. Language-based information-flow security. *IEEE J. Selected Areas in Communications*, 21:5–19, January 2003.

[14] J. Todd Wittbold and Dale M. Johnson. Information flow in nondeterministic systems. In *IEEE Symposium on Security and Privacy*, pages 144–161, 1990.

[15] A. Zakinthinos and E.S. Lee. A general theory of security properties. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 94–102, Oakland, CA, USA, 1997.