The Framework of Selective Interleaving Functions and the Modular Assembly Kit

Heiko Mantel*
RWTH Aachen University, Germany
mantel@cs.rwth-aachen.de

ABSTRACT

The Framework of Selective Interleaving Functions and the Modular Assembly Kit for Security Properties both provide a basis for the uniform representation and formal analysis of noninterference-like properties. In this article, we clarify the relationship between these two frameworks. Our main result is that each property that can be represented in the Framework of Selective Interleaving Functions can also be represented in the assembly kit. In fact, the latter framework is strictly more expressive, which we demonstrate by several example properties.

Categories and Subject Descriptors

D.2.4 [**Software Engineering**]: Software/Program Verification–Formal Methods; F.3.1 [**Logics and Meanings of Programs**]: Specifying and Verifying and Reasoning about Programs–Specification techniques

General Terms

Security, verification

Keywords

Formal specification and analysis of security properties, information flow security, noninterference

1. INTRODUCTION

Noninterference [6] provides a basis for formally analyzing the security of information systems. Numerous variants of the original noninterference property were proposed, often with the aim to better address concurrent and distributed systems. Generalized noninterference and restrictiveness [18], noninference [22], and separability [19] are prominent examples of such properties, and the number of variants is still growing (see, e.g., [9, 30]). This rich body of work gives one much flexibility in the choice of a security property and of a corresponding analysis technique when investigating the security of a given system. However, unless the relationship between the different properties is sufficiently clear, this variety makes it also difficult to determine the property that is most appropriate for analyzing a given system. Therefore, the comparison of noninterference-like properties has been a focal research topic.

For simplifying such analyzes and comparisons, several frameworks were developed. McLean proposed the Framework of Selective Interleaving Functions [19], Focardi and Gorrieri introduced the process algebra SPA [3], Zakinthinos and Lee put forward Low-Level Equivalence Sets [33], Ryan and Schneider used common notions of process equivalence [27], and the author developed the Modular Assembly Kit for Security Properties [11]. The detailed analysis of noninterference-like properties with these frameworks deepened the community's understanding of the existing properties and also inspired the definition of novel properties. The resulting achievements include taxonomies of noninterference-like properties, verification techniques (unwinding theorems) as well as compositionality results.

McLean's Framework of Selective Interleaving Functions (abbreviated by *FSIF*) provides schemata for representing noninterference-like properties that are restrictive enough to investigate entire classes of properties at once. This overcomes the need for investigating similar properties individually when deriving compositionality results and in comparisons to other properties. McLean's work has been quite influential in the information-flow-security community, and it is unfortunate that most subsequently proposed frameworks do not support the derivation of parametric compositionality results. It was the author's main motivation for developing the Modular Assembly Kit for Security Properties (abbreviated by *MAKS*) to overcome the limited expressiveness of the *FSIF* without giving up the ability of investigating entire classes of properties at once. While the latter goal was clearly achieved (see, e.g., [12, 14]), a rigorous investigation of the former aspect has remained an open issue. The current article resolves this issue by clarifying the relationship between the *FSIF* and the *MAKS* with respect to expressiveness. This formal comparison turned out to be more involved than initially expected, which is due to the different flavor of the basic concepts in the two frameworks.

For enabling a formal comparison between the two frameworks, we had to develop an event-based variant of the *FSIF*. In summary, our main contributions are: (1) the definition

*Most of this work has been performed while being a member of the Information Security Group at the ETH Zurich. The author gratefully acknowledges support by the DFG.

FMSE'05, November 11, 2005, Fairfax, Virginia, USA.

of a variant of the *FSIF* for the model of event systems, (2) a theorem showing that all properties that can be represented in the *FSIF* can also be represented in the *MAKS*, and (3) the observation that some noninterference-like properties can be represented in the *MAKS* but not in the *FSIF*.

We argue in detail for the adequacy of our variant of the *FSIF* and explain why it captures the original framework more adequately than a previously proposed variant [32]. Our variant incorporates formal definitions of range and domain restrictions, which goes beyond [19, 20] where these notions were only introduced by example. We illustrate the limited expressiveness of the *FSIF* with several example properties and, based on these investigations, point out possible directions for improving its expressiveness.

2. PRELIMINARIES

2.1 Event Systems

We employ a trace-based system model throughout this article. A *trace* is a sequence of events that models one possible execution sequence of a given system. An *event* is a term modeling an atomic action like, e.g., sending or receiving a message. We use $\langle\rangle$ to denote the empty trace and separate the occurrences of events in a trace by commas (e.g., $\langle send(m, c), rcv(m, c')\rangle$ can be read as: a message m is sent on channel c and then forwarded on channel c').

For a given system, we distinguish between input, output, and internal events. The underlying intuition is that input events are controlled by the environment while output and internal events are controlled by the system. We do *not* make the assumption that input events are always enabled. This makes our observations applicable to systems that are input total as well as to systems that are not input total.

In summary, an *event system* [18] *ES* is a tuple (E, I, O, Tr) where E is a *set of events*, $I, O \subseteq E$, respectively, are disjoint *sets of input and output events*, and $Tr \subseteq E^*$ is the *set of possible traces*, i.e. a set of finite sequences over E. Each trace $\tau \in Tr$ models a possible behavior of *ES*. The set Tr must be closed under prefixes, i.e. any prefix of a trace in Tr must also be in Tr. Event systems allow for the specification of nondeterministic systems where nondeterminism is reflected by the choice between the different events that are enabled. The model of event systems is a possibilistic system model that leaves it unspecified how nondeterministic choices are resolved.

In this article, we adopt the notation used above. That is, *ES* denotes an event system (E, I, O, Tr). The *projection* $t|_{E'}$ of a trace $t \in E^*$ to a set $E' \subseteq E$ results from t by removing all events that are *not* in E'. Given two traces $t_1 \in E_1^*$ and $t_2 \in E_2^*$ over two disjoint sets of events (i.e. $E_1 \cap E_2 = \emptyset$), an *interleaving* of t_1 and t_2 is a trace $t \in (E_1 \cup E_2)^*$ with $t|_{E_1} = t_1$ and $t|_{E_2} = t_2$. We denote the set of all interleavings of t_1 and t_2 by $interleaving(t_1, t_2)$.

2.2 Noninterference-Like Properties

Following the definition of noninterference by Goguen and Meseguer [6], various other noninterference-like properties have been proposed (see [15] for an overview). In this section, we recall the four properties that McLean represented in his framework [19, 20]. We assume a two-level flow policy that forbids information flow from a high level H to a low level L, which is the simplest setting in which the problem of information flow security can be studied. Moreover, we assume a function $dom : E \to \{L, H\}$ that associates each event with one of these security domains and use H and L to denote the set of high-level events and the set of low-level events, respectively. Further properties are discussed in Section 5 where we investigate the limitations of the *FSIF*.

Separability [19, 33] captures noninterference in a very intuitive way. Satisfying separability means for a system that it could have been built from two disconnected components, a high-level component and a low-level component, without any communication lines in between. As the two components cannot communicate with each other, it is intuitively clear that there is no danger of information leaking from the high level to the low level. The requirement is somewhat restrictive though, because information cannot flow from the low-level component to the high-level component either, although such information flow is usually unproblematic.

If one views $\{\tau|_H \mid \tau \in Tr\}$ and $\{\tau|_L \mid \tau \in Tr\}$ as the sets of possible traces of a low-level component and a high-level component, respectively, then the requirement that there is no communication between two parallel components is equivalent to the requirement that all interleavings of possible traces of the two components are contained in Tr.

Definition 1. *An event system ES satisfies* separability *(denoted SEP(ES)) if and only if the following condition holds:* $\forall \tau_l, \tau_h \in Tr : \forall \tau \in interleaving(\tau_h|_H, \tau_l|_L) : \tau \in Tr$.

Generalized noninterference [18] was one of the first variants of noninterference for nondeterministic systems. This property requires that for all perturbations of the occurrences of high-level input events in a possible trace it must be possible to modify the occurrences of internal high-level events and high-level output events such that the result is again a possible trace. There are several variants of generalized noninterference whose definitions differ in which perturbations of occurrences of high-level input events must be considered and where occurrences of internal high-level events and high-level output events may be corrected (see [15] for an overview). The variant that we consider here, perturbs the given trace τ_l by modifying occurrences of high-level input events such that the resulting sequence of high-level input events corresponds to the one in some other possible trace τ_{hi}, and it permits corrections at arbitrary places.

Definition 2. *ES satisfies* interleaving-based generalized noninterference *(denoted IBGNI***(ES)) if and only if*

$$\forall \tau_l, \tau_{hi} \in Tr : \forall t \in interleaving(\tau_{hi}|_{H\cap I}, \tau_l|_L) : \exists \tau \in Tr : \tau|_{(H\cap I)\cup L} = t.$$

Noninference ensures that low-level users cannot deduce that progress has been made in the high-level computation. That is, for every observation that a low-level user can possibly make, there must be a possible trace that yields the same observation and in which no high-level events occur. The variant of noninference usually used today [19, 33] has evolved from three different information flow properties, namely Jacob's *ignorance of progress* [10], O'Halloran's *noninference* [22], and O'Halloran's *weak ignorance of progress* [22].

Definition 3. *A system ES satisfies* noninference *(denoted NF(ES)) if and only if the following condition holds:* $\forall \tau \in Tr : \tau|_L \in Tr$.

Generalized noninference [19] is a weak variant of noninference, which ensures that a low-level user cannot deduce that high-level input has been received.

DEFINITION 4. *A system ES satisfies* generalized noninference *(denoted GNF(ES)) if and only if the following condition holds:* $\forall \tau \in Tr: \exists \tau' \in Tr: [\tau'|_{H \cap I} = \langle\rangle \wedge \tau'|_L = \tau|_L]$.

For more detailed discussions of these and many other noninterference-like properties, we refer to [15, 11, 20].

3. THE FSIF

McLean proposed the *FSIF* for simplifying the comparison of noninterference-like properties and for investigating the preservation of these properties under composition. A selective interleaving function (abbreviated *sif*) is a function that takes two traces as arguments and returns a third trace. Each *sif* belongs to one or more types that prescribe how the returned trace depends on the argument traces. A noninterference-like property is expressed in the *FSIF* by the requirement that the set of traces must be closed under *sifs* of a particular type. This allows one to compare properties by comparing the types used in their representation. Types are the key concept for representing and analyzing properties in the *FSIF*.

3.1 Representing Properties

The *FSIF* was originally developed for a state-based system model. For making a formal comparison with the *MAKS* possible, we had to migrate the *FSIF* to the system model on which the *MAKS* is based. In the following, we introduce our event-based variant of the *FSIF* and argue why it properly reflects the original framework.

DEFINITION 5. *Let* $\kappa : E \to \{0,1,2\}$ *be a function. A function* $f : (E^* \times E^*) \to E^*$ *is a* selective interleaving function *of type* F_κ *if and only if*

$$\forall t, t_1, t_2 \in E^*: [f(t_1,t_2)=t \Rightarrow (t|_{E_1^\kappa} = t_1|_{E_1^\kappa} \wedge t|_{E_2^\kappa} = t_2|_{E_2^\kappa})]$$

where $E_0^\kappa, E_1^\kappa, E_2^\kappa \subseteq E$ *are defined by* $E_j^\kappa = \{e \in E \mid \kappa(e) = j\}$ *for* $j \in \{0,1,2\}$.

The type imposes constraints on how *sifs* construct traces from their arguments. The result $f(t_1,t_2)$ must equal t_1 (t_2) in the occurrences of events in E_1^κ (E_2^κ) and these must have the same order as in the respective argument trace. Otherwise, there are no restrictions on the relative ordering of events $e_1, e_2 \in E$ with $\kappa(e_1) \neq \kappa(e_2)$ and occurrences of events in E_0^κ may be freely inserted.

DEFINITION 6. *ES is* closed *under a set of sifs* $\mathcal{F}$ *if and only if* $\forall f \in \mathcal{F}: \forall \tau_1, \tau_2 \in Tr: f(\tau_1,\tau_2) \in Tr$.[1]

EXAMPLE 1. *For representing separability, one uses the function* κ_{SEP} *that returns* 1 *(2) if the argument is a high-level event (low-level event). If a set of traces Tr is closed under a sif f of type* $F_{\kappa_{SEP}}$ *then, for each pair of traces* $\tau_h, \tau_l \in Tr$, *there is some interleaving of* $\tau_h|_H$ *and* $\tau_l|_L$, *namely* $f(\tau_h,\tau_l)$, *that is a possible trace in Tr. Being closed under* some *sif of type* $F_{\kappa_{SEP}}$ *is a necessary but not a sufficient requirement for satisfying separability. In order to satisfy separability, ES must be closed under the set of* all *sifs of type* $F_{\kappa_{SEP}}$ *(see Theorem 2).*

[1]For brevity, we also write that *ES* is closed under f meaning *ES* is closed under $\{f\}$.

EXAMPLE 2. *For representing interleaving-based generalized noninterference, one uses the function* κ_{IBGNI^*} *that returns* 1 *(2) if the argument is a high-level input event (low-level event) and returns* 0 *if the argument is an internal high-level event or a high-level output event. If a system is closed under all sifs of type* $F_{\kappa_{IBGNI^*}}$ *then it also satisfies IBGNI**. *The implication in the other direction does not hold in general. To see this, we define a system* $ES_2 = (E_2, I_2, O_2, Tr_2)$ *by* $I_2 = \{li\}$, $O_2 = \{ho\}$, $E_2 = I_2 \cup O_2$, *and* $Tr_2 = \{\langle\rangle, \langle li\rangle, \langle ho\rangle\}$. *We consider the two-level security policy with* $dom(li) = L$ *and* $dom(ho) = H$. *As there are no high-level input events, IBGNI*$^*(ES_2)$ *holds trivially. However,* ES_2 *is not closed under the sif* $f_2 : (E_2^* \times E_2^*) \to E_2^*$ *of type* $F_{\kappa_{IBGNI^*}}$ *that is defined in the table below because* $\langle ho\rangle, \langle li\rangle \in Tr_2$, *but* $f_2(\langle ho\rangle, \langle li\rangle) = \langle ho, li\rangle \notin Tr_2$.

f_2	$\langle\rangle$	$\langle ho\rangle$	$\langle li\rangle$
$\langle\rangle$	$\langle\rangle$	$\langle\rangle$	$\langle li\rangle$
$\langle ho\rangle$	$\langle\rangle$	$\langle\rangle$	$\langle ho, li\rangle$
$\langle li\rangle$	$\langle li\rangle$	$\langle ho, li\rangle$	$\langle li\rangle$

While Example 1 illustrates that it does not suffice to require *Tr* to be closed under a single *sif*, Example 2 shows that, if E_0^κ is not empty then being closed under *all sifs* of a particular type is a too restrictive requirement. We introduce the notion of a covering set to soften the latter requirement.

DEFINITION 7. *A set* $\mathcal{F}$ *of sifs of type* F_κ covers type F_κ[2] *if and only if*

$$\forall t_1, t_2 \in E^*: \forall t \in interleaving(t_1|_{E_1^\kappa}, t_2|_{E_2^\kappa}): \exists f \in \mathcal{F}: f(t_1,t_2)|_{E_1^\kappa \cup E_2^\kappa} = t\ .$$

THEOREM 1. *The set of all sifs of type* F_κ *covers* F_κ.

PROOF. Let E, $\kappa : E \to \{0,1,2\}$, $t_1, t_2 \in E^*$, and $t \in interleaving(t_1|_{E_1^\kappa}, t_2|_{E_2^\kappa})$ be arbitrary. Take an arbitrary *sif* g of type F_κ and define $f : E^* \times E^* \to E^*$ by $f(t_1', t_2') = g(t_1', t_2')$ for $t_1', t_2' \in E^*$ if $t_1' \neq t_1$ or $t_2' \neq t_2$ and by $f(t_1,t_2) = t$. Then f is a *sif* of type F_κ with $f(t_1,t_2)|_{E_1^\kappa \cup E_2^\kappa} = t$. □

The following schema can now be used for uniformly representing noninterference-like properties:

ES is closed under a covering set $\mathcal{F}$ *of sifs of type* F_κ. (1)

The schema is parametric in the event system *ES* and in the type F_κ. For representing a given property, one only needs to define an appropriate type. We illustrate this for separability and interleaving-based generalized noninterference.

THEOREM 2. *SEP(ES) holds if and only if ES is closed under some covering set* $\mathcal{F}$ *of sifs of type* $F_{\kappa_{SEP}}$ *where* κ_{SEP} *is defined like in Example 1.*

THEOREM 3. *IBGNI***(ES) if and only if ES is closed under some covering set* $\mathcal{F}$ *of sifs of type* $F_{\kappa_{IBGNI^*}}$ *where* κ_{IBGNI^*} *is defined like in Example 2.*

Theorems 2 and 3 are proved in Section A.1 with the help of our main result.

Unfortunately, Schema (1) is not expressive enough for capturing noninference or generalized noninference. As a

[2]For brevity, we say $\mathcal{F}$ is a *covering set of sifs of type* F_κ meaning $\mathcal{F}$ is a set of *sifs* of type F_κ that covers F_κ.

solution, McLean points out two alternatives for modifying the schema, namely (a) restricting the range of *sifs* in the set $\mathcal{F}$ and (b) restricting the domain of *sifs* in the set $\mathcal{F}$. Unfortunately, he introduces these concepts only by example without giving formal definitions. We provide formal definitions, which are in accordance with McLean's examples, and argue that, while each alternative offers a solution for representing noninference and generalized noninference, the second alternative is preferable to the first.

3.2 Range Restrictions

The following two theorems illustrate how noninference and generalized noninference can be represented by restricting the range of *sifs*. The proofs are in Section A.1.

THEOREM 4. *Let $\kappa_{NF} : E \to \{0,1,2\}$ be the function that returns 0 (2) if the argument is a high-level event (low-level event). NF(ES) holds if and only if ES is closed under some covering set $\mathcal{F}$ of sifs of type $F_{\kappa_{NF}}$ where $\forall f \in \mathcal{F} : \forall t_1, t_2 \in E^* : f(t_1,t_2)|_{E_0^{\kappa_{NF}}} = \langle\rangle$.*

THEOREM 5. *GNF(ES) holds if and only if ES is closed under some covering set $\mathcal{F}$ of sifs of type $F_{\kappa_{NF}}$ where $\forall f \in \mathcal{F} : \forall t_1, t_2 \in E^* : f(t_1,t_2)|_{E_0^{\kappa_{NF} \cap I}} = \langle\rangle$.*

For representing noninference and generalized noninference, Theorems 4 and 5 restrict the range of *sifs* in $\mathcal{F}$. That is, Schema (1) is *strengthened* by an additional requirement. This approach leads to a rather weak schematic part in the representation of properties. In fact, the schematic part for representing noninference and generalized noninference in Theorems 4 and 5 is trivially satisfied.

THEOREM 6. *If $\kappa : E \to \{0,1,2\}$ is a function with $\forall e \in E : \kappa(e) \neq 1$ or $\forall e \in E : \kappa(e) \neq 2$ then ES is closed under some covering set of sifs of type F_κ.*

PROOF. Follows from Definitions 6 and 7. □

It is a direct consequence of Theorem 6 that the type $F_{\kappa_{NF}}$ is not very helpful in the analysis of these properties. Fortunately, the use of this type can be avoided in the representation by using domain restrictions instead of range restrictions.

3.3 Domain Restrictions

We restrict the domain of the first universal quantifier in the coverage requirement.

DEFINITION 8. *A set $\mathcal{F}$ of sifs of type F_κ* covers type F_κ under the domain restriction E_i *($E_i \subseteq E$) if and only if*

$$\forall t_1 \in (E \setminus E_i)^* : \forall t_2 \in E^* : \forall t \in \mathit{interleaving}(t_1|_{E_1^\kappa}, t_2|_{E_2^\kappa}) : \exists f \in \mathcal{F} : f(t_1,t_2)|_{E_1^\kappa \cup E_2^\kappa} = t \ .$$

This leads to the following *relaxation* of Schema (1):

ES is closed under a set $\mathcal{F}$ of sifs of type F_κ that covers F_κ under the domain restriction E_i. (2)

For representing a given property with this schema, one needs to define a type and a domain restriction. The following two theorems illustrate this for noninference and generalized noninference. The proofs are in Section A.1.

THEOREM 7. *NF(ES) holds if and only if ES is closed under some set $\mathcal{F}$ of sifs of type $F_{\kappa_{SEP}}$ that covers F_κ under the domain restriction E.*

THEOREM 8. *GNF(ES) holds if and only if ES is closed under some set $\mathcal{F}$ of sifs of type $F_{\kappa_{IBGNI^*}}$ that covers F_κ under the domain restriction E.*

The representation of noninference and generalized noninference with domain restrictions involves more restrictive, i.e. more meaningful, types than the representation with range restrictions in Theorems 4 and 5, respectively. The more restrictive types immediately reveal that *SEP(ES)* implies *NF(ES)* (cf. Theorems 2, 7, and 9) and that *IBGNI*(ES)* implies *GNF(ES)* (cf. Theorems 3, 8, and 9). This was not so obvious from the representation with range restrictions.

Observe also that the same domain restriction, i.e. E, is used in Theorems 7 and 8. We presume that this is the only nontrivial domain restriction foreseen in the *FSIF*, which is in-line with the examples given in [19, 20]. Having no domain restriction is equivalent to the trivial domain restriction $\emptyset$ (see Theorem 9 below). This observation allows one to also represent separability and interleaving-based generalized noninterference with Schema (2).

THEOREM 9. *A set $\mathcal{F}$ of sifs of type F_κ covers F_κ if and only if it covers F_κ under the domain restriction $\emptyset$.*

PROOF. Follows from Definitions 7 and 8. □

This leads to the following final schema for representing properties in the *FSIF*:

ES is closed under a set $\mathcal{F}$ of sifs of type F_κ that covers F_κ under the domain restriction E_i where $E_i \in \{\emptyset, E\}$. (3)

Permitting other domain restrictions than $\emptyset$ and E is one possibility for increasing the expressiveness of the *FSIF*. We discuss further possibilities in Section 5.4.

3.4 A Prior Variant of the FSIF

In his thesis [32], Zakinthinos proposes a variant of the *FSIF* that substantially differs from the one proposed here. The main difference in the closure requirement is that Zakinthinos requires a set of traces to be closed under some *sif* of the given type (rather than under a set of *sifs*). However, being closed under a single *sif* of type κ_{SEP} does not necessarily imply that separability is satisfied by the given system (cf. Example 1).[3] Separability requires all possible interleavings of a high-level trace with a low-level trace to be possible traces. This can be expressed by requiring closure under a sufficiently large set of *sifs* where each *sif* is responsible for the construction of a particular interleaving of two given traces. Our coverage requirement ensures that for each interleaving of a high-level trace with a low-level trace, the set contains a *sif* that constructs this interleaving. Requiring all interleavings to be possible is in accordance with Guttman and Nadel's earlier observation that a security property needs to prevent attackers not only from obtaining information about whether confidential events have occurred or not, but also about the order in which events have occurred [8].

Another problem results from the definition of *sifs* in [32]. Rather than demanding $f(t_1,t_2)|_{E_1^\kappa} = t_1|_{E_1^\kappa}$, Zakinthinos requires that $f(t_1,t_2)|_{E_1^\kappa \cap I} = t_1|_{E_1^\kappa \cap I}$ and $f(t_1,t_2)|_{E_1^\kappa \cap O} = t_1|_{E_1^\kappa \cap O}$ hold. However, these two conditions do not imply $f(t_1,t_2)|_{E_1^\kappa} = t_1|_{E_1^\kappa}$ and, hence, do not ensure that input and output events in E_1^κ occur in the right order in $f(t_1,t_2)$.

[3] The representation of separability in [32] is not adequate.

$$
\begin{aligned}
R_{\mathcal{V}}(Tr) &\equiv \forall \tau \in E^* : (\tau \in Tr \Rightarrow \exists \tau' \in E^* : (\tau' \in Tr \wedge \tau'|_C = \langle\rangle \wedge \tau'|_V = \tau|_V)\\
I_{\mathcal{V}}(Tr) &\equiv \forall \alpha, \beta \in E^* : \forall c \in C :\\
&\quad ((\beta.\alpha \in Tr \wedge \alpha|_C = \langle\rangle)\\
&\quad \Rightarrow \exists \alpha', \beta' \in E^* : (\beta'.\langle c\rangle.\alpha' \in Tr \wedge \beta'|_{V\cup C} = \beta|_{V\cup C} \wedge \alpha'|_{V\cup C} = \alpha|_{V\cup C}))\\
IA^{\rho}_{\mathcal{V}}(Tr) &\equiv \forall \alpha, \beta \in E^* : \forall c \in C :\\
&\quad ((\beta.\alpha \in Tr \wedge \alpha|_C = \langle\rangle \wedge Adm^{\rho}_{\mathcal{V}}(Tr, \beta, c))\\
&\quad \Rightarrow \exists \alpha', \beta' \in E^* : (\beta'.\langle c\rangle.\alpha' \in Tr \wedge \beta'|_{V\cup C} = \beta|_{V\cup C} \wedge \alpha'|_{V\cup C} = \alpha|_{V\cup C}))
\end{aligned}
$$

$$
\text{where } Adm^{\rho}_{\mathcal{V}}(Tr, \beta, e) \equiv \exists \gamma \in E^* : (\gamma.\langle e\rangle \in Tr \wedge \gamma|_{\rho(\mathcal{V})} = \beta|_{\rho(\mathcal{V})}).
$$

Figure 1: Formal definitions of the BSPs *R*, *I*, and IA^{ρ} (see Definition 10)

4. COMPARISON TO THE ASSEMBLY KIT

4.1 The MAKS

The Modular Assembly Kit for Security Properties supports a modular representation of noninterference-like properties. The representation of a given property consists of two elements, a set $\mathcal{VS}$ of views, defining a security policy, and a security predicate *SP*, giving a definition of secure information flow that is parametric in a view. A security predicate is, again, defined in a modular fashion by composing one or more basic security predicates (abbreviated BSP). A collection of predefined BSPs exists, where each BSP imposes rather primitive restrictions on the information flow. The schema for representing properties in the *MAKS* is:

$$
\text{BSP}^1_{\mathcal{V}}(Tr) \wedge \ldots \wedge \text{BSP}^n_{\mathcal{V}}(Tr) \text{ holds for each view } \mathcal{V} \in \mathcal{VS} \quad (4)
$$

where $\text{BSP}^1, \ldots, \text{BSP}^n$ are the BSPs from which *SP* is assembled.

Before illustrating how the security properties from Section 2.2 can be represented with this schema, we introduce the basic concepts of the *MAKS* to the extent necessary.

A *view* defines the secrets and the observational capabilities of the attacker. This is achieved by identifying the set of all events that introduce secrets into the system and the set of all events whose occurrences are visible to the attacker. As a convention, we denote the set of *confidential* events by C and the set of *visible* events by V (possibly with sub-/superscripts and primes). The intersection of C and V must be empty because, otherwise, there would be an immediate security breach. However, there may be events whose occurrences are neither confidential nor visible. We denote this set by N (for *non*-confidential/*non*-visible).

DEFINITION 9. *A* view $\mathcal{V} = (V, N, C)$ *in* E *is a triple such that* V, N, C *forms a disjoint partition of* E.

A *basic security predicate* BSP is a primitive closure condition on sets of traces that is parametric in a view. The basic idea of possibilistic information flow security is that if the set of traces is closed then, for any given observation of the attacker, so many traces could have possibly generated the observation that the attacker is unable to deduce secret information from his observations. Technically, the closure condition is defined based on two constructions, a *perturbation* and a set of permissible *corrections*, each being a transformation on traces. The closure condition is that one must be able to correct each perturbation of each possible trace to another possible trace solely by applying permissible corrections. As a convention, neither perturbations nor corrections affect occurrences of visible events in a given trace and, hence, the original trace and the correction of the perturbed trace look the same to the attacker. Depending on the perturbation, BSPs are classified into one of two dimensions. BSPs from the first dimension perturb a trace by deleting occurrences of confidential events and thereby ensure that an attacker cannot deduce that a particular confidential event must have occurred. The reasoning is as follows: If a set of traces is closed under a BSP from the first dimension then for each observation of the attacker that can be generated by a possible trace in which a confidential event occurs, there is another possible trace that generates the same observation and in which the confidential event does not occur. BSPs from the second dimension perturb a trace by inserting occurrences of confidential events and thereby prevent an attacker from deducing that a particular confidential event cannot have occurred.

DEFINITION 10. *The BSPs R (for Removal), I (for Insertion), and* IA^{ρ} *(for Insertion of* ρ*-Admissible events) are defined in Figure 3.4.*

The BSP R perturbs a given trace by removing all occurrences of confidential events. The BSPs I and IA^{ρ} perturb a trace $\beta.\alpha$ by inserting a single occurrence of a confidential event at a position where it is not followed by other occurrences of confidential events ($\alpha|_C = \langle\rangle$). The difference between the two BSPs is that IA^{ρ} inserts a confidential event c only if it is ρ-admissible at this position, which is the case if c occurs after some possible trace γ that equals β in its projection to the set $\rho(\mathcal{V}) \subseteq E$. All three BSPs permit corrections that modify occurrences of events in N at arbitrary positions in the perturbed trace.

We are now ready to show how separability, interleaving-based generalized noninterference, noninference, and generalized noninference can be represented.

THEOREM 10 ([15]). *Define two views by* $\mathcal{H} = (L, \emptyset, H)$ *and* $\mathcal{HI} = (L, H \setminus I, H \cap I)$. *Moreover, let* ρ_C *be the function from views in* E *to subsets of* E *that is defined by* $\rho_C((V, N, C)) = C$. *Then the following equivalences are valid:*

$$
\begin{aligned}
SEP(ES) &\Leftrightarrow R_{\mathcal{H}}(Tr) \wedge IA^{\rho_C}_{\mathcal{H}}(Tr) && (5)\\
IBGNI^*(ES) &\Leftrightarrow R_{\mathcal{HI}}(Tr) \wedge IA^{\rho_C}_{\mathcal{HI}}(Tr) && (6)\\
NF(ES) &\Leftrightarrow R_{\mathcal{H}}(Tr) && (7)\\
GNF(ES) &\Leftrightarrow R_{\mathcal{HI}}(Tr) && (8)
\end{aligned}
$$

Interestingly, the representation of the four properties in the *MAKS* reveals the same facts as the representation in the *FSIF* with domain restrictions in Section 3.3: $SEP(ES) \Rightarrow NF(ES)$ and $IBGNI^*(ES) \Rightarrow GNF(ES)$ follow from the fact

$$
\begin{aligned}
BSD_{\mathcal{V}}(Tr) &\equiv \forall \alpha,\beta \in E^* : \forall c \in C : ((\beta.\langle c\rangle.\alpha \in Tr \wedge \alpha|_C = \langle\rangle) \\
&\qquad \Rightarrow \exists \alpha' \in E^* : (\beta.\alpha' \in Tr \wedge \alpha'|_{V\cup C} = \alpha|_{V\cup C})) \\
BSI_{\mathcal{V}}(Tr) &\equiv \forall \alpha,\beta \in E^* : \forall c \in C : ((\beta.\alpha \in Tr \wedge \alpha|_C = \langle\rangle) \\
&\qquad \Rightarrow \exists \alpha' \in E^* : (\beta\langle c\rangle.\alpha' \in Tr \wedge \alpha'|_{V\cup C} = \alpha|_{V\cup C})) \\
BSIA^{\rho}_{\mathcal{V}}(Tr) &\equiv \forall \alpha,\beta \in E^* : \forall c \in C : ((\beta.\alpha \in Tr \wedge \alpha|_C = \langle\rangle \wedge Adm^{\rho}_{\mathcal{V}}(Tr,\beta,c)) \\
&\qquad \Rightarrow \exists \alpha' \in E^* : (\beta\langle c\rangle.\alpha' \in Tr \wedge \alpha'|_{V\cup C} = \alpha|_{V\cup C}))
\end{aligned}
$$

Figure 2: Formal definitions of the BSPs *BSD*, *BSI*, and *BSIA*$^{\rho}$ (see Definition 11)

that the first conjunct in the representation of the two related properties is identical, i.e., $R_{\mathcal{H}}(Tr)$ and $R_{\mathcal{HI}}(Tr)$, respectively. Moreover, we have $SEP(ES) \Rightarrow IBGNI^*(ES)$ and $NF(ES) \Rightarrow GNF(ES)$ [15].

Numerous other properties have been represented in the *MAKS* [15] and this provided a basis for the derivation of unwinding theorems [12] and of compositionality results [14]. There are extensions of the *MAKS* for capturing declassification [13] and encrypted communication [9]. The *MAKS* has also been applied in concrete case studies [16, 17, 29].

4.2 Representation Theorems

Separability and interleaving-based generalized noninterference can be represented in the *FSIF* without range or domain restrictions (see Theorems 2 and 3). The representations of these properties in the *MAKS* (Theorem 10) follow a common pattern, namely $R_{\mathcal{V}}(Tr) \wedge IA^{\rho_C}_{\mathcal{V}}(Tr)$, and differ only in the view. The following theorem generalizes this observation by showing that the set of all properties that can be represented in the *FSIF* without range or domain restrictions coincides with the set of properties that can be represented with this pattern in the *MAKS*. This is the first of the two main theorems of this article.

THEOREM 11. *Let $\kappa : E \to \{0,1,2\}$ be a function, $\mathcal{V} = (V,N,C)$ be a view in E, and ρ_C be like in Theorem 10. If $E^{\kappa}_0 = N$, $E^{\kappa}_1 = C$, and $E^{\kappa}_2 = V$ then the following two propositions are equivalent:*

1. *ES is closed under some covering set of sifs of type F_{κ}.*
2. *$R_{\mathcal{V}}(Tr)$ and $IA^{\rho_C}_{\mathcal{V}}(Tr)$ hold.*

The proof of Theorem 11 is in Appendix A.2.

Noninference and generalized noninference are represented with the domain restriction E in the *FSIF* (Theorems 7 and 8) and their representation in the *MAKS* (Theorem 10) also follows a common pattern, namely $R_{\mathcal{V}}(Tr)$. The following theorem generalizes this observation by showing that the set of all properties that can be represented in the *FSIF* with the domain restriction E coincides with the set of properties that can be represented with this pattern in the *MAKS*.

THEOREM 12. *Let κ, $\mathcal{V}$, ρ_C, E^{κ}_0, E^{κ}_1, and E^{κ}_2 be defined like in Theorem 11. If $E^{\kappa}_0 = N$, $E^{\kappa}_1 = C$, and $E^{\kappa}_2 = V$ then the following two propositions are equivalent:*

1. *ES is closed under a set $\mathcal{F}$ of sifs of type F_{κ} that covers F_{κ} under the domain restriction E.*
2. *$R_{\mathcal{V}}(Tr)$ holds.*

The proof of Theorem 12 is in Appendix A.3.

As a consequence of Schema (3), Theorem 11, and Theorem 12, we obtain the following corollary:

COROLLARY 1. *A property can be represented in the FSIF if and only if it can be represented in the MAKS in one of the forms $R_{\mathcal{V}}(Tr)$ and $R_{\mathcal{V}}(Tr) \wedge IA^{\rho_C}_{\mathcal{V}}(Tr)$.*

5. LIMITATIONS OF THE FSIF

In this section, we explore the current limitations of the *FSIF*. We discuss three classes of properties that can be represented in the *MAKS* but not in the *FSIF*. To this end, we vary the schema $R_{\mathcal{V}}(Tr) \wedge IA^{\rho_C}_{\mathcal{V}}(Tr)$ in three different ways, namely, (1) by dropping the ρ_C-admissibility condition, (2) by modifying the function ρ_C, and (3) by restricting the permissible corrections. We also sketch possibilities for improving the expressiveness of the *FSIF*.

5.1 Dropping the Admissibility Condition

Dropping $Adm^{\rho}_{\mathcal{V}}(Tr,\beta,c)$ in the definition of IA^{ρ} results in the BSP I. By replacing IA^{ρ_C} with I in the *MAKS* representation of $IBGNI^*(ES)$ (cf. Theorem 10), we obtain a property that is equivalent to $IBGNI(ES)$, the variant of interleaving-based generalized noninterference defined in [33]. As $IBGNI(ES)$ is equivalent to $R_{\mathcal{HI}}(Tr) \wedge I_{\mathcal{HI}}(Tr)$ [15], $IBGNI^*(ES)$ is equivalent to $R_{\mathcal{HI}}(Tr) \wedge IA^{\rho_C}_{\mathcal{HI}}(Tr)$, and $GNF(ES)$ is equivalent to $R_{\mathcal{HI}}(Tr)$, we have $IBGNI(ES) \Rightarrow IBGNI^*(ES)$ and $IBGNI(ES) \Rightarrow GNF(ES)$.

5.2 Modifying the function ρ_C

The function ρ is a parameter in the definition of the BSP IA^{ρ}. It is instantiated with ρ_C (defined by $\rho_C((V,N,C)) = C$) in the representations of $SEP(ES)$ and $IBGNI^*(ES)$ (cf. Theorem 10). Other choices are possible. For instance, the function ρ_E (defined by $\rho_E((V,N,C)) = V \cup N \cup C$) can be used to represent the perfect security property [33]. That is, $PSP(ES)$ is equivalent to $R_{\mathcal{H}}(Tr) \wedge IA^{\rho_E}_{\mathcal{H}}(Tr)$ [15]. Another example is the function ρ_{UI} (defined by $\rho_{UI}(V,N,C) = C \cup (V \cap UI)$ where $UI \subseteq I$ is a set of user inputs), which can be used to represent nondeducibility for outputs [8]. That is, $NDO^*(ES)$ is equivalent to $R_{\mathcal{H}}(Tr) \wedge IA^{\rho_{UI}}_{\mathcal{H}}(Tr)$ [15]. The representation of these properties in the *MAKS* immediately reveals the following implications: $SEP(ES) \Rightarrow NDO^*(ES)$, $NDO^*(ES) \Rightarrow PSP(ES)$, and $PSP(ES) \Rightarrow NF(ES)$.

5.3 Restricting the Permissible Corrections

The BSPs R, I, and IA^{ρ} permit corrections that modify occurrences of events in the set N of the given view and leave occurrences of events in $V \cup C$ unchanged. The permissible corrections can be further restricted by disallowing modifications before the first position where a change has been caused by the perturbation. The underlying intuition is that corrections should causally depend on the perturbation. Benefits of such a restriction are that it often facilitates the preservation of the resulting property under composition and also prevents some subtle dangers of information leakage

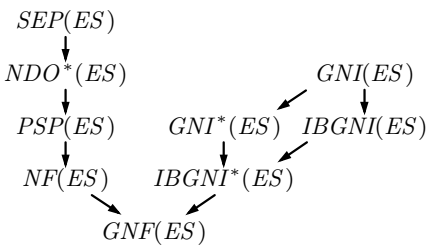
(see Section 3.4.4 in [15] for further details). By disallowing non-causal corrections, we obtain the following BSPs:

DEFINITION 11. *The BSPs BSD (for Backwards Strict Deletion), BSI (for Backwards Strict Insertion), and BSIA$^{\rho}$ (for Backwards Strict Insertion of ρ-Admissible events) are defined in Figure 4.*

The original definition of generalized noninterference [18] can be expressed with these BSPs: $GNI(ES)$ is equivalent to $BSD_{\mathcal{HI}}(Tr) \wedge BSI_{\mathcal{HI}}(Tr)$. One can define another sensible variant of generalized noninterference by $GNI^*(ES) \equiv BSD_{\mathcal{HI}}(Tr) \wedge BSIA^{\rho_C}_{\mathcal{HI}}(Tr)$. The relationship of $GNI(ES)$ to $GNI^*(ES)$ is analogous to the one of $IBGNI(ES)$ to $IBGNI^*(ES)$. In particular $GNI^*(ES)$ perturbs a trace by inserting a high-level input event only if the event is ρ_C-admissible at the given position. The representation of these properties in the *MAKS* immediately reveals the following implications: $GNI(ES) \Rightarrow IBGNI(ES)$, $GNI(ES) \Rightarrow GNI^*(ES)$, and $GNI^*(ES) \Rightarrow IBGNI^*(ES)$.[4]

5.4 Possibilities for Improving the FSIF

The diagram below summarizes the ordering induced by the implications between noninterference-like properties that we have derived so far. Based on this taxonomy and the observations made in the previous paragraphs, we discuss possibilities for modifying the *FSIF* in order to increase its expressiveness. However, a full elaboration of such extensions is outside the scope of the current article.



According to Theorem 3, $IBGNI^*(ES)$ is equivalent to: Tr is closed under a covering set of *sifs* of type $F_{\kappa_{IBGNI^*}}$. One needs to strengthen the closure requirement for representing $IBGNI(ES)$, which the implication $IBGNI(ES) \Rightarrow IBGNI^*(ES)$ already suggests, and – as elaborated in Section 5.1 – this strengthening must correspond to dropping the ρ_C-admissibility condition. One obvious candidate for a solution is to modify the closure requirement to

$$\forall f \in \mathcal{F} : \forall \tau_1 \in E^* : \forall \tau_2 \in Tr : f(\tau_1, \tau_2) \in Tr$$

Note that $\tau_1 \in Tr$ is not required, which is the difference to Definition 6. As a consequence, the trace τ_2 is perturbed by an arbitrary sequence of events in E_1^κ, i.e. an arbitrary sequence of high input events.

Being closed under a covering set of *sifs* of type $F_{\kappa_{SEP}}$ is equivalent to $SEP(ES)$ (cf. Theorem 2). In order to represent $NDO^*(ES)$ and $PSP(ES)$, we have to relax this requirement as $SEP(ES)$ implies $NDO^*(ES)$ and $PSP(ES)$. As elaborated in Section 5.2, this relaxation must correspond to exchanging the parameter ρ_C with ρ_E and ρ_{UI}, respectively. A candidate for a solution, e.g., for representing $BSIA^{\rho_E}$ would be to permit partial functions in the definition of *sifs*. This would allow one to restrict the *sifs* under considerations to ones that interleave two traces t_1, t_2 only if they have a particular form, i.e. $t_1 = \beta.\langle c \rangle$ and $t_2 = \beta.\alpha$.

In order to represent $GNI(ES)$ and $GNI^*(ES)$, one would have to strengthen the closure requirement for $IBGNI^*(ES)$ such that only causal corrections are permitted – as elaborated in Section 5.3. This seems the greatest challenge. We do not yet see any elegant possibilities for improving the expressiveness of *FSIF* in this direction.

[4]Note that $BSD_{\mathcal{HI}}(Tr) \Rightarrow R_{\mathcal{HI}}(Tr)$ follows from the definitions of R and BSD.

6. RELATED WORK

There are several other frameworks in which noninterference-like properties can be analyzed. Roscoe and his coworkers show how noninterference can be captured through determinism [26, 24, 25]. This leads to very rigorous notions of noninterference, but the drawback is that only very limited forms of nondeterministic behavior are permitted. In [3], Focardi and Gorrieri represent properties in the model of labeled transition systems and introduce the process algebra SPA, a variant of Milner's CCS [21], as a syntactic framework. The results derived in their framework include taxonomies of noninterference-like properties and various compositionality results. There are several extensions of the basic framework, e.g., for capturing declassification [2] or mobility [5]. Peri, Wulf, and Kienzle represent noninterference-like properties in a many-sorted predicate logic [23]. This is a very expressive framework (though the authors limit their investigation to the four properties already considered in [19]), but seems too general to derive parametric compositionality results. Zakinthinos and Lee [32, 33] introduce the concept of Low-Level Equivalence Sets and, based on this concept, a uniform schema for representing noninterference-like properties. They derive a taxonomy of noninterference-like properties, unwinding theorems, and compositionality results. However, the schema seems not suitable for the derivation of parametric results, and it is not expressive enough to represent properties like, e.g., separability or the perfect security property. In order to express these properties, Zakinthinos and Lee violate their schema. Ryan and Schneider elaborate a correspondence between noninterference-like properties and various notions of process equivalence [27, 28]. By exploiting this correspondence, they analyze several noninterference-like properties, compare them to each other and derive unwinding results as well as compositionality results. Focardi and Martinelli [4] propose a uniform schema for representing security properties. They illustrate how this schema can be instantiated to capture several noninterference-like properties as well as other security properties. Based on their uniform representation, they clarify the relationship between these security properties. All of these frameworks primarily aim at possibilistic properties. Beyond this, there are also probabilistic variants of noninterference (cf., e.g., [31, 7, 1]).

In [32], Zakinthinos argues that every property representable in the *FSIF* can also be represented in his framework. However, his variant of the *FSIF* for event systems does not adequately reflect the original framework (cf. Section 3.4).

7. CONCLUSION

In this article, we clarified the relationship between the Framework of Selective Interleaving Functions [19, 20] and the Modular Assembly Kit for Security Properties [11, 14, 15]. Our main result is that the *MAKS* is strictly more expressive than the *FSIF* in its current form.

We showed that all noninterference-like properties that can be represented in the *FSIF* can also be represented in the *MAKS* and that there are properties that can be represented in the *MAKS* but not in the *FSIF*. To enable this formal comparison, we had to develop a variant of the *FSIF* for the model of event systems. We explained why our variant adequately reflects the original framework and showed that our variant is adequate for representing all properties that were represented in the original *FSIF*. McLean introduced the notions of range restrictions and domain restrictions only by example. We showed how these notions can be formalized and found that domain restrictions better support a comparison of different properties. The variant of the *FSIF* in this article is an improvement of an earlier attempt by Zakinthinos [32] that did not capture the original framework in an adequate way (see Section 6).

We investigated three classes of noninterference-like properties that cannot be represented in the *FSIF*. Based on the representation of these properties in the *MAKS*, we illustrated for which concepts of the *MAKS* there is no counterpart in the *FSIF*. This also inspired some ideas for modifying the *FSIF* in order to improve its expressiveness. It would be interesting to investigate such possibilities in more detail, but this remains a task for the future. Another interesting question is how the two frameworks compare with respect to the compositionality results that one can derive.

8. REFERENCES

[1] Backes, M., and Pfitzmann, B. Computational Probabilistic Non-Interference. *International Journal of Information Security (IJIS) 3*, 1 (2004), 42–60.

[2] Bossi, A., Piazza, C., and Rossi, S. Modelling Downgrading in Information Flow Security. In *Proceedings of the IEEE Computer Security Foundations Workshop* (2004), pp. 187–201.

[3] Focardi, R., and Gorrieri, R. A Classification of Security Properties for Process Algebras. *Journal of Computer Security 3*, 1 (1995), 5–33.

[4] Focardi, R., and Martinelli, F. A Uniform Approach to the Definition of Security Properties. In *Proceedings of FM'99 – Formal Methods (vol. 1)* (1999), vol. 1708 of *LNCS*, pp. 794–813.

[5] Focardi, R., and Rossi, S. Information Flow Security in Dynamic Contexts. In *Proceedings of the 15th IEEE Computer Security Foundations Workshop* (2002), pp. 307–319.

[6] Goguen, J. A., and Meseguer, J. Security Policies and Security Models. In *Proceedings of the IEEE Symposium on Security and Privacy* (1982), pp. 11–20.

[7] Gray, J. W. Toward a Mathematical Foundation for Information Flow Security. In *Proceedings of the IEEE Symposium on Security and Privacy* (1991), pp. 21–34.

[8] Guttman, J. D., and Nadel, M. E. "What Needs Securing?". In *Proceedings of the IEEE Computer Security Foundations Workshop* (1988), pp. 34–57.

[9] Hutter, D., and Schairer, A. Possibilistic Information Flow Control in the Presence of Encrypted Communication. In *Proceedings of the European Symposium on Research in Computer Security* (2004), vol. 3193 of *LNCS*, pp. 209–224.

[10] Jacob, J. On the Derivation of Secure Components. In *Proceedings of the IEEE Symposium on Security and Privacy* (1989), pp. 242–247.

[11] Mantel, H. Possibilistic Definitions of Security – An Assembly Kit. In *Proceedings of the IEEE Computer Security Foundations Workshop* (2000), pp. 185–199.

[12] Mantel, H. Unwinding Possibilistic Security Properties. In *Proceedings of the European Symposium on Research in Computer Security* (2000), vol. 1895 of *LNCS*, pp. 238–254.

[13] Mantel, H. Information Flow Control and Applications – Bridging a Gap. In *Proceedings of FME 2001: Formal Methods for Increasing Software Productivity* (2001), vol. 2021 of *LNCS*, pp. 153–172.

[14] Mantel, H. On the Composition of Secure Systems. In *Proceedings of the IEEE Symposium on Security and Privacy* (2002), pp. 88–104.

[15] Mantel, H. *A Uniform Framework for the Formal Specification and Verification of Secure Information Flow*. PhD thesis, Saarland University, Saarbrücken, Germany, 2003.

[16] Mantel, H., and Sabelfeld, A. A Generic Approach to the Security of Multi-threaded Programs. In *Proceedings of the 14th IEEE Computer Security Foundations Workshop* (2001), pp. 126–142.

[17] Mantel, H., Schairer, A., Kabatnik, M., Kreutzer, M., and Zugenmaier, A. Using Information Flow Control to Evaluate Access Protection of Location Information in Mobile Communication Networks. Tech. Rep. 159, Computer Science Department, University of Freiburg, 2001.

[18] McCullough, D. Specifications for Multi-Level Security and a Hook-Up Property. In *Proceedings of the IEEE Symposium on Security and Privacy* (1987), pp. 161–166.

[19] McLean, J. D. A General Theory of Composition for Trace Sets Closed under Selective Interleaving Functions. In *Proceedings of the IEEE Symposium on Research in Security and Privacy* (1994), pp. 79–93.

[20] McLean, J. D. A General Theory of Composition for a Class of "Possibilistic" Security Properties. *IEEE Transaction on Software Engineering 22*, 1 (1996), 53–67.

[21] Milner, R. *Communication and Concurrency*. International Series in Computer Science. Prentice Hall, 1989.

[22] O'Halloran, C. A Calculus of Information Flow. In *Proceedings of the European Symposium on Research in Computer Security* (1990), pp. 147–159.

[23] Peri, R. V., Wulf, W. A., and Kienzle, D. M. A Logic of Composition for Information Flow Predicates. In *Proceedings of the 9th IEEE Computer Security Foundations Workshop* (1996), pp. 82–93.

[24] Roscoe, A. W. CSP and Determinism in Security Modelling. In *Proceedings of the IEEE Symposium on Security and Privacy* (1995), pp. 114–127.

[25] Roscoe, A. W., and Goldsmith, M. H. What is intransitive noninterference? In *Proceedings of the 12th IEEE Computer Security Foundations Workshop* (1999), pp. 228–238.

[26] Roscoe, A. W., Woodcock, J. C. P., and Wulf, L. Non-interference through Determinism. In *Proceedings of the European Symposium on Research in Computer Security* (1994), vol. 875 of *LNCS*, pp. 33–53.

[27] Ryan, P. Y. A., and Schneider, S. A. Process Algebra and Non-interference. In *Proceedings of the 12th IEEE Computer Security Foundations Workshop* (1999), pp. 214–227.

[28] Ryan, P. Y. A., and Schneider, S. A. Process Algebra and Non-Interference. *Journal of Computer Security 9*, 1/2 (2001), 75–103.

[29] Schäfer, I. Information Flow Control for Multiagent Systems, A Case Study in Comparison Shopping. Master's thesis, Universität Rostock, 2004.

[30] v. Oheimb, D. Information Flow Control Revisited: Noninfluence = Noninterference + Nonleakage. In *Proceedings of the European Symposium on Research in Computer Security* (2004), vol. 3193 of *LNCS*, pp. 225–243.

[31] Wittbold, J. T., and Johnson, D. M. Information Flow in Nondeterministic Systems. In *Proceedings of the IEEE Symposium on Research in Security and Privacy* (1990), pp. 144–161.

[32] Zakinthinos, A. *On the Composition of Security Properties*. PhD thesis, Graduate Department of Electrical and Computer Engineering, University of Toronto, 1996.

[33] Zakinthinos, A., and Lee, E. S. A General Theory of Security Properties. In *Proceedings of the IEEE Symposium on Security and Privacy* (1997), pp. 94–102.

APPENDIX

A. PROOFS

A.1 Representation of the Example Properties

Proof of Theorem 2. Theorem 10 shows $SEP(ES)$ and $R_{\mathcal{H}}(Tr) \wedge IA^{PC}_{\mathcal{H}}(Tr)$ to be equivalent. Theorem 11 implies that the latter statement is equivalent to the requirement

that *ES* is closed under some covering set of *sifs* of type $F_{\kappa_{SEP}}$. □

PROOF OF THEOREM 3. Theorem 10 shows $IBGNI^*(ES)$ and $R_{\mathcal{HI}}(Tr) \wedge IA^{\rho_C}_{\mathcal{HI}}(Tr)$ to be equivalent. Theorem 11 implies that the latter statement is equivalent to the requirement that *ES* is closed under some covering set of *sifs* of type $F_{\kappa_{IBGNI^*}}$. □

PROOF OF THEOREM 4. Firstly, assume that there is a covering set $\mathcal{F}$ of *sifs* of type $F_{\kappa_{NF}}$ such that *ES* is closed under $\mathcal{F}$ and $f(t_1,t_2)|_{E_0^{\kappa_{NF}}} = \langle\rangle$ holds for all $f \in \mathcal{F}$ and all $t_1, t_2 \in E^*$. Let $\tau \in Tr$ (if $Tr = \emptyset$ then the statement holds trivially) and $f \in \mathcal{F}$ ($\mathcal{F}$ is nonempty because it covers $F_{\kappa_{NF}}$) be arbitrary. Since f is a *sif* and $E_2^{\kappa_{NF}} = L$, we have $f(\langle\rangle,\tau)|_L = \tau|_L$. Since $E_0^{\kappa_{NF}} = H$, we have $f(\langle\rangle,\tau)|_H = \langle\rangle$ according to our assumptions about $\mathcal{F}$. From $E_1^{\kappa_{NF}} = \emptyset$, we obtain $f(\langle\rangle,\tau) = \tau|_L$. Since *ES* is closed under $\mathcal{F}$, $\tau|_L \in Tr$ holds. Hence, *NF*(*ES*) holds.

Secondly, assume that *NF*(*ES*) holds. Define

$$\mathcal{F} = \left\{ f \text{ is a } \textit{sif} \text{ of type } F_{\kappa_{NF}} \;\middle|\; \begin{array}{l} \forall t_1, t_2 \in E^* : \\ f(t_1,t_2)|_{E_0^{\kappa_{NF}}} = \langle\rangle \end{array} \right\}.$$

Let $f \in \mathcal{F}$ and $\tau_1, \tau_2 \in Tr$ be arbitrary. Since $f(\tau_1,\tau_2) = \tau_2|_L$ (follows from $E_1^{\kappa_{NF}} = \emptyset$ and our definition of $\mathcal{F}$) and $\tau_2|_L \in Tr$ (follows from *NF*(*ES*)), we have $f(\tau_1,\tau_2) \in Tr$. Hence, *ES* is closed under $\mathcal{F}$. It remains to show $\mathcal{F}$ covers $F_{\kappa_{NF}}$. Let $t_1, t_2 \in E^*$ and $t \in \textit{interleaving}(t_1|_{E_1^{\kappa_{NF}}}, t_2|_{E_2^{\kappa_{NF}}})$ be arbitrary. Since $E_1^{\kappa_{NF}} = \emptyset$, we have $t = t_2|_{E_2^{\kappa_{NF}}}$. Define f by: $\forall t_1', t_2' \in E^* : f(t_1',t_2') = t_2'|_{E_2^{\kappa_{NF}}}$. Obviously, $f \in \mathcal{F}$ and $f(t_1,t_2) = t_2|_{E_2^{\kappa_{NF}}} = t$ hold. Hence, $\mathcal{F}$ covers $F_{\kappa_{NF}}$. □

PROOF OF THEOREM 5. Firstly, assume that there is a covering set $\mathcal{F}$ of *sifs* of type $F_{\kappa_{NF}}$ such that *ES* is closed under $\mathcal{F}$ and $f(t_1,t_2)|_{E_0^{\kappa_{NF}} \cap I} = \langle\rangle$ holds for all $f \in \mathcal{F}$ and all $t_1, t_2 \in E^*$. Let $\tau \in Tr$ and $f \in \mathcal{F}$ be arbitrary. Since f is a *sif* and $E_2^{\kappa_{NF}} = L$, we have $f(\langle\rangle,\tau)|_L = \tau|_L$. Since $E_0^{\kappa_{NF}} = H$, $f(\langle\rangle,\tau)|_{H\cap I} = \langle\rangle$ follows from our assumptions about $\mathcal{F}$. For $\tau' = f(\langle\rangle,\tau)$, we have $\tau' \in Tr$ (*ES* is closed under $\mathcal{F}$), $\tau'|_L = \tau|_L$, and $\tau'|_{H\cap I} = \langle\rangle$. Hence, *GNF*(*ES*) holds.

Secondly, assume that *GNF*(*ES*) holds. Define $\mathcal{F} = \{f\}$ where

$$f(t_1,t_2) = \begin{cases} t_2|_{E_2^{\kappa_{NF}}} & \text{if } t_2 \notin Tr \\ t_2' & \text{if } t_2 \in Tr \\ & \text{where } t_2' \in E^* \text{ is a trace with} \\ & t_2' \in Tr,\ t_2'|_L = t_2|_L, \text{ and} \\ & t_2'|_{H\cap I} = \langle\rangle \text{ (t_2' with these prop-} \\ & \text{erties exists because } \textit{GNF}(\textit{ES}) \\ & \text{holds)} \end{cases}$$

Hence, *ES* is closed under $\mathcal{F}$. It remains to show $\mathcal{F}$ covers $F_{\kappa_{NF}}$. Let $t_1, t_2 \in E^*$ and $t \in \textit{interleaving}(t_1|_{E_1^{\kappa_{NF}}}, t_2|_{E_2^{\kappa_{NF}}})$ be arbitrary. Since $E_1^{\kappa_{NF}} = \emptyset$, we have $t = t_2|_{E_2^{\kappa_{NF}}}$. Since $f(t_1,t_2)|_{E_1^{\kappa_{NF}} \cup E_2^{\kappa_{NF}}} = t_2|_{E_2^{\kappa_{NF}}} = t$ and t_1, t_2, t were chosen arbitrarily, we conclude that $\mathcal{F}$ covers $F_{\kappa_{NF}}$. □

PROOF OF THEOREM 7. *NF*(*ES*) and $R_{\mathcal{H}}(Tr)$ are equivalent (cf. Theorem 10). The latter statement is equivalent to the requirement that *ES* is closed under some set of *sifs* of type $F_{\kappa_{SEP}}$ that covers $F_{\kappa_{SEP}}$ under the domain restriction E (cf. Theorem 12). □

PROOF OF THEOREM 8. *GNF*(*ES*) and $R_{\mathcal{HI}}(Tr)$ are equivalent (cf. Theorem 10). The latter statement is equivalent to the requirement that *ES* is closed under some set of *sifs* of type $F_{\kappa_{IBGNI^*}}$ that covers $F_{\kappa_{IBGNI^*}}$ under the domain restriction E (cf. Theorem 12). □

A.2 First Representation Theorem

We first prove a lemma that is used in the proof of Theorem 11.

LEMMA 1. *Let $\mathcal{V} = (V, N, C)$ be a view in E and ρ_C be defined like in Theorem 10. Define $\kappa : E \to \{0,1,2\}$ by $E_0^\kappa = N$, $E_1^\kappa = C$, and $E_e^\kappa = V$.*

If $R_{\mathcal{V}}(Tr) \wedge IA^{\rho_C}_{\mathcal{V}}(Tr)$ holds then, for every $n \in I\!N$, there is a set $\mathcal{F}^n \subseteq ((\bigcup_{n' \le n} E^{n'}) \times E^*) \to E^*$ for which the following three propositions hold:

1. *Each $f \in \mathcal{F}^n$ approximates a sif of type F_κ, i.e.*

$$\forall f \in \mathcal{F}^n : \forall t_1 \in \textstyle\bigcup_{n' \le n} E^{n'} : \forall t_2 \in E^* : \\ [(f(t_1,t_2))|_{E_1^\kappa} = t_1|_{E_1^\kappa} \wedge (f(t_1,t_2))|_{E_2^\kappa} = t_2|_{E_2^\kappa}] \tag{9}$$

2. *$\mathcal{F}^n$ approximates the coverage requirement, i.e.*

$$\forall \iota \in ((\textstyle\bigcup_{n' \le n} E^{n'}) \times E^*) \to (E_1^\kappa \cup E_2^\kappa)^* : \\ [(\forall t_1' \in \textstyle\bigcup_{n' \le n} E^{n'} : \forall t_2' \in E^* : \\ \iota(t_1', t_2') \in \textit{interleaving}(t_1'|_{E_1^\kappa}, t_2'|_{E_2^\kappa})) \\ \Rightarrow \exists f \in \mathcal{F}^n : \forall t_1 \in \textstyle\bigcup_{n' \le n} E^{n'} : \forall t_2 \in E^* : \\ f(t_1,t_2)|_{E_1^\kappa \cup E_2^\kappa} = \iota(t_1,t_2)] \tag{10}$$

3. *ES is closed under every $f \in \mathcal{F}^n$, i.e.*

$$\forall f \in \mathcal{F}^n : \forall t_1 \in \textstyle\bigcup_{n' \le n} E^{n'} : \forall t_2 \in E^* : \\ [(t_1 \in Tr \wedge t_2 \in Tr) \Rightarrow f(t_1,t_2) \in Tr] \tag{11}$$

PROOF. The proof proceeds by induction on n.

Base case ($n = 0$): $\bigcup_{n' \le n} E^{n'} = \{\langle\rangle\}$ holds. We construct $\mathcal{F}^0$ by $\mathcal{F}^0 = \{f^0\}$ where f^0 is defined as follows for $t_1 \in \bigcup_{n' \le n} E^{n'}$ (i.e. $t_1 = \langle\rangle$) and $t_2 \in E^*$:

$$f^0(t_1,t_2) = \begin{cases} t_2|_{E_2^\kappa} & \text{if } t_2 \notin Tr \\ t_2' & \text{if } t_2 \in Tr \\ & \text{where } t_2' \in E^* \text{ is some trace with} \\ & t_2' \in Tr,\ t_2'|_{E_2^\kappa} = t_2|_{E_2^\kappa}, \text{ and } t_2'|_{E_1^\kappa} = \\ & \langle\rangle \text{ (existence of such a } t_2' \text{ follows} \\ & \text{from } E_1^\kappa = C,\ E_2^\kappa = V, \text{ and } R_{\mathcal{V}}(Tr) \end{cases}$$

From the construction of f^0, we immediately obtain (9) and (11) for $n = 0$.

As $n = 0$, there is only one function $\iota^0 \in ((\bigcup_{n' \le 0} E^{n'}) \times E^*) \to (E_1^\kappa \cup E_2^\kappa)^*$ such that $\forall t_1' \in \bigcup_{n' \le n} E^{n'} : \forall t_2' \in E^* : \iota^0(t_1',t_2') \in \textit{interleaving}(t_1'|_{E_1^\kappa}, t_2'|_{E_2^\kappa})$ and $\iota^0(t_1',t_2') = t_2|_{E_2^\kappa}$. By construction, we have that $f^0(t_1,t_2)|_{E_1^\kappa \cup E_2^\kappa} = f^0(\langle\rangle,t_2)|_{E_1^\kappa \cup E_2^\kappa} = t_2|_{E_2^\kappa} = \iota^0(t_1,t_2)$ holds for all $t_1 \in E^0$ and $t_2 \in E^*$. Hence, (10) holds for $n = 0$.

Step case ($n > 0$): We construct $\mathcal{F}^n$ as follows:

$$\mathcal{F}^n = \left\{ f_\iota^n \;\middle|\; \begin{array}{l} f_\iota^n : ((\bigcup_{n' \le n} E^{n'}) \times E^*) \to E^* \\ \wedge\ \iota : ((\bigcup_{n' \le n} E^{n'}) \times E^*) \to (E_1^\kappa \cup E_2^\kappa)^* \\ \wedge\ (\forall t_1', t_2' \in E^* : \\ \quad \iota(t_1', t_2') \in \textit{interleaving}(t_1'|_{E_1^\kappa}, t_2'|_{E_2^\kappa})) \\ \wedge\ \mathrm{Def}_{f_\iota^n} \end{array} \right\}$$

where $\mathrm{Def}_{f_\iota^n}$ defines f_ι^n as follows for $t_1 \in \bigcup_{n'<n} E^{n'}$ and $t_2 \in E^*$:

$$\mathrm{Def}_{f_\iota^n} \equiv f_\iota^n(t_1,t_2) = \begin{cases} \iota(t_1,t_2) & \text{if } t_1 \notin Tr \text{ or } t_2 \notin Tr \\ t_{t_1,t_2} & \text{if } t_1,t_2 \in Tr \\ & \text{where } t_{t_1,t_2} \in E^* \text{ is a} \\ & \text{trace with } t_{t_1,t_2} \in Tr \text{ and} \\ & t_{t_1,t_2}|_{E_1^\kappa \cup E_2^\kappa} = \iota(t_1,t_2) \end{cases}$$

The validity of (9), (10), and (11) follows immediately from this construction. It only remains to prove that $f_\iota^n(t_1,t_2)$ is well defined, i.e. that there always is a trace $t_{t_1,t_2} \in Tr$ such that $t_{t_1,t_2}|_{E_1^\kappa \cup E_2^\kappa} = \iota(t_1,t_2)$. For proving the existence of t_{t_1,t_2}, we make a case distinction on (1) $t_1 \in \bigcup_{n'<n} E^{n'}$ and (2) $t_1 \in E^n$:

Case 1 ($t_1 \in \bigcup_{n'<n} E^{n'}$): There exists a function $\iota' : ((\bigcup_{n'<n} E^{n'}) \times E^*) \to (E_1^\kappa \cup E_2^\kappa)^*$ that equals ι on the restricted domain $(\bigcup_{n'<n} E^{n'}) \times E^*$, i.e. $\iota'(t_3,t_4) = \iota(t_3,t_4)$ for all $t_3 \in \bigcup_{n'<n} E^{n'}$ and $t_4 \in E^*$. According to the induction assumption, there is a function $f_{\iota'}^{n-1} \in \mathcal{F}^{n-1}$ such that, for all $t_3 \in \bigcup_{n'<n} E^{n'}$ and $t_4 \in E^*$, $f_{\iota'}^{n-1}(t_3,t_4)|_{E_1^\kappa \cup E_2^\kappa} = \iota'(t_3,t_4)$ holds and $t_3,t_4 \in Tr$ implies $f_{\iota'}^{n-1}(t_3,t_4) \in Tr$. We choose $t_{t_1,t_2} = f_{\iota'}^{n-1}(t_1,t_2)$. Consequently, $t_{t_1,t_2} \in Tr$ holds. Moreover, $t_{t_1,t_2}|_{E_1^\kappa \cup E_2^\kappa} = \iota(t_1,t_2)$ as $t_{t_1,t_2}|_{E_1^\kappa \cup E_2^\kappa} = \iota'(t_1,t_2)$ and ι' equals ι for the restricted domain $(\bigcup_{n'<n} E^{n'}) \times E^*$.

Case 2 ($t_1 \in E^n$): Let $t_1' \in E^{n-1}$ and $e \in E$ be defined by $t_1 = t_1'.\langle e\rangle$. We make another case distinction on e: (2a) $e \notin E_1^\kappa$ and (2b) $e \in E_1^\kappa$.

Case 2a ($e \notin E_1^\kappa$): There is a function $\iota' : ((\bigcup_{n'<n} E^{n'}) \times E^*) \to (E_1^\kappa \cup E_2^\kappa)^*$ such that $\iota'(t_1',t_2) = \iota(t_1,t_2)$ and $\iota'(t_3,t_4) \in interleaving(t_3|_{E_1^\kappa}, t_4|_{E_2^\kappa})$ holds for all $t_3 \in \bigcup_{n'<n} E^{n'}$ and $t_4 \in E^*$. According to the induction assumption, there is a function $f_{\iota'}^{n-1} \in \mathcal{F}^{n-1}$ such that, for all $t_3 \in \bigcup_{n'<n} E^{n'}$ and $t_4 \in E^*$, $f_{\iota'}^{n-1}(t_3,t_4)|_{E_1^\kappa \cup E_2^\kappa} = \iota'(t_3,t_4)$ holds and $t_3,t_4 \in Tr$ implies $f_{\iota'}^{n-1}(t_3,t_4) \in Tr$. For $t_{t_1,t_2} = f_{\iota'}^{n-1}(t_1',t_2)$, we obtain $t_{t_1,t_2} \in Tr$ and $t_{t_1,t_2}|_{E_1^\kappa \cup E_2^\kappa} = \iota'(t_1',t_2) = \iota(t_1,t_2)$.

Case 2b ($e \in E_1^\kappa$): Define $\alpha,\beta \in E^*$ by $\iota(t_1'.\langle e\rangle, t_2) = \beta.\langle e\rangle.\alpha$ and $\alpha|_{E_1^\kappa} = \langle\rangle$. There is a function $\iota' : ((\bigcup_{n'<n} E^{n'}) \times E^*) \to (E_1^\kappa \cup E_2^\kappa)^*$ such that $\iota'(t_1',t_2) = \beta.\alpha$ and $\iota(t_3,t_4) \in interleaving(t_3|_{E_1^\kappa}, t_4|_{E_2^\kappa})$ hold for all $t_3 \in \bigcup_{n'<n} E^{n'}$ and $t_4 \in E^*$. According to the induction assumption, there is a function $f_{\iota'}^{n-1} \in \mathcal{F}^{n-1}$ such that, for all $t_3 \in \bigcup_{n'<n} E^{n'}$ and $t_4 \in E^*$, $f_{\iota'}^{n-1}(t_3,t_4)|_{E_1^\kappa \cup E_2^\kappa} = \iota'(t_3,t_4)$ holds and $t_3,t_4 \in Tr$ implies $f_{\iota'}^{n-1}(t_3,t_4) \in Tr$. Since $f_{\iota'}^{n-1}(t_1',t_2)|_{E_1^\kappa \cup E_2^\kappa} = \iota'(t_1',t_2) = \beta.\alpha$, there are $\alpha',\beta' \in E^*$ with $\beta'.\alpha' = f_{\iota'}^{n-1}(t_1',t_2)$, $\beta'|_{E_1^\kappa \cup E_2^\kappa} = \beta|_{E_1^\kappa \cup E_2^\kappa}$, $\alpha'|_{E_2^\kappa} = \alpha|_{E_2^\kappa}$, and $\alpha'|_{E_1^\kappa} = \langle\rangle$. From $t_1'|_{E_1^\kappa} = \beta'|_{E_1^\kappa}$ and $t_1'.\langle e\rangle \in Tr$, we obtain $Adm_{\mathcal{V}}^{\rho C}(Tr,\beta',e)$. We conclude from $IA_{\mathcal{V}}^{\rho C}(Tr)$ that there are $\alpha'',\beta'' \in E^*$ with $\beta''.\langle e\rangle.\alpha'' \in Tr$, $\beta''|_{E_1^\kappa \cup E_2^\kappa} = \beta'|_{E_1^\kappa \cup E_2^\kappa}$, $\alpha''|_{E_2^\kappa} = \alpha'|_{E_2^\kappa}$, and $\alpha''|_{E_1^\kappa} = \langle\rangle$. For $t_{t_1,t_2} = \beta''.\langle e\rangle.\alpha''$, we obtain $t_{t_1,t_2} \in Tr$ and $t_{t_1,t_2}|_{E_1^\kappa \cup E_2^\kappa} = (\beta''.\langle e\rangle.\alpha'')|_{E_1^\kappa \cup E_2^\kappa} = (\beta'.\langle e\rangle.\alpha')|_{E_1^\kappa \cup E_2^\kappa} = (\beta.\langle e\rangle.\alpha)|_{E_1^\kappa \cup E_2^\kappa} = \iota(t_1,t_2)$. □

Note that $R_{\mathcal{V}}(Tr)$ is applied only in the base case of the proof for Lemma 1 and that $IA_{\mathcal{V}}^{\rho C}(Tr)$ is applied only in Case 2b of the step case.

We are now ready to prove Theorem 11.

Proof of Theorem 11. Firstly, assume ES is closed under some covering set $\mathcal{F}$ of *sifs* of type F_κ. We have to show that $R_{\mathcal{V}}(Tr)$ and $IA_{\mathcal{V}}^{\rho C}(Tr)$ hold.

For proving $R_{\mathcal{V}}(Tr)$, let $\tau \in Tr$ be arbitrary (if $Tr = \emptyset$ then the proposition holds trivially). Let $f \in \mathcal{F}$ be arbitrary and define $\tau' = f(\langle\rangle,\tau)$. As ES is closed under $\mathcal{F}$ and $\langle\rangle,\tau \in Tr$ holds ($\langle\rangle \in Tr$ because ES is closed under prefixes), we have $\tau' \in Tr$. As f is a *sif* of type F_κ, we have $\tau'|_C = \tau'|_{E_1^\kappa} = f(\langle\rangle,\tau)|_{E_1^\kappa} = \langle\rangle$ and $\tau'|_V = \tau'|_{E_2^\kappa} = f(\langle\rangle,\tau)|_{E_2^\kappa} = \tau|_{E_2^\kappa} = \tau|_V$. Thus, $R_{\mathcal{V}}(Tr)$ holds.

For proving $IA_{\mathcal{V}}^{\rho C}(Tr)$, let $\alpha,\beta \in E^*$ and $c \in C$ be arbitrary. Assume $\beta.\alpha \in Tr$, $\alpha|_C = \langle\rangle$, and $Adm_{\mathcal{V}}^{\rho C}(Tr,\beta,c)$. Hence $\alpha|_{E_1^\kappa} = \langle\rangle$ and $c \in E_1^\kappa$. Since $Adm_{\mathcal{V}}^{\rho C}(Tr,\beta,c)$ there is a trace $\gamma \in E^*$ with $\gamma.\langle c\rangle \in Tr$ and $\gamma|_{E_1^\kappa} = \beta|_{E_1^\kappa}$. Consequently, $(\beta.\langle c\rangle.\alpha)|_{E_1^\kappa \cup E_2^\kappa} \in interleaving((\gamma.\langle c\rangle)|_{E_1^\kappa}, (\beta.\alpha)|_{E_2^\kappa})$ holds. Since $\mathcal{F}$ covers F_κ, there is a *sif* $f \in \mathcal{F}$ such that $(f(\gamma.\langle c\rangle, \beta.\alpha))|_{E_1^\kappa \cup E_2^\kappa} = (\beta.\langle c\rangle.\alpha)|_{E_1^\kappa \cup E_2^\kappa}$. Let $\beta',\alpha' \in E^*$ be the subsequences of $f(\gamma.\langle c\rangle, \beta.\alpha)$ before and after the last occurrence of c, respectively, i.e. $f(\gamma.\langle c\rangle, \beta.\alpha) = \beta'.\langle c\rangle.\alpha'$ and $\alpha'|_{E_1^\kappa} = \langle\rangle$. Since ES is closed under $\mathcal{F}$, we obtain $\beta'.\langle c\rangle.\alpha' \in Tr$. From $\beta'.\langle c\rangle.\alpha' \in Tr$, $\alpha'|_C = \langle\rangle$, and $(\beta'.\langle c\rangle.\alpha')|_{V \cup C} = (\beta.\langle c\rangle.\alpha)|_{V \cup C}$ we conclude that $IA_{\mathcal{V}}^{\rho C}(Tr)$ holds.

Secondly, assume $R_{\mathcal{V}}(Tr)$ and $IA_{\mathcal{V}}^{\rho C}(Tr)$. We have to show that ES is closed under some covering set $\mathcal{F}$ of *sifs* of type F_κ. We choose $\mathcal{F} = \bigcup_{n \in \mathbb{N}} \mathcal{F}^n$ where $\mathcal{F}^n$ is defined like in the proof of Lemma 1. Hence, ES is closed under $\mathcal{F}$. It remains to prove that $\mathcal{F}$ covers F_κ. Let $t_1,t_2 \in E^*$ and $t \in interleaving(t_1|_{E_1^\kappa}, t_2|_{E_2^\kappa})$ be arbitrary. Let $n \in \mathbb{N}$ be the length of t_1 (i.e. $t_1 \in E^n$). Then there is a function $\iota \in ((\bigcup_{n' \le n} E^{n'}) \times E^*) \to (E_1^\kappa \cup E_2^\kappa)^*$ such that $\forall t_1' \in \bigcup_{n' \le n} E^{n'} : \forall t_2' \in E^* : \iota(t_1',t_2') \in interleaving(t_1'|_{E_1^\kappa}, t_2'|_{E_2^\kappa})$ and $\iota(t_1,t_2) = t$. According to Lemma 1 there is a $f \in \mathcal{F}$ with $f(t_1,t_2)|_{E_1^\kappa \cup E_2^\kappa} = \iota(t_1,t_2) = t$. Hence, $\mathcal{F}$ covers F_κ. □

A.3 Second Representation Theorem

Proof of Theorem 12. Firstly, assume ES is closed under some set $\mathcal{F}$ of *sifs* of type F_κ that covers F_κ under the domain restriction E. Hence, $\mathcal{F}$ cannot be empty. Let $\tau \in Tr$ be arbitrary (if $Tr = \emptyset$ then the proposition holds trivially). Let $f \in \mathcal{F}$ be arbitrary and define $\tau' = f(\langle\rangle,\tau)$. As ES is closed under $\mathcal{F}$ and $\langle\rangle,\tau \in Tr$ holds ($\langle\rangle \in Tr$ because ES is closed under prefixes), we have $\tau' \in Tr$. As f is a *sif* of type F_κ, we have $\tau'|_C = \tau'|_{E_1^\kappa} = f(\langle\rangle,\tau)|_{E_1^\kappa} = \langle\rangle$ and $\tau'|_V = \tau'|_{E_2^\kappa} = f(\langle\rangle,\tau)|_{E_2^\kappa} = \tau|_{E_2^\kappa} = \tau|_V$. Thus, $R_{\mathcal{V}}(Tr)$.

Secondly, assume $R_{\mathcal{V}}(Tr)$. We choose $\mathcal{F} = \mathcal{F}^0$ where $\mathcal{F}^0$ is like in the proof of Lemma 1. Hence, ES is closed under $\mathcal{F}$. It remains to prove that $\mathcal{F}$ covers F_κ under the domain restriction E. Let $t_1 \in (E \setminus E)^*$ (i.e. $t_1 = \langle\rangle$), $t_2 \in E^*$, and $t \in interleaving(t_1|_{E_1^\kappa}, t_2|_{E_2^\kappa})$ be arbitrary. Then there is a function $\iota \in ((\bigcup_{n' \le 0} E^{n'}) \times E^*) \to (E_1^\kappa \cup E_2^\kappa)^*$ such that $\forall t_1' \in \bigcup_{n' \le 0} E^{n'} : \forall t_2' \in E^* : \iota(t_1',t_2') \in interleaving(t_1'|_{E_1^\kappa}, t_2'|_{E_2^\kappa})$ and $\iota(t_1,t_2) = t$. According to Lemma 1 there is a $f \in \mathcal{F}$ with $f(t_1,t_2)|_{E_1^\kappa \cup E_2^\kappa} = \iota(t_1,t_2) = t$. □