A comparison of semantic models for intransitive noninterference*

Ron van der Meyden

School of Computer Science and Engineering,
University of New South Wales
meyden@cse.unsw.edu.au

Abstract. Noninterference is a notion of information flow security, originally defined for transitive information flow policies. A number of different definitions of noninterference have been proposed for intransitive policies. These definitions are stated with respect to several different semantic models, including state machines with observations on states, state machines with outputs associated to actions, and process algebras. The paper studies the relationship between these definitions and models. Several mappings are defined that transform one semantic model into another, and the correspondences between the definitions under these mappings are precisely characterized. In particular, the paper considers definitions of intransitive noninterference due to Haigh and Young (1987), Roscoe and Goldsmith (1999) and van der Meyden (2007).

1 Introduction

A key concern of research on formal verification of computer security has been various notions of information flow, referred to by the general term *noninterference* [GM82]. Much of this area has been concerned with a very simple information flow security policy involving just two security domains High and Low, with information permitted to flow from Low to High, but not *vice versa*. A richer class of policies deals with partially ordered sets of security domains, with information permitted to flow upwards in the partial order but not down. A feature of this class of policies is that the relation describing permitted information flows is transitive.

It has been argued that there are also applications where the relation describing permitted information flows is not transitive. An example of this is systems involving High and Low security domains, as in the simplest policy, together with a *downgrader*. Here, one would like to specify that information may flow from High to the downgrader, and from the downgrader to Low, but not directly from High to Low. In other words, any information flow from High to Low must be mediated by the downgrader. This constraint makes the relation on domains that represents the security policy *intransitive*.

* Version of December 29, 2007. Thanks to Stanford University for hosting a sabbatical visit during which this research was conducted. Work of the author supported by an Australian Research Council Discovery grant.

The classical definition of noninterference [GM82], proposed originally for policies assumed to be transitive, has been felt to be inadequate for intransitive policies. A variant definition of noninterference designed for intransitive polices was first proposed by Haigh and Young [HY87], and further propounded by Rushby [Rus92] in a paper that corrected and extended Haigh and Young's results.

There have been several criticisms of this variant definition, however. Roscoe and Goldsmith [RG99] criticized it on the basis of the failure of their attempt to use it to formalise reasoning about the downgrader policy. This lead them to propose an alternative definition grounded in the idea of Low-determinism. More recently, the definition was criticized on different grounds in [vdM07], where it is argued that it permits information flows that are counter to an informal understanding of the policy. A number of different definitions are proposed in [vdM07] that resolve this problem, and it is shown that these definitions lead to a more satisfactory treatment of *unwinding* [GM84,Rus92], a proof technique for non-interference, and *access control systems* [Rus92], a general class of systems that may be shown to be secure by this technique. The relationship between these variant definitions and those of Roscoe and Goldsmith was left open in this work.

In this paper we conduct a careful comparison of these alternative definitions. One of the obstacles to this is that they are cast in terms of different semantic models. Rushby considers the definitions of Haigh and Young for two different state machine models with notions of action and observation, with observations associated in one model to states and in the other model to actions (akin to the Moore-Mealy distinction on finite state automata). The definitions of [vdM07] are stated using the state-observed version of Rushby's model. On the other hand, Roscoe and Goldsmith work in the process algebra CSP. They actually have two different definitions of security: one deals only with actions, the other is intended for systems in which observations (or in their nomenclature, 'signal events') are of concern.

We address this diversity by defining mappings between these different semantic models, and then studying how the definitions of intransitive noninterference are related under these mappings. There turns out to be a close correspondence between the variants of the range of definitions with respect to the state-observed and action-observed versions of state machines. We give a translation from the action-observed to the state-observed model that preserves all the definitions of Goguen and Meseguer, Haigh and Young and van der Meyden. Moreover, we show that results concerning unwinding and access control systems can be transferred from one domain to the other by means of this translation.

The relationship to the Roscoe and Goldsmith definitions is more complex. On one way of translating *state-observed* state machines to CSP, one of Roscoe and Goldsmith's definitions corresponds precisely to the classsical definition of noninterference intended for transitive systems (but applied to intransitive systems.) However, this notion does not take observations into account. It turns out that RG's second definition cannot be applied to the processes produced by this

first translation. We can, however, construct another translation from *action observed* state machines to CSP that does allow the application of RG's second definition — in this case we find that RG's definition corresponds a weaker notion than under the first translation.

The structure of the paper is as follows. The state-observed systems model is defined in Section 2, as well as the associated notions of noninterference due to Goguen and Mesguer, Haigh and Young and van der Meyden. Section 3 defines the action-observed state machine model, and presents a similar set of definitions of noninterference on this model. Section 4 shows how action-observed systems may be mapped to the state-observed model, and shows that the definitions of noninterference on the two models correspond precisely under this mapping. Moreover, it is shown in Section 5 that the related notions of (weak) unwinding and the closely related notion of access control interpretability, are also preserved under the mapping from state- to action-observed machines. This enables a transfer of some results of van der Meyden [vdM07] from one domain to the other. Finally, Section 6 considers mappings from state- and action-observed systems to a process algebraic model and characterizes Roscoe and Goldsmith's definitions under these mappings. Some remaining questions for future work are presented in Section 7.

2 State-Observed Systems

We begin by recalling the classical definitions of noninterference for transitive and intransitive policies [GM82,HY87,Rus92], and several new definitions proposed by van der Meyden [vdM07]. The present section considers these definitions on state observed machines; in the following section we treat similar definitions in action observed machines.

The *state-observed* machine model [Rus92] for these definitions consists of deterministic machines of the form $\langle S, s_0, A, \mathtt{step}, \mathtt{obs}, \mathtt{dom}\rangle$, where S is a set of states, $s_0 \in S$ is the *initial state*, A is a set of actions, $\mathtt{dom} : A \to D$ associates each action to an element of the set D of security domains, $\mathtt{step} : S \times A \to S$ is a deterministic transition function, and $\mathtt{obs} : S \times D \to O$ maps states to an observation in some set O, for each security domain. We may also refer to security domains more succinctly as "agents". We write $s \cdot \alpha$ for the state reached by performing the sequence of actions $\alpha \in \mathit{Actions}^*$ from state s, defined inductively by $s \cdot \epsilon = s$, and $s \cdot \alpha a = \mathtt{step}(s \cdot \alpha, a)$ for $\alpha \in A^*$ and $a \in A$. Here ϵ denotes the empty sequence.

Noninterference policies, the class of security policies we consider for these machines, are relations $\rightarrowtail \subseteq D \times D$. Intuitively, $u \rightarrowtail v$ means that "actions of domain u are permitted to interfere with domain v", or "information is permitted to flow from domain u to domain v". Since a domain should be allowed to interfere with, or have information about, itself, this relation is assumed to be reflexive.

Transitive noninterference policies have been given a formal semantics using a definition based on a "purge" function. Given a set $E \subseteq D$ of domains and a

sequence $\alpha \in A^*$, we write $\alpha \upharpoonright E$ for the subsequence of all actions a in α with $\mathtt{dom}(a) \in E$. Given a policy $\rightarrowtail$, we define the function $\mathtt{purge} : A^* \times D \to A^*$ by

$$\mathtt{purge}(\alpha, u) = \alpha \upharpoonright \{v \in D \mid v \rightarrowtail u\}.$$

For clarity, we may use subscripting of agent arguments of functions, writing, e.g., $\mathtt{purge}(\alpha, u)$ as $\mathtt{purge}_u(\alpha)$, and $\mathtt{obs}_u(s)$ for $\mathtt{obs}(s, u)$.

Definition 1. *A system M is* P-secure *with respect to a policy $\rightarrowtail$ if for all sequences $\alpha, \alpha' \in A^*$ such that $\mathtt{purge}_u(\alpha) = \mathtt{purge}_u(\alpha')$, we have $\mathtt{obs}_u(s_0 \cdot \alpha) = \mathtt{obs}_u(s_0 \cdot \alpha')$.*

This can be understood as saying that agent u's observation depends only on the sequence of interfering actions that have been performed. By idempotence of $\mathtt{purge}_u$, this definition is equivalent to the classical formulation, according to which the system M is secure when for all $\alpha \in A^*$ and domains $u \in D$, we have $\mathtt{obs}_u(s_0 \cdot \alpha) = \mathtt{obs}_u(s_0 \cdot \mathtt{purge}_u(\alpha))$. This formulation can be understood as saying that each agent's observations are as if only interfering actions had been performed. We note that we will apply P-security to intransitive policies as well as the transitive policies for which it was originally intended.

Haigh and Young [HY87] generalised the definition of the purge function to intransitive policies as follows. Intuitively, the intransitive purge of a sequence of actions with respect to a domain u is the largest subsequence of actions that could form part of a causal chain of effects (permitted by the policy) ending with an effect on domain u. More formally, the definition makes use of a function $\mathtt{sources} : A^* \times D \Rightarrow \mathcal{P}(D)$ defined inductively by $\mathtt{sources}(\epsilon, u) = \{u\}$ and

$$\mathtt{sources}(a\alpha, u) = \mathtt{sources}(\alpha, u) \cup \{\mathtt{dom}(a) \mid \exists v \in \mathtt{sources}(\alpha, u)(\mathtt{dom}(a) \rightarrowtail v)\}$$

for $a \in A$ and $\alpha \in A^*$. Intuitively, $\mathtt{sources}(\alpha, u)$ is the set of domains v such that there exists a sequence of permitted interferences from v to u within α. The *intransitive purge* function $\mathtt{ipurge} : A^* \times D \to A^*$ is then defined inductively by $\mathtt{ipurge}(\epsilon, u) = \epsilon$ and

$$\mathtt{ipurge}(a\alpha, u) = \begin{cases} a \cdot \mathtt{ipurge}(\alpha, u) & \text{if } \mathtt{dom}(a) \in \mathtt{sources}(a\alpha, u) \\ \mathtt{ipurge}(\alpha, u) & \text{otherwise} \end{cases}$$

for $a \in A$ and $\alpha \in A^*$. An alternative, equivalent formulation that we will find useful is the following: given a set $X \subseteq D$, define $\mathtt{ipurge}_X(\alpha)$ inductively by $\mathtt{ipurge}_X(\epsilon) = \epsilon$ and

$$\mathtt{ipurge}_X(\alpha a) = \begin{cases} \mathtt{ipurge}_{X \cup \{\mathtt{dom}(a)\}}(\alpha) \cdot a & \text{if } \exists u \in X(\mathtt{dom}(a) \rightarrowtail u) \\ \mathtt{ipurge}_X(\alpha) & \text{otherwise} \end{cases}$$

Then $\mathtt{ipurge}_u(\alpha)$ is identical to $\mathtt{ipurge}_{\{u\}}(\alpha)$.

Haigh and Young's [HY87] definition of security can then be formulated by using the intransitive purge function in place of the purge function:

Definition 2. *M is IP-secure with respect to a policy $\rightarrowtail$ if for all $u \in D$ and all sequences $\alpha, \alpha' \in A^*$ with $\mathtt{ipurge}_u(\alpha) = \mathtt{ipurge}_u(\alpha')$, we have $\mathtt{obs}_u(s_0 \cdot \alpha) = \mathtt{obs}_u(s_0 \cdot \alpha')$.*

It can be seen that $\mathtt{ipurge}_u(\alpha) = \mathtt{purge}_u(\alpha)$ when $\rightarrowtail$ is transitive, so IP-security is in fact a generalisation of the definition of security for transitive policies.

These definitions are critiqued in [vdM07], where several alternatives are proposed. All are based on a concrete model of the maximal amount of information that an agent may have after some sequence of actions has been performed, and state that an agent's observation may not give it more than this maximal amount of information. The definitions differ in the modelling of the maximal information, but all take the view that an agent increases its information either by performing an action or by receiving information transmitted by another agent.

In the first model of the maximal information, what is transmitted when an agent performs an action is information about the actions performed by other agents. The following definition expresses this in a weaker way than the ipurge function.

Given sets X and A, let the set $\mathcal{T}(X, A)$ be the smallest set containing X and such that if $x, y \in \mathcal{T}$ and $z \in A$ then $(x, y, z) \in \mathcal{T}$. Intuitively, the elements of $\mathcal{T}(X, A)$ are are binary trees with leaves labelled from X and interior nodes labelled from A.

Given a policy $\rightarrowtail$, define, for each agent $u \in D$, the function $\mathtt{ta}_u : A^* \rightarrow \mathcal{T}(\{\epsilon\}, A)$ inductively by $\mathtt{ta}_u(\epsilon) = \epsilon$, and, for $\alpha \in A^*$ and $a \in A$,

1. if $\mathtt{dom}(a) \not\rightarrowtail u$, then $\mathtt{ta}_u(\alpha a) = \mathtt{ta}_u(\alpha)$,
2. if $\mathtt{dom}(a) \rightarrowtail u$, then $\mathtt{ta}_u(\alpha a) = (\mathtt{ta}_u(\alpha), \mathtt{ta}_{\mathtt{dom}(a)}(\alpha), a)$.

Intuitively, $\mathtt{ta}_u(\alpha)$ captures the maximal information that agent u may, consistently with the policy $\rightarrowtail$, have about the past actions of other agents. (The nomenclature is intended to be suggestive of *transmission* of information about *actions*.) Initially, an agent has no information about what actions have been performed. The recursive clause describes how the maximal information $\mathtt{ta}_u(\alpha)$ permitted to u after the performance of α changes when the next action a is performed. If a may not interfere with u, then there is no change, otherwise, u's maximal permitted information is increased by adding the maximal information permitted to $\mathtt{dom}(a)$ at the time a is performed (represented by $\mathtt{ta}_{\mathtt{dom}(a)}(\alpha)$), as well the fact that a has been performed. Thus, this definition captures the intuition that an agent may only transmit information that it is permitted to have, and then only to agents with which it is permitted to interfere.

Definition 3. *A system M is TA-secure with respect to a policy $\rightarrowtail$ if for all agents u and all $\alpha, \alpha' \in A^*$ such that $\mathtt{ta}_u(\alpha) = \mathtt{ta}_u(\alpha')$, we have $\mathtt{obs}_u(s_0 \cdot \alpha) = \mathtt{obs}_u(s_0 \cdot \alpha')$.*

Intuitively, this says that each agent's observations provide the agent with no more than the maximal amount of information that may have been transmitted to it, as expressed by the functions $\mathtt{ta}$.

The second of van der Meyden's definitions uses the following notion of *view*. The definition uses an absorbtive concatenation function $\circ$, defined over a set X by, for $s \in X^*$ and $x \in X$, by $s \circ x = s$ if x is equal to the final element of s (if any), and $s \circ x = s \cdot x$ (ordinary concatenation) otherwise. Define the view of domain u with respect to a sequence $\alpha \in A^*$ using the function[1] $\mathtt{view}^s_u : A^* \to O(A \cup O)^*$ (where O is the set of observations in the system), defined by

$$\begin{aligned} \mathtt{view}^s_u(\epsilon) &= \mathtt{obs}_u(s_0), \text{ and} \\ \mathtt{view}^s_u(\alpha a) &= \begin{cases} \mathtt{view}^s_u(\alpha)\, a\, \mathtt{obs}_u(s_0 \cdot \alpha) & \text{if } \mathtt{dom}(a) = u, \\ \mathtt{view}^s_u(\alpha) \circ \mathtt{obs}_u(s_0 \cdot \alpha) & \text{otherwise} \end{cases} \end{aligned}$$

That is, $\mathtt{view}^s_u(\alpha)$ is the sequence of all observations and actions of domain u in the run generated by α, compressed by the elimination of stuttering observations. Intuitively, $\mathtt{view}^s_u(\alpha)$ is the complete record of information available to agent u in the run generated by the sequence of actions α. The absorbtive concatenation is intended to capture that the system is asynchronous, with agents not having access to a global clock. Thus, two periods of different length during which a particular observation obtains are not distinguishable to the agent.

Given a policy $\rightarrowtail$, for each domain $u \in D$, define the function $\mathtt{to}^s_u : A^* \to \mathcal{T}(O(A \cup O)^*, A)$ by $\mathtt{to}^s_u(\epsilon) = \mathtt{obs}_u(s_0)$ and

$$\mathtt{to}^s_u(\alpha a) = \begin{cases} \mathtt{to}^s_u(\alpha) & \text{if } \mathtt{dom}(a) \not\rightarrowtail u, \\ (\mathtt{to}^s_u(\alpha), \mathtt{view}_{\mathtt{dom}(a)}(\alpha), a) & \text{otherwise.} \end{cases}$$

Intuitively, this definition takes the model of the maximal information that an action a may transmit after the sequence α to be the fact that a has occurred, together with the information that $\mathtt{dom}(a)$ *actually* has, as represented by its view $\mathtt{view}_{\mathtt{dom}(a)}(\alpha)$. By contrast, TA-security uses in place of this the maximal information that $\mathtt{dom}(a)$ *may* have. (The nomenclature '$\mathtt{to}$' is intended to be suggestive of *transmission* of information about *observations*.)

We will also consider a slight variant of this definition. Given a policy $\rightarrowtail$, for each domain $u \in D$, define the function $\mathtt{ito}^s_u : A^* \to \mathcal{T}(O(A \cup O)^*, A)$ by $\mathtt{ito}^s_u(\epsilon) = \mathtt{obs}_u(s_0)$ and

$$\mathtt{ito}^s_u(\alpha a) = \begin{cases} \mathtt{ito}^s_u(\alpha) & \text{if } \mathtt{dom}(a) \not\rightarrowtail u, \\ (\mathtt{ito}^s_u(\alpha), \mathtt{view}_{\mathtt{dom}(a)}(\alpha), a) & \text{if } \mathtt{dom}(a) = u, \\ (\mathtt{ito}^s_u(\alpha), \mathtt{view}_{\mathtt{dom}(a)}(\alpha a), a) & \text{otherwise.} \end{cases}$$

This definition is just like that of $\mathtt{to}^s$, with the difference that the information that may be transmitted to u by an action a such that $\mathtt{dom}(a) \rightarrowtail u$ but $\mathtt{dom}(a) \neq u$, includes the observation $\mathtt{obs}_{\mathtt{dom}(a)}(s_0 \cdot \alpha a)$ obtained in domain $\mathtt{dom}(a)$ immediately after the occurrence of action a. Intuitively, the definition of security based on this notion will allow that the action a transmits not just the information observable to $\mathtt{dom}(a)$ at the time that it is invoked, but also the new

[1] We superscript some definitions for state-observed systems in this section by s in order to distinguish them from variants for action-observed systems, to be defined in the following section.

information that it computes and makes observable in $\mathtt{dom}(a)$. This information is not included in the value $\mathtt{ito}^s_{\mathtt{dom}(a)}(\alpha)$ itself, since the definition of security will state that the the new observation may depend only on this value. The nomenclature in this case is intended to be suggestive of *immediate* transmission of information about observations.

We may now base the definition of security on either the function $\mathtt{to}^s$ or $\mathtt{ito}^s$ rather than $\mathtt{ta}$.

Definition 4. *The system M is TO-secure with respect to $\rightarrowtail$ if for all domains $u \in D$ and all $\alpha, \alpha' \in A^*$ with $\mathtt{to}^s_u(\alpha) = \mathtt{to}^s_u(\alpha')$, we have $\mathtt{obs}_u(s_0 \cdot \alpha) = \mathtt{obs}_u(s_0 \cdot \alpha')$.*

The system M is ITO-secure with respect to $\rightarrowtail$ if for all domains $u \in D$ and all $\alpha, \alpha' \in A^*$ with $\mathtt{ito}^s_u(\alpha) = \mathtt{ito}^s_u(\alpha')$, we have $\mathtt{obs}_u(s_0 \cdot \alpha) = \mathtt{obs}_u(s_0 \cdot \alpha')$.

The following result shows how these definitions are related:

Theorem 1 ([vdM07]). *For state-observed systems, with respect to a given policy $\rightarrowtail$, P-security implies TO-security implies ITO-security implies TA-security implies IP-security.*

Examples showing that all these notions are distinct are presented in [vdM07].

3 Action Observed Systems

The definitions of the previous section are concerned with a state-observed machine model. Rushby [Rus92] also deals with a variant of the machine model in which observation are associated to actions rather than states. In this section we describe a formulation of our definitions in this model, and show how these variants are related to the state-observed versions.

An *action-observed* machine is a tuple $\langle S, s_0, A, \mathtt{step}, \mathtt{out}, \mathtt{dom}\rangle$, where all the components are as in the state observed system model, except that the observation function $\mathtt{obs}$ is replaced by a function $\mathtt{out} : S \times A \to O$. Intuitively, if s is a state and a is an action, then $\mathtt{out}(s, a)$ is the observation made in domain $\mathtt{dom}(a)$ when action a is performed.

The notions of P-security and IP-security are defined by Rushby for action-observed machines by

1. M is P-secure with respect to a policy $\rightarrowtail$ if for all $\alpha, \alpha' \in A^*$, $u \in D$ and $a \in A$ with $\mathtt{dom}(a) = u$, if $\mathtt{purge}_u(\alpha) = \mathtt{purge}_u(\alpha')$ then $\mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{out}(s_0 \cdot \alpha', a)$.
2. M is IP-secure with respect to a policy $\rightarrowtail$ if for all $\alpha, \alpha' \in A^*$, $u \in D$ and $a \in A$ with $\mathtt{dom}(a) = u$, if $\mathtt{ipurge}_u(\alpha) = \mathtt{ipurge}_u(\alpha')$ then $\mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{out}(s_0 \cdot \alpha', a)$.

We can also adapt TO-security, ITO-security and TA-security to the action observed system model. First, the notion of view is adapted to the action-observed system model by defining $\mathtt{view}^a_u : A^* \to (A \cup O)^*$ for $u \in D$ inductively

by $\mathtt{view}^a_u(\epsilon) = \epsilon$, and

$$\mathtt{view}^a_u(\alpha a) = \begin{cases} \mathtt{view}^a_u(\alpha) \cdot a \cdot \mathtt{out}(s_0 \cdot \alpha, a) & \text{if } \mathtt{dom}(a) = u \\ \mathtt{view}^a_u(\alpha) & \text{otherwise.} \end{cases}$$

That is, the view of an agent is just the sequence of actions that the agent has performed, together with the outputs obtained from those actions.

We may also define for the action-observed model variants $\mathtt{to}^a_u, \mathtt{ito}^a_u : A^* \to \mathcal{T}((A \cup O)^*, A)$ of the functions $\mathtt{to}^s_u$ and $\mathtt{ito}^s_u$ used in the definitions of TO-security and ITO-security.. We define $\mathtt{to}^a_u$ by $\mathtt{to}^a_u(\epsilon) = \epsilon$ and

$$\mathtt{to}^a_u(\alpha a) = \begin{cases} \mathtt{to}^a_u(\alpha) & \text{if } \mathtt{dom}(a) \not\rightarrowtail u, \\ (\mathtt{to}^a_u(\alpha), \mathtt{view}^a_{\mathtt{dom}(a)}(\alpha a), a) & \text{if } \mathtt{dom}(a) = u, \\ (\mathtt{to}^a_u(\alpha), \mathtt{view}^a_{\mathtt{dom}(a)}(\alpha), a) & \text{if } u \neq \mathtt{dom}(a) \rightarrowtail u. \end{cases}$$

Similarly, $\mathtt{ito}^a_u$ is defined by ${}^a_u(\epsilon) = \epsilon$ and

$$\mathtt{ito}^a_u(\alpha a) = \begin{cases} \mathtt{ito}^a_u(\alpha) & \text{if } \mathtt{dom}(a) \not\rightarrowtail u, \\ (\mathtt{ito}^a_u(\alpha), \mathtt{view}^a_{\mathtt{dom}(a)}(\alpha a), a) & \text{otherwise.} \end{cases}$$

We can then formulate the definitions of security of Section 2 on action-observed machines M by

1. M is TA-secure with respect to $\rightarrowtail$ if for all $\alpha, \alpha' \in A^*$, $u \in D$ and $a \in A$ with $\mathtt{dom}(a) = u$, if $\mathtt{ta}_u(\alpha) = \mathtt{ta}_u(\alpha')$ then $\mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{out}(s_0 \cdot \alpha', a)$.
2. M is TO-secure with respect to $\rightarrowtail$ if for all $\alpha, \alpha' \in A^*$, $u \in D$ and $a \in A$ with $\mathtt{dom}(a) = u$, if $\mathtt{to}^a_u(\alpha) = \mathtt{to}^a_u(\alpha')$ then $\mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{out}(s_0 \cdot \alpha', a)$.
3. M is ITO-secure with respect to $\rightarrowtail$ if for all $\alpha, \alpha' \in A^*$, $u \in D$ and $a \in A$ with $\mathtt{dom}(a) = u$, if $\mathtt{ito}^a_u(\alpha) = \mathtt{ito}^a_u(\alpha')$ then $\mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{out}(s_0 \cdot \alpha', a)$.

There is a subtle difference in the definition of $\mathtt{to}^a_u$ to what one might have expected, based on the definition of $\mathtt{to}^s_u$. Note that whereas there was a single case for $\mathtt{to}^s_u(\alpha a)$ when $\mathtt{dom}(a) \rightarrowtail u$, the definition of $\mathtt{to}^a_u$ breaks this into two cases, depending on whether $\mathtt{dom}(a) = u$. In case $\mathtt{dom}(a) = u$, agent u is treated as receiving the information $\mathtt{view}^a_{\mathtt{dom}(a)}(\alpha a)$ as a result of performing the action, whereas other agents with which domain $\mathtt{dom}(a)$ may interfere are treated as receiving the information $\mathtt{view}^a_{\mathtt{dom}(a)}(\alpha)$. That is, agent $\mathtt{dom}(a)$ is considered to have received the output resulting from performing action a, but this output is not included in the information considered by $\mathtt{to}^a$ to have been transmitted to other agents.[2]

The reason for this difference with the definition of $\mathtt{to}^s_u$ is that whereas $\mathtt{to}^s_u(\alpha)$ is used in the definition of TO-security to state what agent u's observation of state $s_0 \cdot \alpha$ may depend upon, $\mathtt{to}^a_u(\alpha)$ is used to state what the output of a

[2] There is some obvious redundancy in the definition, and we could use just $\mathtt{out}(s_0 \cdot \alpha, a)$ in place of $\mathtt{view}^a_{\mathtt{dom}(a)}(\alpha a)$ in the case $\mathtt{dom}(a) = u$, but we keep the definition in the present form to facilitate comparison.

further action a in state $s_0 \cdot \alpha$ may depend upon. Thus, we include all outputs from u's actions in the sequence αa in the case $\mathtt{dom}(a) = u$.

As in the state-observed case, the definition of ITO-security is intended to allow that new observable information computed by an action a is transmitted to other agents with which $\mathtt{dom}(a)$ may interfere. This is captured by transmitting $\mathtt{view}^a_{\mathtt{dom}(a)}(\alpha a)$ to such agents.

These definitions of security are related exactly like their variants for the state observed model.

Theorem 2. *For the action-observed definitions, with respect to a given policy $\rightarrowtail$, P-security implies TO-security implies ITO-security implies TA-security implies IP-security. These implications are all strict.*

We defer the proof of the implications to the next section. For strictness of the implications we have the following examples.

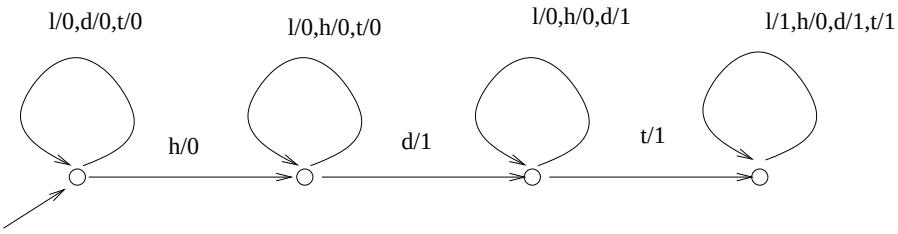

Fig. 1. TO-secure but not P-secure.

To see that TO-security does not imply P-security, consider the system M in Figure 1, where there are domains H, D, L with sets of actions $\{h\}$, $\{d, t\}$ and $\{l\}$, respectively. Let the policy be given by $H \rightarrowtail D \rightarrowtail L$. Intuitively, d tests whether action h has occurred, and t transmits to L the fact that h has occurred, once this fact is known to D. The action l tests whether this fact has been transmitted to L. This system is TO-secure. Since all outputs to H are 0, there is nothing to check for H. In case of D, we check the contrapositive of the definition. Note that if we have $\mathtt{out}(s_0 \cdot \alpha, d) = 0$ and $\mathtt{out}(s_0 \cdot \alpha', d) = 1$, then α does not contain h but α' does contain h. But since $H \rightarrowtail D$, it then follows that $\mathtt{to}^a_D(\alpha')$ contains an h but $\mathtt{to}^a_D(\alpha')$ does not. Similarly, if $\mathtt{out}(s_0 \cdot \alpha, t) = 0$ and $\mathtt{out}(s_0 \cdot \alpha', t) = 1$, then α' contains a h followed by a d, but α does not. In either case, it follows that $\mathtt{to}^a_D(\alpha') \neq \mathtt{to}^a_D(\alpha')$. Similarly, for L, if $\mathtt{out}(s_0 \cdot \alpha, l) = 0$ and $\mathtt{out}(s_0 \cdot \alpha', l) = 1$, then α' contains an h, a later d which gives output 1, and a still later t, but α' does not contain such a subsequence. In particular, it follows that $\mathtt{to}^a_L(\alpha')$ contains a view component containing a subsequence of the form $d1 \ldots t$, but $\mathtt{to}^a_L(\alpha')$ does not. Hence $\mathtt{to}^a_L(\alpha') \neq \mathtt{to}^a_L(\alpha')$. This completes the argument that the system is TO-secure. To see that it is not P-secure, note that $\mathtt{purge}_L(hdt) = dt = \mathtt{purge}_L(dt)$ but $\mathtt{out}(s_0 \cdot hdt, l) = 1$ and $\mathtt{out}(s_0 \cdot dt, l) = 0$.

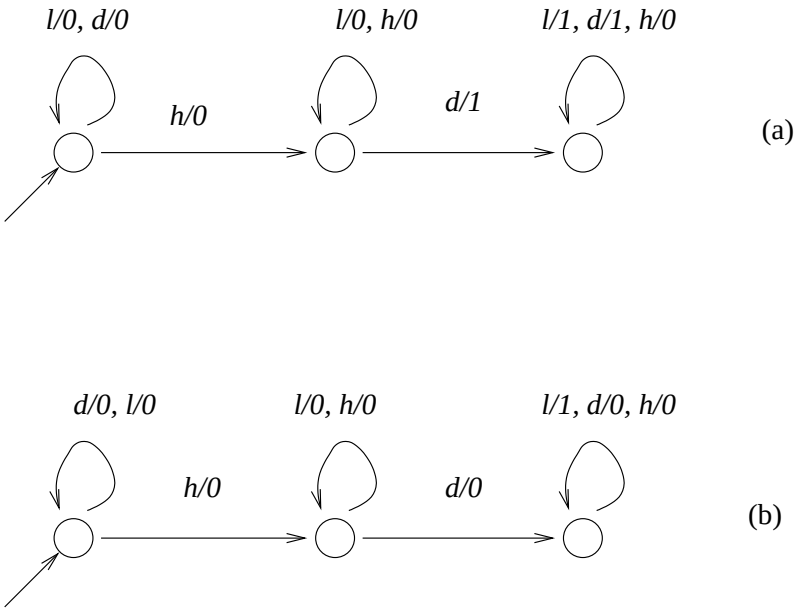

Fig. 2. (a) ITO-secure but not TO-secure, (b) TA-secure, but not ITO-secure.

To see that that ITO-security does not imply TO-security, consider the system M in Figure 2(a), where we assume that there are actions h, d, l of domains H, D, L, respectively. Let the policy be given by $H \rightarrowtail D \rightarrowtail L$. Intuitively, the action d tests whether there has been an occurrence of h, and immediately transmits this information to L if so, where it can be observed via the action l. The system M is ITO-secure. To see this, note that $\mathtt{out}(s_0 \cdot \alpha, l) = 1$ iff α can be written as $\alpha_1 d \alpha_2$ and $\mathtt{out}(s_0 \cdot \alpha_1, d) = 1$. Since $D \rightarrowtail L$, this in turn means that the occurrence of d with output 1 is apparent in $\mathtt{ito}_L(\alpha)$. It follows that $\mathtt{ito}_L(\alpha) = \mathtt{ito}_L(\alpha')$ implies $\mathtt{out}(s_0 \cdot \alpha, l) = \mathtt{out}(s_0 \cdot \alpha', l)$. For domains H and D, the definition of ITO-security is trivial from the fact that there is just one possible observation. To see that M is not TO-secure, note that $\mathtt{to}_L(d) = (\epsilon, \epsilon, d) = (\mathtt{to}_L(h), \mathtt{view}^a_D(h), d) = \mathtt{to}_L(hd)$, but $\mathtt{out}(s_0 \cdot d, l) = 0$ and $\mathtt{out}(s_0 \cdot hd, l) = 1$.

To show that TA-security does not imply ITO-security, consider the system in Figure 2(b) with actions h, d, l of domains H, D, L respectively, and the policy $H \rightarrowtail D \rightarrowtail L$. In this system, action d informs L whether there has been an occurrence of h, although this information is never known to D itself. This system is not ito-secure, for $\mathtt{ito}_L(d) = (\epsilon, d0, d) = (\mathtt{ito}_L(h), \mathtt{view}^a_D(hd), d) = \mathtt{ito}_L(hd)$, but $\mathtt{out}(s_0 \cdot d, l) = 0$ and $\mathtt{out}(s_0 \cdot hd, l) = 1$. However, this system is TA-secure. For, if $\mathtt{obs}_L(s_0 \cdot \alpha) \neq \mathtt{obs}_L(s_0 \cdot \alpha')$, then one of α, α' contains h and a later d, and the other does not. It follows that $\mathtt{ta}_L(\alpha) \neq \mathtt{ta}_L(\alpha')$. The definition of TA-security holds trivially for the domains H and D since they have only one possible observation.

Finally, to show that IP-security does not imply TA-security, consider an example similar to that in [vdM07]. Consider domains H_1, H_2, D_1, D_2, L, with actions $A = \{h_1, h_2, d_1, d_2, l\}$, respectively. Let the policy by given by $H_i \rightarrowtail D_i \rightarrowtail L$ for $i = 1, 2$. Let the set of states of the system be A^*, initial state ϵ, the step function be given by $\mathtt{step}(\alpha, a) = \alpha a$ for $\alpha \in A^*$ and $a \in A$, and let outputs be given by $\mathtt{out}(\alpha, l) = \mathtt{ipurge}_L(\alpha)$, and $\mathtt{out}(\alpha, a) = \perp$ for all other actions a. This system is trivially IP-secure. On the other hand, $\mathtt{ta}_L(h_1h_2d_1d_2) = \mathtt{ta}_L(h_2h_1d_1d_2)$ and $\mathtt{ipurge}_L(h_1h_2d_1d_2) = h_1h_2d_1d_2 \neq h_2h_1d_1d_2 = \mathtt{ipurge}_L(h_2h_1d_1d_2)$. Hence $\mathtt{out}(s_0{\cdot}h_1h_2d_1d_2, l) \neq \mathtt{out}(s_0{\cdot}h_2h_1d_1d_2, l)$, so the system is not TA-secure.

We have used a uniform format based on intuitions concerning maximal message passing to give the definitions of TA-security, TO-security and ITO-security. It will be useful in what follows to have the following simpler characterization of ITO-security.

Define the set of observations for an agent u in a system M to be the set of outputs obtainable by the agent, i.e., $O_u = \{\mathtt{out}(s, a) \mid s \in S, a \in A_u\}$. Say that a system has *disjoint observations* if for all agents $u \neq v$, we have $O_u \cap O_v = \emptyset$. A system may always be transformed into one with disjoint observations simply by renaming observations. It is easy to show that this does not affect the satisfaction of any of the definitions of security.

Given a sequence $\alpha = a_1 \dots a_n \in A^*$, write $\mathtt{trace}(\alpha)$ for the sequence $a_1o_1a_2o_2 \dots a_no_n$ where $o_k = \mathtt{out}(s_0 \cdot (a_1 \dots a_{k-1}), a_k)$, for $k = 1 \dots n$. Given a policy $\rightarrowtail$, and supposing that the system has disjoint observations, for each agent u, define the set $\mathtt{nf}(u) = \bigcup_{v \not\rightarrowtail u} A_v \cup O_v$. That is, this is the set of all actions and observations of agents that may not interfere with u. Given a sequence σ of actions and observations, write $\sigma \setminus \mathtt{nf}(u)$ for the subsequence of all actions and observations not in $\mathtt{nf}(u)$. This is the sequence of all actions and observations of agents that may interfere with u.

Proposition 1. *Suppose that M is a system with disjoint observations. Then M is ITO-secure with respect to $\rightarrowtail$ iff for all $\alpha, \alpha' \in A^*$, agents u and actions $a \in A_u$, if $\mathtt{trace}(\alpha)\setminus\mathtt{nf}(u) = \mathtt{trace}(\alpha')\setminus\mathtt{nf}(u)$ then $\mathtt{out}(s_0{\cdot}\alpha, a) = \mathtt{out}(s_0{\cdot}\alpha', a)$.*

Proof. We show by induction on the combined length of α and α' that $\mathtt{ito}_u(\alpha) = \mathtt{ito}_u(\alpha')$ iff $\mathtt{trace}(\alpha) \setminus \mathtt{nf}(u) = \mathtt{trace}(\alpha') \setminus \mathtt{nf}(u)$. The claim is trivial for $\alpha = \alpha' = \epsilon$. Consider sequences αa and α', where $\alpha, \alpha' \in A^*$ and $a \in A$, and the claim holds for sequences of shorter combined length. If $\mathtt{dom}(a) \not\rightarrowtail u$, then the result follows straightforwardly from the induction hypothesis and the facts that $\mathtt{ito}_u(\alpha a) = \mathtt{ito}_u(\alpha)$ and $\mathtt{trace}(\alpha a) \setminus \mathtt{nf}(u) = \mathtt{trace}(\alpha) \setminus \mathtt{nf}(u)$.

Suppose therefore that $\mathtt{dom}(a) \rightarrowtail u$ and $\mathtt{ito}_u(\alpha a) = \mathtt{ito}_u(\alpha')$. Then α' cannot be ϵ, and we may assume without loss of generality that the last action b in α' has $\mathtt{dom}(b) \rightarrowtail u$, else we may apply the previous case. It follows that $b = a$, and we may write $\alpha' = \beta a$. Hence $\mathtt{ito}_u(\alpha) = \mathtt{ito}_u(\beta)$ and $\mathtt{view}^a_{\mathtt{dom}(a)}(\alpha a) = \mathtt{view}^a_{\mathtt{dom}(a)}(\beta a)$. From the former it follows by the induction hypothesis that $\mathtt{trace}(\alpha) \setminus \mathtt{nf}(u) = \mathtt{trace}(\beta) \setminus \mathtt{nf}(u)$, and from the latter we obtain that $\mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{out}(s_0 \cdot \beta, a)$ – write o for this output value. Thus $\mathtt{trace}(\alpha a) \setminus \mathtt{nf}(u) = (\mathtt{trace}(\alpha) \setminus \mathtt{nf}(u))ao = (\mathtt{trace}(\beta) \setminus \mathtt{nf}(u))ao = (\mathtt{trace}(\beta a) \setminus \mathtt{nf}(u))$, as required.

Conversely, suppose that $\mathtt{dom}(a) \rightarrowtail u$ and $\mathtt{trace}(\alpha a) \setminus \mathtt{nf}(u) = \mathtt{trace}(\alpha') \setminus \mathtt{nf}(u)$. Then a must occur in α', and we may assume that it is the last action, else we may apply the case for $\mathtt{dom}(a) \not\rightarrowtail u$. So write $\alpha' = \beta a$. We then have that $\mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{out}(s_0 \cdot \beta, a)$ – write o for this output value – and also that $\mathtt{trace}(\alpha) \setminus \mathtt{nf}(u) = \mathtt{trace}(\beta) \setminus \mathtt{nf}(u)$. By the induction hypothesis, $\mathtt{ito}_u(\alpha) = \mathtt{ito}_u(\beta)$. Moreover,

$$\begin{aligned}
\mathtt{view}^a_{\mathtt{dom}(a)}(\alpha a) &= \mathtt{trace}(\alpha a) \restriction (A_{\mathtt{dom}(a)} \cup O_{\mathtt{dom}(a)}) \\
&= (\mathtt{trace}(\alpha a) \setminus \mathtt{nf}(u)) \restriction (A_{\mathtt{dom}(a)} \cup O_{\mathtt{dom}(a)}) \\
&= (\mathtt{trace}(\beta a) \setminus \mathtt{nf}(u)) \restriction (A_{\mathtt{dom}(a)} \cup O_{\mathtt{dom}(a)}) \\
&= \mathtt{view}^a_{\mathtt{dom}(a)}(\alpha a).
\end{aligned}$$

It follows that $\mathtt{ito}_u(\alpha a) = \mathtt{ito}_u(\beta a)$, as required. □

4 Mapping Action-Observed to State-Observed Systems

In the previous sections we have defined a spectrum of definitions of security for state- and action-observed systems. We now consider the relationships between these definitions on the two models under a transformation from action-observed to state-observed systems. We show that our usage of common names for definitions on the two different models is justified, by showing that similarly named notions of security correspond under this transformation. As a consequence of this result, we may derive Theorem 2 from Theorem 1.

Given an action-observed machine $M = \langle S, s_0, A, \mathtt{step}, \mathtt{out}, \mathtt{dom}\rangle$, define the state observed machine $F_{as}(M) = \langle S', s'_0, A, \mathtt{step}', \mathtt{obs}, \mathtt{dom}\rangle$ by

1. $S' = S \times (D \rightarrow O \cup \{\bot\})$,
2. $s'_0 = (s_0, f_0)$, where $f_0(d) = \bot$ for all $d \in D$,
3. $\mathtt{step}'((s, f), a)) = (\mathit{step}(s, a), f[\mathtt{dom}(a) \rightarrow \mathtt{out}(s, a)])$,
4. $\mathtt{obs}_u((s, f)) = f(u)$.

Here, we write $f[u \mapsto x]$ for the function f' that is identical to f except that $f'(u) = x$. Intuitively, in a state (s, f), the value $f(d)$ for a domain d represents the observation most recently obtained in domain d, and is $\bot$ if there has been no observation in domain d.

The following result states relationships between the definitions of security on the two types of model under this mapping.

Theorem 3. *Let X be any of P, TO, ITO, TA, or IP. Then an action-observed machine M is X-secure (with respect to the action-observed definitions) iff $F_{as}(M)$ is X-secure (with respect to the state-observed definitions).*

Thus, there is a direct correspondence between similarly named definitions on the two models, and the use of a common nomenclature is justified. Theorem 2 follows immediately from Theorem 3 and Theorem 1. For the proof of Theorem 3, it is useful to identify a number of properties of a generalization of the notions of security that we are considering. Let $X_u : A^* \rightarrow Z$, for $u \in D$, be a family

of functions mapping sequences of actions to some set Z. We say that a state-observed system M is X-secure when for all $\alpha, \alpha' \in A^*$ and $u \in D$, if $X_u(\alpha) = X_u(\alpha')$ then $\mathtt{obs}_u(s_0 \cdot \alpha) = \mathtt{obs}_u(s_0 \cdot \alpha')$. Similarly, an action-observed system M is X-secure when for all $\alpha, \alpha' \in A^*$ and $u \in D$ and $a \in A$ with $\mathtt{dom}(a) = u$, if $X_u(\alpha) = X_u(\alpha')$ then $\mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{out}(s_0 \cdot \alpha', a)$.

It can be seen that the definitions of P-security, TA-security and IP-security in both state- and action-observed systems correspond precisely to X-security with respect to the functions `purge`, `ta` and `ipurge`, respectively. In the case of TO-security and ITO-security, the definitions of the view functions, hence the functions `to` and `ito`, differ somewhat in the action and state observed cases, so it is not immediate that a similar uniform characterization applies. We discuss this case below.

Consider the following properties of the functions X:

(Consistency with $\rightarrowtail$) If $dom(a) \not\rightarrowtail u$ then $X_u(\alpha a) = X_u(\alpha)$;

(Nontriviality) If $\mathtt{dom}(a) \rightarrowtail u$ then $X_u(\alpha a) \neq X_u(\epsilon)$;

(Additivity) If $\mathtt{dom}(a) \rightarrowtail u$ and $\mathtt{dom}(b) \rightarrowtail u$ and $X_u(\alpha a) = X_u(\alpha' b)$ then $a = b$ and $X_u(\alpha) = X_u(\alpha')$;

(Locality) If $\mathtt{dom}(a) = u$ and $X_u(\alpha) = X_u(\alpha')$ then $X_u(\alpha a) = X_u(\alpha' a)$;

These abstract properties are useful to establish a transfer of X-security between an action-observed system M and the system $F_{as}(M)$.

Lemma 1. *Let M be an action-observed system.*

1. *If X is consistent with $\rightarrowtail$, nontrivial and additive, and M is X-secure then $F_{as}(M)$ is X-secure.*
2. *If X satisfies locality and $F_{as}(M)$ is X-secure then M is X-secure.*

Proof. An easy induction shows that $s'_0 \cdot \alpha = (s_0 \cdot \alpha, f_\alpha)$, where for each $u \in D$, if $\alpha = \beta a \gamma$, where a is the rightmost element of domain u in α, then $f_\alpha(u) = out(s_0 \cdot \beta, a)$, and $f_\alpha(u) = \perp$ if there is no such a.

For part (1), assume that M is X-secure and that X is consistent with $\rightarrowtail$, nontrivial and additive. We show that $F_{as}(M)$ is X-secure, i.e., that for all $u \in D$ and $\alpha, \alpha' \in A^*$, if $X_u(\alpha) = X_u(\alpha')$ then $\mathtt{obs}_u(s'_0 \cdot \alpha) = \mathtt{obs}_u(s'_0 \cdot \alpha')$. The proof is by induction on the combined length of α and α'. The base case of $\alpha = \alpha' = \epsilon$ is trivial. Consider sequences αa and α', where $a \in A$ and $X_u(\alpha a) = X_u(\alpha')$, such that the claim holds for sequences of shorter combined length. We have to show that $\mathtt{obs}_u(s'_0 \cdot \alpha a) = \mathtt{obs}_u(s'_0 \cdot \alpha')$. There are two cases, depending on whether $\mathtt{dom}(a) \rightarrowtail u$.

1. Suppose $\mathtt{dom}(a) \not\rightarrowtail u$. Then by consistency with $\rightarrowtail$ we have $X_u(\alpha) = X_u(\alpha a) = X_u(\alpha')$. Hence, by the induction hypothesis, $\mathtt{obs}_u(s'_0 \cdot \alpha) = \mathtt{obs}_u(s'_0 \cdot \alpha')$. It also follows from $\mathtt{dom}(a) \not\rightarrowtail u$ that $\mathtt{dom}(a) \neq u$. Thus, by construction of $F_{as}(M)$, we have $\mathtt{obs}_u(s'_0 \cdot \alpha a) = \mathtt{obs}_u(s'_0 \cdot \alpha)$. Thus, $\mathtt{obs}_u(s'_0 \cdot \alpha) = \mathtt{obs}_u(s'_0 \cdot \alpha')$, as required.

2. Suppose $\mathtt{dom}(a) \rightarrowtail u$. By nontriviality, we get from $X_u(\alpha a) = X_u(\alpha')$ that α' is not ϵ. If the last action b in α' does not satisfy $\mathtt{dom}(b) \rightarrowtail u$, then we can swap αa and α' and apply the previous case. We may therefore assume that $\alpha' = \beta b$ for some b with $\mathtt{dom}(b) \rightarrowtail u$. It now follows from additivity that $a = b$ and $X_u(\alpha) = X_u(\beta)$. We obtain from this by the induction hypothesis that $\mathtt{obs}_u(s'_0 \cdot \alpha) = \mathtt{obs}_u(s'_0 \cdot \beta)$, and by the X-security of M that $\mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{out}_u(s_0 \cdot \beta, a)$. In case $\mathtt{dom}(a) \neq u$, it follows from the former that $\mathtt{obs}_u(s'_0 \cdot \alpha a) = \mathtt{obs}_u(s'_0 \cdot \alpha) = \mathtt{obs}_u(s'_0 \cdot \beta) = \mathtt{obs}_u(s'_0 \cdot \beta a)$. In case $\mathtt{dom}(a) = u$, we have from the latter that $\mathtt{obs}_u(s'_0 \cdot \alpha a) = \mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{out}_u(s_0 \cdot \beta, a) = \mathtt{obs}_u(s'_0 \cdot \beta a)$.

This completes the proof that $F_{as}(M)$ is X-secure .

For part (2), suppose that $F_{as}(M)$ is X-secure, and that X satisfies locality. Let $\alpha, \alpha' \in A^*$ and $u \in D$ be such that $X_u(\alpha) = X_u(\alpha')$. We show that for $a \in A$ with $\mathtt{dom}(a) = u$ that $\mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{out}_u(s_0 \cdot \alpha', a)$. Since $\mathtt{dom}(a) = u$, we have by locality that $X_u(\alpha a) = X_u(\alpha' a)$. Hence by X-security of $F_{as}(M)$ and construction of this system, we have $\mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{obs}_u(s'_0 \cdot \alpha a) = \mathtt{obs}_u(s'_0 \cdot \alpha' a) = \mathtt{out}(s_0 \cdot \alpha', a)$, as required for X-security of M. □

Lemma 2. *The families of functions* `purge`, `ta` *and* `ipurge` *satisfy consistency with* $\rightarrowtail$*, nontriviality, additivity and locality.*

Proof. In most cases, the proof is an easy check. We show just the case of additivity for `ipurge`. Suppose that $\mathtt{dom}(a) \rightarrowtail u$, $\mathtt{dom}(b) \rightarrowtail u$ and $\mathtt{ipurge}_u(\alpha a) = \mathtt{ipurge}_u(\alpha' b)$. Then $\mathtt{ipurge}_{\{u,\mathtt{dom}(a)\}}(\alpha)a = \mathtt{ipurge}_{\{u,\mathtt{dom}(b)\}}(\alpha')b$, so we must have $a = b$. Consequently, $\mathtt{dom}(a) = \mathtt{dom}(b)$, and we have $\mathtt{ipurge}_{\{u,\mathtt{dom}(a)\}}(\alpha) = \mathtt{ipurge}_{\{u,\mathtt{dom}(a)\}}(\alpha')$. We now use the fact that if $X \subseteq Y$ then $\mathtt{ipurge}_X(\mathtt{ipurge}_Y(\beta)) = \mathtt{ipurge}_X(\beta))$ to conclude that $\mathtt{ipurge}_u(\alpha) = \mathtt{ipurge}_{\{u\}}(\mathtt{ipurge}_{\{u,\mathtt{dom}(a)\}}(\alpha)) = \mathtt{ipurge}_{\{u\}}(\mathtt{ipurge}_{\{u,\mathtt{dom}(a)\}}(\alpha')) = \mathtt{ipurge}_u(\alpha')$. □

Theorem 3 is immediate from Lemma 1 and Lemma 2 for the cases of X equal to one of `purge`, `ta` and `ipurge`. For the cases of TO-security and ITO-security, we show the result directly. The following lemma is helpful for the proof.

Lemma 3. *Given a policy* $\rightarrowtail$*, for* $\alpha \in A^*$*, let* $\mathtt{view}^a_u(\alpha)$ *and* $\mathtt{to}^a_u(\alpha)$ *and* $\mathtt{ito}^a_u(\alpha)$ *be as defined in an action-observed system* M *and let* $\mathtt{view}^s_u(\alpha)$ *and* $\mathtt{to}^s_u(\alpha)$ $\mathtt{ito}^s_u(\alpha)$ *be be as defined in the corresponding system* $F_{as}(M)$*. Then for all* $\alpha, \beta \in A^*$*, we have*

(1) $\mathtt{view}^s_u(\alpha) = \bot\mathtt{view}^a_u(\alpha)$*, and*
(2) $\mathtt{to}^a_u(\alpha) = \mathtt{to}^a_u(\beta)$ *implies* $\mathtt{to}^s_u(\alpha) = \mathtt{to}^s_u(\beta)$*, and*
(3) $\mathtt{ito}^a_u(\alpha) = \mathtt{ito}^a_u(\beta)$ *implies* $\mathtt{ito}^s_u(\alpha) = \mathtt{ito}^s_u(\beta)$*.*

Proof. The proof for claim (1) is by induction on α. In the base case, we have $\mathtt{view}^a_u(\epsilon) = \epsilon$ and $\mathtt{view}^s_u(\epsilon) = \bot$, so the claim holds. Consider a sequence $\alpha \in A^*$ for which the claim holds, and let $a \in A$. We show that $\mathtt{view}^s_u(\alpha a) = \bot\mathtt{view}^a_u(\alpha a)$. There are two cases, depending on whether $\mathtt{dom}(a) = u$.

1. If $\mathtt{dom}(a) = u$, then we have $\mathtt{view}^s_u(\alpha a) = \mathtt{view}^s_u(\alpha)\, a\, O_u(s'_0 \cdot \alpha a)$ and $\mathtt{view}^a_u(\alpha a) = \mathtt{view}^a_u(\alpha)\, a\, \mathtt{out}(s_0 \cdot \alpha, a)$. The claim is now immediate from the induction hypothesis and the fact that $O_u(s'_0 \cdot \alpha a) = \mathtt{out}(s_0 \cdot \alpha, a)$, by construction of $F_{as}(M)$.
2. If $\mathtt{dom}(a) \neq u$, then we have $\mathtt{view}^s_u(\alpha a) = \mathtt{view}^s_u(\alpha) \circ O_u(s_0 \cdot \alpha a)$. Let β be the largest prefix of α ending in an action b with $\mathtt{dom}(b) = u$, or, if no such action exists, let $\beta = \epsilon$. By construction of $F_{as}(M)$, for all $\gamma \in A^*$ such that $\beta \leq \gamma \leq \alpha a$ we have $O_u(s'_0 \cdot \beta) = O_u(s'_0 \cdot \gamma)$. Hence $\mathtt{view}^s_u(\alpha a) = \mathtt{view}^s_u(\alpha)$. By the induction hypothesis, we have $\mathtt{view}^s_u(\alpha) = \bot\mathtt{view}^a_u(\alpha)$, and, since $\mathtt{dom}(a) \neq u$, we have $\mathtt{view}^a_u(\alpha) = \mathtt{view}^a_u(\alpha a)$. Thus, $\mathtt{view}^s_u(\alpha a) = \bot\mathtt{view}^a_u(\alpha a)$.

For part (2), we also proceed by induction. The base case of $\alpha = \beta = \epsilon$ is trivial. Consider sequences αa and β, where $a \in A$ and the claim holds for sequences of shorter combined length. Suppose that $\mathtt{to}^a_u(\alpha a) = \mathtt{to}^a_u(\beta)$. We have to show $\mathtt{to}^s_u(\alpha a) = \mathtt{to}^s_u(\beta)$,

In case $\mathtt{dom}(a) \not\rightarrowtail u$, we have $\mathtt{to}^a_u(\alpha a) = \mathtt{to}^a_u(\alpha)$, so $\mathtt{to}^a_u(\alpha) = \mathtt{to}^a_u(\beta)$. By the induction hypothesis, we obtain $\mathtt{to}^s_u(\alpha) = \mathtt{to}^s_u(\beta)$. But, in this case $\mathtt{to}^s_u(\alpha a) = \mathtt{to}^s_u(\alpha)$, so we have $\mathtt{to}^s_u(\alpha a) = \mathtt{to}^s_u(\beta)$ as required.

Alternately, suppose $\mathtt{dom}(a) \rightarrowtail u$. Then we cannot have $\beta = \epsilon$. Let b be the final action of β and write $\beta = \alpha' b$. If $\mathtt{dom}(b) \not\rightarrowtail u$ then we can swap the role of αa and β and apply the previous case. Assume, therefore, that also $\mathtt{dom}(b) \rightarrowtail u$. Then by definition of $\mathtt{to}^a$, we obtain from $\mathtt{to}^a_u(\alpha a) = \mathtt{to}^a_u(\alpha' b)$ that $\mathtt{to}^a_u(\alpha) = \mathtt{to}^a_u(\alpha')$, and $a = b$. Moreover, we have either $\mathtt{view}^a_{\mathtt{dom}(a)}(\alpha) = \mathtt{view}^a_{\mathtt{dom}(a)}(\alpha')$, if $\mathtt{dom}(a) \neq u$, or $\mathtt{view}^a_u(\alpha a) = \mathtt{view}^a_u(\alpha' a)$, if $\mathtt{dom}(a) = u$. In either case, $\mathtt{view}^a_{\mathtt{dom}(a)}(\alpha) = \mathtt{view}^a_{\mathtt{dom}(a)}(\alpha')$, so we get from part (1) that $\mathtt{view}^s_{\mathtt{dom}(a)}(\alpha) = \mathtt{view}^s_{\mathtt{dom}(a)}(\alpha')$. By the induction hypothesis, we have from $\mathtt{to}^a_u(\alpha) = \mathtt{to}^a_u(\alpha')$ that $\mathtt{to}^s_u(\alpha) = \mathtt{to}^s_u(\alpha')$. Thus,

$$\begin{aligned}\mathtt{to}^s_u(\alpha a) &= (\mathtt{to}^s_u(\alpha), \mathtt{view}^s_{\mathtt{dom}(a)}(\alpha), a)\\ &= (\mathtt{to}^s_u(\alpha'), \mathtt{view}^s_{\mathtt{dom}(a)}(\alpha'), a)\\ &= \mathtt{to}^s_u(\alpha' a),\end{aligned}$$

as required.

The argument for part (3) follows the same pattern as that for part (2). We consider just the case of the inductive step for sequences αa and β. where $\mathtt{dom}(a) \rightarrowtail u$. Here, we may again assume that $\beta = \alpha' a$. If we have $\mathtt{ito}^a_u(\alpha a) = \mathtt{ito}^a_u(\alpha' a)$, then it follows that $\mathtt{ito}^a_u(\alpha) = \mathtt{ito}^a_u(\alpha')$, and $\mathtt{view}^a_{\mathtt{dom}(a)}(\alpha a) = \mathtt{view}^a_{\mathtt{dom}(a)}(\alpha' a)$. By the induction hypothesis, it follows that $\mathtt{ito}^s_u(\alpha) = \mathtt{ito}^s_u(\alpha')$. By part (1), we may conclude that $\mathtt{view}^s_{\mathtt{dom}(a)}(\alpha a) = \mathtt{view}^s_{\mathtt{dom}(a)}(\alpha' a)$, which also implies $\mathtt{view}^s_{\mathtt{dom}(a)}(\alpha) = \mathtt{view}^s_{\mathtt{dom}(a)}(\alpha')$. Thus, in either case of $\mathtt{dom}(a) = u$ or $\mathtt{dom}(a) \neq u$, we have that $\mathtt{ito}^s_u(\alpha a) = \mathtt{ito}^s_u(\alpha' a)$. □

We note that the converses to parts (2) and (3) of Lemma 3 do not hold in general. Consider the system in Figure 3, where $\mathtt{dom}(l) = L$ and $\mathtt{dom}(h) = H$

and we assume $H \not\rightarrowtail L$. Here we have

$$\begin{aligned}\mathtt{to}^s_L(hl) &= (\mathtt{to}^s_L(h), \mathtt{view}^s_L(h), l)\\ &= (\mathtt{to}^s_L(\epsilon), \bot, l)\\ &= (\mathtt{to}^s_L(\epsilon), \mathtt{view}^s_L(\epsilon), l)\\ &= \mathtt{to}^s_L(l)\ .\end{aligned}$$

But

$$\begin{aligned}\mathtt{to}^a_L(hl) &= (\mathtt{to}^a_L(h), \mathtt{view}^a_L(hl), l)\\ &= (\mathtt{to}^a_L(\epsilon), l1, l)\\ &= (\epsilon, l1, l)\end{aligned}$$

and

$$\begin{aligned}\mathtt{to}^a_L(l) &= (\mathtt{to}^a_L(\epsilon), \mathtt{view}^a_L(l), l)\\ &= (\epsilon, l0, l)\end{aligned}$$

so $\mathtt{to}^a_L(hl) \neq \mathtt{to}^a_L(l)$. Similarly, $\mathtt{ito}^s_L(hl) = \mathtt{ito}^s_L(l)$, but $\mathtt{ito}^a_L(hl) = (\epsilon, l1, l) \neq (\epsilon, l0, l) = \mathtt{ito}^a_L(l)$.

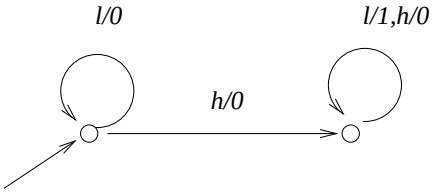

Fig. 3. A system showing inequivalence of $\mathtt{to}^a$ ($\mathtt{ito}^a$) and $\mathtt{to}^s$ ($\mathtt{ito}^s$).

Lemma 4. *The action-observed system M is TO-secure with respect to $\rightarrowtail$ iff $F_{as}(M)$ is TO-secure with respect to $\rightarrowtail$.*

Proof. Throughout, we interpret $\mathtt{to}^a$, $\mathtt{view}^a$ with respect to M and $\rightarrowtail$ and $\mathtt{to}^s$, $\mathtt{view}^s$ with respect to $F_{as}(M)$ and $\rightarrowtail$.

Suppose first that M is TO-secure with respect to $\rightarrowtail$. We claim the following: for all $\alpha, \beta \in A^*$, if $\mathtt{to}^s_u(\alpha) = \mathtt{to}^s_u(\beta)$ then (1) $\mathtt{obs}_u(s'_0 \cdot \alpha) = \mathtt{obs}_u(s'_0 \cdot \beta)$, and (2) $\mathtt{to}^a_u(\alpha) = \mathtt{to}^a_u(\beta)$. The TO-security of $F_{as}(M)$ with respect to $\rightarrowtail$ then follows immediately using part (1).

The proof of the claim is by induction on the combined length of α and β. The base case of $\alpha = \beta = \epsilon$ is trivial. Consider sequences $\alpha a, \beta$ with $\mathtt{to}^s_u(\alpha a) = \mathtt{to}^s_u(\beta)$, where $a \in A$ and the claim holds for sequences of shorter combined length. We consider two cases, depending on whether $\mathtt{dom}(a) \rightarrowtail u$.

1. Suppose $\mathtt{dom}(a) \not\rightarrowtail u$. Then $\mathtt{to}^s_u(\alpha a) = \mathtt{to}^s_u(\alpha)$. Thus, since $\mathtt{to}^s_u(\alpha a) = \mathtt{to}^s_u(\beta)$ we have $\mathtt{to}^s_u(\alpha) = \mathtt{to}^s_u(\beta)$. By the induction hypothesis, we have (1) $\mathtt{obs}_u(s'_0 \cdot \alpha) = \mathtt{obs}_u(s'_0 \cdot \beta)$, and (2) $\mathtt{to}^a_u(\alpha) = \mathtt{to}^a_u(\beta)$. Now $\mathtt{dom}(a) \not\rightarrowtail u$ implies $\mathtt{dom}(a) \neq u$, so by construction of $F_{as}(M)$ we have $\mathtt{obs}_u(s'_0 \cdot \alpha a) =$

$\mathtt{obs}_u(s'_0{\cdot}\alpha)$, and it follows that $\mathtt{obs}_u(s'_0{\cdot}\alpha a) = \mathtt{obs}_u(s'_0{\cdot}\beta)$. Finally, $\mathtt{to}^a_u(\alpha a) = \mathtt{to}^a_u(\alpha)$, so also $\mathtt{to}^a_u(\alpha a) = \mathtt{to}^a_u(\beta)$. This establishes (1) and (2) for the case of sequences αa and β.

2. Suppose $\mathtt{dom}(a) \rightarrowtail u$. Then $\mathtt{to}^s_u(\alpha a) = (\mathtt{to}^s_u(\alpha), \mathtt{view}^s_{\mathtt{dom}(a)}(\alpha), a)$. Thus, if $\mathtt{to}^s_u(\alpha a) = \mathtt{to}^s_u(\beta)$, then β is not ϵ. Let b be the final action in β. If $\mathtt{dom}(b) \not\rightarrowtail u$ then we may apply the previous case with the roles of αa and β swapped. We may therefore assume that $\mathtt{dom}(b) \rightarrowtail u$, and write $\beta = \alpha' b$, so that $\mathtt{to}^s_u(\beta) = (\mathtt{to}^s_u(\alpha'), \mathtt{view}^s_{\mathtt{dom}(b)}(\alpha'), b)$. It then follows from $\mathtt{to}^s_u(\alpha a) = \mathtt{to}^s_u(\beta)$ that $\mathtt{to}^s_u(\alpha) = \mathtt{to}^s_u(\alpha')$, $a = b$ and $\mathtt{view}^s_{\mathtt{dom}(a)}(\alpha) = \mathtt{view}^s_{\mathtt{dom}(a)}(\alpha')$. From the first of these, we obtain by the induction hypothesis that (1) $\mathtt{obs}_u(s'_0 \cdot \alpha) = \mathtt{obs}_u(s'_0 \cdot \alpha')$, and (2) $\mathtt{to}^a_u(\alpha) = \mathtt{to}^a_u(\alpha')$. From the last, we obtain by Lemma 3 that $\mathtt{view}^a_{\mathtt{dom}(a)}(\alpha) = \mathtt{view}^a_{\mathtt{dom}(a)}(\alpha')$. We now consider two subcases, depending on whether $\mathtt{dom}(a) = u$.
   (a) If $\mathtt{dom}(a) = u$, then we obtain from (2) and the TO-security of M that $\mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{out}(s_0 \cdot \alpha', a)$. It follows that (1′) $\mathtt{obs}_u(s'_0 \cdot \alpha a) = \mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{out}(s_0 \cdot \alpha', a) = \mathtt{obs}_u(s'_0 \cdot \alpha' a)$. Also, using (1′) and the fact that $\mathtt{view}^a_{\mathtt{dom}(a)}(\alpha) = \mathtt{view}^a_{\mathtt{dom}(a)}(\alpha')$, we get that $\mathtt{view}^a_{\mathtt{dom}(a)}(\alpha a) = \mathtt{view}^a_{\mathtt{dom}(a)}(\alpha)\, a\, \mathtt{out}(s_0{\cdot}\alpha, a) = \mathtt{view}^a_{\mathtt{dom}(a)}(\alpha)\, a\, \mathtt{out}(s_0{\cdot}\alpha', a) = \mathtt{view}^a_{\mathtt{dom}(a)}(\alpha' a)$. Using this, we get (2′)

$$\begin{aligned}\mathtt{to}^a_u(\alpha a) &= (\mathtt{to}^a_u(\alpha), \mathtt{view}^a_{\mathtt{dom}(a)}(\alpha a), a)\\ &= (\mathtt{to}^a_u(\alpha'), \mathtt{view}^a_{\mathtt{dom}(a)}(\alpha' a), a)\\ &= \mathtt{to}^a_u(\alpha').\end{aligned}$$

   (b) If $\mathtt{dom}(a) \neq u$, then by (1) and construction of $F_{as}(M)$, we get (1′) $\mathtt{obs}_u(s'_0 \cdot \alpha a) = \mathtt{obs}_u(s'_0 \cdot \alpha) = \mathtt{obs}_u(s'_0 \cdot \alpha') = \mathtt{obs}_u(s'_0 \cdot \alpha' a)$. Also we have (2′)

$$\begin{aligned}\mathtt{to}^a_u(\alpha a) &= (\mathtt{to}^a_u(\alpha), \mathtt{view}^a_{\mathtt{dom}(a)}(\alpha), a)\\ &= (\mathtt{to}^a_u(\alpha'), \mathtt{view}^a_{\mathtt{dom}(a)}(\alpha'), a)\\ &= \mathtt{to}^a_u(\alpha' a).\end{aligned}$$

   In either case, the statements (1′) and (2′) give what we require for the case of sequences αa and $\alpha' a = \beta$.

Conversely, suppose that $F_{as}(M)$ is TO-secure with respect to $\rightarrowtail$. We show M is TO-secure with respect to $\rightarrowtail$. Let $\alpha, \beta \in A^*$ be sequences and $a \in A$ such that $\mathtt{dom}(a) = u$ and $\mathtt{to}^a_u(\alpha) = \mathtt{to}^a_u(\beta)$. We have to show $\mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{out}(s_0 \cdot \beta, a)$. By Lemma 3, we have $\mathtt{to}^s_u(\alpha) = \mathtt{to}^s_u(\beta)$. Since $F_{as}(M)$ is TO-secure with respect to $\rightarrowtail$, this implies $\mathtt{view}^s_u(\alpha) = \mathtt{view}^s_u(\beta)$ — see [vdM07] for proof. Thus (using the fact that $\mathtt{dom}(a) = u$, we have $\mathtt{to}^s_u(\alpha a) = (\mathtt{to}^s_u(\alpha), \mathtt{view}^s_u(\alpha), a) = (\mathtt{to}^s_u(\beta), \mathtt{view}^s_u(\beta), a) = \mathtt{to}^s_u(\beta a)$. By TO-security of $F_{as}(M)$, construction of this system, and the fact that $\mathtt{dom}(a) = u$, we have $\mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{obs}_u(s'_0 \cdot \alpha a) = \mathtt{obs}_u(s'_0 \cdot \beta a) = \mathtt{out}(s_0 \cdot \beta, a)$, as required. □

We have a similar result for ITO-security.

Lemma 5. *The action-observed system M is ITO-secure with respect to $\rightarrowtail$ iff $F_{as}(M)$ is ITO-secure with respect to $\rightarrowtail$.*

Proof. Throughout, we interpret action-observed notions with respect to M and state-observed notions with respect to $F_{as}(M)$.

Suppose first that $F_{as}(M)$ is ITO-secure with respect to $\rightarrowtail$. To show that M is ITO-secure with respect to $\rightarrowtail$, consider an agent u, sequences $\alpha, \beta \in A^*$ such that $\mathtt{ito}^a_u(\alpha) = \mathtt{ito}^a_u(\beta)$, and an action a with $\mathtt{dom}(a) = u$. We show that $\mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{out}(s_0 \cdot \beta, a)$. Now, by Lemma 3 part (3), we have $\mathtt{ito}^s_u(\alpha) = \mathtt{ito}^s_u(\beta)$. By ITO-security of $F_{as}(M)$, this implies that $\mathtt{view}^s_u(\alpha) = \mathtt{view}^s_u(\beta)$. (The proof is a simple induction.) Since $\mathtt{dom}(a) = u$, we obtain that $\mathtt{ito}^s_u(\alpha a) = (\mathtt{ito}^s_u(\alpha), \mathtt{view}^s_u(\alpha), a) = (\mathtt{ito}^s_u(\beta), \mathtt{view}^s_u(\beta), a) = \mathtt{ito}^s_u(\beta a)$. Using ITO-security and definition of $F_{as}(M)$, it follows that $\mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{obs}_u(s'_0 \cdot \alpha a) = \mathtt{obs}_u(s'_0 \cdot \beta a) = \mathtt{out}(s_0 \cdot \beta, a)$, as required.

Conversely, suppose that M is ITO-secure with respect to $\rightarrowtail$. We claim that for all $\alpha, \beta \in A^*$ and agents u, if $\mathtt{ito}^s_u(\alpha) = \mathtt{ito}^s_u(\beta)$ then $\mathtt{ito}^a_u(\alpha) = \mathtt{ito}^a_u(\beta)$. The ITO-security of $F_{as}(M)$, i.e., the fact that $\mathtt{ito}^s_u(\alpha) = \mathtt{ito}^s_u(\beta)$ implies $\mathtt{obs}_u(s'_0 \cdot \alpha) = \mathtt{obs}_u(s'_0 \cdot \beta)$ then follows from the fact that $\mathtt{ito}^a_u(\alpha) = \mathtt{ito}^a_u(\beta)$ implies $\mathtt{obs}_u(s'_0 \cdot \alpha) = \mathtt{obs}_u(s'_0 \cdot \beta)$. For, since M is ITO-secure, we have by a simple induction that $\mathtt{ito}^a_u(\alpha) = \mathtt{ito}^a_u(\beta)$ implies that $\mathtt{view}^a_u(\alpha) = \mathtt{view}^a(\beta)$. In particular, the last output from an action in domain u (or $\perp$ if there is no such action) obtained in the sequence α is identical to the last such output (or $\perp$) obtained in the sequence β. But these values are equal to $\mathtt{obs}_u(s'_0 \cdot \alpha)$ and $\mathtt{obs}_u(s'_0 \cdot \beta)$, respectively, so must be identical.

To prove the claim, we proceed by induction on the combined length of α and β. The base case is trivial. Consider sequences αa and β where the claim holds for shorter sequences, and suppose $\mathtt{ito}^s_u(\alpha a) = \mathtt{ito}^s_u(\beta)$.

First, if $\mathtt{dom}(a) \not\rightarrowtail u$, then $\mathtt{ito}^s_u(\alpha) = \mathtt{ito}^s_u(\alpha a) = \mathtt{ito}^s_u(\beta)$, so $\mathtt{ito}^a_u(\alpha) = \mathtt{ito}^a_u(\beta)$, by the inductive hypothesis. Since $\mathtt{ito}^a_u(\alpha a) = \mathtt{ito}^a_u(\alpha)$, this implies $\mathtt{ito}^a_u(\alpha a) = \mathtt{ito}^a_u(\beta)$.

In case $\mathtt{dom}(a) \rightarrowtail u$, we obtain that β is not ϵ, so may write $\beta = \alpha' b$ for some $b \in A$. If $\mathtt{dom}(b) \not\rightarrowtail u$ we may apply the previous case, so we may assume that $\mathtt{dom}(b) \rightarrowtail u$. It now follows from $\mathtt{ito}^s_u(\alpha a) = \mathtt{ito}^s_u(\alpha' a)$ that $\mathtt{ito}^s_u(\alpha) = \mathtt{ito}^s_u(\alpha')$ and either $\mathtt{view}^s_{\mathtt{dom}(a)}(\alpha) = \mathtt{view}^s_{\mathtt{dom}(a)}(\alpha')$ (if $\mathtt{dom}(a) = u$), or $\mathtt{view}^s_{\mathtt{dom}(a)}(\alpha a) = \mathtt{view}^s_{\mathtt{dom}(a)}(\alpha' a)$ (otherwise). Further, we have from $\mathtt{ito}^s_u(\alpha a) = \mathtt{ito}^s_u(\alpha' a)$ by ITO-security of $F_{as}(M)$ that $\mathtt{obs}_u(s'_0 \cdot \alpha a) = \mathtt{obs}_u(s'_0 \cdot \alpha' a)$. Thus, in fact we also have $\mathtt{view}^s_{\mathtt{dom}(a)}(\alpha a) = \mathtt{view}^s_{\mathtt{dom}(a)}(\alpha' a)$ in the case where $\mathtt{dom}(a) = u$. By Lemma 3, we obtain that $\mathtt{view}^a_{\mathtt{dom}(a)}(\alpha a) = \mathtt{view}^a_{\mathtt{dom}(a)}(\alpha' a)$. Since $\mathtt{ito}^s_u(\alpha) = \mathtt{ito}^s_u(\alpha')$, we have by the induction hypothesis that $\mathtt{ito}^a_u(\alpha) = \mathtt{ito}^a_u(\alpha')$. It follows that $\mathtt{ito}^a_u(\alpha a) = \mathtt{ito}^a_u(\alpha' a)$, as required. □

This completes the proof of Theorem 3.

5 Unwinding and Access Control

Rushby [Rus92] proves a number for results concerning IP-security: soundness for a proof method called *unwinding*, and that a particular class of systems – access control systems consistent with a policy – are IP-secure. These results

are improved in [vdM07] where it is shown that unwinding is in fact sound and complete for the stronger notion of TA-security, and that with an intuitive change to the definition of access control system, there is also a close correspondence between TA-security and a system's interpretability as an access control system.

The improved results are established in [vdM07] only for the state-observed model, whereas Rusbhy discusses versions of his results for both state and action-observed systems. He proves the state-observed version by reference to a modification of the (partially automated in PVS) proofs for the action-observed version. In this section, we show that the translation of the previous section provides a way to transfer the improved results from state to action-observed systems.

5.1 State-Observed Unwinding

We begin by recalling the relevant definitions and results for state-observed systems.

Suppose we have for each domain u an equivalence relation $\sim_u$ on the states of M. If M is a state-observed system, we say the family $\{\sim_u\}_{u \in D}$ is an *unwinding* with respect to $\rightarrowtail$ if it satisfies the following conditions.

OC: If $s \sim_u t$ then $\mathtt{obs}_u(s) = \mathtt{obs}_u(t)$.	(Output Consistency)
SC: If $s \sim_u t$ then $s \cdot a \sim_u t \cdot a$.	(Step Consistency)
LR: If $\mathtt{dom}(a) \not\rightarrowtail u$ then $s \sim_u s \cdot a$.	(Left Respect)

Proposition 2. *([GM84] and [Rus92]) A system M is P-secure with respect to the transitive policy $\rightarrowtail$ iff there exists an unwinding on M with respect to $\rightarrowtail$.*

For intransitive noninterference Rushby introduces the following condition:

WSC: If $s \sim_u t$ and $s \sim_{dom(a)} t$ then $s \cdot a \sim_u t \cdot a$. (Weak Step Consistency)

Define a *weak unwinding* on a system M with respect to a policy $\rightarrowtail$ to be a family of relations $\sim_u$, for $u \in D$, satisfying OC,WSC and LR.

Given a system $M = \langle S, s_0, \mathtt{step}, \mathtt{obs}, \mathtt{dom}\rangle$ with actions A, define the "unfolded" system $\mathtt{uf}_s(M) = \langle S', s_0', \mathtt{step}', \mathtt{obs}', \mathtt{dom}\rangle$ with actions A having the same domains as in M, by $S' = A^*$, $s_0' = \epsilon$, $\mathtt{step}'(\alpha, a) = \alpha a$, and $\mathtt{obs}_u'(\alpha) = \mathtt{obs}_u(s_0 \cdot \alpha)$, where $s_0 \cdot \alpha$ is computed in M. Intuitively, this construction unfolds the graph of M into an infinite tree. Then we have the following.

Theorem 4. *([vdM07]) M is TA-secure with respect to $\rightarrowtail$ iff there exists a weak unwinding on $\mathtt{uf}_s(M)$ with respect to $\rightarrowtail$.*

We remark that bisimulation does not preserve existence of a weak unwinding (see [vdM07] for an example), so it turns out that the reference to $\mathtt{uf}_s(M)$ cannot be replaced by M in this result.

5.2 Action-observed Unwinding

We now turn to a consideration of action-observed versions of unwinding and access control, showing that the results of the previous subsections can be carried across to the state-observed domain by means of the transformation of Section 4.

Let M be an action-observed system. Suppose we have for each domain u an equivalence relation $\sim_u$ on the states of M. We say the family $\{\sim_u\}_{u \in D}$ is an *action-observed weak unwinding* with respect to $\rightarrowtail$ if it satisfies WSC, LR and

OC_a: if $s \sim_u t$ and $\mathtt{dom}(a) = u$ then $\mathtt{out}(s, a) = \mathtt{out}(t, a)$.

The definition of unfolding for action-observed systems is the expected: if $M = \langle S, s_0, A, \mathtt{step}, \mathtt{out}, \mathtt{dom}\rangle$, then the unfolding $\mathtt{uf}_a(M) = \langle S', s'_0, A, \mathtt{step}', \mathtt{out}', \mathtt{dom}\rangle$ where

1. $S' = A^*$,
2. $s'_0 = \epsilon$,
3. $\mathtt{step}'(\alpha, a) = \alpha a$,
4. $\mathtt{out}'(\alpha, a) = \mathtt{out}(s_0 \cdot \alpha, a)$.

We then have the following characterization of TA-security using unwinding, mirroring Theorem 4.

Theorem 5. *Let M be an action-observed system. Then M is TA-secure iff there exists an action-observed unwinding $\sim$ on $\mathtt{uf}_a(M)$.*

The proof uses the following lemmas. For the following, define the *reachable* fragment of a system $M = \langle S, s_0, \mathtt{step}, \mathtt{out}, \mathtt{dom}\rangle$ to be the system $\mathtt{reach}(M)$ obtained from M by restricting the set of states to the reachable states $S' = \{s_0 \cdot \alpha \mid \alpha \in A^*\}$ (and restricting each of the other components accordingly). Define two systems $M = \langle S, s_0, \mathtt{step}, \mathtt{out}, \mathtt{dom}\rangle$ and $M' = \langle S', s'_0, \mathtt{step}', \mathtt{out}', \mathtt{dom}'\rangle$ to be *isomorphic* if they have the same set of actions A, $\mathtt{dom} = \mathtt{dom}'$, and there exists a bijection $\iota : S \to S'$ such that $\iota(s_0) = s'_0$, and for all states $s \in S$ and actions a, we have $\iota(\mathtt{step}(s, a)) = \mathtt{step}'(\iota(s), a)$ and $\mathtt{out}(s, a) = \mathtt{out}'(\iota(s), a)$.

Lemma 6. *If M is an action-observed system, then $\mathtt{uf}_s(F_{as}(M))$ is isomorphic to $\mathtt{reach}(F_{as}(\mathtt{uf}_a(M)))$.*

Proof. For each $\alpha \in A^*$, let $f_\alpha : D \to O \cup \{\bot\}$ be the function such that $(s_0, f_0) \cdot \alpha = (s_0 \cdot \alpha, f_\alpha)$ in $F_{as}(M)$. Note that states of $\mathtt{uf}_s(F_{as}(M))$ have the form $\alpha \in A^*$, and observations are determined by $\mathtt{obs}_u^{\mathtt{uf}_s(F_{as}(M))}(\alpha) = \mathtt{obs}_u^{F_{as}(M)}((s_0, f_0) \cdot \alpha) = f_\alpha(u)$.

Since the states of $\mathtt{uf}_a(M)$ have the form $\alpha \in A^*$, the states of $F_{as}(\mathtt{uf}_a(M))$ have the form (α, f) where $f : D \to O \cup \{\bot\}$. In fact, we claim that for each α there is a unique f such that (α, f) is reachable in $F_{as}(\mathtt{uf}_a(M))$, viz., $f = f_\alpha$. The proof of this is by induction on the length of α. The case of $\alpha = \epsilon$ is trivial. Suppose now that $(\alpha a, f)$ is reachable in $F_{as}(\mathtt{uf}_a(M))$, where $\alpha \in A^*$ and $a \in A$. By construction and the induction hypothesis, it can only be reached

by a path in which the final step is $(\alpha, f_\alpha) \cdot a = (\alpha a, f)$. Hence $f = f_\alpha[\mathtt{dom}(a) \mapsto \mathtt{out}^{\mathtt{uf}_a(M)}(\alpha, a)] = f_\alpha[\mathtt{dom}(a) \mapsto \mathtt{out}^M(s_0 \cdot \alpha, a)] = f_\alpha$.

It is now straightforward to check that the mapping ι mapping A^* to the states of $F_{as}(\mathtt{uf}_a(M))$, defined by $\iota(\alpha) = (\alpha, f_\alpha)$, is an isomorphism from $\mathtt{uf}_s(F_{as}(M))$ to $\mathtt{reach}(F_{as}(\mathtt{uf}_a(M)))$. It is plainly a bijection, and we have $\iota(\epsilon) = (\epsilon, f_\epsilon) = (s_0, f_0)$. For the transitions, $\iota(\alpha \cdot a) = \iota(\alpha a) = (\alpha a, f_{\alpha a}) = (\alpha, f_\alpha) \cdot a = \iota(\alpha) \cdot a$. For the observations, we have $\mathtt{obs}_u^{F_{as}(\mathtt{uf}_a(M))}(\iota(\alpha)) = \mathtt{obs}_u^{F_{as}(\mathtt{uf}_a(M))}((\alpha, f_\alpha)) = f_\alpha(u) = \mathtt{obs}_u^{\mathtt{uf}_s(F_{as}(M))}(\alpha)$. □

Say that an action-observed system $M = \langle S, s_0, \mathtt{next}, \mathtt{out}\rangle$ is *output-recording* if there exists a family of functions $\kappa_u : S \to O \cup \{\bot\}$ for $u \in D$, such that

1. $\kappa_u(s_0) = \bot$,
2. if $\mathtt{dom}(a) = u$ then $\kappa_u(s \cdot a) = \mathtt{out}(s, a)$,
3. if $\mathtt{dom}(a) \neq u$ then $\kappa_u(s \cdot a) = \kappa(s)$.

Intuitively, this says that the state records the most recent output obtained in each domain. Note that the system $\mathtt{uf}_a(M)$ is action-recording for each action-observed system M.

Lemma 7. *Let M be an action-observed system.*

1. *If there exists an action-observed weak unwinding on M then there exists a state-observed weak unwinding on $F_{as}(M)$ (hence on $\mathtt{reach}(M)$).*
2. *If M is output-recording and there exists a state-observed weak unwinding on $F_{as}(M)$ ($\mathtt{reach}(F_{as}(M))$) then there there exists an action-observed weak unwinding on M ($\mathtt{reach}(M)$).*

Proof. Suppose that $\{\sim_u^a\}_{u \in D}$ is an action-observed unwinding on M. Define the relations $\sim_u^s$ on the states of $F_{as}(M)$ by $(s, f) \sim_u^s (t, g)$ if $s \sim_u^a t$ and $f(u) = g(u)$. We show that this gives a weak unwinding on $F_{as}(M)$. It is plain that $\sim_u^s$ is an equivalence relation.

1. **OC$_s$:** Suppose that $(s, f) \sim_u^s (t, g)$. Then $\mathtt{obs}_u((s, f)) = f(u) = g(u) = \mathtt{obs}_u((t, g))$ by definition.
2. **WSC:** Suppose $(s, f) \sim_u^s (t, g)$ and $(s, f) \sim_{\mathtt{dom}(a)}^s (t, g)$. From the former we obtain $s \sim_u^a t$ and $f(u) = g(u)$. From the latter, we obtain $s \sim_{\mathtt{dom}(a)}^a t$. Since $\sim^a$ is a weak unwinding, we get from this that $\mathtt{out}(s, a) = \mathtt{out}(t, a)$. Also, we have by WSC that $s \cdot a \sim_u^a t \cdot a$. By the above, $f[\mathtt{dom}(a) \to \mathtt{out}(s, a)](u) = g[\mathtt{dom}(a) \to \mathtt{out}(s, a)](u)$. This shows that $(s, f) \cdot a \sim_u^s (t, g) \cdot a$.
3. **LR:** Suppose $\mathtt{dom}(a) \not\leadsto u$. Then because $\sim_u^a$ is an unwinding, we have $s \sim_u^a s \cdot a$. Since we must have $\mathtt{dom}(a) \neq u$, we have $f[\mathtt{dom}(a) \mapsto \mathtt{out}(s, a)](u) = f(u)$. Hence $(s, f) \sim_u^a (s \cdot a, f[\mathtt{dom}(a) \mapsto \mathtt{out}(s, a)]) = (s, f) \cdot a$.

For part (2), suppose that M is output recording, witnessed by the functions κ_u. Let $\sim^s$ be a state-observed weak unwinding on $F_{as}(M)$ (resp. $\mathtt{reach}(F_{as}(M))$). Define the relations $\sim_u^a$ on the states of M (resp. $\mathtt{reach}(M)$), for $u \in D$, by $s \sim_u^a t$ if $(s, f_s) \sim_u^s (t, f_t)$, where for a state s the function $f_s : D \to O \cup \{\bot\}$

is defined by $f_s(u) = \kappa_u(s)$. Note that it follows from the properties of κ that if s is reachable then so is (s, f_s), so this is well-defined in case $\sim^s$ is an unwinding just on $\mathtt{reach}(F_{as}(M))$. It also follows from the properties of κ that $(s, f_s) \cdot a = (s \cdot a, f_{s \cdot a})$.

We show that $\sim^a$ is an action-observed unwinding on M.

1. **OC_a:** Suppose that $s \sim^a_u t$ and let $a \in A$ with $\mathtt{dom}(a) = u$. By definition of $\sim^a$, we have $(s, f_s) \sim^s_u (t, f_t)$. Since $\sim^s$ is a weak unwinding, we have by WSC that $(s, f_s) \cdot a \sim^s_u (t, f_t) \cdot a$, i.e.

$$(s \cdot a, f_s[\mathtt{dom}(a) \mapsto \mathtt{out}(s, a)]) \sim^s_u (t \cdot a, f_t[\mathtt{dom}(a) \mapsto \mathtt{out}(s, a)]),$$

   hence, by OC_s, we obtain that

$$f_s[\mathtt{dom}(a) \mapsto \mathtt{out}(s, a)](u) = f_t[\mathtt{dom}(a) \mapsto \mathtt{out}(s, a)](u),$$

   i.e., $\mathtt{out}(s, a) = \mathtt{out}(t, a)$.
2. **WSC:** Suppose that $s \sim^a_u t$ and $s \sim^a_{\mathtt{dom}(a)} t$. By definition, $(s, f_s) \sim^s_u (t, f_t)$ and $(s, f_s) \sim^s_{\mathtt{dom}(a)} (t, f_t)$. Thus, by WSC for $\sim^s$, we have $(s, f_s) \cdot a \sim^s_u (t, f_t) \cdot a$, i.e., $(s \cdot a, f_{s \cdot a}) \sim^s_u (t \cdot a, f_{t \cdot a})$. This means that $s \cdot a \sim^a_u t \cdot a$, as required for WSC for $\sim^a$.
3. **LR:** If $\mathtt{dom}(a) \not\leadsto u$, then by LR for $\sim^s$, we have $(s, f_s) \sim^s_u (s, f_s) \cdot a = (s \cdot a, f_{s \cdot a})$. By definition, we obtain $s \sim^a_u s \cdot a$, as required for LR for $\sim^a$.

□

To prove Theorem 5, we may now note that the following are equivalent:

1. The action-observed system M is TA-secure;
2. The state-observed system $F_{as}(M)$ is TA-secure (by Theorem 3);
3. there exists a (state-observed) unwinding on $\mathtt{uf}_s(F_{as}(M))$ (by Theorem 6);
4. there exists a (state-observed) unwinding on $\mathtt{reach}(F_{as}(\mathtt{uf}_a(M)))$ (by Lemma 6);
5. there exists an (action-observed) unwinding on $\mathtt{uf}_a(M)$ (by Lemma 7, the fact that $\mathtt{uf}_a(M)$ is action-recording, and the fact that $\mathtt{reach}(\mathtt{uf}_a(M)) = \mathtt{uf}_a(M)$).

5.3 State-Observed Weak Access Control Systems

Weak access control systems are a class of systems, introduced in [vdM07], modifying a definition from [Rus92], that gives semantic content to the Bell-La Padula model. In this subsection, we recall a result from [vdM07], relating state-observed weak access control systems to TA-security. The following subsection develops an action-observed version of this result.

A *system with structured state* is a machine $\langle S, s_0, A, \mathtt{step}, \mathtt{obs}, \mathtt{dom} \rangle$ together with

1. a set N of *names*,
2. a set V of *values*, and functions

3. $\texttt{contents} : S \times N \to V$, with $\texttt{contents}(s, n)$ interpreted as the value of object n in state s,
4. $\texttt{observe} : D \to \mathcal{P}(N)$, with $\texttt{observe}(u)$ interpreted as the set of objects that domain u can observe, and
5. $\texttt{alter} : D \to \mathcal{P}(N)$, with $\texttt{alter}(u)$ interpreted as the set of objects whose values domain u is permitted to alter.

For a system with structured state, when $u \in D$ and s is a state, write $\texttt{oc}_u(s)$ for the function mapping $\texttt{observe}(u)$ to values, defined by $\texttt{oc}_u(s)(n) = \texttt{contents}(s, n)$ for $n \in \texttt{observe}(u)$. Intuitively, $\texttt{oc}_u(s)$ captures all the content of the state s that is observable to u. Using this, we may define a binary relation $\sim_u^{\texttt{oc}}$ of *observable content equivalence* on S for each domain $u \in D$, by $s \sim_u^{\texttt{oc}} t$ if $\texttt{oc}_u(s) = \texttt{oc}_u(t)$.

The following conditions express that the machine operates in accordance with the intuitive interpretations of this extra structure:

RM1. If $s \sim_u^{\texttt{oc}} t$ then $\texttt{obs}_u(s) = \texttt{obs}_u(t)$.
RM2′. For all actions a, states s, t and names $n \in \texttt{alter}(dom(a))$, if $s \sim_{\texttt{dom}(a)}^{\texttt{oc}} t$ and $\texttt{contents}(s, n) = \texttt{contents}(t, n)$ we have $\texttt{contents}(s{\cdot}a, n) = \texttt{contents}(t{\cdot}a, n)$.
RM3. If $\texttt{contents}(s \cdot a, n) \neq \texttt{contents}(s, n)$ then $n \in \texttt{alter}(dom(a))$.

The first of these says that an agent's observation depends only on the values of the objects observable to the agent. The condition RM2′ says that the the value $\texttt{contents}(s \cdot a, n)$ may depend both on $\texttt{contents}(s, n)$ and $\texttt{oc}_{\texttt{dom}(a)}(s)$. This is a weakening of a similar condition RM2 used by Rushby, which did not allow the dependence on $\texttt{contents}(s, n)$. The third says that if an action can change the value of an object, then the agent of that action is in fact permitted to alter that object. We define a system with structured states to be a *weak access control system* if it satisfies conditions RM1,RM2′, and RM3.

Finally, we consider the condition:

AOI. If $\texttt{alter}(u) \cap \texttt{observe}(v) \neq \emptyset$ then $u \rightarrowtail v$.

Intuitively, this says that the ability to write to a value that an agent can observe counts as a way to interfere with that agent.

We also introduce a related notion on systems without structured states, that expresses that the system behaves as if it were an access control system. Say that a system M with states S *admits a weak access control interpretation consistent with* $\rightarrowtail$ if there exists a set of names N, a set of values V and functions $\texttt{observe} : D \times S \to \mathcal{P}(N)$, $\texttt{alter} : D \times S \to \mathcal{P}(N)$ and $\texttt{contents} : N \times S \to V$, with respect to which M is a weak access control system satisfying the condition AOI.

The following result extends Theorem 4, and shows that there is a close correspondence between TA-security, weak access control interpretations, and weak unwindings.

Theorem 6. *The following are equivalent*

1. *M is TA-secure with respect to $\rightarrowtail$,*
2. *$\mathtt{uf}_s(M)$ admits a weak access control interpretation consistent with $\rightarrowtail$,*
3. *there exists a weak unwinding on $\mathtt{uf}_s(M)$ with respect to $\rightarrowtail$.*

In some special cases, access control interpretability is also sufficient for TO-security. Say that a system with structured states is *fully observable* if in all states s we have $\mathtt{obs}_u(s) = \mathtt{oc}_u(s)$. Note that this means that the relations $\sim_u^{\mathtt{oc}}$ and $\approx_u^{\mathtt{obs}}$ coincide.

Corollary 1. *If M is a fully observable weak access control system consistent with $\rightarrowtail$ then M is TO-secure with respect to $\rightarrowtail$.*

5.4 Action-Observed Weak Access Control Systems

We now develop an action-observed version of the result of the previous subsection on the equivalence between acccess control interpretability and TA-security.

Define an action-observed weak access control system, to be an action observed system with structured state (defined exactly as for state-observed systems), that satisifies conditions RM2′, RM3 and the following variant of RM1:

$\mathrm{RM1}_a$. If $s \sim_u^{\mathtt{oc}} t$ and $\mathtt{dom}(a) = u$ then $\mathtt{out}(s, a) = \mathtt{out}(t, a)$.

Compatibility of such a system with a policy $\rightarrowtail$ is defined exactly as in the state-observed case.

The following results show that it is possible to transfer an access control interpretation of a system from action- to state-observed systems and vice-versa.

Proposition 3. *Let action-observed system M have a weak access control interpretation compatible with $\rightarrowtail$. Then $F_{as}(M)$ has a weak access control interpretation compatible with $\rightarrowtail$.*

Proof. Let the system M, together with the set of names N and functions $(\mathtt{alter}, \mathtt{observe}, \mathtt{contents})$ be an weak access control system compatible with $\rightarrowtail$. We define a set of names N' and functions $(\mathtt{alter}', \mathtt{observe}', \mathtt{contents}')$ that make $F_{as}(M)$ a weak access control system. Specifically,

1. $N' = N \cup D$ (we assume that $N \cap D = \emptyset$; if not, rename N to make this the case).
2. $\mathtt{alter}'(u) = \mathtt{alter}(u) \cup \{u\}$ for all $u \in D$,
3. $\mathtt{observe}'(u) = \mathtt{observe}(u) \cup \{u\}$ for all $u \in D$,
4. for all states (s, f) of $F_{as}(M)$
   (a) $\mathtt{contents}'((s, f), n) = \mathtt{contents}(s, n)$ for $n \in N$,
   (b) $\mathtt{contents}'((s, f), u) = f(u)$ for $u \in D$.

We check that this is a weak access control interpretation compatible with $\rightarrowtail$.

For RM1, suppose that $(s, f) \sim_u^{\mathtt{oc}} (s', f')$. Since $u \in \mathtt{observe}'(u)$, we must have $\mathtt{contents}'((s, f), u)) = \mathtt{contents}'((s', f'), u))$, i.e. $f(u) = f'(u)$. By construction of $F_{as}(M)$, this means $\mathtt{obs}_u((s, f)) = \mathtt{obs}_u((s', f'))$.

For RM2′, suppose that $m \in \mathtt{alter}(u)$ and $\mathtt{contents}'((s,f),m) = \mathtt{contents}'((s',f'),m)$ and $(s,f) \sim^{\mathtt{oc}}_{\mathtt{dom}(a)} (s',f')$. Let $g = f[\mathtt{dom}(a) \mapsto \mathtt{out}(s,a)]$ and $g' = f'[\mathtt{dom}(a) \mapsto \mathtt{out}(s',a)]$, so that $(s,f)\cdot a = (s\cdot a, g)$ and $(s',f')\cdot a = (s'\cdot a, g')$. From $(s,f) \sim^{\mathtt{oc}}_{\mathtt{dom}(a)} (s',f')$ we deduce that $s \sim^{\mathtt{oc}}_{\mathtt{dom}(a)} s'$ (with respect to $\mathtt{observe}$). The conclusion from $\mathtt{contents}'((s,f),m) = \mathtt{contents}'((s',f'),m)$ depends on whether $m \in N$ or $m = u$. If $m \in N$ then we have $\mathtt{contents}(s,m) = \mathtt{contents}(s,m)$, and by the fact we have a weak access control interpretation on M, we have $\mathtt{contents}(s\cdot a, m) = \mathtt{contents}(s'\cdot a, m)$, which implies that $\mathtt{contents}'((s,f)\cdot a, m) = \mathtt{contents}'((s',f')\cdot a, m)$. If $m = u$ then we have $f(u) = f'(u)$. There are now two possibilities: either $\mathtt{dom}(a) = u$ or $\mathtt{dom}(a) \neq u$. If $\mathtt{dom}(a) = u$ then by RM1$_a$, and the fact that $s \sim^{\mathtt{oc}}_{\mathtt{dom}(a)} s'$ we have that $\mathtt{out}(s,a) = \mathtt{out}(s',a)$. Thus $\mathtt{contents}'((s,f)\cdot a, m) = g(u) = \mathtt{out}(s,a) = \mathtt{out}(s',a) = g'(u) = \mathtt{contents}'((s',f')\cdot a, m)$. If $\mathtt{dom}(a) \neq u$ then $\mathtt{contents}'((s,f)\cdot a, m) = g(u) = f(u) = f'(u) = g'(u) = \mathtt{contents}'((s',f')\cdot a, m)$. Thus, in each case we have $\mathtt{contents}'((s,f)\cdot a, m) = \mathtt{contents}'((s',f')\cdot a, m)$, as required for RM2′.

For RM3, suppose that $m \notin \mathtt{alter}(\mathtt{dom}(a))$. There are two possibilities, depending on whether $m \in N$ or $m \in D$. If $m \in N$ then we have $\mathtt{contents}(s\cdot a, m) = \mathtt{contents}(s,m)$ by the weak access control structure on M. This implies $\mathtt{contents}'((s,f)\cdot a, m) = \mathtt{contents}'((s,f),m)$. Alternately, if $m = u \in D$, then we have $\mathtt{dom}(a) \neq u$, and $\mathtt{contents}'((s,f)\cdot a, u) = f[\mathtt{dom}(a) \mapsto \mathtt{out}(s,a)](u) = f(u) = \mathtt{contents}'((s,f),u)$, also as required.

For compatibility with $\rightarrowtail$, suppose that $\mathtt{observe}'(u) \cap \mathtt{alter}'(v) \neq \emptyset$. This could be because $u = v$, so $u \in \mathtt{observe}'(u) \cap \mathtt{alter}'(v)$, but in this case $u \rightarrowtail u$ by reflexivity. On the other hand, if $u \neq v$, then we have $\mathtt{observe}'(u) \cap \mathtt{alter}'(v) = \mathtt{observe}(u) \cap \mathtt{alter}(v) \neq \emptyset$, so $u \rightarrowtail v$ by the fact that the access control struture on M is compatible with $\rightarrowtail$. □

We remark that a weak access control interpretation on a system M yields a weak access control interpretation on $\mathtt{reach}(M)$, simply by restriction of all functions to the set of reachable states.

Proposition 4. *Let M be an output-recording action-observed system, such that $F_{as}(M)$ (or $\mathtt{reach}(F_{as}(M))$) has a (state-observed) weak access control interpretation compatible with $\rightarrowtail$. Then M (respectively, $\mathtt{reach}(M)$) has a weak access control interpretation compatible with $\rightarrowtail$.*

Proof. We first note that if a state-observed system has a weak access control interpetation given by the set of names N and functions ($\mathtt{alter}, \mathtt{observe}, \mathtt{contents}$), then there exists such an interpretation with the property that $\mathtt{observe}(u) \subseteq \mathtt{alter}(u)$. This can be seen by inspecting the construction given in Proposition 11 of [vdM07], which has this property. Suppose that we have such a weak access control interpretation on $F_{as}(M)$.

Let $\kappa : S \rightarrow$ be the witness to the fact that M is output-recording. As in Lemma 7, for s a state of M define $f_s : D \rightarrow O \cup \{\perp\}$ by $f_s(u) = \kappa_u(s)$. Then the transition function of $F_{as}(M)$ satisfies $(s,f)\cdot a = (s\cdot a, f_{s\cdot a})$.

We define an access control structure on M with the same set of names N, the same functions $\mathtt{alter}$ and $\mathtt{observe}$, but function $\mathtt{contents}'$ defined by

$\mathtt{contents}'(s,n) = \mathtt{contents}((s,f_s),n)$. We show that this gives a weak access control interpetation on M compatible with $\rightarrowtail$. The compatibility is trivial from the compatibility of the access control structure on $F_{as}(M)$, since we have used the same function $\mathtt{observe}$ and $\mathtt{alter}$.

For RM2$'$, suppose that $n \in \mathtt{alter}(\mathtt{dom}(a))$, $\mathtt{contents}'(s,n) = \mathtt{contents}'(s',n)$ and $s \sim_u^{\mathtt{oc}} s'$. Then by definition, we have $\mathtt{contents}((s,f_s),n) = \mathtt{contents}'(s,n) = \mathtt{contents}'(s',n) = \mathtt{contents}((s',f_{s'}),n)$. Similarly, $(s,f_s) \sim_u^{\mathtt{oc}} (s',f_{s'})$. By RM2$'$ for $F_{as}(M)$, we obtain that $\mathtt{contents}((s\cdot a, f_{s\cdot a}),n) = \mathtt{contents}((s,f)\cdot a, n) = \mathtt{contents}((s',f')\cdot a, n) = \mathtt{contents}((s'\cdot a, f_{s'\cdot a}),n)$. This yields $\mathtt{contents}'(s\cdot a,n) = \mathtt{contents}'(s'\cdot a,n)$, as required.

We now use the conclusion of the previous paragraph to obtain RM1$_a$. Suppose that $s \sim_u^{\mathtt{oc}} s'$ and let $\mathtt{dom}(a) = u$. Since we have assumed that $\mathtt{observe}(u) \subseteq \mathtt{alter}(u)$, it follows by RM2$'$ that $s\cdot a \sim_u^{\mathtt{oc}} s'\cdot a$. By definition, this means that $(s\cdot a, f_{s\cdot a}) \sim_u^{\mathtt{oc}} (s'\cdot a, f_{s'\cdot a})$. By RM1 for $F_{as}(M)$, we have $\mathtt{obs}_u((s\cdot a, f_{s\cdot a})) = \mathtt{obs}_u((s'\cdot a, f_{s'\cdot a}))$. That is, $\mathtt{out}(s,a) = f_{s\cdot a}(u) = f_{s'\cdot a}(u) = \mathtt{out}(s',a)$, as required for RM1$_a$ on M.

For RM3, suppose $n \notin \mathtt{alter}(\mathtt{dom}(a))$. By RM3 on $F_{as}(M)$, we have $\mathtt{contents}((s\cdot a, f_{s\cdot a}),n) = \mathtt{contents}((s,f_s)\cdot a, n) = \mathtt{contents}((s,f_s),n)$. That is, $\mathtt{contents}'(s\cdot a,n) = \mathtt{contents}'(s,n)$, as required.

The argument from a weak access control structure on $\mathtt{reach}(F_{as}(M))$ to one on $\mathtt{reach}(M)$ is identical: we just need to note that if s is a reachable state of M then (s,f_s) is a reachable state of $F_{as}(M)$, by the properties of κ_u. □

We now obtain that the following are equivalent:

1. M is TA-secure with respect to $\rightarrowtail$,
2. $F_{as}(M)$ is TA-secure with respect to $\rightarrowtail$, (by Theorem 5)
3. there exists a (state-observed) weak access control interpretation on $\mathtt{uf}_s(F_{as}(M))$ consitent with $\rightarrowtail$, (by Theorem 6),
4. there exists a (state-observed) weak access control interpretation on $\mathtt{reach}(F_{as}(\mathtt{uf}_a(M)))$ consistent with $\rightarrowtail$, (by Lemma 6),
5. there exists an (action-observed) weak access control interpretation on $\mathtt{uf}_a(M)$ consistent with $\rightarrowtail$ (by Proposition 3, Proposition 4, the fact that $\mathtt{uf}_a(M)$ is output-recording, and the fact that $\mathtt{reach}(\mathtt{uf}_a(M)) = \mathtt{uf}_a(M)$).

Thus, we have the following result, analogous to Theorem 6.

Theorem 7. *Let M be an action-observed system. The following are equivalent:*

1. *M is TA-secure with respect to $\rightarrowtail$,*
2. *$\mathtt{uf}_a(M)$ admits a weak access control interpretation consistent with $\rightarrowtail$,*
3. *there exists a weak unwinding on $\mathtt{uf}_a(M)$ with respect to $\rightarrowtail$.*

6 A Comparison with Roscoe and Goldsmith

As already mentioned above, Roscoe and Goldsmith [RG99] (RG) have also proposed alternative definitions of intransitive noninterference. In this section we compare their definitions to those discussed above.

One impediment to a direct comparison is that RG work in the context of the process algebra CSP, which provides a rather different semantic basis for their definitions. Amongst the differences is that CSP makes no distinction between actions and observations. Nevertheless, RG aim to deal with a class of systems with observation-like events, but need to handle this through a variant of their main definition.

CSP provides an algebraic process notation based in an alphabet Σ. The elements of Σ are called "actions" in the literature, but to distinguish them from actions as in our state-machine model, we call them "events". From this basis, together with some special atomic processes such as `Stop` (a process which does nothing), the algebra is built up from operations including $a \rightarrow P$ (do event a first, then follow with the behaviour of process P), $P \sqcap Q$ (nondeterministically choose between doing P or Q), $P \,||_X\, Q$ (run processes P and Q in parallel, with events in X required to synchronise) and $P \setminus X$ (run process P, but hide any events in X).

In order to apply CSP to a security setting, RG assume the existence of a partition Π on Σ. They suppose that $\rightarrowtail$ is a binary relation on Π. Thus, Π corresponds to the set of security domains and the elements of $U \in \Pi$ correspond to the sets of events visible in domain U. In order to define whether a process is secure, they first introduce two operators on processes P:

1. the *lazy abstraction* operator $\mathcal{L}$ taking as an additional argument a set X of events, defined by
$$\mathcal{L}_X(P) = (P \,||_X\, \texttt{Chaos}_X) \setminus X$$
2. the *mixed abstraction* operator $\mathcal{M}$, taking as additional arguments sets of events X and S, defined by
$$\mathcal{M}^S_X(P) = (P \,||_{X \setminus S}\, \texttt{Chaos}_{X \setminus S}) \setminus X$$

Here $\texttt{Chaos}_X$ is the process defined by

$$\texttt{Chaos}_X = (\sqcap_{a \in X}\ a \rightarrow \texttt{Chaos}_X) \sqcap \texttt{Stop}.$$

That is, $\texttt{Chaos}_X$ presents either a finite number of events from X and stops, or presents an infinite number of events from X. The intuition for the lazy abstraction $\mathcal{L}_X(P)$ is that it represents the view of a user that is able to interact with the system P only through events $\overline{X}$, on the assumption that the behaviour of the other users, who may control any of the events X, is unknown. The mixed abstraction $\mathcal{M}^S_X(P)$ is intended to deal with situations where the events in S are "signal" events that are visible to, but not under the control of the users. Here, the signal events of the other users represented by X are still hidden from the view, but are not under the control of the process $\texttt{Chaos}_{X \setminus S}$, so cannot be blocked.

There are several semantics for the CSP process notation. The semantics most appropriate for use with the abstraction operators to give a definition of security is the *stable-failures* semantics (see [Ros97] for discussion of this point).

$$(a \to P) \xrightarrow{a} P \qquad P \sqcap Q \xrightarrow{\tau} P \qquad P \sqcap Q \xrightarrow{\tau} Q$$

$$\frac{P \xrightarrow{a} P', \quad a \notin X}{P \,||_X\, Q \xrightarrow{a} P' \,||_X\, Q} \qquad \frac{Q \xrightarrow{a} Q', \quad a \notin X}{P \,||_X\, Q \xrightarrow{a} P \,||_X\, Q'}$$

$$\frac{P \xrightarrow{a} P', \quad Q \xrightarrow{a} Q', \quad a \in X}{P \,||_X\, Q \xrightarrow{a} P' \,||_X\, Q'} \qquad \texttt{Stop} \,||_X\, \texttt{Stop} \xrightarrow{\tau} \texttt{Stop}$$

$$\frac{P \xrightarrow{a} Q, \quad a \in X}{P \setminus X \xrightarrow{\tau} Q \setminus X} \qquad \frac{P \xrightarrow{a} Q, \quad a \notin X}{P \setminus X \xrightarrow{a} Q \setminus X}$$

Fig. 4. Operational semantics for CSP

This can be conveniently stated in conjunction with an operational semantics for processes in terms of the space of *labelled transition systems*.

Labelled transition systems over an alphabet of events Σ are tuples of the form $L = \langle S, s_0, \rightarrow \rangle$, where S is a set of states, s_0 is an initial state, and $\rightarrow \subseteq S \times (\Sigma \cup \{\tau\}) \times S$ is a transition relation on S with edges labelled from the alphabet Σ or by a special event τ representing an *internal transition* of the system. We write $s \xrightarrow{a} t$ when $(s, a, t) \in \rightarrow$. A *run* of L is a sequence $r = s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n$. The corresponding *trace* of r is the sequence $a_1 \ldots a_n$ with any occurrences of τ deleted. The run r is *stable* if there does not exist an internal transition from s_n, i.e., for no state t do we have $s \xrightarrow{\tau} t$. A *failure* in an LTS L is a pair (σ, X) where σ is a trace and X is a subset of Σ such that there exists a stable run r of L with trace σ and final state s such that for no $a \in X$ is there a state t such that $s \xrightarrow{a} t$. We write $\texttt{traces}(L)$ for the set of traces of L and $\texttt{failures}(L)$ for the set of failures of L. Process terms can be associated with a labelled transition system by the rules in Figure 4. Given a process term P, the stable failures semantics of P is the pair $(\texttt{traces}(L), \texttt{failures}(L))$ where L is the LTS associated to P.

A CSP process P is *deterministic* if for all $\sigma \in \texttt{traces}(P)$ and all $c \in \Sigma$, it is not the case that both $\sigma c \in \texttt{traces}(P)$ and $(\sigma, \{c\}) \in \texttt{failures}(P)$. RG similarly define a process P to be *locally deterministic* in a set C of events, if for all $\sigma \in \texttt{traces}(P)$ and all $c \in C$, it is not the case that both $\sigma c \in \texttt{traces}(P)$ and $(\sigma, \{c\}) \in \texttt{failures}(P)$. Intuitively, this says that if c is possible after a run with trace σ, then all stable runs with trace σ can be extended by c.

RG's definition of security is now given as follows. Given the partition Π of Σ, a policy $\rightarrowtail \subseteq \Pi \times \Pi$ and an element U of Π, the set $\texttt{noflow}(V)$ is defined to be the union of the sets of events $V \in \Pi$ such that $V \not\rightarrowtail U$.

Definition 5. *A pair (P, Π) consisting of a process P and a partition Π of its alphabet is* $\mathcal{L}$-secure *with respect to $\rightarrowtail$ if $\mathcal{L}_{\texttt{noflow}(U)}(P)$ is locally deterministic in U for all $U \in \Pi$.*

Intuitively, $\mathcal{L}$-security says that for all domains $U \in \Pi$, the possibility of events U depends only on the past observed events from $\Sigma \setminus \texttt{noflow}(U)$, and

cannot be affected by the behaviour of the processes that are not permitted to communicate directly with U (as modelled by $\mathtt{Chaos}_{\mathtt{noflow}(U)}$).

RG also allude to a definition based on the mixed abstraction operator, but do not give a formal statement.[3] We assume here that what is meant is the following:

Definition 6. *(P, Π) is $\mathcal{M}^S$-secure (where S is a set of signals), if $\mathcal{M}^S_{\mathtt{noflow}(U)}(P)$ is locally deterministic in U for all $U \in \Pi$.*

In order to compare our definitions on state and action-observed systems with these definitions for CSP, we need to translate between these semantic domains. We choose here to translate our systems models to CSP, since the latter seems to be more general. However, as shown in [MZ06] there are several plausible candidates for a security-preserving translation between state-based models and process algebraic models.

We begin with a consideration of mappings from state-observed systems. Suppose we are given a state-observed system $M = \langle S, s_0, A, \mathtt{step}, \mathtt{obs}, \mathtt{dom}\rangle$ with set of domains D and observations O. Given an agent u, write O_u for the set of possible observations of agent u, i.e., $O_u = \{\mathtt{obs}_u(s) \mid s \in S\}$. As noted above, we may assume without loss of generality that if $u \neq u'$ then O_u and $O_{u'}$ are disjoint.

Our first translation constructs a pair $\mathtt{T}_1(M) = (\mathtt{L}_1(M), \Pi(M))$ consisting of an LTS $\mathtt{L}_1(M)$ and a partition $\Pi(M)$ over the alphabet $A \cup O$. The LTS $\mathtt{L}_1(M) = \langle S, s_0, \rightarrow\rangle$ has the same states and initial state as M, and transition relation defined by $s \xrightarrow{x} t$ if either $x \in A$ and $t = s \cdot x$ or $x \in O$ and $s = t$ and $\mathtt{obs}_u(s) = x$ for some $u \in D$. That is, we add self-looping transitions to each state for each of the observations that are made in these states. The partition $\Pi(M)$ consists of the the sets $E_u = A_u \cup O_u$ for $u \in D$. Thus, both the actions and observations of domain u are treated as events of a domain in $\mathtt{T}_1(M)$. This translation, in effect, makes the observations *optional* in $\mathtt{T}_1(M)$, whereas they compulsory in M — note that we have treated them so in the definition of the functions $\mathtt{view}_u$. However, exactly this approach to translation has been shown to yield exact correspondences for a range of security definitions in [MZ06]. The intuition for these correspondences is that a system is *insecure* if it is *possible* for a user to obtain enough information to deduce a secret, so the possibility of observations is all that needs to be preserved in a translation. It is therefore appropriate to investigate the impact of this translation for the RG definitions.

We also need to translate policies $\rightarrowtail \subseteq D \times D$ into RG's format. This is done by defining $\mathtt{T}_1(\rightarrowtail) = \{(E_u, E_v) \mid u \rightarrowtail v\}$.

The following result of RG (Theorem 1 in [RG99]) gives a simpler characterization of $\mathcal{L}$-security that will be useful in what follows. If σ is a sequence of events and X is a set of events, write $\sigma \setminus X$ for the subsequence of events in σ that are not in X.

[3] They say only "Given a subset S of events of the alphabet of P that are signals, it is clear how to define a corresponding notion using mixed abstraction", and confine their attention for the rest of the paper to the lazy abstraction.

Proposition 5. *If P is deterministic, then (P, Π) is secure with respect to $\rightarrowtail$ iff for all domains $U \in \Pi$ and $\sigma, \sigma' \in \mathtt{traces}(P)$, if $\sigma \setminus \mathtt{noflow}(U) = \sigma' \setminus \mathtt{noflow}(U)$ then for all $c \in U$, $\sigma c \in \mathtt{traces}(P)$ iff $\sigma' c \in \mathtt{traces}(P)$.*

We can then obtain the following correspondence:

Proposition 6. *$\mathtt{T}_1(M)$ is $\mathcal{L}$-secure with respect to $\mathtt{T}_1(\rightarrowtail)$ iff M is P-secure with respect to $\rightarrowtail$.*

Proof. It is easily seen that $\mathtt{L}_1(M)$ is deterministic, so we we may apply Proposition 5 to $\mathtt{T}_1(M)$. We first assume that M is not P-secure with respect to $\rightarrowtail$, and show that $\mathtt{T}_1(M)$ is not $\mathcal{L}$-secure with respect to $\mathtt{T}_1(\rightarrowtail)$. By assumption, there exist $\alpha, \alpha' \in A^*$ and $u \in D$ such that $\mathtt{purge}_u(\alpha) = \mathtt{purge}_u(\alpha')$ but $\mathtt{obs}_u(s_0 \cdot \alpha) \neq \mathtt{obs}_u(s_0 \cdot \alpha')$. Note that α and α' are both traces of $\mathtt{T}_1(M)$, and $\alpha \setminus \mathtt{noflow}(E_u) = \mathtt{purge}_u(\alpha) = \mathtt{purge}_u(\alpha') = \alpha' \setminus \mathtt{noflow}(E_u)$. We also have that $\alpha\, \mathtt{obs}_u(s_0 \cdot \alpha) \in \mathtt{traces}(\mathtt{T}_1(M))$ but not $\alpha'\, \mathtt{obs}_u(s_0 \cdot \alpha) \in \mathtt{traces}(\mathtt{T}_1(M))$. Since $\mathtt{obs}_u(s_0 \cdot \alpha) \in E_u$ it follows that $\mathtt{T}_1(M)$ is not $\mathcal{L}$-secure with respect to $\mathtt{T}_1(\rightarrowtail)$.

Conversely, suppose that $\mathtt{T}_1(M)$ is not $\mathcal{L}$-secure with respect to $\mathtt{T}_1(\rightarrowtail)$. By way of witness for this fact, let σc and σ' be traces of $\mathtt{T}_1(M)$, where $\sigma' c$ is not a trace, $\sigma \setminus \mathtt{noflow}(E_u) = \sigma' \setminus \mathtt{noflow}(E_u)$, and $c \in E_u$. Note first that whenever σ is a trace and $a \in A$, then σa is also a trace. Since $\sigma' c$ is not a trace, we must have $c \in O_u$. Define α to be the subsequence of σ consisting of actions in A and similarly let α' be the subsequence of actions in σ'. Note that $\mathtt{purge}_u(\alpha) = \alpha \setminus \mathtt{noflow}(E_u) = (\sigma \setminus (\mathtt{noflow}(E_u)) \setminus O$ and similarly for σ', so we obtain that $\mathtt{purge}_u(\alpha) = \mathtt{purge}_u(\alpha')$. Since all the transitions in $\mathtt{T}_1(M)$ labelled from O are self-loops, we have that αc and α' are traces of $\mathtt{T}_1(M)$, but not $\alpha' c$. It follows that $\mathtt{obs}_u(s_0 \cdot \alpha) = c \neq \mathtt{obs}_u(s_0 \cdot \alpha')$, so α and α' provide a witness showing that M is not P-secure with respect to $\rightarrowtail$. □

Thus, on this translation, RG's definition says that a system is secure if observations in a domain u depend only on the actions that have been performed in domains permitted to interfere with u — although the events in such domains also contain observations, no dependence on these observations is allowed. However, there seem to be some intuitive grounds that RG's definitions are intended to allow such dependencies.

This suggests that we should also consider the mixed abstraction, which is designed to deal with "signal events", for which the intuition seems to be similar to observations. It would seem that we should take S, the set of signal events, to be equal to O. One immediate obstacle to such an application of the mixed abstraction to $\mathtt{T}_1(M)$, however, is that the self-loops for the observations in this process generate τ-transitions (in fact, *divergences*, i.e. infinite sequences of τ transitions) in the mixed abstraction at every state, so that the process has no stable failures. Roscoe [Ros97](p. 307) stipulates that the mixed abstraction is inapplicable to such processes.

We therefore would need another translation to be able to apply the mixed abstraction to state-observed systems. If we are to capture any dependencies

on observations, moreover, the translation should treat observations as obligatory rather than optional. How to define a translation that achieves this is far from clear. The divergence-causing self-loops, or something like them, seem to be necessary to capture the asynchronous nature of observation in the state-observed model. Consider a transition from state s to state t on action a, where the observations made by the agents A, $B, \ldots$ are $\mathtt{obs}_A(t) = o_A$ and $\mathtt{obs}_B(s) = \mathtt{obs}_B(t) = o_B, \ldots$. One way we could represent this in the LTS is by means of a sequence of transitions

$$s \xrightarrow{a} (s, a, A) \xrightarrow{o_A} (s, a, B) \xrightarrow{o_B} \ldots t.$$

However, there is then the risk of the translation failing because the occurrence of the event o_B would signal to B that someone has performed an action, even if $dom(a) \not\rightarrowtail B$. To avoid this, we need that the event o_B is continuously available between states s and t. This seems to require a construction that will cause divergences in the abstraction.

It therefore does not seem that the mixed abstraction is applicable to any class of processes that can model state-observed systems. We therefore turn instead to a consideration of a translation from the action-observed model, which the CSP theory seems better suited to handle.

Suppose we are given an action-observed system $M = \langle S, s_0, A, \mathtt{step}, \mathtt{out}, \mathtt{dom}\rangle$ with domains D and observations O. As above, we assume without loss of generality that the sets $O_u = \{\mathtt{out}(s, a) \mid s \in S, a \in A, \mathtt{dom}(a) = u\}$ of possible observations of agents $u \in D$ are disjoint. Our second translation constructs a pair $\mathtt{T}_1(M) = (\mathtt{L}_3(M), \Pi(M))$ consisting of an LTS $\mathtt{L}_3(M)$ and a partition $\Pi(M)$ over the alphabet $A \cup O$. The LTS $\mathtt{L}_3(M) = \langle S', s_0, \rightarrow\rangle$ has

- States $S' = S \cup S \times A$,
- initial state s_0,
- transition relation defined by $s \xrightarrow{x} t$ if either
  - $s \in S$, $x \in A$ and $t = (s, x)$, or
  - $s = (z, a) \in S \times A$ and $x \in O$ and $t = z \cdot a \in S$ and $x = \mathtt{out}(z, a)$.

That is, we represent the occurrence of action a in state s with output o to $\mathtt{dom}(a)$ by means of a sequence of two transitions:

$$s \xrightarrow{a} (s, a) \xrightarrow{o} s \cdot a.$$

As above, the partition $\Pi(M)$ consists of the the sets $E_u = A_u \cup O_u$ for $u \in D$. We define $\mathtt{T}_2(\rightarrowtail)$ to be identical to $\mathtt{T}_1(\rightarrowtail)$.

The following relates $\mathcal{M}^O$-security to our notions of security on action-observed machines.

Proposition 7. *Let M be an action-observed machine with observations O. Then $\mathtt{T}_2(M)$ is $\mathcal{M}^O$-secure with respect to $\mathtt{T}_2(\rightarrowtail)$ iff M is ITO-secure.*

Proof. For $u \in D$, write $\mathtt{nf}(u)$ for $\bigcup_{v \not\rightarrowtail u} A_v \cup O_v$. Note that this is precisely $\mathtt{noflow}(E_u)$. We prove that $\mathtt{T}_2(M)$ is $\mathcal{M}^O$-secure with respect to $\mathtt{T}_2(\rightarrowtail)$ iff for all

$u \in D$ and for all sequences $\alpha, \alpha' \in A^*$ with $\mathtt{trace}(\alpha)\setminus\mathtt{nf}(u) = \mathtt{trace}(\alpha')\setminus\mathtt{nf}(u)$ and $a \in A_u$ we have $\mathtt{out}(s_0 \cdot \alpha, a) = \mathtt{out}(s_0 \cdot \alpha', a)$. By Proposition 1, this is equivalent to ITO-security.

Suppose first that there exist sequences α, α' and $a \in A_u$ such that $\mathtt{trace}(\alpha)\setminus\mathtt{nf}(u) = \mathtt{trace}(\alpha')\setminus\mathtt{nf}(u)$ but $\mathtt{out}(s_0{\cdot}\alpha, a) \neq \mathtt{out}(s_0{\cdot}\alpha', a)$. We show $\mathcal{M}^O_{\mathtt{nf}(u)}(\mathtt{T}_2(M))$ is not locally deterministic. Let r be a run of $\mathtt{T}_2(M) \,||_{\mathtt{nf}(u)\setminus O}\, \mathtt{Chaos}_{\mathtt{nf}(u)\setminus O}$ in which the first component has trace $\mathtt{trace}(\alpha)a$ and the second component offers actions from $\mathtt{nf}(u)$ as needed to coordinate with this behavior of the first component, and then proceeds to $\mathtt{Stop}$. Let r' be a run similarly constructed with respect to α'. Then, at the end of r, the first component is in state $(s_0 \cdot \alpha, a)$, so the only event enabled is $\mathtt{out}(s_0 \cdot \alpha, a)$, and similarly for the second run with respect to $\mathtt{out}(s_0 \cdot \alpha', a)$. Both runs generate the trace $\sigma = \mathtt{trace}(\alpha)a \setminus \mathtt{nf}(u)$ of $\mathcal{M}^O_{\mathtt{nf}(u)}(\mathtt{T}_2(M))$. The first gives that $\sigma\,\mathtt{out}(s_0 \cdot \alpha, a)$ is a trace of $\mathcal{M}^O_{\mathtt{nf}(u)}(\mathtt{T}_2(M))$ and the second gives that $(\sigma, \{\mathtt{out}(s_0 \cdot \alpha, a)\})$ is a failure. Hence $\mathcal{M}^O_{\mathtt{nf}(u)}(\mathtt{T}_2(M))$ is not $\mathcal{M}^O$-secure.

Conversely, suppose that $\mathtt{T}_2(M)$ is not $\mathcal{M}^O$-secure. By way of witness, let σ be a trace of $\mathcal{M}^O_{\mathtt{nf}(u)}(\mathtt{T}_2(M))$ and $c \in E_u$ be an event such that σc is a trace of this process but $(\sigma, \{c\})$ is a failure. Note that any trace must be a sequence of alternating actions and observations in E_u. Thus, if $c \in A_u$, then either $\sigma = \epsilon$ or the final event in σ is an observation. This means that at the end of any run with trace σ, the first process in the composition $\mathtt{T}_2(M) \,||_{\mathtt{nf}(u)\setminus O}\, \mathtt{Chaos}_{\mathtt{nf}(u)\setminus O}$ must be in a state $s \in S$. But this means that $c \in A_u$ is enabled at s, so $(\sigma, \{c\})$ cannot be a failure. We must therefore have that $c \in O_u$ and that the last event of σ is an action $a \in A_u$. Let r be a run of $\mathtt{T}_2(M) \,||_{\mathtt{nf}(u)\setminus O}\, \mathtt{Chaos}_{\mathtt{nf}(u)\setminus O}$ that can be extended by the event c and similarly let r' be a run giving rise to the failure $(\sigma, \{c\})$. Define αa and $\alpha' a$ to be the sequences $\mathtt{trace}(r) \setminus O$ and $\mathtt{trace}(r') \setminus O$ in A^*, respectively. Then at the end of runs r and r', the first process in the composition must be at the states $(s_0{\cdot}\alpha, a)$ and $(s_0{\cdot}\alpha', a)$, respectively. It follows that $c = \mathtt{out}(s_0 \cdot \alpha, a) \neq \mathtt{out}(s_0 \cdot \alpha', a)$ since these are the only events available at these states. Since $\mathtt{trace}(\alpha) \setminus \mathtt{nf}(u) = \sigma = \mathtt{trace}(\alpha') \setminus \mathtt{nf}(u)$, this provides a witness showing that the condition of the characterization is false.

This result shows that the meaning of RG's definitions is sensitive to how one chooses to model systems. Whereas, on our first translation, $\mathcal{L}$-security corresponds to P-security on both state and action observed systems, on this second translation, the more general notion of $\mathcal{M}^O$-security corresponds to the weaker notion of ITO-security.

7 Conclusion

We have established several correspondences between a range of definitions of intransitive noninterference based in several different semantic models of systems: state-observed state-machines, action-observed state-machines, and the process algebra CSP. The notions of P-security, TO-security, ITO-security, TA-security and IP-security as defined on action-observed systems correspond directly to

the similarly named notions as defined on state-observed systems, under a natural transformation from the action-observed to the state-observed domain. RG's notions of security on CSP processes correspond either to P-security or ITO-security, depending on which of two natural mappings from state-machines to CSP one uses.

Our results have left open a number of questions. In dealing with state- and action observed systems, we confined our attention to deterministic systems. It remains to explore the generalization of the definitions we have considered to nondeterministic systems and systems that are not input-enabled, as has been studied for IP-security by von Oheimb [Ohe04]. His work should be reconsidered in the light of our results. More generally, one could consider extensions of these definitions to the richer semantic framework of process algebra. We have shown some specific correspondences with RG's definitions under our mappings of deterministic state- and action-observed machines, but it would also be of interest to give definitions in the process algebraic setting that correspond to the other definitions of noninterference we have discussed. A deeper exploration of the question raised above concerning the interpretability of the state-observed model in a process algebraic setting would also be of interest.

A starting point of RG's work was difficulties that they had in treating the specific example of downgraders using IP-security. We have not attempted to address this particular example in our work, so also leave open this issue of pragmatics and applications for the spectrum definitions we have studied.

However, the correspondence between $\mathcal{M}^O$-security and ITO-security on action-observed systems brings out a point that may not have been apparent from RG's presentation: in a chain such as $H \rightarrowtail D \rightarrowtail L$, according to $\mathcal{M}^O$-security, the mere fact that D has *observed* information about H is sufficient for L to be permitted to know this information. In particular, anything that D learns from the observation $\mathtt{out}(s, a)$ obtained as a result of performing the action a is permitted by $\mathcal{M}^O$-security to have been transmitted to L without further activity by D.

If one considers this behaviour insecure, we note that it would be even more pronounced in any CSP modelling of state-observed systems, where it would be possible for D to observe consequences of H actions without performing any actions. Here, H could maintain security of the system by sending information directly to L, but ensuring that this information is simultaneously reflected in D's observations. This may amount to an intuitive notion of security if D's ability to *audit* information flows from H to L is of primary concern, but it would not correspond to a policy that requires D to *control* such information flows. We note that both TA-security and IP-security permit D to transmit information to L that D has not observed, so are even more permissive than $\mathcal{M}^O$-security in this regard. Indeed, RG's treatment of downgraders is based on $\mathcal{L}$-security (equivalent to the strongest notion of P-security) but needs to make some very strong assumptions, viz. that the content being downgraded is encoded in the name of the action.

To better understand these definitions, it would be very helpful to have a set of worked examples to clarify the circumstances under which the use of each is appropriate, as well as to elucidate the role that they might play in the derivation of other desirable properties of systems. For the specific case of downgraders, recent work of Chong and Myers [CM04], Mantel and Sands [MS04], Bossi et al [BPR04] and Sabelfeld and Sands [SS05] is of relevance.

References

[BPR04] A. Bossi, C. Piazza, and S. Rossi. Modelling downgrading in information flow security. In *Proc. IEEE Computer Security Foundations Workshop*, pages 187–201, 2004.

[CM04] S. Chong and A. C. Myers. Security policies for downgrading. In *11th ACM Conf. on Computer and Communications Security (CCS)*, Oct 2004.

[GM82] J.A. Goguen and J. Meseguer. Security policies and security models. In *Proc. IEEE Symp. on Security and Privacy*, pages 11–20, Oakland, 1982.

[GM84] J.A. Goguen and J. Meseguer. Unwinding and inference control. In *IEEE Symp. on Security and Privacy*, 1984.

[HY87] J.T. Haigh and W.D. Young. Extending the noninterference version of MLS for SAT. *IEEE Trans. on Software Engineering*, SE-13(2):141–150, Feb 1987.

[MS04] H. Mantel and D. Sands. Controlled declassification based on intransitive noninterference. In *Proc. Asian Symp. on Programming Languages and Systems*, volume 3302 of *LNCS*, pages 129–145. Springer-Verlag, November 2004.

[MZ06] R. van der Meyden and C. Zhang. A comparison of semantic models for noninterference. In *Proc. Workshop on Formal Aspects of Security and Trust, Hamilton, Ontario, Canada,*, LNCS. Springer, August 2006. to appear, extended version at `http://www.cse.unsw.edu.au/~meyden/research/publications.html`.

[Ohe04] D. von Oheimb. Information flow control revisited: Noninfluence = Noninterference + Nonleakage. In *Computer Security – ESORICS 2004*, volume 3193 of *LNCS*, pages 225–243, 2004.

[RG99] A. W. Roscoe and M. H. Goldsmith. What is intransitive noninterference? In *IEEE Computer Security Foundations Workshop*, pages 228–238, 1999.

[Ros97] A.W Roscoe. *The theory and practice of concurrency.* International Series in Computer Science. Prentice Hall, 1997.

[Rus92] J. Rushby. Noninterference, transitivity, and channel-control security policies. Technical Report CSL-92-02, SRI International, Dec 1992.

[SS05] Andrei Sabelfeld and David Sands. Dimensions and principles of declassification. In *Proceedings of the 18th IEEE Computer Security Foundations Workshop*, pages 255–269. IEEE Computer Society Press, 2005.

[vdM07] R. van der Meyden. What, indeed, is intransitive noninterference (extended abstract). In J. Biskup and J. Lopez, editors, *Proc. European Symp. on Research in Computer Security (ESORICS)*, volume 4734 of *Springer LNCS*, pages 235–250, 2007. Full paper at `http://www.cse.unsw.edu.au/~meyden/research/publications.html`.