# Semantics for Specialising Attack Trees based on Linear Logic [1]

Ross Horne[a], Sjouke Mauw[b], Alwen Tiu[a]

[a] *School of Computer Science and Engineering, Nanyang Technological University, Singapore*
[b] *CSC/SnT, University of Luxembourg*

**Abstract**

Attack trees profile the sub-goals of the proponent of an attack. Attack trees have a variety of semantics depending on the kind of question posed about the attack, where questions are captured by an attribute domain. We observe that one of the most general semantics for attack trees, the multiset semantics, coincides with a semantics expressed using linear logic propositions. The semantics can be used to compare attack trees to determine whether one attack tree is a *specialisation* of another attack tree. Building on these observations, we propose two new semantics for an extension of attack trees named *causal attack trees*. Such attack trees are extended with an operator capturing the causal order of sub-goals in an attack. These two semantics extend the multiset semantics to sets of series-parallel graphs closed under certain graph homomorphisms, where each semantics respects a class of attribute domains. We define a sound logical system with respect to each of these semantics, by using a recently introduced extension of linear logic, called MAV, featuring a non-commutative operator. The non-commutative operator models causal dependencies in causal attack trees. Similarly to linear logic for attack trees, implication defines a decidable preorder for specialising causal attack trees that soundly respects a class of attribute domains.

*Keywords:* attack trees, semantics, causality, linear logic, calculus of structures

## 1. Introduction

Attack trees were introduced by Schneier [1] as a means of representing the profile of an attacker. To achieve some goal such as obtaining a company secret, the proponent of an attack may proceed according to sub-goals. In the original model proposed by Schneier, there are two ways of breaking down a goal: either a goal is composed of multiple sub-goals, all of which must be achieved; or a goal can be tackled by one of several alternative sub-goals, where the success of one sub-goal is sufficient. These we refer to as conjunctive and disjunctive refinements, respectively.

In this work, we consider attack trees extended with a sequential refinement [2], called *causal attack trees*. Sequential refinement, like conjunctive refinement, requires that all subgoals must be achieved, but furthermore some sub-goals must be achieved before other sub-goals may be tackled. For example, an attacker must first successfully enter a datacenter before attempting to install malicious hardware. Sequential refinement is distinguished from conjunctive refinement, since the latter assumes multiple sub-goals are attempted concurrently and independently by the attacker.

We build on a line of work investigating the semantics of attack trees initiated by Mauw and Oostdijk [3], that explores conditions under which two attack trees can be considered to represent the same attack. If an algorithm or security expert modifies an attack tree, a semantics is desirable to know whether properties of the attack tree are preserved in the new attack tree. A subtlety is that the appropriate semantics depend on the kind of question the attack tree is used to resolve, where questions are characterised by *attribute domains*. For example, for Boolean attribute

[1] ©2016. This manuscript version is made available under the CC-BY-NC-ND 4.0 license. This is a technical report supporting journal article: Semantics for Specialising Attack Trees based on Linear Logic. Fundamenta Informaticae 153 (2017) 57-86. DOI:10.3233/FI-2017-1531. IOS Press. Corresponding author: Ross Horne.
*Email address:* rhorne@ntu.edu.sg (Ross Horne)

domains such as "whether special equipment is required", a semantics based on classical propositional logic [4] suffices. A more general semantics based on multisets [3] covers a wider class of questions, called *distributive attribute domains*, such as "minimum cost of attack" or "maximum damage of attack".

Related work [2] extends the multiset semantics to causal attack trees by using series-parallel graphs that represent not only the actions of the attack, but also causal dependencies between actions. Under the previously established equational semantics for causal attack trees, given two causal attack trees and an attribute domain sound with respect to the semantics, if two causal attack trees are equivalent then the attribute domain will always give the same answer for both trees.

In practice, however, attack trees are not static snapshots, but dynamic descriptions of a system's vulnerabilities that evolve under influence of the system's development and new insights in the adversary's capabilities. New attacks will be added over time and existing attacks will be made more specific. An equational semantics does not comply to such an incremental development of attack trees because it only allows one to compare whether two trees are exactly equivalent. This implies the need for a semantics that explicitly addresses the question of whether one attack tree is a specialisation of another attack tree. In this article, we address this need by studying various semantics that preorder attack trees, hence are suitable for specialising attack trees. This is a challenging problem, since, in particular for causal attack trees, certain attribute domains that have the same equational semantics have subtly distinct inequational semantics.

The semantics introduced in this work are based on a logical system in which implication captures specialisation. Such a logical foundation for causal attack trees has practical benefits. The logic is defined by a decidable proof calculus that can be used as the basis of tools [5] to assist the security expert working with causal attack trees. Apart from the above mentioned scenario in which a dynamic attack becomes more and more specialised over time, there are other use cases for an inequational semantics. The security expert may begin with a large attack tree representing many system vulnerabilities and prune the attack tree just for an aspect of the system under consideration. Alternatively, tools may guide the security expert when constructing a large attack tree for a system, working bottom-up from a forest of attack trees which must be soundly reflected in the combined attack tree. Another use case for specialisation is when one attack tree represents the capabilities of an attacker and another attack tree represents the vulnerabilities of a system, in which case implication can be used to resolve queries such as "is the attacker capable of attacking the system?".

The logical semantics we propose is based on an extension of linear logic [6] with a non-commutative operator representing sequentiality, called MAV [7, 8]. A philosophical benefit of having a logical semantics for causal attack trees is that a logical system provides an objective justification for a choice of semantics. The rules of MAV are not designed based on human intuition, but are given canonically when investigating the proof theory of a logic with sequential composition and non-deterministic choice. Thus, not only does the logic provide a sound methodology for specialising and querying causal attack trees, but the semantics itself is fundamentally consistent.

**Outline.** Section 2 provides examples of causal attack trees, along with intuitive explanations for why implications between causal attack trees are of significant impact for the attack tree community. Section 3 is mainly background on attack trees with conjunctive and disjunctive refinement only. The key observation is that the established multiset semantics for attack trees coincides with a semantics based on a fragment of linear logic. The main technical contribution is in Sections 4 and 5, which explore semantics for causal attack trees. Section 4 concerns two semantics, based on sets of graphs closed under certain graph homomorphisms, and their sound and complete axiomatisations. The axiomatisations are used to identify two classes of attribute domain for which each semantics can be soundly applied. Section 5 introduces a logical system for which implication in the logic is sound with respect to the proposed semantics for causal attack trees, and thereby also sound with respect to the two classes of attribute domain identified.

## 2. A Case for the Specialisation of Causal Attack Trees

This section intuitively discusses motivating examples of causal attack trees and associated attribute domains. An explanation is provided for how one causal attack trees may be a specialisation of another causal attack tree. A case is made for a system formalising specialisation preorders that are sound with respect to attribute domains. Formal definitions are postponed until later sections.

Consider the example of a causal attack tree in Fig. 1, adapted from [9]. The attack tree features three types of refinement: the root node in the example with arc between the branches represents a conjunctive refinement, the

node with no arc represents a disjunctive refinement, the node with a directed arc pointing to the right represents a sequential refinement.

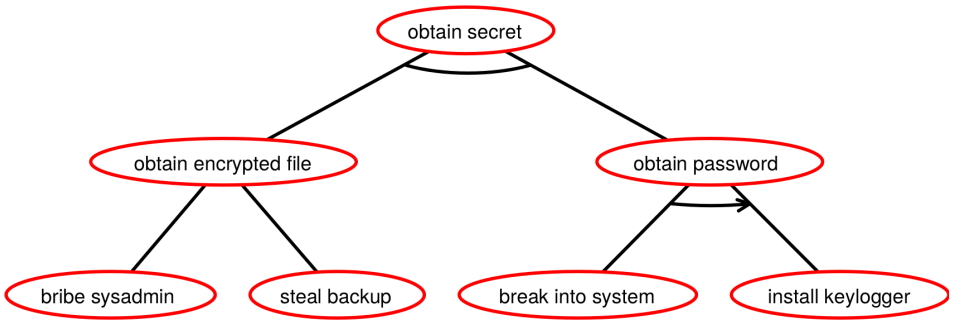



Figure 1: Attack tree for accessing encrypted data.

The goal at the root of the tree is to obtain some secret data. To achieve this, an encrypted file should be obtained *and* a password should be obtained. The goal of obtaining the encrypted file is disjunctively refined to either bribing the sysadmin *or* stealing a backup. The goal of obtaining the password is sequentially refined such that the attacker must first break into the system *before* installing a key logger.

Under certain assumptions, the causal attack tree in Fig. 1 can be specialised to the causal attack tree in Fig. 2. In the specialised tree, "steal backup" can only be performed after breaking into the system. Intuitively, the tree in Fig. 2 represents a more restricted attack pattern. In Fig. 2, there is an extra dependency between the actions "break into system" and "steal backup" not present in Fig. 1.

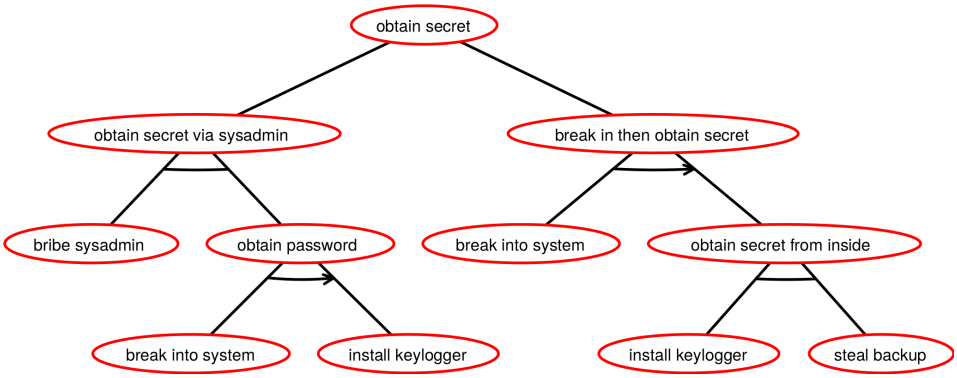



Figure 2: Specialised causal attack tree where the backup can only be stolen from inside the system.

*Soundness with respect to attribute domains.* Whether the specialisation from Fig. 1 to Fig. 2 is sound depends on the attribute domains of interest. The criterion for soundness, that is formally defined in later sections is as follows.

> If one tree specialises another tree, then the specialisation is sound with respect to an attribute domain whenever there is a correlation between the values at the root of the trees, for any assignment of values to basic actions at the leaves.

If we are interested in the "minimal attack time", we expect that in the more restricted attack tree in Fig. 2 any attack will take at least as long as in the original tree. This is due to the implicit assumption that in Fig. 1 "break into system" and "steal backup" can be conducted concurrently; whereas in Fig. 2 the same actions are causally dependent on each other, hence must be performed sequentially. When independent actions occur concurrently the minimum

attack time is the maximum of the times assigned to the sub-goals; whereas when causally dependent actions occur in sequence the minimum attack time should be the sum of the minimum attack time of each sub-goal.

Suppose the minimum times to achieve each basic action are assigned as follows.

$$\boxed{\text{bribe sysadmin}} \mapsto 25 \quad \boxed{\text{steal backup}} \mapsto 5 \quad \boxed{\text{break into system}} \mapsto 9 \quad \boxed{\text{install keylogger}} \mapsto 2$$

For the above assignment the tree in Fig. 2 has a higher minimum attack time, as verified by the following calculations. The formula on the left is the minimum attack time for Fig. 1 and the formula on the right is the minimum attack time for Fig. 2.

$$\max\{\min\{25, 5\}, 9 + 2\} = 11 \leq 14 = \min\{\max\{25, 9 + 2\}, 9 + \max\{2, 5\}\}$$

Indeed, for any assignment of non-negative times, the minimum attack time will always be at least as high for the attack tree in Fig. 2 compared to Fig. 1. Thereby an attacker constrained according to Fig. 2 will never be capable of attacking quicker than an attacker following the profile in Fig. 1.

Two natural questions, clarified in this work, are: *(i)* how do we determine when one causal attack tree is a sound specialisation of another causal attack tree; and *(ii)* what attribute domains are soundly preserved with respect to such specialisations. These questions contrast to previous work on causal attack trees [2], where only equivalence and the preservation of equality with respect to attribute domains is considered. With respect to equality, we can say nothing about the relationship between the two example trees in Fig. 1 and 2.

*Distinct classes of attribute domain.* Not all attribute domains have the same algebraic properties, hence one specialisation preorder does not fit all purposes. We contrast the "minimum attack time" attribute domain to other attribute domains that are also sound for the specialisations between Fig. 1 and 2, but unsound for other specialisations.

Now suppose that we are interested in the "minimum number of experts required to conduct an attack". For this attribute domain, the same expert can conduct actions in sequence, hence sequential composition is interpreted using max. However, if actions are conducted independently and concurrently then the total number of experts required is the sum of the number of experts required for each concurrent sub-goal. For this domain we assume that for conjunctive refinements the experts must be prepared to conduct actions independently and concurrently.

For the "minimum number of experts" attribute domain we can guarantee that the causal attack tree in Fig. 1 always requires at least as many experts compared to the causal attack tree in Fig. 2. Hence specialising Fig. 1 to Fig. 2 is sound with respect to the "minimum number of experts" attribute domain. Consider the same causal attack trees and consider the following assignment of values to basic actions under the "minimum number of experts" attribute domain.

$$\boxed{\text{bribe sysadmin}} \mapsto 3 \quad \boxed{\text{steal backup}} \mapsto 1 \quad \boxed{\text{break into system}} \mapsto 2 \quad \boxed{\text{install keylogger}} \mapsto 1$$

For the above assignment of values, the number of attackers required for the trees in Fig. 1 and Fig. 2 can be compared by the following calculation. The formulas on the left and right represent the minimum number of experts for Fig. 1 and Fig. 2, respectively.

$$\min\{3, 1\} + \max\{2, 1\} = 3 \geq 2 = \min\{3 + \max\{2, 1\}, \max\{2, 1 + 1\}\}$$

For the specialisation from Fig. 1 to Fig. 2, for both attribute domains discussed, the values at the root nodes are correlated for any assignment of values to basic actions at the leaves. Thereby both attribute domains discussed are sound for this specialisation, although the numerical order for each of the two attribute domains is opposite to the other: the specialised attack in Fig. 2 always requires at least as much time, but never more experts.

In the body of this paper, a formal logical system is introduced that can be used to decide whenever such sound specialisations exist, without requiring to manually check soundness for each new combination of specialisation and attribute domain. Without a supporting decision procedure, it may not be immediately obvious when specialisations are sound for one attribute domain but not another.

Consider the causal attack trees in Fig. 3. According to the "minimum number of experts" attribute domain, the causal attack tree in Fig. 3(a) is a sound specialisation of the causal attack tree in Fig. 2. Any assignment of values will lead to at least as many experts being required for Fig. 3(a) compared to Fig. 2. In contrast, for the "minimum

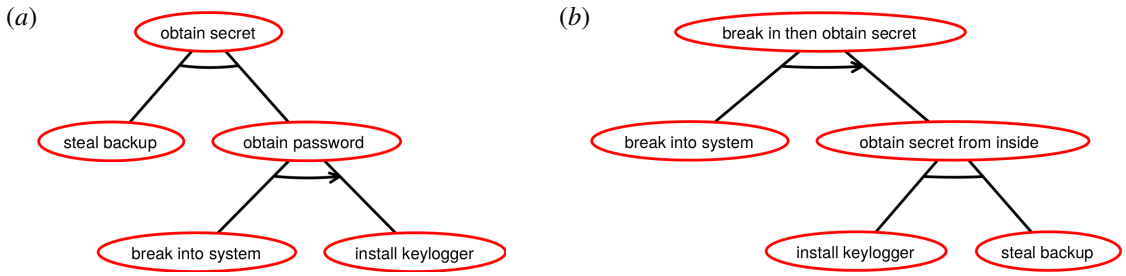



Figure 3: Two causal attack trees: (a) is a sound specialisation of Fig. 2 with respect to "minimum number of experts"; (b) is a sound specialisation of Fig. 1 with respect to "minimum attack time".

attack time" attribute domain, there exist distinct assignments of values to basic actions at the leaves that result in both smaller and greater times for Fig. 3(a) compared to Fig. 2. Given the first assignment of attack times provided above, the minimum attack time for Fig. 3(a) would be $\max\{5, 9 + 2\} = 11$ which is less than the minimum attack time of 14 for Fig. 2. However if all assignments of values are kept the same except the time to steal a backup, which is increased such that $\boxed{\text{steal backup}} \mapsto 35$, then the minimum attack time for Fig. 3(a) becomes $\max\{35, 9 + 2\} = 35$ and the minimum attack time for Fig. 2 becomes $\min\{\max\{25, 9 + 2\}, 9 + \max\{35, 2\}\} = 25$. Hence, in general, there is no correlation between the minimum attack times in Fig. 3(a) and Fig. 2. This example demonstrates that, for a specialisation that is unsound for an attribute domain, assignments of values in the attribute domain may give answers in one tree uncorrelated with answers for the other tree.

A similar story applies for the causal attack tree in Fig. 3(b), which is a sound specialisation of Fig. 1 for the "minimum attack time" attribute domain. According to the "minimum attack time" attribute domain, for any assignment of non-negative times to actions, the minimum attack time for Fig. 3(b) will always be greater than the minimum attack time for Fig. 1. In contrast, for the "minimum number of experts" attribute domain, assignments of values to basic actions in Fig. 3(b) and Fig. 1 give uncorrelated answers.

This work focusses on two specialisation preorders that are sound with respect to each of two attribute domains discussed so far. Other attribute domains are also sound with respect to these specialisation preorders. For example, when the "minimum number of experts" increases between Fig. 2 and Fig. 3(a), the "minimum amount of time to make all attacks possible" decreases. As a guideline, both specialisation preorders covered in this work are sound with respect to classes of attribute domains that involve non-deterministic choices. For instance, the "minimum attack time" attribute domain assumes that whenever there is a choice, only the branch taking the least time need be considered. Thus the two classes of attribute domain in this work do not cover all useful attribute domains. For example, attribute domains that involve probabilities, such as [9], have finer algebraic properties than the attribute domains covered in this work; hence demand finer specialisation preorders. For probabilistic attribute domains, answers may depend partially on multiple branches of a choice, rather than on non-deterministic choices. This highlights further the importance of understanding which semantics and techniques are relevant for a particular choice of attribute domain.

Manually verifying the soundness of each potential specialisation for each attribute domain is a laborious and error-prone task. Fortunately, we can devise procedures for deciding whether one tree is a sound specialisation of the other for a class of attribute domains. The decision procedure works by reducing specialisation to implication in a logical system. We will return to the above motivating examples in Section 5 where the logical system is introduced.

## 3. Denotational, Logical and Algebraic Semantics for Attack Trees

This section introduces attack trees [1] and connects three equivalent formulations of the multiset semantics [3] for attack trees. The multiset semantics is typically used to identify when two attack trees represent the same attack while being sensitive to the possibility that some resources, such as cash or one-time access codes, are not reusable once spent.

The three semantics presented can be classified as denotational, logical and algebraic. The denotational semantics is based on sets of multisets denoting normal forms for attack trees. The logical semantics is based on a fragment of

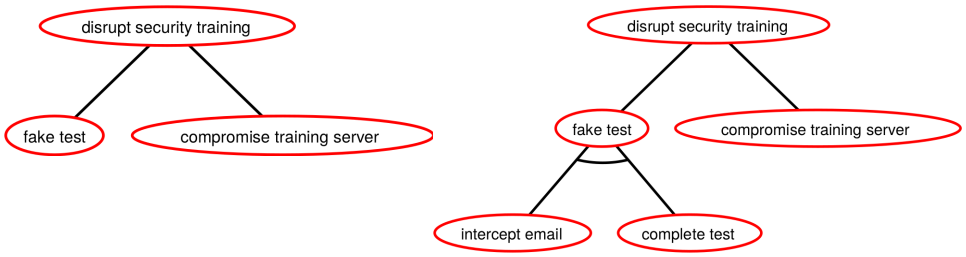



Figure 4: Example of evolving attack tree, indicating two refinement steps for goal “disrupt security training”.

linear logic [6], from which decision procedures can be extracted. The algebraic semantics highlights the commutative idempotent semiring structure of the multiset semantics, which can be used to check whether quantities assigned to attributes called *attribute domains* respect the multiset semantics.

Most of this section is background material introducing attack trees and linear logic. The novelty is a characterization of the multiset semantics through linear logic, which allows us to establish a new proof that the multiset semantics and algebraic semantics coincide. The connection with linear logic justifies generalisations made in later sections.

### *3.1. The Syntax of Attack Trees and Action Refinement*

The syntax for *attack trees* is defined by the following grammar.

$$\begin{array}{rcll} t & ::= & \boxed{\mathrm{a}} & \text{basic action} \\ & | & t \vartriangle t & \text{conjunctive refinement} \\ & | & t \triangledown t & \text{disjunctive refinement} \end{array}$$

The atoms $\boxed{\mathrm{a}}$ are drawn from a set of *basic actions*, which represent possible attacks by the proponent, such as “disrupt security training” or “set up back door admin account”. Conjunctive and disjunctive refinement are used to break down basic actions into sub-goals, to be more precise about attacks of the proponent. For conjunctive refinement $\vartriangle$, the proponent must perform both actions for the attack to be successful. For disjunctive refinement $\triangledown$, the proponent has two possible avenues of attack, but only one need be completed for the attack to be successful.

In graphical representations of attack trees, such as in Fig. 4, a node with an arc represents a conjunctive refinement, while a node with no arc represents disjunctive refinement. This graphical convention is used in supporting tools for attack trees [5].

The attack trees in Fig. 4, model part of a scenario where all employees must undergo online training in security practices of the company. Now suppose that an attacker aims to target an employee, such as a newly appointed VP, who is yet to undergo training. To increase the chances of the new VP not adhering sufficiently to best security practices, perhaps storing unencrypted company secrets, the attacker begins with the basic action: “disrupt security training”. This basic action can be disjunctively refined to two basic actions “fake test” and “compromise server for training”. In the former case, the attack can be refined further to two actions that the attacker can perform to fake the test, say “intercept training email” and “complete test on behalf of employee”. Further refinements are possible to indicate how the various goals are achieved. The two refinements steps are captured graphically in the trees in Fig. 4.

The tree on the right in Fig. 4 is a graphical representation of the following term.

$$\Big(\boxed{\text{intercept training email}} \vartriangle \boxed{\text{complete test}}\Big) \triangledown \boxed{\text{compromise server for training}}$$

Notice that the labels at the non-leaf nodes are not recorded in the syntax. The labels at these nodes simply recall the basic action that was refined, assisting the domain expert when they recall the intuition behind each refinement step. For example, “fake test”, a basic action in the tree on the left, is disjunctively refined in the tree on the right. Thus the

following mapping from basic actions to attack trees can also be recovered from the diagram in Fig. 4.

$$\boxed{\text{disrupt security training}} \mapsto \boxed{\text{fake test}} \triangledown \boxed{\text{compromise server for training}}$$
$$\boxed{\text{fake test}} \mapsto \boxed{\text{intercept training email}} \vartriangle \boxed{\text{complete test}}$$

We refer to such mappings as *action refinements*, to avoid confusion with *specialisation* which we introduce in this paper. All semantics in this work respect action refinement, in the sense that if two trees are related in the semantics then they have the same relation after applying the same action refinement to both trees.

### *3.2. An Established Multiset Semantics for Attack Trees*

In original work on the semantics of attack trees [3], attack trees are given a denotational semantics based on sets of multisets. This semantics is expressed using a distributive product as follows.

**Definition 1.** *Given two sets of multisets $V$ and $W$, define their* distributive product *as follows, where $\uplus$ is multiset union.*

$$V \bowtie W = \{v \uplus w \colon v \in V, w \in W\}$$

*The* multiset semantics *for attack trees is defined by the following function $[\![ \cdot ]\!]_{\mathcal{M}}$ that maps terms representing attack trees to sets of multisets.*

$$\left[\!\!\left[\boxed{a}\right]\!\!\right]_{\mathcal{M}} = \{\{\!|a|\!\}\} \qquad [\![t \triangledown u]\!]_{\mathcal{M}} = [\![t]\!]_{\mathcal{M}} \cup [\![u]\!]_{\mathcal{M}} \qquad [\![t \vartriangle u]\!]_{\mathcal{M}} = [\![t]\!]_{\mathcal{M}} \bowtie [\![u]\!]_{\mathcal{M}}$$

Notice that $\bowtie$ is associative and commutative, and distributes over set union, as follows.

$$U \bowtie (V \cup W) = (U \bowtie V) \cup (U \bowtie W)\,.$$

The above distributivity property allows disjunctive refinement to distribute over conjunctive refinement. Thereby the following attack trees are equivalent in the multiset semantics.

$$\left[\!\!\left[\left(\boxed{\text{fake test}} \triangledown \boxed{\text{compromise server for training}}\right) \vartriangle \boxed{\text{steal ID}}\right]\!\!\right]_{\mathcal{M}}$$
$$= \left[\!\!\left[\left(\boxed{\text{fake test}} \vartriangle \boxed{\text{steal ID}}\right) \triangledown \left(\boxed{\text{compromise server for training}} \vartriangle \boxed{\text{steal ID}}\right)\right]\!\!\right]_{\mathcal{M}}$$

Intuitively, two avenues of attack are presented. In both attacks, stealing the ID documents of the employee is necessary. Both attack trees represent the same attack scenario and have the same multiset-based denotation.

### *3.3. A Brief Overview of Multiplicative Additive Linear Logic*

A key observation enabling this paper is that implication in linear logic [6] can systematise the multiset semantics. Linear logic was designed to control the use of assumptions in proof search. In classical logic, since "$P$ and $P$" is equivalent to $P$, assumptions can be reused during proof search, formalised by the *contraction* rule. Also, not all assumptions need be used, formalised by the *weakening* rule.

$$\frac{\Gamma \vdash P, P, \Delta}{\Gamma \vdash P, \Delta} \qquad \frac{\Gamma, P, P \vdash \Delta}{\Gamma, P \vdash \Delta}\ \textit{contraction} \qquad\qquad \frac{\Gamma \vdash \Delta}{\Gamma, P \vdash \Delta} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash P, \Delta}\ \textit{weakening}$$

Rules here are expressed in the sequent calculus, where a *sequent* of the form $\Gamma \vdash \Delta$ consists of two multisets of propositions $\Gamma$ and $\Delta$. The standard interpretation of a sequent is that if we assume that the conjunction of every proposition left of the turnstile holds, the disjunction of the formulae right of the turnstile must hold.

In contrast to classical logic, linear logic *forbids* both contraction and weakening. A consequence is that *linear negation*, indicated by an over line, becomes a synthetic operator distinct from classical negation. This leads to distinct types of conjunction and disjunction in linear logic. *Multiplicative* conjunction, $P \otimes Q$ (called "times"), has De Morgan dual multiplicative disjunction, $P \parallel Q = \overline{\overline{P} \otimes \overline{Q}}$ (called "par"). *Additive* disjunction, $P \oplus Q$ (called "plus"), has a De Morgan dual additive conjunction, $P \,\&\, Q = \overline{\overline{P} \oplus \overline{Q}}$ (called "with").

$$\frac{}{a \vdash a}\ id \qquad \frac{\Gamma \vdash P, \Delta}{\Gamma, \overline{P} \vdash \Delta}\ neg \qquad \frac{\Gamma, P \vdash \Delta}{\Gamma \vdash \overline{P}, \Delta}\ neg$$

$$\frac{\Gamma_1 \vdash \Delta_1 \quad \Gamma_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\ mix \qquad \frac{\Gamma, P, Q \vdash \Delta}{\Gamma, P \otimes Q \vdash \Delta}\ \otimes_L \qquad \frac{\Gamma_1 \vdash P, \Delta_1 \quad \Gamma_2 \vdash Q, \Delta_2}{\Gamma_1, \Gamma_2 \vdash P \otimes Q, \Delta_1, \Delta_2}\ \otimes_R$$

$$\frac{\Gamma \vdash \Delta, P \quad P, \Gamma' \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\ cut \qquad \frac{\Gamma, P \vdash \Delta \quad \Gamma, Q \vdash \Delta}{\Gamma, P \oplus Q \vdash \Delta}\ \oplus_L \qquad \frac{\Gamma \vdash P_i, \Delta}{\Gamma \vdash P_1 \oplus P_2, \Delta}\ \oplus_R,\ i \in \{1, 2\}$$

Figure 5: Sequent calculus for MALL

The multiplicative-additive fragment of linear logic we consider is called MALL. Propositions in MALL are given by the following grammar, where $a$ ranges over basic actions.

$$P ::= a \mid \overline{P} \mid P \otimes P \mid P_1 \oplus P_2$$

The proof system for MALL is given in Figure 5, formalised in the sequent calculus. The identity axiom (*id*) states that a basic action can be derived from itself. The two negation rules express the duality of the propositions to the right of the turnstile (in a disjunctive context) and propositions to the left of the turnstile (in a conjunctive context). To understand the rules of MALL, contrast to the classical case with contraction and weakening. In the classical setting, the multiplicative and additive conjunctions $P \otimes Q$ and $\overline{\overline{P} \oplus \overline{Q}}$ coincide, as do the disjunctions $P \oplus Q$ and $\overline{\overline{P} \otimes \overline{Q}}$. However, since contraction and weakening are omitted from linear logic, the multiplicative and additive operators are distinguished. For more details on MALL, we recommend the paper on MAV [8], the logical system used later in this paper, which begins by introducing MALL. The *mix* rule is not essential for plain attack trees, but enables conservative extensions introduced in later sections.

A rule is said to be *admissible* whenever any proposition provable in a system using the rule can also be proven without using the rule. A central established result in linear logic, is that the cut rule is *admissible* for MALL. This established result is due to Girard [6].

**Theorem 2** (Girard 1987)**.** *The cut rule is admissible.*

Cut elimination is the corner stone of a proof calculus, with consequences including the transitivity of entailment and the consistency of the logical system. Furthermore, the sequent calculus has the property that the cut-free system has a finite search space, hence proof search is suited to implementing decision procedures. Left-introduction rules, indicated by the subscript $L$, are invertible, meaning that applying them eagerly does not affect provability; while their right-introduction rules, indicated by subscript $R$, are non-invertible meaning they trigger branching in proof search.

### *3.4. Correspondence between the Multiset and Linear Logic Semantics*

We show here an embedding of attack trees into positive linear logic propositions, i.e., propositions without negation, that preserves the multiset semantics. The embedding is defined as follows.

$$[\![\boxed{\text{a}}]\!]_{\mathcal{L}} = \text{a} \qquad [\![t \triangledown u]\!]_{\mathcal{L}} = [\![t]\!]_{\mathcal{L}} \oplus [\![u]\!]_{\mathcal{L}} \qquad [\![t \triangle u]\!]_{\mathcal{L}} = [\![t]\!]_{\mathcal{L}} \otimes [\![u]\!]_{\mathcal{L}}$$

For an example of how entailment in linear logic is useful, consider for example the attack tree for disrupting the security training of new employees in Fig. 4. The attack tree $\boxed{\text{intercept email}} \triangle \boxed{\text{complete test}}$ is a *specialisation* of $\left(\boxed{\text{intercept email}} \triangle \boxed{\text{complete test}}\right) \triangledown \boxed{\text{compromise server}}$, in the sense that the former has less choice between avenues of attack. In the former, the proponent does not attempt to compromise the server directly. In the multiset semantics, this specialisation is captured by the following subset inclusion.

$$\{\{\!|\text{intercept email}, \text{complete test}|\!\}\} \subseteq \{\{\!|\text{intercept email}, \text{complete test}|\!\}, \{\!|\text{compromise server}|\!\}\}$$

In the linear logic semantics the specialisation is captured by the following entailment.

$$\text{intercept email} \otimes \text{complete test} \vdash (\text{intercept email} \otimes \text{complete test}) \oplus \text{compromise server}$$

An advantage of entailment is a decision procedure can be extracted from the sequent calculus presentation of rules. A derivation for the above example is provided below.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\text{intercept email} \vdash \text{intercept email}}\ id \quad \dfrac{}{\text{complete test} \vdash \text{complete test}}\ id}{\text{intercept email}, \text{complete test} \vdash \text{intercept email} \otimes \text{complete test}}\ \otimes_R}{\text{intercept email}, \text{complete test} \vdash (\text{intercept email} \otimes \text{complete test}) \oplus \text{compromise server}}\ \oplus_R}{\text{intercept email} \otimes \text{complete test} \vdash (\text{intercept email} \otimes \text{complete test}) \oplus \text{compromise server}}\ \otimes_L$$

To establish the correspondence between the linear logic embedding and sets of mulisets, we show that positive propositions, i.e., propositions without negation using only $\oplus$ and $\otimes$, can be interpreted as a set of multisets of propositional atoms.

**Definition 3.** *The following function maps positive propositions to sets of multisets:*

$$m(a) = \{\!\{a\}\!\} \qquad m(P \oplus Q) = m(P) \cup m(Q) \qquad m(P \otimes Q) = m(P) \bowtie m(Q)$$

Using cut elimination, the next lemma can be easily proven with structural induction on $P$. Its proof exploits the existence of *focused proofs* [10] for valid linear logic formulae, where invertible rules are applied (reading the proof bottom up) before the non-invertible rules.

**Lemma 4.** *For any positive propositions $P_1, \ldots, P_n, P$, where $n \geq 1$, if $P_1, \ldots, P_n \vdash P$ is provable, then $m(P_1) \bowtie \cdots \bowtie m(P_n) \subseteq m(P)$.*

*Proof.* By induction on the height of the proof $\Pi$ of $P_1, \ldots, P_n \vdash P$, where invertible rules are applied eagerly. In the case where all $P_i$'s are atomic, we prove this lemma using Lemma **??**. Otherwise, the proof $\Pi$ must end with either $\oplus_L$ or $\otimes_L$. Assume without loss of generality that the last rule of $\Pi$ applies to $P_1$.

- Suppose $P_1 = P_1' \otimes P_1''$ and $\Pi$ is
$$\dfrac{\begin{matrix}\Pi' \\ P_1', P_1'', \ldots, P_n \vdash P\end{matrix}}{P_1' \otimes P_1'', \ldots, P_n \vdash P}\ \otimes_L.$$
By Definition 3 and the induction hypothesis, we have
$$m(P_1' \otimes P_1'') \bowtie \cdots \bowtie m(P_n) = m(P_1') \bowtie m(P_1'') \bowtie \cdots \bowtie m(P_n) \subseteq m(P).$$

- Suppose $P_1 = P_1' \oplus P_1''$ and $\Pi$ is
$$\dfrac{\begin{matrix}\Pi_1 & \Pi_2 \\ P_1', \ldots, P_n \vdash P & P_1'', \ldots, P_n \vdash P\end{matrix}}{P_1' \oplus P_1'', \ldots, P_n \vdash P}\ \oplus_L$$
By the induction hypotheses, we have:
$$m(P_1') \bowtie \cdots \bowtie m(P_n) \subseteq m(P) \qquad \text{and} \qquad m(P_1'') \bowtie \cdots \bowtie m(P_n) \subseteq m(P)$$
and therefore
$$\begin{aligned} m\left(P_1' \oplus P_1''\right) \bowtie \cdots \bowtie m(P_n) = \ & (m\left(P_1'\right) \cup m\left(P_1''\right)) \bowtie \cdots \bowtie m(P_n) \\ = \ & m\left(P_1'\right) \bowtie \cdots \bowtie m(P_n) \cup m\left(P_1''\right) \bowtie \cdots \bowtie m(P_n) \subseteq m(P). \end{aligned}$$

□

The next lemma expresses the converse of Lemma 4.

**Lemma 5.** *If $\{\!|a_1, \ldots, a_n|\!\} \in m(P)$ for some positive proposition $P$ then $a_1, \ldots, a_n \vdash P$ is provable.*

*Proof.* By structural induction on $P$.

- If $P = a$ for some atomic proposition $a$, then $\{\!|a_1, \ldots, a_n|\!\} = \{\!|a|\!\}$. In this case, the proof can be constructed by an application of the *id* rule.
- Suppose $P = P_1 \otimes P_2$. Then $m(P) = m(P_1) \bowtie m(P_2)$. In this case, $\{\!|a_1, \ldots, a_n|\!\}$ can be partitioned into $\{\!|a_{i_1}, \ldots, a_{i_k}|\!\} \in m(P_1)$ and $\{\!|a_{j_1}, \ldots, a_{j_m}|\!\} \in m(P_2)$. By the induction hypotheses, we have that $a_{i_1}, \ldots, a_{i_k} \vdash m(P_1)$ and $a_{j_1}, \ldots, a_{j_k} \vdash m(P_1)$ are provable, and

$$\frac{\begin{array}{cc}\vdots & \vdots \\ a_{i_1}, \ldots, a_{i_m} \vdash P_1 & a_{i_1}, \ldots, a_{i_m} \vdash P_1\end{array}}{a_1, \ldots, a_n \vdash P_1 \otimes P_2} \otimes_R$$

- Suppose $P = P_1 \oplus P_2$. Then $m(P) = m(P_1) \cup m(P_2)$. If $\{\!|a_1, \ldots, a_n|\!\} \in m(P_1)$ then we construct the proof as follows:

$$\frac{\begin{array}{c}\vdots \\ a_1, \ldots, a_n \vdash P_1\end{array}}{a_1, \ldots, a_n \vdash P_1 \oplus P_2} \oplus_R$$

  where the premise is provable by the induction hypothesis. The other case where $\{\!|a_1, \ldots, a_n|\!\} \in m(P_2)$ can be proved similarly.

□

**Lemma 6.** *For positive propositions, if $m(P_1) \bowtie \cdots \bowtie m(P_n) \subseteq m(P)$ then $P_1, \ldots, P_n \vdash P$.*

*Proof.* Given a proposition $Q$, let $|Q|$ denote its size, i.e., the number of symbols occurring in $Q$. This lemma is proved by induction on the multiset of proposition sizes: $\{\!|\, |P_1|, \ldots, |P_n| \,|\!\}$.

In the case where every $P_i$ is an atomic proposition, this lemma follows immediately from Lemma 5. If at least one $P_i$ is not atomic, then we proceed by induction as follows (assume w.l.o.g. $i = 1$):

- $P_1 = P_1' \otimes P_1''$. In this case, we have

$$m(P_1) \bowtie \cdots \bowtie m(P_n) = m(P_1') \bowtie m(P_1'') \bowtie \cdots \bowtie m(P_n) \subseteq m(P)\,.$$

  The proof is then constructed as follows:

$$\frac{\begin{array}{c}\vdots \\ P_1', P_1'', \ldots, P_n \vdash P\end{array}}{P_1' \otimes P_1'', \ldots, P_n \vdash P} \otimes_L$$

  where the premise is provable by the induction hypothesis.

- $P_1 = P_1' \oplus P_1''$. In this case, we have

$$m(P_1) \bowtie \cdots \bowtie m(P_n) = m(P_1') \bowtie \cdots \bowtie m(P_n) \cup m(P_1'') \bowtie \cdots \bowtie m(P_n) \subseteq m(P)\,.$$

  That is, we have $m\left(P_1'\right) \bowtie \cdots \bowtie m(P_n) \subseteq m(P)$ and $m\left(P_1''\right) \bowtie \cdots \bowtie m(P_n) \subseteq m(P)$. The proof in this case is constructed as follows:

$$\frac{\begin{array}{cc}\vdots & \vdots \\ P_1', \ldots, P_n \vdash P & P_1'', \ldots, P_n \vdash P\end{array}}{P_1' \oplus P_1'', \ldots, P_n \vdash P}$$

  where the premises are provable by applying the induction hypothesis.

| | |
|---|---|
| $\triangle$ defines a semigroup: | $(t \triangle u) \triangle v \cong t \triangle (u \triangle v)$ |
| $\nabla$ defines a commutative semigroup: | $(t \nabla u) \nabla v \cong t \nabla (u \nabla v) \qquad t \nabla u \cong u \nabla t$ |
| $\triangle$ distributes over $\nabla$: | $(t \nabla u) \triangle v \cong (t \triangle v) \nabla (u \triangle v)$ |
| $\triangle$ is commutative and $\nabla$ is idempotent: | $t \triangle u \cong u \triangle t \qquad t \nabla t \cong t$ |

Figure 6: The identities of a commutative idempotent semiring ($\cong$) closed under congruence.

□

Lemmas 4 and 6 establish that, for positive propositions, the multiset semantics coincides with entailment in linear logic, as summarised by the following theorem.

**Theorem 7.** *Given attack trees t and u, $[\![t]\!]_{\mathcal{M}} \subseteq [\![u]\!]_{\mathcal{M}}$ if and only if $[\![t]\!]_{\mathcal{L}} \vdash [\![u]\!]_{\mathcal{L}}$.*

*Proof.* Firstly, establish $m([\![t]\!]_{\mathcal{L}}) = [\![t]\!]_{\mathcal{M}}$ by structural induction over the attack tree. Thereby, for any two attack trees $t$ and $u$, $[\![t]\!]_{\mathcal{M}} \subseteq [\![u]\!]_{\mathcal{M}}$ holds if and only if $m([\![t]\!]_{\mathcal{L}}) \subseteq m([\![u]\!]_{\mathcal{L}})$, where $m(\,\cdot\,)$ is as in Definition 3. By Lemma 4 and Lemma 6, $m([\![t]\!]_{\mathcal{L}}) \subseteq m([\![u]\!]_{\mathcal{L}})$ if and only if $[\![t]\!]_{\mathcal{L}} \vdash [\![u]\!]_{\mathcal{L}}$. □

**Corollary 8.** *Given attack trees t and u, $[\![t]\!]_{\mathcal{M}} = [\![u]\!]_{\mathcal{M}}$ if and only if $[\![t]\!]_{\mathcal{L}} \vdash [\![u]\!]_{\mathcal{L}}$ and $[\![u]\!]_{\mathcal{L}} \vdash [\![t]\!]_{\mathcal{L}}$.*

### *3.5. An Algebraic Semantics and Attribute Domains*

In the previous sections, we directly established the equivalence of a semantics for attack trees based on multisets and a semantics based on positive propositions in linear logic. We now present an algebraic semantics that also coincides with the multiset semantics for attack trees.

The algebraic characterisation is in terms of a particular class of semirings. A semiring is two semigroups with (associative) binary operators $\nabla$ and $\triangle$ such that $\nabla$ is commutative and distributes over $\triangle$. When additionally $\nabla$ is idempotent and $\triangle$ is commutative, we call such a structure a commutative idempotent semiring. Axioms for commutative idempotent semirings are presented in Fig 6. A preorder over semirings is defined such that $t \preceq u$ if and only if $t \nabla u \cong u$.

The fact that (the positive fragment of) MALL conservatively extends a commutative idempotent semiring is folklore. The proof relies on Theorem 37.

**Proposition 9.** *For attack trees t and u, $[\![t]\!]_{\mathcal{L}} \vdash [\![u]\!]_{\mathcal{L}}$ if and only if $t \preceq u$.*

**Corollary 10.** *For attack trees t and u, $[\![t]\!]_{\mathcal{L}} \vdash [\![u]\!]_{\mathcal{L}}$ and $[\![u]\!]_{\mathcal{L}} \vdash [\![t]\!]_{\mathcal{L}}$ if and only if $t \cong u$.*

*Proof.* For each equation $v \cong w$ in Fig. 6, we have $[\![v]\!]_{\mathcal{L}} \vdash [\![w]\!]_{\mathcal{L}}$. Since the entailment relation $\vdash$ is a pre-congruence, and is both reflexive and transitive, it follows that for every equation $v \cong w$ derivable from the equations in Fig. 6 we have $[\![v]\!]_{\mathcal{L}} \vdash [\![w]\!]_{\mathcal{L}}$. Now assume that $t \nabla u \cong u$ hence $[\![t]\!]_{\mathcal{L}} \oplus [\![u]\!]_{\mathcal{L}} \vdash [\![u]\!]_{\mathcal{L}}$ is provable, and observe that $[\![t]\!]_{\mathcal{L}} \vdash [\![t]\!]_{\mathcal{L}} \oplus [\![u]\!]_{\mathcal{L}}$ is directly provable; hence by an application of the cut rule, we have $[\![t]\!]_{\mathcal{L}} \vdash [\![u]\!]_{\mathcal{L}}$.

For the converse direction, proceed by induction over the structure of a proof of $[\![t]\!]_{\mathcal{L}} \vdash [\![u]\!]_{\mathcal{L}}$ to establish that $u \cong t \nabla u$ holds in the algebra, where $t$ and $u$ are attack trees. Most cases are immediate, except the case for $\otimes_R$.

Consider the case where, $[\![t_1]\!]_{\mathcal{L}}, \ldots [\![t_m]\!]_{\mathcal{L}}, [\![u_1]\!]_{\mathcal{L}}, \ldots [\![u_n]\!]_{\mathcal{L}} \vdash [\![t]\!]_{\mathcal{L}} \otimes [\![u]\!]_{\mathcal{L}}$ follows, by rule $\otimes_R$, from $[\![t_1]\!]_{\mathcal{L}}, \ldots [\![t_m]\!]_{\mathcal{L}} \vdash [\![t]\!]_{\mathcal{L}}$ and $[\![u_1]\!]_{\mathcal{L}}, \ldots [\![u_n]\!]_{\mathcal{L}} \vdash [\![u]\!]_{\mathcal{L}}$. By assuming the induction hypothesis, $(t_1 \triangle \ldots \triangle t_m) \nabla t \cong t$ and $(u_1 \triangle \ldots \triangle u_n) \nabla u \cong u$, from which the following equality can be derived, as required.

| | | |
|---|---|---|
| $t \triangle u \cong$ | $((t_1 \triangle \ldots \triangle t_m) \nabla t) \triangle u$ | by induction |
| $\cong$ | $(t_1 \triangle \ldots \triangle t_m \triangle u) \nabla (t \triangle u)$ | distributivity |
| $\cong$ | $(t_1 \triangle \ldots \triangle t_m \triangle ((u_1 \triangle \ldots \triangle u_n) \nabla u)) \nabla (t \triangle u)$ | by induction |
| $\cong$ | $(t_1 \triangle \ldots \triangle t_m \triangle u_1 \triangle \ldots \triangle u_n) \nabla (t_1 \triangle \ldots \triangle t_m \triangle u) \nabla (t \triangle u)$ | distributivity |
| $\cong$ | $(t_1 \triangle \ldots \triangle t_m \triangle u_1 \triangle \ldots \triangle u_n) \nabla (((t_1 \triangle \ldots \triangle t_m) \nabla t) \triangle u)$ | distributivity |
| $\cong$ | $(t_1 \triangle \ldots \triangle t_m \triangle u_1 \triangle \ldots \triangle u_n) \nabla (t \triangle u)$ | by induction |

Hence by induction on the size of the proof if $[\![t]\!]_{\mathcal{L}} \vdash [\![u]\!]_{\mathcal{L}}$ then $t \preceq u$. □

Commutative idempotent semirings are ubiquitous structures in computer science arising as part of the algebraic theory of various metrics including min-plus, max-plus semirings, and the distributive lattices of classical propositional logic. Such semirings can be used to specify *attribute domains* over attack trees, where an attribute domain is an assignment of values in a domain to basic actions and functions over the domain to refinement operators.

**Definition 11.** *Suppose $\mathcal{D} = (D, f, g)$ is an attribute domain with two binary operators over $D$. Assume also that $\vartheta$ is a valuation mapping basic actions to $D$. The interpretation function $I^{\mathcal{D}}_{\vartheta}(\,\cdot\,)$ from attack trees to $D$ is defined as follows.*

$$I^{\mathcal{D}}_{\vartheta}\big(\boxed{a}\big) = \vartheta(\boxed{a}) \qquad I^{\mathcal{D}}_{\vartheta}(t \triangledown u) = f(I^{\mathcal{D}}_{\vartheta}(t)\,, I^{\mathcal{D}}_{\vartheta}(u)) \qquad I^{\mathcal{D}}_{\vartheta}(t \triangle u) = g(I^{\mathcal{D}}_{\vartheta}(t)\,, I^{\mathcal{D}}_{\vartheta}(u))$$

*An attribute domain $\mathcal{D}$ ordered by $\leq_{\mathcal{D}}$ is sound with respect to specialisation order $\preceq$, whenever for all attack trees $t$ and $u$ and valuations $\vartheta$, if $t \preceq u$ then $I^{\mathcal{D}}_{\vartheta}(t) \leq_{\mathcal{D}} I^{\mathcal{D}}_{\vartheta}(u)$.*

Although often implicitly defined, attribute domains occur frequently in the attack trees literature (for an extensive literature overview see [11]). As an example, consider the domain $\mathcal{B} = (\mathbb{R}, \min, \max)$, where $\vartheta$ is a valuation assigning the estimated time to perform a basic action in an attack. The interpretation function $I^{\mathcal{B}}_{\vartheta}(\,\cdot\,)$ calculates the "minimum time to perform an attack", assuming actions in a conjunctive refinement can be performed concurrently. The soundness of specialisation, defined using entailment in linear logic, with respect to this "minimum time to perform an attack" attribute domain is established as follows.

**Proposition 12.** *For all attack trees $t$ and $u$ and all $\vartheta$, if $[\![t]\!]_{\mathcal{L}} \vdash [\![u]\!]_{\mathcal{L}}$ then $I^{\mathcal{B}}_{\vartheta}(t) \geq I^{\mathcal{B}}_{\vartheta}(u)$.*

To see why the direction of implication is opposite to the order over real numbers in the above soundness property, observe the following. Since $t \preceq u$ only if $t \triangledown u \cong u$ and since $\triangledown$ is interpreted by min, in this domain if $t \preceq u$, then $\min\big(I^{\mathcal{B}}_{\vartheta}(t)\,, I^{\mathcal{B}}_{\vartheta}(u)\big) = I^{\mathcal{B}}_{\vartheta}(u)$, which holds if and only if $I^{\mathcal{B}}_{\vartheta}(t) \geq I^{\mathcal{B}}_{\vartheta}(u)$.

Attribute domains such as $(\mathbb{R}, \min, +)$ representing the "minimal cost of an attack" do not depend on whether actions in a conjunctive refinement are performed concurrently. In contrast, notice that the "minimum time to perform an attack" attribute domain $(\mathbb{R}, \min, \max)$ depends on sub-goals in a conjunctive refinement being performed concurrently and independently. However, it is clearly not always the case that sub-goals can be performed concurrently and independently. For example, in Fig. 4 it may be that "intercept email" must be achieved **before** "complete test". Attribute domains sensitive to causal dependencies between actions demand the semantics in the next section.

## 4. An Extension of Attack Trees Respecting Causality

Attack trees have various extensions for elaborating on the attack scenario profiled. The extension we consider in this section is for attack trees with a specialisation of conjunctive refinement called *sequential refinement*. Sequential refinement explicitly indicates where there are causal dependencies between actions of an attacker, thus actions cannot be performed concurrently. For example, an attacker may hijack a TCP session **before** attempting to access a company database. Due to the causal dependencies between events that must be performed sequentially, we call such attack trees with sequential refinement *causal attack trees*.

As highlighted at the end of Section 3 some quantitative questions, such as those involving timing or conflicting resources, are sensitive to whether or not there are causal dependencies between actions. A finer semantics than the multiset semantics is required to distinguish sequential refinement from conjunctive refinement, and to recover soundness with respect to such attribute domains. Semantics proposed in related work [2] assume that sequential and conjunctive refinement are unrelated, in the sense that there is no property like distributivity defining the interaction between the two operators. We discover in this section, when sequential refinement specialises conjunctive refinement, more than one semantics may be appropriate depending on the relevant class of attribute domain.

The grammar of attack trees is extended with sequential refinement as follows.

$$\begin{array}{rcll} t ::= & \boxed{\mathrm{a}} & & \text{basic action} \\ \mid & t \triangle t & & \text{conjunctive refinement} \\ \mid & t \triangledown t & & \text{disjunctive refinement} \\ \mid & t \cdot t & & \text{sequential refinement} \end{array}$$

Sequential refinement is sometimes called "sequential and" [2]. However we avoid this terminology, since in later sections we reveal that the operator can be interpreted by a self-dual operator, hence is neither strictly conjunctive nor strictly disjunctive in logical nature.

In this section, we introduce two semantics for causal attack trees based on sets of graphs closed under certain graph homomorphisms.

### *4.1. Preliminaries on Series-Parallel Graphs*

We proceed with the preliminaries for our proposed semantics for causal attack trees. Causal attack trees are denoted by particular sets of graphs, called labelled series-parallel graphs [12]. Series-parallel graphs are DAGs constructed from single vertices, labelled with basic actions, using graph constructors composing graphs in series and parallel. For parallel composition, no new edges are added between two disjoint graphs; and for sequential composition, new directed edges are added from every node in one graph to every node in the other graph, modelling causal dependencies.

A labelled digraph is defined as follows. Labels range over basic actions in this work.

**Definition 13.** *A labelled digraph is a triple $(V, E, \mu)$ where $V$ is a set of vertices $E \subseteq V \times V$ is a set of edges and $\mu$ is a function from $V$ to a set of labels.*

We require the definition of disjoint union and sequential composition of labelled digraphs. The sequential composition is the least digraph extending the disjoint union such that there is an edge from every vertex in the left graph to every vertex in the right graph.

**Definition 14.** *If $G = (V_0, E_0, \mu_0)$ and $H = (V_1, E_1, \mu_1)$ are labelled digraphs then their disjoint union $G \uplus H$ is the labelled digraph $(V, E, \mu)$, where $V = V_0 \times \{0\} \cup V_1 \times \{1\}$, $E \subseteq V \times V$ is the least set such that if $(x, y) \in E_0$ then $((x, 0), (y, 0)) \in E$ and if $(x, y) \in E_1$ then $((x, 1), (y, 1)) \in E$, and $\mu$ is such that for all $x \in V_0$, $\mu((x, 0)) = \mu_0(x)$ and for all $x \in V_1$, $\mu((x, 1)) = \mu_1(x)$. The sequential composition of $G$ and $H$ is defined by the labelled digraph $(V, E \cup ((V_0 \times \{0\}) \times (V_1 \times \{1\})), \mu)$ and denoted $GH$.*

A series-parallel graph is a labelled digraph that can be constructed from smaller graphs by composing them in series and parallel, where parallel composition is disjoint union (hence there are no new edges between the graphs) and sequential composition is defined as above.

**Definition 15.** *A labelled series-parallel graph is a digraph $(V, E, \mu)$ that can be constructed as follows.*

- *The empty digraph $(\emptyset, \emptyset, \emptyset)$ is a series-parallel graph, denoted by $\epsilon$.*
- *The singleton digraph with label $\boxed{a}$ denoted $(\{x\}, \emptyset, \mu)$, where $\mu\colon x \mapsto \boxed{a}$ for some vertex $x$ is a series-parallel graph. This singleton graph is denoted by $G_a$.*
- *If $G = (V_0, E_0, \mu_0)$ and $H = (V_1, E_1, \mu_1)$ are series-parallel graphs, then their disjoint union $G \uplus H$ and sequential composition $GH$ are series-parallel graphs.*

By construction, a series-parallel graph is acyclic and transitively closed.

The semantics for causal attack trees proposed by Jhawar et al. [2] is based on sets of labelled series-parallel graphs, inspired by an axiomatisation of pomsets due to Gischer [13]. Established definitions [2] are in terms of graphs with labelled edges rather than vertices; which can always be transformed to series-parallel graphs of the above form using an algorithm due to Valdes et al. [12]. Gischer [13] also provides a more "permissive" semantics for pomsets, which can be captured by using morphisms called *smoothing homomorphisms* [14]. A *smoothing homomorphism* is a label-preserving bijection between the vertices of two graphs that preserves the set of edges. Thereby the structure of the graph is preserved but new directed edges may be added to the target graph.

**Definition 16.** *Consider two labelled digraphs $G = (V_0, E_0, \mu_0)$ and $H = (V_1, E_1, \mu_1)$. A smoothing homomorphism $f$ from $G$ to $H$ is a bijection between $V_0$ and $V_1$ such that for all vertices $x, y \in V_0$, $\mu_0(x) = \mu_1(f(x))$, and if $(x, y) \in E_0$ then $(f(x), f(y)) \in E_1$.*

$$
\begin{array}{rl}
\text{Sequential refinement defines a semigroup:} & t \cdot (u \cdot v) \cong (t \cdot u) \cdot v \\
\text{Sequential refinement distributes over } \triangledown\text{:} & t \cdot (u \triangledown v) \cong (t \cdot u) \triangledown (t \cdot v) \\
& (t \triangledown u) \cdot v \cong (t \cdot v) \triangledown (u \cdot v)
\end{array}
$$

Figure 7: Equational axioms for sequential refinement.

We employ denotations based on sets of series-parallel graphs that are closed with respect to smoothing homomorphisms, called ideals and filters. Ideals are defined using smoothing homomorphisms such that, if a graph is in an ideal, then all graphs with the same structure as a graph and possibly **more** edges are also in the ideal.

**Definition 17.** *An ideal is a set of labelled digraphs $S$ such that if $H \in S$ and $G$ is a labelled digraph such that there exists a smoothing homomorphism from $H$ to $G$, then $G \in S$. The ideal closure of a set of labelled digraphs $T$ is the least ideal $S$ such that $T \subseteq S$ denoted $\iota(T)$.*

For filters the direction of smoothing homomorphisms are reversed compared to ideals. If a graph is in a filter then all graphs with the same structure as a graph with **fewer** edges are also in the filter, as defined using smoothing homomorphism in the following.

**Definition 18.** *A filter is a set of labelled digraphs $S$ such that if $H \in S$ and $G$ is a labelled digraph such that there exists a smoothing homomorphism from $G$ to $H$, then $G \in S$. The filter closure of a set of labelled digraphs $T$ is the least filter $S$ such that $T \subseteq S$ denoted $\phi(T)$.*

Isomorphisms, defined next, characterise when two graphs have the same structure. By identifying graphs with their equivalence classes the identity of vertices are abstracted away. We also extend disjoint union and parallel composition point-wise to sets of graphs.

**Definition 19.** *Two labelled digraphs are isomorphic whenever there is a bijection $f$ between vertices such that both $f$ and its inverse $f^{-1}$ define smoothing homomorphisms. For $G_a$, the equivalence class of all isomorphic graphs is denoted $[G_a]$.*

*The distributive product of two sets of series-parallel graphs $S \bowtie T$, is defined as the point-wise disjoint union $\{G \uplus H \colon G \in S, H \in T\}$. Point-wise sequential composition of two sets of series-parallel graphs $S \triangleleft T$ is defined as $\{GH \colon G \in S, H \in T\}$.*

The distributive product of filters, naturally extending the distributive product of multisets, is a filter. In contrast, the distributive product of ideals is not an ideal. The following proposition clarifies which constructions preserve ideals and filters.

**Proposition 20.** *The equivalence class $[G_a]$ of all singleton graphs with label $a$ is a filter and an ideal. If $S$ and $T$ are filters then the distributive product $S \bowtie T$ is a filter and $S \cup T$ is a filter. If $S$ and $T$ are ideals then the point-wise sequential composition $S \triangleleft T$ is an ideal and $S \cup T$ is an ideal.*

### *4.2. Limitations of an Intermediate Semantics regarding Specialisation*

The definitions in the previous section are sufficient to introduce several graph-based semantics. An *intermediate semantics* for causal attack trees, established in related work [2], is defined as follows.

**Definition 21** (intermediate semantics)**.** *The intermediate semantics is defined by the following embedding from causal attack trees to sets of labelled series-parallel graphs.*

$$
\begin{array}{lcllcl}
[\![ \boxed{a} ]\!]_{\mathcal{M}} & = & [G_a] & [\![t \triangledown u]\!]_{\mathcal{M}} & = & [\![t]\!]_{\mathcal{M}} \cup [\![u]\!]_{\mathcal{M}} \\
[\![t \vartriangle u]\!]_{\mathcal{M}} & = & [\![t]\!]_{\mathcal{M}} \bowtie [\![u]\!]_{\mathcal{M}} & [\![t \cdot u]\!]_{\mathcal{M}} & = & [\![t]\!]_{\mathcal{M}} \triangleleft [\![u]\!]_{\mathcal{M}}
\end{array}
$$

Omitting sequential refinement, the above semantics coincides with the multiset semantics for attack trees in Def. 1. This is because edges are used for sequential refinement only and equivalence classes of labelled digraphs with no edges coincide with multisets. Thus, by Proposition 12, conjunctive and disjunctive refinement form an idempotent commutative semiring as defined by $\cong$ for attack trees in Fig. 6. For sequential refinement, the equations in Fig. 7 hold, making sequential refinement a semi-group that distributes over disjunctive refinement. Indeed this algebraic characterisation is sound and complete for the intermediate semantics, as follows.

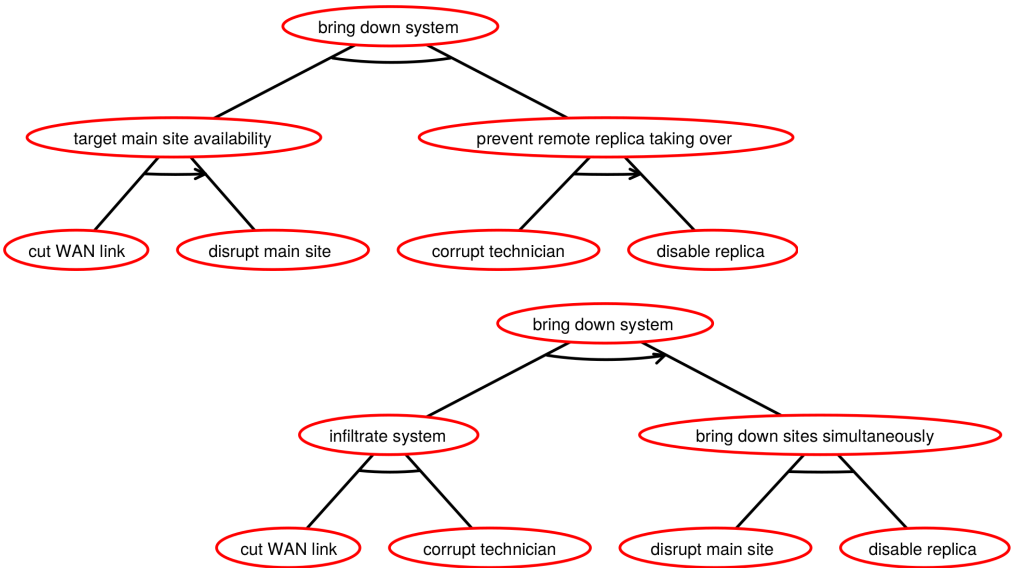


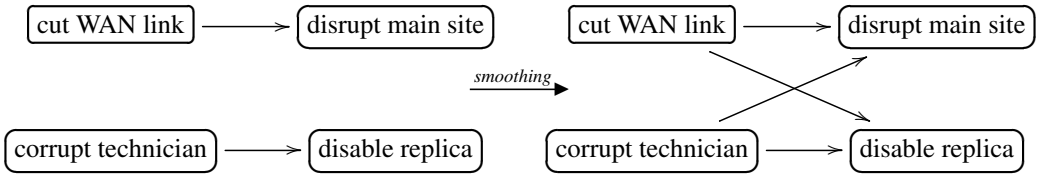

Figure 8: Causal attack trees unrelated by the intermediate semantics, but related for ideals ($\succeq_{\mathcal{I}}$) and filters ($\preceq_{\mathcal{F}}$).

**Theorem 22** (Jhawar et al. 2015)**.** *For causal attack trees $t$ and $u$, $[\![t]\!]_{\mathcal{M}} = [\![u]\!]_{\mathcal{M}}$ if and only if $t = u$ by the equations in Fig. 6 and 7.*

A limitation of the above intermediate semantics is that it does not accommodate relationships between causal attack trees that are not equivalent, as required for a *specialisation* preorder.

For example, we argue that the causal attack trees in Fig. 8 should be related by a specialisation preorder. Notice the exchange of the conjunctive and sequential refinements. The second tree in Fig. 8 represents a specialisation of the first tree. For the first tree, the attacker must "*cut the WAN link*" **before** "*disrupting the main site*" and also should "*corrupt a technician*" **before** "*disabling a replica*". In contrast, the second tree in Fig. 8 implies that both "*cutting the WAN link*" and "*corrupting a technician*" are both prerequisites for "*disrupting the main site*" and also for "*disabling a replica*". Thus the second tree of Fig. 8 represents strictly more causal dependencies than the first tree, and thereby greater synchronisation between sub-goals.

Smoothing homomorphisms, Def. 16, are used to accommodate specialisations of the form of Fig. 8. There is a smoothing homomorphism from the series-parallel graph denoting the first tree to the series-parallel graph denoting the second tree. This smoothing homomorphism, adding two directed edges, is represented below.




The series-parallel graphs above are represented up to graph isomorphism classes leaving only labels.

By definition, ideals and filters are closed under smoothing homomorphisms. Consequently, if we close the intermediate semantics in Def. 21 for filters, then the two example trees in Fig. 8 are ordered by subset inclusion. In the following, $\phi$ is the filter closure operator from Def. 18.

$$\phi\Big(\Big[\!\Big[\Big(\boxed{\text{cut WAN link}}\cdot\boxed{\text{disrupt main site}}\Big)\triangle\Big(\boxed{\text{corrupt technician}}\cdot\boxed{\text{disable replica}}\Big)\Big]\!\Big]_{\mathcal{M}}\Big)$$
$$\subseteq \phi\Big(\Big[\!\Big[\Big(\boxed{\text{cut WAN link}}\triangle\boxed{\text{corrupt technician}}\Big)\cdot\Big(\boxed{\text{disrupt main site}}\triangle\boxed{\text{disable replica}}\Big)\Big]\!\Big]_{\mathcal{M}}\Big)$$

If we close the intermediate semantics in Def. 21 for ideals, then the reverse subset inclusion holds. In the following, $\iota$ is the ideal closure operator from Def. 17.

$$\iota\left(\left[\!\left[\left(\boxed{\text{cut WAN link}} \vartriangle \boxed{\text{corrupt technician}}\right) \cdot \left(\boxed{\text{disrupt main site}} \vartriangle \boxed{\text{disable replica}}\right)\right]\!\right]_{\mathcal{M}}\right)$$
$$\subseteq \iota\left(\left[\!\left[\left(\boxed{\text{cut WAN link}} \cdot \boxed{\text{disrupt main site}}\right) \vartriangle \left(\boxed{\text{corrupt technician}} \cdot \boxed{\text{disable replica}}\right)\right]\!\right]_{\mathcal{M}}\right)$$

*4.3. Two Semantics for Specialising Causal Attack Trees*

Notice that, by Proposition 20, point-wise sequential composition preserves ideals and the distributive product preserves filters. However, the point-wise sequential composition of filters is not necessarily a filter. Similarly, the distributive product of ideals is not necessarily an ideal. Therefore, to obtain a filter extending the point-wise sequential composition of filters, the filter closure function $\phi$ must be applied to close the set of graphs. Similarly, to obtain an ideal extending the distributive product of ideals, the ideal closure function $\iota$ is applied.

The above observations lead us to the following two denotational semantics for causal attack trees.

**Definition 23** (ideal semantics)**.** *The ideal semantics is defined by the following mapping from causal attack trees to ideals.*

$$\left[\!\left[\boxed{a}\right]\!\right]_{\mathcal{I}} = [G_a] \qquad [\![t \triangledown u]\!]_{\mathcal{I}} = [\![t]\!]_{\mathcal{I}} \cup [\![u]\!]_{\mathcal{I}}$$
$$[\![t \vartriangle u]\!]_{\mathcal{I}} = \iota([\![t]\!]_{\mathcal{I}} \bowtie [\![u]\!]_{\mathcal{I}}) \qquad [\![t \cdot u]\!]_{\mathcal{I}} = [\![t]\!]_{\mathcal{I}} \triangleleft [\![u]\!]_{\mathcal{I}}$$

**Definition 24** (filter semantics)**.** *The filter semantics is defined by the following mapping from causal attack trees to filters.*

$$\left[\!\left[\boxed{a}\right]\!\right]_{\mathcal{F}} = [G_a] \qquad [\![t \triangledown u]\!]_{\mathcal{F}} = [\![t]\!]_{\mathcal{F}} \cup [\![u]\!]_{\mathcal{F}}$$
$$[\![t \vartriangle u]\!]_{\mathcal{F}} = [\![t]\!]_{\mathcal{F}} \bowtie [\![u]\!]_{\mathcal{F}} \qquad [\![t \cdot u]\!]_{\mathcal{F}} = \phi([\![t]\!]_{\mathcal{F}} \triangleleft [\![u]\!]_{\mathcal{F}})$$

The ideal semantics is an established process model [13], whereas the filter semantics is new. In the filter semantics, the reversing of the direction of a smoothing homomorphism compared to the ideal semantics affects how $\vartriangle$ and $\cdot$ exchange with each other, as in Fig. 8.

The difference between the ideal and filter semantics is clarified by considering a sound and complete axiomatisation for each semantics. To accommodate the specialisations due to smoothing homomorphisms, define $\preceq_{\mathcal{F}}$ and $\preceq_{\mathcal{I}}$ to be the least precongruences satisfying the equations in Fig. 6 and Fig. 7, and the inequations in Fig. 9. Congruences over filters and ideals are related to respective precongruences such that $t \preceq_{\mathcal{F}} u$ whenever $t \triangledown u \cong_{\mathcal{F}} u$ and $t \preceq_{\mathcal{I}} u$ whenever $t \triangledown u \cong_{\mathcal{I}} u$. Notice that in Fig. 9 the filter and ideal semantics differ only in the direction in which sequential and conjunctive refinement exchange with each other.

The significance of the above mentioned difference between the ideal and filter semantics, was discussed in the motivation, Section 2. For some attribute domains sequential and conjunctive refinement exchange in one direction, but for others the direction is opposite. This is discussed more formally in the next section, when defining attribute domains for causal attack trees.

Note the rules defining how conjunctive and sequential refinement are related are instances of the first rule, i.e. $(t \cdot v) \vartriangle (u \cdot w) \preceq_{\mathcal{F}} (t \vartriangle u) \cdot (v \vartriangle w)$ for the filter semantics, without certain sub-goals. It is semantically neat to add an attack tree that is the unit for both conjunctive and sequential refinement, denoted by $\{\epsilon\}$ and representing the sub-goal that automatically succeeds. However, unit attack trees are avoided in the literature, hence are also avoided in the current work.

For ideals, the axiomatisation given by $\cong_{\mathcal{I}}$ or $\preceq_{\mathcal{I}}$, using Fig. 6, Fig. 7 and Fig. 9, is sound and complete with respect to the ideal semantics. This result was established for pomsets [13], and is formally stated below for causal attack trees.

**Theorem 25** (Gischer 1988)**.** *For causal attack trees t and u, $[\![t]\!]_{\mathcal{I}} \subseteq [\![u]\!]_{\mathcal{I}}$ if and only if $t \preceq_{\mathcal{I}} u$.*

The proof of the corresponding soundness and completeness result for the filter semantics, with respect to $\cong_{\mathcal{I}}$ or $\preceq_{\mathcal{I}}$, is similar to the soundness and completeness result for ideals. The difference is that the switch of direction for smoothing homomorphisms in the definition of filters requires that some cases must be revisited.

**Theorem 26** (Soundness)**.** *For causal attack trees t and u, if $t \preceq_{\mathcal{F}} u$ then $[\![t]\!]_{\mathcal{F}} \subseteq [\![u]\!]_{\mathcal{F}}$.*

| Exchange in the ideal semantics | Exchange in the filter semantics |
|---|---|
| $(t \vartriangle u) \cdot (v \vartriangle w) \preceq_{\mathcal{I}} (t \cdot v) \vartriangle (u \cdot w)$ | $(t \cdot v) \vartriangle (u \cdot w) \preceq_{\mathcal{F}} (t \vartriangle u) \cdot (v \vartriangle w)$ |
| $(t \vartriangle u) \cdot w \preceq_{\mathcal{I}} t \vartriangle (u \cdot w)$ | $t \vartriangle (u \cdot w) \preceq_{\mathcal{F}} (t \vartriangle u) \cdot w$ |
| $u \cdot (v \vartriangle w) \preceq_{\mathcal{I}} v \vartriangle (u \cdot w)$ | $v \vartriangle (u \cdot w) \preceq_{\mathcal{F}} u \cdot (v \vartriangle w)$ |
| $t \cdot u \preceq_{\mathcal{I}} t \vartriangle u$ | $t \vartriangle u \preceq_{\mathcal{F}} t \cdot u$ |

Figure 9: Inequational axioms for the exchange of conjunctive and sequential refinement.

*Proof.* Most cases are immediate, we consider two cases that differ compared to that of Gischer [13].

Consider the equation $(t \cdot u) \cdot v \cong_{\mathcal{F}} t \cdot (u \cdot v)$. Observe that if $G \in [\![(t \cdot u) \cdot v]\!]_{\mathcal{F}}$ then there exist graphs $H$, $J$, $K$, $L$ and smoothing homomorphisms such that $H \in [\![t \cdot u]\!]_{\mathcal{F}}$ and $J \in [\![v]\!]_{\mathcal{F}}$ and $f \colon G \to HJ$, and $K \in [\![t]\!]_{\mathcal{F}}$ and $L \in [\![u]\!]_{\mathcal{F}}$ and $g \colon J \to KL$. Furthermore, $H\,(KL)$ is isomorphic to $(HK)\,L$; hence we can construct a smoothing homomorphism $h \colon G \to (HK)\,L$. Hence, since $(HK)\,L \in [\![t \cdot (u \cdot v)]\!]_{\mathcal{F}}$, by definition, $G \in [\![t \cdot (u \cdot v)]\!]_{\mathcal{F}}$. The converse direction is symmetric, hence $[\![(t \cdot u) \cdot v]\!]_{\mathcal{F}} = [\![t \cdot (u \cdot v)]\!]_{\mathcal{F}}$.

Consider the soundness of $(t \cdot v) \vartriangle (u \cdot w) \preceq_{\mathcal{F}} (t \vartriangle u) \cdot (v \vartriangle w)$. If $G \in [\![(t \cdot u) \vartriangle (v \cdot w)]\!]_{\mathcal{F}}$ then $G = H \uplus J$ such that $H \in [\![t \cdot u]\!]_{\mathcal{F}}$ and $J \in [\![v \cdot w]\!]_{\mathcal{F}}$. Hence there exist labelled series-parallel graphs $K$, $L$, $M$ and $N$ and smoothing homomorphisms $f \colon H \to KL$ and $g \colon J \to MN$ such that $K \in [\![t]\!]_{\mathcal{F}}$, $L \in [\![u]\!]_{\mathcal{F}}$, $M \in [\![v]\!]_{\mathcal{F}}$ and $N \in [\![w]\!]_{\mathcal{F}}$. Hence there is a smoothing homomorphisms from $G$ to $(KL) \uplus (MN)$. Observe that the function on vertices defined such that $((v, x), y) \mapsto ((v, y), x)$ is a smoothing homomorphism from $(KL) \uplus (MN)$ to $(K \uplus M)\,(L \uplus N)$. Hence there exists a smoothing homomorphism from $G$ to $(K \uplus M)\,(L \uplus N)$. Since, $K \uplus M \in [\![t \vartriangle u]\!]_{\mathcal{F}}$ and $L \uplus N \in [\![v \vartriangle w]\!]_{\mathcal{F}}$, we have that $(K \uplus M)\,(L \uplus N) \in [\![(t \vartriangle u) \cdot (v \vartriangle w)]\!]_{\mathcal{F}}$. Hence $G \in [\![(t \vartriangle u) \cdot (v \vartriangle w)]\!]_{\mathcal{F}}$. Thereby $[\![(t \cdot v) \vartriangle (u \cdot w)]\!]_{\mathcal{F}} \preceq_{\mathcal{F}} [\![(t \vartriangle u) \cdot (v \vartriangle w)]\!]_{\mathcal{F}}$. □

For completeness, we require the following lemma. This lemma concerns restricted trees called *codot* terms [13]. Codot terms are formed from the grammar $t ::= \boxed{\text{a}} \mid t \vartriangle t \mid t \cdot t$ using only conjunctive and sequential refinement, without disjunctive refinement. The direction of smoothing homomorphisms in this lemma is opposite to the corresponding case for the established construction for ideals, accounting for the direction of smoothing homomorphisms in the definition of filters.

**Lemma 27.** *For codot terms t and v, there exist labelled series-parallel graphs G and H such that $\phi(\{G\}) = [\![t]\!]_{\mathcal{F}}$ and $\phi(\{H\}) = [\![v]\!]_{\mathcal{F}}$. Furthermore, if $[\![t]\!]_{\mathcal{F}} \subseteq [\![v]\!]_{\mathcal{F}}$ then there exists a smoothing homomorphism from G to H.*

The above lemma is used in the proof of the following result that also concerns only codot terms. Other than the direction of smoothing homomorphisms in the above lemma, the details of the proof are identical to the proof of the corresponding lemma for the ideal construction [13].

**Lemma 28.** *If t and v are codot terms, if $[\![t]\!]_{\mathcal{F}} \subseteq [\![v]\!]_{\mathcal{F}}$ then either $[\![t]\!]_{\mathcal{F}} = [\![v]\!]_{\mathcal{F}}$ or there exists a codot term u such that $[\![t]\!]_{\mathcal{F}} \subseteq [\![u]\!]_{\mathcal{F}}$ and $[\![u]\!]_{\mathcal{F}} \subset [\![v]\!]_{\mathcal{F}}$ and $u \preceq_{\mathcal{F}} v$.*

Completeness of the filter semantics follows from the above lemma by the following argument. In the following, $\bigvee_{1 \leq i \leq m} t_i$ represents $t_1 \triangledown t_2 \triangledown \ldots \triangledown t_m$.

**Theorem 29** (Completeness)**.** *Given two causal attack trees t and u, if $[\![t]\!]_{\mathcal{F}} \subseteq [\![u]\!]_{\mathcal{F}}$ then $t \preceq_{\mathcal{F}} u$.*

*Proof.* Consider causal attack trees such that $[\![t]\!]_{\mathcal{F}} \subseteq [\![u]\!]_{\mathcal{F}}$. Observe that, due to the distributivity of $\triangledown$ over both $\vartriangle$ and $\cdot$, there exist codot terms $t_i$, for $1 \leq i \leq m$ and $u_j$ for $1 \leq j \leq n$ such that $t \cong_{\mathcal{F}} \bigvee_{1 \leq i \leq m} t_i$ and $u = \bigvee_{1 \leq j \leq n} u_j$ and $\left[\!\!\left[\bigvee_{1 \leq i \leq m} t_i\right]\!\!\right]_{\mathcal{F}} \subseteq \left[\!\!\left[\bigvee_{1 \leq j \leq n} u_j\right]\!\!\right]_{\mathcal{F}}$. For each $i$, $[\![t_i]\!]_{\mathcal{F}} \subseteq \bigcup_{1 \leq i \leq n} [\![t_i]\!]_{\mathcal{F}} = [\![t]\!]_{\mathcal{F}}$ hence $[\![t_i]\!]_{\mathcal{F}} \subseteq [\![u]\!]_{\mathcal{F}}$. Observe, that by Lemma 27 there exists labelled series-parallel graph $G$ such that $\phi(\{G\}) = [\![t_i]\!]_{\mathcal{F}}$ hence, since $[\![t_i]\!]_{\mathcal{F}} \subseteq [\![u]\!]_{\mathcal{F}}$ clearly $G \in [\![u]\!]_{\mathcal{F}} = \left[\!\!\left[\bigvee_{1 \leq j \leq n} u_j\right]\!\!\right]_{\mathcal{F}} = \bigcup_{1 \leq j \leq n} \left[\!\!\left[u_j\right]\!\!\right]_{\mathcal{F}}$; thereby there exists $j$ such that $G \in \left[\!\!\left[u_j\right]\!\!\right]_{\mathcal{F}}$. Since $G \in \left[\!\!\left[u_j\right]\!\!\right]_{\mathcal{F}}$ clearly $\phi(\{G\}) \subseteq \left[\!\!\left[u_j\right]\!\!\right]_{\mathcal{F}}$ since $\left[\!\!\left[u_j\right]\!\!\right]_{\mathcal{F}}$ is a filter, by Lemma 20; hence $[\![t_i]\!]_{\mathcal{F}} \subseteq \left[\!\!\left[u_j\right]\!\!\right]_{\mathcal{F}}$. By applying Lemma 28 repeatedly, there exists a sequence of terms $v^i_1, v^i_2, \ldots v^i_{k_i}$ such that $\left[\!\!\left[v^i_\ell\right]\!\!\right]_{\mathcal{F}} \subset \left[\!\!\left[v^i_{\ell+1}\right]\!\!\right]_{\mathcal{F}}$ and $v^i_1 = t_i$ and $v^i_{k_i} = u_j$ and $v^i_\ell \preceq_{\mathcal{F}} v^i_{\ell+1}$. The sequence is finite, since $u_j$ is finite and at least one edge is removed at each step. Hence $t_i \preceq_{\mathcal{F}} u_j$, from which

$t_i \triangledown \bigvee_{1\le k\le n} u_k \cong_{\mathcal{F}} t_i \triangledown u_j \triangledown \bigvee_{1\le k\le n} u_k \cong_{\mathcal{F}} u_j \triangledown \bigvee_{1\le k\le n} u_k \cong_{\mathcal{F}} \bigvee_{1\le k\le n} u_k$, forall $i$, by definition of $\preceq_{\mathcal{F}}$. Furthermore, $(\bigvee_{1\le i\le k+1} t_i) \triangledown u = (\bigvee_{1\le i\le k} t_i) \triangledown t_{k+1} \triangledown u \cong_{\mathcal{F}} (\bigvee_{1\le i\le k} t_i) \triangledown u$ and $(\bigvee_{1\le i\le 1} t_i) \triangledown u = t_1 \triangledown u \cong u$, hence by induction $(\bigvee_{1\le i\le m-1} t_i) \triangledown u \cong u$. Therefore $t \preceq_{\mathcal{F}} u$. □

*4.4. Soundness of Specialisations for Classes of Attribute Domain*

Attribute domains can be checked for soundness with respect to the specialisation preorder defined by the filter and ideal semantics. The choice of semantics depends on how the direction of the preorder for specialising attack trees exchange sequential and conjunctive refinement and also how the interpretation of disjunctive refinement affects the direction of the preorder.

Attribute domains and interpretation functions are extended to include sequential refinement according to the following definitions.

**Definition 30.** *Suppose $\mathcal{D} = (D, f, g, h)$ is an attribute domain with three binary operators over D. Assume also that $\vartheta$ is a valuation mapping basic actions to D. The interpretation function $I^{\mathcal{D}}_{\vartheta}(\,\cdot\,)$ from attack trees to D is defined as follows.*

$$I^{\mathcal{D}}_{\vartheta}\big(\boxed{a}\big) = \vartheta(\boxed{a}) \qquad I^{\mathcal{D}}_{\vartheta}(t \triangledown u) = f(I^{\mathcal{D}}_{\vartheta}(t), I^{\mathcal{D}}_{\vartheta}(u))$$

$$I^{\mathcal{D}}_{\vartheta}(t \vartriangle u) = g(I^{\mathcal{D}}_{\vartheta}(t), I^{\mathcal{D}}_{\vartheta}(u)) \qquad I^{\mathcal{D}}_{\vartheta}(t \cdot u) = h(I^{\mathcal{D}}_{\vartheta}(t), I^{\mathcal{D}}_{\vartheta}(u))$$

*As for attack trees, we say that a semantics given by a specialisation preorder $\preceq$ is sound with respect to an attribute domain $\mathcal{D}$ ordered by $\preceq_{\mathcal{D}}$, whenever for all causal attack trees t and u and valuations $\vartheta$, if $t \preceq u$ then $I^{\mathcal{D}}_{\vartheta}(t) \preceq_{\mathcal{D}} I^{\mathcal{D}}_{\vartheta}(u)$.*

Now consider the attribute domain $\mathcal{A} = (\mathbb{N}, \min, +, \max)$, which can represent the minimum number of experts required to perform an attack. This attribute domain assumes concurrent actions must be performed independently by separate experts, as indicated by the interpretation of conjunctive refinement as addition. Observe that the interpretation of sequential and conjunctive refinement distribute over each other as follows:

$$\begin{aligned} I^{\mathcal{A}}_{\vartheta}((t \cdot u) \vartriangle (v \cdot w)) &= \max\Big(I^{\mathcal{A}}_{\vartheta}(t), I^{\mathcal{A}}_{\vartheta}(u)\Big) + \max\Big(I^{\mathcal{A}}_{\vartheta}(v), I^{\mathcal{A}}_{\vartheta}(w)\Big) \\ &\ge \max\Big(I^{\mathcal{A}}_{\vartheta}(t) + I^{\mathcal{A}}_{\vartheta}(v), I^{\mathcal{A}}_{\vartheta}(u) + I^{\mathcal{A}}_{\vartheta}(w)\Big) = I^{\mathcal{A}}_{\vartheta}((t \vartriangle v) \cdot (u \vartriangle w)) \end{aligned}$$

Also notice that, for numbers $m$ and $n$, $m \ge n$ whenever $\min(m, n) = n$. Hence, since $t \preceq_{\mathcal{F}} u$ is defined such that $t \triangledown u \cong_{\mathcal{F}} u$, clearly $I^{\mathcal{A}}_{\vartheta}(t \triangledown u) = I^{\mathcal{A}}_{\vartheta}(u)$, if and only if $\min\Big\{I^{\mathcal{A}}_{\vartheta}(t), I^{\mathcal{A}}_{\vartheta}(u)\Big\} = I^{\mathcal{A}}_{\vartheta}(u)$, if and only if $I^{\mathcal{A}}_{\vartheta}(t) \ge I^{\mathcal{A}}_{\vartheta}(u)$. This leads us to the following property, indicating that attribute domain $\mathcal{A}$ is sound with respect to the filter semantics.

**Proposition 31.** *For causal attack trees t and u, for all $\vartheta$, if $t \preceq_{\mathcal{F}} u$ then $I^{\mathcal{A}}_{\vartheta}(t) \ge I^{\mathcal{A}}_{\vartheta}(u)$.*

In contrast, the attribute domain $\mathcal{B} = (\mathbb{R}, \min, \max, +)$ is sound with respect to the ideal based semantics. This domain can represent the minimum time required to complete some attack, where instead sequential refinement is interpreted using $+$. Consequently, the direction in which sequential and conjunctive refinement distribute over each other is reversed with respect to the numerical ordering as follows $I^{\mathcal{B}}_{\vartheta}((t \vartriangle v) \cdot (u \vartriangle w)) \ge I^{\mathcal{B}}_{\vartheta}((t \cdot u) \vartriangle (v \cdot w))$. This leads us to the following property, indicating that attribute domain $\mathcal{B}$ is sound with respect to the ideal semantics.

**Proposition 32.** *For causal attack trees t and u, for all $\vartheta$, if $t \preceq_{\mathcal{I}} u$ then $I^{\mathcal{B}}_{\vartheta}(u) \ge I^{\mathcal{B}}_{\vartheta}(t)$.*

Notice that in both attribute domains the interpretation of disjunctive refinement can be switched to max, to obtain two further domains. The domain $\mathcal{C} = (\mathbb{N}, \max, +, \max)$ can be used to measure the "required number of experts on duty to counter any attack". This attribute domain is sound with respect to the ideal semantics, but with respect to reverse numerical order as follows.

**Proposition 33.** *For causal attack trees t and u, for all $\vartheta$, if $t \preceq_{\mathcal{I}} u$ then $I^{\mathcal{C}}_{\vartheta}(t) \le I^{\mathcal{C}}_{\vartheta}(u)$.*

Similarly, the domain $\mathcal{E} = (\mathbb{R}, \max, \max, +)$ could represent the "time required to make all attacks possible", and is sound with respect to the filter semantics, as follows.

**Proposition 34.** *For causal attack trees t and u, for all $\vartheta$, if $t \preceq_{\mathcal{F}} u$ then $I^{\mathcal{E}}_{\vartheta}(t) \le I^{\mathcal{E}}_{\vartheta}(u)$.*

*Action refinement in terms of sound attribute domains.* Recall that nodes in a causal attack tree represent the refinement of a basic action into multiple sub-goals. Therefore, any semantics for causal attack trees should accommodate what is known as *action refinement* [15] in the process calculus literature. Action refinement can be expressed as a special attribute domain $\mathcal{T} = (\mathbb{T}, \triangledown, \triangle, \cdot)$, where the domain $\mathbb{T}$ is the space of all causal attack trees and each of conjunctive, disjunctive and sequential refinement are interpreted using themselves. Each valuation in this attribute domain defines an action refinement. Thereby, soundness of a semantics with respect to action refinement is expressed as follows.

**Proposition 35.** *For any causal attack trees $t$ and $u$ and valuation $\vartheta$ mapping basic actions to $\mathbb{T}$, if $t \preceq u$ then $I^{\mathcal{T}}_{\vartheta}(t) \preceq I^{\mathcal{T}}_{\vartheta}(u)$, where $\preceq \in \{\preceq_{\mathcal{F}}, \preceq_{\mathcal{I}}\}$.*

## 5. Logical Systems for Specialising Causal Attack Trees

For attack trees, the multiset semantics coincide with implication for an embedding of attack trees in linear logic. A natural extension of this observation is to consider whether the ideal and filter semantics, introduced in the previous section for causal attack trees, have an underlying logical system. A logical system sound with respect to each of the ideal and filter semantics is indeed enabled by an extension of linear logic with a non-commutative operator, called MAV [8]. We briefly introduce the system MAV, which is expressed using a generalisation of the sequent calculus, called the *calculus of structures*. We then return to motivating examples from earlier in the paper, and demonstrate how the logical system can be used to decide whether one causal attack tree specialises another causal attack tree.

### *5.1. A Conservative Extension of MALL with Sequential Composition*

MAV is an extension of MALL with a non-commutative operator called *seq* (";"). Seq was introduced in the system BV [7], which is a subsystem of MAV. The following grammar defines propositions.

$$P ::= \mathrm{I} \mid a \mid \overline{a} \mid P \parr P \mid P \otimes P \mid P \oplus P \mid P \mathbin{\&} P \mid P \mathbin{;} P$$

Observe that we write propositions in negation normal form, where linear negation is pushed to the atomic propositions. The operators $\parr$ and & are de Morgan dual to $\otimes$ and $\oplus$ respectively, while seq and the unit are self-dual, as defined by the following function from propositions to propositions.

$$\overline{\overline{a}} = a \qquad \overline{\mathrm{I}} = \mathrm{I} \qquad \overline{(P \otimes Q)} = \overline{P} \parr \overline{Q} \qquad \overline{(P \parr Q)} = \overline{P} \otimes \overline{Q}$$

$$\overline{P \mathbin{;} Q} = \overline{P} \mathbin{;} \overline{Q} \qquad \overline{(P \oplus Q)} = \overline{P} \mathbin{\&} \overline{Q} \qquad \overline{(P \mathbin{\&} Q)} = \overline{P} \oplus \overline{Q}$$

The semantics of MAV is defined by a term rewriting system modulo an equational theory. The rewrite rules and equational theory are presented in Fig. 10. All rules can be applied in any context, where a context $C\{\ \}$ is a proposition with one hole $\{\cdot\}$ in which any proposition can be plugged, as defined by the following grammar: $C\{\ \} ::= \{\cdot\} \mid P \odot C\{\ \} \mid C\{\ \} \odot P$, where $\odot \in \{;, \parr, \otimes, \oplus, \&\}$.

We also have the following congruence relation that serves a similar role to the exchange rule in the sequent calculus. The main difference compared to sequents is that structural rules reordering propositions can be applied deep within any context. The equational system ensures that $(P, ;, \mathrm{I})$ is a monoid, and both $(P, \parr, \mathrm{I})$ and $(P, \otimes, \mathrm{I})$ are commutative monoids.

$$(P \parr Q) \parr R \equiv P \parr (Q \parr R) \qquad P \parr Q \equiv Q \parr P \qquad P \parr \mathrm{I} \equiv P$$

$$(P \otimes Q) \otimes R \equiv P \otimes (Q \otimes R) \qquad P \otimes Q \equiv Q \otimes P \qquad P \otimes \mathrm{I} \equiv P$$

$$(P \mathbin{;} Q) \mathbin{;} R \equiv P \mathbin{;} (Q \mathbin{;} R) \qquad \mathrm{I} \mathbin{;} P \equiv P \qquad P \mathbin{;} \mathrm{I} \equiv P$$

In the calculus of structures, a *proof* is a derivation that reduces to the unit, where the unit represents a successfully completed proof. Rewrite relation $\longrightarrow$ denotes its own reflexive and transitive closure.

$$\mathrm{I} \,\&\, \mathrm{I} \longrightarrow \mathrm{I} \quad \textit{tidy} \qquad\qquad \overline{a} \parallel a \longrightarrow \mathrm{I} \quad \textit{atomic interaction}$$

$$(P \otimes Q) \parallel R \longrightarrow P \otimes (Q \parallel R) \quad \textit{switch}$$

$$(P \,;\, Q) \parallel (R \,;\, S) \longrightarrow (P \parallel R) \,;\, (Q \parallel S) \quad \textit{sequence}$$

$$P \oplus Q \longrightarrow P \quad \textit{left} \qquad\qquad P \oplus Q \longrightarrow Q \quad \textit{right}$$

$$(P \,\&\, Q) \parallel R \longrightarrow (P \parallel R) \,\&\, (Q \parallel R) \quad \textit{external}$$

$$(P \,;\, Q) \,\&\, (R \,;\, S) \longrightarrow (P \,\&\, R) \,;\, (Q \,\&\, S) \quad \textit{medial}$$

Figure 10: Term rewriting system modulo an equational theory for MAV.

**Definition 36.** *A derivation $P \longrightarrow Q$ is any sequence of Fig. 10, modulo the structural congruence $\equiv$. For any proposition P, if there exists a derivation of the form $P \longrightarrow \mathrm{I}$, then we write $\vdash P$, and say that P is provable.*

*Linear implication in MAV, written $P \multimap Q$, is defined by $\overline{P} \parallel Q$.*

An established result [8], is a generalisation of *cut elimination* to the setting of MAV, which implies the transitivity of implication and consistency of MAV.

**Theorem 37** (Cut elimination, Horne 2015)**.** *For any proposition P, if $\vdash C\{\, P \otimes \overline{P} \,\}$, then $\vdash C\{\, \mathrm{I} \,\}$.*

*5.2. Embeddings of Causal Attack Trees in Non-Commutative Logic*

The logical system MAV provides us with methods for deciding whether one tree is a specialisation of another according to the ideal and filter semantics. Recall that a fragment of MALL coincides with the multiset semantics for attack trees. Since MAV is a conservative extension of MALL with a sequential operator, which models the causal order of events, MAV is a natural setting for systematising the semantics for causal attack trees. This section highlights that the two semantics for causal attack trees, the ideal and filter semantics, can be systematised soundly by two fragments of MAV.

The first logical system, which will be used to systematise the ideal semantics for causal attack trees, is defined as follows.

**Definition 38** (ideal system)**.** *The ideal system is defined by the following function from causal attack trees to propositions.*

$$\begin{array}{rclcrcl} [\![ \boxed{a} ]\!]_{\mathcal{LI}} & = & a & \qquad & [\![ t \bigtriangledown u ]\!]_{\mathcal{LI}} & = & [\![t]\!]_{\mathcal{LI}} \oplus [\![u]\!]_{\mathcal{LI}} \\ [\![ t \bigtriangleup u ]\!]_{\mathcal{LI}} & = & [\![t]\!]_{\mathcal{LI}} \parallel [\![u]\!]_{\mathcal{LI}} & & [\![ t \cdot u ]\!]_{\mathcal{LI}} & = & [\![t]\!]_{\mathcal{LI}} \,;\, [\![u]\!]_{\mathcal{LI}} \end{array}$$

The systematisation of filters expressed using MAV is defined as follows. Notice that the operator $\otimes$ is used for filters, as opposed to $\parallel$ in the system for ideals, to interpret conjunctive refinement.

**Definition 39** (filter system)**.** *The filter system is defined by the following function from causal attack trees to propositions.*

$$\begin{array}{rclcrcl} [\![ \boxed{a} ]\!]_{\mathcal{LF}} & = & a & \qquad & [\![ t \bigtriangledown u ]\!]_{\mathcal{LF}} & = & [\![t]\!]_{\mathcal{LF}} \oplus [\![u]\!]_{\mathcal{LF}} \\ [\![ t \bigtriangleup u ]\!]_{\mathcal{LF}} & = & [\![t]\!]_{\mathcal{LF}} \otimes [\![u]\!]_{\mathcal{LF}} & & [\![ t \cdot u ]\!]_{\mathcal{LF}} & = & [\![t]\!]_{\mathcal{LF}} \,;\, [\![u]\!]_{\mathcal{LF}} \end{array}$$

Before proving that the ideal and filter systems are sound for their respective semantics, we provide some examples. These examples demonstrate how the logical system can be used to check when one causal attack tree soundly specialises another.

Recall the motivating examples in Section 2. In Section 2, we claim that the attack tree in Fig. 2 is a specialisation of Fig. 1 for both attribute domains presented. The causal attack tree from Fig. 1 is recalled below, using the syntax of causal attack trees.

$$\Big(\boxed{\text{bribe}} \bigtriangledown \boxed{\text{steal}}\Big) \bigtriangleup \Big(\boxed{\text{break in}} \cdot \boxed{\text{install}}\Big)$$

For convenience, we also recall the causal attack tree from Fig. 2.

$$\big(\boxed{\text{bribe}} \vartriangle \big(\boxed{\text{break in}} \cdot \boxed{\text{install}}\big)\big) \triangledown \big(\boxed{\text{break in}} \cdot \big(\boxed{\text{steal}} \vartriangle \boxed{\text{install}}\big)\big)$$

Notice we abbreviate the labels for basic actions to aid the readability of propositions below.

By Proposition 32 the ideal semantics is sound with respect to the "minimum attack time" attribute domain. Thus, to check the soundness of the specialisation of Fig. 1 to Fig. 2 for the "minimum attack time" attribute domain, it is sufficient to establishing the following implication between embedding of Fig. 2 and Fig. 1 according to the ideal system.

$$\vdash (\text{bribe} \parallel (\text{break in} \mathbin{;} \text{install})) \oplus (\text{break in} \mathbin{;} (\text{steal} \parallel \text{install})) \multimap (\text{bribe} \oplus \text{steal}) \parallel (\text{break in} \mathbin{;} \text{install})$$

The proof of the above implication, as a derivation using the rules of MAV to I, is presented in Fig. 11.

$$\begin{array}{c}
(\text{bribe} \parallel (\text{break in} \mathbin{;} \text{install})) \oplus (\text{break in} \mathbin{;} (\text{steal} \parallel \text{install})) \multimap (\text{bribe} \oplus \text{steal}) \parallel (\text{break in} \mathbin{;} \text{install}) \\
= \text{by de Morgan dualities and definition of implication} \\
\big(\big(\overline{\text{bribe}} \otimes \big(\overline{\text{break in}} \mathbin{;} \overline{\text{install}}\big)\big) \,\&\, \big(\overline{\text{break in}} \mathbin{;} \big(\overline{\text{steal}} \otimes \overline{\text{install}}\big)\big)\big) \parallel (\text{bribe} \oplus \text{steal}) \parallel (\text{break in} \mathbin{;} \text{install}) \\
\downarrow \text{by } \textit{external} \text{ rule} \\
\big(\big(\overline{\text{bribe}} \otimes \big(\overline{\text{break in}} \mathbin{;} \overline{\text{install}}\big)\big) \parallel (\text{bribe} \oplus \text{steal}) \parallel (\text{break in} \mathbin{;} \text{install})\big) \,\&\, \big(\big(\overline{\text{break in}} \mathbin{;} \big(\overline{\text{steal}} \otimes \overline{\text{install}}\big)\big) \parallel (\text{bribe} \oplus \text{steal}) \parallel (\text{break in} \mathbin{;} \text{install})\big) \\
\downarrow \text{by } \textit{left} \text{ and } \textit{right} \text{ rules} \\
\big(\big(\overline{\text{bribe}} \otimes \big(\overline{\text{break in}} \mathbin{;} \overline{\text{install}}\big)\big) \parallel \text{bribe} \parallel (\text{break in} \mathbin{;} \text{install})\big) \,\&\, \big(\big(\overline{\text{break in}} \mathbin{;} \big(\overline{\text{steal}} \otimes \overline{\text{install}}\big)\big) \parallel \text{steal} \parallel (\text{break in} \mathbin{;} \text{install})\big) \\
\downarrow \text{by } \textit{switch} \text{ rule} \\
\big(\big(\overline{\text{bribe}} \parallel \text{bribe}\big) \otimes \big(\big(\overline{\text{break in}} \mathbin{;} \overline{\text{install}}\big) \parallel (\text{break in} \mathbin{;} \text{install})\big)\big) \,\&\, \big(\big(\overline{\text{break in}} \mathbin{;} \big(\overline{\text{steal}} \otimes \overline{\text{install}}\big)\big) \parallel \text{steal} \parallel (\text{break in} \mathbin{;} \text{install})\big) \\
\downarrow \text{by } \textit{sequence} \text{ rule} \\
\big(\big(\overline{\text{bribe}} \parallel \text{bribe}\big) \otimes \big(\big(\overline{\text{break in}} \parallel \text{break in}\big) \mathbin{;} \big(\overline{\text{install}} \parallel \text{install}\big)\big)\big) \,\&\, \big(\big(\overline{\text{break in}} \parallel \text{break in}\big) \mathbin{;} \big(\big(\overline{\text{steal}} \otimes \overline{\text{install}}\big) \parallel \text{steal} \parallel \text{install}\big)\big) \\
\downarrow \text{by } \textit{switch} \text{ rule} \\
\big(\big(\overline{\text{bribe}} \parallel \text{bribe}\big) \otimes \big(\big(\overline{\text{break in}} \parallel \text{break in}\big) \mathbin{;} \big(\overline{\text{install}} \parallel \text{install}\big)\big)\big) \,\&\, \big(\big(\overline{\text{break in}} \parallel \text{break in}\big) \mathbin{;} \big(\big(\overline{\text{steal}} \parallel \text{steal}\big) \otimes \big(\overline{\text{install}} \parallel \text{install}\big)\big)\big) \\
\downarrow \text{by } \textit{atomic interaction} \text{ rule} \\
\text{I}
\end{array}$$

Figure 11: Derivation establishing that Fig. 2 specialises Fig. 1 in the ideal semantics.

The motivating section also claims that Fig. 1 soundly specialises Fig. 2 with respect to the "minimum number of experts" attribute domain. By Proposition 31, the filter semantics is sound with respect to the "minimum number of experts" attribute domain. Hence we can formally verify the claim using the filter system, by establishing the following implication.

$$\vdash (\text{bribe} \oplus \text{steal}) \otimes (\text{break in} \mathbin{;} \text{install}) \multimap (\text{bribe} \otimes (\text{break in} \mathbin{;} \text{install})) \oplus (\text{break in} \mathbin{;} (\text{steal} \otimes \text{install}))$$

Observe that the order of the implication between the embedding of the trees is opposite to the direction for the ideal semantics. This is reflected in the reversal of the preorder of the "minimum number of experts" attribute domain compared to the "minimum attack time" attribute domain in the worked example in Section 2. Furthermore, the proof of the implication is quite different, hence this implication holds for different reasons for the filter system compared to the ideal system.

Recall that the tree in Fig. 3(a) is a sound specialisation of Fig. 2 with respect to the "minimum number of experts" attribute domain. This is verified by the following implication, between the filter embeddings of the trees in Fig. 3(a) and Fig. 2.

$$\vdash \text{steal} \otimes (\text{break in} \mathbin{;} \text{install}) \multimap (\text{bribe} \otimes (\text{break in} \mathbin{;} \text{install})) \oplus (\text{break in} \mathbin{;} (\text{steal} \otimes \text{install}))$$

In contrast, there is no sound specialisation, in either direction, between the embedding of the same trees according to the ideal system. However, Fig. 3(b) does soundly specialise Fig. 1 according to the "minimum attack time" attribute domain. This is verified by the following implication, between the ideal embedding of the trees in Fig. 3(b) and Fig. 1.

$$\vdash (\text{bribe} \oplus \text{steal}) \parallel (\text{break in} \mathbin{;} \text{install}) \multimap \text{break in} \mathbin{;} (\text{steal} \parallel \text{install})$$

Notice that there is no proof in MAV of either implication between the filter embedding of Fig. 3(b) and Fig. 1. Since MAV is decidable, a decision procedure can be extracted from the deductive system, which can be used as the basis of a tool for automatically checking such scenarios.

*Soundness of each logical system.* Each logical embedding of attack trees is sound with respect to its corresponding semantics. To establish the soundness of the filter semantics we require the following lemmas. The proof exploits the fact that a derivation beginning with a linearly negated embedding of a causal attack tree in the filter system uses only rules that are sound with respect to the filter semantics; hence is established by induction over the length of a deriation.

**Lemma 40.** *If* $\overline{[\![t]\!]_{\mathcal{LF}}} \longrightarrow Q$*, then there exists* $u$ *such that* $Q \equiv \overline{[\![u]\!]_{\mathcal{LF}}}$ *and* $[\![t]\!]_{\mathcal{F}} \subseteq [\![u]\!]_{\mathcal{F}}$.

*Proof.* Proceed by induction over the length of the derivation. The base case, $[\![t]\!]_{\mathcal{LF}} \equiv Q$ follows since conjunctive refinement is associative and commutative, sequential refinement is associative, and all units can be removed from $Q$ to obtain an embedding of a causal attack trees. For the inductive cases, the only rules of MAV (from Fig. 10) that can be applied in a derivation are the *sequence*, *external* and *medial* rules.

Assume that $\overline{[\![s]\!]_{\mathcal{LF}}} \longrightarrow \overline{[\![C\{\, (t \cdot u) \vartriangle (v \cdot w) \,\}]\!]_{\mathcal{LF}}} = C'\{\, ([\![t]\!]_{\mathcal{LF}} \,;\, [\![u]\!]_{\mathcal{LF}}) \parallel ([\![v]\!]_{\mathcal{LF}} \,;\, [\![w]\!]_{\mathcal{LF}}) \,\}$ holds such that $[\![s]\!]_{\mathcal{F}} \subseteq [\![C\{\, (t \cdot u) \parallel (v \cdot w) \,\}]\!]_{\mathcal{F}}$. Suppose that the *sequence* rule is applied as follows.

$$\begin{array}{rcl} C'\{\, ([\![t]\!]_{\mathcal{LF}} \,;\, [\![u]\!]_{\mathcal{LF}}) \parallel ([\![v]\!]_{\mathcal{LF}} \,;\, [\![w]\!]_{\mathcal{LF}}) \,\} & \longrightarrow & C'\{\, ([\![t]\!]_{\mathcal{LF}} \parallel [\![v]\!]_{\mathcal{LF}}) \,;\, ([\![u]\!]_{\mathcal{LF}} \,;\, [\![w]\!]_{\mathcal{LF}}) \,\} \\ & = & \overline{[\![C\{\, (t \vartriangle v) \cdot (u \vartriangle w) \,\}]\!]_{\mathcal{LF}}} \end{array}$$

Since $C\{\, (t \cdot u) \vartriangle (v \cdot w) \,\} \preceq_{\mathcal{F}} C\{\, (t \vartriangle v) \cdot (u \vartriangle w) \,\}$ and preorder $\preceq_{\mathcal{F}}$ is sound with respect to the filter semantics the inclusion $[\![C\{\, (t \cdot u) \vartriangle (v \cdot w) \,\}]\!]_{\mathcal{F}} \subseteq [\![C\{\, (t \vartriangle v) \cdot (u \vartriangle w) \,\}]\!]_{\mathcal{F}}$ holds. Thereby $[\![s]\!]_{\mathcal{F}} \subseteq [\![C\{\, [\![(t \vartriangle v) \cdot (u \vartriangle w)]\!]_{\mathcal{LF}} \,\}]\!]_{\mathcal{F}}$ as required. The three other cases for the *sequence* rule are similar, except that unit propositions are introduced then removed when deriving the remaining inequalities relating conjunctive and sequential refinement.

Assume $\overline{[\![s]\!]_{\mathcal{LF}}} \longrightarrow \overline{C\{\, [\![(t \triangledown u) \vartriangle v]\!]_{\mathcal{LF}} \,\}} = C'\{\, ([\![t]\!]_{\mathcal{LF}} \,\&\, [\![u]\!]_{\mathcal{LF}}) \parallel [\![v]\!]_{\mathcal{LF}} \,\}$ and $[\![s]\!]_{\mathcal{F}} \subseteq [\![C\{\, [\![(t \triangledown u) \vartriangle v]\!]_{\mathcal{LF}} \,\}]\!]_{\mathcal{F}}$. Suppose that the *external* rule is applied as follows.

$$\begin{array}{rcl} C'\{\, ([\![t]\!]_{\mathcal{LF}} \,\&\, [\![u]\!]_{\mathcal{LF}}) \parallel [\![v]\!]_{\mathcal{LF}} \,\} & \longrightarrow & C'\{\, ([\![t]\!]_{\mathcal{LF}} \parallel [\![v]\!]_{\mathcal{LF}}) \,\&\, ([\![u]\!]_{\mathcal{LF}} \parallel [\![v]\!]_{\mathcal{LF}}) \,\} \\ & = & [\![C\{\, (t \vartriangle v) \triangledown (u \vartriangle v) \,\}]\!]_{\mathcal{LF}} \end{array}$$

Since $C\{\, [\![(t \triangledown u) \vartriangle v]\!]_{\mathcal{LF}} \,\} \preceq_{\mathcal{F}} C\{\, (t \vartriangle v) \triangledown (u \vartriangle v) \,\}$ holds and $\preceq_{\mathcal{F}}$ is sound with respect to the filter semantics the inclusion $[\![C\{\, [\![(t \triangledown u) \vartriangle v]\!]_{\mathcal{LF}} \,\}]\!]_{\mathcal{F}} \subseteq [\![C\{\, (t \vartriangle v) \triangledown (u \vartriangle v) \,\}]\!]_{\mathcal{F}}$ holds. Thereby $[\![s]\!]_{\mathcal{F}} \subseteq [\![C\{\, (t \vartriangle v) \triangledown (u \vartriangle v) \,\}]\!]_{\mathcal{F}}$ as required. The case for the medial rule follows similarly, observing that $(t \cdot v) \triangledown (v \cdot w) \preceq_{\mathcal{F}} (t \triangledown u) \cdot (v \triangledown w)$ is derivable using the distributivity of $\cdot$ and idempotency of $\triangledown$. □

The proof of the following lemma is similar to the above except that for the ideal embedding only the *sequence*, *left* and *right* rules need to be considered. All three rules respect the converse of subset inclusion for ideals.

**Lemma 41.** *If* $[\![t]\!]_{\mathcal{LI}} \longrightarrow Q$*, there exists* $u$ *such that* $Q \equiv [\![u]\!]_{\mathcal{LI}}$ *and* $[\![t]\!]_{\mathcal{I}} \subseteq [\![u]\!]_{\mathcal{I}}$.

The proofs rely on the following definition and lemma, which is used for proving cut elimination for MAV (Theorem 37) proven in related work [8].

**Definition 42.** *An n-ary killing context* $\mathcal{T}\{\ \}$ *is a context with n holes such that:*

- *if* $n = 1$*, then* $\mathcal{T}\{\ \} = \{\, \cdot \,\}$ *where* $\{\, \cdot \,\}$ *is a hole into which any proposition can be plugged;*
- *if* $m \geq 1$ *and* $n \geq 1$*, then if* $\mathcal{T}^0\{\ \}$ *is an m-ary killing context and* $\mathcal{T}^1\{\ \}$ *is an n-ary killing context, then* $\mathcal{T}^0\{\ \} \,\&\, \mathcal{T}^1\{\ \}$ *is an* $(m + n)$*-ary killing context.*

The following lemma is used to reorganise proofs to simulate the application of rules as in the sequent calculus, in the more general setting of the calculus of structures. This lemma is required to guide the induction in the theorems that follow.

**Lemma 43** (Splitting)**.** *The following statements hold.*

- *If* $\vdash (P \otimes Q) \parallel R$*, then there exist propositions* $V_i$ *and* $W_i$ *such that* $\vdash P \parallel V_i$ *and* $\vdash Q \parallel W_i$*, where* $1 \leq i \leq n$*, and n-ary killing context* $\mathcal{T}\{\ \}$ *such that* $R \longrightarrow \mathcal{T}\{\ V_1 \parallel W_1, \ldots, V_n \parallel W_n\ \}$*.*
- *If* $\vdash (P\ ;\ Q) \parallel R$*, then there exist propositions* $V_i$ *and* $W_i$ *such that* $\vdash P \parallel V_i$ *and* $\vdash Q \parallel W_i$*, where* $1 \leq i \leq n$*, and n-ary killing context* $\mathcal{T}\{\ \}$ *such that* $R \longrightarrow \mathcal{T}\{\ V_1\ ;\ W_1, \ldots, V_n\ ;\ W_n\ \}$*.*
- *If* $\vdash (P \oplus Q) \parallel R$*, then there exist propositions* $V_i$ *such that either* $\vdash P \parallel V_i$ *or* $\vdash Q \parallel V_i$*, where* $1 \leq i \leq n$*, and n-ary killing context* $\mathcal{T}\{\ \}$ *such that* $R \longrightarrow \mathcal{T}\{\ V_1, \ldots, V_n\ \}$*.*
- *If* $\vdash a \parallel U$*, then there exists n-ary killing context* $\mathcal{T}\{\ \}$ *such that* $U \longrightarrow \mathcal{T}\{\ \overline{a}, \ldots, \overline{a}\ \}$*.*
- *If* $\vdash \overline{a} \parallel U$*, then there exists n-ary killing context* $\mathcal{T}\{\ \}$ *such that* $U \longrightarrow \mathcal{T}\{\ a, \ldots, a\ \}$*.*
- *If* $\vdash (P\ \&\ Q) \parallel R$ *then both* $\vdash P \parallel R$ *and* $\vdash Q \parallel R$ *hold.*

The soundness result for the logical embedding $[\![\ \cdot\ ]\!]_{\mathcal{LF}}$ with respect to the filter semantics, is established by the following theorem. The proof idea is that, since causal attack trees are encoded using only $\otimes$, $\oplus$ and ;, we can always apply a technique called *splitting*, developed to prove Theorem 37, to reorganise a proof around the structure of the causal attack tree, such that the soundness of the attack tree depends on the soundness of smaller sub-trees. Furthermore, Lemma 40 ensures that the steps taken to reorganise the proof are also sound. Thereby the proof proceeds by induction on the structure of the attack tree in the conclusion of the implication.

**Theorem 44.** *For causal attack trees t and u, if* $\vdash [\![t]\!]_{\mathcal{LF}} \multimap [\![u]\!]_{\mathcal{LF}}$ *in* MAV *then* $[\![t]\!]_{\mathcal{F}} \subseteq [\![u]\!]_{\mathcal{F}}$*.*

*Proof.* Consider the base case when $\vdash [\![t]\!]_{\mathcal{LF}} \multimap \left[\!\left[\boxed{\mathrm{a}}\right]\!\right]_{\mathcal{LF}}$, where $a$ is an atomic proposition. By definition, $\vdash \overline{[\![t]\!]_{\mathcal{LF}}} \parallel a$, hence by Lemma 43, there exists killing context $\mathcal{T}\{\ \}$ such that $\overline{[\![t]\!]_{\mathcal{LF}}} \longrightarrow \mathcal{T}\{\ \overline{a}, \overline{a}, \ldots, \overline{a}\ \}$. By Lemma 40, $[\![t]\!]_{\mathcal{F}} \subseteq \left[\!\left[\boxed{\mathrm{a}} \triangledown \boxed{\mathrm{a}} \ldots \triangledown \boxed{\mathrm{a}}\right]\!\right]_{\mathcal{F}}$. Now $\left[\!\left[\boxed{\mathrm{a}} \triangledown \boxed{\mathrm{a}} \ldots \triangledown \boxed{\mathrm{a}}\right]\!\right]_{\mathcal{F}} = \left[\!\left[\boxed{\mathrm{a}}\right]\!\right]_{\mathcal{F}}$ from the definitions. Hence $[\![t]\!]_{\mathcal{F}} \subseteq \left[\!\left[\boxed{\mathrm{a}}\right]\!\right]_{\mathcal{F}}$, as required.

Consider the case when $\vdash [\![t]\!]_{\mathcal{LF}} \multimap [\![u \vartriangle v]\!]_{\mathcal{LF}}$. By definition, $\vdash \overline{[\![t]\!]_{\mathcal{LF}}} \parallel ([\![u]\!]_{\mathcal{LF}} \otimes [\![v]\!]_{\mathcal{LF}})$, hence by Lemma 43 there exist propositions $R_1^i$ and $R_2^i$ such that $\vdash \overline{R_1^i} \parallel [\![u]\!]_{\mathcal{LF}}$ and $\vdash \overline{R_2^i} \parallel [\![v]\!]_{\mathcal{LF}}$, for $1 \leq i \leq n$, and the following derivation holds: $\overline{[\![t]\!]_{\mathcal{LF}}} \longrightarrow \mathcal{T}\left\{ \overline{R_1^1} \parallel \overline{R_2^1}, \overline{R_1^2} \parallel \overline{R_2^2}, \ldots, \overline{R_1^n} \parallel \overline{R_2^n} \right\}$. Hence, by Lemma 40, there exist $w_1^i$ and $w_2^i$ such that $\left[\!\left[w_1^i\right]\!\right]_{\mathcal{LF}} = R_1^i$ and $\left[\!\left[w_2^i\right]\!\right]_{\mathcal{LF}} = R_2^i$, for all $1 \leq i \leq n$, and the following equivalence holds.

$$\mathcal{T}\left\{ \overline{R_1^1} \parallel \overline{R_2^1}, \overline{R_1^2} \parallel \overline{R_2^2}, \ldots, \overline{R_1^n} \parallel \overline{R_2^n} \right\} \equiv \left[\!\left[\left(w_1^1 \vartriangle w_2^1\right) \triangledown \left(w_1^2 \vartriangle w_2^2\right) \triangledown \ldots \left(w_1^n \vartriangle w_2^n\right)\right]\!\right]_{\mathcal{LF}}$$

Furthermore $[\![t]\!]_{\mathcal{F}} \subseteq \left[\!\left[\left(w_1^1 \vartriangle w_2^1\right) \triangledown \left(w_1^2 \vartriangle w_2^2\right) \triangledown \ldots \left(w_1^n \vartriangle w_2^n\right)\right]\!\right]_{\mathcal{F}}$. As the induction hypothesis, assume that $\left[\!\left[w_1^i\right]\!\right]_{\mathcal{F}} \subseteq [\![u]\!]_{\mathcal{F}}$ and $\left[\!\left[w_2^i\right]\!\right]_{\mathcal{F}} \subseteq [\![v]\!]_{\mathcal{F}}$. Thereby we have that $\left[\!\left[w_1^i \vartriangle w_2^i\right]\!\right]_{\mathcal{F}} \subseteq [\![u \vartriangle v]\!]_{\mathcal{F}}$, for all $i$; so $\left[\!\left[\left(w_1^1 \vartriangle w_2^1\right) \triangledown \ldots \left(w_1^n \vartriangle w_2^n\right)\right]\!\right]_{\mathcal{F}} = \left[\!\left[w_1^1 \vartriangle w_2^1\right]\!\right]_{\mathcal{F}} \cup \ldots \left[\!\left[w_1^n \vartriangle w_2^n\right]\!\right]_{\mathcal{F}} \subseteq [\![u \vartriangle v]\!]_{\mathcal{F}}$. Thereby $[\![t]\!]_{\mathcal{F}} \subseteq [\![u \vartriangle v]\!]_{\mathcal{F}}$, as required.

Consider the case when $\vdash [\![t]\!]_{\mathcal{LF}} \multimap [\![u \cdot v]\!]_{\mathcal{LF}}$. By definition, $\vdash \overline{[\![t]\!]_{\mathcal{LF}}} \parallel ([\![u]\!]_{\mathcal{LF}}\ ;\ [\![v]\!]_{\mathcal{LF}})$, hence by Lemma 43 there exist $n$-ary killing context $\mathcal{T}\{\ \}$ and propositions $R_1^i$ and $R_2^i$ such that $\vdash \overline{R_1^i} \parallel [\![u]\!]_{\mathcal{LF}}$ and $\vdash \overline{R_2^i} \parallel [\![v]\!]_{\mathcal{LF}}$, for $1 \leq i \leq n$, and $\overline{[\![t]\!]_{\mathcal{LF}}} \longrightarrow \mathcal{T}\left\{ \overline{R_1^1}\ ;\ \overline{R_2^1}, \overline{R_1^2}\ ;\ \overline{R_2^2}, \ldots, \overline{R_1^n}\ ;\ \overline{R_2^n} \right\}$. Hence, by Lemma 40, there exist $w_1^i$ and $w_2^i$ such that $\left[\!\left[w_1^i\right]\!\right]_{\mathcal{LF}} = R_1^i$ and $\left[\!\left[w_2^i\right]\!\right]_{\mathcal{LF}} = R_2^i$, for all $1 \leq i \leq n$, and the following equivalence holds.

$$\mathcal{T}\left\{ \overline{R_1^1}\ ;\ \overline{R_2^1}, \overline{R_1^2}\ ;\ \overline{R_2^2}, \ldots, \overline{R_1^n}\ ;\ \overline{R_2^n} \right\} \equiv \left[\!\left[\left(w_1^1 \cdot w_2^1\right) \triangledown \left(w_1^2 \cdot w_2^2\right) \triangledown \ldots \left(w_1^n \cdot w_2^n\right)\right]\!\right]_{\mathcal{LF}}$$

Furthermore $[\![t]\!]_{\mathcal{F}} \subseteq \left[\!\left[\left(w_1^1 \cdot w_2^1\right) \triangledown \left(w_1^2 \cdot w_2^2\right) \triangledown \ldots \left(w_1^n \cdot w_2^n\right)\right]\!\right]_{\mathcal{F}}$. As the induction hypothesis, assume that $\left[\!\left[w_1^i\right]\!\right]_{\mathcal{F}} \subseteq [\![u]\!]_{\mathcal{F}}$ and $\left[\!\left[w_2^i\right]\!\right]_{\mathcal{F}} \subseteq [\![v]\!]_{\mathcal{F}}$. Thereby we have that $\left[\!\left[w_1^i \cdot w_2^i\right]\!\right]_{\mathcal{F}} \subseteq [\![u \cdot v]\!]_{\mathcal{F}}$, for all $i$; hence the following inclusion of ideals holds: $\left[\!\left[\left(w_1^1 \cdot w_2^1\right) \triangledown \left(w_1^2 \cdot w_2^2\right) \triangledown \ldots \left(w_1^n \cdot w_2^n\right)\right]\!\right]_{\mathcal{F}} \subseteq [\![u \cdot v]\!]_{\mathcal{F}}$. Thereby $[\![t]\!]_{\mathcal{F}} \subseteq [\![u \cdot v]\!]_{\mathcal{F}}$, as required.

Consider the case when $\vdash [\![t]\!]_{\mathcal{LF}} \multimap ([\![u \triangledown v]\!]_{\mathcal{LF}})$. By definition $\vdash \overline{[\![t]\!]_{\mathcal{LF}}} \parallel ([\![u]\!]_{\mathcal{LF}} \oplus [\![v]\!]_{\mathcal{LF}})$ holds hence by Lemma 43 there exist propositions $S_i$ such that either $\vdash S_i \parallel [\![u]\!]_{\mathcal{LF}}$ or $\vdash S_i \parallel [\![v]\!]_{\mathcal{LF}}$, for $1 \leq i \leq n$; and killing context $\mathcal{T}\{\ \}$ such that the following derivation holds: $\overline{[\![t]\!]_{\mathcal{LF}}} \longrightarrow \mathcal{T}\left\{ \overline{S_1}, \overline{S_2}, \ldots, \overline{S_n} \right\}$. Hence, there exist $w_i$ such that $[\![w_i]\!]_{\mathcal{LF}} \equiv S_i$ and $[\![w_1 \triangledown w_2 \triangledown \ldots \triangledown w_n]\!]_{\mathcal{LF}} \equiv \mathcal{T}\left\{ \overline{S_1}, \overline{S_2}, \ldots, \overline{S_n} \right\}$, by Lemma 40; and furthermore $[\![t]\!]_{\mathcal{F}} \subseteq [\![w_1 \triangledown w_2 \triangledown \ldots \triangledown w_n]\!]_{\mathcal{F}}$. As the induction hypothesis, assume that either $[\![w_i]\!]_{\mathcal{F}} \subseteq [\![u]\!]_{\mathcal{F}}$ or $[\![w_i]\!]_{\mathcal{F}} \subseteq [\![v]\!]_{\mathcal{F}}$ hence $[\![w_i]\!]_{\mathcal{F}} \subseteq [\![u]\!]_{\mathcal{F}} \cup [\![v]\!]_{\mathcal{F}} = [\![u \triangledown v]\!]_{\mathcal{F}}$, for all $i$, and thereby $[\![w_1 \triangledown w_2 \triangledown \ldots \triangledown w_n]\!]_{\mathcal{F}} \subseteq [\![u \triangledown v]\!]_{\mathcal{F}}$. From the above $[\![t]\!]_{\mathcal{F}} \subseteq [\![u \triangledown v]\!]_{\mathcal{F}}$ as required.

This covers all constructs, thereby the result follows by induction on the structure of the causal attack tree in the conclusion of the linear implication. □

The soundness result for the logical embedding $[\![\cdot]\!]_{\mathcal{LI}}$ with respect to the ideal semantics is established by a similar technique to the above theorem. Notice that there are no & operators in $\overline{[\![t]\!]_{\mathcal{LI}}}$ and no rule of MAV can introduce & operators during proof search. This simplifies the proof, since splitting is applied with respect to the trivial context with one hole and no & operators.

**Theorem 45.** *For causal attack trees t and u, if* $\vdash [\![t]\!]_{\mathcal{LI}} \multimap [\![u]\!]_{\mathcal{LI}}$ *in* MAV *then* $[\![t]\!]_{\mathcal{I}} \subseteq [\![u]\!]_{\mathcal{I}}$.

*Proof.* Consider the base case when $\vdash [\![t]\!]_{\mathcal{LF}} \multimap \left[\!\!\left[ \boxed{\text{a}} \right]\!\!\right]_{\mathcal{LF}}$, where $a$ is an atomic proposition. By definition, $\vdash \overline{[\![t]\!]_{\mathcal{LF}}} \parallel a$ and observe that there are no & operators in $\overline{[\![t]\!]_{\mathcal{LF}}}$. Hence, by Lemma 43, $\overline{[\![t]\!]_{\mathcal{LF}}} \longrightarrow \overline{a}$. By Lemma 41, $\left[\!\!\left[ \boxed{\text{a}} \right]\!\!\right]_{\mathcal{I}} \subseteq [\![t]\!]_{\mathcal{I}}$, as required.

Consider the case when $\vdash [\![t]\!]_{\mathcal{LF}} \multimap [\![u \vartriangle v]\!]_{\mathcal{LF}}$. By definition, $\vdash \overline{[\![t]\!]_{\mathcal{LF}}} \parallel ([\![u]\!]_{\mathcal{LF}} \otimes [\![v]\!]_{\mathcal{LF}})$, hence by Lemma 43 there exist propositions $R_1$ and $R_2$ such that $\vdash \overline{R_1} \parallel [\![u]\!]_{\mathcal{LF}}$ and $\vdash \overline{R_2} \parallel [\![v]\!]_{\mathcal{LF}}$, for $1 \leq i \leq n$, and the following derivation holds: $\overline{[\![t]\!]_{\mathcal{LF}}} \longrightarrow \overline{R_1} \parallel \overline{R_2}$. Hence, by Lemma 41, there exist $w_1$ and $w_2$ such that $[\![w_1]\!]_{\mathcal{LF}} = R_1$ and $[\![w_2]\!]_{\mathcal{LF}} = R_2$ and $\overline{R_1} \parallel \overline{R_2} \equiv [\![w_1 \vartriangle w_2]\!]_{\mathcal{LF}}$ and furthermore $[\![w_1 \vartriangle w_2]\!]_{\mathcal{I}} \subseteq [\![t]\!]_{\mathcal{I}}$. As the induction hypothesis, assume that $[\![u]\!]_{\mathcal{I}} \subseteq [\![w_1]\!]_{\mathcal{I}}$ and $[\![v]\!]_{\mathcal{I}} \subseteq [\![w_2]\!]_{\mathcal{I}}$. Thereby we have that $[\![u \vartriangle v]\!]_{\mathcal{I}} \subseteq [\![w_1 \vartriangle w_2]\!]_{\mathcal{I}}$. Thereby $[\![u \cdot v]\!]_{\mathcal{I}} \subseteq [\![t]\!]_{\mathcal{I}}$, as required. The case for sequential refinement is similar.

Consider the case when $\vdash [\![t]\!]_{\mathcal{LF}} \multimap [\![u \triangledown v]\!]_{\mathcal{LF}}$. By definition $\vdash \overline{[\![t]\!]_{\mathcal{LF}}} \parallel ([\![u]\!]_{\mathcal{LF}} \,\&\, [\![v]\!]_{\mathcal{LF}})$ holds hence by Lemma 43 there $\vdash \overline{[\![t]\!]_{\mathcal{LF}}} \parallel [\![u]\!]_{\mathcal{LF}}$ or $\vdash \overline{[\![t]\!]_{\mathcal{LF}}} \parallel [\![v]\!]_{\mathcal{LF}}$ holds. As the induction hypothesis, assume that both $[\![u]\!]_{\mathcal{I}} \subseteq [\![t]\!]_{\mathcal{I}}$ and $[\![v]\!]_{\mathcal{I}} \subseteq [\![t]\!]_{\mathcal{I}}$. Thereby $[\![u \triangledown v]\!]_{\mathcal{I}} = [\![u]\!]_{\mathcal{I}} \cup [\![v]\!]_{\mathcal{I}} \subseteq [\![t]\!]_{\mathcal{I}}$, as required.

This covers all constructs, thereby the result follows by induction on the structure of the causal attack tree in the conclusion of the linear implication. □

*Partial distribution of seq over disjunctive refinement.* The two logical systems, defined using embeddings of causal attack trees as propositions in MAV, are subtly finer than their respective graph-based semantics. The logical systems can be regarded as a semantics for causal attack trees, while the graph-based semantics can, by Theorems 44 and 45, be considered to be sound attribute domains for the respective logical semantics.

Although the logical systems are not complete for their corresponding graph-based semantics in this paper, tighter graph-based semantics can be devised. Objectively speaking, MAV is fundamentally consistent, due to cut elimination (Theorem 37); whereas we are unaware of an objective justification for the graph-based semantics in Defs. 23 and 24. Subjectively speaking, an operational interpretation of attack trees, discussed in this section, justifies the partial distributivity properties.

To see why partial distributivity is desirable, consider the following attack:

$$\boxed{\text{descend into vault}} \cdot \left( \boxed{\text{override using laptop}} \triangledown \boxed{\text{smash with hammer}} \right)$$

which suggests that the attacker descends into the vault and then decides whether to access a safe by either using a laptop or a hammer. Contrast the above intuition to the following specialised causal attack tree.

$$\left( \boxed{\text{descend into vault}} \cdot \boxed{\text{override using laptop}} \right) \triangledown \left( \boxed{\text{descend into vault}} \cdot \boxed{\text{smash with hammer}} \right)$$

The causal attack tree above suggests that the attacker commits to packing either a laptop or a hammer, but not necessarily both, when they set off to descend into the vault. The subtlety in these scenarios is, when reading operationally,

the point at which choices are made between disjunctively refined sub-goals affect the success of the attack. The reader familiar with branching time equivalences, such as bisimulation, will observe a similarity in reasoning. Observe that the above argument works both backwards and forwards along the causal link indicated by the *sequential* operator.

## 6. Future Work

We have demonstrated that the logical system MAV can be used to provide a semantics for specialising causal attack trees. This observation opens up several avenues of investigation regarding the semantics of variants of attack trees using fragments and extensions of MAV.

Attack-defense trees not only capture the goals of attackers, but also countermeasures to defend against an attack, and possibly further counterattacks. Attack-defense trees have been provided with a multiset semantics [16], that, as for attack trees, coincides with the positive propositions in linear logic. Thus the multiset semantics for attack-defense trees can be captured using linear logic. However, attack-defense trees can be equipped with a wide range of attribute domains [11], some of which could benefit from a different fragment of linear logic than the positive fragment only. We envision a thorough analysis of the relationship between attack-defense trees with various assignments of linear logic operators to nodes, and attributes that are compatible with such assignments. Related work [17], exploring attack-defense trees with sequential refinement operators, would be suited to a semantics using MAV.

The study of the system MAV was instigated when investigating *provenance diagrams* with a "was derived from" relation [14, 18]. The "was derived from" relation in provenance diagrams indicates "artefacts" that were used to produce another artefact. For example, in a database transaction, a value written may depend on other values that were read. The "was derived from" relation should respect causality, since only artefacts that contributed to the creation of an artefact should be indicated, rather than recording a snapshot of the state of the entire system. The semantics for causal attack trees in this work are also relevant as semantics for provenance diagrams. Both "why" provenance and "how" provenance [19] can be captured, where "why" provenance is given by "was derived from" corresponding to sequential refinement, and "how" provenance is given by provenance semirings [20] captured by conjunctive and disjunctive refinement. Attack trees and provenance diagrams are closely related, since both profile processes. Future work includes establishing the common semantic foundations and applications for attack trees and provenance diagrams.

## 7. Conclusion

This work introduces two semantics for causal attack trees, which are attack trees extended with sequential refinement. The limitation of semantics previously proposed based on sets of series-parallel graphs is that, while equality of causal attack trees can be assessed, such a semantics is not suitable for defining a preorder over causal attack trees. A preorder is useful for scenarios where two trees are not equivalent, but one is a *specialisation* of the other. This work identifies the notion of *specialising* attack trees as being central to attack tree methodology: attack trees evolve as more knowledge is gathered; or are pruned for more specific scenarios.

More than one semantics for specialising causal attack trees is required since different attribute domains can give rise to distinct algebraic properties. The semantics proposed in this paper are referred to as the *ideal semantics* and *filter semantics* for specialisation. Both semantics proposed in this paper close sets of series-parallel graphs using "smoothing" homomorphisms, Def. 16, that permit causal dependencies to be strengthened. Both semantics have a sound and complete axiomatisation, given by Theorems 25, 26 and 29. The first theorem is an established result for pomsets, while the later two are technical contributions of this paper.

For both semantics a logical system is introduced. Each logical system is shown to be sound with respect to the corresponding semantics, by Theorems 44 and 45. The definitions, algebra and logical systems corresponding to each of the two semantics are summarised in the table below.

| name of class | semantics | sound and complete algebra | sound logical system |
|---|---|---|---|
| ideal semantics | $[\![]\!]_{\mathcal{I}}$ Def. 23 | $\preceq_{\mathcal{I}}$ Thm. 25 | $[\![]\!]_{\mathcal{LI}}$ Def. 38 Thm. 45 |
| filter semantics | $[\![]\!]_{\mathcal{F}}$ Def. 24 | $\preceq_{\mathcal{F}}$ Thms. 26 and 29 | $[\![]\!]_{\mathcal{LF}}$ Def. 39 Thm. 44 |

The above results are key for recommending the methodology of specialising attack trees. Given two causal attack trees, a tool can be used to embed the trees in the logical system and thereby determine whether they are related. For example, in the case of the ideal semantics, for causal attack trees $t$ and $u$, if $\vdash [\![t]\!]_{\mathcal{LI}} \multimap [\![u]\!]_{\mathcal{LI}}$ holds, then by

Theorem 45 and Theorem 25, both $t \preceq_I u$ and $[\![t]\!]_I \subseteq [\![u]\!]_I$ hold. Thus $t$ is a sound specialisation of $u$ for attribute domains respecting the ideal semantics. For instance, $t$ is a sound specialisation of $u$ for both the attribute domains "minimum attack time" and "guards to counter all attacks".

Examples of attribute domains that are sound for each semantics are presented in the table below.

| ideal semantics | filter semantics |
|---|---|
| "minimum attack time" Prop. 32 | "minimum number of experts required" Prop. 31 |
| "guards to counter attacks" Prop. 33 | "time to make attacks possible" Prop. 34 |

The above list of attribute domains respecting the ideal and filter semantics is far from being exhaustive. Note, for some useful attribute domains, such as those involving probabilities, finer algebraic properties hold, emphasising the need for a sound methodology for specialising attack trees.

Both logical systems are defined using fragments of MAV, which extends linear logic with a non-commutative operator modelling causal dependencies. Thus the results summarised in the tables above are natural extensions of observations regarding the multiset semantics for attack trees (Definition 1), linear logic (Theorem 7) and commutative idempotent semirings (Proposition 12). All semantics in this work permit action refinement (Proposition 35). Action refinement is central to the attack tree methodology, permitting a basic action to be replaced with a tree of sub-goals.

## Acknowledgment

Horne and Tiu receive support from MOE Tier 2 grant MOE2014-T2-2-076. Tiu receives support from the National Research Foundation Singapore under its National Cybersecurity R&D Program (Award No. NRF2014NCR-NCR001-30). Mauw received funding from the Fonds National de la Recherche Luxembourg, grant C11/IS/1183245 (ADT2P), and the European Commissions Seventh Framework Programme (FP7/2007-2013) under grant agreement number 318003 (TREsPASS).

## References

[1] Schneier B. Attack trees. Dr Dobb's journal. 1999;24(12):21–29.
[2] Jhawar R, Kordy B, Mauw S, Radomirović S, Trujillo-Rasua R. Attack trees with sequential conjunction. In: Proc. IFIPSec'15. vol. 455 of IFIP AICT; 2015. p. 339–353.
[3] Mauw S, Oostdijk M. Foundations of attack trees. In: Proc. ICISC'05. vol. 3935 of LNCS; 2006. p. 186–198.
[4] Rehák M, et al. Runtime Monitoring and Dynamic Reconfiguration for Intrusion Detection Systems. In: Proc. RAID'09. vol. 5758 of LNCS; 2009. p. 61–80.
[5] Kordy B, Kordy P, Mauw S, Schweitzer P. ADTool: Security Analysis with Attack-Defense Trees. In: Proc. QEST'13. vol. 8054 of LNCS; 2013. p. 173–176.
[6] Girard JY. Linear logic. Theoretical computer science. 1987;50(1):1–101.
[7] Guglielmi A. A system of interaction and structure. ACM Transactions on Computational Logic. 2007;8(1):1.
[8] Horne R. The Consistency and Complexity of Multiplicative Additive System Virtual. Scientific Annals of Computer Science. 2015;25(2):245.
[9] Jürgenson A, Willemson J. Serial Model for Attack Tree Computations. In: Proc. ICISC'09. vol. 5984 of LNCS; 2010. p. 118–128.
[10] Andreoli JM. Logic programming with focusing proofs in linear logic. Journal of Logic and Computation. 1992;2(3):297–347.
[11] Kordy B, Mauw S, Schweitzer P. Quantitative questions on Attack-Defense Trees. In: Proc. ICISC'12. vol. 7839 of LNCS; 2013. p. 49–64.
[12] Valdes J, Tarjan RE, Lawler EL. The recognition of series parallel digraphs. In: Proc. STOC'79. ACM; 1979. p. 1–12.
[13] Gischer JL. The equational theory of pomsets. Theor Comput Sci. 1988;61(2):199–224.
[14] Ciobanu G, Horne R. A provenance tracking model for data updates. In: Proc. FOCLASA'12. vol. 91 of EPTCS; 2012. p. 31–44.
[15] Van Glabbeek R, Goltz U. Refinement of actions and equivalence notions for concurrent systems. Acta Informatica. 2001;37(4-5):229–327.
[16] Kordy B, Mauw S, Radomirović S, Schweitzer P. Attack-Defense trees. Journal of Logic and Computation. 2014;24(1):55–87.
[17] Aslanyan Z, Nielson F, Parker D. Quantitative Verification and Synthesis of Attack-Defence Scenarios. In: CSF'16. IEEE; 2016. p. 105–119.
[18] Cheney J, Ahmed A, Acar UA. Provenance as dependency analysis. Mathematical Structures in Computer Science. 2011;21(6):1301–1337.
[19] Cheney J, Chiticariu L, Tan WC. Provenance in Databases: Why, How, and Where. Foundations and Trends in Databases. 2009;1(4):379–474.
[20] Green TJ, Karvounarakis G, Tannen V. Provenance semirings. In: Proc. PODS'07. ACM; 2007. p. 31–40.