A Formal Approach to Analyzing Cyber-Forensics Evidence

Erisa Karafili[1], Matteo Cristani[2], and Luca Viganò[3]

[1] Department of Computing, Imperial College London, UK
e.karafili@imperial.ac.uk
[2] Dipartimento di Informatica, Università di Verona, Italy
matteo.cristani@univr.it
[3] Department of Informatics, King's College London, UK
luca.vigano@kcl.ac.uk

Abstract. The frequency and harmfulness of cyber-attacks are increasing every day, and with them also the amount of data that the cyber-forensics analysts need to collect and analyze. In this paper, we propose a formal analysis process that allows an analyst to filter the enormous amount of evidence collected and either identify crucial information about the attack (e.g., when it occurred, its culprit, its target) or, at the very least, perform a pre-analysis to reduce the complexity of the problem in order to then draw conclusions more swiftly and efficiently. We introduce the Evidence Logic $\mathcal{EL}$ for representing simple and derived pieces of evidence from different sources. We propose a procedure, based on monotonic reasoning, that rewrites the pieces of evidence with the use of tableau rules, based on relations of trust between sources and the reasoning behind the derived evidence, and yields a consistent set of pieces of evidence. As proof of concept, we apply our analysis process to a concrete cyber-forensics case study.

1 Introduction

The frequency and harmfulness of cyber-attacks are increasing every day, and with them also the amount of data that cyber-forensics analysts need to collect and analyze. In fact, forensics investigations often produce an enormous amount of evidence. The pieces of evidence are produced/collected by various sources, which can be humans (e.g., another analyst) or forensic tools such as intrusion detection system (IDS), traceback systems, malware analysis tools, and so on.

When a forensics analyst evaluates a cyber-attack, she first collects all the relevant evidence containing information about the attack and then checks the sources of the evidence in order to evaluate their reliability and to resolve possible inconsistencies arising from them. Based on the collected information, which might be different depending on the information sources and the trust relation between the analyst and the sources, the analyst might reconstruct different, possibly faulty, courses of events. State of the art approaches don't really manage to cope well with such situations.

To reason about the collected evidence, we need to formalize the fact that the analyst trusts more some sources than others for particular pieces of evidence, e.g., a source S_1 is more trusted than another source S_2 for attack similarity as tool S_1 specializes in malware analysis whereas tool S_2 specializes in deleted data. We also need to distinguish between the evidence and its interpretation that an analyst may consider in order to perform a correct analysis and attribution of the cyber-attack.

Our main contribution in this paper is the introduction of the *Evidence Logic* $\mathcal{EL}$, which allows an analysts to represent the different pieces of evidence, together with their sources and relations of trust, and reason about them by eliminating the conflicting pieces of evidence during the analysis process.

As a concrete motivating example, consider the data breach of the Democratic National Committee (DNC) network, during the last US presidential campaign, when Wikileaks and other websites published several private emails in October and November 2016. DNC used the services of a cyber-security company, CrowdStrike, to mitigate the attacks and to conduct a forensics investigation. CrowdStrike stated that the main attack occurred between March and April 2016, and identified it as a *spear phishing* campaign that used Bitly accounts to shorten malicious URLs. The phishing campaign was successful as different IDs and passwords were collected.

However, TheForensicator, an anonymous analyst, stated that the attack actually occurred on the 5th of July 2016, not in March/April, as the metadata released by an alleged attacker were created on the 5th of July 2016, and the data-leak occurred physically as the data were transferred at the speed of around 23 MB/s, and this speed is possible only during a physical access. Another cyber-security company, FireEye, stated that it is possible to have a non-physical data-transfer speed of 23 MB/s. What should an analyst conclude from these discording statements and pieces of evidence? How can a decision be made?

$\mathcal{EL}$ is able to deal with this type of discordances, and based on relations of trust on the sources and reasonings, to arrive at a certain conclusion. $\mathcal{EL}$ is composed of three separate layers: the first layer $\mathcal{EL}_E$ deals with pieces of evidence, the second layer $\mathcal{EL}_I$ focuses on the evidence interpretations, and the third layer $\mathcal{EL}_R$ focuses on the reasoning involved in the evidence. Reasoning with $\mathcal{EL}$ amounts to applying a rewriting system that spans formulas in all three levels to reach a conclusion, ruling out discordances and inconsistencies. Applying the reasoning process of $\mathcal{EL}$ to the different pieces of evidence from the various sources, the analyst can decide the type of the attack and when it occurred. For instance, regarding the speed of transferability, if the analyst trusts FireEye more than TheForensicator, then she does not take into consideration the evidence that the data transfer was physical. Hence, she concludes that the attack occurred during March/April 2016 and not in July 2016.

We proceed as follows. In Section 2 and Section 3, we give the syntax and semantics of the Evidence Logic $\mathcal{EL}$, respectively. In Section 4, we introduce the rules of the rewriting system of $\mathcal{EL}$ and we give a concrete procedure that uses the rewriting rules to prove the satisfiability of a given $\mathcal{EL}$-theory (which

is a finite and non-empty set of formulas of the three layers of $\mathcal{EL}$). We prove the rewriting system to be sound and the procedure to be correct (the proofs of the theorems are given in the Appendix). Section 6 concludes the paper by discussing related work and future work.

2 The Syntax of the Evidence Logic $\mathcal{EL}$

The *Evidence Logic* $\mathcal{EL}$ that we propose enables a cyber-forensics analyst to represent the various plausible pieces of evidence that she collected from different sources and reason about them. To that end, the analyst should distinguish between the evidence and its interpretation. In a nutshell:

- *evidence* represents information related to the attack, where a given (piece of) evidence usually represents an event, its occurrence and the source of the information of the occurrence of the event (which can be another analyst, a cyber-forensics tool, etc.);
- *evidence interpretation* represents what the analyst thinks[4] about the occurrence of an event e and about the occurrences of the events causing e.

$\mathcal{EL}$ contains two types of well-formed formulas: *labeled formulas*, to formalize the pieces of evidence and interpretation, and *relational formulas*, to formalize relations of *trust* between sources of evidence and their reasonings. $\mathcal{EL}$ also contains a rewriting system (composed of a set of tableau rules) to build the analyst's interpretations from forensics evidence. For the sake of readability, we omit to model explicitly the analyst who is reconstructing and attributing the cyber-attack, but we simply silently assume her existence.

$\mathcal{EL}$ is composed of three separate layers: the first layer $\mathcal{EL}_E$ shows how the well-formed formulas for pieces of evidence are built, the second layer $\mathcal{EL}_I$ focuses on the evidence interpretations, and the third layer $\mathcal{EL}_R$ focuses on the reasoning involved in the evidence. In the following, we discuss each of the layers in detail.

2.1 $\mathcal{EL}_E$: Evidence

Definition 1. *Given* $t, t_1, \ldots t_n \in T$, $a, a_1, \ldots a_n \in Ag$, $r_1, r_2 \in \mathcal{R}$, $p \in Vars_S$ *and* $\phi, \phi_1, \ldots, \phi_n \in Lit$, *the set* ρ *of formulas of* $\mathcal{EL}_E$ *is*

$$\rho ::= a : (t : \phi) \;\big|\; a : (t : \phi)\ [a_1 : (t_1 : \phi_1) \mid \ldots \mid a_n : (t_n : \phi_n)]_r \;\big|\; a_1 \lhd_p a_2 \;\big|\; r_1 \prec r_2$$

We introduce all these notions, and the four kinds of formulas, step by step. A piece of evidence asserts what a source thinks about the temporal occurrence of an event, i.e., whether an event occurred or not in a particular instance of time. To formalize this, we use two finite[5] and disjoint sets of labels,

[4] We deliberately use the verb "thinking" to avoid suggesting any epistemic or doxastic flavor, as in $\mathcal{EL}$ we do not consider the modalities of knowledge or belief.

[5] In principle, there is nothing in our logic that prevents us from considering countable, possibly infinite, sets of labels, but here we consider finite sets for simplicity.

– *source labels* $Ag = \{a_1, a_2, \ldots, a_n\}$ for forensic sources, which we call *agents*, regardless of whether they are human or not, and
– *temporal labels* $T = \{t_1, t_2, \ldots, t_m\}$ for instants of time,

along with

– a set of *propositional variables* $Vars = \{p_1, p_2, \ldots, p_n\}$ that represent the occurrences of *forensics events* (so that p represents the occurrence of an event and $\neg p$ represents that p does not occur),
– a set of *reasoning rules* (or simply *reasonings*) $\mathcal{R} = \{r_1, r_2, \ldots, r_l\}$ that represent the reasoning used by the agents to conclude further evidence.

The set of *literals* $Lit = \{p_1, \neg p_1, \ldots, p_n, \neg p_n\}$ consists of each propositional variable and its negation. We write ϕ, possibly subscripted, to denote a literal.

Instants of time are labels associated to elements of a single given stream. Thus, the labels that represent the instants of time cannot be processed for consistency, and no assertions regarding relations between them is allowed.

Example 1. Consider again the motivating example that we discussed in the Introduction. The set of agents is composed of the analyst (whose existence we silently assume) and the sources CrowdStrike (CS), TheForensicator (TF) and FireEye (FE); thus, $Ag = \{CS, TF, FE\}$. The sources make statements about events occurring in two instants of time: "March/April 2016" and "5th of July 2016" represented respectively by t_1 and t_2. □

We formalize two different types of evidence: simple and derived one. The *simple evidence* is a labeled formula of the form

$$a : (t : \phi)\,,$$

expressing that the agent represented by the source label a thinks that the literal ϕ is true at the instant of time represented by the temporal label t. For short, we will say that a thinks that ϕ is true at t.

Example 2. The simple evidence $FE : (t_2 : SpeedTr(23MB/s))$ expresses that FE states that the non-physical transferability speed, $SpeedTr$, can be $23MB/s$ at t_2. □

The *derived evidence* is a labeled formula of the form

$$a : (t : \phi)\ [a_1 : (t_1 : \phi_1) \mid a_2 : (t_2 : \phi_2) \mid \ldots \mid a_n : (t_n : \phi_n)]_r\,,$$

expressing that agent a thinks that ϕ is true at instant of time t *because* of reasoning r, where a_1 thinks that ϕ_1 is true at t_1, … and a_n thinks that ϕ_n is true at t_n. In other words, based on r, a thinks that ϕ is *caused*[6] by $\phi_1, \cdots, \phi_n$

[6] We use the term "cause" to describe the events that an agent thinks were the preconditions for a certain derived evidence. In this work, we will not focus on the causality relationships between events.

(with their respective time instants and agents). The reasoning r of the derived evidence $a : (t : \phi)$ is composed of simple and/or derived pieces of evidence. We include a constraint in our syntax that does not permit cycles between derived pieces of evidence, so that if $a_i : (t_i : \phi_i)\ [\cdots \mid a_j : (t_j : \phi_j) \mid \ldots]_r$, then we do not accept in our language the formula $a_j : (t_j : \phi_j)\ [\cdots \mid a_i : (t_i : \phi_i) \mid \ldots]_{r'}$.

A reasoning r can be used by different agents to arrive at the same conclusion (derived evidence), using the same pieces of evidence. An agent can use different reasonings $r_i, \cdots, r_j$ to conclude the same derived evidence, where the pieces of evidence used by the reasonings are different from one reasoning to the other.

Example 3. *CS* says that the *Attack* occurred at time t_1, based on reasoning r_1 and *CS*'s evidence about a spear phishing campaign *SpPhish* and its success *SucPhish* at t_1. The latter is based on r_2 and *CS*'s pieces of evidence that in t_1: the malicious link was clicked *LinkCl*, the malicious form was filled *FFill*, and the data were stolen *DStolen*. We thus have:

$$CS : (t_1 : Attack)\ [CS : (t_1 : SpPhish) \mid CS : (t_1 : SucPhish)]_{r_1}$$
$$CS : (t_1 : SucPhish)\ [CS : (t_1 : LinkCl) \mid CS : (t_1 : FFill) \mid CS : (t_1 : DStolen)]_{r_2}$$

Instead, *TF* says that based on r_3 the *Attack* occurred at t_2 because the metadata *MetaC* were created at t_2 and the access was physical *PhysA*. The latter is true because *TF* states that it is not true that *SpeedTr* is $23MB/s$:

$$TF : (t_2 : Attack)\ [TF : (t_2 : MetaC) \mid TF : (t_2 : PhysA)]_{r_3}$$
$$TF : (t_2 : PhysA)\ [TF : (t_2 : \neg SpeedTr(23MB/s))]_{r_4}$$

□

To allow an analyst to distinguish the events that can be expressed by a simple or derived evidence, the set of propositional variables *Vars* is composed by two disjoint subsets $Vars_S$ and $Vars_D$ that respectively represent the events that can be part of simple and derived evidence, i.e., $Vars = Vars_S \cup Vars_D$ with $Vars_S \cap Vars_D = \emptyset$. By extension, we write $\phi \in Lit_S$ if ϕ is p or $\neg p$ with $p \in Vars_S$, and $\phi \in Lit_D$ if ϕ is p or $\neg p$ with $p \in Vars_D$.

Hence, if $\phi \in Lit_S$, then $a : (t : \phi)$ is a simple evidence, whereas if $\phi \in Lit_D$ and $\phi_i \in Lit$ for $i \in \{1, \ldots, n\}$, then $a : (t : \phi)\ [a_1 : (t_1 : \phi_1) \mid \ldots \mid a_n : (t_n : \phi_n)]_r$ is a derived evidence. For simplicity, we will assume that a variable that represents an event given by a simple evidence is part of $Vars_S$ and that a variable that represents an event given by a derived evidence is part of $Vars_D$.

Example 4. The variables of the events of our example are divided in the two following disjoint subsets: $Vars_S = \{SpPhish, LinkCl, FFill, DStolen, MetaC, SpeedTr(23MB/s)\}$ and $Vars_D = \{Attack, SucPhish, PhysA\}$. □

The temporal labels can have *temporal constraints* such as $t_1 \leq t$ or $t_n < t$. As we consider time to be *linear* and every instant of time is mapped to only one element of the natural numbers, our syntax doesn't need to include a precedence relation, as it represents the classical precedence relation between natural numbers.

In addition to ordering events with respect to time, the analyst can consider the *trust(worthiness)* relations that she has with the sources with respect to their assertions in the simple evidence, i.e., she might think that one source is more reliable than another one with respect to a particular event (and its negation). For instance, a_i might be more trustworthy than a_j with respect to an event p (and thus also $\neg p$), where $p \in \mathit{Vars}_S$. In general, if there exists a trust relation between two agents $a_i, a_j \in \mathit{Ag}$ for an event $p \in \mathit{Vars}_S$, then we have that either a_i is more trustworthy than a_j with respect to p, or a_j is more trustworthy than a_i with respect to p. We formalize this by introducing the *trust relation* $\lhd : \mathit{Ag} \times \mathit{Ag} \times \mathit{Vars}_S$. Then, the *relational formula* $a_i \lhd_p a_j$ expresses that a_j is more trustworthy than a_i with respect to p.

Example 5. We write $TF \lhd_{\mathit{SpeedTr}(23MB/s)} FE$ to formalize that the analyst trusts FE more than TF w.r.t. the simple evidence $\mathit{SpeedTr}(23MB/s)$. □

The analyst can also consider the trust(worthiness) relations about the reasonings she used. In particular, given two conflicting derived pieces of evidence that use two different reasonings, the analyst can consider one reasoning to be more trustworthy than the other one. We formalize this by introducing the *trust relation* $\prec$: $\mathcal{R} \times \mathcal{R}$. Then, the *relational formula* $r_i \prec r_j$ expresses that reasoning r_j is more trustworthy than reasoning r_i.

2.2 $\mathcal{EL}_I$: Evidence Interpretation

An *evidence interpretation* (or simply *interpretation*) is what the cyber-forensics analyst thinks that is plausibly true. To formalize this, the second level $\mathcal{EL}_I$ of $\mathcal{EL}$ employs a simplified variant of *Linear Temporal Logic (LTL)*. $\mathcal{EL}_I$ inherits from $\mathcal{EL}_E$ the temporal labels T, the reasonings $\mathcal{R}$ and the propositional variables *Vars* (and thus also the literals *Lit*).

Definition 2. *Given* $t, t_1, \ldots t_n \in T$, $\phi, \phi_1, \ldots, \phi_n \in \mathit{Lit}$, $r \in \mathcal{R}$ *and* $\phi' \in \mathit{Lit}_D$, *the set* φ *of formulas of* $\mathcal{EL}_I$, *called* interpretations, *is*

$$\varphi ::= \; t : \phi \mid t_1 : \phi_1 \wedge t_2 : \phi_2 \wedge \ldots \wedge t_n : \phi_n \rightarrow_r t : \phi'$$

$t : \phi$ means that the analyst thinks that ϕ is true at t, whereas $t_1 : \phi_1 \wedge \ldots \wedge t_n : \phi_n \rightarrow_r t : \phi'$ means that the analyst thinks that ϕ' is true at the instant of time t, based on reasoning r, if ϕ_i is true at t_i for all $i \in \{1, \ldots, n\}$. An interpretation expresses a positive event (the occurrence of an event, e.g., $t : p$) or a negative event (the non occurrence of an event, e.g., $t : \neg p$). The interpretations of the temporalized logic $\mathcal{EL}_I$ that express positive events represent the *plausible pieces of evidence* and help the analyst to perform a correct analysis.

2.3 $\mathcal{EL}_R$: Evidence Reasoning

The third layer $\mathcal{EL}_R$ of $\mathcal{EL}$ is the *reasoning layer* and deals with the reasoning behind the derived evidence. Also $\mathcal{EL}_R$ uses LTL and inherits from $\mathcal{EL}_E$ the temporal labels T, the reasonings $\mathcal{R}$ and the propositional variables *Vars*.

Definition 3. *Given $t \in T$, $\phi \in Lit_D$ and $r, r_k, \ldots, r_l \in \mathcal{R}$, the set ψ of formulas of $\mathcal{EL}_R$ is*

$$\psi ::= \ (t : \phi)_r \mid (t : \phi)_{r,r_k,\ldots,r_l}.$$

The *reasoning* involves only derived pieces of evidence, which we can divide in two types. The *first type of derived evidence*, $(t : \phi)_r$, is composed of only simple pieces of evidence; in this case, the only reasoning is the one made by the agent that states the derived evidence $a : (t : \phi)\ [a_1 : (t_1 : \phi_1) \mid \ldots \mid a_j : (t_j : \phi_j)]_r$, where $\phi_i \in Lit_S$ for $i \in \{1, \ldots, j\}$. The *second type of derived evidence*, $(t : \phi)_{r,r_k,\cdots,r_l}$, is composed of simple and derived pieces of evidence; in this case, the reasoning involves the one of the agent stating the derived evidence, $a : (t : \phi)\ [a_1 : (t_1 : \phi_1) \mid \ldots \mid a_j : (t_j : \phi_j)]_r$, as well as all the reasonings involved in the derived pieces of evidence $\phi_i \in Lit$ for $i \in \{1, \ldots, j\}$ that are part of reasoning r. The first type is clearly a special case of the second one, but we keep both for the sake of understandability.

3 The Semantics of the Evidence Logic $\mathcal{EL}$

Definition 4. *The* plausible pieces of evidence *are a finite stream of temporal instants in which at every instant of time we may associate a finite number of occurrences or not occurrences of an event.*

The agents are associated to a given finite set of values, and the trust relationship between agents is interpreted as a partial order on the agents. The same applies to the reasonings: they are associated to a finite set of values and the trust relationship between them is interpreted as a partial order on the reasonings.

Definition 5. *A model of the evidence language $\mathcal{EL}$ is a tuple*

$$\mathfrak{M} = \{Ag^{\mathfrak{I}}, \mathcal{F}^{\mathfrak{I}}, \mathcal{PO}^{\mathfrak{I}}, \mathcal{TR}^{\mathfrak{I}}, Vars^{\mathfrak{I}}, \mathcal{R}^{\mathfrak{I}}, \mathfrak{I}\}$$

where:

- $\mathfrak{I}$ *is the interpretation function, where we interpret time as natural numbers, i.e., $t^{\mathfrak{I}} \in \mathbb{N}$ for every $t \in T$;*
- $Ag^{\mathfrak{I}} = \{{a_1}^{\mathfrak{I}}, \ldots {a_n}^{\mathfrak{I}}\} = \{a_1, \ldots a_n\} = Ag$ *is a set of agents;*
- $\mathcal{F}^{\mathfrak{I}}$ *is a function that maps pairs of instants of time and formulas to True or False (this mapping is used in the second layer of $\mathcal{EL}$, where we have $t : \phi$);*
- $\mathcal{PO}^{\mathfrak{I}} = \{{\lhd_{p_i}}^{\mathfrak{I}}\}$ *is a set of trust relationships between agents, where for every $p \in Vars_S$, if ${\lhd_p}^{\mathfrak{I}} \in \mathcal{PO}^{\mathfrak{I}}$, then ${\lhd_p}^{\mathfrak{I}} = \{({a_i}^{\mathfrak{I}}, {a_j}^{\mathfrak{I}}) \mid a_i \lhd_p a_j\}^*$, where $*$ is the transitive closure of $\lhd$;*
- $\mathcal{TR}^{\mathfrak{I}} = \{\prec^{\mathfrak{I}}\}$ *is a set of trust relationship between reasonings, where for every $r \in \mathcal{R}$, if $\prec^{\mathfrak{I}} \in \mathcal{TR}^{\mathfrak{I}}$, then $\prec^{\mathfrak{I}} = \{({r_i}^{\mathfrak{I}}, {r_j}^{\mathfrak{I}}) \mid r_i \prec r_j\}^*$, where $*$ is the transitive closure of $\prec$;*
- $Vars^{\mathfrak{I}} = Vars = \{p_1, \cdots, p_n\}$ *is a set of events;*
- $\mathcal{R}^{\mathfrak{I}} = \mathcal{R} = \{r_1, r_2, \cdots, r_n\}$ *is a set of reasoning rules.*

Slightly abusing notation, we use $Ag^{\mathfrak{I}}$ to denote also a set of functions, each function ${a_i}^{\mathfrak{I}} : \mathbb{N} \times Lit \rightarrow \{True, False\}$ associating to an instant of time t a set of formulas that are true at t, where ${a_i}^{\mathfrak{I}}(t, p) = True$ when $a_i : (t : p)$ is asserted, ${a_i}^{\mathfrak{I}}(t, p) = False$ when $a_i : (t : \neg p)$ is asserted, ${a_i}^{\mathfrak{I}}(t, \neg p) = True$ when $a_i : (t : \neg p)$ is asserted, ${a_i}^{\mathfrak{I}}(t, \neg p) = False$ when $a_i : (t : p)$ is asserted. The same applies to $\mathcal{R}^{\mathfrak{I}}$, each function ${r_i}^{\mathfrak{I}} : \mathbb{N} \times Lit \rightarrow \{True, False\}$ such that $(t, p)_{{r_i}^{\mathfrak{I}}} = True$ when $(t : p)_{r_i}$ is asserted, $(t, p)_{{r_i}^{\mathfrak{I}}} = False$ when $(t : \neg p)_{r_i}$ is asserted, $(t, \neg p)_{{r_i}^{\mathfrak{I}}} = True$ when $(t : \neg p)_{r_i}$ is asserted, $(t, \neg p)_{{r_i}^{\mathfrak{I}}} = False$ when $(t : p)_{r_i}$ is asserted. Thus, ${a_i}^{\mathfrak{I}}$ and $\mathcal{F}^{\mathfrak{I}}$ both associate to every t a set of formulas that are true at t; the difference is that we use the ${a_i}^{\mathfrak{I}}$ in the evidence layer $\mathcal{EL}_E$ and $\mathcal{F}^{\mathfrak{I}}$ in the interpretation layer $\mathcal{EL}_I$.

In order to avoid having clear contradictions in the models, we constrain the functions $Ag^{\mathfrak{I}}$ and $\mathcal{R}^{\mathfrak{I}}$ as follows:

($COND_1$): If $a^{\mathfrak{I}}(t, p) = True$, then $a^{\mathfrak{I}}(t', p) = False$ for all $t' \neq t$.
($COND_2$): If $(t, p)_{r^{\mathfrak{I}}} = True$, then $(t', p)_{r^{\mathfrak{I}}} = False$ for all $t' \neq t$.
($COND_3$): Every ${\lhd_p}^{\mathfrak{I}}$ is an irreflexive and antisymmetric relation.
($COND_4$): Every $\prec^{\mathfrak{I}}$ is an irreflexive and antisymmetric relation.

A *$\mathcal{EL}$-theory* is built by using $\mathcal{EL}$ to express a finite and non-empty set of formulas of the three layers, including the trust relationships.

4 The Rewriting System of the Evidence Logic $\mathcal{EL}$

In this section, we introduce the *rewriting system* of $\mathcal{EL}$, which, as proved in Theorem 1, is sound. Given pieces of evidence, the rewriting system yields a consistent set of pieces of evidence by translating pieces of evidence into interpretations and reasonings, and resolving their discordances. In particular, the rewriting system uses the tableau rules in Table 1 and applies them via the procedure in Algorithm 1: given a $\mathcal{EL}$-theory $\mathcal{E}$, which is a non-empty set of formulas, the rewriting system generates a new set of formulas $\widehat{\mathcal{E}}$ that replaces $\mathcal{E}$, where the single rewritings correspond to interpretations and reasonings of the theory. More specifically, the rewriting system takes a $\mathcal{EL}$-theory of the first level and rewrites it into a $\mathcal{EL}$-theory of the second and third level, until all the formulas are interpreted, by adding formulas to the theory or eliminating formulas from the theory, with the use of *insertion* or *elimination rules*.

The rules in Table 1 have as premises (above the line) a set of formulas and a $\mathcal{EL}$-theory $\mathcal{E}$, although we don't show $\mathcal{E}$ for readability, and as conclusion (below the line) $\mathcal{E} \cup \{\phi\}$ or $\mathcal{E} \setminus \{\phi\}$, depending on whether the rule is an insertion or elimination rule that respectively inserts or eliminates ϕ. The *insertion* rule introduce formulas for resolving temporal discordances and interpreting pieces of evidence. The *elimination* rules resolve discordances of event occurrences by deleting formulas based on the trust relations among agents and reasonings. The *closure rules* are part of the elimination rules, and discover discordances in $\mathcal{E}$ that cannot be solved, eliminate all the formulas of the set $\mathcal{E}$ and give as result the empty set $\perp$.

Table 1. Rules of the rewriting system of $\mathcal{EL}$

$$\frac{a : (t : \phi)}{\mathcal{E} \cup \{t : \phi\}}\ \mathcal{L}_1 \qquad \frac{(t : \phi)_{r,\cdots,r_n}}{\mathcal{E} \cup \{t : \phi\}}\ \mathcal{L}'_1$$

$$\frac{a : (t : \phi)\ [a_1 : (t_1 : \phi_1) \mid \cdots \mid a_n : (t_n : \phi_n)]_r}{\mathcal{E} \cup \{a_i : (t_i : \phi_i)\}_{\forall i \in \{1,\cdots,n\}\ \phi_i \in Lit_S} \cup \{t_1 : \phi_1 \wedge \cdots \wedge t_n : \phi_n \rightarrow_r t : \phi\}}\ \mathcal{L}_2$$

$$\frac{t_1 : \phi_1 \wedge \cdots \wedge t_n : \phi_n \rightarrow_r t : \phi \quad t_1 : \phi_1 \quad \cdots \quad t_n : \phi_n}{\mathcal{E} \cup \{(t : \phi)_r\}}\ (\rightarrow)$$

$$\frac{t_1 : \phi_1 \wedge \cdots \wedge t_n : \phi_n \rightarrow_r t : \phi \quad (t_1 : \phi_1)_{r_1/\emptyset} \quad \cdots \quad (t_n : \phi_n)_{r_n/\emptyset}}{\mathcal{E} \cup \{(t : \phi)_{r,r_1/\emptyset,\cdots,r_n/\emptyset}\}}\ (\rightarrow')$$

$$\frac{a_1 \lhd_p a_2 \quad a_2 \lhd_p a_3}{\mathcal{E} \cup \{a_1 \lhd_p a_3\}}\ \textsc{Trans}\lhd \qquad \frac{r_1 \prec r_2 \quad r_2 \prec r_3}{\mathcal{E} \cup \{r_1 \prec r_3\}}\ \textsc{Trans}\prec$$

$$\frac{a_1 : (t_1 : \phi) \quad a_2 : (t_2 : \phi)}{\mathcal{E} \cup \{a_1 : (t_2 : \neg\phi), a_2 : (t_1 : \neg\phi)\}}\ \mathcal{D}_1[*] \qquad \frac{(t_1 : \phi)_{r_1} \quad (t_2 : \phi)_{r_2}}{\mathcal{E} \cup \{(t_2 : \neg\phi)_{r_1}, (t_1 : \neg\phi)_{r_2}\}}\ \mathcal{D}'_1[*]$$

$$\frac{(t_1 : \phi)_{r_1,r_i,\cdots,r_j} \quad (t_2 : \phi)_{r_2,r_m,\cdots,r_n}}{\mathcal{E} \cup \{(t_2 : \neg\phi)_{r_1,r_i,\cdots,r_j}, (t_1 : \neg\phi)_{r_2,r_m,\cdots,r_n}\}}\ \mathcal{D}''_1[*]$$

$$\frac{a_2 \lhd_p a_1 \quad a_1 : (t : \phi) \quad a_2 : (t : \neg\phi)}{\mathcal{E} \setminus \{a_2 : (t : \neg\phi)\}}\ \mathcal{D}_2[\circ] \qquad \frac{r_2 \prec r_1 \quad (t : \phi)_{r_1} \quad (t : \neg\phi)_{r_2}}{\mathcal{E} \setminus \{(t : \neg\phi)_{r_2}\}}\ \mathcal{D}'_2$$

$$\frac{r_2 \prec r_1 \quad (t : \phi)_{r_1,r_i,\cdots,r_j} \quad (t : \neg\phi)_{r_2,r_m,\cdots,r_n} \quad \Delta}{\mathcal{E} \setminus \{(t : \neg\phi)_{r_2,r_m,\cdots,r_n}, \Delta\}}\ \mathcal{D}''_2[\star]$$

$$\frac{a : (t_1 : \phi) \quad a : (t_2 : \phi)}{\bot}\ \mathcal{C}_C[*] \qquad \frac{(t_1 : \phi)_{r,\cdots,r_i} \quad (t_2 : \phi)_{r,\cdots,r_j}}{\bot}\ \mathcal{C}'_C[*]$$

$$\frac{a \lhd_p a}{\bot}\ \mathcal{C}_T \qquad \frac{r \prec r}{\bot}\ \mathcal{C}'_T \qquad \frac{t : \phi \quad t : \neg\phi}{\bot}\ \mathcal{C}_P$$

where $[*] = [t_1 \neq t_2]$, $[\circ] = [\phi \in Lit_S,\ \phi$ is p or $\neg p]$, and $[\star] = [\Delta = \bigcup_{(t' : \psi)_\varrho \in \mathcal{E}\ s.t.\ r_2 \in \varrho} (t' : \psi)_\varrho]$

4.1 Rewriting Rules

We now explain the rules in more detail, starting from the transformation rules that transform the formulas into the various layer formulas.

Rule $\mathcal{L}_1$ transforms a simple evidence into a temporal formula of the interpretation layer, whereas $\mathcal{L}'_1$ transforms formulas of the reasoning layer into a temporal formula of the interpretation layer.

Example 6. An application of $\mathcal{L}_1$ in our use example is:

$$\frac{FE : (t_2 : SpeedTr(23MB/s))}{\mathcal{E} \cup \{t_2 : SpeedTr(23MB/s)\}}\ \mathcal{L}_1$$ □

Rule $\mathcal{L}_2$ transforms a derived evidence into an interpretation formula and, if possible, also introduces new pieces of evidence. Thus, given a derived evidence $a : (t : \phi)\ [a_1 : (t_1 : \phi_1) \mid \cdots \mid a_n : (t_n : \phi_n)]_r$, $\mathcal{L}_2$ inserts the temporal formula $t_1 : \phi_1 \wedge \ldots \wedge t_n : \phi_n \rightarrow_r t : \phi$ and all $a_i : (t_i : \phi_i)$ for $\phi_i \in Lit_S$ and $i \in \{1, \ldots, n\}$. Note that rule $\mathcal{L}_2$ inserts in the theory only the simple pieces of evidence that

were part of the reasoning, and not the derived ones, as we expect the pieces of evidence that are part of their reasonings to be part of the theory too.

Example 7. In our example, $\mathcal{L}_2$ is applied to all derived pieces of evidence given by the sources. When it applies to *CS*'s first evidence, it transforms only the simple evidence about the spear phishing campaign, but not the successful phishing evidence as it is a derived one. The same occurs to *TF*'s first evidence where just *MetaC* is introduced:

$$\frac{CS : (t_1 : Attack)\ [CS : (t_1 : SPhish) \mid CS : (t_1 : SucPhish)]_{r_1}}{\mathcal{E} \cup \{CS : (t_1 : SPhish)\} \cup \{t_1 : SPhish \wedge t_1 : SucPhis \rightarrow_{r_1} t_1 : Attack\}}\ \mathcal{L}_2$$

$$\frac{TF : (t_2 : Attack)\ [TF : (t_2 : MetaC) \mid TF : (t_2 : PhysA)]_{r_3}}{\mathcal{E} \cup \{TF : (t_2 : MetaC)\} \cup \{t_2 : MetaC \wedge t_2 : PhysA \rightarrow_{r_3} t_2 : Attack\}}\ \mathcal{L}_2$$

$$\frac{TF : (t_2 : PhysA)\ [TF : (t_2 : \neg SpeedTr(23MB/s))]_{r_4}}{\mathcal{E} \cup \{TF : (t_2 : \neg SpeedTr(23MB/s))\} \cup \{t_2 : \neg SpeedTr(23MB/s) \rightarrow_{r_4} t_2 : PhysA\}}\ \mathcal{L}_2$$

Applying $\mathcal{L}_2$ to the second evidence of *CS* yields $\mathcal{E} \cup \{CS : (t_1 : LinkCl),\ CS : (t_1 : FFill),\ CS : (t_1 : DStolen)\}$ and $t_1 : LinkCl \wedge t_1 : FFill \wedge t_1 : DStolen \rightarrow_{r_2} t_1 : SucPhish$. □

Rules $(\rightarrow)$ and $(\rightarrow')$ transform the interpretation formulas introduced by $\mathcal{L}_2$ into reasoning formulas (derived evidence of the two types).

Example 8. Applying $(\rightarrow)$ to *CS*'s derived pieces of evidence yields:

$$\frac{t_1 : LinkCl \wedge t_1 : FFill \wedge t_1 : DStolen \rightarrow_{r_2} t_1 : SucPhish \quad t_1 : LinkCl \quad t_1 : FFill \quad t_1 : DStolen}{\mathcal{E} \cup \{(t_1 : SucPhish)_{r_2}\}}\ (\rightarrow)$$

Applying $(\rightarrow')$ to the second type of derived evidence for *CS* yields

$$\frac{t_1 : SPhish \wedge t_1 : SucPhish \rightarrow_{r_1} t_1 : Attack \quad t_1 : SPhish \quad (t_1 : SucPhish)_{r_2}}{\mathcal{E} \cup \{(t_1 : Attack)_{r_1, r_2}\}}\ (\rightarrow')$$ □

The $\lhd$ and $\prec$ relations are transitive ones. Trans$\lhd$ and Trans $\prec$ extend the trust relations between agents and reasonings, e.g., if a_1 is less trusted than a_2 with respect to p, and a_2 is less trusted than a_3 with respect to p, then Trans$\lhd$ inserts into the theory the conclusions that a_1 is less trusted than a_3 with respect to p (the same applies to $\prec$ with Trans $\prec$).

The discordance resolution rules resolve temporal and factual discordances, where events are instantaneous and not recurring. A *temporal discordance* about an event occurs when two agents state that it occurred in two different instants of time, e.g., Alice states that x occurred at t_1 and Bob states that it occurred at t_2. A *factual discordance* about an event occurs when there are inconsistent statements about the occurrence of an event at an instant of time, e.g., Alice states that at t occurred p and Bob states that at t did not occur p.

Rules $\mathcal{D}_1$, $\mathcal{D}'_1$ and $\mathcal{D}''_1$ transform temporal discordances into factual ones, where $\mathcal{D}_1$ works with simple pieces of evidence, $\mathcal{D}'_1$ with derived pieces of evidence of the first type, and $\mathcal{D}''_1$ with derived pieces of evidence of the second type (note

that $\mathcal{D}'_1$ is a special case of $\mathcal{D}''_1$). Thus, if the $\mathcal{EL}$-theory $\mathcal{E}$ contains the evidence belonging to two different agents about the same event p, occurring at two different instants, then the evidence of the occurrence or not of p with respect to both agents and both instants of time are inserted in the theory.

Rule $\mathcal{D}_2$, $\mathcal{D}'_2$ and $\mathcal{D}''_2$ solve the factual discordances based on the relations of trust, where $\mathcal{D}_2$ eliminates from the theory the evidence of the less trusted agent, whereas $\mathcal{D}'_2$ and $\mathcal{D}''_2$ eliminate the evidence of the less trusted reasoning. $\mathcal{D}''_2$ eliminates also every evidence that has inside its reasoning the removed evidence, as captured by the side condition where Δ is the set of all derived pieces of evidence that have r_2 in their reasonings: $\Delta = \bigcup_{(t':\psi)_\varrho \in \mathcal{E}\ s.t.\ r_2 \in \varrho} (t' : \psi)_\varrho]$, where $\varrho = \{r_k, \cdots, r_l\}$.

Example 9. $\mathcal{D}_2$ solves the discordance of the speed transfer:

$$\frac{TF \lhd_{SpeedTr(23MB/s)} FE \quad FE : (t_2 : SpeedTr(23MB/s)) \quad TF : (t_2 : \neg SpeedTr(23MB/s))}{\mathcal{E} \setminus \{TF : (t_2 : \neg SpeedTr(23MB/s))\}}\ \mathcal{D}_2 \quad \square$$

The rewriting system has five closure rules that correspond to five discordances that cannot be solved resulting in the empty theory $\perp$. $\mathcal{C}_C$ applies when an agent contradicts herself, $\mathcal{C}'_C$ when a reasoning contradicts itself. $\mathcal{C}_T$ and $\mathcal{C}'_T$ apply when an agent/reasoning is more trusted than herself/itself (we avoid these types of conflicts in the semantics thanks to $COND_1$ and $COND_2$, where $\lhd$ and $\prec$ are irreflexive). Finally, $\mathcal{C}_P$ captures contradictions of the second layer, where two temporal formulas state the occurrence and non occurrence of an event at the same instant of time. This discordance occurs when we were not able to solve it using the trust relations.

Theorem 1. *The rewriting system of $\mathcal{EL}$ is sound.*

The proof of the theorem is in the Appendix.

4.2 Rewriting Procedure

We give a procedure that uses the rewriting rules to prove the satisfiability of a given $\mathcal{EL}$-theory. This procedure defines an order of application of the rules that rewrites the $\mathcal{EL}$-theory as defined in Algorithm 1. Theorem 2 tells us that the procedure is correct (the theorem is proved in the Appendix).

Given a $\mathcal{EL}$-theory, the procedure starts by generating all the trust relations, applying (TRANS$\lhd$) and (TRANS $\prec$). Any contradiction that exists between trust relations is immediately captured by $\mathcal{C}_T$ and $\mathcal{C}'_T$. $\mathcal{L}_2$ is applied to transform any derived evidence into its interpretations. If needed, $\mathcal{D}_1$ and $\mathcal{D}_2$ are applied. At this point all possible simple pieces of evidence are generated. Any contradiction between first layer formulas is captured by $\mathcal{C}_C$. Afterwards, $\mathcal{L}_1$ transforms any simple evidence into second layer formulas that are used by ($\rightarrow$) to obtain reasoning layer formulas. $\mathcal{D}'_1$ and $\mathcal{D}'_2$ are applied to solve discordances between reasoning layer formulas based on the reasonings' trust relations. The result of the previous rules is used by ($\rightarrow'$) to generate reasoning layer formulas from derived pieces of evidence of the second type. If any discordance arises, it is solved

Algorithm 1 Algorithm for the Rewriting Procedure

while We can apply TRANS$\lhd$, TRANS $\prec$ rules **do**
  Apply TRANS$\lhd$ and TRANS $\prec$ rules
end while
Apply $\mathcal{C}_T$ and $\mathcal{C}'_T$; **if** we have $\bot$, **then** We do not have a model. Exit! **endif**
while We can apply $\mathcal{L}_2$ rule **do** Apply $\mathcal{L}_2$ rule **end while**
while We can apply $\mathcal{D}_1$, $\mathcal{D}_2$ rules **do** Apply $\mathcal{D}_1$, $\mathcal{D}_2$ rules **end while**
Apply $\mathcal{C}_C$; **if** we have $\bot$, **then** We do not have a model. Exit! **endif**
while We can apply $\mathcal{L}_1$ rule **do** Apply $\mathcal{L}_1$ rule **end while**
while We can apply $(\rightarrow)$ rule **do** Apply $(\rightarrow)$ rule **end while**
while We can apply $\mathcal{D}'_1$, $\mathcal{D}'_2$ rules **do** Apply $\mathcal{D}'_1$, $\mathcal{D}'_2$ rules **end while**
while We can apply $(\rightarrow')$ rule **do** Apply $(\rightarrow')$ rule **end while**
while We can apply $\mathcal{D}''_1$, $\mathcal{D}''_2$ rules **do** Apply $\mathcal{D}''_1$, $\mathcal{D}''_2$ rules **end while**
Apply $\mathcal{C}'_C$; **if** we have $\bot$, **then** We do not have a model. Exit! **endif**
while We can apply $\mathcal{L}'_1$ rule **do** Apply $\mathcal{L}'_1$ rule **end while**
Apply $\mathcal{C}_P$; **if** we have $\bot$, **then** We do not have a model. Exit! **endif**

by $\mathcal{D}''_1$ and $\mathcal{D}''_2$, where rule $\mathcal{D}''_2$ not only takes out the not preferred evidence, but also any derived evidence that uses it as a precondition. If no contradiction between reasoning rules is captured by $\mathcal{C}'_C$, then $\mathcal{L}'_1$ transforms all reasoning layer formulas into interpretation layer ones. If $\mathcal{C}_P$ applies, then there is a contradiction and we have $\bot$, else no further transformation can be done, and the resulting set of formulas is the model of $\mathcal{EL}$-theory.

Example 10. By applying the procedure we find that $(\rightarrow)$ can be applied only to *CS*'s pieces of evidence as the derived ones of *TF* are missing their premises, removed by $\mathcal{D}_2$. Applying $\mathcal{L}'_1$ yields $t_1 : \mathit{Attack}$ and the analyst concludes that the attack occurred during March/April 2016. □

Theorem 2. *The order of the rules in Algorithm 1 used by the rewriting procedure is correct.*

5 A Detailed Case Study: Attribution of a Cyber-Attack

The Evidence Logic $\mathcal{EL}$ can be used in diverse application areas where there is a need to analyze and reason about conflicting data/knowledge. In this section, as a concrete proof of concept to show how to apply $\mathcal{EL}$ during the investigations on a cyber-attack, we discuss a cyber-forensics case study in which the analyst needs to collect various pieces of evidence and analyze them to decide who performed the attack; this process is called *attribution* of the attack to a particular entity.

As we remarked above, forensics investigations typically produce an enormous amount of evidence that need to be analyzed. The pieces of evidence are produced/collected by various sources, which can be humans (e.g., another analyst) or forensic tools such as intrusion detection system (IDS), traceback systems, malware analysis tools, and so on. The analyst trusts more some sources than others for particular pieces of evidence, e.g., source S_1 is more trusted than

$$\frac{A_1 : t : Culprit(C, Attack)[S_1 : t : sIP(Attack, IP) \mid S_1 : t : Geoloc(IP, C) \mid S_2 : t : Cap(C, Attack)]_{r_1}}{\begin{array}{c}\mathcal{E} \cup \{S_1 : t : sIP(Attack, IP), S_1 : t : Geoloc(IP, C)\} \cup \\ \{\ t : sIP(Attack, IP) \wedge t : Geoloc(IP, C) \wedge t : Cap(C, Attack) \rightarrow_{r_1} t : Culprit(C, Attack)\}\end{array}}\ \mathcal{L}_2 \quad (1)$$

$$\frac{A_2 : t : Culprit(C, Attack)[S_2 : t : Motive(C, Attack) \mid S_2 : t : Cap(C, Attack)]_{r_2}}{\mathcal{E} \cup \{\ t : Motive(C, Attack) \wedge t : Cap(C, Attack) \rightarrow_{r_2} t : Culprit(C, Attack)\}}\ \mathcal{L}_2 \quad (2)$$

$$\frac{A_3 : t : \neg Culprit(C, Attack)[S_3 : t : \neg Cap(C, Attack) \mid S_4 : t : \neg Fin(C, Attack)]_{r_3}}{\begin{array}{c}\mathcal{E} \cup \{S_4 : t : \neg Fin(C, Attack)\} \cup \\ \{t : \neg Cap(C, Attack) \wedge t : \neg Fin(C, Attack) \rightarrow_{r_3} t : \neg Culprit(C, Attack)\}\end{array}}\ \mathcal{L}_2 \quad (3)$$

$$\frac{S_2 : t : Cap(C, Attack)[S_6 : t_1 : Admit(C, Attack') \mid S_1 : t : Sim(Attack, Attack')]_{r_5}}{\begin{array}{c}\mathcal{E} \cup \{S_6 : t_1 : Admit(C, Attack'), S_1 : t : Sim(Attack, Attack')\} \cup \\ \{\ t_1 : Admit(C, Attack') \wedge t : Sim(Attack, Attack') \rightarrow_{r_5} t : Cap(C, Attack)\}\end{array}}\ \mathcal{L}_2 \quad (4)$$

$$\frac{S_3 : t : \neg Cap(C, Attack)[S_6 : t_1 : Admit(C, Attack') \mid S_5 : t : \neg Sim(Attack, Attack')]_{r_6}}{\begin{array}{c}\mathcal{E} \cup \{S_6 : t_1 : Admit(C, Attack'), S_5 : t : \neg Sim(Attack, Attack')\} \cup \\ \{\ t_1 : Admit(C, Attack') \wedge t : \neg Sim(Attack, Attack') \rightarrow_{r_6} t : \neg Cap(C, Attack)\}\end{array}}\ \mathcal{L}_2 \quad (5)$$

$$\frac{A_4 : t : \neg Culprit(C, Attack)[S_1 : t : sIP(Attack, IP) \mid S_1 : t : Geoloc(IP, C) \mid S_7 : t : Spoofed(IP)]_{r_4}}{\begin{array}{c}\mathcal{E} \cup \{S_1 : t : sIP(Attack, IP), S_1 : t : Geoloc(IP, C), S_7 : t : Spoofed(IP)\} \cup \\ \{\ t : sIP(Attack, IP) \wedge t : Geoloc(IP, C) \wedge t : Spoofed(IP) \rightarrow_{r_4} t : \neg Culprit(C, Attack)\}\end{array}}\ \mathcal{L}_2 \quad (6)$$

$$\frac{S_2 : t : Motive(C, Attack)[S_5 : t : EConf(C, Victim)]_{r_7}}{\mathcal{E} \cup \{S_5 : t : EConf(C, Victim)\} \cup \{\ t : EConf(C, Victim) \rightarrow_{r_7} t : Motive(C, Attack)\}}\ \mathcal{L}_2 \quad (7)$$

$$\frac{S_5 \lhd_{Sim} S_1 \quad S_1 : t : Sim(Attack, Attack') \quad S_5 : t : \neg Sim(Attack, Attack')}{\mathcal{E} \setminus \{S_5 : t : \neg Sim(Attack, Attack')\}}\ \mathcal{D}_2 \quad (8)$$

$$\frac{\begin{array}{c}t_1 : Admit(C, Attack') \qquad t : Sim(Attack, Attack') \\ t_1 : Admit(C, Attack') \wedge t : Sim(Attack, Attack') \rightarrow_{r_5} Cap(C, Attack)\end{array}}{\mathcal{E} \cup \{(t : Cap(C, Attack))_{r_5}\}}\ (\rightarrow) \quad (9)$$

$$\frac{\begin{array}{c}t : sIP(Attack, IP) \qquad t : Geoloc(IP, C) \qquad t : Spoofed(IP) \\ t : sIP(Attack, IP) \wedge t : Geoloc(IP, C) \wedge t : Spoofed(IP) \rightarrow_{r_4} t : \neg Culprit(C, Attack)\end{array}}{\mathcal{E} \cup \{(t : \neg Culprit(C, Attack))_{r_4}\}}\ (\rightarrow) \quad (10)$$

$$\frac{t : EConf(C, Victim) \rightarrow_{r_7} t : Motive(C, Attack) \quad t : EConf(C, Victim)}{\mathcal{E} \cup \{(t : Motive(C, Attack))_{r_7}\}}\ (\rightarrow) \quad (11)$$

$$\frac{\begin{array}{c}t : sIP(Attack, IP) \qquad t : Geoloc(IP, C) \qquad (t : Cap(C, Attack))_{r_5} \\ t : sIP(Attack, IP) \wedge t : Geoloc(IP, C) \wedge t : Cap(C, Attack) \rightarrow_{r_1} t : Culprit(C, Attack)\end{array}}{\mathcal{E} \cup \{(t : Culprit(C, Attack))_{r_1, r_5}\}}\ (\rightarrow') \quad (12)$$

$$\frac{\begin{array}{c}(t : Motive(C, Attack))_{r_7} \qquad (t : Cap(C, Attack))_{r_5} \\ t : Motive(C, Attack) \wedge t : Cap(C, Attack) \rightarrow_{r_2} t : Culprit(C, Attack)\end{array}}{\mathcal{E} \cup \{(t : Culprit(C, Attack))_{r_2, r_7, r_5}\}}\ (\rightarrow') \quad (13)$$

$$\frac{r_1 \prec r_4 \quad (t : Culprit(C, Attack))_{r_1, r_5} \quad (t : \neg Culprit(C, Attack))_{r_4}}{\mathcal{E} \setminus \{(t : Culprit(C, Attack))_{r_1, r_5}\}}\ \mathcal{D}_2'' \quad (14)$$

$$\frac{r_4 \prec r_2 \quad (t : Culprit(C, Attack))_{r_2, r_7, r_5} \quad (t : \neg Culprit(C, Attack))_{r_4}}{\mathcal{E} \setminus \{(t : \neg Culprit(C, Attack))_{r_4}\}}\ \mathcal{D}_2'' \quad (15)$$

Fig. 1. Application of the Rewriting Procedure

source S_5 for attack similarity as tool S_1 specializes in malware analysis whereas tool S_5 specializes in deleted data. The collected evidence can be conflicting or bring to conflicting results. The $\mathcal{EL}$ Logic represents the evidence, together with its sources and relations of trust, and reasons about it, by eliminating the conflicting evidence and helping the analyst during the analysis process.

Suppose the analyst has collected (from analysts $A_1, \ldots, A_4$ and sources S_1, S_2, S_3) and is analyzing, using $\mathcal{EL}$, the following pieces of evidence, representing

events related to the attack that occurred (for the sake of space, we give a simplified but realistic version of the evidence that can be easily extended).

$A_1 : t :\ Culprit(C, Attack)[S_1 : t : sIP(Attack, IP) \mid S_1 : t : Geoloc(IP, C) \mid S_2 : t : Cap(C, Attack)]_{r_1}$
$A_2 : t :\ Culprit(C, Attack)[S_2 : t : Motive(C, Attack) \mid S_2 : t : Cap(C, Attack)]_{r_2}$
$A_3 : t :\ \neg Culprit(C, Attack)[S_3 : t : \neg Cap(C, Attack) \mid S_4 : t : \neg Fin(C, Attack)]_{r_3}$
$A_4 : t :\ \neg Culprit(C, Attack)[S_1 : t : sIP(Attack, IP) \mid S_1 : t : Geoloc(IP, C) \mid S_7 : t : Spoofed(IP)]_{r_4}$
$S_1 : t :\ sIP(Attack, IP)$
$S_1 : t :\ Geoloc(IP, C)$
$S_2 : t :\ Cap(C, Attack)[S_6 : t_1 : Admit(C, Attack') \mid S_1 : t : Sim(Attack, Attack')]_{r_5}$
$S_3 : t :\ \neg Cap(C, Attack)[S_6 : t_1 : Admit(C, Attack') \mid S_5 : t : \neg Sim(Attack, Attack')]_{r_6}$
$S_2 : t :\ Motive(C, Attack)[S_5 : t : EConf(C, Victim)]_{r_7}$
$S_5 \lhd_{Sim} S_1 \qquad r_1 \prec r_4 \qquad r_4 \prec r_2 \qquad r_2 \prec r_3$

$sIP(Attack, IP)$ means that the *Attack* came from *IP*; $Geoloc(IP, C)$ that *IP* has country C as geographical location; $Cap(C, Attack)$ that country C has the capability of conducting the *Attack*. Analyst A_1 states that (based on reasoning r_1), given country C is capable of performing the *Attack* (stated by S_2) and it came from *IP* located in C (stated by S_1), then C performed (is the culprit of) the attack, i.e., $Culprit(C, Attack)$. A_2 states that C is the culprit (based on r_2), as it has the capability of and the motive $Motive(C, Attack)$ for performing it (both stated by S_2). A_3 states that C is not the culprit (based on r_3), as it is not capable of and (as stated by S_4) does not have the financial resources $Fin(C, Attack)$ for commissioning the attack. A_4 states that C is not the culprit (based on r_4), as the *IP*'s are *Spoofed* (stated by S_7), so their geolocation cannot be used. Source S_1 states that the *IP* from which the attack originated is located in C. S_2 states that C is capable (based on r_5), as C admitted to be the culprit of a previous attack, i.e., $Admit(C, Attack')$, at t_1 (stated by S_6), and the latter is similar (*Sim*) to *Attack* (stated by S_1). S_3 states that C is not capable of performing *Attack* (based on r_6), as $Attack'$ that C admitted to have performed, is not similar to *Attack* (stated by S_5). S_2 states that C has motive for the attack (based on r_7), as C has an economical conflict *EConf* with the attack *Victim* (stated by S_5). Our analyst trusts more source S_1 than S_5 for the similarity between attacks, and reasoning r_3 more than r_2, r_2 more than r_4 and r_4 more than r_1.

The simple pieces of evidence of this use case are:

$$Vars_S = \{sIP(Attack, IP), Geoloc(IP, C), Fin(C, Attack), Admit(C, Attack'), Sim(Attack, Attack'), Spoofed(IP), EConf(C, Victim)\}.$$

Let us now apply $\mathcal{EL}$'s rewriting procedure. We start with rules TRANS$\lhd$ and TRANS $\prec$: the first cannot be applied, the second yields $r_1 \prec r_2$, $r_1 \prec r_3$ and $r_4 \prec r_3$. Neither $\mathcal{C}_T$ nor $\mathcal{C}'_T$ can be applied. We show the application of $\mathcal{L}_2$ to the pieces of evidence in (1)–(7) in Fig. 1. In (8) rule $\mathcal{D}_2$ eliminates $S_5 : t : \neg Sim(Attack, Attack')$. No contradiction is captured by $\mathcal{C}_C$, and $\mathcal{L}_1$ transforms

all first layer formulas into second layer ones:

$$\mathcal{E} \cup \{t : sIP(Attack, IP), t : Geoloc(IP, C), t : \neg Fin(C, Attack), t : Spoofed(IP)$$
$$t_1 : Admit(C, Attack'), t : Sim(Attack, Attack'), t : EConf(C, Victim)\}.$$

(9)–(11) show applications of $(\rightarrow)$ to any evidence that has its premises in the theory. $\mathcal{D}'_1$ and $\mathcal{D}'_2$ cannot be applied as there is no temporal/factual discordance between derived pieces of evidence of the first type. Applying $(\rightarrow')$ produces derived pieces of evidence of the second type for A_1 and A_2 as shown in (12)–(13). A_3's evidence is not derived as C is capable to perform the attack. Rule $\mathcal{D}''_1$ cannot be applied. Rule $\mathcal{D}''_2$ is applied, as shown in (14)-(15), to the conflicting pieces of evidence where the reasonings' trust relations apply. Finally, $\mathcal{L}'_1$ transforms all third layer formulas into second layer ones:

$$\mathcal{E} \cup \{t : sIP(Attack, IP), t : Geoloc(IP, C), t : \neg Fin(C, Attack), t : Spoofed(IP)$$
$$t_1 : Admit(C, Attack'), t : Sim(Attack, Attack'), t : Cap(C, Attack),$$
$$t : EconfConflict(C, Victim), t : Motive(C, Attack), t : Culprit(C, Attack)\}.$$

The analyst, given the result of the procedure, concludes that the culprit of the *Attack* is C.

The question of "who performed the attack" is, in general, not an easy one to answer, but we believe that $\mathcal{EL}$ can be successfully used to analyze and filter the large amount of cyber-forensics evidence that an analyst needs to deal with. At the very least, $\mathcal{EL}$ allows an analyst to perform a first, formal filtering of the evidence and obtain different plausible conclusions, which the analyst can then further investigate.

6 Related Work and Concluding Remarks

When we introduced $\mathcal{EL}$ and discussed how it allows analysts to reason about simple and derived evidence given by different sources, we deliberately did not use the notion of "belief". We chose to do so as the main scope of our work is not to consider modalities of knowledge or belief, but to introduce a procedure that analyzes and filters the potentially enormous amount of forensics evidence, eliminate discordances and reach conclusions. The notion of evidence (both simple and derived evidence) can be represented quite naturally as agents' beliefs and, in fact, the reasoning process in $\mathcal{EL}$ could be considered a belief revision process. However, our procedure, differently from the belief revision process, uses a monotonic reasoning, does not distinguish between beliefs and knowledge, is based on the notion of trust, and does not apply the principle of minimal change.

Belief revision is the process of integrating new information with existing beliefs or knowledge [9, 17, 7, 4, 10]. It is performed based on the knowledge and beliefs of the user and the beliefs of other agents announced, privately [1, 6] or publicly [14, 5], and it uses non-monotonic reasoning. In our approach, we use *monotonic reasoning* as we expect only the final set, that represents our theory, to be consistent. Our procedure deals with conflicting pieces of evidence, which are analyzed by expanding or contracting the evidence set. In case of unsolved

inconsistencies, our theory is empty. The procedure does not incorporate every incoming information in the evidence set, but rather the new evidence is included or not depending on the trust relations. This is different from the classical AGM belief revision [2], where the *principle of minimal belief change* applies.

Our analysis can be seen as a revision procedure, where we do not distinguish between beliefs and knowledge. Thus, all the pieces of evidence can be treated as beliefs, and there is no space for personal or common belief/knowledge. Some works have considered belief revision that uses relation of trust between agents [13, 12, 8, 3]. However, not much effort has been devoted to working with a relation of trust relative to the reasoning used to arrive to certain conclusions. Our trust relations do not have a grading system, like the one in [13], which is difficult to define for cyber-forensics data, but use comparable trust between the sources based on the evidence, similar to [12], where a notion of trust restricted to a domain of expertise is used. As future work, we plan to use Bayesian belief networks [8], and the Dempster-Shafer theory to quantify the level of trust for the evidence, and to enrich our framework with trust reinforcement mechanisms.

To the best of our knowledge, the only attempt at using belief revision during cyber-attacks' investigations is [15, 16], where a *probabilistic structure argumentation* framework is used to analyze contradictory and uncertain data. Our procedure does not deal with probabilities, but with preferences between sources and reasoning rules. We believe this to be a more accommodating approach, especially for the main use case, investigations of cyber-attacks, where calculating and revising probabilities is resource consuming. The framework of [15, 16] allows attackers to use, during the deceptive attempts, the well-known *specificity criteria*, i.e., the rule that uses more pieces of evidence is preferable. We avoid this type of deceptive attempts as the trust relations are given by the analyst.

$\mathcal{EL}$ is based on LTL. Another approach is to use *Temporal Defeasible Logic* [4], where knowledge is represented as norms with temporal scope [11]. For the sake of simplicity, our stream of time is discrete and provided initially. As future work, we plan to consider the flow of time as not provided and as non-discrete in order to have temporal relations between labels that represent the instants of time.

Another distinctive feature of our approach with respect to the rest of the literature that focuses on agents' trust relations and their reputation systems is the fact that we engage not only with the trust between agents, but also with the reasoning behind the evidence. Hence, even when a particular agent is not trusted, if the reasoning behind the evidence is sound, we might take it into account. The notion of trust, also seen as preference, is subjective to the analyst, and we assume that agents are sincere, and thus share all their information. As future work, we plan to incorporate both a reputation revision process, where the trustworthiness and reliability of the sources is analyzed and revised based on past experience, and private/public announcements. Finally, on the theoretical side, we plan to investigate the completeness of the rewriting system and algorithm, whereas on the practical side, we plan to fully automate our analysis process and to perform an evaluation analysis on real evidence of cyber-attacks.

Acknowledgments

Erisa Karafili was supported by the European Union's H2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 746667.

References

1. Ågotnes, T., Balbiani, P., van Ditmarsch, H., Seban, P.: Group announcement logic. J. Applied Logic **8**(1), 62–81 (2010)
2. Alchourròn, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. Journal of Symbolic Logic **50**, 510–530 (1985)
3. Alechina, N., Jago, M., Logan, B.: Preference-based belief revision for rule-based agents. Synthese **165**(2), 159–177 (Nov 2008)
4. Augusto, J.C., Simari, G.R.: Temporal defeasible reasoning. Knowl. Inf. Syst. **3**(3), 287–318 (2001)
5. Balbiani, P., van Ditmarsch, H., Herzig, A., de Lima, T.: A Tableau Method for Public Announcement Logics. In: Tableaux. pp. 43–59. Springer (2007)
6. Balbiani, P., Guiraud, N., Herzig, A., Lorini, E.: Agents that speak: modelling communicative plans and information sources in a logic of announcements. In: AAMAS 2011, Volume 1-3. pp. 1207–1208 (2011)
7. Baltag, A., Smets, S.: Conditional doxastic models: A qualitative approach to dynamic belief revision. Electr. Notes Theor. Comput. Sci. **165**, 5–21 (2006)
8. Barber, K.S., Kim, J.: Belief revision process based on trust: Agents evaluating reputation of information sources. In: AGENTS 2000. pp. 73–82 (2000)
9. van Benthem, J.: Dynamic logic for belief revision. Journal of Applied Non-Classical Logics **17**(2), 129–155 (2007)
10. Dix, J., Hansson, S.O., Kern-Isberner, G., Simari, G.R.: Belief change and argumentation in multi-agent scenarios. Annals of Mathematics and Artificial Intelligence **78**(3), 177–179 (2016)
11. Governatori, G., Terenziani, P.: Temporal extensions to defeasible logic. In: Australian Conference on Artificial Intelligence. pp. 476–485 (2007)
12. Hunter, A., Booth, R.: Trust-sensitive belief revision. In: IJCAI 2015. pp. 3062–3068 (2015)
13. Lorini, E., Jiang, G., Perrussel, L.: Trust-based belief change. In: ECAI 2014 - Including PAIS 2014. pp. 549–554 (2014)
14. Plaza, J.: Logics of public communications. Synthese **158**(2), 165–179 (2007)
15. Shakarian, P., Simari, G.I., Moores, G., Parsons, S.: Cyber attribution: An argumentation-based approach. In: Cyber Warfare - Building the Scientific Foundation. pp. 151–171 (2015)
16. Shakarian, P., Simari, G.I., Moores, G., Paulo, D., Parsons, S., Falappa, M.A., Aleali, A.: Belief revision in structured probabilistic argumentation - model and application to cyber security. Ann. Math. Artif. Intell. **78**(3-4), 259–301 (2016)
17. Van Ditmarsch, H., van Der Hoek, W., Kooi, B.: Dynamic epistemic logic, vol. 337. Springer Science & Business Media (2007)

A Appendix: Soundness of the Rewriting System and Correctness of the Algorithm

In this appendix, we prove the soundness of the rewriting system of $\mathcal{EL}$ and the correctness of Algorithm 1. Given a theory $\mathcal{E}$ and $\mathcal{EL}$'s rewriting system, the application of at least one of its closure rules generates an empty set. In fact, every theory that contains $\perp$ is equivalent to the empty theory. When the input theory is not empty and has *no contradiction*, then the theory rewritten by $\mathcal{EL}$ should give as result a non empty theory.

As usual in *tableau rewriting systems*, we define three fundamental notions: open, closed, and exhausted theories. A theory is *closed* when it contains a contradiction and it is *open* when it does not. A theory is *exhausted* when it is a fixpoint with respect to the rewriting process, i.e., by applying the rewriting system to an exhausted theory $\mathcal{E}$, we always obtain $\mathcal{E}$. Under the grounded semantics introduced in Section 3, we prove the soundness of the rewriting system by showing that open theories have models under the semantics, and closed ones have not. Thus, when we find an open and exhausted theory, we can prove the existence of a model.

We show now that the rules that rewrite a theory $\mathcal{E}$ into $\widehat{\mathcal{E}}$ without introducing $\perp$ constitute by themselves a sound system. The proofs of Lemma 1 and Lemma 2 are straightforward and are omitted for the sake of space.

Lemma 1. *If a satisfiable $\mathcal{EL}$-theory $\mathcal{E}$ is rewritten into an exhausted theory $\widehat{\mathcal{E}}$, without using the closure rules, then $\widehat{\mathcal{E}}$ entails consequence C only when C is a consequence of $\mathcal{E}$.*

Lemma 2. *If an unsatisfiable $\mathcal{EL}$-theory $\mathcal{E}$ is rewritten into an exhausted theory $\widehat{\mathcal{E}}$, then $\widehat{\mathcal{E}}$ is empty.*

Lemma 3. *Given a satisfiable theory $\mathcal{E}$, the rewriting system $\mathcal{EL}$ rewrites the theory in an open and exhausted one.*

Proof. This lemma is proved by contradiction. Assume that $\mathcal{E}$ is non empty and satisfiable, and is rewritten by $\mathcal{EL}$ into an exhausted closed theory $\widehat{\mathcal{E}}$. Starting from a satisfiable theory $\mathcal{E}$ there are five cases of rewriting it in a contradictory theory that gives as result $\perp$. In the definition of model in Section 3, we introduced four conditions that constrain the behavior of interpretations. Below we provide the complete analysis only for the first case (that is provided as a consequence of $COND_1$). The other cases are a natural extension of this one and are omitted for the sake of space. The first case occurs when applying the $\mathcal{C}_C$ rule. We have that: $a^{\mathfrak{I}}(t_1, p) = True$, and $a^{\mathfrak{I}}(t_2, p) = True$. $COND_1$ implies that a propositional variable referred to an agent a can be true in only one instant of time, thus, $\mathcal{E}$ is not satisfiable. ■

Lemma 3 introduces a result for $\mathcal{EL}$-soundness as it guarantees that irregardless of the order in which we apply the rules, we catch a contradiction at a given point. Thus, as a direct consequence of the grounded semantics, of Lemma 2 and of Lemma 3 we obtain Theorem 1.

When a sound rewriting system exists in a logical reasoning system we always have a method to deliver satisfiability check of a theory. In this case, we apply the rules to a theory until we reach a fixpoint. If we aim at developing an effective method, however, we need to provide a proof of termination for such a method. For the rewriting system of $\mathcal{EL}$ we can prove that a simple approach, based on the execution of the rules in a given order, is sufficient to provide an effective method for satisfiability checking. This is the result of correctness of Algorithm 1. Firstly, in Lemma 4, we prove that the existence of $\bot$ in a theory, if not introduced by default, is the consequence of the application of the rules in a specific order.

Lemma 4. *If a satisfiable $\mathcal{EL}$-theory $\mathcal{E}$ is rewritten into a contradictory $\widehat{\mathcal{E}}$, then we have:*

1. *rewritten the theory by using $\mathcal{L}_1$ before $\mathcal{D}_1$ and $\mathcal{D}_2$, or*
2. *rewritten the theory by using $\mathcal{L}'_1$ before $\mathcal{D}'_1$, $\mathcal{D}''_1$, $\mathcal{D}'_2$ and $\mathcal{D}''_2$, or*
3. *rewritten the theory by using $(\rightarrow)$ before $\mathcal{D}_1$ and $\mathcal{D}_2$, or*
4. *rewritten the theory by using $(\rightarrow')$ before $\mathcal{D}'_1$ and $\mathcal{D}'_2$, or*
5. *applied (TRANS$\lhd$) after $\mathcal{D}_1$ and $\mathcal{D}_2$, and*
6. *applied (TRANS $\prec$) after $\mathcal{D}'_1$, $\mathcal{D}''_1$, $\mathcal{D}'_2$ and $\mathcal{D}''_2$.*

Proof. There are two cases when $\widehat{\mathcal{E}}$ is empty: either (1) the theory $\mathcal{E}$ is empty or (2) a closure rule was used. The first case is not possible by definition of the theory, as we assume that $\mathcal{E}$ is not empty. The second case occurs if at least one of the five closure rules applied. Suppose by contradiction that rule $\mathcal{C}_T$ or $\mathcal{C}'_T$ is used to compute the contradictory $\widehat{\mathcal{E}}$. The application of this rule leads to a contradiction as $\mathcal{E}$ is satisfiable, whilst $\mathcal{C}_T$ or $\mathcal{C}'_T$ are applied when there is a contradiction in the theory. The same applies for rules $\mathcal{C}_C$ and $\mathcal{C}'_C$.

Suppose, ad absurdum, that $\mathcal{C}_P$ leads to a contradictory $\widehat{\mathcal{E}}$. The premises of $\mathcal{C}_P$ are obtained using rules $\mathcal{L}_1$, $\mathcal{L}'_1$, $\mathcal{L}_2$, $(\rightarrow)$ and $(\rightarrow')$. The first case for having a contradiction captured by $\mathcal{C}_P$ is when $\mathcal{L}_1$ is applied before $\mathcal{D}_1$ and $\mathcal{D}_2$. This happens because a contradiction is found that in fact was solved by $\mathcal{D}_1$ and $\mathcal{D}_2$, as $\mathcal{E}$ is satisfiable. The second case for having a contradiction captured by $\mathcal{C}_P$ is when $\mathcal{L}'_1$ is applied before $\mathcal{D}'_1$, $\mathcal{D}''_1$, $\mathcal{D}'_2$ and $\mathcal{D}''_2$. This happens because a contradiction is found that in fact was solved by $\mathcal{D}'_1$, $\mathcal{D}''_1$, $\mathcal{D}'_2$ and $\mathcal{D}''_2$, as $\mathcal{E}$ is satisfiable. The third case for having a contradiction captured by $\mathcal{C}_P$ is when $(\rightarrow)$ is applied before $\mathcal{D}_1$ and $\mathcal{D}_2$. This happens because the formulas that were introduced produce the contradictions that in fact were solved by $\mathcal{D}_1$ and $\mathcal{D}_2$, as $\mathcal{E}$ is satisfiable. The fourth case is similar to the third and occurs when $(\rightarrow')$ is applied before $\mathcal{D}'_1$ and $\mathcal{D}'_2$. The fifth case for having a contradiction captured by $\mathcal{C}_P$ is when TRANS$\lhd$ is applied after $\mathcal{D}_1$ and $\mathcal{D}_2$. This happens because the contradictions found could be solved by $\mathcal{D}_1$ and $\mathcal{D}_2$ if the TRANS$\lhd$ rule was applied before, as $\mathcal{E}$ is satisfiable. The sixth case for having a contradiction captured by $\mathcal{C}_P$ is when TRANS $\prec$ is applied after $\mathcal{D}'_1$, $\mathcal{D}''_1$, $\mathcal{D}'_2$ and $\mathcal{D}''_2$. This happens because the contradictions found could be solved by $\mathcal{D}'_1$, $\mathcal{D}''_1$, $\mathcal{D}'_2$ and $\mathcal{D}''_2$ if the TRANS $\prec$ rule was applied before, as $\mathcal{E}$ is satisfiable. ■

We are now able to prove that the rewriting procedure introduced in Section 4.2 establishes satisfiability as defined in Definition 5. We prove that the provided *specific* order of application of the rewriting rules determines the existence of a model. Given Theorem 1, we prove that the rewriting given by Algorithm 1 is exhausted. Theorem 2 follows by applying Lemma 3 and Lemma 4.

Proof (Theorem 2). Based on the semantics introduced in Section 3 and given that Algorithm 1 applies the rules in the order specified in Lemma 4, we show that every theory that is unsatisfiable is rewritten by Algorithm 1 in a closed one, and consequently, every open theory resulting by the rewriting procedure, is also exhausted. We prove this by induction on the theory construction.

The base cases occur for a relational formula, a simple evidence, or a derived one. For lack of space, we omit the proofs as they follow quite straightforwardly by the definitions of relational formula, simple evidence, derived evidence, and the $\lhd$ and $\prec$ relations.

For the inductive step, we assume that $\mathcal{E}$ is formed by either n relational formulas, or a blend of n formulas, and that we know that the claim is true for $n-1$ formulas, and we show that the claim then holds also for n formulas.

Assume that $\mathcal{E}$ is formed by n different relational formulas. The only rules that can be applied are TRANS$\lhd$ and TRANS $\prec$, and the algorithm applies them. If $\mathcal{E}$ is unsatisfiable, then $\widehat{\mathcal{E}}$ is empty as rule $\mathcal{C}_T$ or rule $\mathcal{C}'_T$ capture any existing contradictions between relational formulas. If $\mathcal{E}$ is satisfiable, then $\widehat{\mathcal{E}}$ is open and exhausted as the algorithm has applied all possible rules.

Assume that $\mathcal{E}$ is formed by n different simple pieces of evidence. The algorithm first tries to apply $\mathcal{D}_1$. If $\mathcal{E}$ is unsatisfiable and there are discordances, then $\widehat{\mathcal{E}}$ is empty, because the algorithm applies $\mathcal{C}_C$. If there are no discordances in $\mathcal{E}$, then the algorithm translates all the rules into second layer formulas, by applying rule $\mathcal{L}_1$. Since $\mathcal{E}$ is unsatisfiable, the algorithm applies the closing rule $\mathcal{C}_P$ to capture the discordances between second layer formulas, and $\widehat{\mathcal{E}}$ is empty. If $\mathcal{E}$ is satisfiable, then the algorithm applies $\mathcal{L}_1$, and $\widehat{\mathcal{E}}$ is open and exhausted as the algorithm has applied all possible rules.

Assume that $\mathcal{E}$ is formed of n different derived pieces of evidence. The algorithm tries to apply the rules in the following order: $\mathcal{L}_2$, $\mathcal{D}_1$, $\mathcal{L}_1$, $(\rightarrow)$, $\mathcal{D}'_1$, $(\rightarrow')$, $\mathcal{D}''_1$ and $\mathcal{L}'_1$. If $\mathcal{E}$ is unsatisfiable, then the algorithm applies one of the closing rules $\mathcal{C}_C$, $\mathcal{C}'_C$ and $\mathcal{C}_P$ to capture the discordances between formulas of the different layers, and $\widehat{\mathcal{E}}$ is empty. If $\mathcal{E}$ is satisfiable, then the algorithm yields a $\widehat{\mathcal{E}}$ that is open and exhausted as it has applied all possible rules.

Assume that $\mathcal{E}$ is formed of n different formulas (pieces of evidence and relational formulas). The algorithm tries to apply all of its rules. If $\mathcal{E}$ is unsatisfiable, then the algorithm applies one of the closing rules to capture the discordances between formulas of the different layers, and $\widehat{\mathcal{E}}$ is empty. We know that our algorithm is able to capture all the contradictions, because the algorithm first applies all the rules that can surface all the possible contradictions and then it applies the appropriate closing rule. If $\mathcal{E}$ is satisfiable, then the algorithm yields a $\widehat{\mathcal{E}}$ that is open and exhausted as it has applied all possible rules. ■