

A Logical Analysis of an Information Filtering Architecture Based on Epistemic Trust Inference

Xu Li, Leendert van der Torre, Liuwen Yu

University of Luxembourg
2, place de l'Université, L-4365 Esch-sur-Alzette, Luxembourg
xu.li@uni.lu, leon.vandertorre@uni.lu, liuwen.yu@uni.lu

Abstract

In agent theory, epistemic trust is used to infer beliefs, for example by filtering out the information the agent receives from untrustworthy agents. Moreover, trust itself can be inferred from other information. We introduce a simple information filtering architecture that clearly distinguishes the relation between the two kinds of inference. We provide a logical analysis of the architecture, based on a new family of input/output logics. We then explore information filtering and belief manipulation within this formal framework. Our key finding is that with this architecture, some of the widely debated logical rules for trust inference are redundant with respect to information-filtering mechanisms and some others are redundant with respect to belief manipulation.

1 Introduction

In agent theory, epistemic trust (Liau 2003) is a relation between two agents, and it is used to infer beliefs. We call the inverse of the trust relation belief manipulation (Hunter, Schwarzentruher, and Tsang 2017; Eiter, Hunter, and Schwarzentruher 2021). This means that if agent X trusts agent Y with respect to the belief in p , agent Y can manipulate X's belief in p .

The trust relation can be conceptualized as two related inference processes. On the one hand, trust itself can be inferred from some information. On the other hand, the inferred trust can be used to infer beliefs by filtering out the information the agent receives from untrustworthy agents.

In this paper, we are interested in combining these two kinds of inferences from an architectural and logical point of view. Our research questions are as follows.

1. How to define an information filtering architecture that clearly distinguishes between the relations involved in the two kinds of inference, in which belief manipulation is the inverse of epistemic trust?
2. How to provide a logical analysis of the architecture in which the optional properties of the two kinds of inferences are represented by optional logical rules?
3. Which of the logical rules are redundant with respect to information-filtering mechanisms, and which ones are redundant for belief manipulation?

We explore information filtering and belief manipulation within the general formal approach of input/output logics (Makinson and Van Der Torre 2000) because information filtering has the general property that the input is not included in the output. In fact, information filtering has the inverse property that the output is included in the input. We therefore introduce a new family of input/output logics to satisfy this property.

We focus exclusively on the I/O-based information filtering architecture and do not consider alternative interpretations (such as power or rights) which may give rise to other logical rules. Moreover, we do not consider other logics of trust in computer science, for example logical approaches to trust management (Becker, Russo, and Sultana 2012).

Our main contributions can be summarized as follows:

1. A new two-step information-filtering architecture separating trust inference from belief acceptance.
2. A comprehensive family of trust I/O logics, including the first semantic characterizations of certain derivation systems.
3. The first study of the manipulation problem in trust-based information filtering.
4. The revelation from formal redundancy results that some of the widely debated logical rules for trust inference are redundant with respect to information-filtering and belief manipulation.

The structure of the paper is as follows. Section 2 introduces pre-logical transformers. Section 3 develops a family of input/output logics for trust inference and then analyzes their behavior through various inference rules. Section 4 applies these logics to a) define an information filtering mechanism and b) examine which trust inference rules are redundant in that context. Section 5 turns to belief manipulation, formally characterizing which formulas are manipulable under different filtering operations. Section 6 discusses related work, and Section 7 concludes with a summary of the paper and our future research.

2 Information Filtering Architecture on a Pre-Logical Level

Makinson and Van der Torre (2000) distinguish between two main kinds of input/output processes. The most studied processes are desiderative transformations where inputs may be

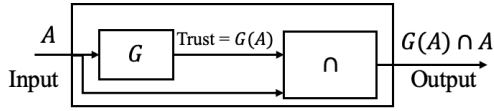


Figure 1: Pre-logical transformer

conditions and outputs express what is deemed desirable under those conditions. In this paper, we consider processes for information filtering. Some inputs pass, others are blocked or modified.

The box may stop some inputs, while letting others through, perhaps in modified form. Inputs may record reports of agents, of the kind ‘according to source i , x is true’, while the box may give as output either x itself, a qualified version of x , or nothing at all, according to the identity of i . Or it might give output x only when at least two distinct sources vouch for it, and so on.

Makinson and Van der Torre (2000) observe that classical logic serves as an inference motor, with premises as inputs and conclusions as outputs. Real-world cases call for reasoning beyond mere classical inference.

It may also be seen in another role, as ‘secretarial assistant’ to some other, perhaps non-logical, transformation engine. The task of logic is one of preparing inputs before they go into the machine, unpacking outputs as they emerge and, less obviously, coordinating the two. The process as a whole is one of inference only when the central transformation is so. In general, it is one of ‘logically assisted’ transformation. That is the general perspective underlying the present paper.

On a pre-logical level, the input/output process can be explained as follows. Consider any set L (not necessarily of formulas), and any relation $G \subseteq L \times L$, which is referred to as a *transformer* (Makinson and Van Der Torre 2000), as illustrated in Figure 1. Given an input $A \subseteq L$, the output of A under G can be understood as $G(A) = \{x : (a, x) \in G \text{ for some } a \in A\}$. The architecture operates in two steps. First, an input set A is passed into transformer G , producing the intermediate output $G(A)$, which represents the set of elements that would be trusted or authorized given A . This stage corresponds to trust inference. The result $G(A)$ is labeled as “Trust” in the figure, and corresponds to the output of the transformer component. In the second stage, this set is intersected with the original input A . The result $G(A) \cap A$ forms the final output of the entire architecture. This final output represents those elements that are both informed and trusted, i.e., what is ultimately accepted.

Consider a school entry policy stipulating that children may enter the school only if accompanied by at least one parent, while the parents themselves are prohibited from entering. A guard enforces this rule strictly: a child without a parent is denied entry and a parent without a child is also not allowed to enter. Let $P = \{p_1, p_2, p_3\}$ be the parents, $C = \{c_1, c_2, c_3\}$ the children, and parent-child re-

lation $G = \{(p_1, c_1), (p_1, c_2), (p_2, c_1), (p_3, c_2)\}$, so that c_1 has parents p_1, p_2 and c_2 has parents p_1, p_3 . Given an input set $A \subseteq P \cup C$ of individuals at the gate, the authorized output is $G(A) = \{x \in C \mid \exists a \in A : (a, x) \in G\}$: a child is admitted if at least one parent is present. Thus, if $A = \{p_1\}$, both c_1 and c_2 may enter, while if $A = \{c_1\}$, nobody enters.

We distinguish between *local* and *global* output. The *local output* of a transformer G is the set of all x such that $(a, x) \in G$ for some a in the input. The *global output* further restricts this by intersecting the local output with the input; if the input represents *informed statements* and the local output represents (*epistemic*) *trust*, then the global output corresponds to *accepted belief*. In the school example, the local output consists of all children authorized to enter given the presence of a parent whereas the global output consists of only those children who are both authorized to enter and are actually present. This captures the notion of a whitelist policy. We also define *manipulability*: an element $x \in L$ is manipulable if some input $A \subseteq L$ satisfies $x \in G(A) \cap A$. Since the global output represents accepted belief, manipulable elements are exactly those that an agent can be made to believe. In the school example, c_3 has no parent linked in G , so for any $A \subseteq P \cup C$, we have $c_3 \notin G(A) \cap A$, therefore c_3 is not manipulable.

3 Inferring Trust

In this section, we will present a family of I/O logics for trust inference. We start by introducing the basics of the I/O logic framework.

We assume a non-empty set PROP of propositional variables or atoms. Let \mathcal{L} be the propositional language generated by PROP, and elements of \mathcal{L} are called formulas. For a finite set A of formulas, $\bigwedge A$ is the conjunction of all formulas in A , and likewise for $\bigvee A$. Note that $\bigwedge \emptyset := \top$ and $\bigvee \emptyset := \perp$. For all sets of formulas $A \cup \{a\}$, $A \vdash a$ denotes that a is a logical consequence of A in classical propositional logic. $Cn(A) = \{a \mid A \vdash a\}$ is the set of all the logical consequences of A . Given two formulas a and b , $a \dashv\vdash b$ abbreviates that $a \vdash b$ and $b \vdash a$.

In the I/O logic framework (Makinson and Van Der Torre 2000), a set G of pairs of formulas is called a generating set. In this paper, a pair $(a, x) \in G$ is interpreted as “if a is informed, then judgment x is trusted”. The idea behind “conditional trust” is that, in some cases, trust may be triggered only if relevant information is provided, as in the following example:

Example 1. Consider the following two statements made by an epidemiologist:

- x : Vaccination reduces the spread of disease.
- y : Economic lockdowns are ineffective in preventing recessions.

In this scenario, the epidemiologist’s claim about vaccination falls within their domain of expertise and is thus trusted unconditionally. In our framework, this is represented by (\top, x) . However, the claim about economic outcomes is outside their field and is trusted only if supported by relevant evidence. Let a be “The epidemiologist has also published

this in in peer-reviewed economic journals.” Then we may represent the conditional trust as (a, y) .

An input is a set of formulas representing informed statements. The question is how to reasonably define the set of formulas making up the output of A under G , which is intended to characterize the set of judgments that can be trusted given information A . The solution can be proposed in terms of either proof theory or semantics.

We first introduce the I/O logics proof theory. The three basic inference rules T, SI (“Strengthening of the Input”), and OEQ (“Output Equivalence”) are listed below. T states that propositional tautologies are always unconditionally trusted. OEQ means that trust does not depend on the syntactic form of the sentence. SI states that trust is monotonic: if x is trusted given information a , then it is still trusted given a more informative statement b .

Remark 1. In SI, the formula b may be inconsistent. Then it is questionable whether x can still be trusted given b . We will address the problem of consistency in the next section, where we propose an information filtering architecture with a consistency check.

Here are the basic rules:

- T From no premise to (\top, \top) .
- SI From (a, x) to (b, x) whenever $b \vdash a$.
- OEQ From (a, x) to (a, y) whenever $x \dashv\vdash y$.

In addition to the basic rules, we consider the following four optional rules. All of them have been discussed in the logical literature on trust. For example, ST abbreviates “Symmetric Trust”. (The unconditional version of) ST is accepted by Liau (2003) as an optional rule for trust inference whereas AND is rejected in the same paper. The reason is that with AND, the conjunction $x \wedge y$ may be inconsistent (and, thus, untrustable), even if both x and y are consistent. Liau (2003) does not discuss a weaker version of AND, DT (“Disjunctive Trust”), first introduced by Dastani et al. 2005, where DT is argued to be a reasonable inference rule for trust. Finally, WO is also discussed in Liau’s paper but is considered to be problematic for trust. Recently, a normal modal logic for “trust in sincerity” was proposed by Leturc and Bonnet (2018) in which all the following rules are valid for “trust in sincerity” except for ST:

- ST From (a, x) to $(a, \neg x)$.
- DT From (a, x) and (a, y) to $(a, x \vee y)$.
- AND From (a, x) and (a, y) to $(a, x \wedge y)$.
- WO From (a, x) to (a, y) whenever $x \vdash y$.

In this paper, we do not aim to settle the dispute over the “correct” inference rules for trust. Instead, we focus on the information filtering architecture (to be introduced) and study the effects of different inference rules on this architecture. As we shall see, some of the above rules are redundant for the purpose of the information filtering architecture.

In this section, we comprehensively study the I/O logics generated by all the basic rules and subsets of the optional rules.

Definition 2. Let $R \subseteq \{\text{ST, DT, AND, WO}\}$. For all generating sets G , $deriv_R(G)$ is the smallest set that contains G , and it is closed under all the basic rules and rules in R . For

all sets of formulas $A \cup \{x\}$, we put $(A, x) \in deriv_R(G)$ iff $(a, x) \in deriv_R(G)$ for some conjunction a of formulas in A . Finally, $deriv_R(G, A) = \{x \mid (A, x) \in deriv_R(G)\}$.

In what follows, we shall write $deriv_{SDW}(G)$ instead of $deriv_{\{\text{ST, DT, WO}\}}(G)$ and similarly for other derivations. That is, each rule is abbreviated by its initial. Our first observation is that ST and WO cannot both be applied to infer trust.

Observation 3. If $R \supseteq \{\text{ST, WO}\}$, then $deriv_R(G, A) = \mathcal{L}$ for all inputs A .

Proof. By the inference rule T, then $(\top, \top) \in deriv_R(G)$. By ST, $(\top, \neg\top) \in deriv_R(G)$. For all formulas x , since $\neg\top \vdash x$, $(\top, x) \in deriv_R(G)$ by WO. Since $\top = \bigwedge \emptyset$, $(A, x) \in deriv_R(G)$, i.e., $x \in deriv_R(G, A)$. Therefore, $deriv_R(G, A) = \mathcal{L}$. \square

It is easy to see that DT is derivable from WO. Therefore, the next observation states that given WO, then DT is redundant for trust inference.

Observation 4. For all inputs A , we have: $deriv_W(G, A) = deriv_{DW}(G, A)$ and $deriv_{AW}(G, A) = deriv_{DAW}(G, A)$.

The next observation states that given ST, then DT and AND are interderivable.

Observation 5. For all inputs A , we have:

$$deriv_{SD}(G, A) = deriv_{SA}(G, A) = deriv_{SDA}(G, A).$$

Proof. It suffices to show that $deriv_{SD}(G) = deriv_{SDA}(G)$ and $deriv_{SA}(G) = deriv_{SDA}(G)$. For the former, it suffices to show that $deriv_{SD}(G)$ is closed under AND. Assume that (a, x) and (a, y) are in $deriv_{SD}(G)$. The following derivation shows that $(a, x \wedge y) \in deriv_{SD}(G)$:

$$\begin{array}{c} \text{ST} \frac{(a, x)}{(a, \neg x)} \quad \frac{(a, y)}{(a, \neg y)} \quad \text{ST} \\ \text{DT} \frac{\quad}{(a, \neg x \vee \neg y)} \\ \text{ST} \frac{\quad}{(a, \neg(\neg x \vee \neg y))} \\ \text{OEQ} \frac{\quad}{(a, x \wedge y)} \end{array}$$

Similarly, $deriv_{SA}(G) = deriv_{SDA}(G)$. \square

According to Observations 3, 4, and 5, there are at most eight mutually different and non-trivial I/O logics generated by the optional rules: $deriv_{\emptyset}$, $deriv_S$, $deriv_D$, $deriv_A$, $deriv_W$, $deriv_{SD}$, $deriv_{DA}$, and $deriv_{AW}$. The remainder of this section focuses on semantic characterization of the eight logics.

We start with $deriv_{AW}$. In (Makinson and Van Der Torre 2000), the derivation $deriv_1$ is defined such that $deriv_1(G)$ is the smallest set that contains G and is closed under T, SI, AND, and WO. The only difference between $deriv_1$ and our $deriv_{AW}$ is that OEQ does not appear in the definition of $deriv_1$. Nevertheless, it is easy to see that OEQ can be derived from WO. Thus, $deriv_1$ and $deriv_{AW}$ are the same.

Observation 6. $deriv_{AW}(G, A) = deriv_1(G, A)$ for all A .

It is established in (Makinson and Van Der Torre 2000) that $deriv_1(G, A) = Cn(G(Cn(A)))$, where $G(B) = \{x \mid (a, x) \in G \text{ for some } a \in B\}$ for all sets B of formulas. Therefore, the semantic characterization of $deriv_{AW}$ is straightforward.

Definition 7. For all inputs A , we put $out_{AW}(G, A) = Cn(G(Cn(A)))$. In addition, $out_{AW}(G) = \{(a, x) \mid x \in out_{AW}(G, a)\}$.¹

Proposition 8. $deriv_{AW}(G, A) = out_{AW}(G, A)$.

Next, we present the semantic characterization for out_\emptyset , out_S , out_D , out_A , out_W . To do this, we need some notations. For all sets B of formulas, let

- $Eq(B) = \{x \mid x \dashv\vdash \top \text{ or } \exists a \in B \text{ such that } a \dashv\vdash x\}$,
- $\overline{B} = \{\neg x \mid x \in B\}$,
- $Disj(B) = \{\bigvee B' \mid \emptyset \neq B' \subseteq B \text{ and } B' \text{ is finite}\}$,
- $Conj(B) = \{\bigwedge B' \mid \emptyset \neq B' \subseteq B \text{ and } B' \text{ is finite}\}$, and
- $Cn^-(B) = \{x \mid x \dashv\vdash \top \text{ or } \exists a \in B \text{ such that } a \vdash x\}$.

Definition 9. For all inputs A , we put:

$$\begin{aligned} out_\emptyset(G, A) &= Eq(G(Cn(A))) \\ out_S(G, A) &= Eq(G(Cn(A))) \cup \overline{Eq(G(Cn(A)))} \\ out_D(G, A) &= Eq(Disj(G(Cn(A)))) \\ out_A(G, A) &= Eq(Conj(G(Cn(A)))) \\ out_W(G, A) &= Cn^-(G(Cn(A))) \end{aligned}$$

For each $R \in \{\emptyset, S, D, A, W\}$, $out_R(G)$ is defined in the same way as before.

The proof of the next proposition is relatively simple and can be found in the extended version of this paper.

Proposition 10. For each $R \in \{\emptyset, S, D, A, W\}$,

$$deriv_R(G, A) = out_R(G, A) \text{ for all inputs } A.$$

The semantic characterization of $deriv_{SD}$ and $deriv_{DA}$ remains to be considered. Next, we consider $deriv_{SD}$. The proof of Proposition 12 can be found in the extended version of this paper.

Definition 11. For all inputs A , let $out_{SD}(G, A) =$

$$\{x \mid \exists A_1, \dots, A_n \subseteq G(Cn(A)) \cup \overline{G(Cn(A))} \text{ such that } n \geq 0, \text{ each } A_i \text{ is finite, and } x \dashv\vdash \bigvee_{1 \leq i \leq n} \bigwedge A_i\}.$$

Proposition 12. $deriv_{SD}(G, A) = out_{SD}(G, A)$.

Finally, we consider $deriv_{DA}$. The proof of Proposition 14 is in the extended version of this paper.

Definition 13. For all inputs A , let $out_{DA}(G, A) =$

$$\{x \mid \exists A_1, \dots, A_n \subseteq G(Cn(A)) \text{ such that } n \geq 1, \text{ each } A_i \text{ is finite, and } x \dashv\vdash \bigvee_{1 \leq i \leq n} \bigwedge A_i\}.$$

Proposition 14. $deriv_{DA}(G, A) = out_{DA}(G, A)$.

¹ $out_{AW}(G, a) = out_{AW}(G, \{a\})$.

We conclude this section with an example that illustrates the difference between the eight I/O logics.

Example 15. Let $G = \{(\top, x), (a, y)\}$ and $A = \{a\}$.

- $out_\emptyset(G, A) = Eq(x, y)$.
- $out_S(G, A) = Eq(x, y) \cup \overline{Eq(x, y)}$.
- $out_D(G, A) = Eq(x, y, x \vee y)$.
- $out_A(G, A) = Eq(x, y, x \wedge y)$.
- $out_W(G, A) = Cn^-(x, y)$.
- $out_{DA}(G, A) = Eq(x, y, x \vee y, x \wedge y)$.
- $out_{AW}(G, A) = Cn(x, y)$.
- $\neg(x \leftrightarrow y) \in out_{SD}(G, A)$.

Note that $\neg(x \leftrightarrow y)$ is not in $out_R(G, A)$ for all $R \neq SD$.

4 Filtering Information by Trust

In the last section, we proposed a family of I/O logics for trust inference. In this section, given a set A of informed statements, we investigate which information contained in A will be believed/accepted by a rational agent. For this, we propose a trust-based information filtering architecture. Our logical analysis shows that some trust inference rules are redundant for our mechanism.

Belief change has been studied extensively in, e.g., belief revision theory (Alchourrón, Gärdenfors, and Makinson 1985) and dynamic epistemic logic (van Ditmarsch, van der Hoek, and Kooi 2008). With both approaches, the incoming information is always completely accepted without scrutiny (in AGM belief revision, the success postulate is included, while in public announcement logic, the announced (propositional) formulas are always believed). However, when trust is considered in the process of belief change, not all incoming information will be believed because “trust plays the role of filtering out noisy information” (Liau 2003).

We present a simple information filtering architecture in the next definition. The core idea is that a piece of information will be believed if (and only if) it is implied by what is both informed and trusted.

Definition 16. Let $R \in \{\emptyset, S, D, A, W, SD, DA, AW\}$. The operation out_R^f is defined as follows.

$$out_R^f(G, A) = \begin{cases} Cn(\emptyset) & \text{if } A \text{ is inconsistent,} \\ Cn(out_R(G, A) \cap Cn(A)) & \text{otherwise.} \end{cases}$$

Note that the above definition ensures that nothing (except for tautologies) will be believed if the input is inconsistent. Let us illustrate the definition with an example:

Example 17. Let $G = \{(\top, x), (a, y)\}$ and $A = \{x, y\}$. Then $out_\emptyset(G, A) = Eq(x)$. Thus, $out_\emptyset^f(G, A) = Cn(x)$.

Next, we list some obvious properties of out_R^f : the rules T, OEQ, WO, DT, and AND are valid for out_R^f . A restricted form of SI is also valid whereas the ST rule is invalid.

Observation 18. Let out_R^f be a filtering operation. The following holds for all generating sets G and inputs A :

- $\top \in out_R^f(G, A)$.
- if $x \vdash y$, then $x \in out_R^f(G, A)$ implies $y \in out_R^f(G, A)$.

- if $x, y \in out_R^f(G, A)$, then $x \vee y, x \wedge y \in out_R^f(G, A)$.
- if $x \in out_R^f(G, A)$, then $x \in out_R^f(G, B)$, provided that $A \subseteq Cn(B)$ and B is consistent.
- $x \in out_R^f(G, A)$ implies that $\neg x \notin out_R^f(G, A)$.

The next three observations are about the redundancy of the AND rule for the information filtering architecture. This fact has been informally mentioned by Liao (2003). Our information filtering mechanism enables us to formally define the notion of “redundancy”, thus we obtain formal results regarding the redundancy of certain rules.

Observation 19. $out_A^f(G, A) = out_A^f(G, A)$.

Proof. We consider only the case when A is consistent. The inclusion $out_\emptyset^f(G, A) \subseteq out_A^f(G, A)$ follows directly from the fact that $out_\emptyset(G, A) \subseteq out_A(G, A)$. It remains to be shown that $out_A^f(G, A) \subseteq out_\emptyset^f(G, A)$. By the definition and the properties of Cn , it suffices to show that $out_A(G, A) \cap Cn(A) \subseteq Cn(out_\emptyset(G, A) \cap Cn(A))$. Let $x \in out_A(G, A) \cap Cn(A)$. If $x \Vdash \top$, then $x \in Cn(out_\emptyset(G, A) \cap Cn(A))$. Otherwise (by the definition of out_A), there must be a finite and non-empty $B \subseteq G(Cn(A))$ such that $x \Vdash \bigwedge B$. It is easy to see that $B \subseteq out_\emptyset(G, A) \cap Cn(A)$. Hence, $x \in Cn(out_\emptyset(G, A) \cap Cn(A))$. \square

Observation 20. $out_{DA}^f(G, A) = out_D^f(G, A)$.

Proof. We consider only the case when A is consistent. We show that $out_{DA}^f(G, A) \subseteq out_D^f(G, A)$ (the other direction is trivial). By the definition and the properties of Cn , it suffices to show that $out_{DA}(G, A) \cap Cn(A) \subseteq Cn(out_D(G, A) \cap Cn(A))$. Let $x \in out_{DA}(G, A) \cap Cn(A)$. The case $x \Vdash \top$ is trivial. We assume $x \not\Vdash \top$. By the definition of out_{DA} , there are finite and nonempty $A_1, \dots, A_n \subseteq G(Cn(A))$ ($n \geq 1$) such that $x \Vdash \bigvee_{1 \leq i \leq n} \bigwedge A_i$. Applying (the distribution law of) propositional logic, this can be rewritten as $x \Vdash \bigwedge_{1 \leq j \leq m} \bigvee B_j$ where each $B_j \subseteq G(Cn(A))$. For each B_j , it is easy to see that $\bigvee B_j \in Disj(G(Cn(A))) \subseteq out_D(G, A)$ and $\bigvee B_j \in Cn(A)$ (since $x \in Cn(A)$). Hence, $\{\bigvee B_j \mid 1 \leq j \leq m\} \subseteq out_D(G, A) \cap Cn(A)$. Thus, $x \in Cn(out_D(G, A) \cap Cn(A))$. \square

Observation 21. $out_{AW}^f(G, A) = out_W^f(G, A)$.

Proof. We consider only the case where A is consistent. We show that $out_{AW}^f(G, A) \subseteq out_W^f(G, A)$ (the other direction is trivial). By the definition and the properties of Cn , it suffices to show that $out_{AW}(G, A) \cap Cn(A) \subseteq Cn(out_W(G, A) \cap Cn(A))$. Let $x \in out_{AW}(G, A) \cap Cn(A)$. If $x \Vdash \top$, then $x \in Cn(out_W(G, A) \cap Cn(A))$. Otherwise, since $x \in out_{AW}(G, A) = Cn(G(Cn(A)))$, there must be $b_1, \dots, b_n \in G(Cn(A))$ ($n \geq 1$) such that $\bigwedge_{1 \leq i \leq n} b_i \vdash x$. Then $x \Vdash \bigwedge_{1 \leq i \leq n} (x \vee b_i)$. For each i , it is the case that $x \vee b_i \in out_W(G, A) = Cn^-(G(Cn(A)))$ and $x \vee b_i \in Cn(A)$ (since $x \in Cn(A)$). Hence, $x \in out_W^f(G, A) = Cn(out_W(G, A) \cap Cn(A))$. \square

According to Observations 19, 20, and 21, there are at most five different filtering logics: out_\emptyset^f , out_S^f , out_D^f , out_W^f , and out_{SD}^f . Next, we show with some examples that they are mutually different.

Example 22. Let $G = \{(\top, \neg a \wedge b), (\top, b), (\top, a \wedge \neg b)\}$ and $A = \{a\}$. We have:

- $out_\emptyset(G, A) = Eq(\neg a \wedge b, b, a \wedge \neg b)$.
Thus, $out_\emptyset^f(G, A) = Cn(\emptyset)$;
- $out_S(G, A) = Eq(\neg a \wedge b, a \vee \neg b, b, \neg b, a \wedge \neg b, \neg a \vee b, \perp)$.
Thus, $out_S^f(G, A) = Cn(a \vee \neg b)$.
- $out_D(G, A) = Eq(Disj(\neg a \wedge b, b, a \wedge \neg b))$.
Therefore, $out_D^f(G, A) = Cn(a \vee b)$.
- Since $a \in out_{SD}(G, A)$, $out_{SD}^f(G, A) = Cn(a)$.

The above example shows that out_\emptyset^f , out_S^f , out_D^f , and out_{SD}^f are different from one other. The independence of out_W^f remains to be shown.

Example 23. Let $G = \{(\top, a), (\top, b)\}$ and $A = \{a \vee c\}$. Then $out_W(G, A) = Cn^-(a, b)$. Therefore, $a \vee c \in out_W^f(G, A)$. However, $a \vee c \notin out_{SD}^f(G, A)$ because $a \vee c \notin out_{SD}(G, A)$. Therefore, $a \vee c$ is in none of $out_\emptyset^f(G, A)$, $out_S^f(G, A)$ or $out_D^f(G, A)$.

5 Belief Manipulation

In the previous section, we proposed a trust-based mechanism for the acceptance or rejection of information by rational agents. A natural question is whether the mechanism can be manipulated. In Example 1, for instance, the epidemiologist may want the public to believe that vaccination has the effect of reducing the spread of disease. In this case, it is meaningful to know whether there are any statements from the epidemiologist that could make the public believe that vaccination has such an effect. In this section, we investigate the problem of belief manipulation with the information filtering architecture. We show that certain information filtering operations are equivalently manipulable in the sense that the sets of manipulable formulas are the same. Furthermore, we characterize the set of manipulable formulas under various filtering operations.

Definition 24. Given a filtering operation out_R^f and a generating set G , a formula x is *manipulable* under G and out_R^f if there exists an input A such that $x \in out_R^f(G, A)$.

Example 25. Let $G = \{(\top, x), (a, y)\}$. The formula x is manipulable under G and out_\emptyset^f because $x \in out_\emptyset^f(G, x)$. On the other hand, $\neg x$ is not manipulable under G and out_\emptyset^f .

Before characterizing the set of manipulable formulas, we first explore the relationship between the sets of manipulable formulas under different filtering operations.

Definition 26. Let two filtering operations out_R^f and $out_{R'}^f$ be given. We say that out_R^f and $out_{R'}^f$ are *equivalently manipulable*, with the notation $out_R^f \equiv out_{R'}^f$, if for all generating sets G , the manipulable formulas under G and out_R^f are the same as those under G and $out_{R'}^f$.

In the last section, we saw that the AND rule is redundant for the information filtering architecture. Thus, the next observation is obvious.

Observation 27. $out_{\emptyset}^f \equiv out_A^f, out_D^f \equiv out_{DA}^f$, and $out_W^f \equiv out_{AW}^f$.

Next, we show that the DT rule is also redundant as far as the manipulable formulas are concerned. For this, we need the next lemma. Recall that a set V of formulas is “maximal consistent” if V is consistent and no proper superset of V is consistent.

Lemma 28. *Let out_R^f and G be given. A formula x ($x \not\vdash \top$) is manipulable under G and out_R^f iff there is a maximal consistent set V such that $x \in Cn(out_R(G, V) \cap V)$.*

Proof. The direction from right to left is trivial. Suppose x is manipulable and $x \not\vdash \top$. Then there exists an A such that $x \in out_R^f(G, A)$. Since $x \not\vdash \top$, A must be consistent. Therefore, $x \in Cn(out_R(G, A) \cap Cn(A))$. Note that, by the Lindenbaum lemma, there exists a maximal consistent set $V \supseteq A$. Since $out_R(G, A) \subseteq out_R(G, V)$ and $Cn(A) \subseteq Cn(V) = V$, $x \in Cn(out_R(G, V) \cap V)$. \square

Observation 29. $out_{\emptyset}^f \equiv out_D^f$.

Proof. We need to show that for all formulas x and generating sets G , x is manipulable under out_{\emptyset}^f and G iff x is manipulable under out_D^f and G .

The case of $x \vdash \top$ is trivial. We consider only the case of $x \not\vdash \top$. The direction from left to right follows from the fact that $out_{\emptyset}^f(G, A) \subseteq out_D^f(G, A)$ for any A . Suppose x is manipulable under out_D^f and G . By Lemma 28, there is a maximal consistent set V such that $x \in Cn(out_D(G, V) \cap V)$. To show that x is manipulable under out_{\emptyset}^f , it suffices to show that $Cn(out_D(G, V) \cap V) \subseteq Cn(out_{\emptyset}(G, V) \cap V)$. By the properties of Cn , it again suffices to show that $out_D(G, V) \cap V \subseteq Cn(out_{\emptyset}(G, V) \cap V)$. For each $y \in out_D(G, V) \cap V$, if $y \vdash \top$ then $y \in Cn(out_{\emptyset}(G, V) \cap V)$. Otherwise, $y \in V$ and there is a finite and nonempty $B \subseteq G(Cn(V))$ such that $y \vdash \bigvee B$. Since $y \in V$, then $\bigvee B \in V$. By the maximal consistent set property, there is a $b \in B$ such that $b \in V$. Note that $b \in G(Cn(V)) \subseteq out_{\emptyset}(G, V)$. Hence, $b \in out_{\emptyset}(G, V) \cap V$. Hence, $y \in Cn(out_{\emptyset}(G, V) \cap V)$. \square

Observation 30. $out_S^f \equiv out_{SD}^f$.

Proof. We need to show that for all formulas x and generating sets G , it holds that x is manipulable under out_S^f and G iff x is manipulable under out_{SD}^f and G .

Following the reasoning used in the proof of Observation 29, we only need to show that $out_{SD}(G, V) \cap V \subseteq Cn(out_S(G, V) \cap V)$ (for any maximal consistent set V). Let $y \in out_{SD}(G, V) \cap V$. Then there exist finite $A_1, \dots, A_n \subseteq G(Cn(V)) \cup \overline{G(Cn(V))}$ such that $y \vdash \bigvee_{1 \leq i \leq n} \bigwedge A_i$. Since y is consistent, $n \neq 0$. Since $y \in V$, $\bigvee_{1 \leq i \leq n} \bigwedge A_i \in V$.

By the maximal consistent set property, there is a $1 \leq j \leq n$ such that $\bigwedge A_j \in V$. Thus, $A_j \subseteq V$. On the other hand, since $A_j \subseteq G(Cn(V) \cup \overline{G(Cn(V))})$, then $A_j \subseteq out_S(G, V)$. Hence, $A_j \subseteq out_S(G, V) \cap V$. Note that $A_j \vdash y$. Therefore, $y \in Cn(out_S(G, V) \cap V)$. \square

According to Observations 27, 29, and 30, there are at most three filtering operations that are not equivalently manipulable: out_{\emptyset}^f , out_S^f , and out_W^f . The next two examples show that the sets of manipulable formulas under the three filtering operations are different to one another.

Example 31. To see $out_{\emptyset}^f \not\equiv out_S^f$, we employ the same G as in Example 25, i.e., $G = \{(\top, x), (a, y)\}$. We have seen that $\neg x$ is not manipulable under G and out_{\emptyset}^f . However, $\neg x \in out_S^f(G, A)$ where $A = \{\neg x\}$.

Example 32. Let $G = \{(p, \perp)\}$. The formula p is manipulable under G and out_W^f since $p \in out_W^f(G, p)$. However, p is not manipulable under out_{\emptyset}^f or out_S^f .

Example 32 shows that the manipulable formulas under out_{\emptyset}^f and out_W^f are different. But what is the exact relationship between them? The next observation answers that question. For any generating set G and formula x , let $G^x = \{(a, b \vee x) \mid (a, b) \in G\}$.

Observation 33. Let G be given. For all formulas x , it is the case that x is manipulable under G and out_W^f iff x is manipulable under G^x and out_{\emptyset}^f .

Proof. It suffices to show that for all inputs A ,

$$x \in out_W^f(G, A) \text{ iff } x \in out_{\emptyset}^f(G^x, A).$$

We consider only the case when A is consistent.

From left to right. Suppose $x \in out_W^f(G, A)$. Then, by definition, there are $a_1, \dots, a_n \in Cn^-(G(Cn(A))) \cap Cn(A)$ such that $\bigwedge_{1 \leq i \leq n} a_i \vdash x$. For each i , without loss of general-

ity, we assume that $a_i \not\vdash \top$. Since $a_i \in Cn^-(G(Cn(A)))$, there must be a $b_i \in G(Cn(A))$ such that $b_i \vdash a_i$. Then $\bigwedge_{1 \leq i \leq n} b_i \vdash x$. Therefore, $\bigwedge_{1 \leq i \leq n} (b_i \vee x) \vdash x$. Note that for each i , it is the case that $b_i \vee x \in G^x(Cn(A))$ and $b_i \vee x \in Cn(A)$. Hence, $x \in out_{\emptyset}^f(G^x, A)$.

From right to left. Since A is consistent, it suffices to show that $out_{\emptyset}(G^x, A) \subseteq out_W(G, A)$. Let $y \in out_{\emptyset}(G^x, A) = Eq(G^x(Cn(A)))$. We consider only the case of $y \not\vdash \top$. Then, there is a set $(a, b \vee x) \in G^x$ such that $a \in Cn(A)$ and $b \vee x \vdash y$. Since $(a, b) \in G$ and $b \vdash y$, then $y \in Cn^-(G(Cn(A))) = out_W(G, A)$. \square

We conclude this section by characterizing the manipulable formulas under out_{\emptyset}^f , out_S^f , and out_W^f . For all generating sets G , let $b(G) = \{a \mid (a, x) \in G\}$.

Proposition 34. *A formula x is manipulable under G and out_{\emptyset}^f (out_S^f , respectively) iff there are $B \subseteq b(G)$ and $H \subseteq G(B)$ ($H \subseteq G(B) \cup \overline{G(B)}$, respectively) such that $B \cup H$ consistent and $x \in Cn(H)$.*

Proof. We show only the case for out_{\emptyset}^f .

From right to left. We show that $x \in out_{\emptyset}^f(G, B \cup H) = Cn(out_{\emptyset}(G, B \cup H) \cap Cn(B \cup H))$. Since $x \in Cn(H)$, it suffices to show that $H \subseteq out_{\emptyset}(G, B \cup H)$ and $H \subseteq Cn(B \cup H)$. The former holds because $H \subseteq G(B) \subseteq out_{\emptyset}(G, B \cup H)$. The latter is trivial.

From left to right. Suppose x is manipulable. If $x \not\vdash \top$, we can assign $B = H = \emptyset$. Assume $x \not\vdash \top$. Then $x \in out_{\emptyset}^f(G, A)$ for some consistent A . Let $B = b(G) \cap Cn(A)$ and $H = G(B) \cap Cn(A)$. It is clear that $B \cup H$ is consistent. To show that $x \in Cn(H)$, it suffices to show that $(out_{\emptyset}(G, A) \cap Cn(A)) \subseteq Cn(H)$. Let $y \in out_{\emptyset}(G, A) \cap Cn(A)$ and $y \not\vdash \top$. Then there is a $z \in G(Cn(A))$ such that $z \not\vdash y$. Since $G(Cn(A)) = G(B)$, then $z \in G(B)$. Since $z \in Cn(A)$, then $z \in H$. Thus, $y \not\vdash z \in Cn(H)$. \square

Proposition 35. *A formula x is manipulable under G and $out_{AW}^f (= out_{W}^f)$ iff there is a $B \subseteq b(G)$ such that $B \cup \{x\}$ is consistent and $x \in Cn(G(B))$.*

Proof. From right to left. It suffices to show that $x \in out_{AW}^f(G, B \cup \{x\})$. Since $B \cup \{x\}$ is consistent, we only need to show that $x \in Cn(Cn(G(Cn(B \cup \{x\}))) \cap Cn(B \cup \{x\}))$. This holds because $x \in Cn(G(B))$ and $x \in Cn(B \cup \{x\})$. From left to right. Suppose x is manipulable. Then there exists an A such that $x \in out_{AW}^f(G, A)$. The case where A is inconsistent is trivial. If A is consistent, then $x \in Cn(Cn(G(Cn(A))) \cap Cn(A))$. Let $B = \{a \in b(G) \mid a \in Cn(A)\}$. It is clear that $B \cup \{x\}$ is consistent (since $B \cup \{x\} \subseteq Cn(A)$) and $G(B) = G(Cn(A))$. Therefore, $x \in Cn(G(B))$. \square

6 Related Work

Logic of Trust. Epistemic trust has been studied in the literature on logic, e.g., (Liau 2003; Dastani et al. 2005; Jiang and Naumov 2022). Most of them studied unconditional trust using modal logic whereas our paper generalizes trust to a conditional setting and employs the I/O logic framework. Compared with the modal logic approach, the I/O logic framework is more flexible in combining different inference rules. Nevertheless, the I/O language is also more restrictive (only propositional trust can be expressed). We expect that some of our I/O logics and filtering operations can be embedded into some modal logics of trust, e.g., in (Liau 2003).

The logics of trust are also studied in the belief revision literature, e.g., (Booth and Hunter 2018) and (Singleton and Booth 2022). Booth and Hunter suggest representing agents’ trust as a partition Π over the set of all valuations (states) on a given finite set of atoms. The semantic intuition is that the agent trusts the source to distinguish between states belonging to different cells in partition Π . Given an informed statement φ , the information filtered by Π is semantically represented by $\Pi(\varphi) = \bigcup\{\Pi(s) \mid s \models \varphi\}$, where s is a state, $\Pi(s)$ is the element of Π containing s , and \models is the satisfaction in classical propositional logic. We remark that this approach can be simulated using the operation out_{SD}^f .

But, due to space limitations, we cannot include the proof.

Belief Manipulation. Belief manipulation is concerned with the ability of one agent to convince another agent to hold certain beliefs (Hunter, Schwarzenruber, and Tsang 2017; Eiter, Hunter, and Schwarzenruber 2021). Similar notions, such as “knowability”, are also studied in dynamic epistemic logic (Balbiani et al. 2008). Strictly speaking, our definition of “manipulable formulas” does not correspond to the exact notion of “belief manipulation” because in our framework, agents’ belief states are not represented and no mechanism for belief change is included. Our emphasis is rather on manipulation during the trust-based information filtering process. We imagine that even if certain formulas cannot be manipulated in our information filtering architecture, one agent (the sender) is still able to make another (the receiver) believe the formulas. To model this, we need to extend our framework with belief revision theory or similar.

Input/Output Logic. I/O logic was originally introduced in (Makinson and Van der Torre 2000). Although I/O logic is an influential framework for normative reasoning, it can also be applied to other domains such as causal reasoning and nonmonotonic logic (Makinson and Van der Torre 2000; Bochman 2005). In the literature, I/O logics without WO have been studied, see (Stolpe 2008) and (Parent and Van der Torre 2019). But, in both (Stolpe 2008) and (Parent and Van der Torre 2019), the AND rule is included. I/O logics without WO and AND are considered in (Farjami 2020) albeit in an algebraic setting.

7 Conclusions and Future Work

Trust can be inferred from other information and plays the role of filtering out noisy information. In this paper, we first developed a family of new I/O logics for trust inference and then presented an information filtering architecture based on trust inference. We also investigated the manipulation problem and we characterized the manipulable formulas in the information filtering architecture. Our main results (see Table 1) concern the redundancy of inference rules for trust. AND is redundant for the information-filtering mechanism. Furthermore, both AND and DT are redundant for belief manipulation.

For our future work, we plan to study the redundancy problem for other inference rules in I/O logic. We may also study a conditional notion of manipulability — we may want to convince others to believe something while not revealing secret information. Finally, we can extend our architecture with belief revision operations.

| | | | | | | | |
|---|-----------------------------|-----------------------------|--------------|------------------------|------------|---------|------------|
| out_{\emptyset} | out_A | out_D | out_{DA} | out_S | out_{SD} | out_W | out_{AW} |
| $out_{\emptyset} = out_A^f$ | $out_D^f = out_{DA}^f$ | out_S^f | out_{SD}^f | $out_W^f = out_{AW}^f$ | | | |
| $out_{\emptyset} \equiv out_A^f \equiv out_D^f \equiv out_{DA}^f$ | $out_S^f \equiv out_{SD}^f$ | $out_W^f \equiv out_{AW}^f$ | | | | | |

Table 1: Summary of main results.

Acknowledgments

We thank the anonymous reviewer for their comments. This work is supported by the Luxembourg National Research Fund (FNR) through the following projects: The Epistemology of AI Systems (EAI) (C22/SC/17111440), DJ4ME – A DJ for Machine Ethics: the Dialogue Jiminy (O24/18989918/DJ4ME), Logical Methods for Deontic Explanations (LoDEx) (INTER/DFG/23/17415164/LoDEx), and Symbolic and Explainable Regulatory AI for Finance Innovation (SERAFIN) (C24/19003061/SERAFIN).

References

- Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *The Journal of Symbolic Logic*, 50(2): 510–530.
- Balbiani, P.; Baltag, A.; van Ditmarsch, H.; Herzig, A.; Hoshi, T.; and de Lima, T. 2008. ‘Knowable’ as ‘Known after an Announcement’. *The Review of Symbolic Logic*, 1(3): 305–334.
- Becker, M. Y.; Russo, A.; and Sultana, N. 2012. Foundations of Logic-Based Trust Management. In *2012 IEEE Symposium on Security and Privacy*, 161–175.
- Bochman, A. 2005. *Explanatory Nonmonotonic Reasoning*. WORLD SCIENTIFIC.
- Booth, R.; and Hunter, A. 2018. Trust as a precursor to belief revision. *Journal of Artificial Intelligence Research*, 61: 699–722.
- Dastani, M.; Herzig, A.; Hulstijn, J.; and van der Torre, L. 2005. Inferring Trust. In Leite, J.; and Torroni, P., eds., *Computational Logic in Multi-Agent Systems*, 144–160. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-31857-6.
- Eiter, T.; Hunter, A.; and Schwarzentruher, F. 2021. How Hard to Tell? Complexity of Belief Manipulation Through Propositional Announcements. In Zhou, Z.-H., ed., *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 1866–1872. International Joint Conferences on Artificial Intelligence Organization. Main Track.
- Farjami, A. 2020. *Discursive Input/output Logic: Deontic Modals, and Computation*. Ph.D. thesis, University of Luxembourg.
- Hunter, A.; Schwarzentruher, F.; and Tsang, E. 2017. Belief Manipulation Through Propositional Announcements. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 1109–1115.
- Jiang, J.; and Naumov, P. 2022. In Data We Trust: The Logic of Trust-Based Beliefs. In Raedt, L. D., ed., *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 2683–2689. International Joint Conferences on Artificial Intelligence Organization. Main Track.
- Leturc, C.; and Bonnet, G. 2018. A Normal Modal Logic for Trust in the Sincerity. In *Proceedings of the 17th International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS ’18, 175–183. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
- Liau, C.-J. 2003. Belief, information acquisition, and trust in multi-agent systems – A modal logic formulation. *Artificial Intelligence*, 149(1): 31–60.
- Makinson, D.; and Van Der Torre, L. 2000. Input/output logics. *Journal of philosophical logic*, 29: 383–408.
- Parent, X.; and van der Torre, L. 2019. Input/output logics without weakening. *Filosofiska Notise*, 6(1): 189–209.
- Singleton, J.; and Booth, R. 2022. Who’s the Expert? On Multi-source Belief Change. In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning*, 331–340.
- Stolpe, A. 2008. Normative Consequence: The Problem of Keeping It Whilst Giving It up. In van der Meyden, R.; and van der Torre, L., eds., *Deontic Logic in Computer Science*, 174–188. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-70525-3.
- van Ditmarsch, H.; van der Hoek, W.; and Kooi, B. 2008. *Dynamic Epistemic Logic*. Springer, Dordrecht.