A Description Language for Similarity, Belief Change and Trust

Aaron Hunter
Department of Computing, British Columbia Institute of Technology, Burnaby, Canada

Keywords: Trust, Belief Revision, Knowledge Representation.

Abstract: We introduce a simple framework for describing and reasoning about situations where an agent receives information reported from external sources, and these reports cause them to change their beliefs. Our framework is inspired by classic action description languages, which use sets of causal statments to specify action effects in terms of transition systems. We suggest that this style of language can effectively capture important properties of similarity and trust, which are required to perform belief revision in practical settings. The language introduced in this paper allows us to specify a similarity relation on states, and it also allows us to explicitly associate an incoming report with a specific formula to be used as the input for a suitable belief revision operator. The result is a flexible framework that can describe a variety of belief change functions, and it can also capture the trust that is held in the reporting agent in a simple and transparent way. We demonstrate the connection with existing trust-influenced models of belief change. We then consider a speculative application where we apply our framework to reason about the correctness of trusted third party protocols. Directions for future work are considered.

1 INTRODUCTION

There is a tradition in formal Knowledge Representation in which compact logic-based languages are used for reasoning about the effects of actions. One influential class of such languages includes the so-called *action languages* (Baral and Gelfond, 1997; Baral et al., 1997). Such languages filled an important role in the literature of the era, by giving declarative representations of important problems in a formal setting. Over time, action languages have become less popular as more sophisticated action formalisms have proven to have greater utility. However, we suggest that this style of formalism can still be valuable to explicitly describe distinct aspects of reasoning. Specifically, we propose that a simple declarative approach can be valuable in specifying the interaction between between trust and belief when agents exchange informative messages.

In this paper, we propose a new language that uses the basic action language framework to describe both similarity between worlds and trust in reports. In this manner, we introduce a simple formal framework that allows for representing and reasoning about communicative actions that selectively impact beliefs based on the knowledge of a reporting agent.

This work makes several contributions to the literature on belief change and trust. The framework introduced provides a mechanism for explicitly specifying a similarity ordering on states, which can be used to define a belief revision operator. At the same time, the framework models knowledge-based trust, by explicitly specifying the connection between reports and belief revision. For example, a report of $\phi \land \psi$ might only cause an agent to revise by ψ in cases where the reporting agent is not trusted to know the truth of ϕ. The final contribution of this paper is really a speculative position for future investigation. We propose that the framework introduced could actually be a useful tool for reasoning about the security of trusted third party protocols. This application is described, but we leave a full treatment of the problem for future work.

2 PRELIMINARIES

Throughout this paper, we are concerned with formal approaches to Knowledge Representation. As such, we assume knowledge of propositional logic as a starting point. We review some basic terminology. A propositional signature $\mathbf{P}$ is a set of propositional variables that can be true or false. A single propositional variable is called an *atomic formula*. A *formula* of propositional logic is defined using the usual logical connectives $\neg$, $\land$, $\lor$. A *literal* is either $p \in \mathbf{P}$ or $\neg p$ where $p \in \mathbf{P}$.

Hunter, A.
A Description Language for Similarity, Belief Change and Trust.
DOI: 10.5220/0012406600003636

In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 3*, pages 870-877
ISBN: 978-989-758-680-4; ISSN: 2184-433X

A propositional interpretation for **P** assigns each variable in **P** a true/false value. We also use the term *state* to refer to a propositional interpretation and we let $2^{\mathbf{P}}$ denote the set of all propositional interpretations. We will also be concerned with beliefs and belief change in this paper. We define a *belief state* to be a set of states; informally the belief state K represents the beliefs of an agent who believes that the actual state of the world must be one of the elements of K.

Broadly, in logical approaches to Knowledge Representation, we think of atomic formulas as representing properties of the world. So a variable like *Rain* might be used as such: it is true in states where it is raining, and it is false in states where it is not raining. In this manner, we are able to represent incomplete beliefs. A belief state that includes some states where *Rain* is true and some states where *Rain* is false will be used to capture an agent's uncertainty about whether or not it is raining.

Finally, as a general statement, throughout this paper we will often be concerned with the notion of trust. We will consider formal notions of trust later, but for now we make a simple disclaimer. When we say that an agent is *trusted over* ϕ in this paper, we simply mean that we will believe ϕ if that agent tells us ϕ is true. However, we informally are thinking about this in terms of perceived knowledge. In other words, we trust an agent on some fact just in case we believe that they have the requisite knowledge to know when that fact is true. This is different from the notion of trust due to honesty. When dealing with honesty, we need to consider the idea that a particular agent may be intentionally deceptive. This introduces different problems that we do address in this paper.

2.1 Action Languages

We briefly describe the action language $\mathcal{A}$. We assume an underlying propositional signature, as well as an underlying set of *action symbols*. A sentence of the action language $\mathcal{A}$ has the form:

$$A \textbf{ causes } L \textbf{ if } P$$

where A is an action, L is a literal and P is a propositional formula. Following the terminology used in the area, we will sometimes refer to atomic formulas as *fluent symbols*; this just reinforces the fact that the truth value of an atomic formula can be changed as actions are executed.

A set of sentences of $\mathcal{A}$ defines a transition system (Gelfond and Lifschitz, 1998). A transition system is simply a graph where the nodes are labelled with states and the edges are labelled with actions. The semantics of $\mathcal{A}$ dictates that for any set S of sentences, the associated transition system will include an edge labelled with A from s_1 to s_2 just in case

- $s_1 \models P$ and$s_2 \models$ Ł
- For all atomic formulas Q that do not occur in L, $s_1 \models Q$ iff $s_2 \models Q$.

Hence, a set of causal sentences serves a single purpose: it describes how the state of the world changes when actions are executed. One advantage of an action language is that it gives a compact, declarative description of action effects that is easy to read and understand.

2.2 Belief Change

Belief revision refers to the process where an agent receives new information, and has to incorporate it with their current beliefs. One important approach is the AGM approach to belief revision is the AGM approach. In the AGM approach, a *belief revision operator* is a function $*$ that maps a belief state K and a formula ϕ to a new belief state $K * \phi$. We say that $*$ is an AGM revision operator if it satisfies a certain set of rationality postulates, which are normally called the AGM postulates. We do not list the postulates here, but instead refer the reader to (Alchourrón et al., 1985) for a complete description of the framework.

While AGM revision operators are defined in terms of a set of rationality postulates, it has also been shown that there is an equivalent semantic characterization. In particular, it has been shown that an operator satisfies the AGM postulates just in case there is a function f that maps each initial belief state K to a total pre-order $\preceq_K$ over states such that $K * \phi$ is the set of minimal states in $\preceq_K$ that satisfy ϕ. In the literature, the function f is called a *faithful assignment*(Katsuno and Mendelzon, 1992).

We can think of the ordering $\preceq_K$ as a plausibility ordering, where a state precedes another if it is considered to be more plausible. The important point for our purposes is that we can determine the outcome of AGM revision by finding all states consistent with the new information that are minimal with respect to the total pre-order.

2.3 Trust

In the original approaches to belief revision, the new information always had to be incorporated into the new belief state. In other words, following revision by ϕ, the underlying agent would always believe ϕ. Of course, in many practical settings, this is not a reasonable assumption; we should only believe the new information if the source of the information is *trusted.* There has been work on integrating trust into belief revision both in the setting of AGM revision (Booth and Hunter, 2018) and also in the setting of modal logics of belief (Liu and Lorini, 2017).

In the present paper, we will introduce sentences that indicate when an action causes an agent to believe a particular formula ϕ. This is how we capture trust: by explicitly specifying when the actions of another agent cause us to believe certain formulas. If we think of the actions as announcements, then we are able to explicitly specify when we believe the information announced by another agent. This is one key aspect of trust, which is important for many applications.

3 DESCRIBING BELIEF CHANGE

3.1 Similarity Descriptions

In this section, we present a simple action language style approach for giving concise descriptions of the similarity between states.

We use the term *similarity description language* to refer to a language that is used to describe the similarity between propositional interpretations. In the following definition, we introduce the similarity description language $\mathcal{D}$.

Definition 1. *A proposition of the similarity description language $\mathcal{D}$ is an expression of the form*

$$\textbf{if } \phi \textbf{ then } \psi \textbf{ adds dissimilarity } i,$$

where ϕ, ψ are conjunctions of literals and $i \in \mathbf{Z}^{\geq 1}$.

In the rule presented in the definition, we refer to ϕ as the head, we refer to ψ as the body, and we refer to i as the increment. A set of propositions of $\mathcal{D}$ is called a *similarity description.*

The semantics of $\mathcal{D}$ is given by associating a distance function d with every similarity description.

Definition 2. *Let SD be a similarity description. The distance function $d_{SD} : 2^{\mathbf{F}} \times 2^{\mathbf{F}} \to \mathbf{Z}^{\geq 0}$ is defined as follows.*

1. $d_{SD}(w,w) = 0$
2. $d_{SD}(w,v) = \sum_{i \in I}$,
   where I is the set of positive integers that appear in propositions in SD of the form

   $$\textbf{if } \phi \textbf{ then } \psi \textbf{ adds dissimilarity } i$$

   where $w \models \phi$ and $v \models \psi$.

Hence, the distance between w and v is calculated by taking the sum of all the increments with heads satisfied by w and bodies satisfied by v.

Proposition 1. *Let $d : 2^{\mathbf{F}} \times 2^{\mathbf{F}} \to \mathbf{Z}^{\geq 0}$ such that $d(w,w) = 0$ for all w. Then $d = d_{SD}$ for some similarity description SD.*

Proof. For each state w, let ϕ_w be the unique conjunction of literals such that, for each atomic formula F:

- F is a conjunct in ϕ_w if F is true in w.
- $\neg F$ is a conjunct in ϕ_w if F is false in w.

Then define SD as follows. For each pair w, v states, SD contains the sentence:

$$\textbf{if } \phi_w \textbf{ then } \phi_v \textbf{ adds dissimilarity } d(w,v).$$

It follows that $d = d_S D$. □

In general, the function d defined by a similarity description does not satisfy the properties that we expect to hold for a distance measure. For example, it need not be transitive and it need not satisfy the triangle inequality. So it is not a distance in the usual sense of the word; it is just a function that gives a natural number output between points. Nevertheless, we suggest that this language does allows for the compact representation of some natural distance functions.

We illustrate some examples that demonstrate particular distance functions that we can capture.

Example 1. *Let* **val** $: \mathbf{F} \to \mathbf{Z}^{\geq 0}$*, so* **val** *is a function that maps every fluent symbol F to some positive integer. Intuitively, we think of* **val** *as assigning some measure of subjective importance to each fluent symbol. The similarity description SD*(**val**) *is defined as follows. For each fluent symbol F, SD*(**val**) *contains the propositions*

$$\textbf{if } F \textbf{ then } \neg F \textbf{ adds dissimilarity } val(F)$$
$$\textbf{if } \neg F \textbf{ then } F \textbf{ adds dissimilarity } val(F)$$

Note that, if **val** *uniformly maps every fluent symbol to 1, then the distance associated with SD*(**val**) *is the Hamming distance between interpretations.*

Example 2. *Suppose that there is some distinguished fluent symbol $U \in \mathbf{F}$, and* **val** *is defined as follows*

$$val(U) = 2^{|\mathbf{F}|}$$
$$val(F \neq U) = 1.$$

In this case, the distance function associated with SD(**val**) *assigns large distances between worlds that disagree on U. We can think of U as a fluent symbol that is very unlikely to change under revision. This weighting captures a parametric difference operator, where one variable is more resistant to change (Peppas and Williams, 2018).*

3.2 A Report Description Language

An *epistemic action signature* is a pair $\langle \mathbf{F}, \mathbf{R} \rangle$ where $\mathbf{F}$ is a propositional signature, and $\mathbf{R}$ is a designated set of $\mathbf{F}$-formulas called *reports*. Informally, these are messages that can be received from some underlying source of information.

We now define a new kind of description language $\mathcal{A}_D$, which we will call a *report description language*.

Definition 3. *A* theory *of the language $\mathcal{A}_D$ is a set of sentences of one of the following forms:*

1. α **causes to believe** ϕ

2. **if** ϕ **then** γ **adds dissimilarity** i

where ϕ, ψ, γ are conjunctions of literals, α is a report, and $i \in \mathbf{Z}^{\geq 1}$. A theory is constrained to have at most one rule of form (1) for each report (up to logical equivalence).

The important feature of the language $\mathcal{A}_D$ is that it describes how reported information will be incorporated into the beliefs of some underlying agent. The following definitions demonstrate how this is done.

Definition 4. *For any theory T of $\mathcal{A}_D$, let $T(D)$ denote the subset of T that consists of all similarity propositions in T. Let d_T denote the distance function defined by $T(D)$.*

Hence, d_T is the distance function defined by the restriction of T to just the similarity sentences. In the following definition, we show how a theory of $\mathcal{A}_D$ can define a revision operator $*$.

Definition 5. *Let T be a theory of $\mathcal{A}_D$. The function $* : 2^S \times \mathbf{R} \rightarrow 2^S$ is defined such that $K * \alpha$ is defined as follows. If (*α **causes to believe** ϕ*) is in T, then $K * \alpha$ is equal to:*

$$\{s \mid s \models \phi \textit{ and for some } w \in K, d(s,w) \textit{ is minimal}\}$$

If there is no such rule with head α in T, then $*$ is the identity function.

Hence, $K * \alpha$ is the set of ϕ-states that are minimally distant from states in K according to the distance function defined by the description. This definition is similar to the definition of distance-based revision found in (Delgrande, 2004), where it is shown that the operator satisfies the AGM postulates under some natural restrictions. But note that we are not revising by α; we are revising only by the formula ϕ that α causes us to believe. In this manner, we are capturing partial trust in the information source. We return to this point in section 4.

3.3 Basic Properties

The report description language $\mathcal{A}_D$ is very flexible, to the point that it is hard to give general properties without restricting the form of theories. Nevertheless, we can give some basic results. We start with the extreme cases.

Proposition 2. *If T is a theory that contains* (α **causes to believe** ϕ) *but it contains no similarity sentences, then $K * \alpha = mod(\phi)$ for all $K \subseteq S$.*

Proof. Since there are no similarity sentences in T, $d_T(w,v) = 0$ for all w, v. So $K * \alpha$ is just the set of models of ϕ. □

On the other hand, we can also consider the case where there are no causal sentences.

Proposition 3. *If T is a theory with no causal sentences, then $K * \phi = K$ for all $K \subseteq S$.*

Proof. If T contains no causal sentences, then there is no effect for any report. Hence there is no change in belief. □

Of course, in between these extreme cases, there are more interesting situations. For any set K of states and any distance function d, define the relation $\preceq_{K,d}$ such that $t_1 \preceq_{K,d} t_s$ just in case the minimum distance from t_1 to K is less than the minimum distance from t_2 to K.

Proposition 4. *Let T be a theory containing the sentence*

ϕ **causes to believe** ϕ

for all formulas ϕ. Let d be the function defined by $T(D)$. If, for each K, $\preceq_{K,d}$ is a total pre-order, then $*$ is an AGM revision operator.

Proof. We can define $*$ to be the revision operator defined by the faithful assignment $K \mapsto \preceq_(K, d)$. □

Hence, if every formula ϕ is taken as evidence of ϕ, then we can define AGM revision operators by carefully specifying the similarity relation to ensure we have a faithful assignment.

From a high-level perspective, the important point here is that the description language we have defined can do two things independently. First, it can be used to specify a similarity relation that is useful for defining revision operators. But independently, the language can be used to specify a relationship between reports and believed outcomes; this is done through the causes-to-believe sentences. This allows us to define situations where the formulas we believe following a report may not be identical to the reports themselves. However, at present, we do not have any formal restriction on this latter connection. The relationship between reports and believed outcomes is completely flexible.

4 REPORTS AND TRUST

4.1 Separating Trust and Similarity

If we just look at the dissimilarity sentences, a theory T of $\mathcal{A}_D$ actually defines an *idealized* belief change operator through distance-based revision. But that is not the operator $*$ that we have associated with T. The operator $*$ is obtained from the idealized operator, but filtered through the trust that is explicitly specified in the causal sentences. If we had several information sources, we could define causal sentences for each. This would give several trust-based operators, based on the same underlying distance function. For now, we stick with a single source.

We look at some examples.

Example 3. *Suppose that we have a rule of this form:*

Rain **causes to believe** *Rain*.

This indicates that the information source is trusted to determine when it is raining. However, suppose that the following rules are not included:

CarBroken **causes to believe** *CarBroken*.
$\neg$*Rain* **causes to believe** $\neg$*Rain*.

This means that we would not trust them when they tell us that our car is broken. Moreover, we would not even trust them if they said it was not *raining.*

In the preceding example, we show how we can filter out the part of the report on which an agent is not trusted. The next example shows how we can represent ignorance.

Example 4. *Consider rules of the form:*

Rain $\wedge$ *CarBroken* **causes to believe** *Rain*
Snow **causes to believe** *Rain*

In this case, we essentially are treating the reporting agent as if they can not tell the difference between different kinds of precipitation. Whether they report snow or rain, we always believe it is raining. Moreover, we do not trust them with respect to the information about our car.

These examples provide some illustrative cases, where our description language can be used to capture interesting relationships between trust and belief change.

4.2 Trust-Influenced Belief Revision

In the literature, several approaches to trust-influenced belief change have appeared. In this section, we briefly show how our work is connected to one of these approaches. Specifically, we look at the so-called trust-senstive belief revision operators of (Booth and Hunter, 2018). In this framework, there is an idealized revision operator, similar to the one defined by the similarity sentences in our setting. Trust-sensitive revision operators are defined with respect to a partition over possible states; we only trust an agent to distinguish between states that are in different partition cells. Hence, there is a division between the specification of trust and the underlying revision operator.

Definition 6. *Let* Π *be a partition over states. Define* B_Π *to be the set of cause-to-believe sentences that includes*

α **causes to believe** ϕ

just in case α *and* ϕ *are maximal conjunctions of literals specifiying states* s *and* t *where* s *and* t *are in different cells of* Π.

Hence, we can use a partition to define a set of causal sentences that captures the division used to define trust-sensitive revision. The following result is straightforward.

Proposition 5. *Let* $*_\Pi$ *be the trust sensitive revision operator, defined the idealized revision operator* $*$ *and the state partition* Π. *If we let* T *consist of the union of* B_Π *and the similarity description SD defining* $*$, *then* $*_\Pi$ *is the revision operator defined by* T.

The point here is that the separation of the trust partition and the revision operator allows us to do both parts independently.

In general, this kind of manual construction can flexibly allow us to specify a variety of trust relationships with the reporting agent.

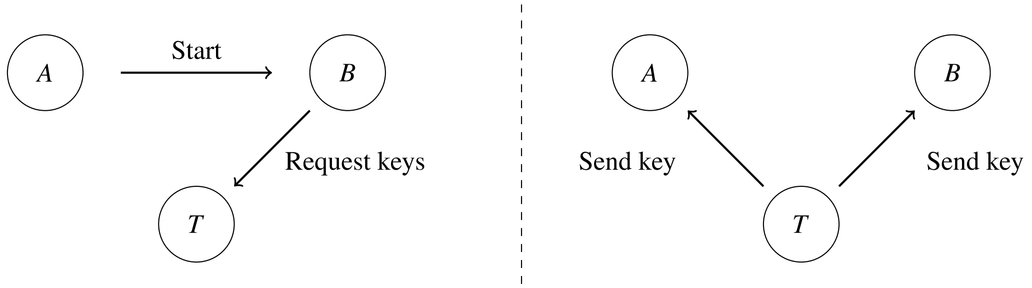
Proposition 6. *Let* P *be a set of formulas. Then there is a theory* T *such that, for each* $\psi \in P$, *if* $\phi \models \psi$ *then* $K * \phi \models \psi$.

So we can define a theory where an agent is trusted just in case they report a formula in the set P; this is similar to the model of trust specified in (Liu and Lorini, 2017). In fact, the proposition ensures we will believe $\psi \in P$ whenever the agent reports something that entails ψ.

5 APPLICATION: TRUSTED THIRD PARTIES

5.1 Protocol Verification

We propose that our framework can be useful for protocol verification. This is an area where logics of belief have been applied in the past, starting with the



Initialization Steps

Sharing Keys

Figure 1: Simple Key Agreement Protocol.

hightly influential work on BAN logic (Burrows et al., 1990). The idea for logic-based protocol verification is to formalize the protocol in terms of the beliefs of the participants. In an authentication protocol, for example, the goal is to prove that some agent has a particular identity.

There are two advantages to using a logic-based approach to protocol verification. The first advantage is that we can give a declarative description of the problem, and then simply consider the beliefs of each agent after the protocols is executed. Hence, the fact that logics are easy to read and understand is a benefit in this context.

The other advantage of a logic-based approach is that it permits precise proofs of correctness for protocols. We know that many protocols are vulnerable to subtle attacks that are hard to predict in advance. As such, the best way to verify the correctness of a protocol is through a formal proof of correctness. In practice, of course, formalizing a protocol in a logic is difficult. Moreover, we need to make some assumptions about the message passing environment that are not always accurate. So this approach to protocol verification is not perfect; but it is one tool that has been used to prove protocol correctness in the past.

For the present paper, we admit that the application to protocol verification is somewhat speculative. We are interested in demonstrating the power of our description language by showing that certain protocols can be modelled and verified. Our focus is specifically on protocols where trust plays an explicit role, as these protocols are very challenging to verify through traditional methods.

5.2 Trusted Third Party Protocols

In network communication, a trusted third party (TTP) is an agent that participates in a protocol to ensure the other parties that the information exchanged is correct. We describe a simple protocol. The protocol involves the exchange of messages between three parties: A, B and T. In this protocol, T is acting as a TTP to allow A and B to establish a session key. This notion has been discussed in (D. Zissis and Koutsabasis, 2011; A. Ulrich and Carle, 2011).

We use the standard notation established in (Burrows et al., 1990) to describe the protocol:

Simple Key Agreement
1. $A \rightarrow B : N_A, A$
2. $B \rightarrow T : N_A, N_B, A, B$
3. $T \rightarrow A : \{K\}_{K_A}$
4. $T \rightarrow B : \{K\}_{K_B}$

In this notation, $A \rightarrow B : M$ means that A sends the message M to the agent B. An expression of the form $\{M\}_K$ denotes the message M encrypted with the key K. Messages of the form N_A are *nonces*, which are random numbers generated at the time of protocol execution. In this protocol, T is a trusted party that is responsible distributing sessions keys for communication between agents. We assume that T shares a secret key with A which is denoted by K_A, as well as a secret key with B which is denoted by K_B. The goal of this protocol is to give A and B a new key that they can use for secure communication. A graphical representation of the protocol is provided in Figure 1.

Proving that this kind of protocol actually works can be difficult. There are at least two challenges. The first is a question of the honesty of T, which we will not address here. Instead, we focus on the problem of knowledge. Why should A and B believe that T has a suitable collection of keys available, which are each secure? This problem requires an analysis of the beliefs of A and B, and how they change when information is exchanged on the network.

5.3 Towards a Formalization

We put aside for the moment the manner in which a TTP might be established. In practice, this might be done through extra-logical means. However, we can give a precise statement of what this means for the agents participating in a protocol.

To prove that a TTP protocol is correct, we simply need to encode the protocol as a set of logical formulas. Consider the Simple Key Agreement protocol from the previous section. In order to show that this protocol is correct, one would need to perform the following steps.

- Formalize the protocol as a sequence of announcements $P_1,\dots,P_n$, made respectively by agents $a_1,\dots,a_n$.
- Formalize the goal of the protocol as another formula G.
- Prove that the goal is believed after the protocol, if we assume the third party is rightly trusted.

This is an established method for verifying cryptographic protocols in epistemic logic. We propose that we can define a variation of this approach for TTP protocols, using our description language.

In order to formalize the Simple Key Agreement protocol at a high level, we assume the propositional vocabulary includes atomic formulas of the form $init(x)$ for $x \in \{A,B\}$. These are formulas that are true when A (resp. B) want to initialize a communication session. We then assume that we have a finite set K of keys. For each key $k \in K$ and each pair of agents x,y we have an atomic formula of the form $\{safe(k,x,y)\}$. Such a formula is true when k is a safe key for communication between x and y.

The Simple Key Agreement protocol can be represented as a sequence of messages exchanged. Each message causes revision by some formula. In order to formalize this protocol in $\mathcal{A}_D$, we need to do three things:

1. Formalize the connection between messages sent and the resulting belief change. This is done through causal sentences. For example, sending message (1) of the protocol should simply cause B to belief that A would like to start a run of the protocol. This kind of arbitrary connection between formulas and belief change can easily be specified in $\mathcal{A}_D$.
2. The causal sentences should also assert that the TTP is trusted on everything they say that is related to the protocol run. This is a critical component.
3. Formalize the underlying belief revision operator using similarity sentences. This is where we capture notions that are general to the application. For example, the connection between agents and keys; this impacts our idealized revision when messages are received.

After specifying all of these components, we are able to specify the stucture of the protocol through report actions for each agent. For agent A, there are just two revisions:

1. A sends M_1 to B.
2. T sends M_4 to A.

From the perspective of A, the protocol is correct if these two revisions cause them to believe that they have a session key that is secure for communication with B. Of course, for this particular protocol, this is not going to be provable. The problem is that there is no connection between the first two messages sent and the second two messages sent; so there is no guarantee that agent a will believe the key they are given is sent from a current run of the protocol.

We conclude the description of this application with one final remark. One might object to our treatment here by saying that we need nested beliefs to reason about protocol verification. For example, we might need A to have beliefs about the beliefs of B. In the present framework, we do not have the capacity to capture nested beliefs in this manner. However, for protocol verification, we generally only require limited (finitely bounded) nesting of belief. It would be possible to address this problem by carefully extending the propositional vocabularly to include formulas about the beliefs of other agents. This kind of approach can be sufficient for the kind of reasoning required for concrete protocols.

We leave a complete treatment of trusted third party protocols in our framework for future work.

6 DISCUSSION

The most closely related work to this paper is the work on using causal rules to model trust and belief change (Hunter, 2021b). In this work, causal rules similar to sentences of $\mathcal{A}$ are used to reason about change in modal logic. In other words, the causal rules specify when an action impacts the truth of a formula like $\Box\phi$. If we interpret the $\Box$ to represent belief, then this is similar to our approach here. However, in the present work, we do not consider modal logics at all; our work is all in the propositional setting of the AGM framework. Moreover, our focus in this paper is more broad; we are not only concerned with how actions impact belief; we are also concerned with giving an simple language for defining similarity relationships.

One important feature of the description language presented that should be emphasized is the connection with trust-sensitive revision, as outlined in section 4. The fact that we can capture an established approach to trust and belief revision illustrates the potential utility of our approach. Moreover, our approach is extremely flexible. Hence, we can easily capture variations on this model of trust by simply imposing different restrictions on our similarity descriptions.

7 CONCLUSION

In this paper, we have introduced a description language for belief change and trust. Our language is based on classic action languages, but it does not involve actions. Instead, it defines similarity between states and connections between reports and belief. The result is a simple language that can flexibly describe belief change where information comes from a partially trusted source.

While this work is inspired by an old tradition in reasoning about action, there has been recent work on related topics. Notably, the language introduced in (Hunter, 2021a) gives a model for reasoning about changes in belief. This work also has connections with a variety of approaches to trust, including both those based on sets of formulas or those based on semantic constraints.

In terms of future work, there are three main directions. The first is characterizing how different belief change postulates can be compactly captured. The second is explicitly specifying how to encode different trust relationships impacting belief change. The third direction for future research is the completion of of speculative application. We are interested in providing a complete approach to the representation and verification of trusted third party protocols using our description language.

REFERENCES

A. Ulrich, R. Holz, P. H. and Carle, G. (2011). Investigating the openpgp web of trust. pages 489–507.

Alchourrón, C. E., Gärdenfors, P., and Makinson, D. (1985). On the logic of theory change: Partial meet functions for contraction and revision. *Journal of Symbolic Logic*, 50(2):510–530.

Baral, C. and Gelfond, M. (1997). Reasoning about effects of concurrent actions. *Journal of Logic Programming*, 31(1-3):85–117.

Baral, C., Gelfond, M., and Provetti, A. (1997). Representing actions: Laws, observations and hypothesis. *Journal of Logic Programming*, 31(1-3):201–243.

Booth, R. and Hunter, A. (2018). Trust as a precursor to belief revision. *J. Artif. Intell. Res.*, 61:699–722.

Burrows, M., Abadi, M., and Needham, R. (1990). A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36.

D. Zissis, D. L. and Koutsabasis, P. (2011). Cryptographic dysfunctionality-a survey on user perceptions of digital certificates.'. *Global Security, Safety and Sustainability and E-Democracy*.

Delgrande, J. (2004). Preliminary considerations on the modelling of belief change operators by metric spaces. In *Proceedings of the 10th International Workshop on Non-Monotonic Reasoning (NMR 2004)*, pages 118–125.

Gelfond, M. and Lifschitz, V. (1998). Action languages. *Linköping Electronic Articles in Computer and Information Science*, 3(16):1–16.

Hunter, A. (2021a). Building trust for belief revision. In *Proceedings of the Pacific Rim Conference on Artificial Intelligence (PRICAI)*, pages 543–555.

Hunter, A. (2021b). On the use of causal rules to specify how trust impacts change in knowledge and belief. In *Proceedings of the 34th Canadian Conference on Artificial Intelligence, Canadian AI 2021*. Canadian Artificial Intelligence Association.

Katsuno, H. and Mendelzon, A. (1992). Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52(2):263–294.

Liu, F. and Lorini, E. (2017). Reasoning about belief, evidence and trust in a multi-agent setting. In *PRIMA 2017: Principles and Practice of Multi-Agent Systems - 20th International Conference*, volume 10621, pages 71–89.

Peppas, P. and Williams, M.-A. (2018). Parametrised difference revision. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 277–286.