A Logical Approach to Promoting Trust over Knowledge to Trust over Action

Aaron Hunter
School of Computing and Academic Studies
British Columbia Institute of Technology
Burnaby, BC, Canada
aaron_hunter@bcit.ca

***Abstract*—We discuss two related forms of trust. One form of trust is related to the perceived knowledge of other agents; we accept the information that another agent provides if we believe they have sufficient expertise in a particular domain. The second form is related to action; we trust another agent to act on our behalf if we believe they will choose acceptable actions. In this paper, we explore the relationship between these two forms of trust. In particular, we use an existing model of trust to demonstrate how trust over knowledge can determine when trust over actions is appropriate. We take a formal approach to this problem, using logic-based tools for representing and reasoning about actions and beliefs to characterize trust over action. While our primary aim is to develop a formal methodology that permits trust over actions to be defined in terms of trust over knowledge, we also consider applications that are both practical and speculative. On the practical side, we consider how our methods can be used to reason about trusted third parties in communication protocols. On the speculative side, we suggest that models of trust have a role to play in the development of ethical decision-making agents.**

I. Introduction

We are concerned with the way that trust over *knowledge* can lead to trust over *actions*. Trust over knowledge refers to the phenomena in which agents only trust others over a restricted domain of expertise. For example, one is likely to trust a medical doctor with respect to human health, but not necessarily with respect to politics. Trust over actions, on the other hand, refers to the willingness of one agent to allow another to act on their behalf. For example, one might trust a personal assistant to make purchasing decisions with a credit card.

In this paper, we take trust over knowledge to be primitive. In other words, we assume as a starting point that each agent has a suitable model that dictates how information from others will be incorporated. The model we use comes out of the belief revision literature in the Artificial Intelligence (AI) community. Given this model, we then set out to specify precisely when another agent can be trusted to choose from the available actions.

We take a formal methods approach to reasoning about trust. The rationale behind this choice is the fact that trusting another to act on your behalf requires a strong guarantee of safety. The strongest such guarantee is a formal proof that the agent being trusted will always choose actions that are in-line with your preferences. As such, we argue that formalizing this kind of reasoning in a logical setting is appropriate.

This paper makes several contributions to the literature on reasoning about trust. First, we provide a concrete approach for defining trust over actions in terms of trust over knowledge. In the process, we demonstrate an important area of application for formal methods developed in the formal AI community. In particular, we demonstrate the utility of so-called *trust partitions* for reasoning about trust in a wider range of applications than originally anticipated. This kind of interdisciplinary connection is important for the formal AI community, as it provides important new applications for reasoning. This is primarily an exploratory paper, in which we consider the utility and feasibility of using trust partitions as the basis for reasoning about trust over actions.

II. Preliminaries

A. Knowledge-Based Trust

Broadly speaking, our work falls under the umbrella of *knowledge-based trust*. Knowledge-based trust refers to the manner in which one agent tries to predict the behaviour of another based on past actions. This is not strictly what we are doing in this paper, as we do not assume any knowledge of past actions. However, we will assume that certain agents are known to have "expertise" in some area; in practice, this known expertise would likely be gleaned from past actions.

One important application of knowledge-based trust is the evaluation of information sources, such as web pages. It has been shown that the reliability of a claim on a web page can accurately be predicted by looking at the accuracy of the other claims on the page, with respect to known online knowledge [5]. This particular work is quantitative, as it provides a measure of likelihood for the accuracy of information. By contrast, our approach will be based on logics of belief.

We are actually interested in taking knowledge-based trust one step further. Work on evaluating information sources is focused on predicting the accuracy of future reports. We are determining how accurately we can predict which actions will be executed in the future.

B. Formal Models of Belief

The formal model of trust discussed in this paper was developed in the literature on belief revision. As such, we

briefly introduce the tradition of work in this area. The most influential approach to belief revision is the so-called AGM approach [1]. In this approach, we assume an underlying propositional vocabulary $\mathbf{F}$. A *state* is a propositional interpretation of $\mathbf{F}$; in other words, a state assigns true/false values to all of the variables in $\mathbf{F}$. By convention, we identify a state s with the set of variables that are assigned the value *true*. We let $2^{\mathbf{F}}$ denote the set of all states. A *belief state* is an element of $2^{\mathbf{F}}$. Intutitively, an agent with the belief state K believes that the actual state of the world is one of the elements of K.

An AGM *belief revision operator* is a function that maps a belief state and a formula to a new belief state, while satisfying the so-called AGM postulates for rational revision. We do not list the postulates here. We remark, however, that every AGM revision operator can be defined in terms of a *plausibility ordering* over states. Hence, belief revision really just involves determining the most plausible states consistent with some new piece of information.

Belief revision is intended to capture the belief change that occurs when an agent observes some information, or receives a report about a static world. But there are other forms of belief change as well, such as belief change caused by an action. Assume there is an underlying set of *action symbols*. The effects of actions can be given by a *transition system*, which is just a directed graph where the nodes are labelled with states and the edges are labelled with actions [6]. Informally, an arrow labelled with A from s to s' means that performing action A in state s will lead to state s'. *Belief progression* refers to the belief change that occurs when an agent predicts the effects of an action by projecting all states in a belief state to the outcome of that action.

In this paper, we will not be directly concerned with belief change, but we will be using many of the notions defined in this section.

C. Trust Partitions

An alternative approach to knowledge-based trust has been defined in the belief change literature [12]. In this approach, trust in an agent is defined in terms of a so-called *trust partition*. Briefly, a trust partition Π_A^B for agent A with respect to B is just a partition on the set of states. If $\Pi_A^B(s, s')$, then A does not trust B to be able to distinguish between states s and s'. This allows us to capture trusted expertise, such as the doctor that is trusted on health but not politics.

Using a trust partition, we can define belief revision that is sensitive to trust. When A is provided a set of states S for revision from B, then A first needs to add all states that B cannot distinguish from the states in S. Hence, we expand the set of states being reported, and then revise. This allows us to trust the doctor on certain distinctions, but not on others.

In this paper, we will use trust partitions for a different purpose. We will use them to help predict what an agent is likely to do, based on their expected preferences. If we already have a trust partition to capture trust over information, we show that the same partition can help inform trust over actions.

D. Motivating Example

We introduce a motivating example to which we will refer throughout the paper as we introduce our formal machinery. The example here is a problem in commonsense reasoning, modified from [10].

Suppose that Alice and Betty are individuals that are given a variety of snacks to eat, and a variety of utensils with which to each them. Some of the snacks are solid candy that can easily be eaten with fingers. Some of the snacks are closer to liquids, such as yogurt or pudding, and these will be messy unless they are eaten with a spoon. Assume that Alice is in a parental role, where she will have to clean up any mess after the snacks are eaten. The question we are interested in is the following: when should Alice *trust* Betty to decide how the food is eaten?

This is a simple example, but it captures an important feature about trust over actions. In the case where all of the snacks are solid, then Alice will generally trust Betty to choose how to eat. In the case where some snacks are liquid, reasoning is required. If there are no spoons available, for example, then Alice might not want Betty to have a free choice of snacks. That is, of course, unless Alice believes Betty will choose the right utensil.

While this example focuses on commonsense reasoning, parallel problems exist in Information Security when we provide users authorization to perform certain tasks.

III. Trust over Actions

A. Action Scenarios

Throughout the remainder of the paper, we assume a fixed underlying propositional vocabulary $\mathbf{F}$. The following notion of an *action scenario* will be important.

Definition 1: An *action scenario* is a tuple $\mathcal{AS} = \langle \mathbf{A}, Act, \mathbf{K}, \Phi, \mathbf{T} \rangle$, where:

- $\mathbf{A}$ is a set of *agents*.
- Act is a set of *action symbols*, with effects given by an underlying transition system T.
- $\mathbf{K}$ is a function mapping each agent A to a belief state K_A.
- Φ is a function mapping each agent A to an AGM revision operator $*_A$
- $\mathbf{T}$ is a trust function that associates a trust partition Π_A^B with each pair of agents A, B.

Following [12], we write $*_A^B$ for the trust-sensitive revision function for A receiving information from B, based on the trust partition Π_A^B. As a short hand, we omit subscripts on K_A when it is clear from the context. In particular, we write $K *_A^B \phi$ rather than $K_A *_A^B \phi$.

Example 1: We can formalize our main example as an action scenario. The underlying propositional signature includes variables C, Y, M. Informally: C is true if there is candy, Y is true if there is yogurt, and M is true if there is a mess of some sort. We use A and B to refer to Alice and Betty, respectively. We have four actions cf, cs, yf, ys that informally represent: eating candy with fingers, eating candy with a spoon, eating

yogurt with fingers, and eating yogurt with a spoon. So we have the following:

- $\mathbf{A} = \{A, B\}$.
- $Act = \{cf, cs, yf, ys\}$.

The transition system giving the action effects encodes two different changes. First, eating either snack makes the corresponding variable become false. Eating yogurt with fingers will also make M true. The transition system may also encode the preconditions for the actions: for example, there must be yogurt present to eat yogurt.

For now, we assume that all Alice and Betty know is that there is no mess. Hence:

- $K_A = K_B = \{\emptyset, \{C\}, \{Y\}, \{C, Y\}\}$.

For simplicity, we specify:

- $*_A = *_B =$ the Dalal revision operator based on Hamming distance [3].

This revision operator is adopted for convenience, as it is easy to define over any propositional vocabulary. The details of the definition are not important; the intuition behind the Dalal operator is that agents will consider a state to be plausible to the extent that it agrees on the truth values that they currently believe.

The only thing that remains is to specify the trust function. We specify different partitions for A and B. Betty has complete trust in Alice, so Π_B^A is the so-called *unit partition* where every state is in a separate cell. On the other hand, Alice does not trust Betty to be able to tell yogurt from candy. So Π_A^B relates any state where C is true to the identical state where Y is true.

B. Preferred States

Informally, we are interested in determining when an agent should trust another to act on their behalf. In other words, given some particular context along with some set of actions, we want to know when an agent A can reasonably allow B to choose the action to be executed. In the abstract, it is not really possible to answer this question unless we have some information about which states an agent prefers. A full discussion of preference is beyond the scope of this paper, and we refer the reader to [8], [13] for some representative work in the area. For our purposes, the following definition is sufficient.

Definition 2: A *preference relation* for an agent A is a total pre-order $<_A$ over all states.

This leads to a natural notion of preference over actions in each state, in that A should prefer actions that lead to preferred states. Specifically, if s is a fixed state, we define $<_A^s$ as follows:

$$a_1 <_A^s a_2 \iff a_1(s) <_A a_2(s).$$

We can extend this to the case where we have a belief state, rather than a fixed state.

Definition 3: Let $\mathcal{AS}$ be an action scenario, and let $<_A$ be a preference relation for some $A \in \mathbf{A}$. The preference structure for A is the set

$$\{<_A^s | \ s \in K_A\}.$$

We write $\Gamma(A)$ as a shorthand for the preference structure for A.

Note that the preference structure for agents may be different, even if the preference relations are the same.

Example 2: In the snack example, assume that Alice prefers states where there is no mess over states where there is a mess. It is easy to verify that, for the initial states $\emptyset$ and $\{C\}$, the induced ordering over actions is empty: no action is preferred over another. For the initial states $\{Y\}$ and $\{C, Y\}$ on the other hand, the ordering over actions can be described as follows:

$$yf < cf = cs = ys.$$

So the preference structure for A in this case contains two orderings.

C. Trust Structures

The preference structure for an agent A indicates which actions they would prefer to execute. This is determined by the preference ordering over states, as well as the initial belief state of A. However, this does not indicate anything about what actions B is likely to execute. The beliefs A holds about B's expertise are captured by a trust partition. The interplay between these two structures can be complicated, but it dictates the situations where A should trust B to choose an action.

First, we introduce the analog of a preference structure for reasoning about another agent's action preferences. If K is a belief state and Π is a partition over states, define

$$K(\Pi) = \{s \mid \Pi(s, s') \text{ for some } s' \in K\}.$$

Hence, $K(\Pi)$ is the set of states that are related to a state in K. In the case of trust partitions, $K(\Pi)$ is the set of states that we do not trust another agent to be able to distinguish.

Definition 4: Let $\mathcal{AS}$ be an action scenario, let $A, B \in \mathbf{A}$ with $A \neq B$ and corresponding preference orderings $<_A$ and $<_B$. The trust structure for A with respect to B is the set

$$\{<_B^s | \ s \in K_A(\Pi_A^B)\}.$$

We write $\Gamma_A(B)$ as a shorthand for the preference structure for A.

Note that $\Gamma_A(B)$ is again a set of preference orderings over actions. However, these orderings are obtained from the initial preference ordering that B has over states. Since we are working from the perspective of A, we assume that all of the states in K_A are possible. But note that A does not trust B to distinguish between the states within a cell of Π_A^B. So if a state s is initially believed to be possible, then B can not be trusted to distinguish it from all states s' such that $\Pi_A^B(s, s')$.

Example 3: Suppose that Betty actually prefers to make a mess; so states where M is true are preferred over states where M is false. We consider two different initial belief states for Alice.

First, suppose that Alice's initial belief state consists of the single state $\{Y\}$, indicating that there is yogurt present. Since Betty prefers to make a mess, she will prefer to execute the action yf, eating the yogurt with her fingers. There is actually a second state to consider as well. As indicated previously,

Betty can not distinguish yogurt from candy. So the trust structure here also includes an ordering which prefers cf, eating candy with her fingers.

Now, suppose that Alice's initial belief state consists of the single state $\{C\}$, indicating that there is candy present. The trust structure here is exactly the same as it was in the previous case, due to the fact that Alice does not trust Betty to distinguish yogurt from candy.

We conclude this section with a remark about the running example. We have seen that, when $\{Y\}$ is the initial belief state, yf is Alice's least preferred action whereas it is Betty's most preferred action. It seems, therefore, that Alice should not trust Betty to act in this situation. On the other hand, if $\{C\}$ is the initial belief state, then Alice has no preference over actions. In this case, she can trust Betty to choose an action.

IV. Structure Evaluation

The discussion of our running example illustrates the basic idea that we can formalize action scenarios and then use induced preferences on states to determine when an agent can be trusted to act. However, in most situations, the realtionship between preference structures is hard to calculate. One case is straightforward:

$$\Gamma(A) = \Gamma_A(B).$$

In this case, the preference structure for A coincides exactly with the trust structure with respect to B. One would expect A and B to choose the same actions in this case, and therefore it is acceptable for A to trust B to choose how to act.

We are interested in identifying situations where the agent be trusted is guaranteed to make a choice that is acceptable.

Definition 5: Let M_1 and M_2 be sets of orderings over actions. We say that M_2 *supports* M_1 in a set of states K just in case, for each $s \in K$, the most preferred executable action in each ordering in M_2 is also maximal among executable actions in each ordering in M_1. We write $M_2 \Rightarrow^K M_1$.

Note that this is a very strong correspondence, as it essentially requires M_2 to select the exact same actions as M_1 in every state under consideration.

A. Basic Results

We list some basic results.

Proposition 1: For any action scenario involving agents A and B, if $\Gamma(A) = \Gamma_A(B)$ then $\Gamma_A(B) \Rightarrow \Gamma(A)$.

This is not a surprising result, as it should be required for any reasonable notion of trust over action. The following result is similarly straightforward.

Proposition 2: For any action scenario where Π_A^B is the unit partition, $\Gamma_A(B) \Rightarrow \Gamma(A)$ if and only if the preference orderings $<_A$ and $<_B$ are identical when restricted to the set $\{a(s) \mid a \in Act, s \in K_A\}$.

In other words, if A trusts B to distinguish between all possible states, then A should only trust B to act if they have the same preferences over actions outcomes on the initial belief state of A.

Proposition 3: For any action scenario where Π_A^B is the trivial partition, $\Gamma_A(B) \Rightarrow \Gamma(A)$ if and only if the set of maximal elements of $<_B$ is the set of $<_A$ maximal elements of $\{a(s) \mid a \in Act, s \in K_A\}$.

So, if B is not trusted to distinguish between any states, then they can only be trusted if they happen to globally prefer the outcomes that A prefers over the possible outcomes from their initial belief state.

Of course, these results are all extreme. The interesting cases fall in between, when A trusts B to distinguish certain states but not others. One idea would be to count isomorphisms between the two sets of orderings, or even partial isomorphisms. We could then identify when the two structures are "close enough" to trust the other agent to act. We leave a complete treatment of these cases for future work.

V. Application: Message Passing Protocols

A. Exchanging Messages

Many cryptographic protocols are simply concerned with passing messages between participants. In this section, we formalize message passing in a transition system framework, and we discuss how our model of trust can be used to understand the role of trusted third parties.

In order to reason about message passing, we introduce a set $\mathbf{M}$ of *messages*. Intuitively, each agent in the underlying set of agents $\mathbf{A}$ has posession of some set of messages. We can use this idea to define a propositional vocabulary. Let

$$\mathbf{M}A = \{M_A \mid M \in \mathbf{B}, A \in \mathbf{A}\}.$$

Intuitively, M_A is true just in case the agent A posesses the message M. Hence, an interpretation over $\mathbf{F}$ completely specifies which agents hold which messages.

With this definition in place, we can follow the approach of [11] to define the effects of actions. The primitive notion here is that of a *message exchange*, which is a triple $\langle A, B, M\rangle$ consisting of agents A, B and a message M. Intuitively, a message exchange represents the exchange of a message M sent from A to B. The meaning of a message exchange is defined by a transition system over $\mathbf{M}A$.

Definition 6: Let $s \in \mathbf{M}A$ and let $\langle A, B, M\rangle$ be a message exchange. Define $s + \langle A, B, M\rangle = s'$ where s' is such that:

1) If $s \models M_A$, then $s' \models M_B$.
2) For $M' \neq M$ and $C \neq B$, $s' \models M'_C \iff s \models M'_C$.

Hence, $+$ is a progression operator that maps a state and a message exchange to the natural resulting state.

B. Protocols

We would like to specify a class of action scenarios for reasoning about message passing protocols. The sort of protocol that we have in mind is of the following form:

General Protocol
1. $A \to B : M_1$
2. $B \to T : M_2$
...
n. $T \to A : M_n$

We are using the traditional notation employed in the protocol verification literature, such as that inspired by the pioneering work on BAN logic[2]. Hence $A \rightarrow B : M$ is interpreted to mean that A sends the message M to the agent B. The agent T stands for a trusted third-party, one that might be used to issue a certificate of identity. A simple example could take the following form:

Simple Identity
1. $A \rightarrow T : B$
2. $T \rightarrow A : K$
3. $A \rightarrow B : \{A\}_K$

In this protocol, A sends T a public identifier for B. Following receipt of this message, T sends A a key K that can be used to encrypt messages for B. In the next step, A sends B a message with their own public identifier encrypted with the key K. There are two prerequisites for this protocol to work. First, A must believe that T has the appropriate K. Second, B must trust T to act on their behalf.

C. Action Scenarios

We now formalize a suitable class of actions scenarios for reasoning about protocols involving trusted third parties. The set of agents $\mathbf{A}$ is $\{A, B, T\}$. As indicated previously, A and B are agents exchanging messages and T is a trusted third party.

The set of action symbols is:

$$Act = \{send(X, Y, M) \mid X, Y \in \mathbf{A}, M \in \mathbf{M}\}.$$

The meaning of the action symbols is given by a transition system. The nodes in the transition system are labelled with states, and the edges are labelled with action symbols. The node s has an edge labelled $send(X, Y, M)$ to the node s' if and only if $s' = s + \langle X, Y, M\rangle$.

The function $\mathbf{K}$ that maps each agent to a set of beliefs is protocol dependent. Recall that the set of propositional variables consists of atoms of the form M_A indicating which agents have which messages. While a full description of an action scenario requires us to specify the initial beliefs of each agent, it is often the case that there are only a handful of important constraints. In the **Simple Identity** protocol, for example, the main constraint is that $s \in K_A$ implies $s \models T_K$. In other words, A must believe that T actually posesses the encryption key for B.

A complete action scenario needs to specify a revision operator $*$ for each agent. As in the motivating example, we could use the Hamming distance between states to define an approach to revision. However, it has been argued in [9] that a more suitable revision operator is the so-called *topological revision operator* based on the number of actions it takes to get from one state to another. This revision operator provides a sensible model of revision if we assume that the only way information is tranferred is through the sending of messages.

The final component of our action scenario is the trust function $\mathbf{T}$ that associates a partition Π with each pair of agents. Again, this is dependent on the protocol. In the **Simple Identity** protocol, the main trust issue is that B must trust T to act. In the case discussed so far, there is only one agent that could possibly request the identity certificate for B; this makes it difficult to see the role played by trust. In practice, there are likely to be several different potential communicators. In this case, we give an example property that we might want to respect.

- If two agents have not been blacklisted, then T should treat them equally. *[Fairness]*

There are certainly more similar properties to respect, but this will be sufficient for discussion.

Assume that there is a set of predicates BL_X for each $X \in \mathbf{A}$ indicating if X has been blacklisted. The important constraint we need for **Simple Identity** is that the partition Π that B associates with T should not distinguish between agents. For example, states where T holds the message A should be indistinguishable from those where T holds the message C (provided that BL_A and BL_C are both false). It is easy to define a class of partitions with this property. If we assume that T has one of these state partitions, then T will satisfy *Fairness*.

In general, if B and T have no preferences over states other than to avoid sending messages to blacklisted agents, then the actions chosen by B and T will be identical. Hence, T can be trusted to act on behalf of B in this setting. If the preference orderings for B and T are different, then we need to compare trust structures as suggested in the previous section. In either case, the action scenario and preference ordering combined will allow us to perform the appropriate evaluation.

D. Protocol Correctness

We conclude this discussion of trusted third parties by considering the notion of protocol correctness. The important feature of a cryptographic protocol is that a complete run of a protocol must guarantee that certain properties are achieved. The correctness of a protocol is proved by demonstrating that any sequence of messages satisfying the protocol must satisfy the stated goals of the protocol. In order to make this more precise, we introduce some terminology. The terminology is inspired by [11], though it is not identical.

A protocol is a template for message passing, defined by a set of expressions as we have given for **Simple Identity**. More precisely, a protocol is a triple (P, G_p, G_e) where:

- P is a function with domain $\{1, \ldots n\}$ for some n such that $P(i) = \langle X, Y, M\rangle$ for some $X, Y \in \mathbf{A}$ and $M \in \mathbf{M}$.
- G_p is a set of states.
- G_e is a set of message exchanges.

The function P specifies the messages to be exchanged by each agent in a run of the protocol. The set G_p is the *information goal* of the protocol; it is specifies the set of goal states. Informally, G_p indicates which messages should be held by each agent after a successful protocol run. The set G_e is the *exchange goal* of the protocol; it specifies which messages need to have been exchanged after a successful protocol run. This is an important distinction. A protocol is only successful if the "right" agents have sent particular messages.

A *trace* is a finite sequence of message exchanges. We say that a trace contains a *run* of P if there is an increasing sequence of natural numbers $p_1, \ldots p_n$ such that $T'[p_i] = P(i)$ for all $i < n$.

The notion of *protocol correctness* is formally defined and explored in [11]. Roughly, a protocol P is *correct* just in case, for every state s and every trace T, if T contains a run of the protocol P, then:

- The state after P completes is in G_p.
- The set if message exchanges in T superset of G_e.

Hence, a protocol is correct if it leads to a goal state and ensures all required messages are exchanged.

We are interested in extending this notion of correctness to reason about trusted third parties. The basic idea is the following. If $\mathcal{P} = \langle P, G_p, G_e \rangle$ is a protocol involving an agent X, we define a new protocol $\mathcal{P}(X/Y)$ that is identical to P except that the agent X is replaced with the agent Y. We say that $\mathcal{P}$ is *safe* under the subsitution X/Y just in case the set of message exchanges sent by X in $\mathcal{P}$ is identical to the set of messages sent by Y in $\mathcal{P}(X/Y)$.

Clearly, the notion of safe substitution provides a sufficient condition for trust in a third party. If T can be trusted to send exactly the same messages that B would have sent in the same circumstance, then T can be trusted. In restricted cases, we suggest that it is actually possible to prove that a subsitution is safe given the full action scenario including a trust partition for T. We leave a specification of these cases for future work.

VI. Speculative Application: Ethical Behaviour

A. Motivation

In order to demonstrate the flexibility of our approach, we consider a very different setting involving trust. There has been a great deal of discussion in the popular media about the risks posed to humanity by intelligent machines. A fundamental question underlying this debate is the following: Can we trust an intelligent machine to make ethical decisions?

In order to facilitate the discussion, we focus on *utilitarian ethics*. This is the theory that the ethically appropriate action is the one that leads to the most "good" overall for all agents. This is called a *consequentialist* theory because it focuses on the outcomes of actions. We introduce a simple example.

Example 4: Consider three agents: Alice, Bob and Craig. Alice is an important individual that needs to get from one place to another periodically. Bob is Alice's peronal assistant, responsible for keeping track of a variety of practical issues. Craig is Alice's driver.

Suppose that Alice needs to get to an important meeting; missing the meeting, or even being late, will have dire consequences. Bob is aware that the brakes on the car are due to be replaced, but it is considered unlikely that they will fail. Craig is not aware of the condition of the car.

The question here is whether or not Bob is under any moral obligation to inform Alice or Craig about the fact that the brakes are due to be replaced. On a utilitarian analysis, the best choice depends on the cost of missing the meeting, the cost of damaging the car, and the risk/cost associated with endangering Alice and Craig. In this ethical theory, the notion of honesty or the treatment of individuals is only an indirect consideration.

B. Ethical Theories: Formal Structure

Every ethical theory involves a set of *contexts*, a set of *agents*, and a set of *actions*. In the best base, we will have a theory that gives a total ordering over all actions available in a given context; this allows us compare the ethical value of each possible choice an agent can make. However, in practice, a total ordering is unlikely. It is more common that an ethical theory provides a partial ordering, allowing us to say "action A is better than action B" in some restricted cases. Sometimes this partial ordering is stated as a partial labelling of "good" or "bad" options.

Example 5: In our motivating example, a utilitarian approach would dictate that the set of contexts includes: a world where the brakes work correctly, a world where the brakes fail causing Alice to miss the meeting due to tardiness, and a world where the brakes fail causing harm to Alice, Craig and possibly others. The set of actions includes: disclosing the issue and keeping the issue secret. In order to perform a utilitarian analysis here, we would need to add a value to each outcome, as well as a probability of each form of brake failure. It would then be easy to automate the process of decision making.

C. Trusting Agents

Fundamentally, the issue of ethical behaviour by artificial agents is concerned with interchangeability. We would like to know if an artificial agent would choose the ethical action, in a manner similar to a human decision maker. We say that an ethical problem is *replaceable* with respect to an agent P if we can toggle P between human and computational agent, and the resulting preference order over actions is unchanged.

Example 6: Consider our motivating example again. Suppose that Bob keeps the secret about the brakes, in the situation where Alice and Craig are human. In a utilitarian analysis, the value associated with a human life is usually set higher than any amount of monetary profit. Hence, if there is a non-zero chance of a fatal crash due to the failed brakes, it follows that the ethically correct decision is to disclose the issue.

Now consider the case where Alice and Craig are machines. Presumably these machines could be backed up and replicated, losing little data if there was a crash. This might be very expensive. However, there would be a monetary gain associated with Alice reaching the meeting that would outweigh this risk. Hence, in principle, the utilitarian analysis could reach different conclusions based solely on exchanging humans with machines. In our terminology, we say that this scenario is not replaceable with respect to Alice or Craig, from a utilitarian perspective.

This kind of situation can be formalized in terms of action scenarios with trust partitions. While ethical theories may

discuss actions and contexts in a general setting, we can formalize these notions precisely with respect to states over a fixed vocabulary. We can then define a value function over all possible states as the basis for our utilitarian ethics. Hence, we can define the set of states that a machine considers possible as well as the preference ordering over all states. With this basic information in place, our formal model of trust should allow us to determine when the machine can be trusted to make ethical decisions.

It is important to note that utilitarian ethics introduces many problems that are independent of trust. For example, in general, it is hard to know where the utility functions come from. On one hand, we could attempt to teach machines to behave morally [15]. But it has also been suggested that computational agents could acquire such functions through machine learning [14]. The approach is promising, but difficult to implement. Following the discussion in this paper, we can not always replace a human with an agent and expect the ethical behaviours to remain constant. This has also been pointed out by Grau in [7], using examples from science fiction. Grau argues that utilitarian ethics can only be applied appropriately when the agents involved have a sense of self. Without such a sense, utlitarian calculations can lead to unintended consequences, as the cold calcuations of utility do not always agree with our intuitions. On the other hand, if a machine is imbued with a sense of self, then self-preservation becomes an issue. The former problem is an artefact of the ethical theory whereas the latter problem is due to the fact that computational agents are only human-replaceable if we value their existence equally.

These concerns with utilitarian ethics are not our concern at present. The advantage of a utilitarian approach is that it allows us to use preferences over states, and then consider when a machine can be trusted to act. This is exactly what our formal approach allows us to do. If we started with a different ethical theory that was not based on an ordering over states, then we would need to reformulate our approach signicantly to address ethical issues. As such, we accept the limitations of the utilitarian approach for the moment.

VII. Discussion

A. Key Assumptions

It is worth noting that we have made several significant assumptions in this paper that may not always be reasonable. First, the "subject" agent A does not have any knowledge of the initial belief state of the "object" agent B. In a sense, A assumes that the initial belief state K_A is correct in that the actual state of the world must be in this set. An alternative approach would be to actually assume that B has a completely different initial belief state and that A has some partial knowledge of this belief state. This situation could be more effectively modelled with Dynamic Epistemic Logic (DEL), where agents have nested beliefs about the beliefs of others [4]. However, we do not take this route, as we are interested in applying our approach in concrete security settings where the computational complexity of reasoning in DEL would be prohibitive.

Another assumption that we have implicitly made is that A is aware of the preference ordering $<_B$. In other words, A is aware of which states B prefers. There are certainly cases where this would not be true, and it might be more appropriate to given A only partial information about $<_B$. This would not require a significant change to our approach.

B. Future Work

There are many directions for future work. From a formal perspective, there is still a great deal of work to be done on the notion of supporting structures. It is also important to relax the rigid definition of trust to allow an agent to trust another in cases where the maximally preferred states are not necessarily aligned. This could take the form of a notion of dominance, where certain structures always choose a "better" action, even if the maximal elements are not identical. It could also involve counting isomorphisms, as previously mentioned. Finally, it would be worthwhile to explore quantitative or probabilistic notions of support.

From a practical perspective, we would like to further develop the use of our framework for the analysis of communication protocols. There are many protocols where one agent trusts another to choose a key, provide a random number, or answer some secret question. In each case, we need to be assured that the agent being allowed to act is suitably trusted.

As a final note on future work, we remark that we have not actually used the revision operators that occur as part of the action scenario in this paper. However, we suggest that communication in general requires revision operators to understand incoming messages and information. In forthcoming work, we will be dealing with trusted action and belief revision simultaneously; we have left the revision operators in the action scenario to maintain consistency with this work.

VIII. Conclusion

In this paper, we have introduced a formal methodology for promoting trust over information to trust over action. Our approach is based on the notion of a *trust partition*, which indicates the situations that another agent is trusted to distinguish. We demonstrate that this simple notion of distinguishability can be combined with a preference ordering over states to predict the actions that an agent will execute. If the predicted actions are acceptable, then we can reasonably allow the agent in question to act on our behalf.

There are many natural applications for this kind of reasoning, and we have explicitly discussed two of them in this paper. The first application considered is the issue of trusted third parties in communication protocols. There are many cases where it is useful to define a protocol that involves a third party that shares keys or other identifiers for communication. Unfortunately, this kind of protocol is vulnerable to many forms of attack. For our purposes, we are not concerned with attacks by intruders, but we are simply concerned with the rationality of trusting the third party in the first place. We

demonstrate that we can formalize protocols in our framework, and we take the first steps towards a full model of verification for this kind of protocol. The fundamental notion to be addressed is whether or not the trusted third party can actually be expected to take actions we would take ourselves.

The second application discussed here is very speculative, but of great emerging importance. As intelligent machines take on an increasing number of roles in society, there is growing concern about the ethical behaviour of these machines. Actually proving that a machine will behave ethically seems quite difficult. However, we have shown that it is actually possible in some cases. If we take a utilitarian view on ethics, then we can reduce ethical considerations to a preference ordering over states. In this case, we can trust a machine to behave ethically when we can trust it to make decisions that are consistent with our own preferences over states. That is precisely what our formal framework allows us to do, which suggests that our model of trust may actually have a role to play in discussing ethical decision making by machines.

We remark that this is a preliminary paper, aiming to set up a general framework for promoting trust over knowledge to trust over actions. The framework is built on logical methods developed in the belief change community, where the emphasis is on formal proofs of correctness and rationality. The applications considered here are described somewhat superficially, and they will be developed more completely in future work.

REFERENCES

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

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

[3] Dalal, M. Investigations into a theory of knowledge base revision *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 475-479, 1988.

[4] van Ditmarsch, H., van der Hoek, W., and Kooi, B. 2007. Dynamic Epistemic Logic. Synthese Library 337, Springer.

[5] Dong, X., Gabrilovich, E., Murphy, K., Dang, V., Horn, W., Lugaresi, C., Sun, S., and Zhang, W. Knowledge-Based Trust: Estimating the Trustworthiness of Web Sources. *IEEE Data Eng. Bull.*, 39(2): 106-117, 2016.

[6] Gelfond, M. and Lifschitz, V. Action Languages. *Linköping Electronic Articles in Computer and Information Science*, 16(3), 1-16, 1998.

[7] Grau, C. 2011. There is no "I" in "Robot". Machine Ethics, 451-473. Cambridge University Press.

[8] Horty, J. 2001. Agency and Deontic Logic. Oxford University Press.

[9] Hunter, A., and Delgrande, J.P. 2007. Belief Change and Cryptographic Protocol Verification. *Proceedings of the National Conference on Artificial Intelligence (AAAI 2007)*.

[10] Hunter, A. Actions, Preferences, and Logic Programs. *Proceedings of the Canadian Conference on AI*, 97-108, 2012.

[11] Hunter, A. Revisiting the Epistemics of Protocol Correctness. *Proceedings of the Canadian Conference on AI*, 256-262, 2013.

[12] Hunter, A., and Booth, R. Trust-Sensitive Belief Revision. *Proceedings of IJCAI*, 3062-3068, 2015.

[13] Lang, J. and van der Torre, L. 2008. From belief change to preference change. *Proceedings of the European Conference on Artificial Intelligence (ECAI 08)*.

[14] Sotala, K. 2016. Defining Human Values for Value Learners. *Proceedings of the AAAI Workshop on AI Ethics*.

[15] Wallach, W. and Allen, C. 2008. Moral Machines: Teaching Robots Right from Wrong Oxford University Press.