



From Knowledge to Trust: A Logical Framework for Pre-trust Computations

Mirko Tagliaferri^(✉) and Alessandro Aldini^(✉)

Urbino University, Urbino, PU, Italy

Abstract. Computational trust is the digital counterpart of the human notion of trust as applied in social systems. Its main purpose is to improve the reliability of interactions in online communities and of knowledge transfer in information management systems. Trust models are typically composed of two parts: a trust computing part and a trust manipulation part. The former serves the purpose of gathering relevant information and then use it to compute initial trust values; the latter takes the initial trust values as granted and manipulates them for specific purposes, like, e.g., aggregation and propagation of trust, which are at the base of a notion of reputation. While trust manipulation is widely studied, very little attention is paid to the trust computing part. In this paper, we propose a formal language with which we can reason about knowledge, trust and their interaction. Specifically, in this setting it is possible to put into direct dependence possessed knowledge with values estimating trust, distrust, and uncertainty, which can then be used to feed any trust manipulation component of computational trust models.

Keywords: Computational trust · Trust logic · Subjective Logic

1 Introduction

Given the growing number of interactions in online communities and of information exchanges in information management systems, it's becoming increasingly important to have security mechanisms that can prevent fraudulent behaviors. However, implementing hard security mechanisms for every possible interaction, e.g. authentication methods or access controls, can be costly and a failure of such mechanisms might put the whole system in jeopardy. For this reason, it is necessary to include other soft security mechanisms in the design of those environments where transactions and exchanges take place [10, 19]. Trust is one form of soft security that can be implemented into a system: trust is a social control mechanism that brings an undoubtedly positive impact on cooperative operations, both by increasing the chances of performing an interaction and by decreasing the chances of having malevolent behaviors during those interactions [1, 18]. Therefore, trust has both a proactive and a control effect over interactions. Computational trust is the digital counterpart of trust as applied in ordinary social communities and computational trust models are soft security mechanisms that implement the notion of trust in digital environments to increase the quantity and quality of interactions¹. Computational trust models are typically

¹ See [2, 3, 5, 14–16, 20] for surveys on computational trust models.

composed of two parts: a trust computing part and a trust manipulation part. While trust manipulation is widely studied, very little attention is paid to the trust computing part. In this paper, we propose a formal language with which it is possible to reason about how knowledge² and trust interact. Specifically, in this setting it is possible to put into direct dependence possessed knowledge with values estimating trust, distrust, and uncertainty, which can then be used to feed the trust manipulation component of any computational trust models. The paper will proceed as follows: in Sect. 2, we discuss the distinction between trust computation and trust manipulation by providing some examples of both components as implemented in computational trust models; in Sect. 3, following on the discussion of Sect. 2, we will show how one well-known model for trust manipulation, i.e. Jøsang's Subjective Logic [11], struggles when dealing with the trust computing component; in Sect. 4, we provide the syntax and semantics of what we will call trust logic. In this logic, we will show how we can talk about the notion of trust and we will then show how the semantical structure in which the logic is interpreted helps in compute trust values that can be fed into Subjective Logic's trust manipulation component; finally, in Sect. 5, we conclude the paper with some general remarks.

2 Trust Computing and Trust Manipulation

A good formalization of the notion of trust must accomplish two goals. The first goal is that of explaining how trust is generated: while it is possible to take trust as a primitive and unexplained notion, it is often preferable to provide a reduction of the notion of trust to more basic concepts, aiding our comprehension of the phenomenon of trust in various contexts. The second goal is that of explaining the dynamics of trust, i.e. how trust evolves under different circumstances: this should help in determining how dynamics in an environment (e.g. a group of friends or a multinational company) influence trust. To each goal corresponds a different component of trust models. Specifically, it is possible to identify a trust computing and a trust manipulation component. The former serves the purpose of gathering relevant information, which is considered basic, and then use it to compute trust values; the latter takes trust values as granted and manipulates them for specific purposes using different operators.

Even though both components are important, authors often concentrate on one or the other: models that concentrate on the trust computing component rely on the fact that when an environment changes, it is possible to repeatedly compute new trust values, therefore no trust manipulation component is needed; models that concentrate on the trust manipulation component can rely on conceptions of trust that take the concept as a primitive or depend for their initial values on other models, therefore neglecting the trust computing component. To highlight the distinction between trust

² It is important to note that, even though we employ the word 'knowledge', we do not commit to any specific epistemological notion of knowledge. In fact, our term might refer to very different conceptions of knowledge and, in some cases, also to beliefs. Although important from a philosophical point of view, this fact does not influence the formal framework we are presenting in the paper.

computing and trust manipulation, we will now provide three examples of trust models and show how each component behaves in those models. To those three models we will then add a fourth one, which will be used as a target for the reflections of the rest of the paper. Note that the models presented were selected for explanatory purposes (i.e. to make clearer the distinction between the two components). No important or specific facts are derived from the models and, therefore, no results of this paper depend on the choices made.

2.1 Marsh's Trust Model

In [13], Stephen Marsh presented the first example of a thorough and detailed analysis of the notion of trust in a computational setting. His system was designed for possible implementations in distributed artificial intelligence and multi-agent systems: the main purpose of his thesis was that of using a formal variant of common-sense trust to increase the quality of the evaluation an autonomous agent should perform to decide whether to collaborate or not with other agents. In this model, it is possible to identify three different forms of trust:

- (1) Basic trust;
- (2) General trust;
- (3) Situational trust.

Basic trust represents the general attitude of an agent, when all his experiences in life are considered; general trust is the overall trust a trustor has in a trustee; situational trust is the specific trust a trustor has in a trustee when a specific collaborative task should take place. Basic trust and general trust are taken as primitives, while situational trust is reduced to more basic concepts.

In Marsh's model, the trust computing component is arrived at through a conceptual analysis of the notion of trust, which either classify trust as a primitive notion or helps in identifying the basic elements that form specific versions of trust (i.e. situational trust). We will now briefly explain how situational trust is analyzed, since this form of trust is at the core of Marsh's model. Specifically, situational trust is computed starting from three basic parameters: $U_x(\alpha)$, the amount of utility agent x gains if situation α occurs; $I_x(\alpha)$, the importance of situation α for agent x ; $\widehat{T}_x(y)$, the general trusting disposition of agent x towards agent y . To obtain the situational trust, those three components are multiplied together.

The trust computing component of Marsh's model is straightforward. Once some basic information is gathered and expressed with numerical values, those values are multiplied to obtain a specific trust value.

What is limited in Marsh's model is the trust manipulation component. This is because the model relies on repeated computations of the trust values, rather than a manipulation of already obtained values. We will show in later sections (Sect. 3) that this is a very severe limiting factor in computational trust models, since it makes it hard to provide evaluations for trust in different scenarios, e.g. when trust is obtained through referrals from one agent to another.

2.2 Yu and Singh's Trust Model

In [21], the authors present a formal framework in which to evaluate the reputation of agents based on witnesses. The model also provides tools to avoid deception in rating provision. Yu and Singh's model is based on Dempster-Shafer theory of evidence and favors the trust manipulation component over the trust computing one. The trust computing component is based on past interactions between agents, where an agent x trusts another agent y if the percentage of positive past experiences over the whole number of recent experiences is superior to a given threshold. The trust computing component of this model is extremely simple and is lacking a proper analysis of trust. Moreover, it relies on the fact that the amount of data on past interactions is big, otherwise the values computed might be misleading. This is troublesome, since in digital environments interactions between the same agents are scarce. However, the strength of the model is associated with its trust manipulation component. In this model, it is possible to aggregate together ratings from different agents. Given a trust net, which is a directed graph representing the referral chains produced by an agent x 's query about the trustworthiness of another agent y , x might obtain a value of the trustworthiness of y by combining the various ratings given to y by all close acquaintances of x in the trust net. The operation is based on Dempster rule of combination and the result is that of combining as a weighted sum all the trust values of other agents into one for the trustor.

2.3 BDI + RePage

In [17], the authors present a sophisticated model for trust and reputation evaluation and propagation. This model has both a simple trust computing and a good trust manipulation component. First, in BDI RePage, there is a clear distinction between a trust and a reputation evaluation, where the former (called image) is interpreted as the trustor's belief in the trustworthiness of the trustee, while the latter is interpreted as the trustor's meta-belief on the beliefs about the trustee of other agents. The trust computing part is based, like in [21], on past experiences, which can be categorized under five different labels (Very Bad, Bad, Neutral, Good, Very Good). Past experiences influence the weight of each label, providing a specific numerical value for each. Those numerical values are then used to generate an image (what we call trust evaluation) of the trustee for the given trustor. This trust evaluation, coupled with the desires, the beliefs and the intentions of the trustor, leads to the decision of collaborating or not. As in [21], the trust computing component suffers from the requirement of having many past interactions and the real strength of the model lies in its trust manipulation component. For such component, this model includes a full logical language that can help reasoning about trust and reputation. The logical framework is that of a hierarchical multi-sorted first-order language. In such language is easy to express formulas that describe, over and above simple properties of objects, the desires, beliefs, intentions and the trust evaluations of a given agent and then, using various logical connectives (e.g. conjunction and disjunction), it is possible to specify different conditions for the presence or absence of trust by the trustor.

2.4 Summing up

We saw three computational trust models and their respective trust computing and trust manipulation components. It has been shown that none of the model can deal perfectly with both components. We will now present a fourth model, which will be the starting point for the reflection made in this paper. This model, i.e. Audun Jøsang's Subjective Logic [11], is extremely well-suited to manipulate trust values using different algebraic operators, but suffers from a poor trust computing component. We will show how the logical language we present in this paper, can be used to implement a good trust computing component in Subjective Logic.

3 Subjective Logic

We described in Sect. 2 the two main components of computational trust models and provided some examples of the role those two components have in such models. In this section, we will introduce a further computational trust model, i.e. Audun Jøsang's Subjective Logic. We will explain why we choose this model as the starting point of our work and why we believe the model requires some improvements.

In Subjective Logic trust is represented as the opinion of agent x about a given proposition p . An opinion has three major components, and a fourth added component which completes the trust evaluation and helps in computing an expected value for trust. The three major components are, respectively, a belief component, a disbelief component and an uncertainty component³, while the fourth added component is labelled as base rate and indicates the prior probability associated with the truth of a proposition when no initial relevant information is available. The belief, disbelief and uncertainty components are additive to one, leading to the fact that Subjective Logic is effectively an extension of traditional probability logics. The additivity principle of the three major components allow also for a nice visualization of opinions through a triangle, which we will call the opinion triangle (Fig. 1). It is possible to observe in the figure that an opinion ω_x is identified through the three major components of belief, disbelief and uncertainty and, after the generic opinion is obtained, it is possible to compute the expected trust value $E(\omega_i)$ using the base rate (indicated in the figure with the letter a): the base rate determines the slope of the projection of the opinion on the base of the triangle and allows to compute an expected value when no uncertainty is taken into account in the valuation of the opinion.

Subjective Logic is a widely employed model to manipulate trust, but is rather ill-suited when it comes to compute initial trust values to be used as inputs to the model. The reason is that the only source of information that Subjective Logic allows to compute trust values is reputation scores based on past interactions. Once it is noticed that different agents might evaluate interaction differently and that reputation scores in

³ Uncertainty plays a crucial role in the formalization of trust. There are many ways this concept has been described formally (see e.g. [6] for a survey). Dealing with trust in a formal way can thus help also understand how uncertainty can be modelled, providing useful insight also for fields other than computer science, e.g. economics.

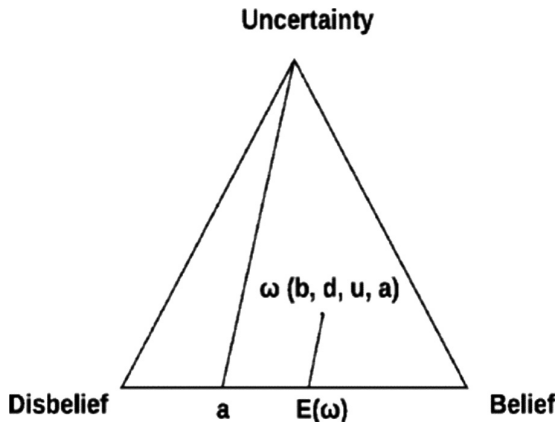


Fig. 1. Opinion triangle where all components used to compute trust are represented.

one context are not easily transferable to other context, the fact that Subjective Logic has no other means to compute initial trust values becomes a big drawback. This is also noted by Jøsang himself:

“The major difficulty with applying subjective logic is to find a way to consistently determine opinions to be used as input parameters. People may find the opinion model unfamiliar, and different individuals may produce conflicting opinions when faced with the same evidence.” [8].

The aim of this paper is specifically to implement a trust computing component that can produce initial trust values which can then be plugged into Subjective Logic.

4 A Language for Trust

We will now introduce the syntax and semantics of a formal language that will allow us to reason about knowledge and trust and, furthermore, to provide a trust computing mechanism that can produce values to be fed into Subjective Logic’s trust manipulation component. This, we believe, is an improvement to Subjective Logic. The leading idea of the framework comes from an insight given by Jøsang:

“...[T]rust ultimately is a personal and subjective phenomenon that is based on various factors or evidence, and that some of those carry more weight than others. Personal experience typically carries more weight than second hand trust referrals or reputation...” [9].

The idea is therefore that of using the expressive power of a formal language to describe the information possessed by an agent and then transform this knowledge into an opinion value about a given proposition with all the three major components of Subjective Logic made explicit.

4.1 Syntax

In our language \mathcal{L} , we start with two sets. The two initial sets are a finite set Ag of agents and a finite set At of atomic propositional constant. Given $p \in At$, $i \in Ag$ and b a rational number in the interval $[0, 1]$ (with 0 and 1 included), the grammar for our language is given by the following BNF:

$$\varphi := p | \neg\varphi | \varphi \wedge \varphi | K_i(\varphi) | \omega_i(\varphi) \geq b$$

All other connectives and operators are defined in the standard way:

- (i) $\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$;
- (ii) $\varphi \rightarrow \psi := \neg\varphi \vee \psi$;
- (iii) $\varphi \leftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$;
- (iv) $F_i(\varphi) := \neg K_i(\neg\varphi)$;
- (v) $\omega_i(\varphi) \leq b := \neg\omega_i(\varphi) \geq -b$;
- (vi) $\omega_i(\varphi) < b := \neg(\omega_i(\varphi) \geq b)$;
- (vii) $\omega_i(\varphi) = b := (\omega_i(\varphi) \geq b) \wedge (\omega_i(\varphi) \leq b)$.

$K_i(\varphi)$ should be intuitively read as “agent i knows that φ ”; we will call such formulas *knowledge formulas*. $\omega_i(\varphi) \geq b$ should be intuitively read as “agent i trusts formula φ to degree at least b ”; we will call such formulas *trust formulas*. The degree to which an agent can trust goes from 0, complete distrust, to 1, complete trust.

4.2 Semantics

The semantics we will provide in this paper is in truth theoretical form and depends on a structure that is a combination of traditional relational structures for modal logics with added components to interpret trust formulas [4, 7, 12]. We will interpret the above presented language in the following structure $M = (S, Cntx, \pi, R_{i1}, \dots, R_{in}, T)$, where S is a finite set of possible states of the system; $Cntx$ is a finite set of contexts, i.e. scenarios in which to evaluate trust; π is a valuation function over the set At , assigning to each atomic proposition in At a set of possible worlds, i.e. $\pi : At \rightarrow \wp(S)$; R_i , one for each agent $i \in Ag$, is an accessibility relation defined over S , i.e. $R_i \subseteq S \times S$; finally, $T = (X_{(i,c)}, \mu_{(i,c,\varphi)})$ is a trust relevance space which determines, for each formula of the language, which propositional constants are relevant and how relevant they are. T has two distinct components: a qualitative relevance component $X_{(i,c)}$ and a quantitative relevance component $\mu_{(i,c,\varphi)}$.

The qualitative relevance component $X_{(i,c)}$, one for each couple of elements $i \in Ag$ and $c \in Cntx$, takes formulas of the language as arguments and returns as values subsets of At , i.e. $X_{(i,c)} : \mathcal{L} \rightarrow \wp(At)$; the quantitative relevance component $\mu_{(i,c,\varphi)}$, which is a family of functions, one for each formula of the language and defined over the same couple of $i \in Ag$ and $c \in Cntx$ of $X_{(i,c)}$, assigns to each relevant atomic proposition p member of the subset of At selected by $X_{(i,c)}$, a rational number in the

range $(0, 1]$, i.e. $\mu_{(i,c,\varphi)} : X_{(i,c)}(\varphi) \rightarrow (0, 1]$ ⁴. The weights assigned must be additive to 1. Since the set of agents and the set of contexts are finite, there is a finite number of trust relevance spaces.

A possible state $s \in S$ represent a way in which the system can be specified; two states differ from one another by what propositions hold in such states. For example, in one state it might hold that it is sunny in San Francisco, while in another state it might hold that it is raining in San Francisco and therefore it is not sunny. If no non-standard conditions⁵ are placed on the description of a system (i.e. proscriptions on propositions that can or can't hold together), there will be exactly $2^{|At|}$ states in S , where $|At|$ is the cardinality of the set of propositional constants.

The valuation function π assigns to each proposition $p \in At$ a set of states; a state is included in the set if, and only if, the proposition holds in the given state.

The accessibility relations R_i connect possible states according to the epistemic status of the agent to whom the accessibility relation is associated. Two states are therefore connected if they are epistemically indistinguishable for the agent (i.e. according to what the agent knows, he can't determine which of two states is the actual one). For example, if an agent only knows the weather of Rome, he can't determine which is the actual state between the state in which it is sunny in San Francisco and the one in which it is raining in San Francisco; therefore, the two states would be connected by an accessibility relation; for the current paper, we are assuming that the accessibility relations are equivalence relations, i.e. reflexive, symmetric and transitive relations.

Finally, the trust relevance space T provides both a qualitative and a quantitative evaluation of the information that will help in determining the exact trust value an agent places in a formula. The qualitative relevance function $X_{(i,c)}$, specifies, for an evaluating agent (the trustor) and an evaluation scenario (the context), which information (under the form of propositional constants) is relevant to compute an actual trust value. The relevance weight assigning functions $\mu_{(i,c,\varphi)}$ provide a quantitative assessment of relevance of a given proposition to trust, assigning specific weights to all propositional constants that are selected by applying $X_{(i,c)}$ to φ . The additivity condition on $\mu_{(i,c,\varphi)}$ is in place to guarantee that in the best-case scenario (the one in which all relevant propositional constants do hold in the state and the agent is aware of all of them) there is full trust on the part of the trustor and moreover, it is never possible, in the system, to exceed full trust.

Once the basic components of the model are given, it is possible to extend the functions π and $\mu_{(i,c,\varphi)}$ to consider other formulas over and above the ones on which those functions are defined. This is fundamental to provide proper satisfiability conditions for the language. We will proceed to give the extensions explicitly: we will label the extension of π with π^{ext} and the extension of $\mu_{(i,c,\varphi)}$ with $\mu_{(i,c,\varphi)}^{\text{ext}}$. Intuitively, the

⁴ It is important to stress that $\mu_{(i,c,\varphi)}$ is not a single function, but a family of functions. Therefore, each member of $\mu_{(i,c,\varphi)}$ assigns a value to the relevant atomic propositions according to the specific formula on which the function depends.

⁵ With standard conditions we refer to conditions such as the principle of non-contradiction, for which a proposition and its negation can't both hold in the same state. An example of a non-standard condition is the fact that one proposition always implies a second one, therefore proscribing states in which the former holds while the latter doesn't.

function π^{ext} assigns to each formula of the language the set of states in which the formula holds, i.e. $\pi^{\text{ext}} : \mathcal{L} \rightarrow \wp(S)$. Put another way, π^{ext} returns, for each formula of the language, the states that are compatible with the truth of such formula. This means that if an agent knows a given formula (i.e. he/she thinks that the formula is true), he/she will consider possible only the states contained in the set identified by π^{ext} . For now, it is only possible to extend π to negations, conjunctions and knowledge formulas, we will then extend it also to trust formulas. The function π^{ext} is defined recursively as follows:

$$\pi^{\text{ext}}(p) = \pi(p) \quad (1)$$

$$\pi^{\text{ext}}(\neg\varphi) = S \setminus \pi^{\text{ext}}(\varphi) \quad (2)$$

here $S \setminus \pi^{\text{ext}}(\varphi)$ is the set-theoretic complement of $\pi^{\text{ext}}(\varphi)$ with respect to the whole set of possible states S .

$$\pi^{\text{ext}}(\varphi \wedge \psi) = \pi^{\text{ext}}(\varphi) \cap \pi^{\text{ext}}(\psi) \quad (3)$$

$$\pi^{\text{ext}}(K_i(\varphi)) = \{s \in S \mid \forall t \in S \text{ s.t. } sR_it, t \in \pi^{\text{ext}}(\varphi)\} \quad (4)$$

The valuation of knowledge formulas should be read intuitively as follows: to check if a given state s is a member of the valuation set of the formula $K_i(\varphi)$, take all states t of the system that are accessible from the state s ; if all those states t are members of the valuation set of the formula φ , then the state s is a member of the valuation set of the formula $K_i(\varphi)$; repeat the process for every state of the system (since the set of states S is finite, the process will eventually end). Before extending π^{ext} also to trust formulas, we are required to extend the family of functions $\mu_{(i,c,\varphi)}$. To obtain such an extension, we define another family of functions $\tau_{(i,c,\varphi)}$. The $\tau_{(i,c,\varphi)}$, one for each formula φ of the language, are defined over the possible states of the system and associate with every state $s \in S$ a rational number in the interval $[0, 1]$, i.e. $\tau_{(i,c,\varphi)} : S \rightarrow [0, 1]$. The family $\tau_{(i,c,\varphi)}$ depends on a trust relevance space T and the members of such family sum up all the relevance weights of the propositional constants that hold in the state s taken as argument. This sum represents the upper bound of trust in φ for each state. Intuitively, a function $\tau_{(i,c,\varphi)}$ indicates how much trust an agent has in the formula φ in each state, if he/she is aware, in the state, of all the relevant information related to that φ (i.e. he/she knows what relevant propositions are true in that state). Another way to put it is the following: if an agent knows exactly which is the current state, $\tau_{(i,c,\varphi)}$ indicates the amount of trust he/she has in φ . Thus, $\tau_{(i,c,\varphi)}$ is an ideal measurement of trust. The function is defined explicitly as follows:

$$\tau_{(i,c,\varphi)}(s) = \sum \mu_{(i,c,\varphi)}(p), \text{ s.t. } s \in \pi^{\text{ext}}(p) \quad (5)$$

Note that since $\mu_{(i,c,\varphi)}$ is additive to 1, we are guaranteed that $\tau_{(i,c,\varphi)}$ itself never exceeds 1. We assume that if the state to which $\tau_{(i,c,\varphi)}$ is applied is not contained in any $\pi^{\text{ext}}(p)$, then $\tau_{(i,c,\varphi)}$ assigns to it the value 0:

$$\text{If there is no } \pi^{\text{ext}}(p) \text{ s.t. } s \in \pi^{\text{ext}}(p), \text{ then } \tau_{(i,c,\varphi)}(s) = 0 \quad (6)$$

We can now define $\mu_{(i,c,\varphi)}^{\text{ext}}$ for all formulas. Intuitively, the function $\mu_{(i,c,\varphi)}^{\text{ext}}$ assigns a trust relevance weight to all formulas of the language, i.e. $\mu_{(i,c,\varphi)}^{\text{ext}} : \mathcal{L} \rightarrow [0, 1]$. Note that $\mu_{(i,c,\varphi)}^{\text{ext}}$ also depends on all the elements on which $\mu_{(i,c,\varphi)}^{\text{ext}}$ depend. This allows for the possibility of taking φ both as parameter and as argument of the function. $\mu_{(i,c,\varphi)}^{\text{ext}}$ takes the sum of the relevance weights of all worlds in which a formula holds, i.e. the members of the valuation of the formula, and then divides the result for the cardinality of the valuation of the formula. It is extremely important to understand the behavior of $\mu_{(i,c,\varphi)}^{\text{ext}}$, because this function is the one properly defining trust in our formal language. The explicit definition of $\mu_{(i,c,\varphi)}^{\text{ext}}$ is the following:

$$\mu_{i,c,\varphi}^{\text{ext}}(\psi) = \left(\sum \tau_{(i,c,\varphi)}(s), \text{ s.t. } s \in \pi^{\text{ext}}(\psi) \right) / |\pi^{\text{ext}}(\psi)| \quad (7)$$

Two remarks must be made about (7): first, it should be noted that the $\mu_{(i,c,\varphi)}^{\text{ext}}$ of a formula, again, never exceeds 1; furthermore, the reader should note that $\pi^{\text{ext}}(\psi)$ is yet to be defined for trust formulas and, therefore, the function $\mu_{(i,c,\varphi)}^{\text{ext}}$ is undefined for this typology of formulas. The reason is that, in our language to obtain the valuation of a trust formula, we need to be able to apply the function $\mu_{(i,c,\varphi)}^{\text{ext}}$ to knowledge formulas first. This should be expected, since the main intuition behind the language presented in the paper is that trust depends on the knowledge of agents.

To properly define $\mu_{(i,c,\varphi)}^{\text{ext}}$ for all formulas, we must return first to the function π^{ext} and see how this function can be fully extended to include in its domain trust formulas. To obtain the valuation π^{ext} for trust formulas, we now present a complete procedure. This procedure indicates what are the step necessary to obtain the set of states in which a given trust formula hold and a by-product of the procedure will be that of understanding how to compute the value of a trust formula in each state.

The procedure has 6 steps:

1. We start with the trust formula $\omega_i(\varphi) \geq b$ that we want to evaluate. We consider the trust relevance space T of the formula φ . Note that the set $X_{i,c}$ of any formula will only contain propositional constants and, particularly, it will not contain any occurrence of trust formulas⁶.

⁶ This is an extremely important point, since it avoids infinite regresses in the procedure and, consequently, in the evaluation process of trust formulas. If we allow arbitrary formulas in the qualitative relevance set, then an infinite regress might happen, for example, if the trust formula we are evaluating ($\omega_i(\varphi) \geq b$) appears in the qualitative relevance set of the formula for which we are trying to determine trust (φ). In such case, when we apply the recursive procedure and try to determine the valuation of $\omega_i(\varphi) \geq b$, the same formula will appear again in the qualitative relevance set of φ , ab infinitum.

2. Given a state $s \in S$, we take the formulas ψ s.t. $s \in \pi^{\text{ext}}(K_i(\psi))$ and ψ does not contain occurrences of trust formulas (i.e. no subformula of the formula is a trust formula)⁷.
3. We take the conjunction of the formulas identified in step 2 and we manipulate it to transform it in the equivalent simplified Conjunctive Normal Form (CNF), i.e. a CNF on which redundant formulas are eliminated and the annihilation and absorption law have been applied⁸.
4. We compute the $\mu_{(i,c,\varphi)}^{\text{ext}}$ of the formula obtained in step 3. This is a rational number in the range $[0, 1]$. Note that, given our conditions during the procedure, we are guaranteed that $\mu_{(i,c,\varphi)}^{\text{ext}}$ is properly defined for such a formula, since there is no occurrence of trust formulas at this point.
5. We compare the value obtained in step 4 with the value b that appears in the trust formula we are evaluating. If the value is more than or equal to b , we say that $\omega_i(\varphi) \geq b$ holds in the state $s \in S$ and we therefore add it to the set $\pi^{\text{ext}}(\omega_i(\varphi) \geq b)$.
6. We return to step 2 and repeat the process for another state of the system. Note that, since the set S of states is finite, the procedure will eventually end.

When the procedure ends, the result is the valuation set of the formula $\omega_i(\varphi) \geq b$, i.e. $\pi^{\text{ext}}(\omega_i(\varphi) \geq b)$. Even if not strictly necessary for the evaluation of the formulas of our language, we also complete the extension of function $\mu_{(i,c,\varphi)}^{\text{ext}}$ to all formulas by giving the valuation function of trust formulas.

We will now provide truth-theoretical conditions for the satisfiability of a formula. Given a model M , a state $s \in S$, a context $c \in \text{Ctx}$ and a rational number b in the range $[0, 1]$, the satisfiability conditions are defined as follows:

- $(M, s, c) \models p$ iff $s \in \pi^{\text{ext}}(p)$;
- $(M, s, c) \models \neg\varphi$ iff $s \in \pi^{\text{ext}}(\neg\varphi)$;
- $(M, s, c) \models \varphi \wedge \psi$ iff $s \in \pi^{\text{ext}}(\varphi \wedge \psi)$;
- $(M, s, c) \models K_i(\varphi)$ iff $s \in \pi^{\text{ext}}(K_i(\varphi))$;
- $(M, s, c) \models \omega_i(\varphi) \geq b$ iff $s \in \pi^{\text{ext}}(\omega_i(\varphi) \geq b)$.

Given the above satisfiability conditions, we define four concepts of validity. A formula is *context-valid* with respect to a structure if, and only if, it is satisfied by every state of the system, once a specific context is given. A formula is *state-valid* with respect to a structure if, and only if, it is satisfied in every context, once a specific state is given. A formula is *model-valid* if, and only if, it is both *context-valid* and *state-valid*. A formula is *fully-valid* if, and only if, it is *model-valid* for all possible models. This structure is sufficient to be able to reason about knowledge and trust.

⁷ This condition is similar in spirit to the one we impose on the qualitative relevance set. See note 3 for further explanations.

⁸ Note that in our language a tautology is the neutral element for conjunction and the destructive element for disjunction, while a contradiction is the neutral element for disjunction and the destructive element for conjunction.

The above presented language is sufficient to reason about knowledge and trust and their interrelationship. The language provides a good way to compute pre-trust values and therefore can be employed as a trust-computing component of a computational trust model. We will now show how it is possible to use this language to feed the trust manipulation component of Jøsang's Subjective Logic.

4.3 From Knowledge to Trust: Pre-trust Computations

The aim of our pre-trust computation is to obtain the three distinct components of Subjective Logic's opinions. Such components are respectively, belief, disbelief and uncertainty. We will now show that obtaining those three components is straightforward in our system, once our semantical structure is given. We start by specifying the three opinion components: "agent i believes in proposition p " (symbolically $b_i(p)$) means that agent i , the trustor, believes, to a given degree, in the truth of proposition p ; "agent i disbelieves in proposition p " (symbolically $d_i(p)$) means that agent i disbelieves, to a given degree, in the truth of proposition p ; finally, "agent i is uncertain about proposition p " (symbolically $u_i(p)$) means that agent i does not possess any relevant information on whether to trust or not the proposition p . In our case, it is possible to connect the amount of information with the cardinality of the set attributed by π^{ext} to the simplified CNF of the known formulas. Specifically, the smaller the cardinality, the higher the amount of information possessed and viceversa.

The three components are obtained in the following way:

1. We start by considering the trust relevance space T corresponding to a given context and a given agent for the formula p .
2. Assuming the actual state is $s \in S$, we take the formulas ψ s.t. $s \in \pi^{\text{ext}}(K_i(\psi))$ and ψ does not contain occurrences of trust formulas.
3. We take the conjunction of the formulas identified in step 2 and we manipulate it to transform it in the equivalent simplified Conjunctive Normal Form (CNF). We label this formula with Φ .
4. We compute $\pi^{\text{ext}}(\Phi)$.
5. For each state $s \in \pi^{\text{ext}}(\Phi)$, compute $\tau_{(i,c,p)}(s)$.
6. Identify the maximum and the minimum value among the results of step 6.
7. $b_i(p)$ is equal to the minimum value identified; $d_i(p)$ is equal to 1 minus the maximum value identified; $u_i(p)$ is equal to the difference between the maximum and the minimum values.

The three components so defined form the basis of subjective logic's opinions and they can therefore be manipulated by Jøsang's model to obtain further trust values. We believe that augmenting subjective logic with the trust computing component that comes from the application of the interpretational structure of the language we proposed is an improvement of the original model presented by Jøsang.

4.4 Summing Up

We will explain the intuitive idea behind the semantical structure we just presented and explain why we believe this structure is able to capture formally the traditional notion

of trust. We will do so by describing how each component of the system contributes to the generation of trust values.

In our language, we take propositions as pieces of information. Each proposition tells us something about the system and, doing so, it helps us in determining the state of the system. The set S of possible states includes, initially, all the possible combinations of propositional constants: this is equivalent to a setting where there is complete ignorance, i.e. the system can be in every possible state. On top of this initial setting we start constructing ideal settings. To do so we first identify which information is relevant for trust in a formula in each context (i.e. evaluation scenario): this role is fulfilled by the structure T , which takes formulas of the language and returns the relevant information for such formulas. Obviously, this step is subjective in nature and this is the reason the X set depends both on agents and contexts. Each agent might consider different information as relevant in different contexts and therefore there is a different X set for each couple (i, c) . However, determining what is relevant isn't sufficient. We are dealing with a formal version of trust and this makes it necessary that trust is somehow measurable. For this reason, we have the μ function, that assigns measures to the relevant information, telling us how much each piece of information is relevant to the trust in a given proposition. Again, this measure is subjective in nature and therefore must depend both on agents and contexts. Once obtained the T structure, we can now compute trust values for ideal situations. This is the role of the τ function. What a τ function does is to take states and determine the trust value in the formula in such a state, if an agent can determine univocally that that state is the only possible one. To this extent, the value of the τ function and that of the μ^{ext} function is the same, if the formula to which μ^{ext} is applied univocally identifies the state to which τ is applied. As we said, though, this is an ideal situation and often, in the real world, agents do not possess enough information to univocally determine a state over the others. For this reason, we introduce the function μ^{ext} , which tells us how much trust must be placed into a given formula, when that formula identifies a subset of S . This is the reason μ^{ext} depends on the function π^{ext} (i.e. the function that identifies the states compatible with the information carried by a given formula). One possible critique is that we take the average of the ideal trust values of the remaining possible states and this is not justified, since we might want to take the minimum or the maximum value among the ones available. To justify our choices, we rely on the principle of sufficient reason. For the principle of sufficient reason, if an agent has no sufficient reasons for preferring a state to another, he should attribute to each state the same probability. Since we are assuming that agents can't determine which is the actual state among all the ones compatible with his/her information (if he/she could determine the actual state, we would again return to the reflections on ideal settings), he/she has no way of preferring one trust value over the others. Note that this might lead, in some scenarios, to trust a formula in a situation where there shouldn't be trust and distrusting a formula in a situation where there should be trust. We believe that this is not fully problematic, since the traditional concept of trust is open to the same issues and therefore it might be that the problem is inherently connected with trust and not specifically with our formalism. Nonetheless, it is possible to change the computation of μ^{ext} to consider optimistic and pessimistic attitudes in the part of the agent. Note, however, that this would only avoid

one part of the problem of misplaced trust by enhancing the other part (e.g. an optimistic approach would avoid the possibility of not trusting when trust should be warranted, at the expense of having many more scenarios in which the agent trusts when trust shouldn't be warranted). For those reasons, at least at the theoretical level of this work, we prefer using the average.

This exhausts the intuitive description of our formal language. We will now include a concrete example, to show a practical application of our language to a real scenario.

4.5 Example

In this section we build a concrete example and show how our formal language (especially the semantical structure) helps in analyzing the example.

Assume we have 2 agents (Anne and Bob) and 2 contexts (Fixing_the_car and Preparing_dinner). Imagine we have a third agent, Charlie, Anne's father and a car mechanic. What we are trying to evaluate in this example is the trust Anne and Bob place in the proposition "Charlie will help me". We will call the agents A and B, the contexts with F and P and the proposition "Charlie will help me" with φ . Imagine the world is completely described by 4 propositional constants: p_1 : Charlie has a master degree in engineering; p_2 : Charlie has taken cooking classes; p_3 : Charlie offers a guarantee when he repairs a car; p_4 : Charlie is a meticulous person.

Given the four propositional constants, we have 16 possible states of the world, representing all the possible combinations of the four propositions. We will label the states with the letter s , with subscripts. The initial valuation π of the propositional constants is the following: $\pi(p_1) = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8\}$; $\pi(p_2) = \{s_1, s_2, s_3, s_4, s_9, s_{10}, s_{11}, s_{12}\}$; $\pi(p_3) = \{s_1, s_2, s_5, s_6, s_9, s_{10}, s_{13}, s_{14}\}$; $\pi(p_4) = \{s_1, s_3, s_5, s_7, s_9, s_{11}, s_{13}, s_{15}\}$.

The R for each agent are the following: $R_A = \{(s_1, s_3), (s_3, s_5), (s_5, s_7), (s_2, s_4), (s_4, s_6), (s_6, s_8), (s_9, s_{11}), (s_{11}, s_{13}), (s_{13}, s_{15}), (s_{15}, s_{17}), (s_{10}, s_{12}), (s_{12}, s_{14}), (s_{14}, s_{16})\}$; $R_B = \{(s_1, s_2), (s_2, s_5), (s_5, s_6), (s_6, s_9), (s_9, s_{10}), (s_{10}, s_{13}), (s_{13}, s_{14}), (s_3, s_4), (s_4, s_7), (s_7, s_8), (s_8, s_{11}), (s_{11}, s_{12}), (s_{12}, s_{15}), (s_{15}, s_{16})\}$ ⁹.

We will now specify the trust relevance space for the three couples (A, F), (B, F) and (A, P) and the proposition φ . We start with the X sets: $X_{(A,F)}(\varphi) = \{p_4\}$; $X_{(B,F)}(\varphi) = \{p_1, p_3, p_4\}$; $X_{(A,P)}(\varphi) = \{p_2, p_4\}$.

We now apply μ : $\mu_{(A,F,\varphi)}(p_4) = 1$; $\mu_{(B,F,\varphi)}(p_1) = 0.5$, $\mu_{(B,F,\varphi)}(p_3) = 0.4$, $\mu_{(B,F,\varphi)}(p_4) = 0.4$; $\mu_{(A,P,\varphi)}(p_2) = 0.6$, $\mu_{(A,P,\varphi)}(p_4) = 0.4$.

We now have all the elements to assess how much trust an agent has in the proposition φ in each possible scenario. Let's imagine we want to evaluate the two formulas $\omega_A(\varphi) \geq 0.5$ and $\omega_B(\varphi) \geq 0.8$. We will evaluate the first formula for two contexts in the same state, i.e. the contexts Fixing_the_car and Preparing_dinner in the

⁹ For this example, we are assuming that the R relations are equivalence relations; however, we omit the couples generated by the reflexive, symmetric and transitive closure of the relations specified in the example. The reader should consider them present.

state s_7 . We will evaluate the second formula for the same context in two different states, i.e. the context `Fixing_the_car` in the states s_2 and s_8 .

Let's try to determine first whether $(M, s_7, F) \models \omega_A(\varphi) \geq 0.5$. Given the structure we presented above, it is possible to derive that in every state Anne knows whether or not her father has a master degree in engineering and whether or not he is meticulous. Therefore, she knows those facts also in s_7 , meaning that s_7 is contained both in the set $\pi^{\text{ext}}(K_A(p_1))$ and $\pi^{\text{ext}}(K_A(p_4))$ ¹⁰. We therefore identified the two propositions p_1 and p_4 . The simplified CNF of the conjunction of those two propositions is just their conjunction $p_1 \wedge p_4$. We now compute the $\mu_{(A,F,\varphi)}^{\text{ext}}$ function of this proposition:

$$\bullet \quad \mu_{(A,F,\varphi)}^{\text{ext}}(p_1 \wedge p_4) = \left(\sum \tau_{(A,F,\varphi)}(s), \text{ s.t. } s \in \pi^{\text{ext}}(p_1 \wedge p_4) \right) / |\pi^{\text{ext}}(p_1 \wedge p_4)|$$

Note that $\pi^{\text{ext}}(p_1 \wedge p_4) = \{s_1, s_3, s_5, s_7\}$ and therefore $|\pi^{\text{ext}}(p_1 \wedge p_4)| = 4$. The $\tau_{(A,F,\varphi)}(s)$ are: $\tau_{(A,F,\varphi)}(s_1) = \tau_{(A,F,\varphi)}(s_3) = \tau_{(A,F,\varphi)}(s_5) = \tau_{(A,F,\varphi)}(s_7) = 1$. This means that $\mu_{(i,c,\varphi)}^{\text{ext}}(p_1 \wedge p_4) = 4/4 = 1$. Therefore, $(M, s_7, F) \models \omega_A(\varphi) \geq 0.5$ holds. This should be expected, because for Anne to trust that Charlie will help in the context of fixing the car the only relevant fact is that Charlie is meticulous. Since she knows in s_7 that Charlie is meticulous, she has full trust in the fact that Charlie will help.

Let's now check $(M, s_7, P) \models \omega_A(\varphi) \geq 0.5$. Given that the knowledge Anne has in state s_7 is not affected by the context, part of the procedure is like the previous one, we therefore must compute:

$$\bullet \quad \mu_{(A,P,\varphi)}^{\text{ext}}(p_1 \wedge p_4) = \left(\sum \tau_{(A,P,\varphi)}(s), \text{ s.t. } s \in \pi^{\text{ext}}(p_1 \wedge p_4) \right) / |\pi^{\text{ext}}(p_1 \wedge p_4)|$$

The $\tau_{(A,P,\varphi)}(s)$ are: $\tau_{(A,P,\varphi)}(s_1) = \tau_{(A,P,\varphi)}(s_3) = 1$ and $\tau_{(A,P,\varphi)}(s_5) = \tau_{(A,P,\varphi)}(s_7) = 0.4$. This means that $\mu_{(A,P,\varphi)}^{\text{ext}}(p_1 \wedge p_4) = 2.8/4 = 0.7$. Therefore $(M, s_7, P) \models \omega_A(\varphi) \geq 0.5$ holds. This result should be expected, because for Anne to trust that Charlie will help in the context of preparing the dinner the relevant facts are that Charlie took cooking classes and that he is meticulous. Since Anne knows that Charlie is meticulous, her trust increases slightly above the indifference threshold. We now check both $(M, s_2, F) \models \omega_B(\varphi) \geq 0.8$ and $(M, s_8, F) \models \omega_B(\varphi) \geq 0.8$.

Note that in our structure, Bob knows whether p_3 holds each state (i.e. if it holds he knows it and if it doesn't hold he knows the negation of it). Again, we will only compute the values, without giving discursive explanations. The reader can check autonomously that the computations are correct. Since p_3 holds in s_2 Bob knows it. We will therefore compute $\mu_{(B,F,\varphi)}^{\text{ext}}(p_3) = \left(\sum \tau_{(B,F,\varphi)}(s), \text{ s.t. } s \in \pi^{\text{ext}}(p_3) \right) / |\pi^{\text{ext}}(p_3)| = 0.7$.

¹⁰ Strictly speaking she also knows the conjunction of the two facts, but we won't consider it, since the conjunction of the two propositions corresponding to our fact would be eliminated during the simplification of the CNF.

Therefore $(M, s_2, F) \models \omega_B(\varphi) \geq 0.8$ does not hold. Finally, in s_2 , p_3 does not hold and therefore Bob knows $\neg p_3$. We then must compute $\mu_{(B,F,\varphi)}^{\text{ext}}(\neg p_3) = (\sum \tau_{(B,F,\varphi)}(s), \text{s.t. } s \in \pi^{\text{ext}}(\neg p_3)) / |\pi^{\text{ext}}(\neg p_3)| = 0.3$. Therefore, $(M, s_8, F) \models \omega_B(\varphi) \geq 0.8$ does not hold.

When evaluating the $\omega_A(\varphi)$ opinion in the context `fixing_the_car` and assuming the actual state is s_7 , we first determine Φ , which is $p_1 \wedge p_4$. We then compute $\pi^{\text{ext}}(p_1 \wedge p_4)$, which is $\{s_1, s_3, s_5, s_7\}$. The next step is that of computing the various $\tau_{(A,F,\varphi)}(s)$, i.e. $\tau_{(A,F,\varphi)}(s_1) = \tau_{(A,F,\varphi)}(s_3) = \tau_{(A,F,\varphi)}(s_5) = \tau_{(A,F,\varphi)}(s_7) = 1$. We can finally determine the various components of the opinion $\omega_A(\varphi)$: $\mathcal{b}_A(\varphi)$ is equal to 1, i.e. the minimum among the values of the $\tau_{(A,F,\varphi)}(s)$; $d_A(\varphi)$ is equal to 0, i.e. 1 minus the maximum among the values of the $\tau_{(A,F,\varphi)}(s)$; $u_A(\varphi)$ is equal to 0, i.e. the maximum minus the minimum. We now give the result for the other three examples we examined, without providing the computations. For the opinion $\omega_A(\varphi)$ in the context `preparing_dinner` and assuming the actual state is s_7 : $\mathcal{b}_A(\varphi)$ is equal to 0.4; $d_A(\varphi)$ is equal to 0; $u_A(\varphi)$ is equal to 0.6. For the opinion $\omega_B(\varphi)$ in the context `fixing_the_car` and assuming the actual state is s_2 : $\mathcal{b}_B(\varphi)$ is equal to 0.4; $d_B(\varphi)$ is equal to 0; $u_B(\varphi)$ is equal to 0.6. For the opinion $\omega_B(\varphi)$ in the context `fixing_the_car` and assuming the actual state is s_8 : $\mathcal{b}_B(\varphi)$ is equal to 0; $d_B(\varphi)$ is equal to 0.4; $u_B(\varphi)$ is equal to 0.6. This last computation exhausts our example.

5 Conclusion and Future Work

In this paper, we first discussed the distinction between the trust computing component and the trust manipulation component of computational trust models. We then showed that classical computational models are often focused only on one of the two components. Specifically, we showed that Audun Jøsang's Subjective Logic, one of the best-suited models for trust manipulation, is lacking a proper trust computing mechanism. To overcome this downside of Subjective Logic, we proposed a logical language that can be employed to reason about knowledge and trust. Moreover, we showed how to move from our logical language to Subjective Logic. This, we believe, is an improvement for Subjective Logic and, generally, for the understanding of computational trust. In the future, the aim is to provide an actual implementation of the language and the ideas contained in this paper. Moreover, we believe it is possible to provide dynamic version of the language, which can take into consideration flow of information and time-related concerns. A final and interesting research direction is that of comparing the language proposed with others formal structures employed to represent uncertainty.

References

1. Arrow, K.: Gifts and exchanges. *Philos. Public Aff.* **1**(4), 343–362 (1972)
2. Artz, D., Gil, Y.: A survey of trust in computer science and the semantic web. *Web Semant.: Sci. Serv. Agents World Wide Web* **5**, 58–71 (2007)

3. Cho, J.H., Chan, K., Adali, S.: A survey on trust modeling. *ACM Comput. Surv.* **48**(2), 1–40 (2015)
4. Fagin, R., Halpern, J.Y.: Reasoning about knowledge and probabilities. *J. ACM* **41**(2), 340–367 (1994)
5. Grandison, T., Sloman, M.: A survey of trust in internet applications. *IEEE Commun. Surv. Tutor.* **3**(4), 2–16 (2000)
6. Halpern, J.Y.: Reasoning about Uncertainty. The MIT Press, Cambridge (2003)
7. Hughes, G., Cresswell, M.: A New Introduction to Modal Logic. Routledge, Abingdon (1996)
8. Jøsang, A., Knapskog, S.J.: A Metric for trusted systems. In: Reinhard, P. (ed.) Proceedings of the 15th IFIP/SEC International Information Security Conference, IFIP (1998)
9. Jøsang, A.: Trust and reputation systems. In: Aldini, A., Gorrieri, R. (eds.) FOSAD 2006-2007. LNCS, vol. 4677, pp. 209–245. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74810-6_8
10. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. *Decis. Support Syst.* **43**(2), 618–644 (2007)
11. Jøsang, A.: Subjective Logic. Springer, Cham (2016). <https://doi.org/10.1007/978-3-319-42337-1>
12. Kripke, S.: Semantical considerations on modal logic. *Acta Philosophica Fennica* **16**, 83–94 (1963)
13. Marsh, S.: Formalising trust as a computational concept. Ph.D. thesis (1994)
14. Mui, L., Halberstadt, A., Mohtashemi, M.: Notions of reputation in multi-agent systems: a review. In: Proceedings of AAMAS 2002, pp. 280–287 (2002)
15. Pinyol, I., Sabat-Mir, J.: Computational trust and reputation models for open multi-agent systems: a review. *Artif. Intell. Rev.* **40**, 1–25 (2013)
16. Pinyol, I., Sabat-Mir, J., Dellunde, P., Paolucci, M.: Reputation-based decisions for logic-based cognitive agents. *Auton. Agents Multi-Agents Syst.* **24**(1), 175–216 (2012)
17. Putnam, R.: Making Democracy Work. Princeton University Press, Princeton (1993)
18. Rasmusson, L., Jansson, S.: Simulated social control for secure internet commerce. In: NSPW Proceedings of the 1996 Workshop on New Security Paradigms, pp. 18–25 (1996)
19. Sabater-Mir, J., Sierra, C.: Review on computational trust and reputation models. *Artif. Intell. Rev.* **24**(1), 33–60 (2005)
20. Tagliaferri, M., Aldini, A.: A Taxonomy of Computational Models for Trust Computing in Decision-Making Procedures (forthcoming)
21. Yu, B., Singh, M.P.: Detecting deception in reputation management. In: Proceedings of AAMAS 2003, pp. 73–80 (2003)