A modal approach to model computational trust

Seifeddine Kramdi

▶ **To cite this version:**

Seifeddine Kramdi. A modal approach to model computational trust. Artificial Intelligence [cs.AI]. Université Paul Sabatier - Toulouse III, 2015. English. NNT : 2015TOU30146 . tel-01328169

HAL Id: tel-01328169
https://theses.hal.science/tel-01328169

Submitted on 7 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Présentée et soutenue le *05/10/2015* **par :**

SEIFEDDINE KRAMDI

A modal approach to model computational trust

JURY

PHILIPPE PALANQUE	Professeur	Examinateur
JAMAL BENTAHAR	Professeur	Examinateur
ANDREAS HERZIG	Directeur de recherche	Directeur
GUILLAUME FEUILLADE	Maître de conferences	co-Directeur
ANDREA TETTAMANZI	Professeur	Rapporteur
NICOLAS MAUDET	Professeur	Rapporteur

École doctorale et spécialité :

MITT : Domaine STIC : Intelligence Artificielle

Unité de Recherche :

Institut de Recherche en informatique de Toulouse (IRIT - UMR 5505)

Directeur(s) de Thèse :

Andreas HERZIG et *Guillaume FEUILLADE*

Rapporteurs :

Nicolas Maudet et *Andrea Tettamanzi*

Acknowledgements

First and foremost, I would like to thank Andi and Guillaume for giving me the opportunity to do this PhD under their supervision. Their support and mentoring extended what any student would expect from the most dedicated supervisors. I would like to ensure them that their teaching and invested time wont fade and will be passed toward any one unlucky enough to be under my supervision in the future.

I would like to thank Nicolas Maudet and Andrea Tettamanzi for accepting to review my thesis, for their detailed notes and relevant remarks. I also thank Philippe Palanque and Jamal Bentahar for the honor of having them in my jury. and joining the reviewer in analyzing and discussing my work. Sharing by the same way their own perspective on the subject.

To all my family and friends, without whom nothing would have been possible, I say to you from the bottom of my heart, thanks !

Finally I would like to thank all my friend and colleagues from the IRIT lab, especially form LILAC, ADRIA and MELODY teams. Your kindness, energy and devotion to science is the best testimony that an enjoyable and productive context of work is possible. Despite the passing of time, may this never change.

Abstract

Le concept de confiance est un concept sociocognitif qui adresse la question de l'interaction dans les systèmes concurrents. Quand la complexité d'un système informatique prohibe l'utilisation de solutions traditionnelles de sécurité informatique en amont du processus de développement (solutions dites de type dur), la confiance est un concept candidat, pour le développement de systèmes d'aide à l'interaction.

Dans cette thèse, notre but majeur est de présenter une vue d'ensemble de la discipline de la modélisation de la confiance dans les systèmes informatiques, et de proposer quelques modèles logiques pour le développement de module de confiance. Nous adoptons comme contexte applicatif majeur, les applications basées sur les architectures orientées services, qui sont utilisées pour modéliser des systèmes ouverts telle que les applications web. Nous utiliserons pour cela une abstraction qui modélisera ce genre de systèmes comme des systèmes multi-agents.

Notre travail est divisé en trois parties, la première propose une étude de la discipline, nous y présentons les pratiques utilisées par les chercheurs et les praticiens de la confiance pour modéliser et utiliser ce concept dans différents systèmes, cette analyse nous permet de définir un certain nombre de points critiques, que la discipline doit aborder pour se développer.

La deuxième partie de notre travail présente notre premier modèle de confiance. Cette première solution basée sur un formalisme logique (logique dynamique épistémique), démarre d'une interprétation de la confiance comme une croyance sociocognitive [15], ce modele présentera une première modélisation de la confiance. Apres avoir prouvé la décidabilité de notre formalisme. Nous proposons une méthodologie pour inférer la confiance en des actions complexes : à partir de notre confiance dans des actions atomiques, nous illustrons ensuite comment notre solution peut être mise en pratique dans un cas d'utilisation basée sur la combinaison de service dans les architectures orientées services [12].

La dernière partie de notre travail consiste en un modèle de confiance , où cette notion sera perçue comme une spécialisation du raisonnement causal tel qu'implémenté dans le formalisme des règles de production [10]. Après avoir adapté ce formalisme au cas épistémique, nous décrivons trois modèles basés sur l'idée d'associer la confiance au raisonnement non monotone. Ces trois modèles permettent respectivement d'étudier comment la confiance est générée, comment elle-même génère les croyances d'un agent et finalement, sa relation avec son contexte d'utilisation.

mot-clé: Confiance computationelle, logique modale, règles de production, raisonnement causal.

Abstract

The concept of trust is a socio-cognitive concept that plays an important role in representing interactions within concurrent systems. When the complexity of a computational system and its unpredictability makes standard security solutions (commonly called **hard security** solutions) inapplicable, computational trust is one of the most useful concepts to design protocols of interaction.

In this work, our main objective is to present a prospective survey of the field of study of computational trust. We will also present two trust models, based on logical formalisms, and show how they can be studied and used. While trying to stay general in our study, we use service-oriented architecture paradigm as a context of study when examples are needed.

Our work is subdivided into three chapters. The first chapter presents a general view of the computational trust studies. Our approach is to present trust studies in three main steps. Introducing trust theories as first attempts to grasp notions linked to the concept of trust, fields of application, that explicit the uses that are traditionally associated to computational trust, and finally trust models, as an instantiation of a trust theory, w.r.t. some formal framework. Our survey ends with a set of issues that we deem important to deal with in priority in order to help the advancement of the field.

The next two chapters present two models of trust. Our first model is an instantiation of Castelfranchi & Falcone's socio-cognitive trust theory [15]. Our model is implemented using a Dynamic Epistemic Logic that we propose. The main originality of our solution is the fact that our trust definition extends the original model to complex action (programs, composed services, etc.) and the use of authored assignment as a special kind of atomic actions. The use of our model is then illustrated in a case study related to service-oriented architecture.

Our second model extends our socio-cognitive definition to an abductive framework, that allows us to associate trust to explanations. Our framework is an adaptation of Bochman's production relations to

the epistemic case. Since Bochman approach was initially proposed to study causality, our definition of trust in this second model presents trust as a special case of causal reasoning, applied to a social context.

We end our manuscript with a conclusion that presents how we would like to extend our work.

Keywords: computational trust, modal logic, production inference relations, causal reasoning.

Contents

Introduction

The acceptance of Internet a mainstream medium of communication triggered a shift paradigm on what is considered an everyday interaction. New types of services are proposed to users everyday, providing them with new means to simplify everyday tasks, and generating new business models to be exploited by service providers.

In our work, we are interested in services that provide us with functionalists related to the construction of virtual communities of users in order to share and cooperate in an effortless way. By virtual communities we encompass such notions as social networks or auctioning sites.

Such new types of interaction bring with it new kinds of risks that the interactors need to handle in order to prevent any harm, especially regarding application that can threaten the privacy of the user or involve him in fraudulent financial transaction.

Such issues are exasperated by the willing of the service provider to propose to his clients, an interaction experience that mimics his "real life" interactions. While this way of *thinking applications* simplifies the user interaction by providing him a mental vocabulary to interact (like putting items in a shopping cart), such analogy often tricks the user to overshadow the risks behind his choices, by relying on his judgment on criteria unsuited for online interaction, providing new type of vulnerabilities to be exploited in cyber crimes.

Computational trust is the field of computer science that tries to tackle such issues by applying principles that we follow in our day to day real interaction to assess risks and decide upon it on how to act in virtual communities. The goal of this field of study is to adapt trust

principles, in addition to classical security solutions, in order to offer an online interactor a coherent and meaningful set of heuristics, to choose his course of action. This task is often presented as complementary to traditional (hard) security approaches.

However, trust is a concept that is both difficult to use, and difficult to describe. A trust model can be implemented in different theories, while referring to different contexts of use. To cope with such difficulties, many trust models are identified with notions related to quality of interaction, notion that is related to the service provider ability to provide what is is expected from the advertised service, given the resources that are available to him. This interpretation of trust skips in the same way the notion of deception and malicious behavior that trust is supposed to help us deal with.

Our work is positioned within the view that malicious behavior is an important subject of a trust study. To deal with it, we argue that a meaningful trust model should take into account the cognitive nature of the interactor, and the possible deceptive intentions of the others.

Our work is divided into three main chapter. Our first chapter takes the form of a global survey on computational trust that presents different perspective on how to model trust, by presenting the modelization process as composed of three parts. At first, an intermediary general trust theory need to be defined. Such semi-formal characterization is conducted either by describing trust properties, or relating trust to a set of concepts that are used to understand how trust is used. While presenting a set of trust theories, we emphasize two of the most influential trust definitions that we encountered, Castelfranchi & Falcone [24] and Marsh [49] models. The second step in modeling trust is to specify a field of application, as a way to describe both the observation and a priori assumption, that are used to assess trust, and the decision process that will use trust to guide interaction. The third step corresponds to the selection of a framework in which a trust theory will be instantiated, in this section we present trust models, categorized by such framework of applications.

The second and third chapters presents our contribution to the filed in the form of two logic-based trust models. While both our models

are designed to deal with interaction in open multi-agent systems, our two models are based on different view of computational trust.

Our first trust model is an instantiation of Castelfranchi & Falcone's socio-cognitive trust theory [15]. Our model is implemented using a dynamic epistemic logic that we propose. The main originality of our solution is the fact that our trust definition extends the original model to complex action (program's, composed services, etc.) and the use of authored assignment as a special kind of atomic actions that described the propositonal changes that an action will conduct, within its syntax. The use of our model is then illustrated in a case study related to service-oriented architecture.

Our second model extends our socio-cognitive definition of trust to an abductive framework. By doing so we can associate to a trust status explanations of such state, providing a better way to assess risks related to the current interaction. Our abductive framework is an epistemic adaptation of Bochmans's production relations [10]. Since Bochman approach was initially proposed to study causality, we feel that it is suited to describe our second view of trust as a special case of causal reasoning, applied to a social context.

We then conclude with a summary of our work, and present how we will extend our results in future studies, in a way that further our understanding of the concept of trust.

Chapter 1

Computational trust: a state of the art

Trust is one of the most studied concepts related to interaction. A commonly accepted view of trust defines it as a notion, used by humans to lower the complexity of decision making in complex social interaction, where being lethargic would be harmful. This view inspired different disciplines to use trust in different systems of interaction. Hence, trust studies can be found in sociology, philosophy, or economics, but also in industrial design, artificial intelligence and computer science (which is our main field of interest).

Computational trust, is the field of research in computer science, that tries to apply trust principles, to *virtual communities*, where "virtual communities" is the umbrella term that encompasses social networks, engineering paradigms (like Service Oriented Architecture), etc. In such applications (as was pointed out in [39]) non expert users, may lack training in assessing what courses of action lead to successful interactions. This is mainly due to the lack of understanding of both risks in interaction and metrics that one may use to assess trust in the different available actions. Implementing trust based is one of the most widely adopted solutions, to find such metrics and assess risks, to equip (human or automated) users with what is needed to make informed choices.

Since our goal is to provide a comprehensive view of the current state

of computational trust studies, we need to define a set of questions that will guide our study. Such questions will help us delimit our scope of study and define our objectives. This step is essential when dealing with such rich and versatile concept trust is. Thus, our goal is to answer the following questions:

What are the criteria to ensure that a trust model, is compatible with a given context of application? One should be aware of limits of trust based solutions and have strict criteria to distinguish applications where trust can be of some help from those where it does not. Also, depending on its uses, trust can be related to different elements of a computational system.

How to conduct a study in computational trust? In order to capitalize upon past studies, one needs to define a modular methodology. Such methodology should help one reuse past results and evaluate ones models w.r.t. others propositions. To do so, we need to take a look at models, properties, common applications and testbeds. Furthermore, defining a set of *best practices* can guide a computer designer or a researcher, on how to start building trust based computation systems in a sound manner.

What are the challenges that need to be faced to study trust? either as a follow-up to influential work or as a response to some applicative needs, researchers interested in computational trust have identified challenges to tackle first, in order to advance the field. identifying those challenges is a crucial element that needs to be dealt with in order to make our study as significant as possible.

Different surveys tried to answer those questions (in more or less depth). One of the most complete was presented by Marsh in [49], where the goal of the author was to define a trust theory, simple and general enough, to be used as a starting point for other researchers (usable in social science studies for example) to construct more complex trust models in an incremental way. Thus, Marsh's inquiry about trust is more prospective in the sense that he focused on answering our first two questions (what is trust and what methodology one should follow

to study it). on the other hand Jøsang presented in [39], a survey that focuses on uses of trust in electronic commerce. By restricting the context of study, he was able to identify categories of trust models, suitable or not for such application. Finally, Sierra & al. in [63] structured their study around a set of dimensional aspects that describe trust models w.r.t. to the type of observations that they take as input and their applicative nature. They were able to present a comprehensive comparison of different trust models, and discuss their applicative contexts.

By comparison, our survey is motivated by the vision that a theory of trust is mainly shaped by choices that are inherent to the way trust will be used. Such choices can be categorized under three main themes:

Trust Construction: In almost all studies that we encountered, the starting point of view of the inquiry, is the social notion of trust. In order to define its computational counterpart, we often restrict the properties and the set of concepts that trust is related to. This process is the first step to define a trust model.

Theory of formalization: Once a conceptual view of trust is defined, a theory of formalization needs to be selected. Choosing a theory instead of another, may change the nature of the concept (from a theoretical or practical point of view), a simple example of this would be the formalization of Castelfranchi & Falcone trust that was conducted in a logical framework in [33] and in a probabilistic framework in [52]. Our approach differs from this two approaches mainly by taking studying trust in complex actions, and taking in account more then one intervening (trustee) in the interaction.

Application framework: Formalizing the context of application defines another restriction upon a definition of trust, where assumption like assuming failure of interaction due to the context, taking into account the capacity of interactors to achieve some tasks or the eventuality of malicious behavior, may change the semantic behind our trust model, and consequently what kind of interactional issues it tackles.

These three aspects of trust correspond to the following three sections, with the remark that the theories of application, will be presented before presenting the theories of formalization of trust, and this in order to introduce first the problems that computational trust is supposed to resolve, before presenting how trust models work.

This chapter will then be concluded by a a discussion that summarize our thoughts about the field of trust studies and presents a set of challenges, that we deem crucial, to the well advancement of the field.

1.1 What is computational trust

Current technologies enable new ways of cooperations. In our daily life, it is virtually impossible to stay away from at least, some sort of virtual communities. Unfortunately, our inadequacy to asses risks and malicious behavior under virtual contexts, grows up with the richness of available services [63].

Security engineers try to deal with this problem defining mechanisms that protect users, predict risks, or at least minimize the negative effects of bad interactions. By their very own nature, such mechanisms show their limitation when implemented in *open system*. Using Internet the main representative of such systems, assumptions like secure communication channel, or the presence of a central authority, that will ensure the correct execution of a policy of interaction and will validate transaction can typically *not* be assumed. Also, traditional security solutions are often constructed to deal with a limited set of possible threats, whose relevance depends on the resources and time that the system designer invests in the design of the solution. Furthermore, threats that target the human factor in computational systems are the most difficult to deal with [4]. This inadequacy of traditional security methods calls for a new approach toward security, that is, more suited to the complexity of current computational systems.

This is where the notion of trust enters the scene, as a way to tackle traditional security drawbacks. The goal of any trust study is to find mechanisms, additional to traditional or ('hard') security mechanisms, that mimic human social interactions and (in particular) trust assess-

ment and thereby help us avoid, or at least minimize the risks of bad interactions [39].

The difficulty to characterize trust is inherent to the properties that such informal concept exhibits. Some of these difficulties can be summarized into:

- the need to describe the relations that exist between trust and other cognitive concepts like regrets, rationality and awareness (defined as the ability of the agent to observe the context of interaction). The more concepts are taken in account in a trust model, the more that model would be complex to study and implement;
- the necessity to define formal objects, that quantify an agent *trust state*, in a way that is pertinent to a decisional framework. For example reducing trust to a probability of successful interaction may seem a little too abstract to be used in many application with intelligent interactors. Knowing that a trust formalization is often used with different decisional processes; and
- finally, the most problematic element is to define a model of trust whose impact upon an interactional system can be identified and studied. Proving that using trust in a given situation would impact positively the experience of the interactor, is a very difficult task to fulfill.

In what will follow, we will present a set of trust definitions, relevant for —but not limited to—the notion of computational trust in open systems. We will emphasize two of the most influential trust definitions that we encountered, the Castelfranchi & Falcone [24] and the *Marsh* [49] models.

1.1.1 Castelfranchi socio-cognitive model of trust

The main goal of Castelfranchi & Falcone (henceforth abbreviated C&F), is to deal with delegation in online interaction. C&F argued that the future of e-commerce is closely related to helping interactor asses the current risk state of any given interaction, which can only be

possible by defining a set of mechanisms that would construct a simple abstraction to the current interaction.

The C&F approach emphasizes the strong relation between trust and delegation. They rely on the view that *trust is the mental background of delegation*. In other words, the decision that an agent (the truster) takes, to delegate a task to an other agent (the trustee), is conditioned by some specific beliefs and goals. This mental state is what they call *trust*.

According to C&F, trust is a predicate Trust(i, j, α, φ), that describes the trust that an agent i (called the truster) has in an agent j (called the trustee), in achieving one of i's goals φ, by executing the action α. such trust is observed if:

1. i has *the goal* φ;
2. i believes that j is *capable* to do α;
3. i believes that j has the *power* to achieve φ by doing α and
4. i believes that j intends to do α.

For example, if i trusts j to send him a book B that he bought, then (1) i's goal is to possess B, and i believes that (2) j is capable to send B, (3) that j's sending B will result in i possessing B (i.e., that the post delivery will not loose the book), and finaly (4) that under the condition that j receives the payment, j has the intention to send B.

C&F trust model is based on a set of assumptions, used to conceptualize the notion of trust [14]:

Assumption 1: Only a cognitive agent can *trust* an other agent: we cannot talk about trust without implying a beliefs and goals of the truster.

Assumption 2: Trust is a mental state, that describes a complex *attitude* of an agent i toward the capability (or the context of execution) and the willingness of an agent j to achieve i's goals, by performing an action. We note that this does not imply that the agent j is necessary cognitive: one can trust a chair to hold him if one sits down on it.

Assumption 3: Trust is the mental counterpart of delegation, where delegation is perceived as a social construct.

C&F observe that future business models and technological advances in communication, will put the human element as a central part of distributed artifical intelligence, or what we call now multi-agent system (*mas*). Thus, *mas* will be a predominant framework to either simulate or to implement such applications [24].

In such context, users loss of landmark to distinguish good from bad interactions, or even to assess risks behind their choices, justifies the use of trust, as one of the key elements in application fields such as electronic commerce [16].

In order to illustrate the use of their model in a decision framework, C&F described a simple extension of their model in [15], where they define the notion of graded trust. This notion of graded trust was used in a decision tree to implement delegation problems.

While they decompose trust into more basic concepts, C&F do not use a specific notion of belief, neither a way to compute quantifications of a graded version of the different concepts involved in the definition.

Adapting the C&F model of trust to different contexts of application and to different background theories was attempted in further studies, the most notable being:

A probability model proposed by Yves Demizeau [52] which used a Bayesian network to concatenate the different beliefs components of C&F definition, into a probability-based trust quantification.

A formalization based on modal logic was presented in [33], where the trust ingredients of the C&F model are decomposed in more fine grained ingredients, implemented in the *Belief, Desire, Intention* (BDI) framework.

A possibilistic model was proposed in [19].

This implementations will be detailed in the next sections. We also note that the trust model presented in chapter 2 of this work, is also a formalization of this definition.

1.1.2 Marsh modelization of trust

The model proposed in [49] by Marsh, is one of the earliest models of computational trust. The main goal of this model is to propose a trust theory, upon which different disciplines, including non technical ones, like social sciences, can formalize their own vision of trust.

The Marsh model only takes into account direct interaction. It defines trust as a numerical value, generally between -1 and 1, that can refer to three types of trust:

Basic trust which models the general trusting disposition of the truster, independently of the identity of the trustee. It is calculated from all the experiences accumulated by the truster. Good experiences lead to a greater disposition to trust, and vice versa. The author uses the notation T_x^t to represent the trust disposition of the truster x at time t.

General trust is the common notion of trust that the truster associates to a trustee but in a context free manner (as opposed to the C&F definition presented before). The notation $T_x(y)^t$ denotes the quantification of the trust that x has toward y at time t, without any context of interaction in mind.

Situational trust quantify the amount of trust that one agent has in another, taking into account a specific situation. The utility of the situation, its importance and the *General trust* are the elements considered in this definition in order to calculate *Situational trust*. Marsh presented the following basic formula to calculate this type of trust:

$$T_x(y,\alpha)^t = U_x(\alpha)^t \times I_x(\alpha)^t \times \widehat{T_x(y)^t}$$

where x is the truster, y the trustee and α the situation at time t. $U_x(\alpha)^t$ represents the utility x gains from α, $I_x(\alpha)^t$ is the importance of the situation α for agent x and $\widehat{T_x(y)^t}$ is the new estimation of general trust after the current interaction; i.e., if t is the current time, the truster x will aggregate all situations $T_x(y,\alpha')^{t'}$, with $\theta < t' < t$ and α' similar or identical to the present situation α. θ

and t define the temporal frame that the agent considers. Only the experiences within that interval of time will be taken into account for the aggregation.
In order to define $\widehat{T_x(y)}$ the author proposes three statistical methods: the mean, the maximum and the minimum. Each method is identified with a different type of truster: the optimistic, that takes the maximum trust value from the last $t - \theta$ experiences that he had, the pessimistic, that uses the minimum trust value, and the realistic, that calculates the value as an average value using the formula $\widehat{T_x(y)} = \frac{1}{|A|} \sum_{\alpha \in A} T_x(y, \alpha)$, where A is the set of situations similar to the present situation α available from the last frame of time.

As presented by Marsh, these trust values are used by the truster to decide if it is worth to cooperate with the trustee or not. But a such decision mechanism needs to take in account other parameters, like the importance of the action to be performed, the risk associated to the situation and the perceived competence of the trustee.

Marsh introduces also two mechanisms for trust deprecation, which are reciprocation and deprecation over time. Reciprocation implements the idea that if an agent does not reciprocate a cooperative behavior, our trust toward him should decrease. Deprecation over time is self explanatory (over time and without interaction, trust over less tested trustee decreases).

Marsh defines a decision process to choose between cooperate or not (a binary choice). The truster decision is based on the notion of threshold of interaction.

We recall that Marsh's main goal was to define a theory of trust that can be extended and adapted to specific situations. To do so, Marsh followed Karl Popper's principles to define scientific theories [49]. Marsh does not see his trust model (or theory) as a definitive one, but hoped to define a model that furthers our understanding of trust, and helps others reuse it by being:

- Circumscriptive, by delimiting the context of study,

- Simple, by following the Occam's Razor,
- Repeatable in different context of study, and
- Flexible, by allowing to link it to further concepts.

As a result, Marsh pursued his study in [48] by relating trust to the concepts of forgiveness and regret after bad interaction.

In contrast to Castelfrenchi & Falcone's model which define trust in term of decomposition into more primitive concepts, Marsh's theory of trust is more interested in its relations to other identified concepts relevant to interaction. Defining types of trust allows to use different sets of properties and related concept, depending on the context of application.

1.1.3 Other approaches

Different studies of trust proposed a characterization of the concept to suit some applicative context. Some of the most notable one that we encountered are:

Adum proposed a trust model based on risk in decision making [43]. Using McKnight and Chervany's work, he defines trust as *the extent to which one party is willing to depend on somebody, or something, in a given situation with a feeling of relative security, even though negative consequences are possible.* This definition is adopted to see trust as a concept that is situational and related to risks. This definition of trust is applied to financial transaction. By restricting the context of application the author is able to quantify risk (using classical gambling theory) by defining the expected monetary value EV of the interaction as:

$$EV = I \sum_{n}^{i=1} p_i G_i \tag{1.1}$$

where p_i is the probability of outcome i and G_i is the gain factor on the monetary investment I in case of outcome i.

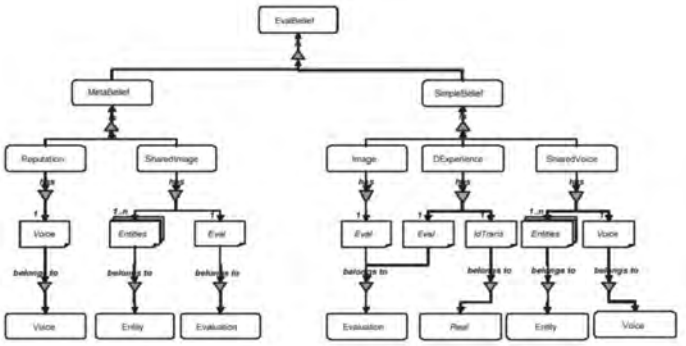

Figure 1.1: The taxonomy of Pinyol Trust language

Upon this definition, Adum identifies some risk attitude, together with risk functions, that depend on the context of interaction, This models assumes a binary outcome $i \in \{s, f\}$. with $G_s \in [0, +\infty]$ being the gain factor (associated to the success outcome s) and $G_f \in [-1, 0]$ considered as the loss factor associated to the failing outcomes f.

The expected gain is then defined as:

$$\begin{aligned} EG &= pG_s + (1-p)G_f \\ &= p(G_s - G_f) + G_f \end{aligned} \tag{1.2}$$

RepAge: In [56], reputation and trust mechanisms are defined as a control artifact, aimed to ensure well-behaving behavior by promoting a certain type of interaction within a society(of agents), by rewarding good behaving agents. In this model, trust is not just a score, but a mental image constructed by the truster. Such mental image corresponds to a set of beliefs about the future performance of the trustee. Those images (corresponding to the notion of role in game theory) and beliefs are ordered within an ontology of concepts(figure 1.1).

The ontology defined by Pinyol is a set of related beliefs(semantically described as a set of atomic predicates) that have some social valuation (meaning that an observation can be used to assert if a the

predicate is true or false), those beliefs are divided into *simpleBelief* and *Metabelief*(a belief about other agent beliefs).

In such ontology, complex concepts (like trust and reputation), are constructed using other concepts that can be evaluated.

The European Commission Joint Research Center (ECJRC) defines trust as the property of a business relationship, such that reliance can be placed on the business partners and the business transactions developed with them [75].

Norton Deutsch presented a descriptive definition of trust applied to psychology [22] that is very similar to Von Neumann & Morgenstern definition of rationality. The definition goes as follows: *" When an individual (the truster) is confronted with an ambiguous path, that may lead to two outcomes (Va+) and (Va-), where the first outcome is perceived as beneficial and the second one is perceived as harmful. If he decides to interact while perceiving that the outcome depends on the behavior of another individual (the trustee), then we say that the truster trusts the trustee."*

As Deutsch states, this definition is descriptive. It underlines the subjectivity of trust, by associating to it two dimensions: our utility function and how we perceive the outcome utility in one hand, and our point of view of the situation (or our capacity to acquire and use observation to derive our subjective trust) in the other hand.

Deutsch was one of the first to note that trust is mostly about how agents use observations (or objective knowledge of a situation) to generate subjective assumptions that define the state of trust of an agent, toward some future interaction, meaning as he puts it, that *"People (and here, we include agents) are in their own ways psychological theorists and tend to make assumptions about the behavior of others."*

Niklas Luhmann studied trust in a more practical way, compared to Deutsch, using a study of the prisoner dilemma. Luhmann

presented trust as a sociological emergent artifact, meant to reduce the complexity of society, where the complexity related to the number of variables that need to be taken in account in order to have a deterministic vision of the world is so high, that an individual needs to reduce such description in order to adapt to his surrounding. Trust is then one of those abstractions, meant to handle risks [46].

Diego Gambetta's definition of trust is one of the most influential. Trust is defined as the threshold of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he monitors the action and in a context in which it affects his own actions. This definition is the basic foundation upon which the Castelfranchi & Falcone definition was built to handle trust in delegation.

1.2 Application fields

As announced in the introduction of this survey, we will now present a set of application fields and theoretical framework, in which trust plays a crucial role. The fact the current section precedes the section where we talk about trust models, is our way to emphasize that trust studies are highly influenced by their application framework, and that trust formalization, is only justifiable if we can show that using it under that context, enhances the performance of the system.

Now the notion of usefulness of a trust study, as the following section will show, can be either directly observed in its implementation, by solving some security or performance issues (as a heuristic, a security policy or a protocol of communication), or it can allow us to understand furthermore how trust and related concepts can be used in some social setting (by means of simulation and prediction models of virtual and real communities).

By an application field, we mean a systematic description of a context of interaction. This definition covers for example social and peer-

to-peer networks, service based applications, in which cognitive agents [1] observe and interact with each other, in order to achieve common or concurrent goals. The main interest in such systems, are related to how agents will decide to interact.

Once an application field is defined, trust is added to the system by identifying it with a heuristic agents use in their interaction protocol. In this setting, the goal of a trust study is to understand how a trust model affects the behavior of the system (hopefully by increasing the performance of that system, for example by decreasing the overall number of failed interaction interactions are more frequent then *Failed* one).

Integrating trust in an applicative field, can be done in different ways that affects the system in different manners. We identify three of these effects:

- *Direct effects*, where trust will be used by a truster agent, as a way to lower the indeterminism of his perception of the system, and provide some insights on how to succeed in more (if not all) his interaction,

- *Indirect effects*, which model the effect of adding trust for both the truster and the trustee. While the truster will integrate trust in his interaction protocol, the trustee will also adapt his protocol of interaction if he would like to continue cooperating with the other agents. The perception of trust will then act as a deterrent, since trust perception influences the utility of trustee agents (especially perceivable in systems with monetary outcomes). Finally,

- *Recursive effects*, as the most complex possible integration of trust: the truster, by knowing that the trustee understands that trust is a component of the truster interaction protocol, may use this information to learn some things about the inner mechanics that govern the trustee, or use it to simplify interaction (for example by taking advantage of trust to minimize the cost of usual verification

[1] We use the term cognitive agent here loosely as a synonym of a complex agent those behavior is not entirely known by the other agents of the systems

on the trustee competences). Meanwhile, the trustee may use this perception either to manipulate how he is perceived in the system to look more trustworthy (in harming scenarios, this is often done with the help of accomplices agents and their recommendations).

Defining which sort of trust integration may give different type of results depends on the system, and may be more or less suited to some application fields.

In the remaining of this section, we will present a set of frameworks used to model such systems, where for each system, we will try to emphasize how trust can modify interaction protocols, both for the truster and the trustee, and how we can measure the impact of such a modification.

1.2.1 Decision theory

Decision theory, in a large sens, encompasses a set of fields, interested in characterizing decision making processes. In other words, decision theory is concerned with goal directed behavior in the presence of options.

Applications of decision theory range from economy to statistics, passing by psychology, political and social science, to philosophy. This plethora of application fields explains the interdisciplinary nature of modern decision theory.

A decision theory can be either normative or descriptive. Normative approaches study how decisions should be made (given some notion of rationality), while a descriptive approach is more interested in how decision are actually made.

The use of norms is the main way a decision theorist describes the context of a decision problem. The notion of norm is not restricted to the notion of rationality, but refers as any type of constrains (usually social, political, ethical, etc.) that are used as a foreground for the process of choosing between different options. Such sets of norms are meant to be used as external variable of the decision study.

Our interest in this survey, is when those norms are partially defined. Such assumption has to be made when we are interested in decision

making under uncertainty. As most of trust definitions showed in the last section, such is where trust can be of use.

A decision theory is characterized by:

- a representation of the notion of alternative options,
- a representation of the preferences between such options (quantitative or qualitative assessment), and
- the set of principles (or rules) used to make the decision.

1.Representing alternatives Representing alternatives can be done using different spaces. The commonly used ones are *alternatives*, *states of nature*, and *decision matrices*.

Alternatives: Alternatives are defined as courses of action that are available to the decision maker at the time of the decision. The set of alternatives can be more or less well-defined. The set of possible alternatives can be either open (new alternatives are discovered over time), or closed (static). We can divide decisions with closed alternative sets into two categories: those with voluntary and those with involuntary closure. In cases of voluntary closure, the decision maker has himself decided to close the set (as a first step in the decision). In cases of involuntary closure, closure has been imposed by others or by impersonal circumstances. It is commonly assumed that the set of alternatives is closed and that its elements are mutually exclusive.

States of nature The effect of a decision depends not only on our choice of an alternative and how we carry it through. It also depends on factors outside of the decision maker's control. Some of these external factors are known, they are the background information that the decision maker assumes. Others are unknown. They depend on what other persons will do, and features off nature that are unknown to the decision maker. As an example, consider the decision to whether or not to go to

an outdoor concert. The outcome (whether I will be satisfied or not) will depend both on natural factors (the weather) and on the behavior of other human beings (how well the band performs). In decision theory, it is common to summarize the various unknown external factors into a number of cases, called *states of nature*.

Decision matrices The standard format for the evaluation-choice routine in (individual) decision theory is that of a decision matrix. In a decision matrix, the alternatives open to the decision maker are tabulated against the possible states of nature. The alternatives are represented by the rows of the matrix, and the states of nature by the columns. Mainstream decision theory is almost exclusively devoted to problems that can be expressed using *utility matrices*. While, most modern decision theoretic methods require numerical information, in many practical decision problems we have much less precise value information (perhaps best expressed by an incomplete preference relation). However, it is much more difficult to construct methods that can deal effectively with non-numerical information.

2.Representing preferences Using preferences helps to define a notion of order among alternatives. Given a preference rule, their is different ways to choose between alternatives:

1. An alternative is (uniquely) best if and only if it is better than all other alternatives. If there is a uniquely best alternative, choose it. There are cases in which no alternative is uniquely best, since the highest position is *shared* by two or more alternatives. In this case, the obvious solution is to pick one of the alternatives, no matter which. In a more general setting the next rule can be used:
2. An alternative is (among the) best if and only if it is at least as good as all other alternatives. If there are alternatives that are best, pick one of them. However, there are cases in which not even this modified rule can be used to guide decision making. An example would be cyclic preferences relations. In

those cases, rationality criteria such as transitivity are often not useful to guide decisions because for a cyclic and transitive relation all choices are equal (or incomparable).

3.Principles of decision Different principles for decision making were proposed over time. One can cite the *Condorcet* methods that uses a three steps decision process that refines at each step, the available alternatives to choose from, *Modern sequential models*, or Non-sequential models. To study the relation between trust and decision theory, we focus our attention on those related to the two most predominant one, *utility maximization*, *Expected utility* and *Bayesianism*.

Utility maximization The rule of maximization is almost endorsed in a universal manner. Once the utility of the decision maker is numerically represented, the basic rule of utility maximization is simple to apply: *choose the alternative with the highest utility. If more than one alternative has the highest utility, pick one of them (no matter which).*

The use of the *rule of maximization*, is justified mostly in economic theory, by assuming that individuals maximize their benefits, as measured in money. Utilitarian moral theory also postulates that individuals should maximize the utility resulting from their actions.

This approach have various withdraw related to the difficulty to design a numerical utility function for the decision maker, while in most real cases, one can act in an irrational manner. To cope with such withdraws, we often assume certain assumptions on the decision maker, these assumptions gave raise to the expected utility method for decision-making, that we present in the next item.

Expected utility Presented as the major paradigm in decision-making under uncertainty, expected utility (EU) theory, assigns to each alternative, a weighted average of its utility values under different states of nature, and the probabilities of

these states are used as weights. This is why EU is called in some cases *probability-weighted utility theory*. This also means that in order to calculate expectation values, one must have access to reasonably accurate estimates of objective probabilities. In some applications of decision theory, these estimates can be based on empirically known frequencies. Such assumption is in many cases too strong to be significant in practice, especially when one can only construct subjective probabilities based on ones own limited observation. The reliability of probability estimations depends on how small is the difference between objective probabilities and subjective ones. This relation between objective and subjective probabilities was one of the main reasons to develop the Bayesian approach.[2]

Baysianism approach to decision making Expected utility theory with both subjective utilities and subjective probabilities is commonly called Bayesian decision theory, or Bayesianism. The idea behind this paradigm is to take in account the subjective nature of decision making. In such paradigm, the decision maker will follow three main principles:

- The decision maker has a coherent (in the mathematical sense) set of probabilistic beliefs.
- The set of probabilistic beliefs of the decision maker is complete. In other words, to each proposition, the decision maker can associate a subjective probability.
- When acquiring a new evidence, the Bayesian decision maker changes his beliefs in accordance with his conditional probabilities, where a conditional probability $P(A \mid B)$, denotes the probability of the proposition A knowing B, $P(A)$ denotes then the probability of A, given everything that we know. with:

$$P(A|B) = P(A \wedge B)/P(B)$$

[2]To be noted that while *utility maximization* applies to deterministic contexts, *expected utility* is seen as a generalization of this principle to contexts with uncertainty

as the only rationality criteria (according to Savage [64]).

- Finally, the decision maker chooses the option with the highest expected utility.

Bayesianism is more popular among statisticians and philosophers than among more practically oriented decision scientists. An important reason for this is that it is much less operative than most other forms of expected utility due to the set of assumptions that it requires. Also, theories based on objective utilities and/or probabilities more often give rise to predictions that can be tested. It is much more difficult to ascertain whether or not Bayesianism is violated.

The drawbacks of decision theory are inherent to how it is used in practice, for example the use of closed sets of alternatives is not suited for the kind of application fields that we are assuming here. In virtual communities, new ways of interaction (and sadly, new mischiefs) are discovered by the user of the system, during its entire life cycle. Assuming the possibilities of interaction is a closed set, especially when defining the security policies of the system, is not a viable assumption.

Another impracticality of decision theory, is that it relies on numerical utility function, which are difficult estimate, and prone to error.

Introducing trust in decision theory framework can be done in different manners. The first way is to perceive trust as a social normative notion. In this case, trust is an external input of the decisional system, upon which we can rely in defining the decision process. Such aspects are mainly found in economics and the theory of organizations, with little to no study available for computational trust.

Another way to see trust, is as an intermediary process in decision making, that simplifies the representation of the alternatives, the space of states of nature. In this case, trust would encapsulate the expertise of the decision maker to leave only in the description of the context of interaction, elements that are pertinent to choose the best alternative. Such uses are associated to probabilistic trust models (which will be presented in the next section). Also, Castelfranchi & Falcone used their trust definition to define a decision making process based on decision

trees and threshold of interaction [24]. This implementation represent a typical use of this principle.

In the next subsection, we will present Game theory as an applicative framework for trust models. It is worth mentioning that Game theory can be seen as part of decision theory, but we argue that it deserves a study by itself due to the tremendous body of work related to the use of trust in this framework.

1.2.2 Game theory

Trust has been extensively studied in economics, and in the context of game theory in particular. The usual game theoretic framework for analyzing trust is that of repeated games, in which some players are uncertain about the payoff structures of their opponents.

Such framework encompass the two major notions that justify the use of trust: uncertainty upon the other players' payoff structures and repetition. These two assumptions justify the presence of a mechanism to cope with such uncertainty. A third characteristic inherent to game theory, is the fact that our payoff is a function of other agent actions, which make cooperation enforcement, an important study of game theory. In this section we will present three games that encompass these three notions (the prisoner dilemma, Bayesian games and repeated games), for each of these games, we discuss the role trust plays.

This also means that we concentrate on the so called strategic games, in which players choose their actions simultaneously. This concept can be formally defined as follows.

Definition 1 *A* strategic game *consists of:*

- *A set of players* $\mathbb{N} = \{1, 2, \cdots, n\}$,
- *for each player* i*, a set of actions (or strategies)* A_i *and*
- *for each player* i*, a utility function* $u_i : A \to \mathbb{R}$*, where* $A = A_1 \times A_2 \times \cdots \times A_n$.

Prisoner dilemma(PD): The prisoner's dilemma is a canonical example of an interaction framework, analyzed in game theory. Such analysis shows why two purely rational individuals might not cooperate, even if it appears that it is in their best interests to do so.

Following the description given in [58], this game presents the following situation: Two members of a criminal gang (A and B) are arrested and imprisoned. Each prisoner is in solitary confinement with no means of speaking to or exchanging messages with the other. The prosecutors do not have enough evidence to convict the pair on the principal charge. They hope to get both sentenced to a year in prison on a lesser charge. Simultaneously, the prosecutors offer each prisoner a Faustian bargain. Each prisoner is given the opportunity either to: betray the other by testifying that the other committed the crime, or to cooperate with the other by remaining silent. Here is the offer: If A and B each betray the other, each of them serves 2 years in prison. If A betrays B but B remains silent, A will be set free and B will serve 3 years in prison (and vice versa). If A and B both remain silent, both of them will only serve 1 year in prison (on the lesser charge).

Using the standard matrix notation for players payoffs, the following matrix represents the game.

	C	D
C	1,1	3,0
D	3,0	2,2

Speaking generally, one might say that an instance of PD is a game in which a *cooperative* outcome is only obtainable when every player violates rational self-interest. There is no accepted pure solution [3].

[3] In game theory, accepted solutions are solutions that satisfy a conceptual notion of rational strategy, in this case, strategies that are a Nash equilibrium

Baysian games In many applicative scenarios of game theory, different players have an asymmetric access to information about some important aspects of the game, including a special case in which some players know more than others and may use this information to get better payoffs. Bayesian games are used to study such situation.

Informally, the central notion of Bayesian games is the notion of player types. It is used to describe a player knoweldge about the other participant of the game. Each player is aware of his own type, but is uncertain about the other players type. The type of a player, will specify his utility function, and the probability distribution that defines his knowledge about the other player's type. The goal of every player is to maximize his payoff for any of the other player's types. A player will then need to form a belief about the other player's strategies, given his knowledge of the type distribution before making any decision. Of course, all beliefs must be consistent.

Definition 2 *A* Bayesian game *consists of a set of players* $N = \{1, 2, \cdots, n\}$*, where to each player* i*, we associate:*

- *a set of possible actions* A_i
- *a set of possible types* T_i
- *a probability function* $p_i : T_i \to \Delta(T_{-i})$*, where* $T_{-i} = \times_{j \in N \setminus \{i\}} T_j$ *and* $\Delta(T_{-i})$ *is the space of probability distributions upon* T_{-i}*.*
- *a utility function* $u_i : A \times T \to \mathbb{R}$*, where* $A = \times_{i \in N} A_i$ *and* $T = \times_{i \in N} T_i$

The main concept of rational solution to Bayesian games is based on the notion of Bayesian equilibrium.

Definition 3 *Any* strategy profile $\sigma^* \in \times_{i \in N} \times_{t_i \in T_i} \Delta(A_i)$ *is a Bayesian equilibrium if for any type* t_i *of any player* i *mixed strategy* $\sigma^\star(a_i \mid t_i)$ *maximizes player* i*'s expected payoff. Where the*

expectation is taken over all combinations of types of the other players.

According to Bayesian decision theory, a rational player should choose a Bayesian equilibrium.

Repeated games To model repeated interactions, game theorists use the concept of repeated games in which the same so-called stage game is repeated a finite or infinite number of times.

To define a repeated game and its equilibrium we need to define the player's strategy sets and payoffs for the entire repeated game, given the strategies and payoffs of its constituent stage game. Therefore, we assume that the stage game is an n-player strategic game and that the action set of each player i, A_i, is finite. The stage game payoffs are defined in the usual way, as maps $u_i : A \to \mathbb{R}$, where $A = \times_{i \in N} A_i$. With A_i, the space of all strategies of player i. Assuming that the players observe each other's realized pure strategies at the end of each stage and are aware of them when choosing the next stage strategies we can proceed as follows: let $a^t = \langle a_1^t, \cdots, a_n^t \rangle$ be the pure strategy profile relized in period t and $h^t = \langle a^0, \cdots, a^{t-1} \rangle$ the history of these profiles for all periods before t. Finally, H^t denotes the set of all such period-t histories.

Then a period-t strategy of player i in the repeated game is any mapping $\sigma_i^t : H^t \to A_i$. A player strategy i for the whole repeated game is then a sequence of such maps $\sigma_i = \{\sigma_i^t\}_{t=1}^{\infty}$.

At all stages of the game players receive some payoffs. Meaning that comparing possible payoff, sums to comparing series. There are three criteria adopted in the literature to do so: time-average, overtaking and discounting criterion. Each one of this criteria correspond to a different view of past, present and future outcomes value.

Concerning the prisoner dilemma, this game can be applied in many applicative context(some of them not involving a prison sentence), like

online e-commerce. In such systems, the traditional way to solve this issue (by assuming mixed strategy games) is not viable (no one is actually seeing as a good solution that a seller flip a coin to know if he should send or not a good, or the buyer to send the money). Trust is then a way to implement common belief of cooperation.

Repeated games emphasize the ties that good behavior may have upon interaction, intuitively, because the players can condition their future play on the past plays of their opponents and can stop to cooperate if the opponents do not play in a specific way. The quality of interaction affects players trust, which in return affects the trustee payoff. This emphasizes a kind of rationality built over time and through interaction. One should also stress that such type of interactions are also very risky, because someone can take advantage of such system (think e.g. eBay frauds), which underlines the necessity to carefully craft a trust system.

Finally, Bayesian games show that uncertainty is difficult to tackle, deciding what to do under such framework is often based on the expected utility principle presented in Section 1.2.1.

We remark that the relation between trust and game theory, can be seen from two perspectives. (1) If studying trust effects of such games is pursued from the point of view of the game designer, who would play the of an external agent that take in account, the game description as a whole, then we trust can be associated to some normative notion. It can take the aspect of a reputation mechanism in some cases. (2) Working with individual trust amounts to defining a localized mechanism, where agents have access to limited information (for example, only the interactions that were conducted by them in repeated games), or they may have access to all information, but adopt a trust modelization that is subjective and based on principles that are proper to them (for example a proper type probability distribution in a Bayesian network).

Game theory is a very successful framework to deal with decision making within uncertainty. Nevertheless it was remarked (i.e., in [1]), that the assumption of such framework, are applicable to system design, where the different utility function of agents are available to the

system designer, such assumptions makes game theory, a very difficult framework to implement trust based solution, to help interaction in open systems.

1.2.3 MAS

A multi-agent system (MAS) is a computerized system composed of multiple interacting intelligent agents within an environment. Multi-agent systems can be used to solve problems that are difficult or impossible for an individual agent or a monolithic system to solve.

MAS can be used to model different types of interaction, which usually correspond to systems with different types of agents. Such large definition makes multi-agent formalization a field that encompasses a large set of visions of what is cooperation, with both abundance of paradigms upon which a formalization can be constructed, and theories of implementation that instantiate those paradigms.

An important class of MAS, is that of open multi-agent systems. In comparison with other MAS paradigms in open MAS, agents can freely join and leave at any time and where the agents may hold different aims and objectives that are not necessary compatible with other agents of the systems. Agents can also assume different identities in the system, even at the same time. These features are sources of the following trust related issues in MAS:

1. In most case, the agents are likely to be self-interested and may be unreliable, such property comes from the fact that such agents, may be representative of concurrent parties (i.e. competitive companies), that needs to cooperate, in order to achieve respectively their personal goals.

2. No agent can know everything about its environment, either due to the unavailability of the information or due to its cost (either in terms of time or in terms of computational or monetary resources), it is also difficult to assume instant monitoring capabilities, due to the high dynamical nature of the system.

3. As a result of the impossibility to track the definitive identity of the agents, it is difficult (or even impossible) to implement a system where a central authority can control all the agents, either by imposing normative restriction on interactions, or mechanisms to sanction bad behavior.

In relevant scenarios, despite the uncertain nature of the interaction system, agents cannot afford not to interact, due to the fact that their goals are unachievable without external help. To decide his course of action, an agent would need to cope with incomplete knowledge about his environment and other agents. As we have seen, trust plays a central role in facilitating these tasks.

In our case, we will illustrate the different formalisms that were used to describe multi-agent systems, and were used to study trust, while emphasizing representation of epistemic agents. We will focus on MAS that are based on logical frameworks.

BDI frameworks What we call the belief, desire, intension (*BDI*) models are descriptions of the agents' cognitive states, constructed upon claims originally proposed by Bratman [13] on the role of intentions in practical reasoning. Specifically, Bratman argued that rational agents will tend to focus their practical reasoning on the intentions they have already adopted, and will tend to bypass full consideration of options that conflict with those intentions.

Bratman's view of agency was implemented in different frameworks. The one of interest here, is Cohen and Levesque's logic, which is the best example of the use of modal logic to represent BDI models. We choose to concentrate on modal logic representation, since it was the framework that we chose to implement the trust model that we present in the next chapter, for more details on such type of logics we recommend [59].

Cohen and Levesque's logic accounts for BDI models, by using basic modal notion of beliefs (using a KD45 operator), the notion of desire is associated to the notion of preferences and goals, while the notion of intention is defined, using different modal operators of the logic:

- dynamic operator as defined in Linear Propositional Dynamic logics (PDL),
- belief operators, and
- preference operators.

The semantics of such logic is based on the notion of frame, as a quadruple $M = \langle W, R, B, P \rangle$, where:

- W is a non-empty set of possible worlds;
- $R : (I \times A) \to (W \times W)$ maps an authored action π, to an accessibility relations R_π;
- $B : I \to (W \times W)$ maps agents i to accessibility relations B_i, used to represent agents beliefs;
- $P : I \to (W \times W)$ maps agents i to accessibility relations P_i, used to represent the agent's preferred worlds.

Such frames satisfy the following constrains:

- $\langle W, R_i \rangle$ is a linear transition system;
- every B_i is serial, transitive and euclidean, which are the standard constrains to define a KD45 belief operator;
- $P_i \subseteq B_i$, for every $i \in I$, which makes the preference of agents, rational, or compatible with their beliefs.

The notion of intention is defined as follows:

1. i has the *achievement goal* if i *prefers that* φ is eventually true and believes that φ is currently false. Formally:

$$\text{AGoal}_i\varphi \equiv \text{Pref}_i F\varphi \wedge \text{Bel}_i\neg\varphi$$

2. i has the *persistent* goal that φ if i has the achievement goal that φ and will keep that goal until it is either fulfilled or believed to be out of reach. Formally:

$$\text{PGoal}_i\varphi \equiv \text{AGoal}_i\varphi \wedge (\text{AGoal}_i\varphi) \cup (\text{Bel}_i\varphi \vee \text{Bel}_i G\neg\varphi)$$

$$\text{Intend}_i\varphi \equiv \text{PGoal}_i\varphi \wedge \text{Bel}_i F\exists\alpha\text{Happ}_{i:\alpha}\varphi$$

Cohen and Levesque succeeded in providing a fine-grained analysis of intention by relating that concept to action, belief and realistic preference. Such definition describes a mental state of an agent that acts, in accordance with his belief and intentions. Under such setting, trust can be characterized as a special case of belief (as we will see in section 1.3.3). From a normative perspective, trust can be defined as a condition of the context of interaction, whose presence compensates partially the uncertainty of the outcome of interaction.

Dynamic epistemic logic *Dynamic Epistemic Logic* (DEL) can be seen as the study of how to reason about model change. The main interest of the field, is to characterize how agents can share knowledge, update it and revise it.

Dynamic epistemic logic is often defined by describing three elements: the *epistemic part* of the logic, or how agents represent their epistemic state, A *dynamic part* , describes external interaction of agents, those interactions can change both the current state of the system, and the agents knowledge. This changes can be either noticed or not by the agents, such interaction can be dynamic operators, or announcement operator. Finally, *the belief change and update element* describes how the dynamic part and the epistemic part interact. Some representative of

- Public announcement logic [73]
- Epistemic PDL [74]
- BMS logic [7]

ABC that we present in the next chapter, is also a dynamic epistemic logic, which uses authored assignment to model interaction in a multi agent system.

Multi-agent systems that can be described in dynamic epistemic logic, can take advantage of a trust model in different ways. Trust can be a conditional element of the framework (implemented as a formula of the logic modeling what an agent knows), to decide

if the disclosure of some information would be judicious or not. Trust can be used to implement a delegation system, where one may delegate action to more efficient agents. Finally, trust can be used within the process of belief change and update, when faced with incoherent information from different sources, to define aggregation processes.

Agent-based negotiation language Sierra & al. presented in [67] a multi agent framework to study trust in agent negotiation. While the trust model was probabilistic (as an implementation of entropy estimation in information theory), the presented MAS, or at least the communication language of the agents was based on a logical framework.

When an agent α is negotiating with an opponent β, they aim to strike a deal $\delta = \langle a, b \rangle$ where a is α's commitment and b is β's.

A denotes the set of all possible commitments by α, and B the set of all possible commitments by β. The agents have two languages, C for communication and L for internal representation of interaction.

L is thus a dynamic language constructed upon the following atomic actions:

$$Atm = \{\text{Offer}, \text{Accept}, \text{Reject}, \text{Withdraw}, \text{Inform}\}$$

with the following syntax and informal meaning:

- Offer(α, β, δ): agent α offers agent β a deal $\delta = \langle a, b \rangle$ with action commitments a for α and b for β.
- *Accept*(α, β, δ): agent α accepts agent β's previously offered deal δ.
- Reject$(\alpha, \beta, \delta, [\text{info}])$: agent α rejects agent β's previously offered deal δ. Optionally, information explaining the reason for the rejection can be given.
- Withdraw$(\alpha, \beta, \delta, [\text{info}])$: agent α breaks down negotiation with β. Extra info justifying the withdrawal may be given here.

- Inform(α, β, [info]) agent α informs β about info.

Where [info] can be either of a (i) dymanic nature (related to an agent process used to solve a problem), or (ii) beliefs of the agent including preferences.

Using these atomic actions that can be performed between agents, a dialog can be constructed in order to trigger reaction from other agents. The existence of a shared protocol between agents, to ensure the coherence of communication need to be assumed (e.g. an offer cannot be accepted if it was not made).

On the other hand, the content language expressed by an agent (of which [info] is a part) is a dynamic epistemic language that allows to represent agents beliefs, knowledge, and different types of preferences and conditionals.

Some simple examples of what can be represented in this language are the following:

- "I prefer red wine to white wine when served meat." as:

  $$\text{Inform}(\alpha, \beta, \text{if Food} = \text{meat then Wine} = \text{red} > \text{Wine} = \text{white})$$

- "I prefer more money to less money" as:

  $$\text{Inform}(\alpha, \beta, \text{soft}(\text{tanh}, \{\text{Money}\}))$$

  where tanh is an ordering function, and soft expresses a preference constrain.

- "I reject your offer since I definitely cannot pay more than 200 euros" as:

  $$\text{Reject}(\alpha, \beta, \text{Money} = 250, \text{hard}(\text{Money} < 200ee, \{\text{Money}\}))$$

  where hard express a practical, or a "deal breaker" constrain.

- and "I prefer red cars to yellow cars" as:

  $$\text{Inform}(\alpha, \beta, \text{if thing} = \text{car then Colour} = \text{Red} > \text{Colour} = \text{Yellow})$$

An agent's set of beliefs, is constructed using a sequence of such formulas. Using this knowledge base, the agent can enter in a negotiation phase, guided by his beliefs (perceived as a probability distribution) estimated using the information theory concept of entropy.

The main issue with such an interaction protocol, comes from the uncertainty related to the current situation, where, the more "focused" an agent is (with less probability dispersion over the possible context states), the more efficient his actions will be. Trust in such situation is associated to a measure of the dispersion of an agent's belief probability.

Trust plays a crucial part in any multi-agent system that models complex agents, interacting within an uncertain and risky context. The relevance of MAS is related to the set of applications that can be implemented in this formalism which ranges over a wide variety of networked computer systems such as Grid computing [25], the Semantic Web [36], ubiquitous computing systems [45], peer-to-peer systems [65], etc.

1.3 Trust models

The previous section presented a set of application fields that did put some light on how trust can be integrated in interaction systems, and how a trust model should tackle in order to be significant. The outcome was the view that trust is either:

- a normative notion, that can be observed, or desired in a cooperative system,
- a process, that is part of a communication protocol, whose goal is to help interacting in uncertain and risky situations,
- or a piece of information that is either acquired, or given, and represents an abstraction of the current state of the system, that is significant for the agent's future tasks.

In this section we present a set of trust models, that will be of these different types. We order them by their theories of implementation, which are:

Social network approaches: that uses an ad hoc aggregation function, to predict the behavior of probabilistic agents. Such aggregation function can not generally be described using a probabilistic semantic.

Probabilistic approaches: the success of probability theory in representing uncertainty makes it a straightforward choice, to implement trust models, statistical inference methods are often used to calculate trust, often as the probability of future successful interactions.

Logical approaches: logical models of trust are of two kind, epistemic models that represent trust as a cognitive notion linked to the belief state of a given agent, or a protocol model, that given the description of a communication system (for example, a sensor network) would infer a statement on interaction to be trustworthy or not.

1.3.1 Social network

Social networks models of trust, target probabilistic peer behavior and are mainly characterized by ad hoc feedback aggregation strategies. Normally they imply the aggregation of all trust and reputation information available in the system.

Beth, Borcherding, and Klein [8] presented one of the early examples of a social trust model that takes in account, both personal interaction and recommendations. In their model, an agent can either enter in direct interaction (d), or recommend an agent(r). A recommended agent can either give his opinion about direct interaction with the trustee(d), or recommend an other agent (r), etc. Thus, a feedback in this model is a binary valuation of an

interaction or a recommendation $W = \{r, d\} \times \{0, 1\}$, for example $\langle d, 1\rangle$ correspond to a successful direct interaction, while $\langle r, 0\rangle$ correspond to a failed recommendation or a recommendation that lead to a bad interaction.

The feedback aggregation algorithm starts by aggregating all direct and indirect interactions with direct acquaintances in the network to form direct trust valuation using the formula $v_d^{ij}(p) = 1 - \alpha^p$, where i is the truster agent, j is the trustee, p is direct positive experience (the amount of $\langle d, 1\rangle$ associated to j) and α being a parameter such that $0 < \alpha < 1$. In this step, if i has experienced at least one bad interaction with j, he should put $v_d^{ij}(p) = 0$. The second step is to evaluate recommendation using v_d^{ij}. If i is the truster and j is a recommender, p is the positive and n the number of negative interactions with j, the recommender, recommendation trust is defined as $v_r^{ij}(p, n) = 1 - \alpha^{p-n}$ if $p > n$ and 0 otherwise. The exact value of parameter α is left to the discretion of the truster.

By means of such computation we can actually order agents of the virtual community in a graph, where the agents ar the nodes of the graph. Between two nodes of this graph there are at most two edges, one representing direct interaction between the source node and the target node, and the second, corresponding to the evaluation of the source node recommendations. A peer that would evaluate trust with another of the network, would take in account, all paths, that goes from him (the truster) to the evaluated node (the trustee) such that we choose paths that are constructed exclusively with recommendation edges, except for the last edge which would represent a direct interaction. Such mechanisms propagate trust along the graph. Given a truster i_0 and a trustee i_k and a *trust path* i $\langle v_r^{i_0 i_1}, v_r^{i_1 i_2}, \cdots, v_r^{i_{k-2} i_{k-1}}, v_d^{i_{k-1} i_k}\rangle$, the quantification of trust expressed by the path is calculated using the function:

$$v_{\text{path } i} = 1 - (1 - v_d^{i_{k-1} i_k})^{v_r^{\overline{i_0 i_{k-1}}}}$$

with $v_r^{\overline{i_0 i_{k-1}}} = \prod_{n=1}^{k-1} v_r^{i_{n-1} i_n}$.

Trust path values are then aggregated in *group path*, where a group is defined by paths that share the same penultimate node. Given a group of m paths $\langle \text{path}_1, \cdots, \text{path}_m \rangle$, the aggregation function is as follows:

$$v_{\text{group}\langle \text{path}_1, \cdots, \text{path}_m \rangle} = \sqrt[m]{\prod_{j=1}^{m} (1 - v_{\text{path } j})}$$

Finally, groups trusts are aggregated to form the trust value. which for m groups, corresponds to $v = 1 - \prod_{i=1}^{m} v_{\text{group i}}$.

From a complexity point of view, this algorithm would perform poorly, w.r.t. the context of application it was proposed for (peer to peer networks) due to the number of peers that can reach thousands of nodes. Such complexity can be lowered, by involving the other peers by propagating the calculation in the network, or introducing heuristics to select paths taken in account. For example Yu and Singh [77] presented a polynomial time feedback aggregation, that sees recommendation and direct interaction as the same notion. While the aggregation processes of this approach is actually similar to Beth's approach, the main difference is that Yu and Singh only aggregate paths that carry maximal values form the source to all the neighbors of the destination node.

Richardson, Agrawal, and Domingos [60] presented an efficient algorithm to calculate trust, by merging a trust multi-graph over a network of agents. A trust multi-graph is simply a graph, where nodes are agents, and labeled edges represent the value of trust that an agent has toward another (where, a trust value is a real number). The aggregation approach considers the matrix $M \equiv [M_{ij}]_{i,j=1}^{N}$, where N is the number of agents of the system. it is supposed that M has been normalized such that for every $1 \leq i, j \leq N$:

$$0 \leq M_{ij} \leq 1 \text{ and } \sum_{k=1}^{N} = 1$$

The elements of the obtained matrix will be used to change its own values, for example, since the value M_{ij} represents the direct trust estimation of i about j, the line M_{iX} of the matrix, corresponds to the opinion of the agent i about the whole network, while the column M_{Xj} corresponds to the opinion of network about j; knowing this, an opinion M_{ij}, altered by the opinion of the network on i (M_{Xi}) which itself is altered by changes in M_{ij}, this means that an aggregation algorithm would iterate the matrix (by aggregating and concatenating trust values), until the Matrix converges.

Richardson, Agrawal, and Domingos matrix based aggregation, can be seen as a generalization of different aggregation algorithms that can be found in the literature. Such algorithms include PageRank, which is the algorithm used by Google to order web pages [54].

Xiong and Liu [76] presented a calculation method that does not require the aggregation of trust valuations within all the agents of the system, but propagates trust directly among the agents. In this work, agents express direct valuations as a rating from the interval $[0, 1]$. The main idea of the algorithm is to compute trust as an average of all the direct feedback that the network can provide about the trustee, while weighting those feedbacks by the trust value of the agents, given those feedbacks. Calculating trust t_j toward the trustee j, is calculated as follow:

$$t_j = \sum_{e \in \text{ incoming}(j)} w_e \frac{t_{\text{source(e)}}}{\sum_{f \in \text{ incoming}(j)} t_{\text{source(e)}}}$$

where incoming(j) is the set of all direct feedback about the agent j, w_e is the feedback of the agent e and $t_{\text{source(e)}}$ is the calculated trust value of the agent e.

The computation of trust is still a time consuming process, due to the iterative task of retrieving feedbacks and recursively calculating trust toward agents of the whole network.

To sum it up, social network approaches to trust (and reputation) valuation, offer a way to provide a decision system, with numerical

parameters to guide interaction, nevertheless, the semantics of these numerical values is difficult to understand, for both defining a threshold of interaction and for comparing trust estimates for different agents. Even if in some contexts, it can be associated to a probability measure, in complex systems, where the semantics of the trust valuation is crucial, it is generally considered that such approaches are not recommended, except with a lot of empirical testing [21].

1.3.2 Probabilistic modelization of trust

Probability theory, is the most widely adopted approach, to model trust, this can be associated to the historical place that probability theory take towards representing systems with uncertainty and risk.

Over time, probability theory matured to embody a set of theories and practices that use it in decision framework. When we talk about a probabilistic approach, we often refers to three different set of tools [61]:

- **Statistical inference:** statistical inference is the process of modeling and estimating a probability function related to a random process (a stochastic system),the modelization process is based on defining the form of the function that would be used to represent the system, while the inference reside on the estimation of such function, together with a quantification of the quality of such estimation (if supported by our decision process).

- **Probability theory:** probability theory it the mathematical field that group the tools that are used to study probability as a mathematical objects, their relations and properties, the main objects of probability theory are the concepts of random variables, stochastic processes and events. to be noted that probability theory generally deals with abstract (perfect) probability functions, different from estimations, as contrast with what statistical inference produces.

- **Decision theory:** the decision process is the theory that adds an applicative decisional vision upon probability theory. Defining a

decision problem sharpen the tools that will be used in statistical inference while estimating the probability function of the studied system, and the assumption that one will use within the realm of probability theory.

Since we associate decision theory to the field of application of trust in this survey. Defining a trust model in our view, amounts to identifying a probability model representing the system, a et of properties of the system as a valuation of trust (i.e., the probability of good interaction, or the average of *quality* of interaction that can be predicted, etc.), together with the statistical inference process, that, in most cases, will provide a way to estimate the probability of the system, using the history of past interactions.

Despotovic and Aberer [20] present a rather simple trust model to manage trust in peer-to-peer networks. Their model identifies trust i, to the subjective probability of the truster to have a successful interaction with the trustee j (denoted θ_j) during the next interaction. Such probability is derived form a history of binary evaluation of the quality of past interactions. A history of interaction is a tuple $\langle x_1, x_2, , \cdots, x_n \rangle$ where $(x_i \in \{0, 1\})$ which was reported respectively by the peers $\langle p_1, p_2, \cdots, p_n \rangle$. Given that a peer p_k may lie about his reported valuation x_k about the trustee j, with a probability l_k, the authors propose the following function as the probability of reporting:

$$P[x_i = e] = \begin{cases} l_k(1 - \theta_j) + (1 - l_k)\theta_j & \text{if } e = 1 \\ l_k\theta_j + (1 - l_k)(1 - \theta_j) & \text{if } e = 0 \end{cases}$$

This is then used to calculate the likelihood of observing the history $\langle p_1, p_2, \cdots, p_n \rangle$, as a function of θ_i, to be

$$L(\theta_j) = P[x_1]P[x_2] \cdots P[x_n]$$

Estimating the trust of i toward j, amounts then to estimate the probability θ_i. This is done by using the maximum likelihood principle, that stipulates that θ_j should be the value that maximizes

the probability to observe the current history, or the value that maximizes $L(\theta)$.

Hang & al. [31] present an approach that is similar to the social network model presented by Beth, Borcherding, and Klein [8]. They study how a client would estimate the trustworthiness of a service provider (or his reputation) based on a history of direct interaction provided by agents called witnesses, where the role of indirect interaction is be played by agents referred too as recommenders.

The main difference between Hang & al.'s and Beth & al.'s models, is that Hang & al.'s model makes the distinction between trust, as constructed from direct interaction (or evidence), from the trustworthiness of the agent that was reported by recommender. These two quantifications are modeled by the mean of two different spaces.

Direct trust is associated to the probability of an expected positive outcome $\alpha = \frac{r}{r+s}$, where r are past direct positive interactions, and s are the failed ones. On the other hand, indirect trust is modeled using a triple $\langle b, d, u \rangle$, where $b + d + u = 1$ that can be interpreted as weights of belief, disbelief, and uncertainty, respectively, of the probability of good interaction. We present here the update function that the authors provide for direct trust.

The authors propose an ad hoc method to update their direct trust $\langle r, s \rangle$ w.r.t. new information provided by referrers in the network, by providing their trust $\langle r', s' \rangle$. Such update is done in two steps:

- Estimating the accuracy of the recommendation, by comparing it to the truster's own estimation. This is done by calculating the ratio between the two trust values $\alpha = \frac{r}{r+s}$ and $\alpha' = \frac{r'}{r'+s'}$

$$q = \frac{\alpha(1-\alpha)}{\alpha'(1-\alpha')}$$

- Updating the trust value α, by concatenating $\langle r, s \rangle$ and $\langle r'q, s'(1-q) \rangle$ into a vector $\langle r + r'q, s + s'(1-q) \rangle$.

The authors also provide different update function, for example to account for trust deprecation overtime, that resemble Marsh's view of trust (as presented in section 1.1.2.)

Chatalic & al. [53] present a parametric trust model, meant to deal with conflict within a peer-to-peer inference system. The goal of trust in this system is to provide a way to deal with inconsistency in query answering in description logic.

In summary, the authors defines trust as the probability that the trustee has, to answer a query (associating database elements, to a query description) in a correct way. The evaluation is then a binary number (1 for success and 0 for fail). This probability function follows a Bernoulli distribution with parameter θ, defined upon the evaluation of the next interaction, as a random variable X:

$$P(X = k) = \begin{cases} \theta & \text{if } k = 1 \\ 1 - \theta & \text{if } k = 0 \end{cases}$$

Estimating (or updating) this probability function, amounts to estimate the value of the parameter θ. To do so, the authors, follow a Bayesian approach to statistical inference, which perceive θ itself as a random variable that follows a probability distribution (in this case, a *normal distribution*). The chosen value of θ will be its expected value $E(\theta)$.

Josan & al. [41] present a model that can be seen as a generalization of the precedent trust model, to graded evaluation (and not only binary evaluations). The approach uses a Dirichlet distribution function to implement for example, a system of evaluation that takes its value from the set {mediocre, bad, average, good, excellent}.

This means that, given a finite valuation space $X = 1, 2, \cdots, k$, trust is characterized by the probability distribution over valuations of X. The only constraint on this probability is the standard additivity property $\sum_{i \in \{1,2,\cdots,k\}} P(X = i) = 1$. This probability is described by a probability vector $\vec{p} = \{P(X = i) \mid 1 \leq i \leq k\}$.

In this model, the history of interaction corresponds to a k-vector $\vec{\alpha} = \{\alpha_i \mid 1 \leq i \leq l\}$. where each α_i can be one of our k valuation. The Dirichlet model that is proposed, is a direct generalization of the binomial modelization of trust

$$f(\vec{p} \mid \vec{\alpha}) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} p(X = i)^{(\alpha_i - 1)}$$
$$\text{where} \begin{cases} P(X = 1), \cdots, P(x = k) \\ \sum_{i=1}^{k} P(x = i) = 1 \\ \alpha_1, \cdots, \alpha_k \end{cases}$$

As seen in the last trust model, each evaluation is associated to a probability distribution toward its estimation, that would represent the expectation that the evaluation will be picked for the future interaction. This expectation in the case of a Dirichlet distribution, corresponds to:

$$E(p_i \mid \vec{\alpha}) = \frac{\alpha_i}{\sum_{i=1}^{k} \alpha_i}$$

The main property of a probabilistic approach to model trust is its ability to quickly converge to a good estimation of the probability model representing the system.

While the success of probability approaches comes form the acceptance that social interaction can be simulated by stochastic systems, such approaches do not take into account the recursive nature of trust, or the ability of the trustee to manipulate the behavior of the truster, in order to misbehave. Logical models, in their epistemic form, try to deal with such issues by studying a more complex models of agents.

1.3.3 Logical modelization of trust

Logical approaches to model trust falls under two main categories, depending on what they are actually trying to model. The *epistemic approach* to trust modeling tries to study the inner state of the system's agents, which comprise how they perceive other agents, how they

update such knowledge and finally, how they use it to decide how they will interact. In such approaches, trust is a cognitive concept, presented as the counter part of the agent actions. On the other hand, *protocol based approach* to trust modeling, is interested in representing the communication protocol of the interaction system as a whole. In such approaches, trust is associated to a set of rules that an agent follows to decide how to interact; in such approaches, trust is associated to the performances of the different agents and their ability to conduct successful interactions, without any regard to their intention behind their behavior. Some representative of this two approaches are:

Herzig & al. [33] present a logical formalization of Castelfranchi & Falcone model of trust, using a BDI framework (see section 1.2 for more details).

We remark that we will only present the formalization of occurrent trust of this (or trust w.r.t. a defined future delegation), since this was the trust that we presented in section 1.1

We recall that according to C&F, trust is a predicate $\text{Trust}(i, j, \alpha, \varphi)$, that describes the trust that an agent i (the truster) has in an agent j (the trustee), in achieving one of i's goals φ, by executing the action α. Such trust is the case if:

- i has *the goal* φ;
- i believes that j is *capable* to do α;
- i believes that j has the *power* to achieve φ by doing α; and
- i believes that j intends to do α.

According to Herzig & al., the notion of belief is associated to the modal system KD45,

$$Believes(i, \varphi) \equiv \text{Bel}_i\varphi$$

On the other hand, goals are perceived as preferences about the future. Its instantiation uses the standard temporal operator Eventuallyφ

(whose semantics is based on transition relations between possible states as seen in [71]). The notion of preferences is Cohen and Levesque's binary preferences $\text{Pref}_i\varphi$, which we present in section 1.2.

$$Goal(i, \varphi) \equiv \text{Pref}_i \text{ F } \varphi$$

The remaining of the predicate involved in C&F definition are defined as follows:

- *j is capable to do* α corresponds the fact that all execution of α by j succeeds:
$$\neg\text{After}_{j:\alpha}\bot$$
- *j has the power to do* α, corresponds to the fact that there is an execution of α by j that makes φ true
$$\text{After}_{j:\alpha}\varphi$$
- finally, *j intends to do* α corresponds to the fact that j prefers to execute α
$$\text{Pref}_j\text{Does}_{j:\alpha}$$

Such formalization allows to model trust in a cognitive agent taking in account the context of execution, the abilities of the agent and his intention in the realization of a given goal. Such complex division allows us to associate to trust (or distrust), conditions related to the environment, the competence of the trustee, and his willingness to cooperate. We remark that the trust model that we present in chapter 2, is an extension of this model, to action compositions [12].

Standers & al. **[69]** present a formalization of trust that is based on the argumentation framework that was proposed by Prade & Amgoud in [3]. The goal of the authors is to construct a trust model that associates to a trust valuation of a given situation, a set of arguments that justify such valuation.

In order to do so, the trust agent first constructs a knowledge base about the other agents in the system. A knowledge base corresponds to a set of tuple k_i, γ_i, where $k_i \in \mathcal{L}$ is a fuzzy formula and $\gamma_i \in [0, 1]$ is a confidence that the agent has in k_i. The valuation of a fuzzy formulas k_i in a world state w is given using a valuation function $v_w : \mathcal{L} \leftrightarrow: [0, 1]$. The value $v_w(k_i)$ quantify the applicability of the fuzzy formulas k_i in the world w.

The knowledge base regenerately contains fuzzy rules, i.e. a material implication from an observation (condition) to an expected/learned effect (conclusion). Such a rule can be partially applicable in a particular world state, instead of just being fully applicable or not at all. such rules are used to describe situations where trust is then a propositional formula that, given the description of the current situation (as a world state), would be reached or not using these rules.

In this framework, when a truster is given the choice to trust or not the trustee, this decision is reached using the following argumentation framework:

Definition 4 *Given a truster $\langle K, G, T\rangle$ where K is a knowledge base, G is a formulas describing his goals and $T = \{t, \neg t\}$ is a set of available option (to trust or not trust the trustee), an argument A in favor of a decision $\alpha \in T$ is a triple $A = \langle S, C, \alpha\rangle$, where*

- *$S \subseteq K$ is the support of the argument, containing the knowledge from the agent's knowledge base K used to predict the consequences of decision α,*
- *$C \subseteq G$ are the consequences of the argument, i.e. the goals reached by the decision α, and*
- *α is the conclusion argument A recommends.*

Moreover, $S \cup d$ entails C, S is minimal, and C is maximal among the sets satisfying the above conditions.

Given the set of all possible argument that can be generated by a truster's knoweldge base K and Goals G to decide to trust or

not a trustee $T = \{t, \neg t\}$, a decision is made by choosing the argument(s) with the higher strength. Given an argument A, such value is calculated using using the level of confidence the truster has in the support of the argument, and the weight or desirability of the outcome.

Jøsan [42] presents a model of trust based on *subjective logic* [40]. Subjective logic is presented as a belief calculus that takes into account that perceptions about the world always are subjective in nature, i.e. such view results form a limit set of observation, that can be used to estimate the probability of the view's validity. This translates into using a belief model that can express degrees of uncertainty about probability estimates. While statistical inference principles are used to estimate the validity of atomic propositions, a logical framework governs the way this proposition are combined to construct the formulas of the language.

Subjective logic is based on the notion of opinion. Where an opinion w_x^A is the opinion owned by A, x is the proposition to which the opinion applies. A is often omitted if we are working with the opinion of a single agent. For example, a binomial opinion (or an opinion that applies to a single proposition, and can be represented as a beta distribution) can be represented by a tuple $w_x = \langle b, d, u \rangle$ such that $b, d, u, a \in [0, 1]$ where $b + d + u = 1$, b is the agent's certainty about x, d correspond to his certainty about $\neg x$, and u codes his uncertainty about his opinion. special reasoning cases depending on the values of b,d and u can give a sense of what does these values corresponds to:

- $b = 1$ corresponds to the classical logical notion of trueness,
- $d = 1$ corresponds to the classical logical notion of falsity,
- $b + d = 1$ corresponds to a traditional probability,
- $b + d < 1$ corresponds to a degree of uncertainty about the opinion, and item $b + d = 0$ corresponds to total uncertainty.

The following table show some operators, proposed to combine opinions (to calculate opinions on more complex formulas):

Name	Operator
Addition	$w^A_{x \vee y} = w^A_x + w^A_y$
Multiplication	$w^A_{x \wedge y} = w^A_x \times w^A_y$
Complement	$w^A_{\neg x} = \neg w^A_x$
Deduction	$w^A_{x \| y} = w^A_x \circledcirc (w^A_{y \| x}, w^A_{y \| \neg x})$
Consensus	$w^{A \Diamond B}_x = w^A_x + w^B_x$

Where for example, the deduction operator $\circledcirc$ is a direct application of *Bayes theorem*. An interpretation of such operators (an others) can be seen in [40].

Trust toward a formula, corresponds to the opinion the truster constructed about it, which can be calculated given his opinions toward atomic propositions of the language.

Krukow & Nielsen [44] presented a trust (or reputation model) to deal with risky interaction in electronic commerce, that may involve the disclosure of private informations to untrusted parties. The authors approach describes a formal declarative language to express interaction policies, such language is based on a (pure-past) variant of linear temporal logic (LTL). Such language can be used by an agent to describe how he should interact with other agents, depending on their behavior in the past and their current state of knowledge (as described by a history of interaction).

A history of interaction correspond to a sequence of events, occurring of events and possible histories. It follows a structure that describes dependencies between events. This structure is called an *event structure*:

Definition 5 *An* event structure *is a triple* $ES = \langle E, \leq, \#) \rangle$ *consisting of a set* E*, and two binary relations on* E*:* $\leq$ *and* $\#$*.*

The elements $e \in E$ are called events, and the relation $\#$, called the conflict relation, is symmetric and irreflexive. The relation $\leq$ is called the (causal) dependency relation, and partially orders E. The dependency relation satisfies the following axiom, for any $e \in E$:

$$\text{the set } [e] = \{e' \in E \mid e' \leq e\} \text{is finite.}$$

The conflict and dependency relations satisfy the following transitivity axiom for any e, e', $e'' \in E$

$$(e \# e' \text{ and } e' \leq e'') \text{ implies } e \# e''$$

Two events are independent *if they are not in either of the two relations.*

With an event structure, past events can be used to ensure that conflicting events will not occur in the future. This is why the logic of policies presented by the authors is parametric in an event structure.

Given an event structure ES, The syntax of their language (denoted $\mathcal{L}(ES)$) is as follows:

$$\psi := e \mid \Diamond e \mid \psi_0 \wedge \psi_1 \mid \neg\psi \mid X^{-1}\psi \mid \psi_0 S \psi_1 \mid \top$$

Where e is an atomic formulas that can be read as "e is an event that was observed in the current working session", while $\Diamond e$, correspond to "e can still occur within the current session", or e is possible. The operator X^{-1} (last time) and S (since) are the usual past-time operators. other standard temporal operators can be defined as abbreviations, for example $F^{-1}(\psi) \equiv \top\ S\ \psi$ denotes the fact that ψ is true at some time in the past, while $G^{-1}(\psi) = \neg F^{-1}(\neg\psi)$ means that ψ is true at all times in the past.

Using formulas from this language (interpreted in linear models of LTL), the authors are able to express different policies of interactions. Some examples of such policies would be of a buyer (in

eBay for example) who can decide to bid on an electronic auction only if the auction is run by a seller that has never failed to send goods for won auctions:

$$\psi^{\text{bids}} \equiv \neg F(\text{time-out})$$

Furthermore, the buyer might require that the seller has never provided negative feedback in auctions where payment was made:

$$\psi^{\text{bids}} \equiv \neg F(\text{time-out}) \wedge G^{-1}(\text{negative} \rightarrow \text{ignore})$$

In this model, trust correspond to the set of policies that an agent would define, in order to describe his way of interacting with other agents, the more restrictive those policies are, the less the agent trusts the others.

Czenko & al. [18] present a logic based trust management system, as a security infrastructure (Called TuLiP)to secure content sharing in peer-to-pper networks. As in the previous work, this trust management infrastructure allows an agent of the network to specify a policy of interaction that will be enforced by the system.

Logical approaches to trust provide a way to represent more complex aspects of trust, and how trust can be integrated in complex systems of interaction. But such model suffer from logic formalization issues like the complexity of reasoning with difficult logics, the difficulty to model the situation and the necessity to manually tweak the formalization, to resolve such issues.

1.4 Discussion and conclusion

In this survey, we tackled the issue of computational trust in a step by step manner. We offered different points of reattachment to make parallels between trust models, where two trust models can share the same conceptual aspects, share the same application field or be implemented within the same theory. We can use those parallels to talk

about computational trust studies. The first element that comes out of our study is the difficulty to study trust in an abstract setting. Computational trust is often a middle element, that helps cognitive agents associate observation and knowledge about the other agents and choose between different acts, Abstracting trust limits the vocabulary that can be used to describe trust formalizations. In another hand, choosing the framework in which trust will be modeled has a great importance in the usability of the model, for example, probability models have better performances and are more easy to implement than logical applications, but are not applicable in all contexts. Finally, efficiency of trust models is often based on an intricate knowledge of the context of application, which lead to ad hoc optimization of the proposed trust model.

While constructing this survey, we identified a list of issues that we deem important to the advancement of the field. We believe that such issues should be dealt with in priority:

How to interface trust with different application fields. Dissociating trust model from an applicative context, even with the difficulty that it implies, is a necessity both to study the properties of a trust model in different context of application, and comparing different models to choose which one is better. This question translates into the interest of the community in defining testbeds to compare trust models w.r.t. a common application field, representatives of such testbeds are [70] [27]. Also, the attempt to define, universal languages to talk about trust in an abstract aspect which where attempted in [56] [57] [68]

Decision process using trust. Since trust can be seen as the cognitive counterpart of cooperative interaction, understanding the semantic of a trust model is an essential part to define a decision process that translate trust values to a sound protocol of interaction. Such understanding would help us understand the risk of a given interaction, for example to define different threshold of interaction, depending on the sensibility of the action to follow. Understanding the semantic of a trust model helps us compare trust

toward different agents and choose for example between different options between them. Such issues where underlined by [49].

Heterogeneity of trust in distributed open systems. Within an open setting like online interaction or service oriented architecture solutions, one can assume that each agent of the system has his own vision toward what construct a trusted interaction, and his proper model of trust. One should take in account this element when developing an agent instantiated to work on such type of systems. In addition to the use of a common vocabulary to talk about trust(as seen in the fist item of this list), the system may require an agent to share information about a trust valuation, while adding arguments on how such trust valuation was calculated [3], or to share feedbacks upon which a trust value was calculated to ensure its compatibility with other agents' valuations [8].

Deal with malicious agents. One of the most spread misconception is to associate trust to performance analyses of interactor. While this distinction is of little importance in simple system, it can be crucial when dealing with cognitive agents with a behavior based on high level concepts of rationality. Failure in interaction can be traced to three main aspects: the ability of the interactor, the context of interaction and finally the intension of the interactor. While the ability of the agent and the context of interaction is a widely studied subject, especially when related to the notion of quality of services QoS [66] [50], trust models should distinguish intentional misbehavior from technical difficulties while interaction. Distinguishing this two different aspects of failure is a key to identify malicious agents that may uses confusion either to mask their bad behavior behind the context, or more dangerously, use our own trust model against us, by adopting deception strategies as seen in [30] [26].

Chapter 2

Trust in complex actions

2.1 Motivation: trust in service compositions

In this chapter, we present our first trust model. We argue for the necessity of a general and formal theory of trust that can be adapted to a specific agent for his context of interaction. We focus on logical theories. As seen in the previous survey, such theories view trust as a particular belief of the truster involving ability and willingness of the trustee to perform some action for the truster. Most of such theories are about trust in an action of an *individual* trustee only. They therefore cannot account for complex interactions in multi-agent systems where an agent may e.g. ask *several* trustees to collaborate by composing their efforts, i.e., by composing their individual actions into complex actions, that might also be called *complex programs*. The ability to take into account such examples seems essential to trust based applications.

In this first model, we provide a general definition of trust in a complex action performed by multiple agents. We choose a formal, logical approach in terms of a simple, decidable logic of action, belief, and choice. The actions of our logic are assignments by an agent i of a propositional variable p to either true or false, respectively noted $i{:}{+}p$

and $i{:}{\rightarrow}p$ [35, 5]. Within the logical language we follow Castelfranchi and Falcone and define the concept of trust as a special kind of belief. We then show how to infer trust for different kinds of complex action operators. For example, we show that agent i's trust that the sequence of actions $\pi_1;\pi_2$ is going to take place cannot be reduced to the conjunction of (1) i's trust that π_1 is going to take place and (2) i's trust that π_2 is going to take place. Indeed, the effect of π_1 might be a precondition of π_2. Instead, i's trust in $\pi_1;\pi_2$ should reduce to (1) i's trust that π_1 is going to take place and (2) i's belief that *after* π_1, *i is going to trust that π_2 is going to take place.*

2.2 ABC: A logic of action, belief and choice

Our logic is a fairly standard multimodal logic of action, belief, and choice that we call ABC.

There are two belief modalities: strong belief, alias certainty, and weak belief, alias plausibility. We will argue that the former is appropriate to capture the truster's belief about the trustee's ability to cooperate, while the latter is appropriate to capture the truster's belief about the trustee's willingness to cooperate (despite his capability to defect). The logic of both certainty and plausibility is the normal modal logic K45; moreover, certainty implies plausibility. Following Cohen and Levesque [17], we suppose that choice is realistic: certainty implies choice (while plausibility does not).

Taking inspiration from boolean games [2] and dynamic epistemic logics [23], we consider that atomic actions are assignments of propositional variables to truth values, with a simple semantics in terms of model updates. Complex actions—alias programs—are then built by means of standard program constructions such as sequential composition. We consider two modalities for such complex actions, one for executability, and one for actual execution. The former enables us to capture C&F's external component of trust, which is about the environment allowing the execution of the trustee's action. The latter enables us to capture their internal component of trust, which is about the trustee's willingness to perform an action. It is therefore related to

the choice modality: an agent only executes an action if he has chosen it, and the other way round, cf.[33].

Belief and choice obey no forgetting and no learning principles that are familiar from dynamic epistemic logics and epistemic temporal logics, cf. [23].

2.2.1 Language

Throughout the chapter, $\mathbb{P} = \{p, q, \dots\}$ is a countably infinite set of *propositional variables* and $\mathbb{I} = \{i, j, k, \dots\}$ is a finite set of *agent names*.

In the epistemic dimension of the language, we have modal operators of certitude $\texttt{Cert}_i$, plausibility $\texttt{Plaus}_i$, and choice $\texttt{Choice}_i$, one per agent $i \in \mathbb{I}$. The formula $\texttt{Cert}_i\varphi$ reads "i strongly believes that φ", $\texttt{Plaus}_i\varphi$ reads "i weakly believes that φ", and $\texttt{Choice}_i\varphi$ reads "i chooses that φ", or "i prefers that φ".

The dynamic dimension of the language is based on *assignments*. $+p$ makes p true and $-p$ makes p false. An *authored assignment* is of the form either $i{:}+p$ or $i{:}-p$, where i is an agent from $\mathbb{I}$ and p is a variable from $\mathbb{P}$. The intuition for the former is that i sets the variable p to true; for the latter it is that i sets p to false. An *atomic action* is a finite set of authored assignments. The biggest set of atomic actions is

$$\Delta = \{i{:}+p : p \in \mathbb{P}, i \in \mathbb{I}\} \cup \{i{:}-p : p \in \mathbb{P}, i \in \mathbb{I}\}$$

Note that Δ could as well be defined as the set of subsets of $\mathbb{I} \times \{+, -\} \times \mathbb{P}$.

Given an atomic action $\delta \subseteq \Delta$ and an agent $i \in \mathbb{I}$, *i's part of δ* is

$$\delta|_i = \{i{:}+p : i{:}+p \in \delta\} \cup \{i{:}-p : i{:}-p \in \delta\}.$$

Let $p \in \mathbb{P}$ be a propositional variable. An atomic action $\delta \in \Delta$ is said to be *consistent in p* iff there are no $i, j \in \mathbb{I}$ such that $i{:}+p, j{:}-p \in \delta$. Otherwise δ is *inconsistent in p*. δ is *consistent* if δ is consistent in each of its variables p.

Beyond the modal operators $\texttt{Cert}_i$ and $\texttt{Choice}_i$, our language has two dynamic modal operators $\langle . \rangle$ and $\langle\!\langle . \rangle\!\rangle$. The first of these operators

is from Propositional Dynamic Logic PDL [32]. Both operators have complex actions as arguments. The set of formula $\langle\pi\rangle\varphi$ reads "the action π is executable and φ is true afterwards". In contrast, $\langle\!\langle\pi\rangle\!\rangle\varphi$ reads "π is executed and φ is true afterwards". The latter implies the former: execution implies executability. It is also clear that the other way round, executability should not imply execution. So we read $\langle i{:}{+}p\rangle\top$ as "i is able to make p true" and $\langle\!\langle i{:}{+}p\rangle\!\rangle\top$ as "i is making p true". The formula $\langle\!\langle i{:}{+}p, j{:}{-}q\rangle\!\rangle\varphi$ expresses that the agents i and j are going to assign the value 'true' to the propositional variable p and 'false' to q, and that afterwards φ will be true; and $\mathtt{Cert}_k\langle\!\langle i{:}{+}p, j{:}{-}q\rangle\!\rangle\varphi$ expresses that agent k believes that this is going to happen. As the reader may have noticed, we drop the set parentheses around the atomic assignments in formulas such as $\langle i{:}{+}p\rangle\top$, $\langle\!\langle i{:}{+}p\rangle\!\rangle\top$ and $\langle\!\langle i{:}{+}p, j{:}{-}q\rangle\!\rangle\varphi$.

Formally, the definition of the set of *actions* (programs) $\mathtt{Prog}$ and the set of *well-formed formulas* of ABC logic is by mutual induction as follows:

$$\begin{array}{lcl} \pi & := & \delta \mid \mathtt{skip} \mid \mathtt{fail} \mid \pi;\pi \mid \mathtt{if}\ \varphi\ \mathtt{then}\ \pi\ \mathtt{else}\ \pi \\ \varphi & := & p \mid \top \mid \neg\varphi \mid \varphi\wedge\varphi \mid \mathtt{Plaus}_i\varphi \mid \mathtt{Cert}_i\varphi \mid \mathtt{Choice}_i\varphi \mid \langle\pi\rangle\varphi \mid \langle\!\langle\pi\rangle\!\rangle\varphi \end{array}$$

where p ranges over $\mathbb{P}$, i over $\mathbb{I}$ and δ over Δ. The set of well-formed formulas is denoted by $\mathcal{L}_{\mathsf{ABC}}$.

Here is an example of a complex action. Let L mean that the light is on. Then the program $\mathtt{if}\ L\ \mathtt{then}\ j{:}{-}L\ \mathtt{else}\ j{:}{+}L$ describes j's action of toggling the light switch.

The formulas without the dynamic operators $\langle.\rangle$ and $\langle\!\langle.\rangle\!\rangle$ is noted $\mathcal{L}_{\mathsf{BC}}$.

For a given formula φ, the set $\mathbb{P}(\varphi)\subseteq\mathbb{P}$ is the set of propositional variables occurring in φ, and $\mathbb{I}(\varphi)\subseteq\mathbb{I}$ is the set of agent names occurring in φ. The sets $\mathbb{P}(\pi)$ and $\mathbb{I}(\pi)$ are defined likewise for programs π. For example, $\mathbb{P}(\langle\!\langle i{:}{+}p\rangle\!\rangle q) = \{p, q\}$ and $\mathbb{I}(\langle\!\langle i{:}{+}p\rangle\!\rangle q) = \{i\}$.

We use the standard abbreviations for $\vee$ and $\rightarrow$. Moreover, $\bot$ abbreviates $\neg\top$, $[\pi]\varphi$ abbreviates $\neg\langle\pi\rangle\neg\varphi$ and $[\![\pi]\!]\varphi$ abbreviates $\neg\langle\!\langle\pi\rangle\!\rangle\neg\varphi$.

2.2.2 Models

The models of our logic are Kripke models with accessibility relations for the belief and choice operators. Just as in dynamic epistemic logics, there is no accessibility relation for the action operators: instead, the semantics of actions is in terms of updates of models. There are moreover two functions which for every world determine the executable atomic assignments and the atomic assignment that is going to be executed after σ, for every sequence of atomic assignments σ. When σ is the empty sequence that function therefore determines the next atomic action that is going to take place.

An ABC model is a six-tuple $M = \langle W, A, BS, BW, C, T, V\rangle$ such that:

- W is a nonempty set of possible worlds or states;
- $A : W \longrightarrow 2^{\Delta}$ maps each world to the set of atomic actions that are physically executable there;
- $BS : \mathbb{I} \longrightarrow 2^{W\times W}$ maps each agent i to his accessibility relation for strong belief BS_i;
- $BW : \mathbb{I} \longrightarrow 2^{W\times W}$ maps each agent i to his accessibility relation for weak belief BW_i;
- $C : \mathbb{I} \longrightarrow 2^{W\times W}$ maps each agent i to his accessibility relation for choice C_i;
- $T : W \times \Delta^* \longrightarrow \Delta$ maps each world w and sequence of atomic actions σ to an action $T(w, \sigma)$ taking place after the sequence σ took place;
- $V : \mathbb{P} \longrightarrow 2^W$ is a valuation function associating to each propositional variable p the set of worlds $V(p)$ where p is true;

and such that the following constraints hold for every $i \in \mathbb{I}$:

1. $BW_i \subseteq BS_i$,
2. $C_i \subseteq BS_i$,

3. if $w' \in BS_i(w)$ then $BS_i(w) = BS_i(w')$, $BW_i(w) = BW_i(w')$, $C_i(w) = C_i(w')$, $A_i(w)|_i = A_i(w')|_i$, and $T(w,\sigma)|_i = T(w',\sigma)|_i$ for every sequence σ,

where $BS_i(w) = \{w' : \langle w, w'\rangle \in BS_i\}$, $BW_i(w) = \{w' : \langle w, w'\rangle \in BW_i\}$, and $C_i(w) = \{w' : \langle w, w'\rangle \in C_i\}$.

$BS_i(w)$ is the set of worlds that are compatible with agent i's strong beliefs at w, etc. The inclusion of the BW accessible worlds in the BS accessible worlds is a natural requirement. The inclusion of the C accessible worlds in the BS accessible worlds means that choice is realistic, in the sense of [17]: at a given world w, among the set $BS_i(w)$ of worlds that are possible for i, $C_i(w)$ is the set of those worlds that i prefers. The last constraint means that the agents introspect their strong and weak beliefs, choices and planned actions. The first requirement that $BS_i(w) = BS_i(w')$ for every $w' \in BS_i(w)$ is nothing but transitivity and euclideanity of the accessibility relation BS_i. The last requirement on the function T means that if after σ, agent i is going to perform his part $\delta|_i$ of δ, then i is going to perform exactly the same action at every world that is compatible with his beliefs. In particular, if i is going to perform $\delta|_i$ now (after the empty sequence of actions nil) then i also believes that he is going to perform $\delta|_i$ now. In other words, agents are aware of what they are going to do.

Beyond transitivity and euclideanity of BS_i it is often moreover assumed that every $BS_i(w)$ is non-empty, i.e., that the relation BS_i is serial. We do not suppose this here because seriality cannot be guaranteed under updates, as we will see below. This means that we allow that an agent may 'get crazy', in the sens that we allow for worlds w where $BS_i(w) = \emptyset$: at w, no possible world is compatible with i's beliefs. Note that unless the accessibility relation is reflexive, the same is the case in dynamic epistemic logics such as public announcement logic, cf. [23, 6].

Definition 6 (pointed model) *The set of* pointed models $\mathcal{PM}$ *is the set of tuples* (M, w_0)*, where* $M = \langle W, A, BS, BW, C, T, V\rangle$ *is an* ABC *model and* $w_0 \in W$ *is a world of* M*.*

2.2.3 Updating models

We now define two basic operations on a given model M: a valuation update (w.r.t. an atomic action δ) and restriction update of M to by subset W' of the set of possible worlds W ('relativization').

Definition 7 (update by an assignment and relativization) *Let $M = \langle W, A, BS, BW, C, T, V\rangle$ be a model. Let $\delta \subseteq \Delta$ be an atomic action. The* update of M by δ *is the model $M^\delta = \langle W, A, BS, BW, C, T^\delta, V^\delta\rangle$ where:*

$T^\delta(w, \sigma) = T(w, (\delta; \sigma))$, *for every $w \in W$ and $\sigma \in \Delta^*$;*

$$V^\delta(p) = \begin{cases} W & \text{if } \langle i, p^+\rangle \in \delta \text{ for some } i \in \mathbb{I} \text{ and } \delta \text{ is consistent in } p; \\ \emptyset & \text{if } \langle i, p^-\rangle \in \delta \text{ for some } i \in \mathbb{I} \text{ and } \delta \text{ is consistent in } p; \\ V(p) & \text{otherwise;} \end{cases}$$

for every $p \in \mathbb{P}$.

Let U be a nonempty subset of W. The relativization of M by U *is the model $M|_U = \langle W, A, BS', BW', C', T, V\rangle$ where for every $i \in \mathbb{I}$:*

$$\begin{aligned} BS'_i(w) &= BS_i(w) \cap U; \\ BW'_i(w) &= BW_i(w) \cap U; \\ C'_i(w) &= C_i(w) \cap U. \end{aligned}$$

Here are two simple examples. The update of M by the empty atomic assignment $\emptyset$ is not M, but $M^\emptyset = \langle W, A, BS, BW, C, T^\emptyset, V^\emptyset\rangle$ with $V^\emptyset = V$ and $T^\delta(w, \sigma) = T(w, (\delta; \sigma))$ for every world $w \in W$ and sequence $\sigma \in \Delta^*$. The relativization of M by W is $M|_W = M$.

The update of a model by an atomic action is clearly a model satisfying the above three constraints that we have imposed on models. The same is the case for its relativization by a subset U of its set of possible worlds W (in particular because U has to be nonempty). As announced in Section 2.2.2, the property of seriality is not preserved under relativization, which is why we did not require it for the belief accessibility relations BS_i.

Relativization and update can be safely permuted: we have $(M|_U)^\delta = (M^\delta)|_U$. We may therefore write $M|_U^\delta$ without harm.

2.2.4 Interpretation of formulas and programs

Formulas are interpreted as subsets of the sets of possible worlds, and programs are interpreted as functions on the set of pointed models. This is done by mutual recursion involving four ingredients.

To start with, the *satisfaction relation* $\models \subseteq \mathcal{PM} \times \mathcal{L}_{\mathsf{ABC}}$ is:

$$\begin{aligned}
M, w \models p &\text{ iff } w \in V(p)\\
M, w \models \top &\\
M, w \models \neg\varphi &\text{ iff } M, w \not\models \varphi\\
M, w \models \varphi \wedge \varphi' &\text{ iff } M, w \models \varphi \text{ and } M, w \models \varphi'\\
M, w \models \mathtt{Cert}_i\varphi &\text{ iff } M, w' \models \varphi \text{ for every } w' \in BS_i(w)\\
M, w \models \mathtt{Plaus}_i\varphi &\text{ iff } M, w' \models \varphi \text{ for every } w' \in BW_i(w)\\
M, w \models \mathtt{Choice}_i\varphi &\text{ iff } M, w' \models \varphi \text{ for every } w' \in C_i(w)\\
M, w \models \langle\pi\rangle\varphi &\text{ iff } \pi \in \mathit{Executable}(M, w) \text{ and } (M, w)^\pi \models \varphi\\
M, w) \models \langle\!\langle\pi\rangle\!\rangle\varphi &\text{ iff } \pi \in \mathit{Happens}(M, w) \text{ and } (M, w)^\pi \models \varphi
\end{aligned}$$

The last two truth conditions involve three functions to be defined now. The function $\mathit{Executable}(M, w)$ returns the set of programs that are executable at world w of M:

$$\begin{aligned}
\delta \in \mathit{Executable}(M, w) &\text{ iff } \delta \in A(w)\\
\mathtt{skip} \in \mathit{Executable}(M, w) &\\
\mathtt{fail} \notin \mathit{Executable}(M, w) &\\
\pi_1; \pi_2 \in \mathit{Executable}(M, w) &\text{ iff } \pi_1 \in \mathit{Executable}(M, w)\\
&\quad \text{and } \pi_2 \in \mathit{Executable}(M, w)\\
\mathtt{if}\ \varphi\ \mathtt{then}\ \pi_1\ \mathtt{else}\ \pi_2 \in \mathit{Executable}(M, w) &\text{ iff } M, w \models \varphi \text{ implies}\\
&\quad \pi_1 \in \mathit{Executable}(M, w)\\
&\quad \text{and } M, w \not\models \varphi \text{ implies}\\
&\quad \pi_2 \in \mathit{Executable}(M, w)
\end{aligned}$$

The function $\mathit{Happens}(M, w)$ returns the set of programs that are ex-

ecuted at w in M:

$$\begin{aligned}
\delta \in \mathit{Happens}(M,w) &\text{ iff } \delta \in A(w) \text{ and } \delta \in T(w,\mathsf{nil})\\
\mathtt{skip} \in \mathit{Happens}(M,w)&\\
\mathtt{fail} \notin \mathit{Happens}(M,w)&\\
\pi_1;\pi_2 \in \mathit{Happens}(M,w) &\text{ iff } \pi_1 \in \mathit{Happens}(M,w)\\
&\qquad \text{and } \pi_2 \in \mathit{Happens}\big((M,w)^{\pi}\big)\\
\mathtt{if}\ \varphi\ \mathtt{then}\ \pi_1\ \mathtt{else}\ \pi_2 \in \mathit{Happens}(M,w) &\text{ iff } M,w \models \varphi \text{ implies } \pi_1 \in \mathit{Happens}(M,w)\\
&\qquad \text{and } M,w \not\models \varphi \text{ implies}\\
&\qquad\qquad \pi_2 \in \mathit{Happens}(M,w)
\end{aligned}$$

It remains to define the update of a pointed model (M,w) by a complex program. The first and the third line make that this is a partial function.

$$\begin{aligned}
(M,w)^{\delta} &= (M|^{\delta}_{\{u\in W:\delta\in A(w) \text{ and } T(u,\mathsf{nil})=\delta\}}, w)\\
(M,w)^{\mathtt{skip}} &= (M,w)\\
(M,w)^{\mathtt{fail}} &= \mathsf{undef}\\
(M,w)^{\pi_1;\pi_2} &= ((M,w)^{\pi_1})^{\pi_2}\\
(M,w)^{\mathtt{if}\ \varphi\ \mathtt{then}\ \pi_1\ \mathtt{else}\ \pi_2} &= \begin{cases}(M,w)^{\pi_1} & \text{if } M,w\models\varphi\\ (M,w)^{\pi_2} & \text{otherwise}\end{cases}
\end{aligned}$$

For example, $(M,w)^{\mathtt{if}\ \top\ \mathtt{then}\ \pi\ \mathtt{else}\ \pi'} = (M,w)^{\pi}$. To see that $(M,w)^{\delta}$ may be undefined suppose $T(w,\mathsf{nil}) \neq \delta$: then the updated and relativized model $M|^{\delta}_{\{u\in W:\delta\in A(w) \text{ and } T(u,\mathsf{nil})=\delta\}}$ does not contain the world w, and therefore $(M|^{\delta}_{\{u\in W:\delta\in A(w) \text{ and } T(u,\mathsf{nil})=\delta\}}, w)$ is not a legal pointed model.

Remark 1 *An alternative to the above definition of update of pointed models would be* $(M,w)^{\delta} = (M|^{\delta}_{\{u\in W:\delta\in A(u)\}}, w)$. *However, then an agent would never learn about the other agents' intentions. Let us motivate this by an example. Suppose agent i knows that p is false, does not know that q, and knows that j can make p true but cannot*

make q false. Moreover, i knows that j wants $p \leftrightarrow q$ to hold: in i's $\neg p \wedge q$ world, j is going to make p true. This situation is expressed by the following formula:

$$\mathtt{Cert}_i \neg p \wedge \neg \mathtt{Cert}_i q \wedge \mathtt{Cert}_i(\langle j{:}{+}p\rangle \top \wedge \neg \langle j{:}{-}q\rangle \top) \wedge \mathtt{Cert}_i(q \leftrightarrow \langle\!\langle j{:}{+}p\rangle\!\rangle \top)$$

Suppose that i learns that j makes p false. As j does this intentionally—i.e., in agreement with his choices—, i may eliminate the $p \wedge \neg q$ world because j's action does not happen there. Then we expect $\langle\!\langle j{:}{+}p\rangle\!\rangle \mathtt{Cert}_i(p \wedge \neg q)$, i.e., after learning the mere occurrence of $j{:}{+}p$, i believes that $p \wedge q$. This is guaranteed by our way of updating pointed models. It is not guaranteed by the above alternative, which, informally speaking, 'keeps too many accessible worlds' in the set $BS_i(w)$.

It might be criticized that our definition too often result in agents having an empty set of accessible worlds $BS_i(w)$. The above definition can be refined in the following way: when at w agent i strongly believes that δ does not happen, i.e., when $M, w \models \mathtt{Cert}_i \neg \langle\!\langle \delta \rangle\!\rangle \top$, and when i learns that δ nevertheless took place then i should not move from the set of accessible worlds $BS_i(w)$ to $BS_i^\delta(w) = \emptyset$, but rather to the set $\{u \in BS_i(w) : \delta \in A(u)\}$. This however comes with some technical complications: just as in dynamic epistemic logics, we would have to create several copies of the worlds, and the updated set of possible worlds would have to be the product of W and the set of agents $\mathbb{I}$.

2.2.5 Logic ABC and logic BC

A $\mathcal{L}_{\mathsf{ABC}}$ formula φ is *valid in an* ABC *model* M if $M, w \models \varphi$ for every world w of M. We say that φ is a global consequence of ψ in ABC, noted $\psi \models_{\mathsf{ABC}} \varphi$, if φ is valid in every model M where ψ is valid. For example, the schema $\langle \pi \rangle \varphi \rightarrow [\pi]\varphi$ is valid: all our programs are deterministic. Moreover, $[\delta]\mathtt{Cert}_j\langle \delta \rangle \top)$ is ABC valid. Note that its generalisation $[\pi]\mathtt{Cert}_j\langle \pi \rangle \top)$ is invalid; to see this, replace π by `if` p `then` $i{:}{-}p$ `else fail` or by $\langle\!\langle \delta \rangle\!\rangle \top$. Another example of a ABC validity is $[i{:}{+}p, j{:}{-}q]\mathtt{Cert}_j(p \wedge \neg q)$.

The fragment of formulas without dynamic operators will be important in the sequel. For φ and ψ are in the fragment $\mathcal{L}_{\mathsf{BC}}$ of $\mathcal{L}_{\mathsf{ABC}}$, when

$\psi \models_{\mathsf{ABC}} \varphi$ then we also say that φ *is a logical consequence of* ψ *in* BC and write $\psi \models_{\mathsf{BC}} \varphi$.

Proposition 1 *The validities of* BC *logic are axiomatized by the principles of the basic modal logic* K *for each of the modal operators:* $\mathtt{Cert}_i$, $\mathtt{Plaus}_i$, $\mathtt{Choice}_i$, *plus the following schemas:*

$$\begin{aligned}
\mathtt{Cert}_i\varphi &\to \mathtt{Plaus}_i\varphi \\
\mathtt{Cert}_i\varphi &\to \mathtt{Choice}_i\varphi \\
\mathtt{Plaus}_i\varphi &\to \mathtt{Cert}_i\mathtt{Plaus}_i\varphi \\
\neg\mathtt{Plaus}_i\varphi &\to \mathtt{Cert}_i\neg\mathtt{Plaus}_i\varphi \\
\mathtt{Choice}_i\varphi &\to \mathtt{Cert}_i\mathtt{Choice}_i\varphi \\
\neg\mathtt{Choice}_i\varphi &\to \mathtt{Cert}_i\neg\mathtt{Choice}_i\varphi
\end{aligned}$$

For example, the following are theorems of our axiomatisation.

$$\begin{aligned}
\big(\mathtt{Cert}_i\varphi \wedge \mathtt{Plaus}_i(\varphi \to \psi)\big) &\to \mathtt{Plaus}_i\psi \\
\big(\mathtt{Cert}_i\varphi \wedge \mathtt{Choice}_i(\varphi \to \psi)\big) &\to \mathtt{Choice}_i\psi \\
\mathtt{Cert}_i\mathtt{Cert}_i\varphi &\leftrightarrow (\mathtt{Cert}_i\varphi \vee \mathtt{Cert}_i\mathtt{Cert}_i\bot) \\
\mathtt{Cert}_i\mathtt{Plaus}_i\varphi &\leftrightarrow (\mathtt{Plaus}_i\varphi \vee \mathtt{Cert}_i\mathtt{Plaus}_i\bot)
\end{aligned}$$

Global logical consequence in BC logic is decidable.

Proposition 2 *For* $\mathcal{L}_{\mathsf{BC}}$ *formulas* φ *and* ψ *it is decidable whether* $\psi \models_{\mathsf{BC}} \varphi$.

2.2.6 Decidability of ABC

In this section we prove that the ABC satifiability problem is decidable. The next propositions collects some valid equivalences by means of which we can almost eliminate the dynamic operators: we may rewrite formulas into a normal form in $\mathcal{L}_{\mathsf{BC}}$ with particular atoms.

Proposition 3 *The following equivalences are valid in* ABC *logic:*

$$\begin{aligned}
\langle\!\langle \pi \rangle\!\rangle \varphi &\leftrightarrow \langle\!\langle \pi \rangle\!\rangle \top \wedge \langle \pi \rangle \varphi \\
\langle\!\langle \mathtt{skip} \rangle\!\rangle \varphi &\leftrightarrow \varphi \\
\langle\!\langle \mathtt{fail} \rangle\!\rangle \varphi &\leftrightarrow \bot \\
\langle\!\langle \pi_1 ; \pi_2 \rangle\!\rangle \varphi &\leftrightarrow \langle\!\langle \pi_1 \rangle\!\rangle \langle\!\langle \pi_2 \rangle\!\rangle \varphi \\
\langle\!\langle \mathtt{if}\ \psi\ \mathtt{then}\ \pi_1\ \mathtt{else}\ \pi_2 \rangle\!\rangle \varphi &\leftrightarrow (\psi \to \langle\!\langle \pi_1 \rangle\!\rangle \varphi) \wedge (\neg\psi \to \langle\!\langle \pi_2 \rangle\!\rangle \varphi)
\end{aligned}$$

By the first equivalence we can 'almost' eliminate the operator $\langle\!\langle \pi \rangle\!\rangle$: we can obtain formulas where $\langle\!\langle \pi \rangle\!\rangle$ is always followed by $\top$. By the remaining equivalences we can obtain that π is atomic. So we can restrict our attention to formulas where all the occurrences of the happens-operator take the form $\langle\!\langle \delta \rangle\!\rangle \top$.

The next proposition parallels the last four equivalences for case of the executability operator.

Proposition 4 *The following equivalences are valid in* ABC *logic:*

$$\begin{aligned}
\langle \mathtt{skip} \rangle \varphi &\leftrightarrow \varphi \\
\langle \mathtt{fail} \rangle \varphi &\leftrightarrow \bot \\
\langle \pi_1 ; \pi_2 \rangle \varphi &\leftrightarrow \langle \pi_1 \rangle \langle \pi_2 \rangle \varphi \\
\langle \mathtt{if}\ \psi\ \mathtt{then}\ \pi_1\ \mathtt{else}\ \pi_2 \rangle \varphi &\leftrightarrow (\psi \to \langle \pi_1 \rangle \varphi) \wedge (\neg\psi \to \langle \pi_2 \rangle \varphi)
\end{aligned}$$

By the above equivalences we can decompose the executability operators $\langle \pi \rangle$ just as in star-free PDL: we can eliminate all the program operators from formulas.

Proposition 5 *The following equivalences are valid in* ABC *logic:*

$$
\begin{aligned}
\langle\delta\rangle\neg\varphi &\leftrightarrow \langle\delta\rangle\top \wedge \neg\langle\delta\rangle\varphi \\
\langle\delta\rangle(\varphi_1 \wedge \varphi_2) &\leftrightarrow \langle\delta\rangle\varphi_1 \wedge \langle\delta\rangle\varphi_2 \\
\langle\delta\rangle\mathtt{Cert}_i\varphi &\leftrightarrow \langle\delta\rangle\top \wedge \mathtt{Cert}_i[\delta]\varphi \\
\langle\delta\rangle\mathtt{Plaus}_i\varphi &\leftrightarrow \langle\delta\rangle\top \wedge \mathtt{Plaus}_i[\delta]\varphi \\
\langle\delta\rangle\mathtt{Choice}_i\varphi &\leftrightarrow \langle\delta\rangle\top \wedge \mathtt{Choice}_i[\delta]\varphi \\
\langle\delta\rangle p &\leftrightarrow \begin{cases} \top & \text{if } \delta \text{ is consistent and there is } i \in \mathbb{I} \text{ such that } i{:}{+}p \in \delta \\ \bot & \text{if } \delta \text{ is consistent and there is } i \in \mathbb{I} \text{ such that } i{:}{-}p \in \delta \\ p & \text{otherwise} \end{cases} \\
\langle\delta\rangle\langle\delta'\rangle\top &\leftrightarrow \langle\delta\rangle\top \wedge \langle\delta'\rangle\top
\end{aligned}
$$

The equivalences for the belief and goal operators correspond to principles of no forgetting and no learning.[1] These equivalences enable us to distribute the $\langle\delta\rangle$ over all the connectives but $\langle\!\langle\pi\rangle\!\rangle$ and $\top$ (where we can only eliminate the dynamic operator if δ is empty). The above propositions 4 and 5 therefore provide a set of reduction axioms for the 'executable' operator $\langle\pi\rangle$ that is complete for all the program operators and for all the formula operators except $\top$ and $\langle\!\langle\pi\rangle\!\rangle$.

Together, propositions 3, 4, and 5 enable us to rewrite any $\mathcal{L}_{\mathsf{ABC}}$ formula into an equivalent formula that is built by means of the boolean operators and the modal operators $\mathtt{Cert}_i$, $\mathtt{Plaus}_i$, and $\mathtt{Choice}_i$ from propositional variables $p \in \mathbb{P}$ and from what we call *dynamic atoms*: formulas of the form either $\langle\delta\rangle\top$, for $\delta \neq \emptyset$, or $\langle\delta_1\rangle\cdots\langle\delta_n\rangle\langle\!\langle\delta\rangle\!\rangle\top$. When $n = 0$ then we identify the latter dynamic atom with $\langle\!\langle\delta\rangle\!\rangle\top$.

It will be convenient to denote dynamic atoms by $\mu\langle\!\langle\delta\rangle\!\rangle\top$, where μ stands for the sequence $\langle\delta_1\rangle\cdots\langle\delta_n\rangle$. Then the language $\mathcal{L}^0_{\mathsf{ABC}}$ of formulas in normal form may be defined by the following BNF:

$$\varphi \;:=\; p \mid \langle\delta\rangle\top \mid \mu\langle\!\langle\delta\rangle\!\rangle\top \mid \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathtt{Cert}_i\varphi \mid \mathtt{Choice}_i\varphi$$

[1]Note that these principles can be generalised to sequences of atomic programs. They are however invalid for conditionals: for example, $\langle\mathtt{if}\ p\ \mathtt{then}\ i{:}{-}p\ \mathtt{else}\ i{:}{+}p\rangle\mathtt{Cert}_i p$ does not imply $\mathtt{Cert}_i[\mathtt{if}\ p\ \mathtt{then}\ i{:}{-}p\ \mathtt{else}\ i{:}{+}p]p$ (to see this, consider the case where p is false: then the first is true, while the second is not necessarily so).

where p ranges over the set of propositional variables $\mathbb{P}$, δ over the set of atomic actions Δ, and μ over the set of sequences $\langle\delta_1\rangle\cdots\langle\delta_n\rangle$. The p, $\langle\delta\rangle\top$, and $\mu\langle\langle\delta\rangle\rangle\top$ are called atomic formulas. If we identify the atomic formulas of $\mathcal{L}^0_{\mathsf{ABC}}$ with propositional variables then we obtain the language of BC.

Let $DA(\varphi)$ be the set of atomic formulas of φ. For example, ...

Let the reduction of the formula φ be $red(\varphi) \in \mathcal{L}^0_{\mathsf{ABC}}$.

Theorem 1 *The equivalence $\varphi \leftrightarrow red(\varphi)$ is ABC valid.*

A reduced formula $red(\varphi)$ can be viewed as a formula of the fragment $\mathcal{L}_{\mathsf{BC}}$ of $\mathcal{L}_{\mathsf{ABC}}$ that is built from the set of proposition variables $\mathbb{P}(\varphi) \cup DA(\varphi)$.

Proposition 6 *The following formulas are ABC valid:*

$$\begin{array}{ll} \langle\emptyset\rangle\top & \\ \langle\langle\delta\rangle\rangle\top \rightarrow \langle\delta\rangle\top & \\ \neg\big(\langle\langle\delta\rangle\rangle \wedge \langle\langle\delta'\rangle\rangle\big) & \textit{for } \delta \textit{ and } \delta' \textit{ different sets} \end{array}$$

The first implication is a consequence of the first item of Proposition 3.

Let us define the set of formulas Γ_φ axiomatizing the relation between the dynamic atoms occurring in φ:

$$\begin{array}{rl} \Gamma_\varphi = & \big\{\langle\emptyset\rangle\top\big\} \cup \\ & \big\{\mu\langle\langle\delta\rangle\rangle\top \rightarrow \langle\delta\rangle\top : \mu\langle\langle\delta\rangle\rangle\top \in DA(\varphi)\big\} \cup \\ & \big\{\neg\big(\mu\langle\langle\delta\rangle\rangle\top \wedge \mu\langle\langle\delta'\rangle\rangle\top\big) : \mu\langle\langle\delta\rangle\rangle\top, \mu\langle\langle\delta'\rangle\rangle\top \in DA(\varphi) \text{ and } \delta \neq \delta'\big\} \end{array}$$

The set Γ_φ is finite, and each of its elements is ABC valid. It provides a background theory under which we may consider dynamic atoms to be just atoms of $\mathcal{L}_{\mathsf{BC}}$.

Theorem 2 *Let $\varphi \in \mathcal{L}_{\mathsf{ABC}}$ and let $red(\varphi) \in \mathcal{L}^0_{\mathsf{ABC}}$ be its reduction. Then φ is ABC valid if and only if $\Gamma_\varphi \models_{\mathsf{BC}} red(\varphi)$, where in the latter consequence problem the atomic formulas of $DA(\varphi)$ occurring in Γ_φ and $red(\varphi)$ are viewed as arbitrary propositional variables.*

Proof 1 *The proof uses that if the sets δ and δ' are syntactically different then the implication $\langle\langle\delta\rangle\rangle\top \wedge \langle\langle\delta'\rangle\rangle\top \to \bot$ is valid in* ABC *logic. From this it follows that the elements of Γ_φ are valid in* ABC *logic.*

Corollary 3 *Validity of a $\mathcal{L}_{\mathsf{ABC}}$ formula is decidable.*

Remark 2 *Suppose not only the set of agents $\mathbb{I}$ is finite, but also the set of propositional variables $\mathbb{P}$. Then the set of atomic actions Δ is finite, too, and we can formulate principles of intentional action: every agent intentionally performs his part of the action δ that is going to take place. Formally:*

$$\langle\langle\delta\rangle\rangle\top \to \mathtt{Cert}_i \bigvee_{\{\delta'\in\Delta:\delta'|_i=\delta|_i\}} \langle\langle\delta'\rangle\rangle\top$$

$$\bigwedge_{\{\delta'\in\Delta:\delta'|_i=\delta|_i\}} \neg\langle\langle\delta'\rangle\rangle\top \to \mathtt{Cert}_i\neg\langle\langle\delta\rangle\rangle\top$$

So when δ occurs then each of the agents has the intention to perform his part of δ, and the other way round, this is a necessary condition of the occurrence of δ. However, we have supposed that $\mathbb{P}$ is countably infinite: the above two principles can therefore not be formulated.

2.3 A logical analysis of trust in ABC

The cognitive theory of Castelfranchi and Falcone, henceforth abbreviated C&F, is one of the most prominent [15, 16]. According to C&F, the trust relation involves a truster i, a trustee j, an action a that is performed by j and a goal φ of i. They defined the predicate `Trust` as a goal together with a particular configuration of beliefs of the trustee. Precisely, i trusts j to do a in order to achieve φ if and only if i has the goal that φ and i believes that:

1. j is willing to perform a,
2. j is capable to perform a,
3. j has the power to achieve φ by doing a.

C&F distinguish external from internal conditions in trust assessment: j's capability to perform a is an external condition, while j's willingness to perform a is an internal condition (being about the trustee's mental state). Finally, j's power to achieve φ by doing a relates internal and external conditions: if j performs a then φ will result. Observe that in the power condition, the result is conditioned by the execution of a; therefore the power condition is independent from the capability condition. In particular, j may well have the power to achieve φ without being capable to perform a: for example, right now I have the power to lift a weight of 50kg, but I am not capable to do this because there is no such weight at hand.

We follow Jones who argued that the core notion of trust need not involve a goal of the truster [38, 37] and consider a simplified version of C&F's definition in terms of a truster, a trustee, an action of the trustee, and an expected outcome of that action.

C&F did not investigate further how goals, capabilities, willingness and power have to be defined. A formal analysis of these notions was undertaken by Herzig et al. [34]. Both C&F and Herzig et al. only considered trust in the atomic action of another agent and did not consider trust in complex actions such as our π. In the sequel we adapt and extend their approach in order to account for complex actions. Our analysis differs in one important respect: while theirs is in terms of a single modal operator of belief, we are going to be more fine-grained and involve two kinds of belief operators: the strong belief operators $\texttt{Cert}_i$ and the weak belief operators $\texttt{Plaus}_i$. That j has the capability to perform a and has the power to achieve φ by doing a are strong beliefs of the trustee, while that j is willing to perform a is a weak belief. To motivate this consider a prisoners dilemma situation where agent 1 trusts agent 2 to cooperate: agent 1 knows that agent 2 may defect, i.e., there is a world that is possible for 1 where 2 is going to defect. However, in the most plausible worlds among the worlds that are possible for 1 agent 2 is going to cooperate.

2.3.1 Reducing trust

Complex action expressions involve multiple agents that occur in the action expressions. We therefore need not identify the trustee as a separate argument of the trust predicate. Our official definition of the trust predicate then becomes:

$$\mathtt{Trust}(i,\pi,\varphi) \stackrel{\text{def}}{=} \mathtt{Plaus}_i\mathtt{CInt}(\pi) \wedge \mathtt{Cert}_i\big(\mathtt{CExt}(\pi) \wedge \mathtt{Cert}_i\mathtt{Res}(\pi,\varphi)\big)$$

where $\mathtt{Plaus}_i$ and $\mathtt{Cert}_i$ are the two modal operators of belief of ABC and where $\mathtt{CInt}(\pi)$, $\mathtt{CExt}(\pi)$, and $\mathtt{Res}(\pi,\varphi)$ respectively correspond to items 1, 2 and 3 in C&F's definition. `CInt` and `CExt` stand for the internal and the external condition in trust assessment.

In the rest of the section we define the predicates $\mathtt{CInt}(\pi)$, $\mathtt{CExt}(\pi)$, and $\mathtt{Res}(\pi,\varphi)$ in ABC logic and study how trust in a complex action can be built from trust in its constituents.

The modal operators $\mathtt{Plaus}_i$ and $\mathtt{Cert}_i$ are already primitives of ABC logic. It remains to define the other components of trust in ABC logic:

$$\begin{aligned}
\mathtt{CExt}(\pi) &\stackrel{\text{def}}{=} \langle\pi\rangle\top \\
\mathtt{CInt}(\pi) &\stackrel{\text{def}}{=} \langle\pi\rangle\top \rightarrow \langle\!\langle\pi\rangle\!\rangle\top \\
\mathtt{Res}(\pi,\varphi) &\stackrel{\text{def}}{=} [\pi]\varphi
\end{aligned}$$

The definition of the internal condition says that if π is executable then π is going to happen. This is actually a bit weaker than C&F's willingness condition. To see this, consider the case where π is an atomic action a of some agent j, written $j{:}a$. If j cannot perform a, i.e., when the external condition fails to hold, then the internal condition is trivially true. There is however no harm here: as $\mathtt{CExt}(j{:}a)$ is false, the trust predicate will be false anyway in that case. In the case where the external condition holds the internal condition reduces to truth of $\langle\!\langle j{:}a\rangle\!\rangle\top$, and as we have seen in Section 2.2.2, when $\langle\!\langle j{:}a\rangle\!\rangle\top$ is true at a possible world w then j performs $j{:}a$ at every world that is chosen by j at w, i.e., j indeed has the intention to perform a at w.

Given the above definitions we obtain:

$$\begin{aligned}\mathtt{Trust}(i,\pi,\varphi) &= \mathtt{Plaus}_i\langle\!\langle\pi\rangle\!\rangle\top \wedge \mathtt{Cert}_i\big(\langle\!\langle\pi\rangle\!\rangle\top \rightarrow \langle\pi\rangle\top\big) \wedge [\pi]\varphi\big)\\ &\leftrightarrow \mathtt{Plaus}_i\langle\!\langle\pi\rangle\!\rangle\top \wedge \mathtt{Cert}_i\big(\langle\pi\rangle\top \wedge [\pi]\varphi\big)\end{aligned}$$

We take this as our official definition of trust in a complex action. In words, i's trust that the complex action π is going to be performed and produces φ reduces to a weak belief of i that π is going to occur and a strong belief of i that π is executable and that φ is among the effects of π.

2.3.2 Trust in complex actions

The following validities allow to build trust in a complex action from trust in its constituents. Their proof makes use of the following ABC theorems:

$$\begin{aligned}[\delta]\mathtt{Cert}_i\psi &\leftrightarrow [\delta]\bot \vee \mathtt{Cert}_i[\delta]\psi\\ [\delta]\mathtt{Plaus}_i\psi &\leftrightarrow [\delta]\bot \vee \mathtt{Plaus}_i[\delta]\psi\\ \mathtt{Cert}_i[\delta]\psi &\rightarrow [\delta]\mathtt{Cert}_i\psi\\ \mathtt{Plaus}_i[\delta]\psi &\rightarrow [\delta]\mathtt{Plaus}_i\psi\\ \langle\delta\rangle\top \wedge [\delta]\varphi &\leftrightarrow \langle\delta\rangle\varphi\end{aligned}$$

which follow from the equivalences in Proposition 5 (precisely, the reduction axioms for `Cert` and `Plaus` and the determinism of δ).

Theorem 4 *The following equivalences are* ABC *valid:*

$$\begin{aligned}\mathtt{Trust}(i,\mathtt{fail},\varphi) &\leftrightarrow \bot\\ \mathtt{Trust}(i,\mathtt{skip},\varphi) &\leftrightarrow \mathtt{Cert}_i\varphi\end{aligned}$$

For sequences and conditionals we only have implications.

Theorem 5 *The following implications are* ABC *valid:*

$$\begin{aligned}\mathtt{Trust}(i,(\pi_1;\pi_2),\varphi) &\rightarrow \mathtt{Trust}(i,\pi_1,\top)\\ \mathtt{Trust}(i,(\delta;\pi),\varphi) &\rightarrow \mathtt{Cert}_i[\delta]\mathtt{Trust}(i,\pi,\varphi)\\ \neg\mathtt{Cert}_i\mathtt{Plaus}_i\bot \wedge \mathtt{Trust}(i,\delta,\top) \wedge \mathtt{Cert}_i[\delta]\mathtt{Trust}(i,\pi,\varphi) &\rightarrow \mathtt{Trust}(i,(\delta;\pi),\varphi)\end{aligned}$$

Proof 2 *The first implication is straightforward. For the second implication we have:*

$$
\begin{aligned}
\mathtt{Trust}(i,(\delta;\pi),\varphi) &\leftrightarrow \mathtt{Plaus}_i\langle\!\langle\delta\rangle\!\rangle\langle\!\langle\pi\rangle\!\rangle\top \wedge \mathtt{Cert}_i\langle\delta\rangle\langle\pi\rangle\top \wedge \mathtt{Cert}_i[\delta][\pi]\varphi \\
&\qquad\qquad \textit{(because } \mathtt{Cert} \textit{ and } [\delta] \textit{ are normal)} \\
&\rightarrow \mathtt{Plaus}_i[\delta]\langle\!\langle\pi\rangle\!\rangle\top \wedge \mathtt{Cert}_i[\delta]\big(\langle\pi\rangle\top \wedge [\pi]\varphi\big) \\
&\qquad\qquad \textit{(because programs are deterministic)} \\
&\rightarrow \mathtt{Cert}_i\mathtt{Plaus}_i[\delta]\langle\!\langle\pi\rangle\!\rangle\top \wedge \mathtt{Cert}_i\mathtt{Cert}_i[\delta]\big(\langle\pi\rangle\top \wedge [\pi]\varphi\big) \\
&\qquad\qquad \textit{(by Proposition 1)} \\
&\rightarrow \mathtt{Cert}_i[\delta]\mathtt{Plaus}_i\langle\!\langle\pi\rangle\!\rangle\top \wedge \mathtt{Cert}_i[\delta]\mathtt{Cert}_i\big(\langle\pi\rangle\top \wedge [\pi]\varphi\big) \\
&\qquad\qquad \textit{(by Proposition 5)} \\
&\leftrightarrow \mathtt{Cert}_i[\delta]\mathtt{Trust}(i,\pi,\varphi)
\end{aligned}
$$

For the third implication:

$$
\begin{aligned}
& \neg\mathtt{Cert}_i\mathtt{Plaus}_i\bot \wedge \mathtt{Trust}(i,\delta,\top) \wedge \mathtt{Cert}_i[\delta]\mathtt{Trust}(i,\pi,\varphi) \\
\leftrightarrow\ & \neg\mathtt{Cert}_i\mathtt{Plaus}_i\bot \wedge \Big(\mathtt{Plaus}_i\langle\!\langle\delta\rangle\!\rangle\top \wedge \mathtt{Cert}_i\big(\langle\delta\rangle\top \wedge [\delta]\top\big)\Big) \wedge \\
& \Big(\mathtt{Cert}_i[\delta]\mathtt{Plaus}_i\langle\!\langle\pi\rangle\!\rangle\top \wedge \mathtt{Cert}_i[\delta]\mathtt{Cert}_i\big(\langle\pi\rangle\top \wedge [\pi]\varphi\big)\Big) \\
& \qquad\qquad \textit{(because } \mathtt{Cert} \textit{ and } [\delta] \textit{ are normal)} \\
\rightarrow\ & \neg\mathtt{Cert}_i\mathtt{Plaus}_i\bot \wedge \mathtt{Plaus}_i\langle\!\langle\delta\rangle\!\rangle\top \wedge \mathtt{Cert}_i\langle\delta\rangle\top \wedge \\
& \mathtt{Cert}_i\big([\delta]\bot \vee \mathtt{Plaus}_i[\delta]\langle\!\langle\pi\rangle\!\rangle\top\big) \wedge \mathtt{Cert}_i\Big([\delta]\bot \vee \mathtt{Cert}_i[\delta]\big(\langle\pi\rangle\top \wedge [\pi]\varphi\big)\Big) \\
& \qquad\qquad \textit{(by Proposition 5)} \\
\rightarrow\ & \neg\mathtt{Cert}_i\mathtt{Plaus}_i\bot \wedge \mathtt{Plaus}_i\langle\!\langle\delta\rangle\!\rangle\top \wedge \mathtt{Cert}_i\langle\delta\rangle\top \wedge \\
& \mathtt{Cert}_i\mathtt{Plaus}_i[\delta]\langle\!\langle\pi\rangle\!\rangle\top \wedge \mathtt{Cert}_i\mathtt{Cert}_i[\delta]\big(\langle\pi\rangle\top \wedge [\pi]\varphi\big) \\
& \qquad\qquad \textit{(because } \mathtt{Cert} \textit{ and } [\delta] \textit{ are normal)} \\
\leftrightarrow\ & \neg\mathtt{Cert}_i\mathtt{Plaus}_i\bot \ \wedge \mathtt{Plaus}_i\langle\!\langle\delta\rangle\!\rangle\top \wedge \mathtt{Cert}_i\langle\delta\rangle\top \wedge \\
& \Big(\mathtt{Cert}_i\mathtt{Plaus}_i\bot \vee \mathtt{Plaus}_i[\delta]\langle\!\langle\pi\rangle\!\rangle\top\Big) \wedge \Big(\mathtt{Cert}_i\mathtt{Cert}_i\bot \vee \mathtt{Cert}_i[\delta]\big(\langle\pi\rangle\top \wedge [\pi]\varphi\big)\Big) \\
& \qquad\qquad \textit{(by Proposition 1)} \\
\rightarrow\ & \mathtt{Plaus}_i\langle\!\langle\delta\rangle\!\rangle\top \wedge \mathtt{Cert}_i\langle\delta\rangle\top \wedge \mathtt{Plaus}_i[\delta]\langle\!\langle\pi\rangle\!\rangle\top \wedge \mathtt{Cert}_i\big([\delta]\langle\pi\rangle\top \wedge [\delta][\pi]\varphi\big) \\
& \qquad\qquad \textit{(because } \mathtt{Cert} \textit{ and } [\delta] \textit{ are normal)} \\
\rightarrow\ & \mathtt{Plaus}_i\langle\!\langle\delta\rangle\!\rangle\langle\!\langle\pi\rangle\!\rangle\top \wedge \mathtt{Cert}_i\big(\langle\delta\rangle\langle\pi\rangle\top \wedge [\delta][\pi]\varphi\big) \\
& \qquad\qquad \textit{(by Proposition 1)} \\
\leftrightarrow\ & \mathtt{Trust}(i,(\delta;\pi),\varphi)
\end{aligned}
$$

Note that the last two equivalences cannot be generalised from atomic δ to conditionals.

Theorem 6 *The following implications are* ABC *valid:*

$$
\begin{aligned}
& \mathtt{Trust}(i,\mathtt{if}\ \psi\ \mathtt{then}\ \pi_1\ \mathtt{else}\ \pi_2,\varphi) \rightarrow \\
& \big(\mathtt{Cert}_i\psi \rightarrow \mathtt{Trust}(i,\pi_1,\varphi)\big) \wedge \big(\mathtt{Cert}_i\neg\psi \rightarrow \mathtt{Trust}(i,\pi_2,\varphi)\big)
\end{aligned}
$$

Proof 3 *It suffices to prove*

$$
\big(\mathtt{Trust}(i,\mathtt{if}\ \psi\ \mathtt{then}\ \pi_1\ \mathtt{else}\ \pi_2,\varphi)\big) \wedge \mathtt{Cert}_i\psi\big) \rightarrow \mathtt{Trust}(i,\pi_1,\varphi).
$$

We prove that under $\texttt{Cert}_i\psi$*, the components of* $\texttt{Trust}\big(i, \texttt{if } \psi \texttt{ then } \pi_1 \texttt{ else } \pi_2, \varphi\big)$ *imply the components of* $\texttt{Trust}(i, \pi_1, \varphi)$*: first,* $\texttt{Plaus}_i\langle\!\langle \texttt{if } \psi \texttt{ then } \pi_1 \texttt{ else } \pi_2 \rangle\!\rangle \top \wedge \texttt{Cert}_i\psi \rightarrow \texttt{Plaus}_i\langle\!\langle \pi_1 \rangle\!\rangle \top$*; second,* $\texttt{Cert}_i\langle \texttt{if } \psi \texttt{ then } \pi_1 \texttt{ else } \pi_2\rangle \top \wedge \texttt{Cert}_i\psi \rightarrow \texttt{Cert}_i\langle\pi_1\rangle\top$*; third,* $\texttt{Cert}_i[\texttt{if } \psi \texttt{ then } \pi_1 \texttt{ else } \pi_2]\top \wedge \texttt{Cert}_i\psi \rightarrow \texttt{Cert}_i[\pi_1]\top$*.*

Let us consider a stronger version of theorem 6 where the consequent is replaced by

$$\texttt{Cert}_i\Big(\big(\psi \rightarrow \texttt{Trust}(i, \pi_1, \varphi)\big) \wedge \big(\neg\psi \rightarrow \texttt{Trust}(i, \pi_2, \varphi)\big)\Big)$$

However, together with $\neg\texttt{Cert}_i\psi \wedge \neg\texttt{Cert}_i\neg\psi$. they imply $\neg\texttt{Cert}_i\neg\texttt{Trust}(i, \pi_1, \varphi) \wedge \neg\texttt{Cert}_i\neg\texttt{Trust}(i, \pi_2, \varphi)$, i.e.,

$$\begin{aligned}&\neg\texttt{Cert}_i\neg\Big(\texttt{Plaus}_i\langle\!\langle \pi_1 \rangle\!\rangle \top \wedge \texttt{Cert}_i(\langle\pi_1\rangle \wedge [\pi_1]\varphi)\Big) \wedge \\ &\neg\texttt{Cert}_i\neg\Big(\texttt{Plaus}_i\langle\!\langle \pi_2 \rangle\!\rangle \top \wedge \texttt{Cert}_i(\langle\pi_2\rangle \wedge [\pi_2]\varphi)\Big)\Big),\end{aligned}$$

which by the introspection principles is equivalent to

$$\Big(\texttt{Plaus}_i\langle\!\langle \pi_1 \rangle\!\rangle \top \wedge \texttt{Cert}_i(\langle\pi_1\rangle \wedge [\pi_1]\varphi)\Big) \wedge \Big(\texttt{Plaus}_i\langle\!\langle \pi_2 \rangle\!\rangle \top \wedge \texttt{Cert}_i(\langle\pi_2\rangle \wedge [\pi_2]\varphi)\Big)\Big),$$

i.e., to $\texttt{Trust}(i, \pi_1, \varphi) \wedge \texttt{Trust}(i, \pi_2, \varphi)$. Therefore, in the case where i is in doubt about the condition of the if-then-else then i has to trust both actions π_1 and π_2 to occur and to achieve φ. This means that π_1 and π_2 are equivalent. Indeed, for example for, if they are both atomic they have to be syntactically identical according to Proposition 6.

2.3.3 Reasoning tasks involving trust

There are at least 3 different reasoning tasks involving the trust predicate.

- Trust validity problem: the problem of deciding validity of formulas containing the trust predicate allows us to analyze the general properties of an execution in some system. This can be related to trust concepts in epistemic games.

- Model checking: decide whether in a current system, trust can be asserted for some actions, and for which reason.
- Abduction: by analyzing the conditions under which the goal is not satisfied, we can define a set of formulas that can be seen as explanations for failure of trust.

Such reasoning tasks come up naturally in various domains and contexts, requiring many variations of the concept of trust. Our logic provides a common trust definition that can be used in different trust-based decision procedures. The instantiation depends on the definition of the reasoning procedure to be used.

Validity problem

When reasoning about trust, checking the validity of a formula allows us to derive general properties from some system specification. When a specific implementation of a system satisfies the specification and the specification implies some property then the implementation is guaranteed to have that property. This appears to be very similar to the game theoretical approaches towards trust.

To illustrate this, we will use an instance of the prisoners dilemma, with two agents *i,j*. Each agent respectively controls the variables $coop_i$ and $coop_j$, where the simple action $\{i : coop_i^+, j : coop_j^+\}$ expresses that the two agent are cooperating. The set of all possible simple actions corresponds to pure strategy profiles.

The uncertainty of i about the actions of j when i decides to cooperate corresponds to the formula

$$\texttt{Cert}_i(\langle\!\langle i : coop_i^+, j : coop_j^+ \rangle\!\rangle \top \vee \langle\!\langle i : coop_i^+, j : coop_j^- \rangle\!\rangle \top).$$

The payoff of i can be associated to the following constrains on his choices: $\texttt{Cert}_i(\langle\!\langle i : coop_i^+, j : coop_j^+ \rangle\!\rangle \top \vee \langle\!\langle i : coop_i^+, j : coop_j^- \rangle\!\rangle \top) \rightarrow \texttt{Choice}_i\langle\!\langle \{i : coop_i^+, j : coop_j^+\} \rangle\!\rangle \top$ and $\texttt{Cert}_i(\langle\!\langle i : coop_i^-, j : coop_j^+ \rangle\!\rangle \top \vee \langle\!\langle i : coop_i^-, j : coop_j^- \rangle\!\rangle \top) \rightarrow \texttt{Choice}_i\langle\!\langle \{i : coop_i^-, j : coop_j^+\} \rangle\!\rangle \top$, meaning that if i chooses to cooperate, he would prefer that the payoff rather comes from the cooperation of j since it maximises his utility, and if i

decides to defect then he would prefer that j cooperates, for the same reason.

Our logic seems to offer the tools that we need in order to analyze the epistemic state of the agents i and j that may lead them to adopt some strategy profile, i.e., if Γ is a formula that describes the current epistemic state of our agents then $\models \Gamma \rightarrow \delta$. This means that γ supports the strategy profile δ, where to match our past example,

$$\delta \in \{\{i : coop_i^{\bullet}, j : coop_j^{\star}\} \mid \bullet, \star \in \{+, -\}\}.$$

Example 1 *Cooperation ($\delta_c = i : coop_i^{+}, j : coop_j^{+}$) can be ensured by the following epistemic state:*

$$\Gamma = \mathtt{Trust}(i, \delta_c) \wedge \mathtt{Trust}(j, \delta_c).$$

The formula Γ states that each agent believes that if he is willing to cooperate then the other agent will do the same.

Example 2 *Using the same idea, the Nash equilibrium*

$$\delta_n = \{i : coop_i^{-}, j : coop_j^{-}\}$$

can be explained by the distrust of each agent toward a cooperative behavior from the other agent. This corresponds to the following formula:

$$\Gamma = \bigwedge_i \mathtt{Cert}_i(\bigvee_{\bullet \in \{+,-\}} \langle\!\langle \{i : coop_i^{+}, j : coop_j^{\bullet}\} \rangle\!\rangle \top \rightarrow \langle\!\langle \{i : coop_i^{+}, j : coop_j^{-}\} \rangle\!\rangle \top) \wedge \mathtt{Choice}_i \langle\!\langle \delta_c \rangle\!\rangle \top.$$

The following example illustrates how complex actions can correspond to strategy profiles in repeated games.

Example 3 *The agent i decides to adopt reciprocity (cooperate if j cooperated last time or else defect), this strategy will correspond to the following formula:*

$$\begin{aligned} \langle\!\langle i : coop_i^{+}, j : coop_j^{+} \rangle\!\rangle \; (\; \langle\!\langle i : coop_i^{+}, j : coop_j^{+} \rangle\!\rangle \top \vee \\ \langle\!\langle i : coop_i^{+}, j : coop_j^{-} \rangle\!\rangle \top \;) \quad \wedge \\ \langle\!\langle i : coop_i^{+}, j : coop_j^{-} \rangle\!\rangle \; (\; \langle\!\langle i : coop_i^{-}, j : coop_j^{+} \rangle\!\rangle \top \vee \\ \langle\!\langle i : coop_i^{-}, j : coop_j^{-} \rangle\!\rangle \top \;) \end{aligned}$$

Model checking problem

The Model checking problem corresponds to the main reasoning procedure for our logic. Suppose given a model $pm \in PM$ that corresponds to the current state in which an epistemic multi-agent system is (see example in our last section). We can verify if:

- An agent i believes that a goal of his can be achieved by executing a complex action π, by checking if $pm \models \texttt{Trust}(i, \pi)$.
- A check whether $pm \models \neg\texttt{Trust}(i, \pi)$ will verify that i does not trust that the action would bring about φ (also given what he knows).

In the second case where $pm \models \neg\texttt{Trust}(i, \pi)$, we may want to check furthermore why trust predicate does not hold:

- $pm \models \texttt{Cert}_i((\texttt{CExt}(\pi) \wedge \texttt{CInt}(\pi)) \rightarrow \texttt{Res}(\pi, \varphi))$ means that I believe the action would not realise the goal;
- $pm \models \texttt{Cert}_i(\neg\texttt{CExt}(\pi) \vee \neg\texttt{CInt}(\pi)))$: means that the action will not occur.

Abductive reasoning

Using the model checking problem and for the sake of analyzing the same system as for the model checking procedure, we can define an abduction procedure, that tries to explain the truth value of the trust predicate.

Definition 8 *Suppose given an agent i and a complex action π. Suppose given two finite sets of formulas, respectively called PE and NE. Positive and negative explanations have to respect the following constrains:*

- *The formulas of PE are mutually exclusive and $\models \varphi' \rightarrow \varphi$ for all $\varphi' \in PE$.*
- *The formulas of NE are mutually excursive and $\models \varphi' \rightarrow \neg\varphi$ for all $\varphi' \in PE$*

Given a model $pm \in PM$, an explanation *of a positive evaluation of the trust of i in π to achieve φ (resp. negative evaluation of trust) is a formula φ' of PE (resp. of NP) such that $pm \models \texttt{Trust}(i, \pi, \varphi) \wedge \langle\langle\pi\rangle\rangle\varphi'$.*

The abductive procedure that we define can be seen as a specific analysis of the primary predicate $Res(\varphi)$ since we are only interested in the relation of the action π with the realisation (or non-realisation) of the goal.

This procedure allows us to help a user to choose between different actions (to be executed), by providing explanations for the result of the model checking procedure.

2.4 Case study: searching for accommodation

Our Goal in this section is to show how our logic can be used to analyze trust assertions, in a SOA context.

To do so, we will use a case study related to the process of searching accommodation that a student would perform using the services that the French student union CROUS offers to him.

In this case study:

- The student is identified as a service consumer that needs to reach a goal (in our case, accessing a list of accommodations potentially of interest to him), and this, by choosing between different processes that help him to reach his goal.
- The CROUS services are identified as a set of composed services that provide the essential information using different compositions. We will define 2 service compositions.

We remark that our logic is used in the process of analyzing the opportunities of interaction and not during the actual interaction. Our model allows the student to:

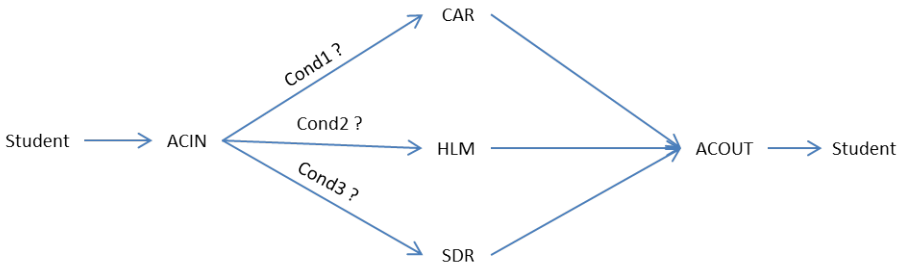

Figure 2.1: composed service : AccIntResearch

- Represent the execution of a composition workflow and the behavior of its different services by using the student knowledge about the modeled system;
- Assert if the student can trust the service composition to achieve his goal and in the same time, fulfilling the requirements that the student would want to see verified (mostly related to security and privacy issues);
- Extract some information about the different properties of interaction (arguments for or against the interaction, recommendations to achieve the goal, etc).

2.4.1 CROUS services presentation

We start by sketching an informal description of the different elements that we use in our case study, followed by a global schema of execution of the study cases; then we show how our logic can be used to provide an analysis.

We start in this section by describing two composed services that the CROUS allows students to use for searching accommodation.

- The first one is the AccIntResearch service depicted in Figure 2.1. This service returns the list of accommodations that the internal services of the CROUS propose. The execution of the composed

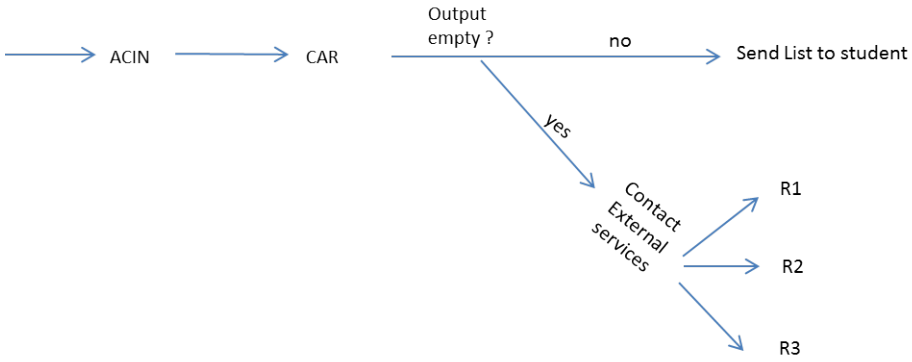

Figure 2.2: composed service : AccExtResearch

service is defined as follows:

1. The first service executed is AcomCrousIN (ACIN) which is a service that takes the information and preferences of the student and defines a set of public sources to contact (in our case, the HLM services, the CROUS Accommodation repository (CAR), schools dormitory services (SDR)).
2. A test is then run for each one of the services that have been identified, if the condition for admission of the student is fulfilled, contact the appropriate service.
3. Each service will then answer to an aggregation service called AccomCrous2 that will sum the results (available accomodations, application files and contact adresses) and sends them back to the student.

- The second one is the AccExtResearch service. This service includes external sources (private renters or estate agencies) into the research process that follows the schema presented in Figure 2.2 :

  The execution of the composed service is defined as follows:

  1. The first service executed is AccomCrous1 which takes the personal information and preferences of the student. This time,

the service will trigger a new searching process (due to the precise desire of the student to allow external sources to be interrogated expressed in his preferences).

2. The AccomCrous1 sends the student information (including email and phone number) to a CROUS repository services, that searches for the accommodations that satisfies the criteria specified by the student.
3. A conditional path is then triggered: if the list of accommodations retrieved by the repository is nonempty, the results are sent to the student (by email), otherwise, the personal information and the request of the student are sent to a list of external renters and agencies (private companies). The condition will be described by the atomic proposition *OutputEmpty*.
4. Each private service R_i will have the choice to either ignore the demand of the student, or to contact him by email or by phone (using the information provided by the CROUS to R_i).

2.4.2 Beliefs and goals of the student

The student (noted s) is considered as an agent in the interaction. His goal is to find a list of accommodations while making sure that his phone number is not provided to a private company (in order to avoid ads for example, or unwanted calls) during the composition. To model the belief of the student, we suppose that he knows the description of the two composed services AccIntResearch and AccExtResearch, which are made public before interaction (either via a public procedure description or proposed by the composition engine).

Thus the student believes that the services involved in the two compositions will behave as stated stated in Section 2.4.1 once he begins the interaction. This means he believes that upon his request the workflow of the composition will be respected. As a consequence his beliefs about the AccExtResearch service depend on his belief about the emptiness of the list of accommodations retrieved by the CROUS services. There are 3 cases:

1. He believes that the list is nonempty. Either he has performed a preliminary search or he got this information via some other way. In this case, his belief is expressed by the logical formula:

$$\texttt{Cert}_s\neg\text{OutputEmpty}$$

2. He believes that the list is empty. In this case, his belief is expressed by the logical formula :

$$\texttt{Cert}_s\text{OutputEmpty}$$

3. He has no belief about the emptiness of the list, which is expressed by the logical formula :

$$(\neg\texttt{Cert}_s\neg\text{OutputEmpty}) \wedge (\neg\texttt{Cert}_s\text{OutputEmpty})$$

The goal of the student is tied with his preferences and the action he requested.

Concerning his preferences, we suppose that the student agrees to share his email and phone number with any public institution but only agrees to share his email with private companies. As a consequence we may infer that one of his goals is to prevent private companies to access his phone number. We use the atomic proposition *PhonePrivate* to represent that a private company is given access to his phone number.

Concerning his request, we assume that the goal of the student is to obtain a list of accommodations. We use the atomic proposition *Accommodation* to represent that the student obtained this list.

With these notation, the goal of the student is represented by the logical formula :

$$\texttt{Goal}(s, \text{Accommodation} \wedge \neg\text{PhonePrivate}) \stackrel{\text{def}}{=} \texttt{Choice}_s\texttt{F} \quad (\text{Accommodation} \wedge\neg\text{PhonePrivate})$$

2.4.3 Trust analysis

The analysis process is the process of satisfiability checking for the trust formula associated to the composed action and goal of the student. If

the trust formula is satisfiable given the modeled beliefs and goals, then the trust is established. In the other case, an in depth analysis of the failure of satisfiability may give the reasons why the trust cannot be established.

AccIntResearch service.

In the case of the AccIntResearch service, the complex action is expressed with the atomic actions in the following way:

AccIntResearch = ACIN; (Cond1?; CAR+Cond2?; HLM+Cond3?; SDR); ACOUT

The trust formula is Trust(s, AccIntResearch, Accommodation$\wedge\neg$PhonePrivate) which is, by our definition :

$$\begin{aligned}&\texttt{Trust}(s, \text{AccIntResearch}, \text{Accommodation} \wedge \neg\text{PhonePrivate}) \stackrel{\text{def}}{=} \\ &\quad \texttt{Goal}(s, \text{Accommodation} \wedge \neg\text{PhonePrivate}) \\ &\quad \wedge\texttt{Cert}_s(\texttt{CExt}(\text{AccIntResearch}) \wedge \texttt{CInt}(\text{AccIntResearch}) \\ &\quad \wedge\texttt{Res}(\text{AccIntResearch}, \text{Accommodation} \wedge \neg\text{PhonePrivate}))\end{aligned}$$

Since no agent involved in the composition is a private company, the privacy goal is fulfilled by default. The satisfiability of this trust formula only depends on whether the agents will effectively perform the actions. In our setting, given the beliefs of the student, this holds and the trust is asserted.

AccExtResearch service.

In the case of the AccIntResearch service, the complex action is expressed with the atomic actions in the following way :

$$\begin{aligned}\text{AccExtResearch} = \; & ACIN; CAR; \texttt{if}\ \text{OutputEmpty} \\ & \texttt{then}\ \text{ContactExternalServices}; (R1 + R2 + R3) \\ & \texttt{else}\ \text{Send to student}\end{aligned}$$

The trust formula is

$$\begin{aligned}&\texttt{Trust}(s, \text{AccExtResearch}, \text{Accommodation} \wedge \neg\text{PhonePrivate}) \stackrel{\text{def}}{=} \\ &\quad \texttt{Goal}(s, \text{Accommodation} \wedge \neg\text{PhonePrivate}) \\ &\quad \wedge\texttt{Cert}_s(\texttt{CExt}(\text{AccExtResearch}) \wedge \texttt{CInt}(\text{AccExtResearch}) \\ &\quad \wedge\texttt{Res}(\text{AccExtResearch}, \text{Accommodation} \wedge \neg\text{PhonePrivate}))\end{aligned}$$

The satisfiability of this trust formula depends on the beliefs of the student about the emptiness of the list of accommodations retrieved by the CROUS. We detail the three cases presented in Section 2.4.2 :

1. $\texttt{Cert}_s\neg$OutputEmpty. Since the student believes that the list is not empty, he believes that the CROUS services will send him the list of accommodation without sending his personal information to any company; thus the trust formula is satisfied.

2. $\texttt{Cert}_s$OutputEmpty. Since the student believes that the list is empty, he believes that the CROUS will send his personal information, including phone number, to the *external services* which include private companies. The trust formula is not satisfied in this case.

3. $(\neg\texttt{Cert}_s\neg\text{OutputEmpty}) \wedge (\neg\texttt{Cert}_s\text{OutputEmpty})$. The student has no belief about the emptiness of the list. As a consequence, he must take into account the two possible executions of the composed service. Since the execution triggered when the list is empty involves sending his personal information to private companies, he cannot trust the composite service for fulfilling his goals. The trust formula is not satisfied.

In the cases 2 and 3, the trust formula is not satisfied. An automated analysis of the formula may point the trust problem. In case 2, the trust problem arises from the workflow of the composite service: the student believes that following this workflow, his phone number will be sent to private companies. In case 3, the trust problem also arises from the workflow of the composite service but it depends on the propositional variable OutputEmpty. Such an analysis can be used to prompt the student about the trust issues in a precise way. For example, he can be prompted in case 2 that his phone number will be provided to private companies whereas in case 3 he may be prompted that if the list is empty, his phone number may be provided to private companies.

Case study conclusion

This case study shows how the trust model can be used in an e-service composition context. This first step is to aggregate the general beliefs of the user, using, amongst other sources, his preference set. This step allows also to compute a part of his goals (in our example his privacy goals). The next step is to add the beliefs and goals about the complex action used to achieve his request. This action can be provided by a service composition algorithm. Assuming that the user trusts the algorithm, this provides the beliefs about the workflow of the composed service. The query itself provides the second part of his goals.

When this components are known, the trust formula can be stated and its satisfiability can be analyzed. The analysis procedure is not given in this document; it should allow, whenever the trust formula is unsatisfiable, to give information to the user about why the trust cannot be achieved.

2.5 Conclusion and discussion

Logic-based approaches for trust offer a robust way to deal with situations of interaction where there is a risk for the trustee. This robustness comes from the fact that while the concept of trust stays encoded in the axiomatization of the logic (in our case, an epistemic concept that links trust to the beliefs and choices of the truster), we can still adapt the logic by integrating more information about the applicative context in which our logic will be used.

Our work in this chapter was used as a trust component in a personal information management system or Pims, where information exchange takes place by composing data access services within a network of pims. In this applicative context, our notion of program is associated to service composition, the goal is associated to query described by a set of data to be retrieved (became known to the query agent) associated to some preferences upon the quality for resources and privacy (conditions on the believes of the agents of the network, about each others), This

use of our logic was illustrated in the previous study case.

While the external conditions of trust are associated to the executability of the different services of the composition and resources availability, the internal condition will be related to the interaction protocol, and the reliability of the description of the different services.

Our next chapter presents an extension of this approach by studying the process in which, trust as a beliefs is actually generated, by doing so, our goal is to define abductive mechanisms that can be usable to justify an agent's 'trust state'.

Chapter 3

Nonmonotonic trust operator

3.1 Motivation: NMR and its view of trust

In the last chapter, we presented an epistemic conceptualization of trust. In that model, trust was viewed as a derivative consequence of the epistemic state of the truster.

This view does not endorse any assumptions on how such knowledge is acquired, albeit its importance to understand how the trust state of an agent is constructed. Furthermore, ambiguities can arise when we emphasize the causes of a truster trusting someone . This comes from the non existence of a clear mechanism in our logic that expresses *trust construction*.

Starting from the point of view that trust plays a major role in dealing with risk under uncertainty, the nature of such uncertainty can be a key to propose more efficient models. Until now, the type of uncertainty that we encountered, is of a combinatorial nature: all possible cases are known to the truster, which as a decision process decides how to act under a total description of the situation. A more interesting context to study trust would be of uncertainty as a lack of descriptive material. A simple example of such type of uncertainty can be found in [72], which can be used to describe different agents

that interact in the same context, without the same vocabulary. In our context, a truster may need to interact with a trustee that uses a different vocabulary to describe the context of interaction, which means that they may have different views on their interactions.

In the last chapter, trust was identified with the epistemic states of the agent, in which our *trust formula* will be satisfied, i.e. to the trust formula $\texttt{Trust}(i, j, \pi)$, we associate the set of all epistemic theories (closed and complete set of $\texttt{Cert}_i\varphi$ formulas) of the truster i, that contain the formulas $\texttt{Trust}(i, j, \pi)$. Our goal in this chapter is to define a mechanism (more precise than logical deduction) to identify in each epistemic state, the induced trust. In this work we consider nonmonotonic reasoning as the framework to implement such mechanism. Our choice is supported by the parallels that can be made between deriving how trust is used in social interaction in *common sense* [51] and *causal* [29] reasoning treatment in nonmonotonic frameworks, in both cases there is incomplete knowledge from which conclusions are drawn that are not supported by classical logical inference.

Henceforth, trust interpreted as a concept related to interaction in a social context, is a specialization of the notion of causality to social interaction.

Our work explores this point of view, identifying trust to a reasoning process, instead of just epistemic states that satisfies it. We present trust as a specialized version of causal reasoning, applied to social interactions [11]. Such specialization means that trust is primary a subjective type of causal reasoning. Also, abstract trust will be associated to the notion of causal theory -which we call, *trust theory*-. Such theories are rules that distinguish logical consequences from trust derived facts. Situational trust is identified to a trust theory, which to each context of application, derives different *trusted facts*.

To do so, we adapt the framework *of production and causal inference* that was introduced by Bochman [9], to work within the context of ABC. Production inference relations are one of the most primary and intuitive extensions of classical logic, to implement nonmonotonic reasoning. They are based on the syntactic notion of production rules (where a production rule $A \Rightarrow B$ will be read: A produces, or causes,

B). Our formalism will be used to implement causal and abductive reasoning done in an epistemic context, in order to define inference procedures related to Trust use.

The following sections will present a brief review of production inference relations interpreted using a nonmonotonic semantics and how it can be used to implement both causal reasoning and abductive reasoning.

We will then show how to adapt production inference relations to work with ABC logic formulas and how to use our *epistemic inference relations* to implement nonmonotonic reasoning within multi-agent systems.

We will use the resulting framework to present three distinct notions of trust.

And finally we will present a multi-agent version of our system, as a new modal logic.

3.2 Production inference relations

Nonmonotonic reasoning consists in 'jumping to conclusions' [28] it is a way of concluding from the absence of information and allows to deduce more from a hypothesis than what classical logic licenses. *Production relations* are variants of Makinson & Van der Torre's *input-output logic* [47]. They are used to specify nonmonotonic reasoning procedure that extend classical propositional calculus to support nonmonotonic consequence relation. Our exposition in this section follows [10], to which we refer the reader for more details about the different proofs related to production relations.

Production relation is a relation defined between propositional formulas. In what follow p,q,$\cdots$ denote atomic propositions, φ,ψ,$\cdots$ denote propositional formulas and $\models$ and $\mathtt{Th}$ denote respectively the classical notions of logical entailment and its corresponding logical closure operator.

Definition 9 *A* production inference *relation is a binary relation* $\Rightarrow$ *defined on the set of classical propositions, that satisfies the following*

postulates:

(Strengthening) *If* $\varphi \models \psi$ *and* $\psi \Rightarrow \theta$, *then* $\varphi \Rightarrow \theta$;

(Weakening) *If* $\varphi \Rightarrow \psi$ *and* $\psi \models \theta$, *then* $\varphi \Rightarrow \theta$;

(And) *If* $\varphi \Rightarrow \psi$ *and* $\varphi \Rightarrow \theta$, *then* $\varphi \Rightarrow \psi \wedge \theta$;

(Truth) $\top \Rightarrow \top$;

(falsity) $\bot \Rightarrow \bot$.

An element $\varphi \Rightarrow \psi$ of $\Rightarrow$ will be called a *production rule.*

It is worth mentioning that the main difference between the classical notion of entailment and the notion of production is that reflexivity ($\varphi \Rightarrow \varphi$) is not guaranteed for the later.

In what follows, production rules are extended to arbitrary sets of propositions as follow: for any set Φ of propositions, we define $\Phi \Rightarrow \psi$ as follows:

$$\Phi \Rightarrow \varphi \text{ iff } \bigwedge u \Rightarrow \varphi, \text{ for some finite } u \subseteq \Phi$$

We associate to $\Rightarrow$, its corresponding *derivability operator*

$$\mathcal{C}(\Phi) = \{\varphi \mid \Phi \Rightarrow \varphi\}$$

.

Remark that $\mathcal{C}$ is monotonic:

$$\text{if } \Gamma \subseteq \Phi, \text{ then } \mathcal{C}(\Gamma) \subseteq \mathcal{C}(\Phi)$$

and deductively closed:

$$\text{for any } \Phi, \mathcal{C}(\Phi) = \mathtt{Th}(\mathcal{C}(\Phi))$$

Even so, $\mathcal{C}$ is still not inclusive: $\Phi \subseteq \mathcal{C}(\Phi)$ does not always hold. Also, it is not idempotent, that is, $\mathcal{C}(\mathcal{C}(\Phi))$ can be different from $\mathcal{C}(\Phi)$.

The main semantics used to interpret production relations is based on the notion of *bimodels*, where a bimodel is just a pair of deductively closed set of formulas, viewed as an initial and a possible final states.

Definition 10 *A pair of consistent, deductively closed sets of formulas will be called a* classical bimodel. *A set of classical bimodels will be called a* classical binary semantics.

We use the symbols u, v, etc. for classical bimodels. A semantical reformulation of the notion of bimodels as a pair of sets of interpretations is possible as follows:

$$\langle u, v\rangle \equiv \langle \{I \mid I \models u\}, \{I \mid I \models v\}\rangle$$

A classical binary semantics can also be seen as a binary relation between deductively closed theories. We will use the notation $u\mathcal{B}v$ to denote the fact that the bimodel $\langle u, v\rangle$ belongs to the classical binary semantics $\mathcal{B}$.

Definition 11 (validity) *A production rule $\varphi \Rightarrow \psi$ will be said to be valid in a classical binary semantics $\mathcal{B}$ if, for any bimodel $\langle u, v\rangle$ from $\mathcal{B}$, $\varphi \in u$ only if $\psi \in v$.*

In regards to the notion of validity, the syntactic and semantics formulations of bimodels are equivalent.

Lemma 7 *For any* classical binary semantics $\mathcal{B}$*, the associated set of production rules that are valid in $\mathcal{B}$, is a production relation (denoted $\Rightarrow_{\mathcal{B}}$).*

A completeness result can be obtained by constructing for any production relation $\Rightarrow$, its *canonical semantics*

$\mathcal{B}_{\Rightarrow} = \{\langle w, \mathcal{C}(w)\rangle \mid w$ is a consistent and deductively closed set of formulas$\}$

Theorem 8 *If $\mathcal{B}_{\Rightarrow}$ is the canonical semantics for a production relation $\Rightarrow$, then, for any set of propositions Φ and any formula φ,*

$$\Phi \Rightarrow \varphi \quad \textit{iff } \varphi \in v, \textit{ for any bimodel } (w, v) \in \mathcal{B}_{\Rightarrow} \textit{ such that } \Phi \subseteq w$$

Corollary 9 *A binary relation $\Rightarrow$ on the set of propositions is a production inference relation if and only if it is determined by a classical binary semantics.*

The notion of *causal theory* is the main way to specify production relations. By a causal theory we mean a set of production rules. Since all postulates for production relations are Horn ones, for any causal theory Δ there exists at least production relation that includes Δ. We will denote it by $\Rightarrow_\Delta$, while $\mathcal{C}_\Delta$ will denote the derivability operator. $\Rightarrow_\Delta$ is the set of all the production rules that can be derived from Δ using the postulates for production relations.

If $\Delta(\Phi)$ denotes the set of all proposition that can be directly produced from Φ by Δ, that is

$$\Delta(\Phi) = \{\psi \mid (\varphi \Rightarrow \psi) \in \Delta, \text{ for some } \varphi \in \Phi\}$$

A description of $\Rightarrow_\Delta$, can be formulated as follows:

Proposition 7 $\mathcal{C}_\Delta(\Phi) = \mathtt{Th}(\Delta(\mathtt{Th}(\Phi)))$

3.2.1 Regular production inference

As presented above, productions relations offer a way to link assumptions to their conclusion in a more deliberate way than the classical entailment relation. Still they may seem too restrictive to handle complex type of reasoning. For example, production relations do not allow to reuse productions as inputs to produce more formulas.

Such mechanism needs to be specified as a postulate. The cut mechanism is a way to implement reusability of rules as follow:

(Cut) If $\varphi \Rightarrow \psi$ and $\varphi \wedge \psi \Rightarrow \theta$, then $\varphi \Rightarrow \theta$

An equivalent constraint can be defined for the production operator:

$$\mathcal{C}(\Phi \cup \mathcal{C}(\Phi)) \subseteq \mathcal{C}(\Phi)$$

A production relation that satisfies the cut mechanism is called a *regular production relation*. This special type of production relation satisfies a certain number of properties. The most notable ones are:

(Transitivity) If $\varphi \Rightarrow \psi$ and $\psi \Rightarrow \theta$, then $\varphi \Rightarrow \theta$. This corresponds to the following constraint on the operator: $\mathcal{C}(\mathcal{C}(\Gamma)) \subseteq \mathcal{C}(\Gamma)$.

(Constraint) If $\varphi \Rightarrow \psi$, then $\varphi \wedge \neg\psi \Rightarrow \bot$. This property introduces a special kind of production rule, $\varphi \Rightarrow \bot$, as a way to describe purely factual informations (by saying that φ is explanatory inconsistent and should not hold in any output state).

(Coherence) If $\varphi \Rightarrow \neg\varphi$, then $\varphi \Rightarrow \bot$. As a special case of Constraint, coherence expresses that if a proposition produces propositions that are incompatible with it, then it is explanatory inconsistent.

Regular production relations can be described using the following notion of theory.

Definition 12 *A set Φ of propositions will be called a* theory *of a production relation, if it is deductively closed, and $\mathcal{C}(\Phi) \subseteq \Phi$.*

Φ will be called a theory of a causal theory Δ, if it is a theory of $\Rightarrow_\Delta$.

Furthermore, theories that are worlds, i.e., a complete deductively closed set of propositions, have a very simple characterization of regular production relations:

Lemma 10 *A world α is a theory of a regular production relation if and only if $\alpha \not\Rightarrow \bot$*

Regular production relations can be characterized using only inclusive bimodels ($\langle u, v\rangle$ such that $v \subseteq u$). The corresponding semantics will be called *consistent* -or *inclusive*- binary semantics.

Theorem 11 *$\Rightarrow$ is a regular production relation if an only if it is generated by a consistent binary semantics.*

We denote $\Rightarrow^r_\Delta$ the least regular production relation containing a causal theory Δ. As a consequence of the above characterization, we obtain a constructive characterization of $\Rightarrow^r_\Delta$.

Proposition 8 *$\Phi \Rightarrow^r_\Delta \varphi$ if and only if $\varphi \in \mathtt{Th}(\Delta(u))$, for any Δ-theory u, such that $\Phi \subseteq u$*

A simple characterization can be obtained by using $Cl(\Phi)$ to denote the least Δ-theory containing Φ.

Corollary 12 *Given $\mathcal{C}^r_\Delta$, a production operator that corresponds to the causal theory Δ:*

$$\mathcal{C}^r_\Delta(\Phi) = \mathtt{Th}(\Delta(Cl(\Phi)))$$

Using regular production relations, we can define a proper notion of equivalence, that allows to group propositions in sets with elements substituable in production rules. Namely two propositions φ and ψ are *production equivalent* with respect to a production relation, if $\top \Rightarrow (\varphi \leftrightarrow \psi)$ holds. Then we have

Lemma 13 *Propositions φ and ψ are production-equivalent with respect to a regular production relation $\Rightarrow$ if and only if any occurrence of φ can be replaced by ψ in the rules of $\Rightarrow$.*

Until now, we considered a monotonic interpretation of production inference, the next definition presents an alternative nonmonotonic semantics, determined by production relations in a natural way, providing a logical basis for a particular form of nonmonotonicity[1] .

Definition 13 .

- *A set of propositions Φ is called an* exact theory*, if it is consistent and $\Phi = \mathcal{C}(\Phi)$.*
- *A set Φ of propositions is an* exact theory *of a* causal theory *Δ, if it is an exact theory of $\Rightarrow_\Delta$*
- *A* general nonmonotonic semantics *of a production inference relation (or a causal theory) is the set of all its exact theories.*

Using a general exact semantics correspond to accepting the explanatory closure assumptions, which stipulates that any proposition

[1]Due to the fact that the production operator C is not reflexive, an important distinction among theories of a production relation can be made.

is actually produced (explained by) propositions that are accepted in this state.

To this point, we showed how to associate to a regular production relation, a general nonmonotonic semantics, without defining a definite notion of validity. This is mainly due to the fact that different notion of validities are candidate. Standard one are based on total, minimal or maximal inclusion, and each one of these validity notions shows pros and cons, depending on the application our logic is meant for. Nevertheless, the nonmonotonic nature of the validity is preserved, since adding new rules to a production relation changes nonmonotonically its semantics.

Exact theories can be seen as fixed points of the production operator $\mathcal{C}$, and since this operator is monotonic and continuous, exact theories (and hence the nonmonotonic semantics) always exist. For regular production relations, the least exact theory coincides with $\mathcal{C}(\top)$, that is, with the set of propositions that are produced by tautologies. In addition, the union of any chain of exact theories (with respect to set inclusion) is an exact theory, so any exact theory is included in a maximal such theory.

Lemma 14 *If $\Rightarrow$ is a regular production relation associated to a production operator $\mathcal{C}$, and $\Phi \subseteq \mathcal{C}(\Phi)$ for any set of formulas Φ, then $\mathcal{C}(\Phi)$ is an exact theory of $\Rightarrow$.*

An equivalent lemma can be defined for causal theories

Lemma 15 *Γ is an exact theory of a causal theory Δ if and only if $\Gamma = \mathtt{Th}(\Delta(\Gamma))$, for any set of formulas Γ.*

Regular production relations are an adequate and maximal logical framework for reasoning with exact theories

Definition 14 *Two causal theories will be called nonmonotonically equivalent if they have the same general nonmonotonic semantics.*

The next lemma links causal theories and regular production relations under general nonmonotonic semantics.

Lemma 16 *Any causal theory Δ is nonmonotonically equivalent to $\Rightarrow^{r}_{\Delta}$.*

The following holds:

Corollary 17 *Regularly equivalent theories are nonmonotonically equivalent.*

But the inverse of the last equivalence does not hold, due to the monotonicity of regular equivalence, that can not be guaranteed when new rules are added to regularly equivalent relations. This shortcoming means that we may need a stronger monotonic notion of equivalence, that will be preserved under the addition of new rules.

Definition 15 *Two causal theories* Φ *and* Γ *will be said to be strongly equivalent if, for any set* Θ *of causal rules,* $\Phi \cup \Theta$ *is nonmonotonically equivalent to* $\Gamma \cup \Theta$*.*

This kind of independence can be seen as context free equivalence between causal theories.

Theorem 18 *Two causal theories are strongly equivalent if and only if they are regularly equivalent.*

The above result implies, in effect, that regular production relations are maximal inference relations that are adequate for reasoning with causal theories: any postulate that is not valid for regular production relations can be 'falsified' by finding a suitable extension of two causal theories that would determine different nonmonotonic semantics, and hence would produce different nonmonotonic conclusions.

Exact theories are fixed points, where all elements can be explained. Such semantics make regular production relation a promising framework to study abductive reasoning.

3.2.2 Abducive production inference

A special case of production relation can be used to implement abductive reasoning.

We start by defining an *abductive framework* as a pair $\mathbb{A} = \langle Cn, \mathcal{A} \rangle$ where Cn is a consequence relation and $\mathcal{A}$ is a set of distinguishable

propositions called *abducibles*. A proposition φ is *explainable* in an abductive framework $\mathbb{A}$ if there exists a consistent set of abducibles $a \subseteq \mathcal{A}$ such that $A \in Cn(a)$.

A correspondence was presented between abductive framework and production relation in [10], where a production relation corresponds to a consequence relation of the abductive framework, and the set of abducives corresponds to the set of propositions that are self-explained $(\varphi \Rightarrow \varphi)$.

In what follows, we define abductive production relations as a special case of regular production relations in which we take into account the role of abducibles.

Definition 16 .

- *A proposition φ will be called an abducible in a production relation $\Rightarrow$, if $\varphi \Rightarrow \varphi$;*
- *A production relation will be called abductive if it is regular and satisfies*

  (Abduction) *If $\varphi \Rightarrow \psi$, then $\varphi \Rightarrow \theta \Rightarrow \psi$ for some abducible θ*

This corresponds to the following condition on the corresponding production operator: for any world α, $\mathcal{C}(\alpha) = \mathcal{C}(\alpha \cap \mathcal{A})$, where $\mathcal{A}$ is the set of abducible of $\Rightarrow$.

Thus, production inference allows to give a syntax-independent representation of abductive reasoning, This comes from the fact that abducibles are not defined as a distinguished set of propositions, but logically as propositions that satisfy certain property (in this case reflexivity).

The well-known 'wet grass' example [55] can be used as an example to illustrate the uses of abductive relations.

Example 4 *Assume that an abductive system $\Rightarrow$ is determined by the set Δ of rules*

$$Rained \Rightarrow Grasswet$$
$$Sprinkler \Rightarrow Grasswet$$
$$Rained \Rightarrow Streetwet$$

and the set abducibles

$$\mathcal{A} = \{Rained, \neg Rained, Sprinklers, \neg Sprinklers, \neg Grasswet\}$$

In this example, Rained is an independent parameter since both Rained and ¬Rained are abducible and hence, self explained (the same goes for Sprinkler), also ¬Grasswet does not require any explanation. However Grasswet will demand explanation, i.e. Under a general nonmonotonic semantics interpretation, any exact theory of ⇒ that verifies Grasswet will need to verify either Rained or sprinkler

One possible way to construct an abductive production relation is to extract it from a regular production relation by identifying a set of abducibles, and restricting it to the subrelation that satisfies the abduction rule.

So, Given a production relation ⇒, we define the following production relation

$$\varphi \Rightarrow^a \psi \equiv (\exists\theta)(\varphi \Rightarrow \theta \Rightarrow \theta \Rightarrow \psi)$$

Theorem 19 *If ⇒ is a regular production relation, then $\Rightarrow^a$ is the greatest abductive production relation included in ⇒.*

A regular production relation and its abductive subrelation share some similar properties like conserving the same abducibles, the same axioms, and the same constraints. Also, using the general nonmonotonic semantics amounts to accepting the explanatory closure assumption, i.e. that any accepted proposition, need to be explained. Which also means in the case of either a finite set of proposition, or infinite set of proposition with a finite number of equivalent classes, one is ensured to encounter abducible within any *production chain.* We can conclude that in many *regular* cases, the general nonmonotonic semantics of a production relation should coincide with the semantics of its abductive subrelation.

Definition 17 *A production relation will be called* quasi-abductive *if it is nonmonotonically equivalent to its abductive subrelation.*

A more general way to identify quasi-abductive production relation is to use a property that guarantee the presence of an abducible, in any 'chain' of production.

Definition 18 *A regular production relation $\Rightarrow$ will be called* well-founded *if any infinite sequence $\{\varphi_0, \varphi_1, \varphi_2, \cdots\}$ of propositions such that $\varphi_{n+1} \Rightarrow \varphi_n$, for every $n \geq 0$, contains an abducible.*

We recall that a regular production relation is finitary if it is a least regular production relation containing some finite causal theory

Theorem 20 *Any finitary regular production relation is well-founded.*

The following theorem ties it all:

Theorem 21 *Any well-founded regular production relation is quasi-abductive.*

The framework of abductive production relations seems well suited to extend the trust framework that we presented in the last chapter by providing a way to explicit which belief of the truster explained his trust. Such beliefs can be seen as abducibles, observation and prior knowledge about the trustee.

Another possible extension of our precedent work is to study how trust affects the truster beliefs. Either as a primitive abducible, or associated to some contextual (core belief). In this context, trust can be seen as a way to specify, another channel of knoweldge acquisition (that can be subjective in nature).

In order to study trust in this framework, a last task need to be achieved, which is to adapt abductive production relations, to our subject vie of agency, this will be done in the next section.

3.3 Epistemic production inference relations

In this section, we we present our contribution as an extension of Bochman's production relations. Our goal is to present a framework

that extends our ABC, to work with the epistemic knowlege of specif agents.

The logical consequence relation of ABC logic and its related closure operator will be denoted respectively $\models_{\mathsf{ABC}}$ and Th_{ABC}, we drop the subscript "ABC " for convenience. We also recall that ABC logic's formulas are interpreted under the set of pointed models $\mathcal{PM}$.

Given an agent i, we define an epistemic consequence relation $\models_i$ as a 4-ary relation, such that for any two pairs of formulas $(\varphi_1, \varphi 2)$ and $(\psi_1, \psi 2)$

$(\varphi_1, \varphi 2) \models_i (\psi_1, \psi 2)$ iff for any $pm \in \mathcal{PM}$,
$pm \models \mathtt{Cert}\varphi_1 \wedge \mathtt{Plaus}\varphi_2$ implies $\quad pm \models \mathtt{Cert}_i\psi_1 \wedge \mathtt{Plaus}_i\psi_2$

The entailement operator will be denoted Th_i

We define *Epistemic production relations*, as 4-ary relations defined on $\mathcal{L}_{\mathsf{ABC}}$, that associate to an epistemic state of an ABC agent, a second epistemic state.

This means that the definitive epistemic state of an agent can no longer be defined by the characterization of his weak and strong belief relation (BW and BS), his epistemic production relation $\Rightarrow_i$needs also to be taken in account.

Definition 19 *An* epistemic production relation *associated to an agent i is a relation $\Rightarrow_i \subseteq (\mathcal{L}_{\mathsf{ABC}} \times \mathcal{L}_{\mathsf{ABC}}) \times (\mathcal{L}_{\mathsf{ABC}} \times \mathcal{L}_{\mathsf{ABC}})$ that satisfies the following properties:*

(Strengthening) *If $\theta_1, \theta_2 \models_i \varphi_1, \varphi_2$ and $(\varphi_1, \varphi_2) \Rightarrow_i (\psi_1, \psi_2)$, then $(\theta_1, \theta_2) \Rightarrow_i (\psi_1, \psi_2)$*

(Weakening) *If $\psi_1, \psi_2 \models_i \theta_1, \theta_2$ and $(\varphi_1, \varphi_2) \Rightarrow_i (\psi_1, \psi_2)$,then $(\psi_1, \psi_2) \Rightarrow_i (\theta_1, \theta_2)$*

(And) *If $(\varphi_1, \varphi_2) \Rightarrow_i (\psi_1, \psi_2)$ and $(\varphi_1, \varphi_2) \Rightarrow_i (\theta_1, \theta_2)$,then $(\varphi_1, \varphi_2) \Rightarrow_i (\psi_1 \wedge \theta_1, \psi_2 \wedge \theta_2)$*

(Truth) $(\top, \top) \Rightarrow_i (\top, \top)$

(Falsity) $(\bot, \bot) \Rightarrow_i (\bot, \bot)$

(Plaus-weakening) *If* $\varphi_1, \varphi_2 \Rightarrow_i \psi_1, \psi_2$ *then* $\varphi_1, \varphi_2 \Rightarrow_i \top, \psi_1$

(Necessitation) *If* $(\varphi_1, \varphi_2) \Rightarrow_i (\psi_1, \psi_2)$ *,then* $(\varphi_1, \varphi_2) \Rightarrow_i (\mathtt{Cert}_i\psi_2, \mathtt{Plaus}_i\psi)$

We will call $\varphi_1, \varphi_2 \Rightarrow_i \psi_1, \psi_2$ an epistemic production rule.

We extend this notion of rule to sets of premises as follows: given two sets of $\mathcal{L}_{\mathsf{ABC}}$ formulas, Γ and Ψ:

$$(\Gamma, \Psi) \Rightarrow_i (\varphi, \psi) \text{ iff } (\wedge\Gamma', \wedge\Psi') \Rightarrow_i (\varphi, \psi)$$

for some finite sets $\Gamma' \subseteq \Gamma$ and $\Psi' \subseteq \Psi$

The next step in adapting production relation to an epistemic view, is to define a corresponding production operator $\mathcal{C}_i$ associated to the agent i. For any pair $\langle \Gamma, \Psi \rangle$ of sets of formulas, $\mathcal{C}_i(\Gamma, \Psi)$ denotes the pair of sets of formulas *produced* by $\langle \Gamma, \Psi \rangle$, that is:

$$\mathcal{C}_i(\Gamma, \Psi) = \{\langle \varphi, \psi \rangle \mid (\Gamma, \Psi) \Rightarrow (\varphi, \psi)\}$$

Lemma 22 *The production operator* $\mathcal{C}_i$ *of an epistemic production relation is* deductively closed *under the entailment operator of* ABC *logic, i.e.,* $\mathcal{C}_i(\Gamma, \Psi) = \langle \Theta_c, \Theta_p \rangle$*, implies that* $\Theta_c = Th_{\mathsf{ABC}}(\Theta_c)$ *and* $\Theta_p = Th_{\mathsf{ABC}}(\Theta_p)$

Proof 4 *Suppose* $\mathcal{C}_i(\Gamma, \Psi) = (\Theta, \chi)$ *and suppose* $\theta \in Th_{\mathsf{ABC}}(\Theta)$*. Then* $\{theta_1, \ldots, \theta_n\} \models \theta$ *for some finite subset* $\{theta_1, \ldots, \theta_n\}$ *of* Θ *such that* $(\theta_k, \chi_k \in \mathcal{C}_i(\Gamma, \Psi)$ *for* $1 \leq k \leq n$*. The latter means that for every* k *there are finite* Γ_k *and* Ψ_k *such that* $(\wedge\,\Gamma_k, \wedge\,\Psi_k) \Rightarrow_i (\theta_k, \chi_k)$ *for some* χ_k*. Then by* ***(And)*** *we have* $(\wedge\,\Gamma_k, \wedge\,\Psi_k) \Rightarrow_i (\wedge_k\,\theta_k, \wedge_k\,\chi_k)$*. Finally,* $(\wedge\,\Gamma_k, \wedge\,\Psi_k) \Rightarrow_i (\theta, \wedge_k\,\chi_k)$ *by* ***(Weakening)****. Therefore* $\theta \in \Theta$*.*

The proof for the plausibility part of the inference is similar.

Our production operator is also monotonic.

Lemma 23 *If* $\Gamma' \subseteq \Gamma$ *and* $\Psi' \subseteq \Psi$ *, then* $\mathcal{C}_i(\Gamma', \Theta) \subseteq \mathcal{C}_i(\Gamma, \Theta)$ *and* $\mathcal{C}_i(\Theta, \Psi) \subseteq \mathcal{C}_i(\Theta, \Psi')$ *for any set of formulas* Θ*.*

Proof 5 *Suppose* $\Gamma' \supseteq \Gamma$ *and suppose* (θ, χ) *in* $\mathcal{C}_i(\Gamma, \Psi)$*. Then* $(\Gamma, \Psi) \Rightarrow_i (\theta, \chi)$*, i.e.,* $(\wedge\,\Gamma_0, \wedge\,\Psi_0) \Rightarrow_i (\theta, \chi)$ *for some finite* $\Gamma_0 \subseteq \Gamma$ *and* $\Psi_0 \subseteq \Psi$*. By definition we then also have* $(\Gamma', \Psi) \Rightarrow_i (\theta, \chi)$*, i.e.,* θ, χ *in* $\mathcal{C}_i(\Gamma', \Psi)$*.*

Still, the operator is not inclusive, neither it is idempotent (for example, let $\Gamma = \Psi$ be the set of ABC theorems and let $\Theta = \chi$ be the set of all ABC formulas.)

In order to define the semantics of our epistemic production relations, we will both present a syntactic (formulas based) and semantics (model based) semantics.

Definition 20 *We will call four consistent deductively closed sets an* epistemic bimodel*(abbreviated ebimodel). A set of ebimodels will be called a* ABC epistemic binary semantics*.*

We will use the notation $(u_1, u_2)\mathcal{E}(v_1, v_2)$ *to denote the fact that the ebimodel* (u_1, u_2, v_1, v_2) *belongs to the epistemic binary semantics* $\mathcal{E}$*.*

An equivalent semantics based on the set of pointed models $\mathcal{PM}$ can be defined as follow:

$$\begin{aligned}\langle \Gamma_c, \Gamma_p, \Psi_c, \Psi_p\rangle \quad \equiv \quad & \langle \{pm \in \mathcal{PM} \mid pm \models_{\mathsf{ABC}} \mathtt{Cert}_i\gamma, \text{ for any } \gamma \in \Gamma_c \\ & \quad \text{and } pm \models_{\mathsf{ABC}} \mathtt{Plaus}_i\gamma', \text{ for any } \gamma' \in \Gamma_p\} \\ & , \{pm \in \mathcal{PM} \mid pm \models_{\mathsf{ABC}} \mathtt{Cert}_i\psi, \text{ for any } \psi \in \Psi_c \\ & \quad \text{and } pm \models_{\mathsf{ABC}} \mathtt{Plaus}_i\psi', \text{ for any } \psi' \in \Psi_c\}\rangle\end{aligned}$$

Since both formulations of the semantics can be interchangeable, we will use the most convenient one to simplify our proofs.

We have now all the necessary ingredients to define the notion of validity.

Definition 21 *An epistemic production rule* $\varphi_1, \varphi_2 \Rightarrow_i \psi_1, \psi_2$ *will be said to be valid in an epistemic binary semantics* $\mathcal{E}$*, if for any ebimodel* $\langle u_1, u_2, v_1, v_2\rangle$ *from* $\mathcal{E}$*,* $\varphi_1 \in u_1$ *and* $\varphi_2 \in u_2$*, only if* $\psi_1 \in v_1$ *and* $\psi_2 \in v_2$*.*

Just as for production relations, an epistemic binary semantics characterizes a specific epistemic production relation.

Lemma 24 *For any epistemic binary semantics* $\mathcal{E}$*, the associated set of epistemic production rules that are valid in* $\mathcal{E}$*, is an epistemic production relation (that we will denote* $\Rightarrow_{\mathcal{E}i}$*)*

A completeness result can be obtained by constructing for any epistemic production relation $\Rightarrow_i$, its canonical semantics

$$\mathcal{E}_{\Rightarrow_i} = \{\langle u_1, u_2, \mathcal{C}_i(u_1, u_2)\rangle\}$$

Theorem 25 *If $\mathcal{E}_{\Rightarrow_i}$ is the epistemic canonical semantics for an epistemic production relation $\Rightarrow_i$, then for any sets of propositions Γ and Ψ and any pair of formulas φ, ψ:*

$\Gamma, \Psi \Rightarrow_i \varphi, \psi$ iff $\varphi \in v_1$ and $\psi \in v_2$, for any ebimodel $(u_1, u_2, v_1, v_2) \in \mathcal{E}_{\Rightarrow_i}$ such that $\Gamma \subseteq u_1$ and $\Psi \subseteq u_2$

Corollary 26 *A 4-ary relation $\Rightarrow$on $\mathcal{L}_{\mathsf{ABC}}$ formulas is an epistemic production inference associated to the agent i, if and only if it is determined by an epistemic binary semantics.*

We will use the notion of *epistemic causal theory*, as an arbitrary set of epistemic production rules. It should be noted again that the notion of causality is associated to the epistemic state of an agent, comprising strong and weak beliefs, this means that such notion of causality is subjective in nature.

We also point out that the notion of causality as defined here, is not temporal, $\langle\varphi, .\rangle \Rightarrow_i \langle\psi, .\rangle$ does not means that i strongly believes that φ occurred before ψ, but rather that i would associate to the observation of ψ (under no specific context), φ as an explanation. This vision defines a hierarchy of dependencies that will be of great use for us, when we will try to use trust and how it can be used to explain the agent's beliefs. The same remark applies to weak beliefs.

Epistemic causal theories can be used to construct epistemic production relations. Let $\Delta(\Gamma, \Psi)$ denote the set of all production rules that can be directly produced from $\langle\Gamma, \Psi\rangle$ by Δ, that is

$$\Delta(\Gamma, \Psi) = \{\langle\psi_1, \psi_2\rangle \mid \varphi_1, \varphi_2 \Rightarrow_i \psi_1, \psi_2 \text{ for some } \varphi_1, \varphi_2 \in \Gamma, \Psi\}$$

We will adopt the following description of $\Rightarrow_{i\Delta}$, for the epistemic production relation that is generated by Δ:

Proposition 9 $\mathcal{C}_\Delta(\Gamma, \Theta) = Th_i(\Delta(Th_i(\Gamma, \Theta)))$

3.3.1 Regular epistemic production inference

A regular epistemic production relation is an epistemic production relation that satisfies the cut rule.

Cut if $\varphi_1, \varphi_2 \Rightarrow_i \psi_1, \psi_2$ and $\varphi_1 \wedge \psi_1, \varphi_2 \wedge \psi_2 \Rightarrow_i \theta_1, \theta_2$, then $\varphi_1, \varphi_2 \Rightarrow_i \theta_1, \theta_2$

This is equivalent to the following property of the operator:

$$\mathcal{C}_i(\Gamma \cup C_1, \Psi \cup C_2) \subseteq \mathcal{C}_i(\Gamma, \Psi)$$

where $\mathcal{C}_i(\Gamma, \Psi) = \langle C_1, C_2 \rangle$.

Adding the cut rule makes some other properties hold:

Transitivity If $\varphi_1, \varphi_2 \Rightarrow_i \psi_1, \psi_2$ and $\psi_1, \psi_2 \Rightarrow_i \theta_1, \theta_2$, then $\varphi_1, \varphi_2 \Rightarrow_i \theta_1, \theta_2$

Constraint If $\varphi_1, \varphi_2 \Rightarrow_i \psi_1, \psi_2$, then $\varphi_1 \wedge \neg\psi_1, \varphi_2 \Rightarrow_i \bot, \psi_2$ and $\varphi_1, \varphi_2 \wedge \neg\psi_2 \Rightarrow_i \psi_1, \bot$

This property introduced a new kind of productions $\varphi_1, \varphi_2 \Rightarrow_i \bot, \psi_2$ and $\varphi_1, \varphi_2 \Rightarrow_i \psi_1, \bot$ as a way to describe purely factual information. By saying that φ_1 (resp. φ_2), is explanatory inconsistent w.r.t. the agents i strong (resp. weak) belief. We will call this special kind of rules, strong and weak contraints.

Coherence If $\varphi_1, \varphi_2 \Rightarrow_i \neg\varphi_1, \psi_2$, then $\varphi_1, \varphi_2 \Rightarrow_i \bot, \psi_2$. The same goes for the second premise of the rule. Which are special kind of contraints expressing that if a production is not compatible with its antecedents, then it is explanatory inconsistent.

Definition 22 .

A pair of sets of ABC *formulas* $\langle \Gamma, \Psi \rangle$ *is a theory of an epistemic production relation, if it is deductively closed, and* $\mathcal{C}_i(\Gamma, \Psi) \subseteq \langle \Gamma, \Psi \rangle$.

Furthermore, theories that describe a unique pair of belief relations BS_i, BW_i —special classes of models in ABC logic, where we have the same belief relation for the agent i— are simple to characterize. We

will call such theory, an epistemic theory. Using theories shows that we are interested in rearranging beliefs using the notion of explanation that is coded within the production relation, we are not generating knoweldge

Lemma 27 *An epistemic theory $\langle \Gamma, \Psi \rangle$ for the agent i is a theory of a regular epistemic production relation, if an only if $\Gamma, \Psi \not\Rightarrow_i \varphi, \bot$ and $\Gamma, \Psi \not\Rightarrow_i \bot, \psi$, for any formulas φ and ψ.*

Regular epistemic production relations can be characterized using only inclusive ebimodels ($\langle u_1, u_2, v_1, v_2 \rangle$ such that $v_1 \subseteq u_1$ and $v_2 \subseteq u_2$). The corresponding semantics will be called and *inclusive epistemic binary semantics.*

Theorem 28 *$\Rightarrow_i$is a regular epistemic production relation if and only if it is generated by an inclusive epistemic binary semantics.*

We will denote $\Rightarrow^r_{i\Delta}$, the least regular production relation containing an epistemic causal theory Δ. We can obtain a constructive characterization of $\Rightarrow^r_{i\Delta}$using the theorem above.

Proposition 10 *$\Gamma, \Psi \Rightarrow^r_{i\Delta} \varphi, \psi$ if and only if, $\langle \varphi, \psi \rangle \in Th_i(\Delta(u_1, u_2))$, for any Δ-theory $\langle u_1, u_2 \rangle$, such that $\Gamma, \Psi \subseteq \langle u_1, u_2 \rangle$.*

Corollary 29

$$\mathcal{C}^r_{i\Delta}(\Gamma, \Psi) = Th_i(\Delta(Cl_e(\Gamma, \Psi)))$$

where $Cl_e(\Gamma, \Psi)$ is the least epistemic Δ-theory containing $\langle \Gamma, \Psi \rangle$.

The next thing to do, is to define a proper notion of equivalence, that allows us to define classes of interchangeable formulas in production rules. Two formulas are certainly (resp. plausibly) equivalent if $\top, \varphi_2 \Rightarrow_i \varphi \leftrightarrow \psi, \psi_2$ (resp. $\varphi_1, \top \Rightarrow_i \psi_1, \varphi \leftrightarrow \psi$), for any formulas $\varphi_1, \varphi_2, \psi_1, \psi_2$.

Lemma 30 *If two formulas are certainly (resp. plausibly) equivalent in $\Rightarrow_i$, they can be interchangeable within the left (resp. right) occurrences in any rules of $\Rightarrow_i$.*

Like production relations, epistemic production relations can characterize a nonmonotonic semantics, where nonmonotonicity is a byproduct of the lack of reflexivity we will talk here about left and right reflexivity of the production operator.

Definition 23 .

- *A pair of sets of formulas* $\langle \Gamma, \Psi \rangle$ *is called an* exact epistemic theory *if it is consistent and* $\langle \Gamma, \Psi \rangle = \mathcal{C}_i(\Gamma, \Psi)$
- *A pair of sets* $\langle \Gamma, \Psi \rangle$ *of formulas is an exact epistemic theory of a causal epistemic theory* Δ *if it is an exact epistemic theory of* $\Rightarrow_{i\Delta}$.
- *A general epistemic nonmonotonic semantics of a production inference relation of an epistemic production relation (or a causal epistemic theory) is the set of all its exact theories.*

By restricting our attention to exact semantics, we are accepting the explanatory closure assumption, meaning that all formulas are produced (or explained) in final states.

Still, the notion of validity can be defined in different manners.

The nonmonotonicity comes from the fact that excluding non-exact epistemic theories, makes the process of adding new rules, changes the semantics of a relation in a nonmonotonic way.

Also, exact epistemic theories are fix points of the production operator $\mathcal{C}_i$, and since this operator is monotonic and continuous, exact epistemic theories (and hence non monotonic epistemic semantics) alway exists. For regular epistemic production relations, the least exact theory coincides with $\mathcal{C}_i(\top, \top)$. In addition, the union of any chain of exact epistemic theories, (with respect to set inclusion), is an exact epistemic theory, so any exact epistemic theory is included in a maximal such theory.

Lemma 31 *If* $\Rightarrow_i$ *is a regular epistemic production relation, and* $\langle \Gamma, \Psi \rangle \subseteq \mathcal{C}_i(\Gamma, \Psi)$*, then* $\mathcal{C}_i(\Gamma, \Psi)$ *is an exact theory of* $\Rightarrow_i$.

An equivalent lemma can be defined for causal theories

Lemma 32 *$\langle \Gamma, \Psi \rangle$ is an exact epistemic theory of a causal theory Δ if and only if $\langle \Gamma, \Psi \rangle = \mathcal{C}_i(\Delta(\Gamma, \Psi))$*

Regular epistemic production relations are an adequate and maximal logic framework for reasoning with exact epistemic theories.

Definition 24 *Two causal epistemic theories will be called nonmonotonically equivalent if they have the same general nonmonotonic semantics.*

The next lemma links epistemic causal theories and regular epistemic production relations under the general nonmonotonic epistemic semantics.

Lemma 33 *Any causal epistemic theories Δ is nonmonotonically equivalent to $\Rightarrow^r_{i\Delta}$.*

The two notions of equivalent may be linked

Corollary 34 *Regulary equivalent theories are nonmonotonically equivalent*

But the converse does not hold (adding rules does change the nonmonotonic semantics in an unpredictable way). We need a stronger monotonic equivalence notion, in order to have an equivalence to the nonmonotonic semantics.

Definition 25 *Two epistemic causal theories Δ and Δ' will be called strongly equivalent, if for any set Θ of causal epistemic rules, $\Delta \cup \Theta$ is nonmonotonically equivalent to $\Delta' \cup \Theta$.*

Such kind of equivalence is a context-free equivalence between epistemic causal theories.

We can show that all those notions of equivalence are the same actually. As a final result, we show that:

Theorem 35 *Two epistemic theories are strongly equivalent if and only if they are regularly equivalent.*

Just as in the propositional case, production relations are maximal inference relations that are adequate for reasoning with causal theories.

Exact theories are also fixed points, where beliefs are augmented with a hierarchy of causality relation, represented by the production relation. Lets take a look at how Abductive reasoning can be implemented in such framework.

3.3.2 Abductive epistemic inference

In this section, we will adapt Bochman's abductive reasoning to the epistemic case of ABC. Our goal is to propose a framework that models how an agent would associate abducibles to his beliefs.

We recall that an abductive framework is a pair $\mathbb{A} = (Cn, \mathcal{A})$ where Cn is a consequence relation and $\mathcal{A}$ is a set of abducibles. In our framework abducibles are couples of ABC formulas. We adopt the same definition of explanability by saying that a pair of formulas are explained if there is a set $a \subseteq \mathcal{A}$ such that $Cn(a)$ includes it.

We define an abductive epistemic production relation as a regular relation that satisfies the abduction principle.

Definition 26 .

- *A pair of formulas $\langle \varphi, \psi \rangle$ will be called an abducible in an epistemic production relation $\Rightarrow_i$, if $\varphi, \psi \Rightarrow_i \varphi, \psi$;*
- *An epistemic production relation will be called abductive epistemic if it is regular and satisfies:*

  (Abduction) *if $\varphi_1, \varphi_2 \Rightarrow_i \psi_1, \psi_2$, then $\varphi_1, \varphi_2 \Rightarrow_i \theta_1, \theta_2 \Rightarrow_i \psi_1, \psi_2$, for some abducible $\langle \theta_1, \theta_2 \rangle$.*

The abduction condition corresponds to the following restriction on the production operator:

$$\text{for any epistemic theory } \mathcal{C}_i(\langle \Gamma, \Psi \rangle) = \mathcal{C}_i(\langle \Gamma, \Psi \rangle \cup \mathcal{A})$$

where $\mathcal{A}$ is the set of abducibles of $\Rightarrow_i$.

Given an epistemic production relation $\Rightarrow_i$, we will define the following production relation:

$$\varphi_1, \varphi_2 \Rightarrow_i^a \psi_1, \psi_2 \equiv (\exists \langle \theta_1, \theta_2 \rangle)(\varphi_1, \varphi_2 \Rightarrow_i \theta_1, \theta_2 \Rightarrow_i \theta_1, \theta_2 \Rightarrow_i \psi_1, \psi_2)$$

Theorem 36 *if $\Rightarrow_i$ is a regular production relation, then $\Rightarrow_i^a$ is the greatest abductive production relation included in $\Rightarrow_i$.*

Epistemic abductive subrelations preserve some properties of the original relation, like sharing the same constraints, axioms and abducibles.

By accepting the explanatory closure assumption, we ensure that in cases where the used set of formulas is finite, we cannot derogate from using abducibles. This means that many regular relations coincide with their abductive subrelation.

Definition 27 *An epistemic production relation will be called* quasi-abductive *if it coincide with its abductive subrelation.*

Now a condition for quasi-abducibility

Definition 28 *A regular epistemic production relation will be called well-founded, if any infinite senquence $\{\langle \varphi_0, \psi_0 \rangle, \langle \varphi_1, \psi_1 \rangle, \langle \varphi_2, \psi_2 \rangle, \cdots\}$ of couples of formulas, such that $\varphi_{n+1}, \psi_{n+1} \Rightarrow_i \psi_n, \psi_n$, for any $n \geq 0$, contains an abducible.*

A regular epistemic production relation is finitary if it is a least regular production relation that contain some finite causal epistemic theories.

Theorem 37 *Any finitary regular production is well-founded*

And finally,

Theorem 38 *Any well-founded regular production relation is quasi-abductive.*

3.4 Nonmonotonic vision of trust

In the last chapter, we presented a formalization of trust as a special kind of belief about a context of interaction. Our ABC logic offers tools to test if an agent has this kind of belief, given his epistemic state. Nonetheless, by its operational nature, trust should be viewed as a special kind of belief, different in nature from an agent's other types of beliefs, for example, in many cases, trust is generated by heuristical interpretation of social norms. Such social norms make us believe that any agent wearing a police uniform would help us. This kind of belief should be distinguishable from belief that is directly acquired for example, like observing that the sky is blue.

In this section, we will use our abductive relation both to separate what we will call **trust belief** from **core belief**, and define a hierarchy of explanability, that allows us to associate to each belief the agent has, an origin from his core or trust belief.

In our view an agent's epistemic state will be divided into:

A core belief , that represents what an agent would assume about the actual interaction context. Such beliefs can be either acquired by observation or interaction,

A trust based belief or *trust belief*, is a special kind of belief, that represents the agent *trust assumptions*. Separating it from *core belief* allow us to manipulate such special kind of belief in a special way. It also emphasizes the fact that trust can be based on other metrics and assumptions than the core beliefs (like different social conventions,etc.), and

A production relation , that links core belief and trust based belief, in order to form the current epistemic state of the agent. The production relation will also provide us with a way to describe how to define a causal dependency between these different type of beliefs, in a way that represents how trust may be used (to implement abductive tasks).

The interaction between core and trust beliefs may differ from an agent to another. Depending on the use of the production relation, we identify three types of interactions:

Core based trust: The first vision sees core trust, as the origin of the trust belief, trust is then used to explain the agent's beliefs.

trust as elemental belief: Trust is an elemental type of belief, that explains any belief of the agent except his core belief, which is taken as self explanatory.

contextual trust: Trust assertions are actually used in parallel with the core knoweldge to justify beliefs.

In what will follow, we will present how we can use epistemic abductive relations to deploy these three kinds of epistemic state. Then we will present how we can integrate our production relations into ABC logic.

3.4.1 Core based trust

In chapter 2, we defined trust as a special kind of belief that is associated to the current state of belief of the truster. Our core based trust approach assumes that trust can explain all the formulas compatible with the agent's epistemic state. Trust assertions are seen as abducibles then. As for trust assertion, they are themselves explainable using the core belief of the agent.

We will denote by $\mathbb{C}_i \subseteq \mathcal{PM}$ the core beliefs of the agent i, and $\mathbb{T}_i \subseteq \mathcal{PM}$ the trust belief. We should note that a syntactic reformulation can be obtained by defining $\mathbb{C}_{si} = \{\langle \varphi, \psi \rangle \mid \mathbb{C}_i \models_{\mathsf{ABC}} \mathtt{Cert}_i\varphi \wedge \mathtt{Plaus}_i\psi\}$ and $\mathbb{T}_{si} = \{\langle \varphi, \psi \rangle \mid \mathbb{T}_i \models_{\mathsf{ABC}} \mathtt{Cert}_i\varphi \wedge \mathtt{Plaus}_i\psi\}$. This reformulation helps us define the set of formulas upon which our solution will define an abductive relation using the entailment operator of ABC logic, corresponding to the epistemic state of i: $\mathbb{C}_i \cup \mathbb{T}_i$, related to the syntactic reformulation $Th_i(\mathbb{C}_{si} \cup \mathbb{T}_{si})$.

Our abductive framework is then a regular production relation, defined on the set of formulas $Th_i(\mathbb{C}_{si} \cup \mathbb{T}_{si})$, such that we define two

sets of abducibles: $\mathcal{A}_T$ and $\mathcal{A}_C$ of resp. trust and core abducibles. We are interested in such abductive relation $\Rightarrow$that are refinement of Bochman's abduction constraint:

Abduction* if $\varphi_1, \varphi_2 \Rightarrow \psi_1, \psi_2$ then there is $\langle c_1, c_2 \rangle \in \mathcal{A}_C$ and $\langle t_1, t_2 \rangle \in \mathcal{A}_T$ such that: $\varphi_1, \varphi_2 \Rightarrow c_1, c_2 \Rightarrow t_1, t_2 \Rightarrow \psi_1, \psi_2$

This corresponds to the abductive framework presented in the last section, with the introduction of two sets of abducibles.

Given the fact that an abductive relation can be defined using a causal theory (a set of production relations), we can define a new epistemic state of an agent, that integrates an abduction mechanism based on core based trust as follows:

Definition 29 (Core based trust Epistemic state($\mathcal{E}_{CT}$)) *A core trust based epistemic state of an agent i, is a structure* $\langle \mathbb{T}, \mathbb{C}, \mathcal{A}_C, \mathcal{A}_T, \Delta \rangle$, *where:*

- $\mathbb{T} \subseteq \mathcal{PM}$ *are the trust beliefs of i.*
- $\mathbb{C} \subseteq \mathcal{PM}$ *the core belief of i.*
- Δ *is a set of production rules on formulas of* ABC *logic.*
- $\mathcal{A}_C$,$\mathcal{A}_T$*are sets of* ABC *formulas, such that* $\mathcal{A}_C \cup \mathcal{A}_T$ *is the set of abducibles of the abductive relation* $\Rightarrow^a_{i\Delta}$ *associated to* Δ *such that* $\Rightarrow^a_{i\Delta}$*satisfies* ***abduction****.

3.4.2 Trust as an elemental belief

In many cases, it is advantageous to see trust as a primitive artifact, that cannot be reduced, nor explained using other objects of an interaction theory. In this second reformulation of an agent's epistemic state, trust suffices to explain the agent's epistemic state assertions, except for the core belief, which is seen as self-explained.

Definition 30 *Elemental Epistemic state($\mathcal{E}_{EL}$)] the elemental epistemic state of an agent i, is a structure* $\langle \mathbb{T}, \mathbb{C}, \mathcal{A}_C, \mathcal{A}_T, \Delta \rangle$, *where:*

- $\mathbb{T} \subseteq \mathcal{PM}$ *are the trust beliefs of i.*

- $\mathbb{C} \subseteq \mathcal{PM}$ *are the core beliefs of i.*
- Δ *is a set of production rules on formulas of* ABC *logic.*
- $\mathcal{A}_C$,$\mathcal{A}_T$*are sets of* ABC *formulas, such that* $\mathcal{A}_C \cup \mathcal{A}_T$ *is the set of abducibles of the abductive relation* $\Rightarrow^a_{i\Delta}$ *associated to* Δ *satisfying:*

  if $\varphi_1, \varphi_2 \Rightarrow \psi_1, \psi_2$ *with* $\langle \psi_1, \psi_2 \rangle \in Th_i(\mathbb{C}_{si} \cup \mathbb{T}_{si}) \setminus \mathbb{C}_{si}$*, there is a* $\langle t_1, t_2 \rangle \in \mathcal{A}_T$ *such that:* $\varphi_1, \varphi_2 \Rightarrow t_1, t_2 \Rightarrow \psi_1, \psi_2$*, else, there is a* $\langle t_1, t_2 \rangle \in \mathcal{A}_T$ *such that:* $\varphi_1, \varphi_2 \Rightarrow c_1, c_2 \Rightarrow \psi_1, \psi_2$

Assuming a system where agents are based on elemental epistemic states, assumes that an agent will base his abductive framework upon trust. It also means that adding elements to core knowledge will expand to the epistemic state only in ways that are compatible with a trust-based explanation.

3.4.3 Contextual trust

A more admissible way to think about a trust based abductive framework, is to define the use of trust as contextual to the basic assumption about the current world. This differentiation between what is often called absolute and circumstantial trust can only be modeled if the current influence of the core belief, upon how trust is interpreted, is explicitly defined by the agent.

We will implement this in the form of families of parametrized abductive relations

The set of abducibles is $\mathcal{A} \subseteq \mathbb{C}_{si} \times \mathbb{T}_{si}$ where abducibles take the form $\langle c_1 \wedge t_1, c_2 \wedge t_2 \rangle$.

Definition 31 (contextual Epistemic state($\mathcal{E}_{CN}$)) *the elemental epistemic state of an agent i, is a structure* $\langle \mathbb{T}, \mathbb{C}, \mathcal{A}_C, \mathcal{A}_T, \Delta \rangle$*, where:*

- $\mathbb{T} \subseteq \mathcal{PM}$ *are the trust beliefs of i.*
- $\mathbb{C} \subseteq \mathcal{PM}$ *are the core beliefs of i.*
- Δ *a set of production rules defines upon the formulas of* ABC *logic.*

- $\mathcal{A}_C$,$\mathcal{A}_T$*are sets of* ABC *formulas, such that* $\mathcal{A}_C \times \mathcal{A}_T$ *is the set of abducibles of the abductive relation* $\Rightarrow^a_{i\Delta}$ *associated to* Δ *satisfying:*

  if $\langle \varphi_1, \varphi_2 \rangle \Rightarrow \langle \psi_1, \psi_2 \rangle$ *then There is* $\langle c_1, c_2 \rangle \in \mathcal{A}_C$ *and* $\langle t_1, t_2 \rangle \in \mathcal{A}_T$ *such that:* $\langle \varphi_1, \varphi_2 \rangle \Rightarrow \langle c_1 \wedge t_1, c_2 \wedge t_2 \rangle \Rightarrow \langle \psi_1, \psi_2 \rangle$

The main change that should be forced, is that the user needs to define his causal theory, using parametrized rules.

3.5 Discussion

In this chapter, we presented a nonmonotonic vision of trust that extends the trust belief principle presented in chapter 2. This nonmonotonic vision divides trust models into two elements, a set of atomic assumptions assimilated to abducibles in abductive frameworks, and a function that transforms beliefs according to those trust assumptions, which is a conditional and epistemic variant of production relations.

This separation provides a better understanding of the process of trust integration. Trust assumptions are objects of trust, while the related production relation is how the agent actually uses trust in his way of reasoning. This provide us with an interesting model of trust, that may be extended in future work.

In this investigation, we introduced a new way to model belief revision based on epistemic production relations. We can see that we can use this method in a framework to include knowledge of different kind and different sources, while separating how to integrate it. Still, priority between those different kind of belief need to be studied.

However, this approach to trust assumes that atomic trust assertions and the production relation are inputs of the model, those two objects characterize how an agent has integrated trust in his process of constructing a mental image of. One may see this process as a result of the agent experience and own effort to induce general rules about how to interact.

Chapter 4

Discussion and conclusion

In this work, we presented a general study of computational trust based on two main claims: trust studies need to be more methodical, which can only be possible by assuming an applicative perception of trust. When applied in complex situations, trust objects need to be explainable, or at least well described in a way that helps the truster construct his decision process upon it in a sound manner.

Following this two claim, we started by presenting a general survey of computational trust. We focused on finding different parallels between trust models in the literature , either by sharing a common trust theory, a field of application, or a theory of instantiation. Such parallels helped us defining a set of research tracks, that we deem important to tackle in order to help the discipline grow.

Due to the applicative incentive to study trust, most of the prominent models are based on ad hoc optimization. Such an approach to studying trust may give short term results, but in the long run, a methodology to study trust is needed in order to be able to construct upon past studies in a modular way. This can only be done by a modular approach to define the component of trust management systems, and defining testbeds that are both meaningful from an applicative point of view and not specific to a unique category of trust models.

We presented then two trust models, meant to represent trust in the

context of multi-agent systems based on cognitive agents. In our first model, we started by proposing a dynamic epistemic logic, with two belief operators and a choice operator, joined with two dynamic operators used to express action composition to represent the multi-agent system. Atomic actions are interpreted as authored assignments, which allows us to make a direct parallel to standard notions of computation like programs or services. We used this framework to present a trust definition that instantiates Castelfranchi&Falcone socio-cognitive trust theory, as a special kind of epistemic state of the truster about the capacity of the trustees to achieve some task, viewed as a composition of actions. After illustrating how trust in such composed action an be retrieved from trust in atomic actions, we illustrated the usage of our logic in a study case, related to service-oriented architectures, where composed actions are related to service composition.

Our second trust model presents in an attempt to extend our first model of trust, to support trust based abductive reasoning. To do so, we extended Bochman's production relations to work with an epistemic language. Since Bochman's framework was designed to study the notion of causality as a nonmonotonic reasoning concept, our definition of trust became a special case of causal reasoning, applied to social interactions. Our framework was then used to present our views on how trust can be used as abducibles, to explain the agents beliefs, or as a consequence of the agents observation and a priori knowledge about the system.

Possible extensions of our work can be of two nature: applicative and theoretical.

Applicative extension that we hope to pursue are:

Defining formal service-oriented architecture: our dynamic epistemic logic offers the necessary expressiveness that one can desire to implement SOA solutions. Our framework should be usable to express service composition, service orchestration, and the content of contract of service between service providers and service consumers. We also believe that our logic offers the necessary tools to implement protocols of interaction, that tackle both functional and security aspects of online interactions, as illusttrated in our

case study.

Defining personal information management systems: our trust model can be used as basis to implement a suggestion module, that a service provider can offer to a user that would interact in risky environment to help him assess threats related to his preserving his privacy and manage access to his personal informations. Such module can be of use in social networks, where suggestions on how and with whom to share information can be crucial to avoid cyber crime, like information theft, cyber fraud, identity usurpation, cyber bullying, etc.

Theoretical extension of our work that we would like to pursue are:

Studying the complexity of ABC logic: while we did prove the decidability of our logic, in oder to emphasize its usability to implement real solution, we need to prove the tractability of our logic. Such results will also further our understanding fo both dynamic epistemic logic class of complexity and the influence of assignment based action, in the tractability of complex dynamic logics.

Extending the set of action constructor in ABC logic: The main drawback of ABC logic, is the lack of a Kleene star, which allow to implement loops. New operators would help us describe more complex action composition. Our goal is to get closer to service composition languages that are already in use (like BPEL [62] for example).

Proofs of claims regarding our abductive framework: While the main claims of the last chapter are still conjecture, we intend to provide their formal proof in future works. We also intend to define a more natural interpretation of production relation within modal languages.

Bibliography

[1] Karl Aberer and Zoran Despotovic. Managing trust in a peer-2-peer information system. In *Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, USA, November 5-10, 2001*, pages 310–317, 2001.

[2] Thomas Ågotnes, Paul Harrenstein, Wiebe van der Hoek, and Michael Wooldridge. Boolean games with epistemic goals. In *Logic, Rationality, and Interaction - 4th International Workshop, LORI 2013, Hangzhou, China, October 9-12, 2013, Proceedings*, pages 1–14, 2013.

[3] Leila Amgoud and Henri Prade. Using arguments for making decisions: A possibilistic logic approach. In *UAI '04, Proceedings of the 20th Conference in Uncertainty in Artificial Intelligence, Banff, Canada, July 7-11, 2004*, pages 10–17, 2004.

[4] Ross J. Anderson. *Security engineering - a guide to building dependable distributed systems (2. ed.)*. Wiley, 2008.

[5] Philippe Balbiani, Andreas Herzig, and Nicolas Troquard. Dynamic logic of propositional assignments: A well-behaved variant of PDL. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*, pages 143–152, 2013.

[6] Philippe Balbiani, Hans van Ditmarsch, Andreas Herzig, and Tiago De Lima. Some truths are best left unsaid. In *Advances in*

Modal Logic 9, papers from the ninth conference on "Advances in Modal Logic," held in Copenhagen, Denmark, 22-25 August 2012, pages 36–54, 2012.

[7] Alexandru Batlag, Lawrence S. Moss, and Slawomir Solecki. The logic of public announcements and common knowledge and private suspicions. In *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-98), Evanston, IL, USA, July 22-24, 1998*, pages 43–56, 1998.

[8] Thomas Beth, Malte Borcherding, and Birgit Klein. Valuation of trust in open networks. In *Computer Security - ESORICS 94, Third European Symposium on Research in Computer Security, Brighton, UK, November 7-9, 1994, Proceedings*, pages 3–18, 1994.

[9] Alexander Bochman. A causal approach to nonmonotonic reasoning. *Artif. Intell.*, 160(1-2):105–143, 2004.

[10] Alexander Bochman. Production inference, nonmonotonicity and abduction. In *AI&M 1-2004, Eighth International Symposium on Artificial Intelligence and Mathematics, January 4-6, 2004, Fort Lauderdale, Florida, USA*, 2004.

[11] Alexander Bochman and Dov M. Gabbay. Causal dynamic inference. *Ann. Math. Artif. Intell.*, 66(1-4):231–256, 2012.

[12] Julien Bourdon, Guillaume Feuillade, Andreas Herzig, and Emiliano Lorini. Trust in complex actions. In *ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings*, pages 1037–1038, 2010.

[13] M. Bratman. *Intention, plans, and practical reason*. Harvard University Press, 1987.

[14] Christiano Castelfranchi and Rino Falcone. *Trust Theory: A Socio-Cognitive and Computational Model*. Wiley Publishing, 1st edition, 2010.

[15] Cristiano Castelfranchi and Rino Falcone. Principles of trust for MAS: cognitive anatomy, social importance, and quantification. In *Proceedings of the Third International Conference on Multiagent Systems, ICMAS 1998, Paris, France, July 3-7, 1998*, pages 72–79, 1998.

[16] Cristiano Castelfranchi and Yao-Hua Tan. The role of trust and deception in virtual societies. In *34th Annual Hawaii International Conference on System Sciences (HICSS-34), January 3-6, 2001, Maui, Hawaii, USA*, 2001.

[17] Philip R. Cohen and Hector J. Levesque. Intention is choice with commitment. *Artif. Intell.*, 42(2-3):213–261, 1990.

[18] Marcin Czenko, Jeroen Doumen, and Sandro Etalle. Trust management in P2P systems using standard tulip. In *Trust Management II - Proceedings of IFIPTM 2008: Joint iTrust and PST Conferences on Privacy, Trust Management and Security, June 18-20, 2008, Trondheim, Norway*, pages 1–16, 2008.

[19] Célia da Costa Pereira, Andrea G. B. Tettamanzi, and Serena Villata. A computational model of trust based on message content and source. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, pages 1849–1850, 2015.

[20] Zoran Despotovic and Karl Aberer. A probabilistic approach to predict peers? performance in P2P networks. In *Cooperative Information Agents VIII, 8th International Workshop, CIA 2004, Erfurt, Germany, September 27-29, 2004, Proceedings*, pages 62–76, 2004.

[21] Zoran Despotovic and Karl Aberer. P2P reputation management: Probabilistic estimation vs. social networks. *Computer Networks*, 50(4):485–500, 2006.

[22] Morton Deutsch. Cooperation and trust: Some theoretical notes. pages 275–319, 1962.

[23] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic.* Springer Publishing Company, Incorporated, 1st edition, 2007.

[24] Rino Falcone and Cristiano Castelfranchi. Social trust: A cognitive approach. In Cristiano Castelfranchi and Yao-Hua Tan, editors, *Trust and Deception in Virtual Societies*, pages 55–90. Springer Netherlands, 2001.

[25] Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *Int. J. High Perform. Comput. Appl.*, 15(3):200–222, August 2001.

[26] Karen Fullam and K. Suzanne Barber. Learning trust strategies in reputation exchange networks. In *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006*, pages 1241–1248, 2006.

[27] Karen Fullam, Tomas B. Klos, Guillaume Muller, Jordi Sabater-Mir, Zvi Topol, K. Suzanne Barber, Jeffrey S. Rosenschein, and Laurent Vercouter. The agent reputation and trust (ART) testbed architecture. In *Artificial Intelligence Research and Development, Proceedings of the 8th International Conference of the ACIA, CCIA 2005, October 26-28, 2005, Alguer, Italy*, pages 389–396, 2005.

[28] Dov M. Gabbay. Theoretical foundations for non-monotonic reasoning in expert systems. In Krysztof R. Apt, editor, *Logics and Models of Concurrent Systems*, pages 439–457. 1985.

[29] Enrico Giunchiglia, Joohyung Lee, Vladimir Lifschitz, Norman McCain, and Hudson Turner. Nonmonotonic causal theories. *Artif. Intell.*, 153(1-2):49–104, 2004.

[30] Dawn G. Gregg and Judy E. Scott. The role of reputation systems in reducing on-line auction fraud. *Int. J. Electron. Commerce*, 10(3):95–120, April 2006.

[31] Chung-Wei Hang, Yonghong Wang, and Munindar P. Singh. An adaptive probabilistic trust model and its evaluation. In *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, May 12-16, 2008, Volume 3*, pages 1485–1488, 2008.

[32] David Harel, Dexter Kozen, and Jerzy Tiuryn. *Dynamic Logic*. MIT Press, 2000.

[33] Andreas Herzig, Emiliano Lorini, Jomi Fred Hübner, Jonathan Ben-Naim, Cristiano Castelfranchi, Robert Demolombe, Dominique Longin, and Laurent Vercouter. Prolegomena for a logic of trust and reputation. In *Third International Workshop on Normative Multiagent Systems - NorMAS 2008, Luxembourg, July 15-16, 2008. Proceedings*, pages 143–157, 2008.

[34] Andreas Herzig, Emiliano Lorini, Jomi Fred Hübner, and Laurent Vercouter. A logic of trust and reputation. *Logic Journal of the IGPL*, 18(1):214–244, 2010.

[35] Andreas Herzig, Emiliano Lorini, Frédéric Moisan, and Nicolas Troquard. A dynamic logic of normative systems. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 228–233, 2011.

[36] Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph. *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC, 2009.

[37] Andrew J. I. Jones. On the concept of trust. *Decision Support Systems*, 33(3):225–232, 2002.

[38] Andrew J.I. Jones and BabakSadighi Firozabadi. On the characterisation of a trusting agent. aspects of a formal approach. In Cristiano Castelfranchi and Yao-Hua Tan, editors, *Trust and Deception in Virtual Societies*, pages 157–168. Springer Netherlands, 2001.

[39] Audun Jøsang. Trust and reputation systems. In *Foundations of Security Analysis and Design IV, FOSAD 2006/2007 Tutorial Lectures*, pages 209–245, 2007.

[40] Audun Jøsang. *Subjective logic*. 2013.

[41] Audun Jøsang and Jochen Haller. Dirichlet reputation systems. In *Proceedings of the The Second International Conference on Availability, Reliability and Security, ARES 2007, The International Dependability Conference - Bridging Theory and Practice, April 10-13 2007, Vienna, Austria*, pages 112–119, 2007.

[42] Audun Jøsang, Ross Hayward, and Simon Pope. Trust network analysis with subjective logic. In *Computer Science 2006, Twenty-Nineth Australasian Computer Science Conference (ACSC2006), Hobart, Tasmania, Australia, January 16-19 2006*, pages 85–94, 2006.

[43] Audun Jøsang and Stéphane Lo Presti. Analysing the relationship between risk and trust. In *Trust Management, Second International Conference, iTrust 2004, Oxford, UK, March 29 - April 1, 2004, Proceedings*, pages 135–145, 2004.

[44] Karl Krukow, Mogens Nielsen, and Vladimiro Sassone. A logical framework for history-based access control and reputation systems. *Journal of Computer Security*, 16(1):63–101, 2008.

[45] John Krumm. *Ubiquitous Computing Fundamentals*. Chapman & Hall/CRC, 1st edition, 2009.

[46] Niklas Luhmann. *Essays on self-reference*. Columbia University Press, 1990.

[47] D. Makinson and L. Van Der Torre. Input/output logics. *Journal of Philosophical Logic*, 29(4):383–408, 2000.

[48] Stephen Marsh and Pamela Briggs. Examining trust, forgiveness and regret as computational concepts. In *Computing with Social Trust*, pages 9–43. 2009.

[49] Stephen Paul Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, April 1994.

[50] E. Michael Maximilien. Multiagent system for dynamic web services selection. In *In Proceedings of 1st Workshop on Service-Oriented Computing and Agent-Based Engineering (SOCABE at AAMAS*, pages 25–29, 2005.

[51] Norman McCain and Hudson Turner. Causal theories of action and change. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference, AAAI 97, IAAI 97, July 27-31, 1997, Providence, Rhode Island.*, pages 460–465, 1997.

[52] Dimitri Melaye and Yves Demazeau. Bayesian dynamic trust model. In *Multi-Agent Systems and Applications IV, 4th International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2005, Budapest, Hungary, September 15-17, 2005, Proceedings*, pages 480–489, 2005.

[53] Gia Hien Nguyen, Philippe Chatalic, and Marie-Christine Rousset. A probabilistic trust model for semantic peer-to-peer systems. In *ECAI 2008 - 18th European Conference on Artificial Intelligence, Patras, Greece, July 21-25, 2008, Proceedings*, pages 881–882, 2008.

[54] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[55] Judea Pearl. Embracing causality in formal reasoning. In Kenneth D. Forbus and Howard E. Shrobe, editors, *AAAI*, pages 369–373. Morgan Kaufmann, 1987.

[56] Isaac Pinyol and Jordi Sabater-Mir. Arguing about reputation: The lrep language. In *Engineering Societies in the Agents World VIII, 8th International Workshop, ESAW 2007, Athens, Greece,*

October 22-24, 2007, Revised Selected Papers, pages 284–299, 2007.

[57] Isaac Pinyol, Jordi Sabater-Mir, and Guifré Cuní. How to talk about reputation using a common ontology: From definition to implementation. pages 90—101, 2007.

[58] William Poundstone. *Prisoner's Dilemma: John von Neuman, Game Theory, and the Puzzle of the Bomb*. Doubleday, New York, 1992.

[59] Anand S. Rao and Michael P. Georgeff. BDI agents: From theory to practice. In *Proceedings of the First International Conference on Multiagent Systems, June 12-14, 1995, San Francisco, California, USA*, pages 312–319, 1995.

[60] Matthew Richardson, Rakesh Agrawal, and Pedro M. Domingos. Trust management for the semantic web. In *The Semantic Web - ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23, 2003, Proceedings*, pages 351–368, 2003.

[61] Enders A Robinson. *Statistical reasoning and decision making*. Goose Pond Press, 1981.

[62] Florian Rosenberg and Schahram Dustdar. Business rules integration in BPEL - A service-oriented approach. In *7th IEEE International Conference on E-Commerce Technology (CEC 2005), 19-22 July 2005, München, Germany*, pages 476–479, 2005.

[63] Jordi Sabater and Carles Sierra. Review on computational trust and reputation models. *Artif. Intell. Rev.*, 24(1):33–60, 2005.

[64] Leonard J. Savage. Difficulties in the theory of personal probability. *Philosophy of Science*, 34(4):pp. 305–310, 1967.

[65] Xuemin Shen, Heather Yu, John Buford, and Mursalin Akon. *Handbook of Peer-to-Peer Networking*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[66] Wang Shou-xin, Zhang Li, and Wang Shuai. A measurement approach of trust relation in web service. *Journal of Communication and Computer*, 6(8):9–17, 2009.

[67] Carles Sierra and John Debenham. An information-based model for trust. *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems - AAMAS '05*, page 497, 2005.

[68] George Spanoudakis and Stephane LoPresti. Web service trust: Towards a dynamic assessment framework. In *Proceedings of the The Forth International Conference on Availability, Reliability and Security, ARES 2009, March 16-19, 2009, Fukuoka, Japan*, pages 33–40, 2009.

[69] Ruben Stranders, Mathijs de Weerdt, and Cees Witteveen. Fuzzy argumentation for trust. In *Computational Logic in Multi-Agent Systems, 8th International Workshop, CLIMA VIII, Porto, Portugal, September 10-11, 2007. Revised Selected and Invited Papers*, pages 214–230, 2007.

[70] George Theodorakopoulos and John S. Baras. On trust models and trust evaluation metrics for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(2):318–328, 2006.

[71] J. van Benthem. *The Logic of Time: A Model-Theoretic Investigation into the Varieties of Temporal Ontology and Temporal Discourse*. Synthese Library. Springer Netherlands, 1991.

[72] Hans van Ditmarsch and Tim French. Semantics for knowledge and change of awareness. *Journal of Logic, Language and Information*, 23(2):169–195, 2014.

[73] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Pieter Kooi. *Dynamic epistemic logic*, volume 337. Springer Science & Business Media, 2007.

[74] Jan Van Eijck and Yanjing Wang. Propositional dynamic logic as a logic of belief revision. In *Logic, language, information and computation*, pages 136–148. Springer, 2008.

[75] Kaiyu Wan and Vasu S. Alagar. An intensional functional model of trust. In *Trust Management II - Proceedings of IFIPTM 2008: Joint iTrust and PST Conferences on Privacy, Trust Management and Security, June 18-20, 2008, Trondheim, Norway*, pages 69–85, 2008.

[76] Li Xiong and Ling Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Trans. Knowl. Data Eng.*, 16(7):843–857, 2004.

[77] Bin Yu and Munindar P. Singh. A social mechanism of reputation management in electronic communities. In *Cooperative Information Agents IV, The Future of Information Agents in Cyberspace, 4th International Workshop, CIA 2000, Boston, MA, USA, July 7-9, 2000, Proceedings*, pages 154–165, 2000.