Reasoning about Cognitive Trust in Stochastic Multiagent Systems

XIAOWEI HUANG, University of Oxford, UK

MARTA KWIATKOWSKA, University of Oxford, UK

MACIEJ OLEJNIK, University of Oxford, UK

We consider the setting of stochastic multiagent systems modelled as stochastic multiplayer games and formulate an automated verification framework for quantifying and reasoning about agents' trust. To capture human trust, we work with a cognitive notion of trust defined as a subjective evaluation that agent A makes about agent B's ability to complete a task, which in turn may lead to a decision by A to rely on B. We propose a probabilistic rational temporal logic PRTL*, which extends the probabilistic computation tree logic PCTL* with reasoning about mental attitudes (beliefs, goals and intentions), and includes novel operators that can express concepts of social trust such as competence, disposition and dependence. The logic can express, for example, that "agent A will eventually trust agent B with probability at least p that B will behave in a way that ensures the successful completion of a given task". We study the complexity of the automated verification problem and, while the general problem is undecidable, we identify restrictions on the logic and the system that result in decidable, or even tractable, subproblems.

CCS Concepts: • **Theory of computation** → **Modal and temporal logics**; *Probabilistic computation*; *Verification by model checking*.

Additional Key Words and Phrases: multi-agent systems, stochastic games, cognitive trust, quantitative reasoning, probabilistic temporal logic

1 INTRODUCTION

Mobile autonomous robots are rapidly entering the fabric of our society, to mention driverless cars and home assistive robots. Since robots are expected to work with or alongside humans in our society, they need to form partnerships with humans, as well as other robots, understand the social context, and behave, and be seen to behave, according to the norms of that context. Human partnerships such as cooperation are based on *trust*, which is influenced by a range of subjective factors that include subjective preferences and experience. As the degree of autonomy of mobile robots increases and the nature of partnerships becomes more complex, to mention shared autonomy, understanding and reasoning about social trust and the role it plays in decisions whether to rely on autonomous systems is of paramount importance. A pertinent example is the recent

Authors' addresses: Xiaowei Huang, University of Oxford, Department of Computer Science, Oxford, UK, xiaowei.huang@live.com; Marta Kwiatkowska, University of Oxford, Department of Computer Science, Oxford, UK, marta.kwiatkowska@cs.ox.ac.uk; Maciej Olejnik, University of Oxford, Department of Computer Science, Oxford, UK, maciej.olejnik@cs.ox.ac.uk.

Tesla fatal car accident while on autopilot mode [37], which is a result of over-reliance ("overtrust") by the driver, likely influenced through his personal motivation and preferences.

Trust is a complex notion, viewed as a belief, attitude, intention or behaviour, and is most generally understood as *a subjective evaluation of a truster on a trustee about something in particular*, e.g., the completion of a task [24]. A classical definition from organisation theory [39] defines trust as *the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that party*. The importance of being able to correctly evaluate and calibrate trust to guide reliance on automation was recognised in [38]. Trust (and trustworthiness) have also been actively studied in many application contexts such as security [30] and e-commerce [14]. However, in this paper we are interested in trust that governs social relationships between humans and autonomous systems, and to this end consider *cognitive trust* that captures the human notion of trust. By understanding how human trust in an autonomous system evolves, and being able to quantify it and reason about it, we can offer guidance for selecting an appropriate level of reliance on autonomy.

The goal of this paper is therefore to develop foundations for *automated, quantitative reasoning* about cognitive trust between (human and robotic) agents, which can be employed to support decision making in dynamic, uncertain environments. The underlying model is that of *multiagent systems*, where agents are autonomous and endowed with individual goals and preferences in the style of BDI logic [44]. To capture uncertainty, we work in the setting of *stochastic* multiagent systems, represented concretely in terms of *concurrent stochastic multiplayer games*, where stochasticity can be used to model, e.g., component failure or environmental uncertainty. This also allows us to represent agent *beliefs* probabilistically. Inspired by the concepts of social trust in [17], we formulate a probabilistic rational temporal logic PRTL* as an extension of the probabilistic temporal logic PCTL* [23] with *cognitive* aspects. PCTL* allows one to express *temporal* properties pertaining to system execution, for example "with probability p, agent A will eventually complete a given task". PRTL* includes, in addition, mental attitude operators (belief, goal, intention and capability), together with a collection of novel *trust* operators (competence, disposition and dependence), in turn expressed using beliefs. The logic PRTL* is able to express properties such as "agent A will eventually trust agent B with probability at least p that B will behave in a way that ensures the successful completion of a given task" that are informally defined in [17]. PRTL* is interpreted over a stochastic multiagent system, where the cognitive reasoning processes for each agent can be modelled based on a *cognitive mechanism* that describes his/her mental state (a set of goals and an intention, referred to as pro-attitudes) and subjective preferences.

Since we wish to model dynamic evolution of beliefs and trust, the mechanisms are history-dependent, and thus the underlying semantics is an infinite branching structure, resulting in undecidability of the general model checking problem for PRTL*. In addition, there are two types of nondeterministic choices available to the agents, those made along the *temporal* or the *cognitive* dimension. By convention, the temporal nondeterminism is resolved using the concept of adversaries and quantifying over them to obtain a fully probabilistic system [22], as is usual for models combining probability and nondeterminism. We use a similar approach for the cognitive dimension, instead of the classical accessibility relation employed in logics for agency, and resolve cognitive nondeterminism by *preference functions*, given as probability distributions that model subjective knowledge about other agents. Also, in contrast to defining beliefs in terms of knowledge [16] and probabilistic knowledge [22] operators, which are based solely on agents' (partial) observations, we additionally allow agents' cognitive changes and subjective preferences to influence their belief.

This paper makes the following original contributions.

- We introduce autonomous stochastic multiagent systems as an extension of stochastic multi-player games with a cognitive mechanism for each agent (a set of goals and intentions).
- We provide a mechanism for reasoning about agent's cognitive states based on preference functions that enables a sound formulation of probabilistic beliefs.
- We formalise a collection of trust operators (competence, disposition and dependence) informally introduced in [17] in terms of probabilistic beliefs.
- We formulate a novel probabilistic rational temporal logic PRTL* that extends the logic PCTL* [23] with mental attitude and trust operators.
- We study the complexity of the automated verification problem for PRTL* and, while the general problem is undecidable, we identify restrictions on the logic and the system that result in decidable, or even tractable, subproblems.

The structure of the paper is as follows. Section 2 gives an overview of related work, and in Section 3 we discuss the concept of cognitive trust. Section 4 presents stochastic multiplayer games and strategic reasoning on them. Section 5 introduces autonomous stochastic multiagent systems, an extension of stochastic multiplayer games with a cognitive mechanism. Section 6 introduces preference functions and derives the semantics of the subjective, probabilistic belief operator. Section 7 defines how beliefs vary with respect to agents' observations. In Section 8 we define trust operators and the logic PRTL*. We consider the interactions of beliefs and pro-attitudes in Section 9 via pro-attitude synthesis. Section 10 gives the undecidable complexity result for the general PRTL* model checking problem and Section 11 presents several decidable logic fragments. We conclude the paper in Section 12.

A preliminary version of this work appeared as [28]. This extended version includes detailed derivations of the concepts, illustrative examples and full proofs of the complexity results omitted from [28].

2 RELATED WORK

The notion of trust has been widely studied in management, psychology, philosophy and economics (see [36] for an overview). Recently, the importance of trust in human-robot cooperation was highlighted in [34]. Trust in the context of human-technology relationships can be roughly classified into three categories: *credentials-based*, *experience-based*, and *cognitive trust*. Credentials-based trust is used mainly in security, where a user must supply credentials in order to gain access. Experience-based trust, which includes reputation-based trust in peer-to-peer and e-commerce applications, involves online evaluation of a trust value for an agent informed by experiences of interaction with that agent. A formal foundation for quantitative reputation-based trust has been proposed in [32]. In contrast, we focus on (quantitative) cognitive trust, which captures the social (human) notion of trust and, in particular, trust-based decisions between humans and robots. The cognitive theory of [17], itself founded on organisational trust of [39], provides an intuitive definition of complex trust notions but lacks rigorous semantics. Several papers, e.g., [25, 26, 29, 41], have formalised the theory of [17] using modal logic, but none are quantitative and automatic verification is not considered. Of relevance are recent approaches [47, 48] that model the evolution of trust in human-robot interactions as a dynamical system; instead, our formalism supports evolution of trust through events and agent interactions.

A number of logic frameworks have been proposed that develop the theory of human decisions [6] for artificial agents, see [40] for a recent overview. The main focus has been on studying the relationships between modalities with various axiomatic systems, but their amenability to automatic verification is arguable because of a complex underlying possible world semantics, to mention the

sub-tree relation of BDI logic [44]. The only attempt at model checking such logics [46] ignores the internal structure of the possible worlds to enable a reduction to temporal logic model checking.

The distinctive aspects of our work is thus a *quantitative* formalisation of *cognitive* trust in terms of *probabilistic* temporal logic, based on a probabilistic notion of *belief*, together with algorithmic complexity of the corresponding *model checking* problem.

3 COGNITIVE THEORY OF SOCIAL TRUST

In the context of automation, trust is understood as delegation of responsibility for actions to the autonomous system and willingness to accept risk (possible harm) and uncertainty. The decision to delegate is based on a *subjective* evaluation of the system's capabilities for a particular task, informed by factors such as past experience, social norms and individual preferences. Moreover, trust is a dynamic concept which evolves over time, influenced by events and past experience. The cognitive processes underpinning trust are captured in the influential theory of social trust by [17], which is particularly appropriate for human-robot relationships and serves as an inspiration for this work.

The theory of [17] views trust as a complex *mental attitude* that is relative to a set of goals and expressed in terms of *beliefs*, which in turn influence decisions about agent's future behaviour. They consider agent A's trust in agent B for a specific goal ψ (goals may be divided into tasks), and distinguish the following core concepts: *competence* trust, where A believes that B is able to perform ψ, and *disposition* trust, where A believes that B is willing to perform ψ. The decision to delegate or rely on B involves a complex notion of trust called *dependence*: A believes that B needs, depends, or is at least better off to rely on B to achieve ψ, which has two forms, *strong* (A needs or depends on B) and *weak* (for A, it is better to rely than not to rely on B). [17] also identify *fulfilment* belief arising in the truster's mental state, which we do not consider.

We therefore work in a stochastic setting (to represent uncertainty), aiming to *quantify* belief probabilistically and express trust as a *subjective, belief-weighted expectation*, informally understood as a degree of trust.

4 STOCHASTIC MULTIAGENT SYSTEMS AND TEMPORAL REASONING

A multiagent system $\mathcal{M}$ comprises a set of agents (humans, robots, components, processes, etc.) running in an environment [16]. To capture random events such as failure and environmental uncertainty that are known to influence trust, we work with *stochastic* multiagent systems, concretely represented using *concurrent stochastic multiplayer games*, where each agent corresponds to a player. Taking such models as a starting point, in this section we gradually extend them to autonomous multiagent systems that support cognitive reasoning.

4.1 Stochastic Multiplayer Games

Given a finite set S, we denote by $\mathcal{D}(S)$ the set of probability distributions on S and by $\mathcal{P}(S)$ the power set of S. Given a probability distribution δ over a set S, we denote by $supp(\delta)$ the support of δ, i.e., the set of elements of S which have positive probability. We call δ a Dirac distribution if $\delta(s) = 1$ for some $s \in S$.

We now introduce concurrent stochastic games, in which several players repeatedly make choices simultaneously to determine the next state of the game.

Definition 4.1. A *stochastic multiplayer game (SMG)* is a tuple $\mathcal{M} = (Ags, S, S_{\text{init}}, \{Act_A\}_{A \in Ags}, T, L)$, where:

- $Ags = \{1, ..., n\}$ is a finite set of players called agents, ranged over by A, B, ...
- S is a finite set of states,

- $S_{\mathrm{init}} \subseteq S$ is a set of initial states,
- Act_A is a finite set of actions for the agent A,
- $T : S \times Act \to \mathcal{D}(S)$ is a (partial) probabilistic transition function, where $Act = \times_{A \in Ags} Act_A$,
- $L : S \to \mathcal{P}(AP)$ is a labelling function mapping each state to a set of atomic propositions taken from a set AP.

We assume that each (global) state $s \in S$ of the system includes a local state of each agent and an (optional) environment state. In every state of the game, each player $A \in Ags$ selects a local action $a_A \in Act_A$ independently and the next state of the game is chosen by the environment according to the probability distribution $T(s, a)$ where $a \in Act$ is the joint action. In other words, the probability of transitioning from state s to state s' when action a is taken is $T(s, a)(s')$. In this paper we will usually omit the environment part of the state.

In concurrent games players perform their local actions simultaneously. *Turn-based* games are a restricted class of SMGs whose states are partitioned into subsets, each of which is controlled by a single agent, meaning that only that agent can perform an action in any state in the partition. Turn-based games can be simulated by concurrent games by requiring that, in states controlled by agent A, A performs an action $a_A \in Act_A$ and the other agents perform a distinguished silent action $\perp$.

Let a_A denote agent A's action in the joint action $a \in Act$. We let $Act(s) = \{a \in Act \mid T(s, a) \text{ is defined}\}$ be the set of *valid* joint actions in state s, and $Act_A(s) = \{a_A \mid a \in Act(s)\}$ be the set of valid actions in state s for agent A. T is called *serial* (or *total*) if, for any state s and joint action $a \in Act$, $T(s, a)$ is defined. We often write $s \longrightarrow_T^a s'$ for a transition from s to s' via action a, provided that $T(s, a)(s') > 0$.

Paths. A *path* ρ is a finite or infinite sequence of states $s_0 s_1 s_2 \dots$ induced from the transition probability function T, i.e., satisfying $T(s_k, a)(s_{k+1}) > 0$ for some $a \in Act$, for all $k \geq 0$. Paths generated by T are viewed as occurring in the *temporal* dimension. We denote the set of finite (resp. infinite) paths of $\mathcal{M}$ starting in s by $\mathrm{FPath}^{\mathcal{M}}(s)$ (resp. $\mathrm{IPath}^{\mathcal{M}}(s)$), and the set of paths starting from any state by $\mathrm{FPath}^{\mathcal{M}}$ (resp. $\mathrm{IPath}^{\mathcal{M}}$). We may omit $\mathcal{M}$ if clear from the context. For any path ρ we write $\rho(k)$ for its $(k + 1)$-th state, $\rho[0..n]$ for the prefix $s_0 \dots s_n$, and $\rho[n..\infty]$ for the suffix $s_n s_{n+1} \dots$ when ρ is infinite. If ρ is finite then we write $last(\rho)$ for its last state and $|\rho|$ for its *length*, i.e., the number of states in ρ. Given two paths $\rho = s_0 \dots s_n$ and $\rho' = s'_0 \dots s'_m$, we write $\rho \cdot \rho' = s_0 \dots s_n s'_1 \dots s'_m$ when $s_n = s'_0$ for their concatenation by an overlapping state, and $\rho\rho' = s_0 \dots s_n s'_0 s'_1 \dots s'_m$ for the regular concatenation.

Strategies. A (history-dependent and stochastic) *action strategy* σ_A of agent $A \in Ags$ in an SMG $\mathcal{M}$ is a function $\sigma_A : \mathrm{FPath}^{\mathcal{M}} \to \mathcal{D}(Act_A)$, such that for all $a_A \in Act_A$ and finite paths ρ it holds that $\sigma_A(\rho)(a_A) > 0$ only if $a_A \in Act_A(last(\rho))$. We call a strategy σ *pure* if $\sigma(\rho)$ is a Dirac distribution for any $\rho \in \mathrm{FPath}^{\mathcal{M}}$. A *strategy profile* σ_D for a set D of agents is a vector of action strategies $\times_{A \in D} \sigma_A$, one for each agent $A \in D$. We let Σ_A be the set of agent A's strategies, Σ_D be the set of strategy profiles for the set of agents D, and Σ be the set of strategy profiles for all agents.

Probability space. In order to reason formally about a given SMG $\mathcal{M}$ we need to quantify the probabilities of different paths being taken. We therefore define a probability space over the set of infinite paths $\mathrm{IPath}^{\mathcal{M}}(s_0)$ starting in a given state $s_0 \in S$, adapting the standard construction from [31]. Our probability measure is based on the function assigning probability to a given finite path $\rho = s_0 \dots s_n$ under strategy $\sigma \in \Sigma$, defined as $\mathrm{Pr}_\sigma(\rho) = \prod_{i=0}^{n-1} \sum_{a \in Act} \sigma(\rho[0..k])(a) \cdot T(s_k, a)(s_{k+1})$. To define measurable sets, for a path ρ we let Cyl_ρ be a basic cylinder, which is a set of all infinite paths starting with ρ. We then set $\mathcal{F}_s^{\mathcal{M}}$ to be the smallest σ-algebra generated by the

Table 1. Payoff of a simple trust game

	share	keep
invest	(20, 20)	(0, 40)
withhold	(10, 5)	(10, 5)

Table 2. Strategies for Alice and Bob

Strategy	*withhold*	*invest*	*keep*	*share*
$\sigma_{passive}$	0.7	0.3		
σ_{active}	0.1	0.9		
σ_{share}			0.0	1.0
σ_{keep}			1.0	0.0

basic cylinders $\{\text{Cyl}_\rho \mid \rho \in \text{FPath}^{\mathcal{M}}(s)\}$ and $\text{Pr}_\sigma^{\mathcal{M}}$ to be the unique measure on the set of infinite paths $\text{IPath}^{\mathcal{M}}(s)$ such that $\text{Pr}_\sigma^{\mathcal{M}}(\text{Cyl}_\rho) = \text{Pr}_\sigma(\rho)$. It then follows that $(\text{IPath}^{\mathcal{M}}(s), \mathcal{F}_s^{\mathcal{M}}, \text{Pr}_\sigma^{\mathcal{M}})$ is a probability space [31].

Example 4.2. We consider a simple (one shot) trust game from [33], in which there are two agents, Alice and Bob. At the beginning, Alice has 10 dollars and Bob has 5 dollars. If Alice does nothing, then everyone keeps what they have. If Alice invests her money with Bob, then Bob can turn the 15 dollars into 40 dollars. After having the investment yield, Bob can decide whether to share the 40 dollars with Alice. If so, each will have 20 dollars. Otherwise, Alice will lose her money and Bob gets 40 dollars.

For the simple trust game, the payoffs of the agents are shown in Table 1. The game has a Nash equilibrium of Alice withholding her money and Bob keeping the investment yield. This equilibrium discourages collaboration between agents and has not been confirmed empirically under the standard economic assumptions of pure self-interest [5].

To illustrate our methods, we construct a stochastic multiplayer game $\mathcal{G}$ with $Ags = \{Alice, Bob\}$, $S = \{s_0, s_1, ..., s_4\}$ with s_0 being the initial state, $Act_{Alice} = \{invest, withhold, \perp\}$, $Act_{Bob} = \{share, keep, \perp\}$ and the transition function defined in an obvious way (see Figure 1). Note that we do not represent payoffs explicitly in our modelling of the trust game, but rather capture them using atomic propositions. For example, $richer_{Alice,Bob}$ is true is state s_1, while $richer_{Bob,Alice}$ holds in s_3. Note also that *Alice* and *Bob* proceed in turns, which is captured through joint actions where the other agent takes the silent action $\perp$. We represent states as pairs:

$$(a_{Alice}, a_{Bob}),$$

where $a_{Alice} \in Act_{Alice}$ is Alice's last action and $a_{Bob} \in Act_{Bob}$ is Bob's last action. For example, $s_0 = (\perp, \perp)$, $s_2 = (invest, \perp)$ and $s_4 = (\perp, share)$.

We now equip agents with strategies. For Alice, we define σ_{active} and $\sigma_{passive}$, where the former corresponds to high likelihood of Alice investing her money and the latter allocates greater probability to withholding it. For Bob, we set two pure strategies σ_{keep} and σ_{share}, corresponding to him keeping and sharing the money with Alice. The strategies are summarised in Table 2. □
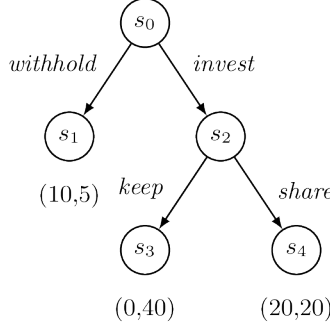
Fig. 1. Simple trust game

4.2 Temporal Reasoning about SMGs

We now recall the syntax of the Probabilistic Computation Tree Logic PCTL* [3, 4][1] for reasoning about temporal properties in systems exhibiting nondeterministic and probabilistic choices. PCTL* is based on CTL* [13] for purely nondeterministic systems and retains its expressive power, additionally extending it with a probabilistic operator $\mathsf{P}^{\bowtie q}\psi$.

Definition 4.3. The syntax of the logic PCTL* is as follows.

$$
\begin{aligned}
\phi &\quad ::= \quad p \mid \neg\phi \mid \phi \vee \phi \mid \forall\psi \mid \mathsf{P}^{\bowtie q}\psi \\
\psi &\quad ::= \quad \phi \mid \neg\psi \mid \psi \vee \psi \mid \bigcirc\psi \mid \psi\mathcal{U}\psi
\end{aligned}
$$

where p is an atomic proposition, $\bowtie \in \{<, \leq, >, \geq\}$, and $q \in [0, 1]$.

In the above, ϕ is a PCTL* (state) formula and ψ an LTL (path) formula. The operator $\forall$ is the (universal) path quantifier of CTL* and $\mathsf{P}^{\bowtie q}\psi$ is the probabilistic operator of PCTL [23], which expresses that ψ holds with probability in relation $\bowtie$ with q. The remaining operators, $\bigcirc\psi$ (next) and $\psi\mathcal{U}\psi$ (until) follow their usual meaning from PCTL and CTL*. The derived operators such as $\phi_1 \wedge \phi_2$, $\Diamond\psi$ (eventually), $\Box\psi$ (globally), $\psi\mathcal{R}\psi$ (release) and $\exists\phi$ (existential path quantifier) can be obtained in the standard way.

Let $\mathcal{M}$ be an SMG. Given a path ρs which has s as its last state, a strategy $\sigma \in \Sigma$ of $\mathcal{M}$, and a formula ψ, we write:

$$
Prob_{\mathcal{M}, \sigma, \rho s}(\psi) \stackrel{\text{def}}{=} \Pr_{\sigma}^{\mathcal{M}}\{\delta \in \text{IPath}_T^{\mathcal{M}}(s) \mid \mathcal{M}, \rho s, \delta \models \psi\}
$$

for the probability of implementing ψ on a path ρs when a strategy σ applies. The relation $\mathcal{M}, \rho, \delta \models \psi$ is defined below. Based on this, we define:

$$
Prob_{\mathcal{M}, \rho}^{min}(\psi) \stackrel{\text{def}}{=} \inf_{\sigma \in \Sigma} Prob_{\mathcal{M}, \sigma, \rho}(\psi),
$$

$$
Prob_{\mathcal{M}, \rho}^{max}(\psi) \stackrel{\text{def}}{=} \sup_{\sigma \in \Sigma} Prob_{\mathcal{M}, \sigma, \rho}(\psi)
$$

as the minimum and maximum probabilities of implementing ψ on a path ρ over all strategies in Σ.

We now give semantics of the logic PCTL* for concurrent stochastic games.

Definition 4.4. Let $\mathcal{M} = (Ags, S, S_{\text{init}}, \{Act_A\}_{A \in Ags}, T, L)$ be an SMG and $\rho \in \text{FPath}_T^{\mathcal{M}}$. The satisfaction relation $\models$ of PCTL* is defined inductively by:

---

[1]Note that we do not consider here the coalition operator, and therefore the logic rPATL* [12] that is commonly defined over stochastic game models.

- $\mathcal{M}, \rho \models p$ if $p \in L(last(\rho))$,
- $\mathcal{M}, \rho \models \neg\phi$ if not $\mathcal{M}, \rho \models \phi$,
- $\mathcal{M}, \rho \models \phi_1 \vee \phi_2$ if $\mathcal{M}, \rho \models \phi_1$ or $\mathcal{M}, \rho \models \phi_2$,
- $\mathcal{M}, \rho \models \forall\psi$ if $\mathcal{M}, \rho, \delta \models \psi$ for all $\delta \in \mathrm{IPath}_T^{\mathcal{M}}(last(\rho))$,
- $\mathcal{M}, \rho \models \mathsf{P}^{\bowtie q}\psi$ if $Prob_{\mathcal{M},\rho}^{opt(\bowtie)}(\psi) \bowtie q$, where

$$opt(\bowtie) = \left\{ \begin{array}{ll} \min & \text{when } \bowtie \in \{\geq, >\} \\ \max & \text{when } \bowtie \in \{\leq, <\} \end{array} \right.$$

and for any infinite continuation $\delta \in \mathrm{IPath}_T^{\mathcal{M}}$ of ρ (i.e., $\delta(0) = last(\rho)$):

- $\mathcal{M}, \rho, \delta \models \phi$ if $\mathcal{M}, \rho \models \phi$,
- $\mathcal{M}, \rho, \delta \models \neg\psi$ if not $\mathcal{M}, \rho, \delta \models \psi$,
- $\mathcal{M}, \rho, \delta \models \psi_1 \vee \psi_2$ if $\mathcal{M}, \rho, \delta \models \psi_1$ or $\mathcal{M}, \rho, \delta \models \psi_2$,
- $\mathcal{M}, \rho, \delta \models \bigcirc\psi$ if $\mathcal{M}, \rho \cdot \delta[0..1], \delta[1..\infty] \models \psi$,
- $\mathcal{M}, \rho, \delta \models \psi_1 \mathcal{U} \psi_2$ if there exists $n \geq 0$ such that $\mathcal{M}, \rho \cdot \delta[0..n], \delta[n..\infty] \models \psi_2$ and $\mathcal{M}, \rho \cdot \delta[0..k], \delta[k..\infty] \models \psi_1$ for all $0 \leq k < n$.

We note that the semantics of state formulas is defined on finite paths (histories) rather than states, whereas the semantics of path formulas is defined on a finite path together with its infinite continuation (rather than a single infinite path). The reason for defining it in such a way is to be consistent with definitions of trust operators (which we introduce in Section 8), whose semantics is dependent on execution history (understood as a sequence of past states of a system).

Below, we write $\overline{<}$ for $>$, $\overline{>}$ for $<$, $\overline{\leq}$ for $\geq$ and $\overline{\geq}$ for $\leq$ (inverting the order), and write $\widehat{<}$ for $\leq$, $\widehat{>}$ for $\geq$, $\widehat{\leq}$ for $<$, and $\widehat{\geq}$ for $>$ (strict/non-strict variants).

PROPOSITION 4.5. *Let $\mathcal{M}$ be an SMG and $\rho \in \mathrm{FPath}^{\mathcal{M}}$. The following equivalences hold for any formula ψ:*

(1) $\mathcal{M}, \rho \models \neg\mathsf{P}^{\bowtie q}\psi$ iff $\mathcal{M}, \rho \models \mathsf{P}^{\widehat{\bowtie}1-q}\neg\psi$
(2) $\mathcal{M}, \rho \models \mathsf{P}^{\bowtie q}\psi$ iff $\mathcal{M}, \rho \models \mathsf{P}^{\overline{\bowtie}1-q}\neg\psi$

Definition 4.6. For a given SMG $\mathcal{M}$ and a formula ϕ of the language PCTL*, the *model checking problem*, written as $\mathcal{M} \models \phi$, is to decide whether $\mathcal{M}, s \models \phi$ for all initial states $s \in S_{\text{init}}$.

Example 4.7. Here we give a few examples of PCTL* formulas that we may wish to check on the trust game from Example 4.2. The formula

$$\mathsf{P}^{\leq 0.9} \bigcirc (a_{Alice} = invest)$$

expresses that the probability of Alice investing in the next step is no greater than 0.9. On the other hand, the formula

$$\mathsf{P}^{\leq 1} \Diamond (a_{Bob} = keep)$$

states (the obvious fact) that the probability of Bob keeping the money in the future is no greater than 1. Finally, the formula

$$\exists \Diamond richer_{Alice,Bob},$$

where $richer_{Alice,Bob}$ is an atomic proposition with obvious meaning, states that eventually a state can be reached where Alice has more money than Bob.

All the above formulas are true when evaluated at the state s_0, given in Fig 1. □

Table 3. Payoff of a simple trust game with trust as a decision factor

	share	keep
invest	(20,20+5)	(0,40-20)
withhold	(10,5)	(10,5)

5 STOCHASTIC MULTIAGENT SYSTEMS WITH THE COGNITIVE DIMENSION

In this section, we present a framework for reasoning about autonomous agents in multiagent systems. The key novelty of our model is the consideration of agents' mental attitudes to enable autonomous decision making, which we achieve by equipping agents with *goals* and *intentions* (also called *pro-attitudes*). We enhance stochastic multiplayer games with a cognitive mechanism to represent reasoning about goals and intentions. The system then evolves along two interleaving dimensions: *temporal*, representing actions of agents in the physical space, and *cognitive*, which corresponds to mental reasoning processes, i.e., goal and intention changes, that determine the actions that agents take.

5.1 Cognitive Reasoning

We now motivate and explain the concepts of the *cognitive state* of an agent and the *cognitive mechanism*, and how they give rise to *partial observability*.

Cognitive State. We assume that each agent has a set of *goals* and a set of *intentions*, referred to as *pro-attitudes* and viewed as high-level concepts. We follow existing literature, see e.g., Bratman [6] and Gollwitzer [21], etc., and identify commitment as the distinguishing factor between goals and intentions, i.e., an intention is a committed plan of achieving some immediate goal. We therefore think of goals as abstract attitudes, for example selflessness or risk-taking, whereas intentions are more concrete and directly influence agents' behaviour. Goals are usually static and independent of external factors, whereas intentions are dynamic, influenced by agent's own goals and by its beliefs about other agents' pro-attitudes. For the purposes of our framework, we assume that agents use action strategies to *implement their intentions* and therefore there exists a one-to-one association between intentions and action strategies. This way, agents' behaviour in the physical space is determined by their mental state.

For an agent A, we use $Goal_A$ to denote its set of goals and Int_A to denote its set of intentions. At any particular time, an agent may have several goals, but can only have a single intention. Goals are not required to satisfy constraints such as consistency.

Definition 5.1. Let A be an agent and let $Goal_A$ and Int_A be its set of goals and intentions, respectively. A *cognitive state* of A consists of a set of goals and an intention, which can be retrieved from global states of the system using the following functions:

- $gs_A : S \to \mathcal{P}(Goal_A)$, i.e., $gs_A(s)$ is a set of agent A's *goals* in state s,
- $is_A : S \to Int_A$, i.e., $is_A(s)$ is the agent A's *intention* in state s.

To illustrate the concepts, we now extend the trust game given in the previous examples to include agents' goals and intentions.

Example 5.2. It is argued in [33] that the single numerical value as the payoff of the trust game introduced in Example 4.2 is an over-simplification. A more realistic utility should include both the payoff and other hypotheses, including trust. An example payoff table is given in Table 3, in which Bob's payoff will increase by 5 to denote that he will gain Alice's trust if sharing the investment yield and decrease by 20 to denote that he will lose Alice's trust if keeping the investment yield

without sharing. With the updated payoffs, the new Nash equilibrium is for Alice to invest her money and Bob to share the investment yield.

The main point for the new payoffs is for the agents to make decisions not only based on the original payoffs, but also based on the trust that the other agent has. This reflects some actual situations in which one agent may want to improve, or at least maintain, the trust of the other agent. In our modelling of such a game, we show that this can be captured by adding the cognitive dimension and assuming that Bob makes decisions by considering additionally whether Alice's trust in him reaches a certain level.

For Alice, we let $Goal_{Alice} = \{passive, active\}$ be two goals which represent her attitude towards investment. Intuitively, $passive$ represents the goal of keeping the cash and $active$ represents the goal of investing. For simplicity, we assume that Alice's intention is determined by her goals and set $Int_{Alice} = \{passive, active\}$. We also assume that Alice uses strategy $\sigma_{passive}$ to implement her $passive$ intention, and σ_{active} to implement her $active$ intention, where the strategies are defined in Example 4.2.

Bob has a set of goals $Goal_{Bob} = \{investor, opportunist\}$, which represent the goals of being an investor pursuing long-term profits and being an opportunist after short-term profits, respectively. As for Alice, Bob's intentions are associated with action strategies, and we have already defined two such strategies: σ_{share}, in which Bob shares the investment yield with Alice, and σ_{keep}, in which Bob keeps all the money for himself. Hence $Int_{Bob} = \{share, keep\}$, with the obvious association. We assume that Bob's intention will be $share$ when he is an investor and his belief in Alice being active is above a certain threshold, and $keep$ otherwise. Intuitively, when he is an $investor$, Bob intends to build a good relationship with Alice (in other words, gain Alice's $trust$), hoping that it will pay off in his future interactions with her.

We extend the trust game $\mathcal{G}$ defined in Example 4.2 by expanding the states to additionally include cognitive states. In particular, each state can now be represented as a tuple:

$$(a_{Alice}, a_{Bob}, gs_{Alice}, gs_{Bob}, is_{Alice}, is_{Bob}),$$

such that a_{Alice} and a_{Bob} are as before and $gs_{Alice} \subseteq Goal_{Alice} \cup \{\bot\}$, $gs_{Bob} \subseteq Goal_{Bob} \cup \{\bot\}$, $is_{Alice} \in Int_{Alice} \cup \{\bot\}$, and $is_{Bob} \in Int_{Bob} \cup \{\bot\}$ denote the cognitive part of the state, namely Alice's and Bob's goals and intentions.   □

Partial Observation. It is common that, in real-world systems, agents are *not* able to fully observe the system state at any given time. In a typical scenario, every agent runs a local protocol, maintains a local state, observes part of the system state by, e.g., sensing devices, and communicates with other agents. It is impractical, and in fact undesirable, from the system designer's point of view, to assume that agents can learn the local states of other agents or learn what the other agents observe. In the context of our work partial observability arises naturally through the cognitive state, which represents an *internal* state of every agent that is, in general, not observable by other agents. We formalise this notion with the following definition.

Definition 5.3. A *partially observable* stochastic multiplayer game (POSMG) is a tuple $\mathcal{M} = (G, \{O_A\}_{A \in Ags}, \{obs_A\}_{A \in Ags})$, where
- $G = (Ags, S, S_{\text{init}}, \{Act_A\}_{A \in Ags}, T, L)$ is an SMG,
- O_A is a finite set of observations for agent A, and
- $obs_A : S \longrightarrow O_A$ is a labelling of states with observations for agent A.

Remark 1. We note that, unlike partially observable Markov processes (POMDPs), in which observations are probability distributions, we follow the setting in [18] and work with deterministic observations. In [20], it is stated without proof that probabilistic observations do not increase the complexity of the problem.

We lift the observations from states to paths in the obvious way. Formally, for a finite path $\rho = s_0...s_n$, we define $obs_A(\rho) = obs_A(s_0)...obs_A(s_n)$.

Remark 2. In our model, an agent remembers both its past observations and the number of states, known as synchronous perfect recall [16]. This assumption is necessary for the definition of belief in Section 7.

Example 5.4. For the trust game from Example 5.2, the agents' observation functions for $A \in \{Alice, Bob\}$ are as follows:

$$obs_A((a_{Alice}, a_{Bob}, gs_{Alice}, gs_{Bob}, is_{Alice}, is_{Bob})) = (a_{Alice}, a_{Bob}, gs_A, is_A),$$

denoting that they cannot observe the opponent's cognitive state but can observe their last action. The set of observations O_A can be easily inferred from this definition.   □

We say $\mathcal{M}$ is *fully observable* if $s = s'$ iff $obs_A(s) = obs_A(s')$ for all $A \in Ags$. In fully observable systems, agents make decisions based on the current state, whereas in partially observable systems decisions must be based on all past observations.

Cognitive Mechanism. While agents interact with each other and with the environment by taking actions in the physical space, they make decisions through cognitive processes that determine their behaviour. Thus, in addition to the temporal dimension of transitions $s \xrightarrow{a}_T s'$, we also distinguish a *cognitive* dimension of transitions $s \longrightarrow_C s'$, which corresponds to mental reasoning processes. The idea is that each temporal transition is preceded by a cognitive transition, which represents an agent's reasoning that led to its decision about which action to take. While transitions in the temporal dimension conform to the transition function T, cognitive changes adhere to the *cognitive mechanism*, which determines for an agent its legal goals and intentions. Formally, we have the following definition of a stochastic game extended with a cognitive mechanism.

Definition 5.5. A *stochastic multiplayer game with the cognitive dimension* (SMG$_\Omega$) is a tuple $\mathcal{M} = (G, \{\Omega_A\}_{A \in Ags}, \{\pi_A\}_{A \in Ags})$, where

- $G = (Ags, S, S_{\text{init}}, \{Act_A\}_{A \in Ags}, T, L, \{O_A\}_{A \in Ags}, \{obs_A\}_{A \in Ags})$ is a POSMG,
- $\Omega_A = \langle \omega_A^g, \omega_A^i \rangle$ is the *cognitive mechanism* of agent A, consisting of a legal goal function $\omega_A^g : S \to \mathcal{P}(\mathcal{P}(Goal_A))$ and a legal intention function $\omega_A^i : S \to \mathcal{P}(Int_A)$, and
- $\pi_A = \langle \pi_A^g, \pi_A^i \rangle$ is the *cognitive strategy* of agent A, consisting of a goal strategy $\pi_A^g : \text{FPath}^{\mathcal{M}} \to \mathcal{D}(\mathcal{P}(Goal_A))$ and an intention strategy $\pi_A^i : \text{FPath}^{\mathcal{M}} \to \mathcal{D}(Int_A)$.

We refer to the SMG G from the above definition as the *induced SMG* from $\mathcal{M}$.

Thus, the SMG$_\Omega$ model generalises the usual notion of multiplayer games by extending states S with agents' cognitive states, and adding for each agent A the cognitive mechanism Ω_A and A's cognitive strategies to enable autonomous decision making. We sometimes refer to the set $\Omega = \{\Omega_A\}_{A \in Ags}$ of cognitive mechanisms of all agents as the cognitive mechanism of the system $\mathcal{M}$.

The *legal goal* (resp. *intention*) function ω_A^g (resp. ω_A^i) specifies legal goal (resp. intention) changes in a given state. Intuitively, those are goal (resp. intention) changes that an agent is allowed (but might not be willing) to make. One possible use of those functions is to enforce application-specific constraints that goals or intentions must satisfy (see Example 5.6).

The *cognitive strategy* π_A determines how an agent's cognitive state evolves over time. Specifically, the goal (resp. intention) strategy π_A^g (resp. π_A^i) specifies the incurred goal (resp. intention) changes (along with their probabilities), which are under agent A's consideration for a given execution history. We sometimes call π_A^g (resp. π_A^i) a *possible goal* (resp. *intention*) function and require that

possible goal (resp. intention) changes are also legal. Formally, for all $\rho s \in \text{FPath}^{\mathcal{M}}$, we must have $supp(\pi_A^g(\rho s)) \subseteq Goal_A(s)$ and $supp(\pi_A^i(\rho s)) \subseteq Int_A(s)$.

Remark 3. We remark that cognitive strategies π_A^g and π_A^i are, in general, not computable. In Section 9 we propose how to realise cognitive strategies so that they can be effectively computed.

We note the following correspondence between the cognitive and temporal dimension: $Goal_A$ and Int_A (regarded as sets) specify all the goals and intentions of a given agent A similarly to the way that Act_A specifies all the actions of A, while the cognitive mechanism Ω identifies possible cognitive transitions in a similar fashion to the transition function T playing the same role for the temporal dimension. Finally, a cognitive strategy gives the probability for agents' goal and intention changes, analogously to an action strategy quantifying the likelihood of actions taken by agents in the temporal dimension.

The following standard assumption ensures that agents' temporal and cognitive transitions, as well as their cognitive strategies, are consistent with their partial observability.

Assumption 1. (Uniformity Assumption) An SMG$_\Omega$ $\mathcal{M} = (Ags, S, S_{\text{init}}, \{Act_A\}_{A \in Ags}, T, L, \{O_A\}_{A \in Ags},$ $\{obs_A\}_{A \in Ags}, \{\Omega_A\}_{A \in Ags}, \{\pi_A\}_{A \in Ags})$ satisfies the *Uniformity Assumption* if the following conditions hold.

- Agents can distinguish states with different sets of joint actions or legal cognitive changes: for any two states s_1 and s_2 and an agent $A \in Ags$, $obs_A(s_1) = obs_A(s_2)$ implies $Act_B(s_1) = Act_B(s_2)$, $\omega_B^g(s_1) = \omega_B^g(s_2)$, and $\omega_B^i(s_1) = \omega_B^i(s_2)$, for all $B \in Ags$.
- Agents can distinguish execution histories which give rise to different cognitive strategies: for any two finite paths ρ_1, ρ_2, $obs_A(\rho_1) = obs_A(\rho_2)$ implies $\pi_A^g(\rho_1)(x) = \pi_A^g(\rho_2)(x)$ and $\pi_A^i(\rho_1)(x) = \pi_A^i(\rho_2)(x)$ for all $x \subseteq Goal_A$.

Given a state s and a set of agent A's goals $x \subseteq Goal_A$, we write $A.g(s, x)$ for the state obtained from s by substituting agent A's goals with x. Similar notation $A.i(s, x)$ is used for the intention change when $x \in Int_A$. Alternatively, we may write $s \longrightarrow_C^{A.g.x} s'$ if $s' = A.g(s, x)$ contains the goal set x for A and $s \longrightarrow_C^{A.i.x} s'$ if $s' = A.i(s, x)$ contains the intention x for A.

Example 5.6. Having extended states of the trust game $\mathcal{G}$ with goals and intentions in Example 5.2, we now make $\mathcal{G}$ into an SMG$_\Omega$ by introducing cognitive transitions, defining the cognitive mechanism and specifying cognitive strategies of agents.

We give a graphical representation of $\mathcal{G}$ in Figure 2. Note that w and i stand for Alice's actions *withhold* and *invest* respectively, whereas s and k denote Bob's actions *share* and *keep*. Cognitive transitions are represented with dashed lines. Temporal actions are annotated with probabilities which reflect the intention (i.e., an action strategy) that an agent has in a given state.

Below we explain how we arrived at such a system. The execution of the game starts by agents choosing their goals. While it may seem unnatural (a more realistic approach would probably involve multiple initial states corresponding to agents having different goals), such a way of modelling plays well with our formalism and does not restrict the generality of our approach. Formally, we specify those cognitive transitions using legal goal function for Alice and Bob as follows:

$$\omega_{Alice}^g(s_0) = \{\{active\}, \{passive\}\},$$
$$\omega_{Bob}^g(s_k) = \{\{investor\}, \{opportunist\}\},$$

where $k \in \{1, 2\}$.

Once their goals are set, agents begin interacting with one another in the physical space by taking actions. However, each action is preceded by an agent determining its intention, which represents
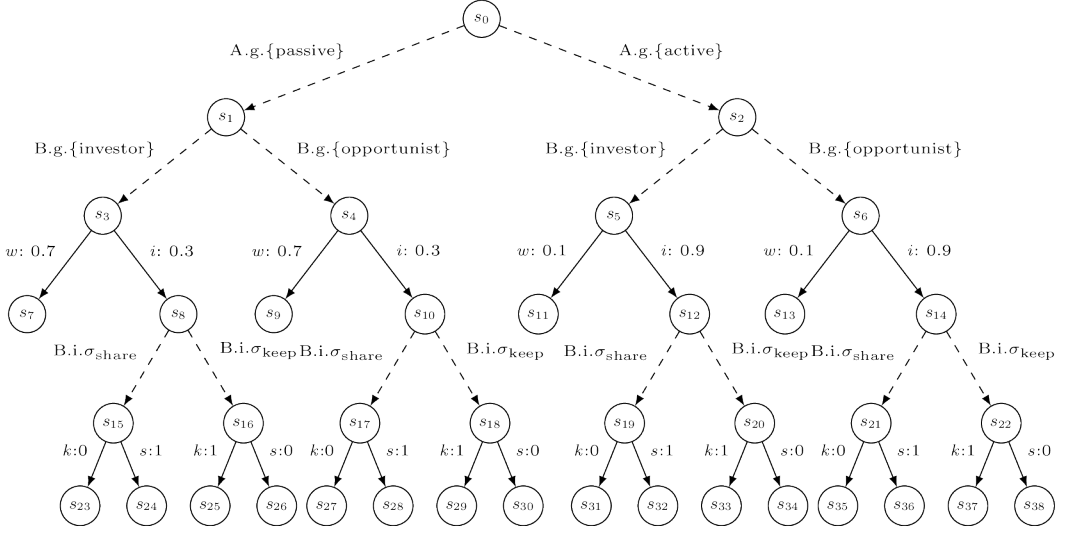
Fig. 2. Trust game with cognitive dimension

the mental reasoning that results in action selection. Note that we do not depict Alice's intention change in Figure 2, the reason being our assumption that Alice's intention is fully determined by her goals.

As mentioned above, Alice's actions are annotated with probabilities, according to our assumption that cognitive state determines agents' behaviour. For example, in states s_3 and s_4, Alice's intention is *passive*, and so the probabilities of withholding and investing the money are given by Table 2. If Alice withholds her money, the game ends. Otherwise, Bob determines his intention (his choice depends on his goals and his belief about Alice's goals) and performs his action of keeping or sharing his profit. Bob's legal intentions are given by his legal intention function, defined as:

$$\omega_{Bob}^i(s_k) = \{share, keep\},$$

where $k \in \{8, 10, 12, 14\}$.

Finally, we define Bob's intention strategy. As mentioned above, Bob takes on intention *share* only if he is an *investor* and he believes that Alice is *active*. The latter depends on Alice's actions in the physical space (since Bob can't observe Alice's cognitive state). In this case, Alice investing her money with Bob increases his belief that she is *active*. We therefore set:

$$\pi_{Bob}^i(s_0 s_1 s_3 s_8) = \langle share \mapsto 1, keep \mapsto 0 \rangle$$
$$\pi_{Bob}^i(s_0 s_1 s_4 s_{10}) = \langle share \mapsto 0, keep \mapsto 1 \rangle$$
$$\pi_{Bob}^i(s_0 s_2 s_5 s_{12}) = \langle share \mapsto 1, keep \mapsto 0 \rangle$$
$$\pi_{Bob}^i(s_0 s_2 s_6 s_{14}) = \langle share \mapsto 0, keep \mapsto 1 \rangle$$

Note that the above strategy is pure. While the framework does not enforce it, we believe defining it in such a way more accurately resembles human cognitive processes. Also, as we will see later, pure intention strategies are more compatible with our trust formulations.

Finally, note that we don't consider goal strategies here. The reason for that is our representation of goals in this example as static mental attitudes which agents possess or not, rather than choose

dynamically. We therefore treat the first two cognitive transitions as nondeterministic for now. In Section 6, we introduce a mechanism which motivates choosing such representation. □

Remark 4. We remark that, while defining the intention strategy as in the above example is easy for simple systems, for more complex games this approach does not scale. In particular, we could consider repeated version of the trust game, for which constructing Bob's intention strategy manually is impractical. However, in Section 7, we formalise the notion of agent's belief and, in Section 9, we propose an intuitive method of constructing the intention strategy efficiently.

Induced SMG. For a concrete system, it is conceptually simpler to assume a certain, predefined interleaving of cognitive and temporal transitions, as we did in Example 5.6. However, in general, such interleaving might be arbitrary since agents may change their mental state at any time. It is therefore often useful to think of a SMG_Ω as a collection of induced SMGs, each corresponding to one configuration of mental states of agents. Those induced SMGs do not differ as far as their components are concerned (i.e., states, actions and transitions are the same), but different mental states give rise to different behaviour. Execution of such an SMG_Ω can then be viewed as starting in one of the induced SMGs, remaining there as long as agents perform temporal actions and moving to a different induced SMG as soon as one agent changes its mental state. In the long run, execution alternates between different standard SMGs, where, at any point, current SMG reflects current mental states of agents, and each temporal transition preserves the current SMG, while each cognitive transition switches to a different SMG. The benefit of such an approach is that each induced SMG can then be reasoned about using standard techniques. We note also that, for the purposes of the logic operators introduced later in the paper, we assume that both temporal and cognitive transitions are available to agents in any state.

Paths. In contrast to the conventional multiplayer games, where each path includes a sequence of temporal transitions, which are consistent with the transition function T, in an SMG_Ω, in view of the active participation of agents' pro-attitudes in determining their behaviour, a path can be constructed by interleaving of temporal *and* cognitive transitions. Each cognitive transition represents a change of an agent's goals or intention. We now extend the definition of a path to allow cognitive transitions.

Definition 5.7. Given a stochastic multiplayer game with the cognitive dimension (SMG_Ω) $\mathcal{M}$, we define a finite, respectively infinite, *path* ρ as a sequence of states $s_0 s_1 s_2 \ldots$ such that, for all $k \geq 0$, one of the following conditions is satisfied:

(1) $s_k \xrightarrow{a}_T s_{k+1}$ for some joint action a such that $a \in Act(s_k)$,
(2) $s_k \xrightarrow{A.g.x}_C s_{k+1}$ for some $A \in Ags$ and $x \subseteq Goal_A(s_k)$,
(3) $s_k \xrightarrow{A.i.x}_C s_{k+1}$ for some $A \in Ags$ and $x \in Int_A(s_k)$.

We reuse the notation introduced in Section 4.1 for regular paths and denote the set of finite (resp. infinite) paths by $FPath^{\mathcal{M}}$ (resp. $IPath^{\mathcal{M}}$), and the set of finite (resp. infinite) paths starting in state s by $FPath^{\mathcal{M}}(s)$ (resp. $IPath^{\mathcal{M}}(s)$).

The first condition represents the standard temporal transition, while the other two stand for cognitive transitions – the former is a goal change, whereas the latter is an intention change.

We often require that both the legal goal function and the legal intention function are serial, i.e., for any state s and any subset x of $Goal_A$, there exists a state $s' \in S$ such that $s \xrightarrow{A.g.x}_C s'$, and for any state s and any intention $x \in Int_A$, there exists a state $s' \in S$ such that $s \xrightarrow{A.i.x}_C s'$. This is a non-trivial requirement, since such states s' can be unreachable via temporal transitions.

Remark 5. We remark that the seriality requirement for the probabilistic transition function is usually imposed for model checking, and by no means reduces the generality of the problem, as we

can introduce absorbing states to accommodate undefined temporal or cognitive transitions in an obvious way.

Deterministic Behaviour Assumption. Recall that intentions of agents are associated with action strategies, thereby determining agents' behaviour in a physical space. Below, we formalise that idea, in addition requiring that the associated strategies are pure, which simplifies results.

Assumption 2. (Deterministic Behaviour Assumption) An SMG_Ω $\mathcal{M}$ satisfies the *Deterministic Behaviour Assumption* if each agent's cognitive state deterministically decides its behaviour in terms of selecting its next local action. In other words, agent's cognitive state induces a pure action strategy that it follows.

Hence, since global states encode agents' cognitive states, under Assumption 2 the transition function T becomes deterministic, i.e., for every state s, there exists a unique joint action a such that next state s' is chosen with probability $T(s, a)(s')$. Therefore, each SMG induced from $\mathcal{M}$, with fixed pro-attitudes, can be regarded as a Markov chain.

Remark 6. We note that a more general version of Assumption 2 is possible, where the action strategy is not assumed to be pure, and our results can be easily adapted to that variant. In fact, action strategies introduced in Example 4.2 are not pure and so the trust game does not satisfy the strict version of the Deterministic Behaviour Assumption as stated above. Example 6.4 illustrates how the calculations can be adapted to handle that.

5.2   Cognitive Reasoning

We can now extend the logic PCTL* with operators for reasoning about agent's cognitive states, resulting in the logic called Probabilistic Computation Tree Logic with Cognitive Operators, $PCTL_\Omega^*$.

Definition 5.8. The syntax of the logic $PCTL_\Omega^*$ is:

$$\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid \forall\psi \mid \mathsf{P}^{\bowtie q}\psi \mid \mathbb{G}_A\phi \mid \mathbb{I}_A\phi \mid \mathbb{C}_A\phi$$
$$\psi ::= \phi \mid \neg\psi \mid \psi \vee \psi \mid \bigcirc\psi \mid \psi\mathcal{U}\psi$$

where $p \in AP$, $A \in Ags$, $\bowtie \in \{<, \leq, >, \geq\}$, and $q \in [0, 1]$.

Intuitively, the newly introduced *cognitive* operators $\mathbb{G}_A\phi$ (goal), $\mathbb{I}_A\phi$ (intention) and $\mathbb{C}_A\phi$ (capability) consider the task expressed as ϕ and respectively quantify, in the cognitive dimension, over *possible changes of goals*, *possible intentions* and *legal intentions*.

The semantics for $PCTL_\Omega^*$ is as follows.

Definition 5.9. Let $\mathcal{M} = (Ags, S, S_{\text{init}}, \{Act_A\}_{A \in Ags}, T, L, \{O_A\}_{A \in Ags}, \{obs_A\}_{A \in Ags}, \{\Omega_A\}_{A \in Ags},$ $\{\pi_A\}_{A \in Ags})$ be a SMG_Ω, ρ a finite path in $\mathcal{M}$ and $s \in S$ such that $\rho s \in FPath^{\mathcal{M}}$. The semantics of previously introduced operators of PCTL* remains the same in $PCTL_\Omega^*$. For the newly introduced cognitive operators, the satisfaction relation $\models$ is defined as follows:

- $\mathcal{M}, \rho s \models \mathbb{G}_A\phi$ if $\forall x \in supp(\pi_A^g(\rho s)) \, \exists s' \in S : s \xrightarrow{A.g.x}_C s'$ and $\mathcal{M}, \rho s s' \models \phi$,
- $\mathcal{M}, \rho s \models \mathbb{I}_A\phi$ if $\forall x \in supp(\pi_A^i(\rho s)) \, \exists s' \in S : s \xrightarrow{A.i.x}_C s'$ and $\mathcal{M}, \rho s s' \models \phi$,
- $\mathcal{M}, \rho s \models \mathbb{C}_A\phi$ if $\exists x \in \omega_A^i(s) \, \exists s' \in S : s \xrightarrow{A.i.x}_C s'$ and $\mathcal{M}, \rho s s' \models \phi$.

Thus, $\mathbb{G}_A\phi$ expresses that ϕ holds in future regardless of agent A changing its goals. Similarly, $\mathbb{I}_A\phi$ states that ϕ holds regardless of A changing its intention, whereas $\mathbb{C}_A\phi$ quantifies over the legal intentions, and thus expresses that agent A could change its intention to achieve ϕ (however, such a change might not be among agent's possible intention changes).

Table 4. Complexity of Markov chain model checking

PCTL	PTIME-complete
PCTL*	PSPACE-complete

Remark 7. We note that, when evaluating PCTL* operators, we assume that agents keep their current mental attitudes, i.e., that the future path is purely temporal. Formally, for a SMG$_\Omega$ $\mathcal{M}$, PCTL* operators should be interpreted over the SMG induced from $\mathcal{M}$. Furthermore, when evaluating PCTL$^*_\Omega$ formulas, we assume agents can change their goals and intentions at any time, in line with the 'induced SMGs' interpretation presented in Section 5.1. That ensures that the cognitive operators can be applied at any point of execution, as well as meaningfully chained, nested or manipulated in any other way.

Remark 8. We comment here about our definition of the semantics of $\mathbb{G}_A\phi$, $\mathbb{I}_A\phi$ and $\mathbb{C}_A\phi$, whereby the changes of goals and intentions do not incur the changes of other components of the state. This can be counter-intuitive for some cases, e.g., it is reasonable to expect that the intention of the agent may change when its goals are changed. We believe that such dependencies are better handled at the modelling stage. For example, a simultaneous change of goals and intention can be modelled as two consecutive cognitive transitions – a goal change followed by an intention change.

Below, we write $\overline{X}\phi$ with $X \in \{\mathbb{G}_A, \mathbb{I}_A, \mathbb{C}_A\}$ for $\neg X \neg \phi$. For instance, formula $\overline{\mathbb{I}_A}\phi$ expresses that it is possible to achieve ϕ by changing agent A's intention. Note that it is not equivalent to $\mathbb{C}_A\phi$, which quantifies over legal, rather than possible, intentions.

Example 5.10. Here we give examples of formulas that we may wish to check on the trust game from Example 5.6. The formula:

$$\mathbb{G}_{Alice}\mathsf{P}^{\leq 0.9}\Diamond(a_{Alice} = invest)$$

expresses that, regardless of Alice changing her goals, the probability of her investing in the future is no greater than 90%. On the other hand, the formula:

$$\mathbb{C}_{Bob}\mathsf{P}^{\leq 0} \bigcirc (a_{Bob} = keep)$$

states that Bob has a legal intention which ensures that he will not keep the money as his next action. Also, the formula:

$$\overline{\mathbb{I}_{Alice}}\exists\Diamond richer_{Alice,Bob},$$

where $richer_{Alice,Bob}$ is an atomic proposition with obvious meaning, states that Alice can find an intention such that eventually a state can be reached where Alice has more money than Bob. Finally, the formula:

$$\overline{\mathbb{I}_{Alice}}\exists\Diamond\mathbb{G}_{Bob}\forall\Diamond\neg richer_{Alice,Bob}$$

expresses that Alice can find an intention such that it is possible to reach a state such that, for all possible Bob's goals, the game will always reach a state in which Bob is no poorer than Alice. □

In this paper we study the model checking problem defined as follows.

Definition 5.11. For a given SMG$_\Omega$ $\mathcal{M}$ and a formula ϕ of the language PCTL$^*_\Omega$ the *model checking problem*, written as $\mathcal{M} \models \phi$, is to decide whether $\mathcal{M}, s \models \phi$ for all initial states $s \in S_{\text{init}}$.

The model checking problem amounts to checking whether a given finite model satisfies a formula. For logics PCTL and PCTL* the model checking problem over Markov chains is known to be decidable, with the complexity results summarised in Table 4. These logics thus often feature

in probabilistic model checkers, e.g., PRISM [35] and Storm [15]. In contrast, the satisfiability problem, i.e., the whether there exists a model that satisfies a given formula, for these logics is an open problem [8]. Therefore. the satisfiability problem for the logic introduced here is likely to be challenging.

Consider a SMG$_\Omega$ $\mathcal{M}$ in which agents never change their mental attitudes. Then, thanks to Assumption 2, all transitions in the system are effectively deterministic and $\mathcal{M}$ can be viewed as a Markov chain. Using complexity results summarised in Table 4 we formalise the above observation with the following theorem.

THEOREM 5.12. *If the cognitive strategies of all agents in a system $\mathcal{M}$ are constant, then the complexity of model checking PCTL$_\Omega^*$ over SMG$_\Omega$ is PSPACE-complete, and model checking PCTL$_\Omega$ over SMG$_\Omega$ is PTIME-complete.*

However, it is often unrealistic to assume that agents' cognitive strategies are constant. In Section 9, we suggest a variation of our model based on agents reasoning about their beliefs and trust, all of whose components are finite, which makes it amenable for model checking.

6 PREFERENCE FUNCTIONS AND PROBABILITY SPACES

In this section, we develop the foundations for reasoning with *probabilistic beliefs*, which we define in Section 7. In order to support *subjective* beliefs, we utilise the concept of *preference functions*, which resolve the nondeterminism arising from agents' cognitive transitions in a similar way to how action strategies resolve nondeterminism in the temporal dimension. This enables the definition of probability spaces to support reasoning about beliefs, which will in turn provide the basis for reasoning about trust. The central model of this paper, autonomous stochastic multiagent systems, is then introduced.

Preference Functions. To define a belief function, the usual approach is to employ for every agent a preference ordering, which is a measurement over the worlds. This measurement is commonly argued to naturally exist with the problem, see [18] for an example. However, the actual definition of such a measurement can be non-trivial, because the number of worlds can be very large or even infinite, e.g. in [18] and in this paper, and thus enumerating all worlds can be infeasible.

In this paper, instead of an (infinite definition of) preference ordering, we resolve the nondeterminism in the system by introducing *preference functions*. The key idea for a preference function is to estimate, for an agent A, the possible changes of goals or intentions of another agent B in a given state using a probability distribution. In other words, preference functions model *probabilistic prior* knowledge of agent A about pro-attitudes of another agent B. That knowledge may be derived from prior experience (through observations), personal preferences, social norms, etc., and will in general vary between agents. A uniformly distributed preference function can be assumed if no prior information is available, as is typical in Bayesian settings.

Resolving the nondeterminism (for temporal dimension – by Assumption 2, for cognitive dimension – by preference functions) allows us to define a family of probability spaces for each agent. Since preference functions vary among agents, probability spaces are also different for every agent. Moreover, each agent has multiple probability spaces, corresponding to its own various cognitive states. Intuitively, every probability space represents agent's *subjective* view on the relative likelihood of different infinite paths being taken in the system.

We are now ready to define the central model of this paper.

Definition 6.1. An *autonomous stochastic multi-agent system* (ASMAS) is a tuple $\mathcal{M} = (G, \{p_A\}_{A \in Ags})$, where

- $G = (Ags, S, S_{\text{init}}, \{Act_A\}_{A \in Ags}, T, L, \{O_A\}_{A \in Ags}, \{obs_A\}_{A \in Ags}, \{\Omega_A\}_{A \in Ags}, \{\pi_A\}_{A \in Ags})$ is an SMG_Ω and
- p_A is a family of preference functions of agent $A \in Ags$, defined as

$$p_A \overset{\text{def}}{=} \{gp_{A,B}, ip_{A,B} \mid B \in Ags \text{ and } B \neq A\},$$

where:

- $gp_{A,B} : S \to \mathcal{D}(\mathcal{P}(Goal_B))$ is a goal preference function of A over B such that, for any state s and $x \in \mathcal{P}(Goal_B)$, we have $gp_{A,B}(s)(x) > 0$ only if $x \in \omega_B^g(s)$, and
- $ip_{A,B} : S \to \mathcal{D}(Int_B)$ is an intention preference function of A over B such that, for any state s and $x \in Int_B$, we have $ip_{A,B}(s)(x) > 0$ only if $x \in \omega_B^i(s)$.

Remark 9. During system execution, preference functions may be updated as agents learn new information through interactions; we will discuss in Section 9 how this can be implemented in our framework.

Intuitively, a preference function provides agent A with a probability distribution over another agent B's changes of pro-attitudes. Naturally, we expect preference functions to be consistent with partial observability. We therefore extend the *Uniformity Assumption* in the following way.

Assumption 3. (Uniformity Assumption II) Let $\mathcal{M} = (Ags, S, S_{\text{init}}, \{Act_A\}_{A \in Ags}, T, L, \{O_A\}_{A \in Ags}, \{obs_A\}_{A \in Ags}, \{\Omega_A\}_{A \in Ags}, \{\pi_A\}_{A \in Ags}, \{p_A\}_{A \in Ags})$ be an ASMAS. For an agent $A \in Ags$ and any two states $s_1, s_2 \in S$, we assume that $obs_A(s_1) = obs_A(s_2)$ implies $gp_{B,A}(s_1)(x) = gp_{B,A}(s_2)(x)$ and $ip_{B,A}(s_1)(x) = ip_{B,A}(s_2)(x)$ for any $B \in Ags$ such that $B \neq A$ and $x \subseteq Goal_A$. That is, agents' preferences over a given agent are the same on all paths which a given agent cannot distinguish.

We also mention that Assumption 2 (Deterministic Behaviour Assumption) extends to ASMAS in a straightforward manner.

Transition Type. In general, in any state of the system, an agent may choose a temporal or a cognitive transition. However, it is often desirable, e.g., when constructing a probability space, to restrict the *type* of transition available to an agent.

Definition 6.2. Given a path $s_0 s_1 s_2 \dots$ in an ASMAS $\mathcal{M}$ satisfying Assumption 2 and an agent $A \in Ags$, we use $tp_A(s_k, s_{k+1})$ to denote the type, as seen by agent A, of the transition that is taken to move from state s_k to s_{k+1}. More specifically, we distinguish five different *transition types*:

- $tp_A(s_k, s_{k+1}) = a$ if $s_k \longrightarrow_T^a s_{k+1}$ for some $a \in Act$,
- $tp_A(s_k, s_{k+1}) = A.g.x$ if $s_k \longrightarrow_C^{A.g.x} s_{k+1}$ for some $x \subseteq \omega_A^g(s_k)$,
- $tp_A(s_k, s_{k+1}) = A.i.x$ if $s_k \longrightarrow_C^{A.i.x} s_{k+1}$ for some $x \in \omega_A^i(s_k)$,
- $tp_A(s_k, s_{k+1}) = B.g$ if $s_k \longrightarrow_C^{B.g.x} s_{k+1}$ for another agent $B \in Ags$ and $x \subseteq \omega_B^g(s_k)$,
- $tp_A(s_k, s_{k+1}) = B.i$ if $s_k \longrightarrow_C^{B.i.x} s_{k+1}$ for another agent $B \in Ags$ and $x \in \omega_B^i(s_k)$.

Remark 10. Assumption 2 guarantees that, if the transition form s_k to s_{k+1} is temporal, then the action a is uniquely determined.

We write $tp_A(\rho) = tp_A(\rho(0), \rho(1)) \cdot tp_A(\rho(1), \rho(2)) \cdot \dots$ for the type of a path ρ. When ρ is a finite path and t is a type of an infinite path, we say that ρ is *consistent* with t if there exists an infinite extension $\rho' = \rho\delta$ of ρ such that $tp_A(\rho') = t$.

We note that the type of agent's own cognitive transitions is defined differently than the type of another agent's cognitive transitions. This stems from our implicit assumption that agents can observe their own cognitive changes, but they cannot, in general, observe another agent's cognitive transitions. Formally, we make the following assumption.

Assumption 4. (Transition Type Distinguishability Assumption) Let $\mathcal{M} = (Ags, S, S_{\text{init}}, \{Act_A\}_{A \in Ags},$ $T, L, \{O_A\}_{A \in Ags}, \{obs_A\}_{A \in Ags}, \{\Omega_A\}_{A \in Ags}, \{\pi_A\}_{A \in Ags}, \{p_A\}_{A \in Ags})$ be an ASMAS. For an agent $A \in$ Ags and any two finite paths ρ_1, ρ_2, we assume that $obs_A(\rho_1) = obs_A(\rho_2)$ implies $tp_A(\rho_1) = tp_A(\rho_2)$. That is, agents can distinguish paths of different types.

This assumption is essential to ensure that the belief function is defined over finite paths in the same probability space.

Initial Distribution. We replace the set of initial states with a more accurate notion of *initial distribution* $\mu_0 \in \mathcal{D}(S_{\text{init}})$, as a common prior assumption of the agents about the system, which is needed to define a belief function.

Probability Spaces. We now define a family of probability spaces for an arbitrary ASMAS. Recall that our ultimate goal is to define the belief function. To motivate the following construction, we mention that the belief function provides a probability distribution over paths which are indistinguishable (i.e., have the same observation) to a given agent. There are two important consequences of this: (i) probability spaces will vary among agents, reflecting the difference in their partial observation, and (ii) rather than defining a single probability space, spanning all the possible paths, for a given agent, we may define many of them, as long as every pair of indistinguishable paths lies in the same probability space. Hence, using the fact that agents can observe the type of transitions (by Assumption 4), we parameterise probability spaces by type, so that each contains paths of unique type.

We begin by introducing an auxiliary transition function, specific to each agent, which will be used to define the probability measure.

Definition 6.3. Let $\mathcal{M} = (Ags, S, S_{\text{init}}, \{Act_A\}_{A \in Ags}, T, L, \{O_A\}_{A \in Ags}, \{obs_A\}_{A \in Ags}, \{\Omega_A\}_{A \in Ags},$ $\{\pi_A\}_{A \in Ags}, \{p_A\}_{A \in Ags})$ be an ASMAS. Based on temporal transition function T and preference functions $\{p_A\}_{A \in Ags}$, we define an auxiliary *transition function* T_A for agent $A \in Ags$ as follows for $s, s' \in S$:

$$T_A(s, s') = \begin{cases} T(s, a)(s') & \text{if } tp_A(s, s') = a \\ gp_{A,B}(s)(x) & \text{if } tp_A(s, s') = B.g \text{ and } s \xrightarrow{B.g.x}_C s' \\ ip_{A,B}(s)(x) & \text{if } tp_A(s, s') = B.i \text{ and } s \xrightarrow{B.i.x}_C s' \\ 1 & \text{if } tp_A(s, s') = A.g.x \text{ for some } x \in \omega_A^g(s) \\ & \text{or } tp_A(s, s') = A.i.x \text{ for some } x \in \omega_A^i(s) \end{cases}$$

The application of the function T_A resolves the nondeterminism in both the temporal and cognitive dimensions and the resulting system is a family of probability spaces, each containing a set of paths of the same type. While the probabilities of temporal transitions are given by the transition function T and computed by all agents in the same way, each agent resolves the nondeterminism in the cognitive dimension differently. For an agent A, all possible cognitive transitions of another agent B have the same type and hence lie in the same probability space; A uses its preference functions to associate probabilities to them. On the other hand, each of A's own possible pro-attitude changes has a different type and lies in a different probability space; A therefore treats it as a deterministic transition and assigns probability 1 to it.

We now construct a probability space for an arbitrary ASMAS $\mathcal{M}$, agent A and a type t in a similar way as in Section 4. The sample space consists of infinite paths of type t starting in one of the initial states, i.e., $\Omega_{\mathcal{M},t} = \bigcup_{s \in S, \mu_0(s) > 0} \{\delta \in \text{IPath}^{\mathcal{M}}(s) \mid tp_A(\delta) = t\}$. We associate a probability to each finite path $\rho = s_0...s_n$ consistent with t via function $\text{Pr}_A(\rho) = \mu_0(\rho(0)) \cdot$ $\prod_{0 \le i \le |\rho| - 2} T_A(\rho(i), \rho(i+1))$. We then set $\mathcal{F}_{\mathcal{M}}$ to be the smallest σ-algebra generated by cylinders

$\{\mathrm{Cyl}_\rho \cap \Omega_{\mathcal{M},t} \mid \rho \in \mathrm{FPath}^{\mathcal{M}}\}$ and $\mathrm{Pr}_A^{\mathcal{M}}$ to be the unique measure such that $\mathrm{Pr}_A^{\mathcal{M}}(\mathrm{Cyl}_\rho) = \mathrm{Pr}_A(\rho)$. It then follows that $(\Omega_{\mathcal{M},t}, \mathcal{F}_{\mathcal{M}}, \mathrm{Pr}_A^{\mathcal{M}})$ is a probability space.

Note that, for agents A and B such that $A \neq B$ in a system $\mathcal{M}$, agents' probability spaces will in general differ in probability measures $\mathrm{Pr}_A^{\mathcal{M}}$ and $\mathrm{Pr}_B^{\mathcal{M}}$, because their preference functions may be different.

Example 6.4. We now define preference functions for Bob and Alice of Example 4.2. For example, setting:

$$gp_{Bob, Alice}(s_0) = \langle passive \mapsto 1/3, active \mapsto 2/3 \rangle$$

indicates that Bob believes Alice is more likely to be *active* than *passive*. Similarly, we give Alice's preference functions. We first note that $obs_{Bob}(s_1) = obs_{Bob}(s_2)$. Therefore, by Assumption 3, $gp_{Alice, Bob}(s_1) = gp_{Alice, Bob}(s_2)$. Setting:

$$gp_{Alice, Bob}(s_k) = \langle investor \mapsto 1/2, opportunist \mapsto 1/2 \rangle,$$

for $k \in \{1, 2\}$, represents that Alice has no prior knowledge regarding Bob's mental attitudes. Finally, we define Alice's intention preference over Bob. Since $obs_{Bob}(s_8) = obs_{Bob}(s_{12})$ and $obs_{Bob}(s_{10}) = obs_{Bob}(s_{14})$, Assumption 3 implies that $ip_{Alice, Bob}(s_8) = ip_{Alice, Bob}(s_{12})$ and $ip_{Alice, Bob}(s_{10}) = ip_{Alice, Bob}(s_{14})$. We may set:

$$ip_{Alice, Bob}(s_k) = \langle share \mapsto 3/4, keep \mapsto 1/4 \rangle \quad \text{for } k \in \{8, 12\},$$
$$ip_{Alice, Bob}(s_k) = \langle share \mapsto 0, keep \mapsto 1 \rangle \quad \text{for } k \in \{10, 14\}$$

to indicate that Alice knows that Bob will keep the money when he is an *opportunist*, but she thinks it is quite likely that he will share his profit when he is an *investor*.

We now compute the probability that Alice and Bob will have the same amount of money at the end of the game. In other words, we want to find the probability that Alice invests her money with Bob and Bob shares his profit with her. As noted above, probability spaces differ between agents. We first perform the computation from Alice's point of view and consider two cases: (i) Alice being *passive* and (ii) Alice being *active*. For (i), letting $\rho_1 = s_0 s_1 s_3 s_8 s_{15} s_{24}$ and $\rho_2 = s_0 s_1 s_4 s_{10} s_{17} s_{28}$, we compute:

$$\mathrm{Pr}_{Alice}(\rho_1) = gp_{Alice, Bob}(s_1)(investor) \cdot (\sigma_{passive}(s_0 s_1 s_3)(invest) \cdot T(s_3, invest)(s_8))$$
$$\cdot\; ip_{Alice, Bob}(s_8)(share) \;\cdot\; (\sigma_{share}(s_0 s_1 s_3 s_8 s_{15})(share) \cdot T(s_{15}, share)(s_{24}))$$
$$= \frac{1}{2} \cdot (\frac{3}{10} \cdot 1) \cdot \frac{3}{4} \cdot (1 \cdot 1) = \frac{9}{80},$$

$$\mathrm{Pr}_{Alice}(\rho_2) = gp_{Alice, Bob}(s_1)(opportunist) \cdot (\sigma_{passive}(s_0 s_1 s_4)(invest) \cdot T(s_4, invest)(s_{10}))$$
$$\cdot\; ip_{Alice, Bob}(s_{10})(share) \;\cdot\; (\sigma_{share}(s_0 s_1 s_4 s_{10} s_{17})(share) \cdot T(s_{17}, share)(s_{28}))$$
$$= \frac{1}{2} \cdot (\frac{3}{10} \cdot 1) \cdot 0 \cdot (1 \cdot 1) = 0.$$

Similarly, in case (ii), letting $\rho_3 = s_0 s_2 s_5 s_{12} s_{19} s_{32}$ and $\rho_3 = s_0 s_2 s_6 s_{14} s_{21} s_{36}$, we compute:

$$\mathrm{Pr}_{Alice}(\rho_3) = gp_{Alice, Bob}(s_2)(investor) \cdot (\sigma_{active}(s_0 s_2 s_5)(invest) \cdot T(s_5, invest)(s_{12}))$$
$$\cdot\; ip_{Alice, Bob}(s_{12})(share) \;\cdot\; (\sigma_{share}(s_0 s_2 s_5 s_{12} s_{19})(share) \cdot T(s_{19}, share)(s_{32}))$$
$$= \frac{1}{2} \cdot (\frac{9}{10} \cdot 1) \cdot \frac{3}{4} \cdot (1 \cdot 1) = \frac{27}{80},$$

$$\mathrm{Pr}_{Alice}(\rho_4) = gp_{Alice, Bob}(s_2)(opportunist) \cdot (\sigma_{active}(s_0 s_2 s_6)(invest) \cdot T(s_6, invest)(s_{14}))$$
$$\cdot\; ip_{Alice, Bob}(s_{14})(share) \;\cdot\; (\sigma_{share}(s_0 s_2 s_6 s_{14} s_{21})(share) \cdot T(s_{21}, share)(s_{36}))$$
$$= \frac{1}{2} \cdot (\frac{9}{10} \cdot 1) \cdot 0 \cdot (1 \cdot 1) = 0.$$

Hence, from Alice's point of view, the probability that both will have 20 dollars at the end of the game is three times higher when she is *active*, and roughly equal to 1/3. This is consistent with our expectations, since the only difference between the two scenarios is the likelihood of her investing, which is three times greater when she is *active*.

We now perform a similar computation for Bob. Again, we consider (i) Bob being an *investor* and (ii) Bob being an *opportunist*. We start with case (i), but we have to be a little more careful now. In order for the result to be meaningful, all paths in which Bob is an *investor* must be in the same probability space. However, currently, this is not the case for paths corresponding to different intention changes for Bob. To fix that, we use Bob's intention strategy π^i_{Bob} to quantify the likelihoods of paths in different probability spaces. We may then compute:

$$
\begin{aligned}
\mathrm{Pr}_{Bob}(\rho_1) &= gp_{Bob,Alice}(s_0)(passive) \cdot (\sigma_{passive}(s_0 s_1 s_3)(invest) \cdot T(s_3, invest)(s_8)) \\
&\quad \cdot\ \pi^i_{Bob}(s_0 s_1 s_3 s_8)(share) \\
&\quad \cdot\ (\sigma_{share}(s_0 s_1 s_3 s_8 s_{15})(share) \cdot T(s_{15}, share)(s_{24})) \\
&= \frac{1}{3} \cdot (\frac{3}{10} \cdot 1) \cdot 1 \cdot (1 \cdot 1) = \frac{1}{10}, \\
\mathrm{Pr}_{Bob}(\rho_3) &= gp_{Bob,Alice}(s_0)(active) \cdot (\sigma_{active}(s_0 s_2 s_5)(invest) \cdot T(s_5, invest)(s_{12})) \\
&\quad \cdot\ \pi^i_{Bob}(s_0 s_2 s_5 s_{12})(share) \\
&\quad \cdot\ (\sigma_{share}(s_0 s_2 s_5 s_{12} s_{19})(share) \cdot T(s_{19}, share)(s_{32})) \\
&= \frac{2}{3} \cdot (\frac{9}{10} \cdot 1) \cdot 1 \cdot (1 \cdot 1) = \frac{3}{5}.
\end{aligned}
$$

In case (ii), Bob chooses intention σ_{keep} with probability 1 and so the probability of him sharing the profit is 0. □

7 REASONING WITH PROBABILISTIC BELIEFS

In presence of partial observation and, resulting from it, imperfect information that agents have about system state, one resorts to using beliefs to represents agents' knowledge. In this section, we define a *probabilistic belief* function be_A of an agent A in an autonomous stochastic multiagent system (ASMAS), which models agents' uncertainty about the system state and its execution history.

In [18], a qualitative belief notion is defined based on a plausibility measure. We consider probability measure instead, and define a *quantitative* belief function expressing that agent A believes ϕ with probability in relation $\bowtie$ to q if it *knows* ϕ with probability in relation $\bowtie$ to q. This can be seen as a quantitative variant of the qualitative definition in [43], according to which agent A believes ϕ if the probability of ϕ is close to 1. It is generally accepted that trust describes agents' subjective probability [19] and therefore needs to be measured quantitatively.

7.1 Belief Function

Before defining the belief function, we establish an equivalence relation $\sim^o_A$ on the set of finite paths of an autonomous system $\mathcal{M}$ for an agent A. We say that paths $\rho, \rho' \in \mathrm{FPath}^{\mathcal{M}}$ are equivalent (from the point of view of A), denoted $\rho \sim^o_A \rho'$, if and only if $obs_A(\rho) = obs_A(\rho')$. Intuitively, each equivalence class contains paths which a given agent cannot differentiate from each other. Note that, by Assumption 4, each equivalence class comprises paths of the same type.

Example 7.1. Recall the trust game $\mathcal{G}$ from Example 5.6 and the observation function defined on $\mathcal{G}$ in Example 5.4. Using that definition, we obtain $obs_{Bob}(s_0 s_1) = obs_{Bob}(s_0 s_2)$, which is intuitively correct, since Bob cannot observe Alice's goals. These two paths form an equivalence class. Similarly,

$obs_{Bob}(s_0s_1s_3s_8) = obs_{Bob}(s_0s_2s_5s_{12})$ and $obs_{Bob}(s_0s_1s_4s_{10}) = obs_{Bob}(s_0s_2s_6s_{14})$, and both pairs of paths form an equivalence class. □

The belief function quantifies agent's belief about the execution history of the system at any given time. Intuitively, at any point during system execution, agent's observation history uniquely identifies an equivalence class (with respect to $\sim_A^o$), consisting of all paths with that observation. The belief function gives a probability distribution on that equivalence class. We define $OPath_A = \{obs_A(\rho) \mid \rho \in \text{FPath}^{\mathcal{M}}\}$ to be a set of all finite observation histories (path observations), and for a given path observation $o \in OPath_A$, $class(o)$ denotes the equivalence class associated with o. Recall that, for any finite path ρ, Cyl_ρ denotes a basic cylinder with prefix ρ.

Definition 7.2. Let $\mathcal{M}$ be an ASMAS and A an agent in $\mathcal{M}$. The *belief function* $be_A : OPath_A \to \mathcal{D}(\text{FPath}^{\mathcal{M}})$ of an agent A is given by:

$$be_A(o)(\rho) = \Pr_A^{\mathcal{M}}(\text{Cyl}_\rho \mid \bigcup_{\rho' \in class(o)} \text{Cyl}_{\rho'}).$$

Hence, given that agent A's path observation is o at some point, their belief that the execution history at that time is ρ is expressed as the *conditional probability* of the execution history being ρ, given that the execution history belongs to the equivalence class $class(o)$. We note that sometimes, e.g., when the observation is clear from the context, we might omit it, and simply write $be_A(\rho)$ for agent A's belief that the execution history at some point is ρ.

Example 7.3. We return again to Example 5.6 and let $\rho_1 = s_0s_1$ and $\rho_2 = s_0s_2$. Recall from Example 7.1 that $obs_{Bob}(\rho_1) = obs_{Bob}(\rho_2)$, and let o_1 denote that common observation. Hence, o_1 is a path observation associated with the equivalence class $\{\rho_1, \rho_2\}$. We compute $be_{Bob}(o_1, \rho_1)$ and $be_{Bob}(o_1, \rho_2)$ below.

$$
\begin{aligned}
be_{Bob}(o_1, \rho_1) &= \Pr_{Bob}^{\mathcal{G}}(\text{Cyl}_{\rho_1} \mid \bigcup_{\rho \in class(o)} \text{Cyl}_\rho) \\
&= \frac{\Pr_{Bob}^{\mathcal{G}}(\text{Cyl}_{\rho_1})}{\Pr_{Bob}^{\mathcal{G}}(\text{Cyl}_{\rho_1}) + \Pr_{Bob}^{\mathcal{G}}(\text{Cyl}_{\rho_2})} \\
&= \frac{gp_{Bob,Alice}(s_0)(passive)}{gp_{Bob,Alice}(s_0)(passive) + gp_{Bob,Alice}(s_0)(active)} \\
&= \frac{1}{3}.
\end{aligned}
$$

Similar computation shows that:

$$be_{Bob}(o_1, \rho_2) = \frac{2}{3}.$$

We also let $\rho_3 = s_0s_1s_3s_8$, $\rho_4 = s_0s_2s_5s_{12}$ and recall from Example 7.1 that $obs_{Bob}(\rho_3) = obs_{Bob}(\rho_4)$. We let o_2 denote that common observation and, performing similar computation as above, we obtain:

$$be_{Bob}(o_2, \rho_3) = \frac{1}{7},$$

$$be_{Bob}(o_2, \rho_4) = \frac{6}{7}.$$

Analogously, letting $\rho_5 = s_0 s_1 s_4 s_{10}$, $\rho_6 = s_0 s_2 s_6 s_{14}$ and $o_3 = obs_{Bob}(\rho_5) = obs_{Bob}(\rho_6)$, we compute:

$$\mathsf{be}_{Bob}(o_3, \rho_5) = \frac{1}{7},$$
$$\mathsf{be}_{Bob}(o_3, \rho_6) = \frac{6}{7}.$$

$\square$

The following theorem gives a recursive definition of the belief function. Intuitively, the beliefs of agent A are represented as a distribution over a set of possible states which are reachable by paths consistent with the current observation history. This belief distribution will be updated in the Bayesian way whenever a new observation is taken.

THEOREM 7.4. *The beliefs* $\mathsf{be}_A(\rho)$ *can be computed recursively over the length of path* ρ:

- *For an initial state* s, $\mathsf{be}_A(s) = \dfrac{\mu_0(s)}{\sum_{s' \in S \wedge obs_A(s')=obs_A(s)} \mu_0(s')}.$
- *For* ρ *a path such that* $|\rho| = k + 2$ *and* $k \geq 0$,

$$\mathsf{be}_A(\rho) = \frac{\mathsf{be}_A(\rho[0..k]) \times T_A(\rho(k), \rho(k+1))}{\sum_{\rho' \in \mathrm{FPath}^{\mathcal{M}} \wedge obs_A(\rho')=obs_A(\rho[0..k])} \sum_{s \in S \wedge obs_A(s)=obs_A(\rho(k+1))} \mathsf{be}_A(\rho') \times T_A(\rho'(k), s)}.$$

7.2 Belief ASMAS

A conceptually simpler construction, defined below, can be effectively used to reason about *a single agent's* understanding about the system, when there is no need for nested reasoning on agents' beliefs. We employ a well-known construction [10] which, for a system $\mathcal{M}$ and a fixed agent A, induces an equivalent *belief ASMAS* $\beta_A(\mathcal{M})$ whose states are *distributions* over states of $\mathcal{M}$ called *belief states*. Intuitively, belief states quantify agent's uncertainty about the current state of a system by specifying their belief, expressed as a probability distribution: although we may not know which of several observationally-equivalent states we are currently in, we can determine the likelihood of being in each one. The belief ASMAS is fully observable but its state space is possibly infinite.

Formally, a belief ASMAS for an ASMAS $\mathcal{M}$ and an agent A is a tuple $\beta_A(\mathcal{M}) = (Ags, \mathcal{D}(S), \mathcal{B}_{\mathrm{init}}, \{Act_A\}_{A \in Ags}, T_A^\beta, L)$, where $\mathcal{B}_{\mathrm{init}} = \{b_{\mathrm{init}}^o \mid o \in O_A\}$ is a set of initial belief states such that:

$$b_{\mathrm{init}}^o(s) = \begin{cases} \dfrac{\mu_0(s)}{\sum_{s' \in S \& obs_A(s')=obs_A(s)} \mu_0(s')} & \text{if } obs_A(s) = o \\ 0 & \text{otherwise.} \end{cases}$$

We now give the transition function T_A^β. Intuitively, from a belief state b, different belief states b' are possible, each corresponding to a unique combination of A's next observation and the type of transition taken. Transition probabilities are then computed based on the temporal transition function T or A's preferences. The definition is split into three parts, reflecting the different types of transitions present in an ASMAS.

First, for $b, b' \in \mathcal{D}(S)$ and $a \in Act$, we have

$$T_A^\beta(b, a)(b') = \sum_{s \in S} b(s) \cdot \Big(\sum_{o \in O_A \& b^{a,o}=b'} \sum_{s' \in S \& b'(s')>0} T(s, a)(s') \Big),$$

where $b^{a,o}$ is a belief state reached from b by performing a and observing o, i.e.,

$$b^{a,o}(s') = \begin{cases} \dfrac{\sum_{s \in S} b(s) \cdot T(s, a)(s')}{\sum_{s \in S} b(s) \sum_{s'' \in S \& obs_A(s'')=o} T(s, a)(s'')} & \text{if } obs_A(s') = o \\ 0 & \text{otherwise.} \end{cases}$$

Intuitively, the belief state reached by agent A from b by performing a and observing o is a probability distribution over states s such that $obs_A(s) = o$, where the probability of each state is weighted according to the probabilities of transitions from states in the support of b to that state.

Second, for $x \subseteq Goal_A$,

$$T_A^\beta(b, A.g.x)(b') = \sum_{s \in S} b(s) \cdot (\sum_{o \in O_A \& b^{A.g.x,o} = b'} \sum_{s' \in S \& b'(s') > 0} (s \xrightarrow{A.g.x}_C s')),$$

where $b^{A.g.x,o}$ is belief reached from b by performing A's goal change into x and observing o, i.e.,

$$b^{A.g.x,o}(s') = \begin{cases} \dfrac{\sum_{s \in S} b(s) \cdot (s \xrightarrow{A.g.x}_C s')}{\sum_{s \in S} b(s) \sum_{s'' \in S \& obs_A(s'')=o}(s \xrightarrow{A.g.x}_C s'')} & \text{if } obs_A(s') = o \\ 0 & \text{otherwise.} \end{cases}$$

Note the we interpret $s \xrightarrow{A.g.x}_C s'$ as having value 0 or 1, depending on whether a transition of a specified type exists or not. The case corresponding to an intention change of agent A is defined analogously.

Third, for $B \in Ags$ such that $B \neq A$,

$$T_A^\beta(b, B.g)(b') = \sum_{s \in S} b(s) \cdot (\sum_{o \in O_A \& b^{B.g,o} = b'} \sum_{s' \in S \& b'(s') > 0} \sum_{x \in \omega_A^g(s)} gp_{A,B}(s)(x) \cdot (s \xrightarrow{B.g.x}_C s')),$$

where $b^{B.g,o}$ is belief reached from b by performing B's goal change into x and observing o, i.e.,

$$b^{B.g,o}(s') = \begin{cases} \dfrac{\sum_{s \in S} b(s) \cdot \sum_x gp_{A,B}(s)(x) \cdot (s \xrightarrow{B.g.x}_C s')}{\sum_{s \in S} b(s) \sum_{s'' \in S \& obs_A(s'')=o} \sum_x gp_{A,B}(s)(x) \cdot (s \xrightarrow{B.g.x}_C s'')} & \text{if } obs_A(s') = o \\ 0 & \text{otherwise.} \end{cases}$$

The case of intention change can be defined in a similar manner.

Example 7.5. We illustrate the above construction on the trust game $\mathcal{G}$ from Example 5.6 by considering the initial few states of the belief ASMAS $\beta_{Bob}(\mathcal{G})$ corresponding to $\mathcal{G}$. Since the game has a single initial state, the belief ASMAS also has a single initial belief state, namely $b_0 = b_{init}^{o_1} = \langle s_0 \mapsto 1 \rangle$.

We now compute possible belief states after the first transition, i.e. Alice's goal change. Recall that Bob cannot distinguish states s_1 and s_2; we let $o_1 = obs_{Bob}(s_1) = obs_{Bob}(s_2)$. Then the new belief state is $b_1 = b_0^{Alice.g,o_1}$, with:

$$b_1(s_1) = b_0^{Alice.g,o_1}(s_1) = \frac{b_0(s_0) \cdot gp_{Bob,Alice}(s_0)(\{passive\})}{b_0(s_0) \cdot (gp_{Bob,Alice}(s_0)(\{passive\}) + gp_{Bob,Alice}(s_0)(\{active\}))}$$

$$= \frac{\frac{2}{3}}{\frac{2}{3} + \frac{1}{3}} = \frac{2}{3},$$

where we used Bob's preference function $gp_{Bob,Alice}$ defined in Example 6.4. Similar computation shows that:

$$b_1(s_2) = \frac{1}{3}.$$

Since Bob has only one possible observation at this point of the game, we expect that b_1 is his only possible belief state after first transition, i.e. $T_{Bob}^\beta(b_0, Alice.g)(b_1) = 1$. Indeed,

$$T_{Bob}^\beta(b_0, Alice.g)(b_1) = b_0(s_0) \cdot (gp_{Bob,Alice}(s_0)(\{passive\}) + gp_{Bob,Alice}(s_0)(\{active\}))$$

$$= 1.$$
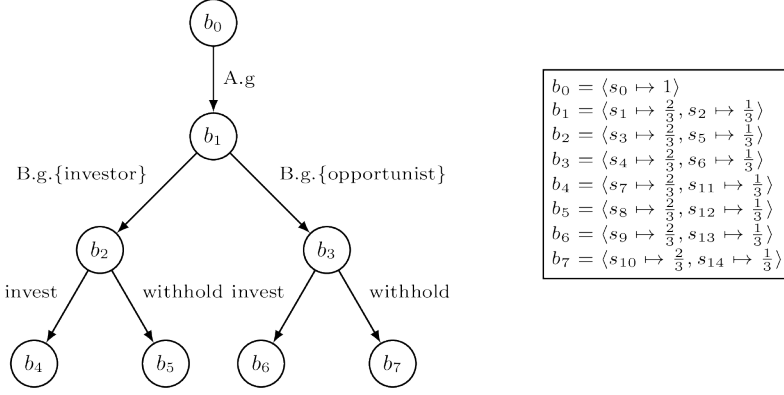
Fig. 3. Belief ASMAS $\beta_{Bob}(\mathcal{G})$

Next, Bob sets his goal. We let $o_2 = obs_{Bob}(s_3) = obs_{Bob}(s_5)$ and $o_3 = obs_{Bob}(s_4) = obs_{Bob}(s_6)$. If Bob observes o_2, then his belief state is $b_2 = b_1^{Bob.g,o_2}$, such that:

$$b_2 = \langle s_3 \mapsto 2/3, s_5 \mapsto 1/3 \rangle.$$

If he observes o_3 then his belief state is $b_3 = b_1^{Bob.g,o_3}$, with:

$$b_3 = \langle s_4 \mapsto 2/3, s_6 \mapsto 1/3 \rangle.$$

The transition function is as follows.

$$T_{Bob}^{\beta}(b_1, Bob.g.\{investor\})(b_2) = 1,$$

$$T_{Bob}^{\beta}(b_1, Bob.g.\{opportunist\})(b_3) = 1.$$

Finally, we consider how belief state changes after Alice's temporal transition. There are four possible observations for Bob at this stage, namely $o_4 = obs_{Bob}(s_7) = obs_{Bob}(s_{11})$, $o_5 = obs_{Bob}(s_8) = obs_{Bob}(s_{12})$, $o_6 = obs_{Bob}(s_9) = obs_{Bob}(s_{13})$, $o_7 = obs_{Bob}(s_{10}) = obs_{Bob}(s_{14})$. Hence there are four possible belief states, $b_4 = b_2^{Bob.g,o_4}$, $b_5 = b_2^{Bob.g,o_5}$, $b_6 = b_3^{Bob.g,o_6}$, $b_7 = b_3^{Bob.g,o_7}$, such that:

$$b_4 = \langle s_7 \mapsto 2/3, s_{11} \mapsto 1/3 \rangle,$$

$$b_5 = \langle s_8 \mapsto 2/3, s_{12} \mapsto 1/3 \rangle,$$

$$b_6 = \langle s_9 \mapsto 2/3, s_{13} \mapsto 1/3 \rangle,$$

$$b_7 = \langle s_{10} \mapsto 2/3, s_{14} \mapsto 1/3 \rangle.$$

Figure 3 depicts belief states of $\beta_{Bob}(\mathcal{G})$ which we have computed.

□

As in our construction of the probability measure $\mathrm{Pr}_{s,\sigma}$ for SMGs in Section 4, a strategy profile σ induces a probability measure $\mathrm{Pr}_{b,\sigma}$ on any belief state b over infinite paths of ASMAS $\beta_A(\mathcal{M})$. We note that, in general, $\beta_A(\mathcal{M})$ has a (continuous) infinite state space.

8 REASONING ABOUT BELIEFS AND TRUST

In this section, we formalise the notions of trust, which are inspired by the social trust theory of [17]. We define trust in terms of the belief function and further extend PCTL$_\Omega^*$ to capture reasoning about beliefs and trust.

In Section 7, a Bayesian-style definition of the belief function is presented. It uses agents' partial observation function and probability measure, which is in turn defined in terms of preference functions introduced in Section 6. In this section, we assume the existence of such a belief setting, i.e., one belief function for every agent, and define operators to reason about agents' beliefs and trusts.

Expressing Trust. Our trust operators follow the intuition of social trust concepts from [17], but the technical definitions are different and more rigorous. We distinguish between two types of trust, *competence*, meaning that agent A believes that B is capable of producing the expected result, and *disposition*, which means that agent A believes that agent B is willing to do what A needs. Moreover, we express trust with the usual *probabilistic* quantification of certainty that we sometimes use in our daily life, e.g., "I am 99% certain that the autonomous taxi service is trustworthy", or "I trust the autonomous taxi service 99%". Our formalisation then captures how the value of 99% can be computed based on the agent's past experience and (social, economic) preferences. Indeed, [17] also provides a justification for quantifying trust with probabilities, e.g. in Section 5, where it is stated that the degree of trust is a function of the subjective certainty of the pertinent beliefs. We do not consider fulfilment belief also discussed in [17].

Beliefs and trust are intimately connected. As agents continuously interact with the environment and with each other, their understanding about the system and the other agents may increase, leading to an update of their beliefs. This change in an agent's beliefs may also lead to an update of trust, which is reflected in the definition of the semantics of our trust operators.

Changes of beliefs and trust will affect goals and intentions, which will be captured via so called pro-attitude synthesis, which is introduced in Section 9. Based on its updated beliefs, an agent may modify its strategy to implement its intention.

8.1 Probabilistic Rational Temporal Logic

We now introduce Probabilistic Rational Temporal Logic (PRTL*) that can express mental attitudes of agents in an ASMAS, as well as beliefs and trust. PRTL* extends the logic PCTL^*_Ω with operators for reasoning about agent's beliefs and cognitive trust.

Definition 8.1. The syntax of the logic PRTL* is as follows:

$$\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid \forall\psi \mid \mathsf{P}^{\bowtie q}\psi \mid \mathbb{G}_A\phi \mid \mathbb{I}_A\phi \mid \mathbb{C}_A\phi \mid \mathbb{B}_A^{\bowtie q}\psi \mid \mathbb{CT}_{A,B}^{\bowtie q}\psi \mid \mathbb{DT}_{A,B}^{\bowtie q}\psi$$
$$\psi ::= \phi \mid \neg\psi \mid \psi \vee \psi \mid \bigcirc\psi \mid \psi\mathcal{U}\psi$$

where $p \in AP$, $A, B \in Ags$, $\bowtie \in \{<, \leq, >, \geq\}$, and $q \in [0, 1]$.

Intuitively, $\mathbb{B}_A^{\bowtie q}\psi$ is the *belief* operator, expressing that agent A believes ψ with probability in relation $\bowtie$ with q. $\mathbb{CT}_{A,B}^{\bowtie q}\psi$ is the *competence trust* operator, expressing that agent A trusts agent B with probability in relation $\bowtie$ with q on its *capability* of completing the task ψ, where capability is understood to be the possibility of taking one of its legal intentions defined with function ω_B^i. $\mathbb{DT}_{A,B}^{\bowtie q}\psi$ is the *disposition trust* operator, expressing that agent A trusts agent B with probability in relation $\bowtie$ with q on its *willingness* to do the task ψ, where the state of willingness is interpreted as unavoidably taking an intention defined with function π_B^i. Recall that we use function ω_B^i for the legal intentions and function π_B^i for the possible intentions, i.e., intentions available assuming agent's willingness. We use $\mathbb{T}$ to range over the two trust operators $\mathbb{CT}$ and $\mathbb{DT}$.

Before we define the semantics, we require additional notation. We write:

$$\text{Pr}_{\mathcal{M},A,\rho}^{max,min}(\psi) \quad \overset{\text{def}}{=} \quad \sup_{\sigma_A \in \Sigma_A} \inf_{\sigma_{Ags\setminus\{A\}} \in \Sigma_{Ags\setminus\{A\}}} \text{Pr}_{\mathcal{M},\sigma,\rho}(\psi),$$

$$\text{Pr}_{\mathcal{M},A,\rho}^{min,max}(\psi) \quad \overset{\text{def}}{=} \quad \inf_{\sigma_A \in \Sigma_A} \sup_{\sigma_{Ags\setminus\{A\}} \in \Sigma_{Ags\setminus\{A\}}} \text{Pr}_{\mathcal{M},\sigma,\rho}(\psi)$$

to denote the *strategic ability* of agent A in implementing ψ on a finite path ρ. Intuitively, $\Pr_{\mathcal{M},A,\rho}^{max,min}(\psi)$ gives a lower bound on agent A's ability to maximise the probability of ψ, while $\Pr_{\mathcal{M},A,\rho}^{min,max}(\psi)$ gives an upper bound on agent A's ability to minimise the probability of ψ.

Moreover, we extend the expression $B.i(s, x)$ to work with finite paths: $B.i(\rho, x)$ for the path $\rho' = \rho s$ and $last(\rho) \longrightarrow_C^{B.i.x} s$. Intuitively, it is the concatenation of path ρ with a cognitive transition $B.i.x$.

For a measurable function $f : \mathrm{FPath}^{\mathcal{M}} \to [0, 1]$, we denote by $E_{be_A}[f]$ the *belief-weighted expectation* of f, i.e.,

$$E_{be_A}[f] = \sum_{\rho \in \mathrm{FPath}^{\mathcal{M}}} be_A(\rho) \cdot f(\rho).$$

With this in mind, we aim to define the new operators so that $\mathbb{B}_A^{\bowtie q}\psi$ normalises the probability measure of the formula $\mathsf{P}^{\bowtie q}\psi$ with agent's probabilistic belief, whereas $\mathbb{T}_{A,B}^{\bowtie q}\psi$ operators consider moreover the possible or legal intention changes of another agent B. The intention changes are conditioned over agent's strategic ability of implementing formula ψ, expressed by $\Pr_{\mathcal{M},A,\rho}^{max,min}(\psi)$ and $\Pr_{\mathcal{M},A,\phi}^{min,max}(\psi)$.

Definition 8.2. Let $\mathcal{M} = (Ags, S, S_{\mathrm{init}}, \{Act_A\}_{A \in Ags}, T, L, \{O_A\}_{A \in Ags}, \{obs_A\}_{A \in Ags}, \{\Omega_A\}_{A \in Ags}, \{\pi_A\}_{A \in Ags}, \{p_A\}_{A \in Ags})$ be an ASMAS and ρ a finite path in $\mathcal{M}$. The semantics of previously introduced operators of PCTL* and PCTL*$_\Omega$ remains the same in PRTL*(see Definitions 4.4 and 5.9). For the newly introduced belief and trust operators, the satisfaction relation $\models$ is defined as follows:

- $\mathcal{M}, \rho \models \mathbb{B}_A^{\bowtie q}\psi$ if

$$E_{be_A}[V_{\mathbb{B},\mathcal{M},\psi}^{\bowtie}] \bowtie q,$$

where the function $V_{\mathbb{B},\mathcal{M},\psi}^{\bowtie} : \mathrm{FPath}^{\mathcal{M}} \to [0, 1]$ is such that

$$V_{\mathbb{B},\mathcal{M},\psi}^{\bowtie}(\rho') = \begin{cases} \Pr_{\mathcal{M},A,\rho'}^{max,min}(\psi) & \text{if } \bowtie \in \{\geq, >\} \\ \Pr_{\mathcal{M},A,\rho'}^{min,max}(\psi) & \text{if } \bowtie \in \{<, \leq\} \end{cases}$$

- $\mathcal{M}, \rho \models \mathbb{CT}_{A,B}^{\bowtie q}\psi$ if

$$E_{be_A}[V_{\mathbb{CT},\mathcal{M},B,\psi}^{\bowtie}] \bowtie q,$$

where the function $V_{\mathbb{CT},\mathcal{M},B,\psi}^{\bowtie} : \mathrm{FPath}^{\mathcal{M}} \to [0, 1]$ is such that

$$V_{\mathbb{CT},\mathcal{M},B,\psi}^{\bowtie}(\rho') = \begin{cases} \sup_{x \in \omega_B^i(last(\rho'))} \Pr_{\mathcal{M},A,B.i(\rho',x)}^{max,min}(\psi) & \text{if } \bowtie \in \{\geq, >\} \\ \inf_{x \in \omega_B^i(last(\rho'))} \Pr_{\mathcal{M},A,B.i(\rho',x)}^{min,max}(\psi) & \text{if } \bowtie \in \{<, \leq\} \end{cases}$$

- $\mathcal{M}, \rho \models \mathbb{DT}_{A,B}^{\bowtie q}\psi$ if

$$E_{be_A}[V_{\mathbb{DT},\mathcal{M},B,\psi}^{\bowtie}] \bowtie q,$$

where the function $V_{\mathbb{DT},\mathcal{M},B,\psi}^{\bowtie} : \mathrm{FPath}^{\mathcal{M}} \to [0, 1]$ is such that

$$V_{\mathbb{DT},\mathcal{M},B,\psi}^{\bowtie}(\rho') = \begin{cases} \inf_{x \in supp(\pi_B^i(\rho'))} \Pr_{\mathcal{M},A,B.i(\rho',x)}^{max,min}(\psi) & \text{if } \bowtie \in \{\geq, >\} \\ \sup_{x \in supp(\pi_B^i(\rho'))} \Pr_{\mathcal{M},A,B.i(\rho',x)}^{min,max}(\psi) & \text{if } \bowtie \in \{<, \leq\} \end{cases}$$

We interpret formulas ϕ in ASMAS $\mathcal{M}$ in a state reached after executing a path ρ, in history-dependent fashion. Recall that this path may have interleaved cognitive and temporal transitions. However, when evaluating a given belief or trust formula, we assume that agents do not change

their mental attitudes (i.e., the future path is purely temporal, contained within a single induced SMG), which motivates using the probability measure defined in Section 4 in the above definition. The belief formula corresponds to the probability of satisfying ϕ in future in the original ASMAS $\mathcal{M}$ weighted by the belief distribution; in other words, it is a belief-weighted expectation of future satisfaction of ϕ, which is subjective, as it is influenced by A's partial observation and its prior knowledge about B encoded in the preference function. The competence trust operator reduces to the computation of optimal probability of satisfying ψ in $\mathcal{M}$ over legal changes of agent's intention, which is again weighted by the belief distribution and compared to the probability bound q. Dispositional trust, on the other hand, computes the optimal probability of satisfying ψ in $\mathcal{M}$ over possible changes of agent's intention, which is weighted by the belief distribution and compared to the probability bound q.

For a relational symbol $\bowtie \in \{\geq, >\}$, the expression $V_{\mathrm{CT}, \mathcal{M}, B, \psi}^{\bowtie}(\rho)$ computes the maximum probability of completing the task ψ on path ρ, among all legal changes of agent B's intention. Therefore, when interpreting formula $\mathbb{CT}_{A,B}^{\bowtie q}\psi$, we assume the optimal capability of agent B. Note that this capability is not necessarily within B's possible intention changes. On the other hand, for $\bowtie \in \{\geq, >\}$, the expression $V_{\mathrm{DT}, \mathcal{M}, B, \psi}^{\bowtie}(\rho)$ computes the minimum probability of completing the task ψ on path ρ that all agent B's possible intentional changes can achieve. Therefore, when interpreting formula $\mathbb{DT}_{A,B}^{\bowtie q}\psi$, we consider all possible states of agent B's willingness. It should be noted that the $\mathbb{T}_{A,B}^{\bowtie q}$ operators cannot be derived from the other operators.

Remark 11. We remark that, while defining competence trust in terms of legal intentions should be rather uncontroversial, the disposition trust operator may be expressed in an alternative way. Given that $\mathbb{DT}_{A,B}^{\bowtie q}$ computes A's trust towards B and that intentional attitudes of B are in general not observable to A, one could argue that it is not realistic to assume that A uses B's intention strategy to compute its trust towards B. It would perhaps be more natural to use A's intention preference function $ip_{A,B}$ instead, since it precisely represents A's expectations about B's mental attitudes. Indeed, it is possible to make such a modification, which we discuss below.

First of all, note that, in the definition of $\mathbb{DT}_{A,B}^{\bowtie q}$ above, we restrict A's knowledge to the support of π_B^i, i.e., to B's possible intentions, rather than allowing A to know B's full intention strategy. As a side note, the consequence of this is that A's disposition trust towards B differs from competence trust only when the set of possible intentions of B is a strict subset of B's legal intentions. In particular, it is the case when B's intention strategy is pure, such as in Example 5.6. Hence, an important practical consideration to bear in mind when constructing instances of ASMAS is to keep cognitive strategies pure, or close to pure (with few intentions in the support). In fact, Section 9 provides an easy way to achieve this.

Going back to the alternative way of defining $\mathbb{DT}_{A,B}^{\bowtie q}$, if we replace B's intention strategy π_B^i by A's intention preferences over B, $ip_{A,B}$, it makes sense to include in our calculations of trust the full probability distribution that $ip_{A,B}$ provides, rather than just its support, as we do for π_B^i. That's because $ip_{A,B}$ represents knowledge that A realistically possesses, so there's no reason to introduce any restrictions on it. We then end up with $\mathbb{DT}_{A,B}^{\bowtie q}$ defined as a *belief-preference-weighted expectation* of agent's strategic ability of implementing a given formula ψ. Note that, with such formulation of $\mathbb{DT}_{A,B}^{\bowtie q}$, the last equivalence of Theorem 8.10 does not hold.

Finally, note that disposition trust operator of one agent may be nested inside a belief operator of another agent, which calls for nested preferences (i.e., preferences over preferences) in an idealised version of $\mathbb{DT}_{A,B}^{\bowtie q}$. We aim to avoid such complexities at this early stages of the development of the framework, and therefore settle for a simplified notion of disposition trust involving intention strategy.

Example 8.3. Here we give examples of formulas that we may wish to verify on the trust game defined in Example 5.6. The formula

$$\mathbb{B}^{\geq 0.6}_{Bob} \bigcirc (a_{Alice} = invest)$$

states that Bob believes that, with probability at least 0.6, Alice will invest the money with him in the next step. On the other hand, the formula

$$\mathbb{DT}^{\geq 0.9}_{Alice,Bob} \Diamond (a_{Bob} = keep)$$

states that Alice can trust Bob with probability no less than 0.9 that he will keep the money for himself. The formula

$$\forall \Box (richer_{Bob,Alice} \rightarrow \mathsf{P}^{\geq 0.9} \Diamond \mathbb{CT}^{\geq 1.0}_{Bob,Alice} richer_{Alice,Bob})$$

states that, at any point of the game, if Bob is richer than Alice, then with probability at least 0.9, in future, he can almost surely, i.e., with probability 1, trust Alice on her ability to become richer.

It is also possible, and often desired, to have a trust operator within the scope of a belief operator. For instance,

$$\mathbb{B}^{\geq 0.7}_{Bob} \mathbb{DT}^{\geq 0.5}_{Alice,Bob} \Diamond (a_{Bob} = share)$$

expresses that Bob believes that, with probability at least 0.7, Alice's trust that he will share the profits with her in the future is at least 0.5. Such construct might represent Bob's reasoning leading to his decision of whether to share his profit or not, where he considers Alice's trust towards him as a determining factor. A similar statement might be expressed by the following formula:

$$\mathbb{B}^{\geq 0.7}_{Bob} \mathbb{B}^{\geq 0.5}_{Alice} \mathbb{I}_{Bob} \exists \Diamond (a_{Bob} = share).$$

The above states that Bob believes that, with probability at least 0.7, Alice believes that, with probability at least 0.5, Bob has a possible intention which ensures that it is possible that he shares the profit. The first formula may be seen as a quantitative version of the second, as it additionally (implicitly, via $\mathbb{DT}$ operator) considers the probability of Bob sharing the profit. However, both formulas are equivalent for the trust game, assuming the action strategies defined in Example 5.2 and intention strategy from Example 5.6. To see that, first note that Bob's intention strategy is pure, hence he has only one possible intention. Second, since the action strategy implementing that intention (whatever it is) is pure, the probability of him sharing the profit is either 1 or 0. Therefore, $\inf_{x \in supp(\pi^i_{Bob}(\rho))} \Pr_{\mathcal{G},Bob.i(\rho,x)}(\Diamond(a_{Bob} = share))$ takes only two values, 1 and 0, and is equal to 1 whenever $\Pr_{\mathcal{G},Alice,\rho}(\mathbb{I}_{Bob} \exists \Diamond(a_{Bob} = share)) = 1$ for any path $\rho = \rho' s_k$ for $k \in \{8, 10, 12, 14\}$. Hence,

$$\mathbb{DT}^{\geq 0.5}_{Alice,Bob} \Diamond (a_{Bob} = share) \equiv \mathbb{B}^{\geq 0.5}_{Alice} \mathbb{I}_{Bob} \exists \Diamond (a_{Bob} = share).$$

In general, however, those two formulas differ in their semantics.   □

Definition 8.4. For a given autonomous stochastic multi-agent system $\mathcal{M}$ and a formula ϕ of the language PRTL*, the *model checking problem*, written as $\mathcal{M} \models \phi$, is to decide whether $\mathcal{M}, s \models \phi$ for all initial states $s \subseteq supp(\mu_0)$.

8.2 Restricting to Deterministic Behaviour Assumption

The above semantics is complex because of the quantification over agents' strategies. However, under Assumption 2, this quantification is not necessary, since the system conforms to a single action strategy, induced from agents' cognitive states, which we call the induced strategy σ_{ind}. As a result, some operators have simpler semantics. We begin with an easy consequence of restricting ourselves to a single action strategy.

PROPOSITION 8.5. *With Assumption 2 on ASMAS $\mathcal{M}$, we have the following equivalence for ρ being a finite path, and ψ being a PRTL* formula.*

$$\mathrm{Pr}^{max,min}_{\mathcal{M},A,\rho}(\psi) = \mathrm{Pr}^{min,max}_{\mathcal{M},A,\rho}(\psi) = \mathrm{Pr}^{max,min}_{\mathcal{M},\rho}(\psi) = \mathrm{Pr}^{min,max}_{\mathcal{M},\rho}(\psi) = \mathrm{Pr}_{\mathcal{M},\sigma_{ind},\rho}(\psi).$$

Moreover, for systems $\mathcal{M}$ satisfying Assumption 2, we will often omit the σ_{ind} component in the last expression above and write $\mathrm{Pr}_{\mathcal{M},\rho}(\psi)$ instead. Intuitively, it represents the probability of ψ being satisfied on a future path in a system $\mathcal{M}$ given that the current path is ρ.

Definition 8.6. Let $\mathcal{M}$ be an ASMAS satisfying Assumption 2 and ρ a finite path in $\mathcal{M}$. The following are several operators whose semantics can be simplified according to Proposition 8.5.

- $\mathcal{M}, \rho \models \mathsf{P}^{\bowtie q}\psi$ if

$$\mathrm{Pr}_{\mathcal{M},\rho}(\psi) \bowtie q.$$

- $\mathcal{M}, \rho \models \mathbb{B}^{\bowtie q}_A \psi$ if

$$E_{\mathsf{be}_A}[V_{\mathbb{B},\mathcal{M},\psi}] \bowtie q,$$

where the function $V_{\mathbb{B},\mathcal{M},\psi} : \mathrm{FPath}^{\mathcal{M}} \to [0,1]$ is such that

$$V_{\mathbb{B},\mathcal{M},\psi}(\rho') = \mathrm{Pr}_{\mathcal{M},\rho'}(\psi)$$

- $\mathcal{M}, \rho \models \mathbb{CT}^{\bowtie q}_{A,B} \psi$ if

$$E_{\mathsf{be}_A}[V^{\bowtie}_{\mathbb{CT},\mathcal{M},B,\psi}] \bowtie q,$$

where the function $V^{\bowtie}_{\mathbb{CT},\mathcal{M},B,\psi} : \mathrm{FPath}^{\mathcal{M}} \to [0,1]$ is such that

$$V^{\bowtie}_{\mathbb{CT},\mathcal{M},B,\psi}(\rho') = \begin{cases} \sup\limits_{x \in \omega^i_B(last(\rho'))} \mathrm{Pr}_{\mathcal{M},B.i(\rho',x)}(\psi) & \text{if } \bowtie \in \{\geq, >\} \\ \inf\limits_{x \in \omega^i_B(last(\rho'))} \mathrm{Pr}_{\mathcal{M},B.i(\rho',x)}(\psi) & \text{if } \bowtie \in \{<, \leq\}. \end{cases}$$

- $\mathcal{M}, \rho \models \mathbb{DT}^{\bowtie q}_{A,B} \psi$ if

$$E_{\mathsf{be}_A}[V^{\bowtie}_{\mathbb{DT},\mathcal{M},B,\psi}] \bowtie q,$$

where the function $V^{\bowtie}_{\mathbb{DT},\mathcal{M},B,\psi} : \mathrm{FPath}^{\mathcal{M}} \to [0,1]$ is such that

$$V^{\bowtie}_{\mathbb{DT},\mathcal{M},B,\psi}(\rho') = \begin{cases} \inf\limits_{x \in supp(\pi^i_B(\rho'))} \mathrm{Pr}_{\mathcal{M},B.i(\rho',x)}(\psi) & \text{if } \bowtie \in \{\geq, >\} \\ \sup\limits_{x \in supp(\pi^i_B(\rho'))} \mathrm{Pr}_{\mathcal{M},B.i(\rho',x)}(\psi) & \text{if } \bowtie \in \{<, \leq\}. \end{cases}$$

Intuitively, given that the action strategy in the system is fixed as σ_{ind}, there is no need to consider agents' strategic abilities, and hence the probability of a given formula ψ being satisfied can be uniquely determined, which simplifies the definitions above.

8.3 A Single-Agent Special Case

The semantics can be further simplified if we work with the single-agent case, in which case we can employ the belief ASMAS construct $\beta_A(\mathcal{M})$ as given in Section 7.2. Recall that this approach explores belief states (distributions over states of $\mathcal{M}$) and that δ_A is a mapping from paths in $\mathcal{M}$ to paths in belief ASMAS $\beta_A(\mathcal{M})$. Therefore, to evaluate belief and trust we can directly employ current belief $last(\delta_A(\rho))$, instead of working with belief distribution be_A defined on finite paths ρ.

Here we only present the semantics of the belief and trust operators.

Definition 8.7. Let $\mathcal{M}$ be an ASMAS satisfying Assumption 2 and ρ a finite path in $\mathcal{M}$. The semantics of the belief and trust operators can be defined on $\beta_A(\mathcal{M})$ as follows.

- $\mathcal{M}, \rho \models \mathbb{B}_A^{\bowtie q} \psi$ if

$$E_{\mathrm{last}(\delta_A(\rho))}[\mathrm{Sat}_\psi] \bowtie q,$$

  where the function $\mathrm{Sat}_\psi : S \to [0, 1]$ is such that $\mathrm{Sat}_\psi(s) = 1$ if $\mathcal{M}, s \models \psi$ and 0 otherwise.
- $\mathcal{M}, \rho \models \mathbb{CT}_{A,B}^{\bowtie q} \psi$ for $\bowtie \in \{\geq, >\}$ if

$$E_{\mathrm{last}(\delta_A(\rho))}[V_{\mathbb{CT}, \mathcal{M}, B, \psi}] \bowtie q$$

  where the function $V_{\mathbb{CT}, \mathcal{M}, B, \psi} : S \to [0, 1]$ is such that

$$V_{\mathbb{CT}, \mathcal{M}, B, \psi}(s) = \sup_{x \in \omega_B^i(s)} \mathrm{Pr}_{\mathcal{M}, B.i(s,x)}(\psi)$$

  and if $\bowtie$ is $\leq$ or $<$ we replace sup with inf in the above.
- $\mathcal{M}, \rho \models \mathbb{DT}_{A,B}^{\bowtie q} \psi$ if for $\bowtie \in \{\geq, >\}$ if

$$E_{\mathrm{last}(\delta_A(\rho))}[V_{\mathbb{DT}, \mathcal{M}, B, \psi}^{\bowtie}] \bowtie q$$

  where the function $V_{\mathbb{DT}, \mathcal{M}, B, \psi}^{\bowtie} : S \to [0, 1]$ is such that

$$V_{\mathbb{DT}, \mathcal{M}, B, \psi}^{\bowtie}(s) = \inf_{x \in \pi_B^i(\rho')} \mathrm{Pr}_{\mathcal{M}, B.i(s,x)}(\psi)$$

  for ρ' such that $\delta_A(\rho') = \delta_A(\rho)$ and $\mathrm{last}(\rho') = s$; if $\bowtie$ is $\leq$ or $<$ we replace inf with sup in the above.

Note that A's belief in ψ is evaluated as an expectation of satisfaction of ψ computed in the current belief state. A's competence trust in B's ability to perform ψ reduces to the expectation of the optimal probability of completing ψ, over all legal changes of B's intention. Similarly, to evaluate A's disposition trust in B's willingness to perform ψ we compute the expectation of the optimal probability of completing ψ, over all possible changes of B's intention π_B^i.

Remark 12. The sets of possible intentions $\pi_B^i(\cdot)$ of agent B can be determined for a given belief state with pro-attitude synthesis defined in Section 9.

8.4 Other Trust Notions

In [17], another concept related to trust called *dependence* is introduced, whose interpretation is that agent A needs agent B (strong dependence), or is better off relying on B (weak dependence). We now propose how those two concepts can be formalised in an ASMAS under Assumption 2.

Definition 8.8. Let $\mathcal{M}$ be an ASMAS satisfying Assumption 2 and ρ a finite path in $\mathcal{M}$. We introduce operators $\mathbb{ST}_{A,B}^{\bowtie q}$ and $\mathbb{WT}_{A,B}^{\bowtie}$ to express strong and weak dependence, with the semantics given by:

- $\mathcal{M}, \rho \models \mathbb{ST}_{A,B}^{\bowtie q} \psi$ if $\mathcal{M}, \rho \models (\neg\mathbb{B}_A^{\bowtie q}\psi \wedge \mathbb{CT}_{A,B}^{\bowtie q}\psi) \vee (\mathbb{B}_A^{\bowtie q}\psi \wedge \neg\mathbb{CT}_{A,B}^{\bowtie q}\psi)$
- $\mathcal{M}, \rho \models \mathbb{WT}_{A,B}^{\bowtie} \psi$ if

$$E_{\mathrm{be}_A}[V_{\mathcal{M}, A, B, \psi}] \bowtie E_{\mathrm{be}_A}[V_{\mathbb{CT}, \mathcal{M}, A, \psi}^{\bowtie}],$$

  where the function $V_{\mathcal{M}, A, B, \psi} : \mathrm{FPath}^{\mathcal{M}} \to [0, 1]$ is such that

$$V_{\mathcal{M}, A, B, \psi}(\rho') = \sum_{x \in \omega_B^i(last(\rho'))} ip_{A,B}(last(\rho'))(x) \cdot \mathrm{Pr}_{\mathcal{M}, B.i(\rho', x)}(\psi).$$

Intuitively, $\mathbb{ST}_{A,B}^{\bowtie q} \psi$ states that, according to agent A's beliefs, either (i) the probability of ψ will not be as required unless agent B takes on an appropriate intention, or (ii) the probability of ψ will be as required unless B takes on a certain intention. In other words, strong dependence of A on B over ψ means that B can control the probability of future satisfaction of ψ. On the other hand,

$\mathbb{WT}_{A,B}^{\bowtie}\psi$ states that, according to A's beliefs and preferences, intentional changes of agent B can bring about better results than any of the available intentional changes of agent A.

We note that alternative definitions of the above notions are possible. For example, we could introduce a non-probabilistic version of strong dependence, based solely on the $\mathbb{C}_A$ operator. We could define

$$\mathcal{M}, \rho \models \mathbb{ST}_{A,B}\phi \text{ if } \mathcal{M}, \rho \models \mathbb{C}_B\overline{\mathbb{C}_A}\neg\phi \wedge \mathbb{C}_B\mathbb{C}_A\phi.$$

Intuitively, here $\mathbb{ST}_{A,B}\phi$ means that B can take on an intention such that, regardless of what intention A takes on, ϕ will not be satisfied, and B can take on another intention, such that A can take on an intention which ensures that ϕ holds. In other words, B can either make it impossible for A to achieve ϕ, or it can allow A to make ϕ true. We can view the above definition as a more objective, absolute interpretation of strong dependence, whereas the probabilistic version represents a subjective notion, influenced by agent's understanding of the system execution. Depending on the use case, either variant of the strong dependence operator might be employed.

Weak dependence can also be interpreted in a different way, especially in systems equipped with a reward structure. The intuitive meaning of the expression *better off* used in the informal definition of weak dependence relates to some implicit notion of agent's well being. Definition 8.8 takes the probability of satisfying a given formula as agent's reward. However, in presence of a reward structure, we could allow ψ to include a reward operator and adapt the definitions of the V-functions to compute expected reward.

Example 8.9. In the trust game, Bob's financial situation is dependent on Alice's actions. We can formally express that in PRTL* with the following formula, where $profit_{Bob}$ is an atomic proposition, true in states in which Bob has more money than in the initial state:

$$\mathbb{ST}_{Bob,Alice} \forall \Diamond profit_{Bob}.$$

It expresses that Bob needs Alice's cooperation to ensure that, at some point in the future, he will have more money than when he started. In fact, Alice's hopes of making a profit depend on Bob's cooperation, which we could express by a formula very similar to the one above.

We remark also that we could equip the trust game with a reward structure that reflects agents' payoffs. We could then express that Bob is weakly dependent on Alice to maximise his payoff with the following formula:

$$\mathbb{WT}_{Bob,Alice} R^{=?}[\Diamond end],$$

where $R^{=?}[\Diamond end]$ returns the expected reward accumulated before the game ends.

8.5 Trust for Systems with Sure Beliefs

We now consider a special class of systems where agent A has sure beliefs, i.e., for all finite paths ρ, we have either $be_A(\rho) = 1$ or $be_A(\rho) = 0$. Intuitively, agent A is sure about the current system state and the execution history. The following theorem shows that, in such systems, agent A's beliefs and trusts can be expressed with other operators.

THEOREM 8.10. *For ASMAS $\mathcal{M}$ in which agent A has sure beliefs, the following equivalences hold for any finite path ρ, formula ψ, and agent $B \neq A$:*

- $\mathcal{M}, \rho \models \mathbb{B}_A^{\bowtie q}\psi$ *if and only if* $\mathcal{M}, \rho \models \mathsf{P}^{\bowtie q}\psi$,
- $\mathcal{M}, \rho \models \mathbb{CT}_{A,B}^{\bowtie q}\psi$ *if and only if* $\mathcal{M}, \rho \models \mathbb{C}_B\mathsf{P}^{\bowtie q}\psi$,
- $\mathcal{M}, \rho \models \mathbb{DT}_{A,B}^{\bowtie q}\psi$ *if and only if* $\mathcal{M}, \rho \models \mathbb{I}_B\mathsf{P}^{\bowtie q}\psi$.

Intuitively, the first equivalence shows that agent A's belief of ψ reduces to computing the probability of satisfying ψ. The other two equivalences concern agent A acting as a trustor and agent B as a trustee. The former indicates that agent A's trust in B's competence in achieving ψ is

equivalent to the existence of B's legal intention which ensures satisfaction of ψ, while the latter means that A's trust in B's disposition towards achieving ψ is equivalent to B ensuring satisfaction of ψ under all possible intentions.

The following proposition shows that the sure-belief system has an equivalent definition as fully observable system. Therefore, Theorem 8.10 also holds for fully observable systems.

PROPOSITION 8.11. *Let $\mathcal{M}$ be an ASMAS and A an agent in $\mathcal{M}$. The following two statements are equivalent:*

- *Agent A has sure beliefs, i.e., for all finite paths ρ, we have either $\mathsf{be}_A(\rho) = 1$ or $\mathsf{be}_A(\rho) = 0$.*
- *Agent A has full observation, i.e., $obs_A(s) = s$ for all $s \in S$.*

9 PRO-ATTITUDE SYNTHESIS

Recall that the cognitive strategies π_A^g and π_A^i are defined on the (possibly infinite) set of paths and may lack a finite representation necessary for model checking. We now formalise an idea, which is informally argued in [7, 50], that the changes of pro-attitudes are determined by the changes of beliefs, but not vice versa.

This will enable us to *synthesize* the cognitive strategies π_A^g and π_A^i by considering agent A's beliefs and trust expressed as formulas of the logic PRTL* and used to constrain pro-attitude changes. In particular, we will associate with each set of goals a condition, expressed as a PRTL* formula, which will *guard* that set of goals. The intuitive interpretation of such an association is that an agent takes a set of goals when the associated condition is satisfied. To guard intentions, we additionally use the agent's own goals, since agents' intentions are influenced by their goals. We note that such a construction closely resembles human reasoning – for example, recall that in Example 5.2 we stated that Bob takes intention *share* when he is an *investor* (i.e., his goal is *investor*) and his belief in Alice being *active* is sufficient. In Example 9.3, we show how to express this statement formally using pro-attitude synthesis, which consists of evaluating Bob's belief in Alice being *active* as a guarding mechanism for his intention to *share* and setting a minimum belief threshold.

Let $\mathcal{L}_A(PRTL^*)$ be the set of formulas of the logic PRTL* that are Boolean combinations of atomic propositions and formulas of the form $\mathbb{B}_A^{\bowtie q}\psi$, $\mathbb{T}_{A,B}^{\bowtie q}\psi$, $\mathbb{B}_A^{\bowtie ?}\psi$ or $\mathbb{T}_{A,B}^{\bowtie ?}\psi$, such that ψ does not contain temporal operators. The formulas $\mathbb{B}_A^{\bowtie ?}\psi$ and $\mathbb{T}_{A,B}^{\bowtie ?}\psi$ denote the quantitative variants of $\mathbb{B}_A^{\bowtie q}\psi$ and $\mathbb{T}_{A,B}^{\bowtie q}\psi$ that yield the actual value of the probability/expectation.

The language $\mathcal{L}_A(PRTL^*)$ allows nested beliefs[2], for example $\mathbb{B}_A^{\geq 0.9}\mathbb{B}_B^{>0.7}\psi$, under the condition that the outermost belief operator is for agent A. The nesting is useful since one may want to reason about, e.g., an agent's belief over the other agent's trust in himself. Moreover, all modal logic formulas in $\mathcal{L}_A(PRTL^*)$ must be in the scope of a belief operator of agent A. This is to ensure that agent A is able to invoke the synthesis with the limited information it has.

Below, we work with an arbitrary ASMAS $\mathcal{M}$ and a set of agents Ags in $\mathcal{M}$.

Definition 9.1. For every agent $A \in Ags$, we define:

- a *goal guard* function $\lambda_A^g : \mathcal{P}(Goal_A) \to \mathcal{L}_A(PRTL^*)$ and
- an *intention guard* function $\lambda_A^i : Int_A \times \mathcal{P}(Goal_A) \to \mathcal{L}_A(PRTL^*)$.

The guard functions are partial functions associating pro-attitudes with conditions expressed using belief and trust formulas that can be evaluated based on a finite execution history. Note

---

[2]Because the semantics of trust is based on the belief function be_A, a trust formula is also called a belief formula in this paper.

how λ_A^i differs from λ_A^g in that it guards intentions by agent's own goals. We call λ_A^g *trivial* if $\lambda_A^g(x) =$ true for all $x \in \mathcal{P}(Goal_A)$. Similarly for λ_A^i.

We recall that $\Omega = \{\langle \omega_A^g, \omega_A^i \rangle\}_{A \in Ags}$ is the cognitive mechanism of $\mathcal{M}$. Additionally, we set $\Lambda = \{\langle \lambda_A^g, \lambda_A^i \rangle\}_{A \in Ags}$ and call it the *guarding mechanism*. In the following, we present an approach to obtain the infinite structure $\Pi = \{\pi_A^g, \pi_A^i\}_{A \in Ags}$, i.e., cognitive strategies of all agents, from finite structures Ω and Λ.

First, for agent $A \in Ags$, we define the *goal evaluation function* $eval_A^g$, such that for $x \subseteq \mathcal{P}(Goal_A)$, $eval_A^g(x) : \text{FPath}^{\mathcal{M}} \to [0, 1]$ is given by:

$$eval_A^g(x)(\rho) = \begin{cases} \mathcal{M}, \rho \models \lambda_A^g(x) & \text{if } x \in \omega_A^g(last(\rho)) \\ 0 & \text{otherwise} \end{cases}$$

Intuitively, $eval_A^g(x)$ evaluates the (probabilistic) satisfiability of agent A's beliefs or trust expressed as $\mathcal{L}_A(PRTL^*)$ formulas given by $\lambda_A^g(x)$. Note that the expressions $\mathbb{B}_A^{\rhd\lhd?}\psi$ and $\mathbb{T}_{A,B}^{\rhd\lhd?}\psi$ return their corresponding probabilistic values, and $\mathcal{M}, \rho \models \mathbb{B}_A^{\rhd\lhd q}\psi$ and $\mathcal{M}, \rho \models \mathbb{T}_{A,B}^{\rhd\lhd q}\psi$ return the value 0 or 1 depending on the verification result.

Similarly, we define the *intention evaluation function* $eval_A^i$, such that for any $x \in Int_A$, $y \subseteq \mathcal{P}(Goal_A)$, $eval_A^i(x, y) : \text{FPath}^{\mathcal{M}} \to [0, 1]$ is given by:

$$eval_A^i(x, y)(\rho) = \begin{cases} \mathcal{M}, \rho \models \lambda_A^i(x, y) & \text{if } x \in \omega_A^i(last(\rho)), \\ 0 & \text{otherwise.} \end{cases}$$

We now use pro-attitude evaluation functions to provide a finite definition of a cognitive strategy.

Definition 9.2. Let $\mathcal{M}$ be an ASMAS with a cognitive mechanism $\Omega = \{\omega_A^g, \omega_A^i\}_{A \in Ags}$ and let $\Lambda = \{\langle \lambda_A^g, \lambda_A^i \rangle\}_{A \in Ags}$ be a guarding mechanism for $\mathcal{M}$. In a finite-memory setting, the *goal strategy* $\pi_A^g : \text{FPath}^{\mathcal{M}} \to \mathcal{D}(\mathcal{P}(Goal_A))$ can be instantiated as follows: for any $x \subseteq Goal_A$ and $\rho \in \text{FPath}^{\mathcal{M}}$, we set

$$\pi_A^g(\rho)(x) = \frac{eval_A^g(x)(\rho)}{\sum_{x \in \omega_A^g(last(\rho))} eval_A^g(x)(\rho)}.$$

Likewise, the *intention strategy* $\pi_A^i : \text{FPath}^{\mathcal{M}} \to \mathcal{D}(Int_A)$ can be instantiated as follows: for any $x \in Int_A$ and $\rho \in \text{FPath}^{\mathcal{M}}$, we set

$$\pi_A^i(\rho)(x) = \frac{eval_A^i(x, y)(\rho)}{\sum_{x \in \omega_A^i(last(\rho))} eval_A^i(x, y)(\rho)},$$

where $y = gs_A(last(\rho))$.

This definition allows us to transform the model $\mathcal{M}$ into $eval(\mathcal{M})$, which, besides the cognitive mechanism Ω, contains the cognitive strategy of each agent. Once $eval(\mathcal{M})$ has been precomputed, model checking a given specification can be carried out on $eval(\mathcal{M})$ instead of $\mathcal{M}$. We refer to this approach as *pro-attitude synthesis before model checking*. Besides formalising the intuition that agent's pro-attitudes are affected by its beliefs, this approach makes the interaction of agents' beliefs and trust possible, without resorting to nested beliefs.

Example 9.3. We show how a cognitive strategy can be constructed from the guarding mechanism for Bob in the trust game $\mathcal{G}$. Since we assume that goals of agents are static throughout the execution of the game, we only concern ourselves with the intention guard function. We recall our informal assumption that Bob's intention will be *share* when he is an investor and his belief in Alice being

active is sufficient, and *keep* otherwise. We formalise it as follows:

$$\lambda^i_{Bob}(share, \{investor\}) = \mathbb{B}^{>0.7}_{Bob} active_{Alice},$$

$$\lambda^i_{Bob}(keep, \{investor\}) = \neg\mathbb{B}^{>0.7}_{Bob} active_{Alice},$$

$$\lambda^i_{Bob}(share, \{opportunist\}) = \bot,$$

$$\lambda^i_{Bob}(keep, \{opportunist\}) = \top,$$

where $active_{Alice}$ holds in states in which Alice's goal is *active* and we used a value 0.7 to represent Bob's belief threshold.

We now synthesize Bob's intention strategy. We let $\rho_1 = s_0 s_1 s_3 s_8$ and $\rho_2 = s_0 s_2 s_5 s_{12}$. We recall from Example 7.1 that $obs_{Bob}(\rho_1) = obs_{Bob}(\rho_2)$ and we let o_1 denote that common observation. By Example 7.3:

$$be_{Bob}(o_1, \rho_1) = 1/7,$$

$$be_{Bob}(o_1, \rho_2) = 6/7.$$

Therefore, since $\mathcal{G}, \rho_1 \models \neg active_{Alice}$ and $\mathcal{G}, \rho_2 \models active_{Alice}$ (below and in what follows, $k \in \{1, 2\}$):

$$\mathcal{G}, \rho_j \models \mathbb{B}^{=6/7}_{Bob} active_{Alice}.$$

Hence

$$eval^i_{Bob}(share, \{investor\})(\rho_k) = 1,$$

$$eval^i_{Bob}(keep, \{investor\})(\rho_k) = 0,$$

and so:

$$\pi^i_{Bob}(\rho_k)(share) = 1,$$

$$\pi^i_{Bob}(\rho_k)(keep) = 0.$$

Likewise, letting $\rho_3 = s_0 s_1 s_4 s_{10}$, $\rho_4 = s_0 s_2 s_6 s_{14}$ and $k \in \{3, 4\}$ one can show that:

$$\pi^i_{Bob}(\rho_j)(share) = 0,$$

$$\pi^i_{Bob}(\rho_j)(keep) = 1.$$

As expected, if Bob is an *opportunist*, he keeps the money for himself. However, when he is an *investor* and Alice invests the money with him, his belief that Alice is *active* is high enough for him to share the profits, hoping to gain Alice's trust. □

Finally, we remark that, while it may not be immediately clear how powerful pro-attitude synthesis is after considering a simple example such as this, full appreciation of it can be gained by applying it to more complex systems. One such system is provided by an iterated trust game, in which Alice and Bob interact with each other repeatedly, in which beliefs and trust evolve dynamically. Rather than specifying cognitive strategies by hand for each iteration, pro-attitude synthesis enables one to encode them in a natural way via guard functions and generate them dynamically. Such an approach scales well to larger systems and closely resembles human reasoning.

Preference Functions Update. During the execution of the system, agents learn new information about other agents by observing their actions. As a result, their understanding of others' behaviour, motivations and cognitive state increases. This new knowledge gained by agents is reflected in their belief function, which could be used to update agents' preference functions. To do that, we extend guarding mechanism of each agent A by introducing guard functions of A over B for every other agent B. In particular, for agents A and B, we define a goal (resp. intention)

guard function $\lambda_{A,B}^g$ (resp. $\lambda_{A,B}^i$) of A over B in a similar way as in Definition 9.1, and then the goal (resp. intention) evaluation function $eval_{A,B}^g$ (resp. $eval_{A,B}^i$) of A over B. Intuitively, $\lambda_{A,B}^g$ (resp. $\lambda_{A,B}^i$) captures A's expectation of how B updates their mental attitudes. In case A does not possess such information, $\lambda_{A,B}^g$ (resp. $\lambda_{A,B}^i$) are trivial. We then define $gp_{A,B}^h$ and $ip_{A,B}^h$ (where h stands for *history*, representing that updated preference functions are history-dependent) by weighting the state-defined preferences with respect to the evaluation functions as follows:

- $gp_{A,B}^h(\rho)(x) = \dfrac{gp_{A,B}(last(\rho))(x) \times eval_{A,B}^g(x)(\rho)}{\sum_{x \subseteq Goal_A} gp_{A,B}(last(\rho))(x) \times eval_{A,B}^g(x)(\rho)}$,

- $ip_{A,B}^h(\rho)(x) = \dfrac{ip_{A,B}(\rho)(x) \times eval_{A,B}^i(x,y)(\rho)}{\sum_{x \in Int_A} ip_{A,B}(\rho)(x) \times eval_{A,B}^i(x,y)(\rho)}$,

where $y = gs_B(last(\rho))$.

Intuitively, the preference functions of agent A over B will be adjusted with respect to its current understanding of the system execution and the knowledge it has acquired throughout, represented by a $\mathcal{L}_A(PRTL^*)$ formula in the guarding functions $\lambda_{A,B}^g$ and $\lambda_{A,B}^i$. The probability space construction from Section 6 can be adapted to work with the updated preference functions p_A^h. Therefore, we have probability spaces for both pro-attitude synthesis and model checking.

Example 9.4. After investing her money with Bob, Alice might be concerned whether she can trust Bob to share his profit with her. Such trust can be formally expressed by a formula $\phi = \mathbb{DT}_{Alice,Bob}^{\geq?}\psi$, where $\psi = \bigcirc(a_{Bob} = share)$ and $\geq$ indicates that Alice wants to find a lower bound on such trust value. We assume that Alice is *active* and let $\rho_1 = s_0 s_2 s_5 s_{12}$, $\rho_2 = s_0 s_2 s_6 s_{14}$, $o_1 = obs_{Alice}(\rho_1) = obs_{Alice}(\rho_2)$ and $k \in \{1, 2\}$. We use Alice's preference function defined in Example 6.4, which we recall to be:

$$gp_{Alice,Bob}(s_k) = \langle investor \mapsto 1/2, opportunist \mapsto 1/2 \rangle.$$

From that, we easily compute Alice's belief upon observing o_1 to be:

$$be_{Alice}(o_1, \rho_1) = 1/2,$$
$$be_{Alice}(o_1, \rho_2) = 1/2.$$

We now compute $V_{\mathbb{DT},\mathcal{G},Bob,\psi}^{\geq}(\rho_k)$. We recall from Example 9.3 that, in state s_{10}, Bob's only possible intention change is $Bob.i.keep$, and, in state s_{14}, his only possible intention change is $Bob.i.share$. Since $\Pr_{\mathcal{G},Bob.i(\rho_k,keep)}(\psi) = 0$ and $\Pr_{\mathcal{G},Bob.i(\rho_k,share)}(\psi) = 1$, we obtain:

$$V_{\mathbb{DT},\mathcal{G},Bob,\psi}^{\geq}(\rho_1) = 0,$$
$$V_{\mathbb{DT},\mathcal{G},Bob,\psi}^{\geq}(\rho_2) = 1.$$

Therefore:

$$\mathcal{G}, \rho_k \models \mathbb{DT}_{Alice,Bob}^{\geq?}\psi = 1/2.$$

Hence, Alice can trust Bob with probability at least 50% in his willingness to share his profit with her.

Note that an alternative way of formulating the trust we wish to compute is to use competence trust instead, in which case $\phi = \mathbb{CT}_{Alice,Bob}^{\geq?}\psi$. Most of the computations stay the same in that case, the exception being the V-functions:

$$V_{\mathbb{CT},\mathcal{G},Bob,\psi}^{\geq}(\rho_1) = 1,$$
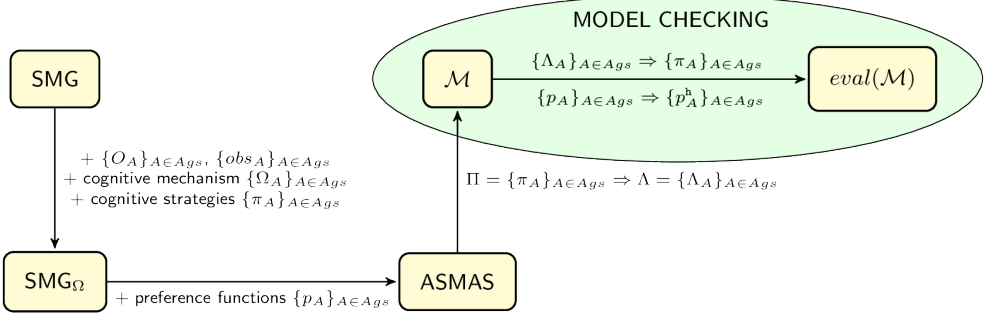$$V_{\mathbb{CT},\mathcal{G},Bob,\psi}^{\geq}(\rho_2) = 1.$$

Fig. 4. Model progression and model checking diagram

Therefore:

$$\mathcal{G}, \rho_k \models \mathbb{CT}^{\geq 1}_{Alice, Bob} \psi.$$

Hence, Alice can trust Bob with certainty on his capability to share his profit with her. We remark that the difference in the value of competence trust and disposition trust is as expected – intuitively, Alice knows for sure that Bob is capable of sharing his profit with her, but she cannot be sure that he is willing to do that.

Finally, we consider Alice's *belief* that Bob will share money with her. We let $\phi = \mathbb{B}^{=?}_{Alice} \psi$ and recall Alice's intention preference function over Bob defined in Example 6.4:

$$ip_{Alice, Bob}(s_i) = \langle share \mapsto 3/4, keep \mapsto 1/4 \rangle \quad \text{for } i \in \{8, 12\},$$
$$ip_{Alice, Bob}(s_i) = \langle share \mapsto 0, keep \mapsto 1 \rangle \quad \text{for } i \in \{10, 14\}.$$

Hence, after Bob settles his intention, Alice's belief is as follows:

$$be_{Alice}(\rho_1 s_{19}) = 3/8,$$
$$be_{Alice}(\rho_1 s_{20}) = 1/8,$$
$$be_{Alice}(\rho_2 s_{21}) = 0,$$
$$be_{Alice}(\rho_2 s_{22}) = 1/2.$$

Therefore:

$$\mathcal{G}, s_0 s_2 \models \mathbb{B}^{=3/8}_{Alice} \psi.$$

Intuitively, due to inaccuracy of Alice's prior knowledge about Bob (encoded in preference functions), her *belief* that Bob will share his profit with her differs from her *trust* towards Bob sharing the money.   □

10 MODEL CHECKING COMPLEXITY

In the next two sections we consider model checking ASMAS against PRTL* formulas. We first present the verification procedure and then analyse its complexity.

We redefine the autonomous stochastic multi-agent system of Definition 6.1 by replacing an infinite structure Π with a finite structure Λ, defined in Section 9.

Our automated verification framework accepts as inputs an ASMAS $\mathcal{M}$ satisfying Assumption 2 and a PRTL* specification formula ϕ. The verification procedure proceeds as follows:

Table 5. Complexity of decidable fragments

BPRTL*	PSPACE-hard
PRTL$^*_1$	PSPACE-complete
PQRTL$^*_1$	PTIME

(1) Perform pro-attitude synthesis on the system $\mathcal{M}$ to obtain $eval(\mathcal{M})$. In the new system, guarding mechanism Λ is replaced by cognitive strategies, Π, of agents. (In this phase, functions be_A are based on preference functions p_A.)
(2) Update the preference functions p_A into p_A^h according to the results of pro-attitude synthesis, as defined in Section 9.
(3) Model check formula ϕ on the system $eval(\mathcal{M})$. (In this phase, the belief functions be_A are based on preference functions p_A^h.)

The checking of expressions $\mathcal{M}, \rho \models \phi$ in Steps 1 and 3 relies on the corresponding preference functions p_A and p_A^h, respectively. However, for simplicity of the notations, we still write them as $\mathcal{M}, \rho \models \phi$. Note that cognitive strategies synthesised in Step 1 generally lack a precise finite representation. Remark 13 discusses a possible approximate solution, while Appendix B presents a restriction on the system which ensures that a finite representation for Π exists. Figure 4 summarises the differences between the models introduced in this paper and outlines the model checking process. Unfortunately, the problem is undecidable in general for both the synthesis step and the model checking step. First of all,

THEOREM 10.1. *The synthesis of pro-attitude functions whose formulas are in the language $\mathcal{L}_A(PRTL^*)$ is undecidable.*

A guarding mechanism $\Lambda = \{\langle \lambda_A^g, \lambda_A^i \rangle\}_{A \in Ags}$ is trivial if, for all agents A, we have $\lambda_A^g(x) = true$ for all $x \in \mathcal{P}(Goal_A)$ and $\lambda_A^i(x, y) = true$ for all $x \in Int_A$ and $y \in \mathcal{P}(Goal_A)$. If the guarding mechanism is trivial then the first two steps can be skipped, and the computation proceeds directly to model checking. In this case, we have

THEOREM 10.2. *Model checking PRTL* is undecidable, even if the guarding mechanism is trivial.*

We note that both of the above problems are undecidable even for formulas concerning beliefs of a single agent. Proofs can be found in Appendix A.

11 DECIDABLE FRAGMENTS

To counter the undecidability results, we explore fragments of the general problem. In this section, we only present proof ideas for the complexity results. The details of the proofs can be found in the Appendix. Table 5 summarises complexity results for the proposed fragments.

11.1 A Bounded Fragment

The bounded fragment, named BPRTL*, allows formulas which 1) do not contain temporal operators $\mathcal{U}$ and $\square$, 2) all $\bigcirc$ operators are immediately prefixed with a probabilistic operator or a branching operator, i.e., in a combination of $P^{\bowtie q} \bigcirc \psi$, $\forall \bigcirc \psi$, or $\exists \bigcirc \psi$, and 3) the nested depth of belief and trust operators is constant. The last constraint is needed to ensure that the complexity is not measured over the nested depth of belief and trust operators. For the upper bound, we have the following result.

THEOREM 11.1. *The complexity of the bounded fragment BPRTL* is in EXPTIME.*

Table 6. Assumptions of decidable fragments

BPRTL*	PRTL$_1^*$	PQRTL$_1^*$
no $\mathcal{U}$ or $\square$ operators	Λ is trivial	$\phi = \square(\psi \Rightarrow \mathsf{P}^{\bowtie q} \Diamond \mathbb{B}_A^{\geq 1} \psi)$
all $\bigcirc$ preceded by P, $\forall$ or $\exists$	$\mathbb{B}_A$, $\mathbb{T}_A$ for single A only	or $\square(\psi \Rightarrow \mathsf{P}^{\bowtie q} \Diamond \mathbb{T}_{A,B}^{\geq 1} \psi)$
constant depth of $\mathbb{B}$, $\mathbb{T}$ nesting	no nesting of $\mathbb{B}$ and $\mathbb{T}$	
	no $\mathbb{B}$ or $\mathbb{T}$ in scope of P	
	constant # of $\mathbb{B}$ and $\mathbb{T}$	
	"expanded system restriction"	

In Appendix B, we present an EXPTIME algorithm which performs the verification procedure from Section 10. It uses boundedness of the logic fragment to limit the length of paths that need to be considered to evaluate a guard or the specification formula. It then constructs an expanded system whose states capture all past observations of all agents, enabling belief and trust formulas to be evaluated locally. Model checking the original system is then reduced to model checking the expanded system, which admits a standard recursive labelling procedure, similar to the one used for PCTL model checking.

For the lower bound, we have the following result.

THEOREM 11.2. *The complexity of the bounded fragment BPRTL* is PSPACE-hard.*

The proof is a reduction from the quantified Boolean formula problem. Given a QBF ϕ, the reduction constructs an ASMAS with two agents, $\forall$ and $\exists$, playing a game of two phases. In the first phase, the agents decide the truth values of the variables (atomic propositions) of ϕ by changing their intentions. Then, in the second phase, the agent $\exists$ attempts to show that a randomly chosen clause of ϕ evaluates to true by proving that one of its literals is true. If they can do that regardless of which clause gets selected, then they have a winning strategy in the game, which is equivalent to ϕ being true. The challenge lies in designing the system so that agents' decisions from phase one are "remembered" until phase two, and devising an appropriate formula to express the existence of a winning strategy for $\exists$. The details of the proof are in Appendix B.

11.2 A Fragment with $\mathcal{U}$ and $\square$ operators

The fragment of this section, named PRTL$_1^*$, works with the $\mathcal{U}$ and $\square$ temporal operators in the specification formula ϕ. With $\mathcal{U}$ and $\square$ operators, the specification formula can express long-run properties about agent's mental attitudes in the system, and therefore this fragment complements the bounded fragment. However, the fragment is subject to other restrictions as follows.

- The guarding mechanism Λ is trivial. That is, the algorithm works with the model checking step without conducting pro-attitude synthesis first.
- The specification formula ϕ is restricted so that 1) it works with a single agent's beliefs and trust, 2) there are no nested beliefs or trust, 3) beliefs and trust cannot be in the scope of a probabilistic operator P, and 4) there is a constant number of belief or trust operators.
- The expanded system, described below, satisfies the restriction that, along any path of the expanded system, the evaluation of belief or trust subformulas for agent A are the same on any two expanded states if the support of the probabilistic distributions over the set of states are the same.

Remark 13. We remark that the pro-attitude synthesis problem is strictly harder than model checking for long-run properties because it needs a finite representation for the set $eval(x)$ of paths. For this reason, this fragment and PRTL_1^* do not handle synthesis. Future work may involve formulating an approach based on order effects [2, 27, 45] to approximate the set $eval(x)$. The focus of this paper is on approaches that can be computed precisely.

The restriction on the expanded system is essential. It results from an investigation into the undecidability, which has established that the undecidable cases arise because of the existence of non-monotonic flows of probability between states in two consecutive visits of the same expanded state. By assuming stability (as we do in this fragment) or monotonicity of the flows, decidability can be achieved. Moreover, we note that this fragment has full expressiveness of LTL. In the following, we assume that it is the agent A's beliefs that the formula ϕ is to describe. We have the following result for the upper bound.

THEOREM 11.3. *The complexity of the fragment* PRTL_1^* *is in PSPACE.*

Given a system $\mathcal{M}$ and a formula ϕ, the algorithm proceeds by first computing those subformulas which do not have belief or trust formulas in their scope. These computations can be done by an adaptation of the usual probabilistic temporal logic model checking. Then, to compute a formula with belief and trust subformulas, we construct an expanded system $\mathcal{M}_A(\phi)$ with the state space $S^\dagger = S \times \mathcal{D}(S) \times R \times \ldots \times R$, where R is the domain of real numbers with fixed precision[3]. Although R is infinite, we need only a finite number of real numbers. The restriction imposed on the system ensures that the system $\mathcal{M}_A(\phi)$ is of size exponential with respect to the size of the system $\mathcal{M}$. However, we do not explicitly construct the system, but instead take an automata-theoretic approach and perform on-the-fly computation by, e.g., two-phase search. The belief and trust formulas can be directly evaluated on the states of the expanded system. The details of the construction are in Appendix C.

We have the following results for the lower bound. Recall that a strongly connected component (SCC) of a graph G is a subgraph of G in which every vertex is reachable from every other vertex.

THEOREM 11.4. *The complexity of the fragment* PRTL_1^* *is PSPACE-hard.*

The proof is a reduction from the language universality problem of nondeterministic finite automata. Given an NFA $A = (Q, q_0, \delta, F)$, we construct a single agent system $\mathcal{M}(A)$ which, starting from an initial state s_0, moves into one of two subsystems $\mathcal{M}_1(A)$ and $\mathcal{M}_2(A)$, with a uniform distribution. The system $\mathcal{M}_1(A)$ simulates the universal language Σ^*, while $\mathcal{M}_2(A)$ simulates the language of A. In other words, every word in Σ^* has a corresponding execution in $\mathcal{M}_1(A)$ and every word in Σ^* that produces a valid run of A has a corresponding execution (one for each unique run) in $\mathcal{M}_2(A)$ which ends in a designated state (one labelled finished) if the corresponding run on A ends in a final state. We define the observation function so that the agent cannot distinguish between paths in $\mathcal{M}_1(A)$ and $\mathcal{M}_2(A)$ corresponding to the same word. This then allows us to prove that the non-universality of the language of A is equivalent to the satisfaction of a carefully designed formula revolving around agent's qualitative belief that the system reaches a designated state. Finally, the reduction is completed by showing that the system $\mathcal{M}(A)$ satisfies the restriction outlined above, which follows by observing that agent's belief stays constant in all SCCs of the expanded system. The details of the proof are in Appendix C.

11.3 A Polynomial Time Fragment

One of the restrictions in the previous fragment is that beliefs and trust cannot be in the scope of a probabilistic operator. In that fragment, this restriction ensures that the complexity is in PSPACE,

---

[3]We assume that each number can be encoded with complexity $O(1)$.

Table 7. Assumptions of the framework

1. Deterministic Behaviour Assumption (Assumption 2)
2. Uniformity Assumptions (Assumptions 1, 3)
3. Transition Type Distinguishability Assumption (Assumption 4)
4. Synchronous perfect recall (Remark 2)
5. PCTL* formulas are evaluated in induced SMGs (Remark 7)
6. Constraints on the language $\mathcal{L}_A(PRTL^*)$
7. Seriality of legal goal and intention functions

because it can be harder to perform quantitative verification of the probabilistic and belief operators. In this section, we partly complement this with a fragment in which the belief or trust operators can be in the scope of a probabilistic operator but need to be qualitative, i.e., almost sureness. We show that this fragment, named $PQRTL_1^*$, is, very surprisingly, polynomial time. Note that, as before, $\mathbb{T}$ ranges over two trust operators $\mathbb{CT}$ and $\mathbb{DT}$.

We consider the model checking step and restrict formulas to be of the form $\Box(\psi \Rightarrow P^{\bowtie q} \Diamond \mathbb{B}_A^{\geq 1} \psi)$ or $\Box(\psi \Rightarrow P^{\bowtie q} \Diamond \mathbb{T}_{A,B}^{\geq 1} \psi)$ such that, in ψ, there are no belief or trust operators and every temporal operator is immediately prefixed with a branching operator, i.e., in the style of CTL. The system $\mathcal{M}$ needs to satisfy the formula $\mathcal{M} \models \Box(\psi \Rightarrow \Box\psi)$, which means that, once ψ holds, it will hold henceforth. The following is the result.

THEOREM 11.5. *The complexity of the fragment $PQRTL_1^*$ is in PTIME.*

We give a polynomial time algorithm in Appendix D. The algorithm first constructs a system $\mathcal{M}^{\#}$ in which two copies of system $\mathcal{M}$ run by synchronising their observations and the second copy avoids the states where ψ is satisfiable. The computation is then to find strongly connected components (SCCs) of $\mathcal{M}^{\#}$ such that a) the formula ψ is satisfiable on some state of the first copy, b) both copies are closed SCCs, and c) the two copies are equivalent in terms of the acceptance probability of words. We show that the checking of formulas $P^{\bowtie q} \Diamond \mathbb{B}_A^{\geq 1} \psi$ or $P^{\bowtie q} \Diamond \mathbb{T}_{A,B}^{\geq 1} \psi$ on states satisfying ψ is equivalent to comparing the reachability probability of these SCCs with the value $1 - q$. The PTIME complexity is due to the fact that the computation of SCCs and checking the three conditions on SCCs can be done in polynomial time; in particular, checking of equivalence of acceptance probability of all words can be done in polynomial time by [49].

12 CONCLUSIONS

The paper proposes an automated verification framework for autonomous stochastic multi-agent systems and specifications given in probabilistic rational temporal logic PRTL*, which includes novel modalities for quantifying and reasoning about agents' cognitive trust. Our model is an extension of stochastic multiplayer games with cognitive reasoning, which specifies how agents' goals and intentions change during system execution, and admits probabilistic beliefs on which the trust concepts are founded. We study computational complexity of the decision problems and show that, although the general problem is undecidable, there are decidable, even tractable, fragments. Their existence is made possible by numerous assumptions and restrictions that we place on our system, which are summarised in Table 7.

As can be seen from the illustrative running example in this paper, the framework is applicable to a wide range of scenarios of human-robot interactions. This includes competitive settings such as the trust game, as well as cooperative scenarios, which are more commonly considered in robotics community, such as a table clearing task [11]. Furthermore, the development of trust sensors (see

[1]) complements our framework very well; they could serve as a source of agent preferences and for validation purposes.

A natural next step following the development of the framework is implementing its techniques in the form of a model checker. To overcome undecidability, a subset of the problem will be considered, in particular, the bounded fragment BPRTL*. Since a finite set of finite paths is sufficient to model check a bounded specification formula, pro-attitude synthesis only needs to consider those finite paths and the resulting cognitive strategies have a finite representation. Preference function update also only needs to be performed on the relevant finite paths. Finally, computing the satisfaction of a specification formula is a recursive procedure that keeps track of the execution history to evaluate belief and trust formulas. Other decidable fragments also provide a basis for implementation, but their strong restrictions make them less practical.

Another interesting direction for future works involves investigating how memory decay can be introduced into the framework. Intuitively, humans tend to remember more recent experiences better and this should be reflected in the semantics of trust. As a side effect, decidability will be achieved if appropriately old memory is discarded. With that, a Bellman operator may be defined which will allow more efficient evaluation of trust.

Finally, it would be interesting to study an axiomatisation for our logic, as well as the satisfiability problem, which we believe to be challenging.

ACKNOWLEDGMENTS

REFERENCES

[1] Kumar Akash, Wan-Lin Hu, Neera Jain, and Tahira Reid. 2018. A Classification Model for Sensing Human Trust in Machines Using EEG and GSR. *CoRR* abs/1803.09861 (2018). arXiv:1803.09861 http://arxiv.org/abs/1803.09861

[2] S.E. Asch. 1946. Forming impression of personality. *The Journal of Abnormal and Social Psychology* 40, 3 (1946), 258–290.

[3] Adnan Aziz, Vigyan Singhal, Felice Balarin, Robert Brayton, and Alberto Sangiovanni-Vincentelli. 1995. It usually works: the temporal logic of stochastic systems. In *Computer Aided Verification*. Springer Berlin/Heidelberg, 155–165.

[4] Christel Baier, Joost-Pieter Katoen, and Kim Guldstrand Larsen. 2008. *Principles of Model Checking*. The MIT Press.

[5] Joyce Berg, John Dickhaut, and Kevin McCabe. 1995. Trust, Reciprocity, and Social History. *Games and Economic Behavior* 10, 1 (1995).

[6] M.E. Bratman. 1987. *Intentions, Plans, and Practical Reason*. Harvard University Press, Massachusetts.

[7] Cristiano Castelfranchi and Fabio Paglieri. 2007. The role of beliefs in goal dynamics: prolegomena to a constructive theory of intentions. *Synthese* 155 (2007), 237–263.

[8] Souymodip Chakraborty and Joost-Pieter Katoen. 2016. On the Satisfiability of Some Simple Probabilistic Logics. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '16)*. ACM, New York, NY, USA, 56–65. https://doi.org/10.1145/2933575.2934526

[9] Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1980. Alternation. *J. ACM* 28, 1 (1980), 114–133.

[10] Krishnendu Chatterjee, Martin Chmelik, and Mathieu Tracol. 2016. What is decidable about partially observable Markov decision processes with ω-regular objectives. *J. Comput. Syst. Sci.* 82, 5 (2016), 878–911. https://doi.org/10.1016/j.jcss.2016.02.009

[11] Min Chen, Stefanos Nikolaidis, Harold Soh, David Hsu, and Siddhartha Srinivasa. 2018. Planning with Trust for Human-Robot Collaboration. *CoRR* abs/1801.04099 (2018). arXiv:1801.04099 http://arxiv.org/abs/1801.04099

[12] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis. 2013. Automatic Verification of Competitive Stochastic Systems. *FMSD* 43, 1 (2013).

[13] E. M. Clarke, O. Grumberg, and D. Peled. 1999. *Model Checking*. The MIT Press.

[14] Brian J. Corbitt, Theerasak Thanasankit, and Han Yi. 2003. Trust and e-commerce: a study of consumer perceptions. *Electronic Commerce Research and Applications* 2, 3 (2003), 203–215.

[15] Christian Dehnert, Sebastian Junges, Joost-Pieter Katoen, and Matthias Volk. 2017. A storm is Coming: A Modern Probabilistic Model Checker. *CoRR* abs/1702.04311 (2017). arXiv:1702.04311 http://arxiv.org/abs/1702.04311

[16] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. 1995. *Reasoning About Knowledge.* MIT Press.

[17] R. Falcone and C. Castelfranchi. 2001. Social trust: A cognitive approach. In *Trust and Deception in Virtual Societies.* Kluwer, 55–90.

[18] Nir Friedman and Joseph Y. Halpern. 1997. Modeling belief in dynamic systems, part I: Foundations. *Artificial Intelligence* 95, 2 (1997), 257–316.

[19] D. Gambetta (Ed.). 1990. *Trust.* Basil Blackwell, Oxford.

[20] Judy Goldsmith and Martin Mundhenk. 1998. Complexity Issues in Markov Decision Processes. In *the Thirteenth Annual IEEE Conference on Computational Complexity.*

[21] P. M. Gollwitzer. 1993. Goal achievement: The role of intentions. *European Review of Social Psychology* 4 (1993), 141–185.

[22] Joseph Y. Halpern and Mark R. Tuttle. 1993. Knowledge, Probability, and Adversaries. *J. ACM* 40, 3 (1993), 917–962.

[23] Hans Hansson and Bengt Jonsson. 1994. A logic for reasoning about time and reliability. *Formal aspects of computing* 6, 5 (1994), 512–535.

[24] Russell Hardin. 2002. *Trust and trustworthiness.* Russell Sage Foundation.

[25] Andreas Herzig, Emiliano Lorini, Jomi F Hübner, and Laurent Vercouter. 2010. A logic of trust and reputation. *Logic Journal of IGPL* 18, 1 (2010), 214–244.

[26] Andreas Herzig, Emiliano Lorini, and Frédéric Moisan. 2013. A simple logic of trust based on propositional assignments. *The Goals of Cognition. Essays in Honor of Cristiano Castelfranchi* (2013), 407–419.

[27] Robin M Hogarth and Hillel J Einhorn. 1992. Order effects in belief updating: The belief-adjustment model. *Cognitive Psychology* 24, 1 (1992), 1–55.

[28] Xiaowei Huang and Marta Kwiatkowska. 2017. Reasoning about cognitive trust in stochastic multiagent systems. In *Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17).*

[29] Audun Jøsang. 2001. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 9, 03 (2001), 279–311.

[30] L. Kagal, T. Finin, and A. Joshi. 2001. Trust-based security in pervasive computing environments. *Computer* 34, 12 (2001), 154 – 157.

[31] John G. Kemeny, J. Laurie Snell, and Anthony W. Knapp. 1967. *Denumerable Markov Chains.* Springer-Verlag.

[32] Karl Krukow, Mogens Nielsen, and Vladimiro Sassone. 2008. Trust models in ubiquitous computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 366, 1881 (2008), 3781–3793.

[33] Benjamin Kuipers. 2016. What is Trust and How Can My Robot Get Some?. In *a Talk at Social Trust in Autonomous Robots, a workshop in Robotics: Science and Systems 2016.* http://qav.comlab.ox.ac.uk/trust_in_autonomy/img/KuipersTrustWorkshop16.pdf

[34] Benjamin Kuipers. 2018. How Can We Trust a Robot? *Commun. ACM* 61, 3 (Feb. 2018), 86–95. https://doi.org/10.1145/3173087

[35] M. Kwiatkowska, G. Norman, and D. Parker. 2011. PRISM 4.0: Verification of Probabilistic Real-time Systems. In *Proc. 23rd International Conference on Computer Aided Verification (CAV'11) (LNCS)*, G. Gopalakrishnan and S. Qadeer (Eds.), Vol. 6806. Springer, 585–591.

[36] Morteza Lahijanian and Marta Kwiatkowska. 2016. Social Trust: a Major Challenge for the Future of Autonomous Systems. In *AAAI Fall Symposium on Cross-Disciplinary Challenges for Autonomous Systems (AAAI Fall Symposium).* AAAI, AAAI Press.

[37] Dave Lee. 2016. US opens investigation into Tesla after fatal crash. *British Broadcasting Corporation (BBC) News* (Jul. 2016). http://www.bbc.co.uk/news/technology-36680043 [Online; posted 1-July-2016; http://www.bbc.co.uk/news/technology-36680043].

[38] John D Lee and Katrina A See. 2004. Trust in automation: Designing for appropriate reliance. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 46, 1 (2004), 50–80.

[39] Roger C Mayer, James H Davis, and F David Schoorman. 1995. An integrative model of organizational trust. *Academy of management review* 20, 3 (1995), 709–734.

[40] John-Jules Ch. Meyer, Jan Broersen, and Andreas Herzig. 2014. BDI Logics. In *Handbook of Epistemic Logic.* College Publications.

[41] J-J Ch Meyer, Wiebe van der Hoek, and Bernd van Linder. 1999. A logical approach to the dynamics of commitments. *Artificial Intelligence* 113, 1 (1999), 1–40.

[42] Azaria Paz. 1971. *Introduction to probabilistic automata (Computer science and applied mathematics).* Academic Press.

[43] Judea Pearl. 1994. Probabilistic semantics for nonmonotonic reasoning: a survey. In *KR 1994.*

[44] Anand S. Rao and Michael P. Georgeff. 1991. Modeling Rational Agents within a BDI-Architecture. In *KR 1991.*

[45] Frank E. Ritter, Josef Nerb, Erno Lehtinen, , and Timothy O'Shea. 2007. *In Order to Learn: How the sequence of topics influences learning.* Oxford University Press.

[46] Klaus Schild. 2000. On the Relationship between BDI Logics and Standard Logics of Concurrency. *Autonomous Agents and Multi-Agent Systems* (2000), 259 – 283.
[47] Tina Setter, Andrea Gasparri, and Magnus Egerstedt. 2016. Trust-Based Interactions in Teams of Mobile Agents. In *American Control Conference*. 6158–6163.
[48] Nicholas Sweet, Nisar R Ahmed, Ugur Kuter, and Christopher Miller. 2016. Towards Self-Confidence in Autonomous Systems. In *AIAA Infotech@ Aerospace*. 1651–1652.
[49] Wen-Guey Tzeng. 1992. A polynomial-time algorithm for the equivalence of probabilistic automata. *SIAM J. Comput.* 21, 2 (1992), 216 – 227.
[50] Wiebe van der Hoek, Wojciech Jamroga, and Michael Wooldridge. 2007. Towards a theory of intention revision. *Synthese* 155, 2 (2007), 265–290.

A UNDECIDABILITY OF THE GENERAL PROBLEM

The undecidability proof is by a reduction from the emptiness problem and strict emptiness problem of probabilistic automata, both known as undecidable problems [42]. A probabilistic automaton PA is a tuple $(Q, A, (M_a)_{a \in A}, q_0, F)$, where

- Q is a finite set of states,
- q_0 is the initial state,
- $F \subseteq Q$ is a set of accepting states,
- A is the finite input alphabet, and
- $(M_a)_{a \in A}$ is the set of transition matrix.

For each $a \in A$, $M_a \in [0, 1]^{Q \times Q}$ defines transition probabilities, such that given $q, q' \in Q$, $M_a(q, q')$ is the probability that q makes a transition to q' when a is the input. For every $q \in Q$ and $a \in A$, we have $\sum_{q' \in Q} M_a(q, q') = 1$. Plainly, given a state q, an input a makes a transition to a distribution on Q, and we further extend M_a to be a transformer from distributions to distributions. Let $\mathcal{D}(Q)$ be the set of all probabilistic distributions on the set Q. Given $\Delta \in \mathcal{D}(Q)$, we write $M_a(\Delta)$ for the distribution transformed from Δ by a, such that for all $q' \in Q$, $M_a(\Delta)(q') = \sum_{q \in supp(\Delta)} \Delta(q) \cdot M_a(q, q')$. Given $w = a_1 \cdot a_2 \cdot \ldots \cdot a_n \in A^*$, we write M_w for the function $M_{a_n} \circ M_{a_{n-1}} \circ \cdots \circ M_{a_1}$ (we assume function application is right associative).

The emptiness problem of a probabilistic automata is defined as follows: Given a probabilistic automaton $PA = (Q, A, (M_a)_{a \in A}, q_0, F)$ and $\epsilon \in [0, 1]$, decide whether there exists a word w such that $M_w(\Delta_0)(F) \geq \epsilon$, where $\Delta_0(q_0) = 1$ and $\Delta_0(q) = 0$ for $q \in Q \setminus \{q_0\}$. Replacing '$\geq$' by a strict inequality '$>$' yields the strict emptiness problem. Both problems are known to be undecidable.

Pro-attitude Synthesis is Undecidable. First of all, we show that the synthesis of pro-attitude functions in which the formulas are in the language $\mathcal{L}_A(PRTL^*)$ is undecidable. Formally, let $PA = (Q, A, (M_a)_{a \in A}, q_0, F)$ be a probabilistic automaton and $AP = \{\text{final}, \text{found}\}$ be a set of atomic propositions. We can then construct an ASMAS $\mathcal{M}(PA) = (Ags, S, S_{init}, Act_1, T, L, O_1, obs_1, \Omega_1, \pi_1, p_1)$ such that

- $Ags = \{1\}$, i.e., this is a single-agent system,
- $S = A \times Q \times \{1, 2\}$,
- $S_{init} = \{(a, q_0, 1)\}$ for some $a \in A$,
- $Act_1 = \{\tau\}$,
- the transition relation is as follows for $k \in \{1, 2\}$:

$$T((a, q, k), \tau, (a', q', k)) = \frac{M_{a'}(q, q')}{\sum_{a' \in A} \sum_{q' \in Q} M_{a'}(q, q')}$$

- found $\in L((a, q, 2))$ for $a \in A$ and $q \in Q$, and final $\in L((a, q, k))$ for $a \in A$, $q \in F$, and $k \in \{1, 2\}$.

Intuitively, the system consists of two subsystems, indexed with the third component of the states (i.e., 1 or 2), running in parallel without any interaction. The observation is defined as follows:

- $O_1 = A$, and
- $obs_1((a, q, k)) = a$ for all $(a, q, k) \in S$.

The moving from the first subsystem to the second subsystem is done by agent 1's intentional changes, which are guarded with a testing on agent 1's beliefs. We only define relevant intentional attitudes in Ω_1 as follows.

- $Int_1 = \{x_1, x_2\}$,

As defined in Section 9, the cognitive strategy π_1 can be obtained from Ω_1 and Λ, with the latter defined as follows for the intentional strategy.

- $\lambda_1^i(x_2) = \mathbb{B}_1^{\geq (>)\epsilon} \mathit{final}$ for some $\epsilon \in [0, 1]$ and $\lambda_1^i(x_1) = \mathit{true}$

The preference functions p_1, the goal attitude $Goal_1$ in Ω_1, and the goal guards λ_1^g are not defined, because they are not used in this reduction. The initial distribution μ_0 is a Dirac distribution over the initial states S_{init}. Therefore, given a PA and a number $\epsilon \in [0, 1]$, the (strict) emptiness of the problem $M_w(\Delta_0)(F) \geq (>)\epsilon$ is equivalent to checking whether $\mathcal{M}(PA) \models \exists \Diamond \overline{\mathbb{I}_1} \mathit{found}$.

Model Checking PRTL* without Pro-Attitudes is Undecidable. In the following, we show that model checking PRTL* is also undecidable, for systems where the guarding mechanism Λ is trivial. Formally, let PA be as before and $AP = \{final\}$ be a set of atomic propositions. We then construct an ASMAS $\mathcal{M}(PA) = (Ags, S, S_{\text{init}}, Act_1, T, L, O_1, obs_1, \Omega_1, \pi_1, p_1)$ such that

- $Ags = \{1\}$, i.e., this is a single-agent system,
- $S = A \times Q$,
- $S_{\text{init}} = \{(a, q_0)\}$ for some $a \in A$, with the initial distribution μ_0 being a Dirac distribution over S_{init},
- $Act_1 = \{\tau\}$,
- the transition relation T is as follows:

$$T((a, q), \tau, (a', q')) = \frac{M_{a'}(q, q')}{\sum_{a' \in A} \sum_{q' \in Q} M_{a'}(q, q')}$$

- $final \in L((a, q))$ for $a \in A$ and $q \in F$.

Agent 1's observation is defined as

- $O_1 = A$, and
- $obs_1((a, q)) = a$ for all $(a, q) \in S$.

We do not need the functions Ω_1, π_1, p_1 in this reduction. Given a PA and a number $\epsilon \in [0, 1]$, the (strict) emptiness of the problem $M_w(\Delta_0)(F) \geq (>)\epsilon$ is equivalent to checking whether $\mathcal{M}(PA) \models \exists \Diamond \mathbb{B}_1^{\geq (>)\epsilon} final$.

B　A DECIDABLE FRAGMENT OF BOUNDED LENGTH

As shown in Section A, the automated verification problem defined in Section 10 is undecidable in general. In this section, we present a fragment of the problem whose computational complexity falls between PSPACE and EXPTIME. Note that in the following discussions on decidable fragments (i.e., Appendix B, C, and D), we only consider dispositional trust formula $\mathbb{DT}_{A,B}^{\triangleright \triangleleft q} \psi$. The competence trust formula $\mathbb{CT}_{A,B}^{\triangleright \triangleleft q} \psi$ can be handled in a similar way.

Bounded Fragment. The bounded fragment works with specification formulas ϕ which do not contain temporal operators $\mathcal{U}$ and $\square$, all $\bigcirc$ operators are immediately prefixed with a probabilistic operator or a branching operator, i.e., in a combination of $\mathsf{P}^{\triangleright \triangleleft q} \bigcirc \psi$, $\forall \bigcirc \psi$, or $\exists \bigcirc \psi$, and the nested depth of belief and trust operators is constant. We remark that the specification formula ϕ can be extended to include subformulas of the form $\mathsf{P}^{\triangleright \triangleleft q}(\psi_1 \mathcal{U}^{\leq k} \psi_2)$, $\forall(\psi_1 \mathcal{U}^{\leq k} \psi_2)$, or $\exists(\psi_1 \mathcal{U}^{\leq k} \psi_2)$. Moreover, the restriction on nested temporal operators can be relaxed by taking the bounded semantics for LTL. We focus on the simpler syntax to ease the notation.

Let $d(\phi)$ be the maximal length of the paths that are needed for the specification formula ϕ. Specifically:

- $d(p) = 0$,
- $d(\phi_1 \vee \phi_2) = \max\{d(\phi_1), d(\phi_2)\}$,
- $d(\neg \psi) = d(\forall \psi) = d(\mathsf{P}^{\triangleright \triangleleft q} \psi) = d(\mathbb{B}_A^{\triangleright \triangleleft q} \psi) = d(\psi)$, and
- $d(\bigcirc \psi) = d(\mathbb{G}_A \psi) = d(\mathbb{I}_A \psi) = d(\mathbb{C}_A \psi) = d(\mathbb{DT}_{A,B}^{\triangleright \triangleleft q} \psi) = d(\psi) + 1$.

B.1 Upper Bound

The algorithm proceeds in three steps according to the verification procedure in Section 10.

B.1.1 Pro-attitude Synthesis. The purpose of the synthesis is to realise the guarding mechanism, i.e., for every $x \in \mathcal{P}(Goal_A)$ and $y \in Int_A$, compute an equivalent representation for the formulas $\lambda^g_A(x)$ and $\lambda^i_A(x, y)$ for all agents A. Here we only consider formulas of the forms $\mathbb{B}^{\blacktriangleright\triangleleft q}_A \phi$ and $\mathbb{DT}^{\blacktriangleright\triangleleft q}_{A,B}\phi$, and claim that formulas of the other forms $\mathbb{B}^{=?}_A \phi$ and $\mathbb{DT}^{=?}_{A,B}\phi$ can be handled by adapting the technique slightly. Without loss of generality, we let $x \in \mathcal{P}(Goal_A)$. By the constraint of the language $\mathcal{L}_A(PRTL^*)$ that no temporal operator can be in the scope of a belief operator, the satisfiability of $M, \rho \models \lambda^g_A(x)$ depends only on those paths of length $|\rho|$. Moreover, by the semantics of the language $PRTL^*$, model checking the bounded specification formula ϕ requires only those paths of length not greater than $\mathsf{d}(\phi) + 1$. Therefore, for this fragment, the synthesis is equivalent to finding a set of paths $eval(x, \mathsf{d}(\phi))$, where

$$eval(x, k) = \{\rho \in \text{FPath}^{\mathcal{M}} \mid |\rho| \le k, x \in Goal_A(last(\rho)), \mathcal{M}, \rho \models \lambda^g_A(x)\}$$

for $1 \le k \le \mathsf{d}(\phi)$.

Given a state $s \in S$, we write $obs(s) = (obs_1(s), ..., obs_n(s))$ for the tuple of agents' observations at state s. For $0 \le k \le \mathsf{d}(\phi) - 1$, we let $O^k = \{\bot\} \cup (O_1 \times ... \times O_n)$ be the set of possible observations at time k, where $\bot$ denotes that agents have not made any observations at time k. The state space of the following expanded system $\mathcal{M}^{\#}$ is $S \times O^0 \times ... \times O^{\mathsf{d}(\phi)-1}$. For a state $s^{\#} = (s, o^0, ..., o^{\mathsf{d}(\phi)-1})$ of $\mathcal{M}^{\#}$, we write $L_{\mathsf{M}}(s^{\#})$ for original state s and $L^k(s^{\#})$ with $0 \le k \le \mathsf{d}(\phi) - 1$ for the observations at time k, i.e., $L^k(s^{\#}) = o^k \in O^k$. Moreover, for a tuple of observations $L^k(s^{\#})$, we write $L^k_A(s^{\#})$ for agent A's observation. These notations can be extended to work with a sequence of states whenever reasonable, e.g., $L_{\mathsf{M}}(s^{\#}_0 s^{\#}_1 ...) = L_{\mathsf{M}}(s^{\#}_0) L_{\mathsf{M}}(s^{\#}_1)...$, etc.

Given $\mathcal{M} = (Ags, S, S_{\text{init}}, \{Act_A\}_{A \in Ags}, T, L, \{O_A\}_{A \in Ags}, \{obs_A\}_{A \in Ags}, \{\Omega_A\}_{A \in Ags}, \{\pi_A\}_{A \in Ags}, \{p_A\}_{A \in Ags})$, we construct a system $\mathcal{M}^{\#} = (Ags, S^{\#}, S^{\#}_{\text{init}}, \{Act_A\}_{A \in Ags}, \{T^{\#}_k\}_{0 \le k \le \mathsf{d}(\phi)-1}, L^{\#}, \{O^{\#}_A\}_{A \in Ags}, \{obs^{\#}_A\}_{A \in Ags}, \{\Omega^{\#}_A\}_{A \in Ags}, \{\pi^{\#}_A\}_{A \in Ags}, \{p^{\#}_A\}_{A \in Ags})$ such that

- $S^{\#} = S \times O^0 \times ... \times O^{\mathsf{d}(\phi)-1}$,
- for $s^{\#} \in S^{\#}$, $s^{\#} \in S^{\#}_{\text{init}}$ whenever $\mu^{\#}_0(s^{\#}) > 0$, where $\mu^{\#}_0(s^{\#}) = \mu_0(L_{\mathsf{M}}(s^{\#}))$ if $L^0(s^{\#}) = obs(L_{\mathsf{M}}(s^{\#}))$ and $L^k(s^{\#}) = \bot$ for $1 \le k \le \mathsf{d}(\phi) - 1$, and $\mu^{\#}_0(s^{\#}) = 0$ otherwise,
- $T^{\#}_k(s^{\#}_1, a, s^{\#}_2) = T(L_{\mathsf{M}}(s^{\#}_1), a, L_{\mathsf{M}}(s^{\#}_2))$ if
  - $L^j(s^{\#}_2) = L^j(s^{\#}_1)$ for all $0 \le j \le k$,
  - $L^{k+1}(s^{\#}_1) = ... = L^{\mathsf{d}(\phi)-1}(s^{\#}_1) = L^{k+2}(s^{\#}_2) = ... = L^{\mathsf{d}(\phi)-1}(s^{\#}_2) = \bot$, and
  - $L^{k+1}(s^{\#}_2) = obs(L_{\mathsf{M}}(s^{\#}_2))$.
  and $T^{\#}_k(s^{\#}_1, a, s^{\#}_2) = 0$ otherwise,
- $L^{\#}(s^{\#}) = L(L_{\mathsf{M}}(s^{\#}))$,
- $O^{\#}_A$ and $obs^{\#}_A$ are to be defined later,
- for $\Omega^{\#}_A$ with $A \in Ags$, we have $Goal^{\#}_A(s^{\#}) = Goal_A(L_{\mathsf{M}}(s^{\#}))$ and $Int^{\#}_A(s^{\#}) = Int_A(L_{\mathsf{M}}(s^{\#}))$ for all $s^{\#} \in S^{\#}$, and
- for $p^{\#}_A$, we have $gp^{\#}_{A,B}(s^{\#}) = gp_{A,B}(L_{\mathsf{M}}(s^{\#}))$ and $gp^{\#}_{A,B}(s^{\#}) = gp_{A,B}(L_{\mathsf{M}}(s^{\#}))$ for all $A, B \in Ags$.

Intuitively, in the new system $\mathcal{M}^{\#}$, agents' observation history are remembered in the state, and for every time $0 \le k \le \mathsf{d}(\phi) - 1$, a separate transition relation $T^{\#}_k$ is constructed. The transition relation $T^{\#}_k$ maintains the previous observation history up to time k and adds a new observation $L^{k+1}(s^{\#}_2)$ to the next state $s^{\#}_2$.

Before evaluating $\mathcal{L}_A(PRTL^*)$ formulas, we define a new belief function $\text{be}_A^{\#}$ on the constructed system $\mathcal{M}^{\#}$. With this function, we can evaluate belief formulas *locally* instead of resorting to the observation history as we did for the original system $\mathcal{M}$. A sequence $\rho^{\#} = s_0^{\#}...s_{\text{d}(\phi)-1}^{\#}$ of states in the expanded system $\mathcal{M}^{\#}$ is a path if, for all $0 \le k \le \text{d}(\phi) - 1$, one of the following conditions holds:

- $T_k^{\#}(s_k^{\#}, a, s_{k+1}^{\#}) > 0$ for some joint action $a \in Act$.
- there exist $B \in Ags$ and $x \in Goal_B(L_{\mathsf{M}}(s_k^{\#}))$ such that
  - $L_{\mathsf{M}}(s_k^{\#}) \overset{B.g.x}{\longrightarrow}_C L_{\mathsf{M}}(s_{k+1}^{\#})$
  - $L^j(s_{k+1}^{\#}) = L^j(s_k^{\#})$ for all $0 \le j \le k$,
  - $L^{k+1}(s_k^{\#}) = ... = L^{\text{d}(\phi)-1}(s_k^{\#}) = L^{k+2}(s_{k+1}^{\#}) = ... = L^{\text{d}(\phi)-1}(s_{k+1}^{\#}) = \bot$, and
  - $L^{k+1}(s_{k+1}^{\#}) = obs(L_{\mathsf{M}}(s_{k+1}^{\#}))$.
- Similar for a transition on which some agent $B \in Ags$ changes its intention.

The following proposition states that there is a 1-1 correspondence relation between paths in $\mathcal{M}$ and paths in $\mathcal{M}^{\#}$.

PROPOSITION B.1. *For every path $\rho^{\#} = s_0^{\#}...s_{\text{d}(\phi)-1}^{\#}$ in $\mathcal{M}^{\#}$ such that $\mu_0^{\#}(s_0^{\#}) > 0$, we have that $L_{\mathsf{M}}(\rho^{\#})$ is an initialised path of $\mathcal{M}$. On the other hand, for every initialised path ρ in $\mathcal{M}$ whose length is no more than $\text{d}(\phi) + 1$, there exists exactly one path $\rho^{\#}$ in $\mathcal{M}^{\#}$ such that $L_{\mathsf{M}}(\rho^{\#}) = \rho$.*

Note that the transition relation of the constructed system $\mathcal{M}^{\#}$ is acyclic, that is, for every state $s^{\#}$, there does not exist a cyclic path $\rho^{\#} \in \text{FPath}^{\mathcal{M}}(s^{\#})$ such that $s^{\#} = \rho^{\#}(m)$ for some $m \ge 1$. Therefore, we can define $\text{clk}(s^{\#})$ as the time in which the state $s^{\#}$ appears, or formally, $\text{clk}(s^{\#}) = k - 1$ for k the greatest number such that $L^k(s^{\#}) \ne \bot$.

We can now define a probabilistic function $T_{k,A}^{\#}$ for the system $\mathcal{M}^{\#}$ and agent $A \in Ags$ as follows, by considering the preference functions of agent A, where $\text{clk}(s_1^{\#}) = k$ and we write $tp(s_1^{\#}, s_2^{\#})$ for $tp(L_{\mathsf{M}}(s_1^{\#}), L_{\mathsf{M}}(s_2^{\#}))$. We note that $s_1^{\#}s_2^{\#}$ needs to be on some path of $\mathcal{M}^{\#}$.

$$T_{k,A}^{\#}(s_1^{\#}, s_2^{\#}) = \begin{cases} T_k^{\#}(s_1^{\#}, a, s_2^{\#}) & \text{if } tp(s_1^{\#}, s_2^{\#}) = a \\ gp_{A,B}(L_{\mathsf{M}}(s_1^{\#}))(x) & \text{if } tp(s_1^{\#}, s_2^{\#}) = B.g \text{ and } L_{\mathsf{M}}(s_1^{\#}) \overset{B.g.x}{\longrightarrow}_C L_{\mathsf{M}}(s_2^{\#}) \\ ip_{A,B}(L_{\mathsf{M}}(s_1^{\#}))(x) & \text{if } tp(s_1^{\#}, s_2^{\#}) = B.i \text{ and } L_{\mathsf{M}}(s_1^{\#}) \overset{B.i.x}{\longrightarrow}_C L_{\mathsf{M}}(s_2^{\#}) \\ 1 & \text{if } tp(s_1^{\#}, s_2^{\#}) = A.g.x \text{ for some } x \in Goal_A(s_1^{\#}) \\ & \text{or } tp(s_1^{\#}, s_2^{\#}) = A.i.x \text{ for some } x \in Int_A(s_1^{\#}) \end{cases}$$

With the non-acyclic property, we can define for every state $s^{\#}$ a reachability probability $\text{rP}_A(s^{\#})$ as follows with respect to a type t of paths.

- $\text{rP}_A(s^{\#}) = \mu_0^{\#}(s^{\#})$ if $\text{clk}(s^{\#}) = 0$
- $\text{rP}_A(s^{\#}) = \sum_{s_1^{\#} \in S^{\#}, \text{clk}(s^{\#}) = \text{clk}(s_1^{\#})+1} \text{rP}_A(s_1^{\#}) \times T_{k,A}^{\#}(s_1^{\#}, s^{\#}) \times (tp(s_1^{\#}, s^{\#}) = t(\text{clk}(s_1^{\#})))$, if $\text{clk}(s^{\#}) > 0$.

The observation function $obs_A^{\#}$ for agent A is defined as follows:

- $obs_A^{\#}(s_1^{\#}) = obs_A^{\#}(s_2^{\#})$ if $obs_A(L_{\mathsf{M}}(s_1^{\#})) = obs_A(L_{\mathsf{M}}(s_2^{\#}))$, $\text{clk}(s_1^{\#}) = \text{clk}(s_2^{\#})$, and for all $0 \le k \le \text{clk}(s_1^{\#})$, $L_A^k(s_1^{\#}) = L_A^k(s_2^{\#})$.

Based on this, $O_A^{\#}$ contains all possible observations of A. Moreover, we let $obs_A^{\#}(s^{\#})$ to be the set of states that are indistinguishable to agent A at state $s^{\#}$. We define a belief function $\text{be}_A^{\#} : S^{\#} \to [0, 1]$ as follows for agent $A \in Ags$:

$$\text{be}_A^{\#}(s^{\#}) = \frac{\text{rP}_A(s^{\#})}{\sum_{s_1^{\#} \in obs_A^{\#}(s^{\#})} \text{rP}_A(s_1^{\#})}$$

For every path $\rho \in \text{FPath}^{\mathcal{M}}$ in the original system $\mathcal{M}$, we construct the following state $e^{\#}(\rho)$ in the expanded system $\mathcal{M}^{\#}$:

$$e^{\#}(\rho) = (last(\rho), obs(\rho(0)), ..., obs(\rho(|\rho| - 1)), \bot, ..., \bot)$$

We have the following proposition to reduce the probability of Pr_A in $\mathcal{M}$ into rP_A in $\mathcal{M}^{\#}$.

PROPOSITION B.2. *For any path ρ of $\mathcal{M}$ such that $|\rho| \leq \text{d}(\phi) + 1$, we have that $\text{Pr}_A(\text{Cyl}_\rho) = \text{rP}_A(e^{\#}(\rho))$, where Cyl_ρ is the basic cylinder with prefix ρ.*

With this proposition, it is not hard to see that $\text{be}_A(\rho) = \text{be}_A^{\#}(e^{\#}(\rho))$. In the following, we inductively define a satisfiability relation $\mathcal{M}^{\#}, s^{\#} \models \phi$ as follows for ϕ a boolean combination of atomic propositions and belief formulas of the form $\mathbb{B}_A^{\bowtie q}\varphi$ or $\mathbb{DT}_{A,B}^{\bowtie q}\varphi$ such that φ is an atemporal formula.

- $\mathcal{M}^{\#}, s^{\#} \models p$ for $p \in AP$ if $p \in L^{\#}(s^{\#})$
- $\mathcal{M}^{\#}, s^{\#} \models \neg\phi$ if not $\mathcal{M}^{\#}, s^{\#} \models \phi$
- $\mathcal{M}^{\#}, s^{\#} \models \phi_1 \vee \phi_2$ if $\mathcal{M}^{\#}, s^{\#} \models \phi_1$ or $\mathcal{M}^{\#}, s^{\#} \models \phi_2$
- $\mathcal{M}^{\#}, s^{\#} \models \mathbb{B}_A^{\bowtie q}\psi$ if

$$\left(\sum_{obs_A^{\#}(s_1^{\#})=obs_A^{\#}(s^{\#})} (\mathcal{M}^{\#}, s_1^{\#} \models \psi) \times \text{be}_A^{\#}(s_1^{\#}) \right) \bowtie q$$

- $\mathcal{M}^{\#}, s^{\#} \models \mathbb{DT}_{A,B}^{\bowtie q}\psi$ if

$$\left(\sum_{obs_A^{\#}(s_1^{\#})=obs_A^{\#}(s^{\#})} \text{be}_A^{\#}(s_1^{\#}) \times \bigotimes_{s_2^{\#} \in \pi_B^i(s_1^{\#})} (T_{k,A}^{\#}(s_1^{\#}, s_2^{\#}) \times \mathcal{M}^{\#}, s_2^{\#} \models \psi) \right) \bowtie q$$

  where $\otimes \equiv \inf$ if $\bowtie \in \{\geq, >\}$, $\otimes \equiv \sup$ if $\bowtie \in \{\leq, <\}$, and the set $\pi_B^i(s_1^{\#})$ contains states $s_2^{\#}$ such that $s_1^{\#}s_2^{\#}$ is on some path of $\mathcal{M}^{\#}$ and $tp(s_1^{\#}, s_2^{\#}) = B.i$.
- The case of $\mathbb{CT}_{A,B}^{\bowtie q}\psi$ can be done similarly as $\mathbb{DT}_{A,B}^{\bowtie q}\psi$.

Now we have the following theorem to reduce the problem of model checking on the original system $\mathcal{M}$ to the model checking on the expanded system $\mathcal{M}^{\#}$.

THEOREM B.3. $\mathcal{M}, \rho \models \phi$ *if and only if* $\mathcal{M}^{\#}, e^{\#}(\rho) \models \phi$, *for ϕ a formula in the language $\mathcal{L}_A(PRTL^*)$.*

Now we can interpret those belief formulas properly with the above theorem, i.e., for each $x \subseteq Goal_A$ and $1 \leq k \leq \text{d}(\phi)$, we compute the set

$$eval^{\#}(x, k) = \{s^{\#} \in S^{\#} \mid \mathcal{M}^{\#}, s^{\#} \models \lambda_A^g(x), \text{clk}(s^{\#}) \leq k\}$$

of states in the expanded system $\mathcal{M}^{\#}$. It is noted that $eval(x, k) = \{\rho \mid |\rho| = k, \exists s^{\#} \in eval^{\#}(x, k) : s^{\#} = e^{\#}(\rho)\}$. But we do not need to compute $eval(x, k)$ because the following procedure will base on $\mathcal{M}^{\#}$ instead of $\mathcal{M}$.

B.1.2 Preference Function Update. This can be done by following the expressions in Section 9. By the system $\mathcal{M}^{\#}$ and the functions $eval^{\#}(x, k)$, we can have the updated preference functions p_A for all $A \in Ags$. Then we can update the transition functions $T_{k,A}^{\#}(s_1^{\#}, s_2^{\#})$ into $T_{k,A}^{\#,*}(s_1^{\#}, s_2^{\#})$ by substituting p_A with the new one. Based on these, we can update the reachability function rP_A into rP_A^* and the belief function $\text{be}_A^{\#}$ into $\text{be}_A^{\#,*}$.

B.1.3 Model Checking Specification Formula. The model checking algorithm proceeds by labelling subformulas of ϕ on the states in $S^\#$. We use the notation $sat(s^\#, \psi)$ to represent that a formula ψ is labeled on the state $s^\#$. The labelling can be done recursively as follows.

- $sat(s^\#, p)$ for $p \in AP$, if $p \in L^\#(s^\#)$,
- $sat(s^\#, \psi_1 \vee \psi_2)$ if $sat(s^\#, \psi_1)$ or $sat(s^\#, \psi_2)$,
- $sat(s^\#, \neg\psi)$ if not $sat(s^\#, \psi)$,
- $sat(s^\#, \forall \bigcirc \psi)$ if $sat(t^\#, \psi)$ for all $t^\#$ such that $T^\#_k(s^\#, a, t^\#) > 0$,
- $sat(s^\#, \mathsf{P}^{\bowtie q} \bigcirc \psi)$ if $prob(s^\#, \psi) \bowtie q$, where the function $prob(s^\#, \psi)$ is obtained as follows.
  - $prob(s^\#, \psi) = \sum_{t^\# \in S^\#} T^\#_k(s^\#, a, t^\#) \times sat(t^\#, \psi)$.

  Intuitively, $prob(s^\#, \psi)$ is the one-step reachability probability of those states satisfying ψ from a given state $s^\#$. Recall that in the expression $T^\#_k(s^\#, a, t^\#)$, the action a is determined by state $s^\#$.
- $sat(s^\#, \mathbb{B}^{\bowtie q}_A \psi)$ if $(\sum_{obs^\#_A(t^\#)=obs^\#_A(s^\#)} sat(t^\#, \psi) \times \mathsf{be}^{\#,*}_A(t^\#)) \bowtie q$
- $sat(s^\#, \mathbb{DT}^{\bowtie q}_{A,B} \psi)$ if $(\sum_{obs^\#_A(t^\#)=obs^\#_A(s^\#)} \mathsf{be}^{\#,*}_A(t^\#) \times \bigotimes_{u^\# \in \pi^i_B(t^\#)} (T^{\#,*}_{k,A}(t^\#, u^\#) \times sat(u^\#, \psi))) \bowtie q$ where $\otimes \equiv \inf$ if $\bowtie \in \{\geq, >\}$, $\otimes \equiv \sup$ if $\bowtie \in \{\leq, <\}$, and the set $\pi^i_B(t^\#)$ contains states $u^\#$ such that
  - $L_\mathsf{M}(t^\#) \xrightarrow{B.i.x}_C L_\mathsf{M}(u^\#)$ for some $x \in Int_B$ such that $u^\# \in eval^\#(x, \mathsf{clk}(t^\#))$,
  - $L^j(u^\#) = L^j(t^\#)$ for all $0 \leq j \leq k$,
  - $L^{k+1}(t^\#) = \ldots = L^{d(\phi)-1}(t^\#) = L^{k+2}(u^\#) = \ldots = L^{d(\phi)-1}(u^\#) = \bot$, and
  - $L^{k+1}(u^\#) = obs(L_\mathsf{M}(u^\#))$.
- the case of $sat(s^\#, \mathbb{CT}^{\bowtie q}_{A,B} \psi)$ can be done similarly as that of $sat(s^\#, \mathbb{DT}^{\bowtie q}_{A,B} \psi)$.
- $sat(s^\#, \mathbb{G}_A \psi)$ with $\mathsf{clk}(s^\#) = k \leq d(\phi)$ if $sat(t^\#, \psi)$ for all $t^\#$ such that
  - $L_\mathsf{M}(s^\#) \xrightarrow{B.g.x}_C L_\mathsf{M}(t^\#)$ for some $x \subseteq Goal_B$ such that $t^\# \in eval^\#(x, \mathsf{clk}(s^\#))$,
  - $L^j(t^\#) = L^j(s^\#)$ for all $0 \leq j \leq k$,
  - $L^{k+1}(s^\#) = \ldots = L^{d(\phi)-1}(s^\#) = L^{k+2}(t^\#) = \ldots = L^{d(\phi)-1}(t^\#) = \bot$, and
  - $L^{k+1}(t^\#) = obs(L_\mathsf{M}(t^\#))$.
- $sat(s^\#, \mathbb{I}_A \psi)$ with $\mathsf{clk}(s^\#) = k \leq d(\phi)$ follows the similar pattern as that of $sat(s^\#, \mathbb{G}_A \psi)$.
- $sat(s^\#, \mathbb{C}_A \psi)$ with $\mathsf{clk}(s^\#) = k \leq d(\phi)$ if $sat(t^\#, \psi)$ for all $t^\#$ such that
  - $L_\mathsf{M}(s^\#) \xrightarrow{B.i.x}_C L_\mathsf{M}(t^\#)$ such that $x \in Int_A$,
  - $L^j(t^\#) = L^j(s^\#)$ for all $0 \leq j \leq k$,
  - $L^{k+1}(s^\#) = \ldots = L^{d(\phi)-1}(s^\#) = L^{k+2}(t^\#) = \ldots = L^{d(\phi)-1}(t^\#) = \bot$, and
  - $L^{k+1}(t^\#) = obs(L_\mathsf{M}(t^\#))$.

It is noted that the above labelling procedure is done locally for all the belief and trust formulas. We have the following theorem to reduce model checking problem to the above labelling algorithm.

THEOREM B.4. $\mathcal{M} \models \phi$ if and only if $sat(s^\#, \phi)$ for all $s^\#$ such that $\mu_0(s^\#) > 0$.

B.1.4 Analysis of Complexity. For the measurement of the complexity, we use the number of states $|S|$ as the size of the system $\mathcal{M}$, and the depth $d(\phi)$ as the size of the specification formula ϕ. Moreover, we assume that the sets $Goal_A$ and Int_A, for all $A \in Ags$, are polynomial with respect to the size of $\mathcal{M}$, and the size of the formulas in the guarding mechanism, measured with the number of operators, are also polynomial with respect to the size of $\mathcal{M}$. As usual, the set of agents Ags is fixed. Also, because the observation function obs is defined on states, the size of $O_1 \times \ldots \times O_n$ is no more than the size of S.

For the complexity of the fragment, we notice that the size of the expanded system $\mathcal{M}^\#$ is polynomial with respect to the size of $\mathcal{M}$ but exponential with respect to the size of the specification formula ϕ.

- For the pro-attitude synthesis, for every $x \in \bigcup_{A \in Ags} \mathcal{P}(Goal_A) \cup Int_A$, the determination of whether $s^{\#} \in eval^{\#}(x, \mathsf{d}(\phi))$ can be done in EXPTIME.
- The update of preference functions $\{p_A\}_{A \in Ags}$ and the model $\mathcal{M}^{\#}$ can be done in EXPTIME.
- For the model checking of the specification formula, the labelling procedure can be done in EXPTIME.

Putting them together, we have that the computational complexity of the bounded fragment of the problem is in EXPTIME. It is noted that, the exponential is with respect to the size of the specification formula. If the formula is fixed, it is in PTIME.

B.2 Hardness Problem

The lower bound can be obtained by a reduction from the satisfiability problem of the quantified Boolean formula (QBF), which is to determine if a QBF formula is satisfiable. A canonical form of a QBF is

$$\phi = Q_1 x_1 ... Q_n x_n : (l_{11} \lor l_{12} \lor l_{13}) \land ... \land (l_{m1} \lor l_{m2} \lor l_{m3})$$

where $Q_i = \exists$ if i is an odd number and $Q_i = \forall$ otherwise, and $l_{kj} \in \{x_1, ..., x_n, \neg x_1, ..., \neg x_n\}$ for all $1 \le k \le m$ and $1 \le j \le 3$. Without loss of generality, we assume that n is an even number. Every QBF formula can be transformed into this format by inserting dummy variables. For a literal l_{kj}, we write $var(l_{kj})$ for its variable and $sign(l_{kj})$ for its sign. For example, if $l_{kj} = \neg x_1$ then $var(l_{kj}) = x_1$ and $sign(l_{kj}) = neg$, and if $l_{kj} = x_n$ then $var(l_{kj}) = x_n$ and $sign(l_{kj}) = pos$.

The problem can be simulated by a game between two agents $\forall$ and $\exists$. The game consists of two phases: in the first phase, agents decide the values of variables $\{x_1, ..., x_n\}$, and in the second phase, agents evaluate the boolean formula $(l_{11} \lor l_{12} \lor l_{13}) \land ... \land (l_{m1} \lor l_{m2} \lor l_{m3})$. We write $odd(k)$ ($even(k)$) to denote that the number k is odd (even). Figure 5 gives a diagram for the construction. The system has a state space $S = \{x_{kb}, x_{kb}^{ha} \mid 1 \le h \le n, 1 \le k \le n+1, a, b \in \{0, 1\}\} \cup \{c_k, c_k^{ha} \mid 1 \le k \le m, 1 \le h \le n, a \in \{0, 1\}\} \cup \{l_{kj}, l_{kj}^{ha} \mid 1 \le k \le m, j \in \{1, 2, 3\}, 1 \le h \le n, a \in \{0, 1\}\} \cup \{v_{ha} \mid 1 \le h \le n, a \in \{0, 1\}\}$.

Before proceeding to the reduction, we recall notations $is_{\exists}(s)$ and $is_{\forall}(s)$ which return the intention at state s for agents $\exists$ and $\forall$, respectively. Assuming that agents' intentions are encoded in the states make it easier to understand the reduction. Similar for the notations $gs_{\exists}(s)$ and $gs_{\forall}(s)$.

In the first phase, the agents $\exists$ and $\forall$ make their decisions by changing their intentions. The states involved in the first phase are $\{x_{kb}, x_{kb}^{ha} \mid 1 \le h \le n, 1 \le k \le n+1, a, b \in \{0, 1\}\}$. Intuitively, the state x_{kb} represents that the value of variable x_k was set to b, and the superscript ha in the state x_{kb}^{ha} represents that the state "remembers" the value a of variable x_h. We let $Int_{\exists} = \{0, 1, 2, 3\}$, where only 0 and 1 will be used in the first phase, and let $Int_{\forall} = \{0, 1\}$. On a state x_{kb} or x_{kb}^{ha}, we define $is_{\exists}(x_{kb}) = is_{\exists}(x_{kb}^{ha}) = b$ for the case of $odd(k)$, and $is_{\forall}(x_{kb}) = is_{\forall}(x_{kb}^{ha}) = b$ for the case of $even(k)$. We do not define $is_{\exists}(x_{kb})$ and $is_{\exists}(x_{kb}^{ha})$ for the cases of $even(k)$ and $is_{\forall}(x_{kb})$ and $is_{\forall}(x_{kb}^{ha})$ for the case of $odd(k)$, since they will not be used in this proof. Intuitively, the intentional attitude reflects agent's choice. Because the value of variable x_k for $odd(k)$ will be decided by agent $\exists$, we define $Int_{\exists}(x_{kb}) = Int_{\exists}(x_{kb}^{ha}) = \{0, 1\}$, so that agent $\exists$ can choose the value by changing its intention. Similar for $Int_{\forall}(x_{kb}) = Int_{\forall}(x_{kb}^{ha}) = \{0, 1\}$ when $even(k)$.

Since we will test the beliefs of agent $\exists$ in the specification formula, only agent $\exists$'s observations are relevant. We let $obs_{\exists}(x_{kb}) = obs_{\exists}(x_{kb}^{ha}) = b$ for any $1 \le k \le n+1$, i.e., it can see both agents' intentions. However, it is not able to distinguish observations between x_{kb} and x_{kb}^{ha}.

The idea of the first phase is that the states of the form x_{kb} contribute to the main computation in which agents use their intentions to decide the values of the variables under their controls. Every time a decision is made, the execution moves to both the next variable $x_{(k+1)b}$ and a helper
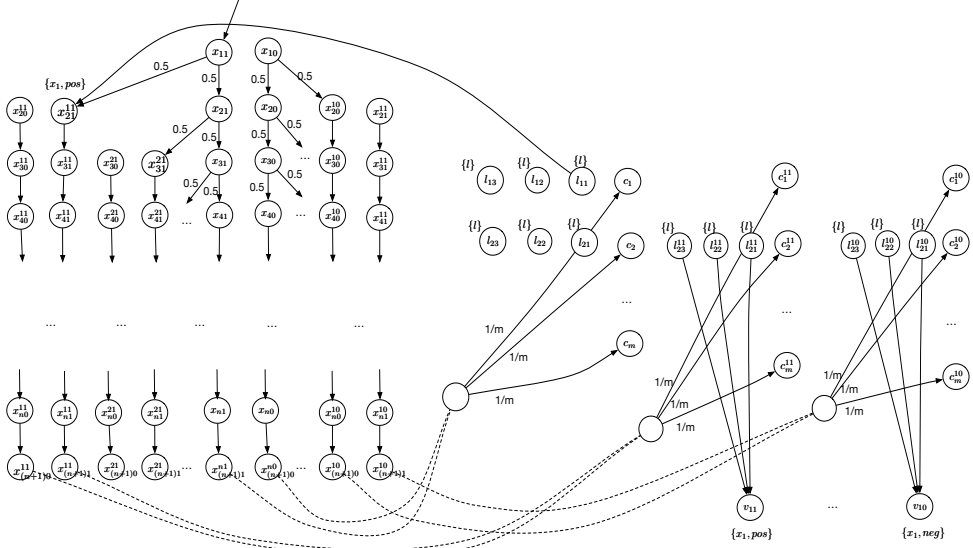
Fig. 5. Diagram for the reduction from QBF problem

computation in which all states are labelled with the fact about the last decision. E.g., in Figure 5, for the first variable x_1, the agent $\exists$ can choose between two states x_{11} and x_{10}, at which the variable x_1 will take different values 1 and 0, respectively. Without loss of generality, if agent $\exists$ chooses x_{11}, then the execution will move into both x_{21} and x_{21}^{11}, where the state x_{21}^{11} remembers that the value of variable x_1 is 1. Therefore, from now on, the execution will be possible on both the main computation from x_{21} and the helper computation from x_{21}^{11}. But the agent $\exists$ can not distinguish these two executions by its observations. Generalising the above, from variable x_1 to x_n, we will need to explore the main computation and the other n helper computations, such that one helper computation is added for each variable.

The execution of the first phase can move into those states $x_{(n+1)b}^{ha}$, where the main computation moves into $x_{(n+1)1}^{n1}$ or $x_{(n+1)0}^{n0}$. By the existence of executions on these states, the system remembers the choices made by the two agents. Then in the second phase, the agent $\exists$ needs to justify that its choices make the formula $(l_{11} \lor l_{12} \lor l_{13}) \land ... \land (l_{m1} \lor l_{m2} \lor l_{m3})$ true whenever the QBF formula is satisfiable. To do this, we let the executions of the first phase move into those states c_k and c_k^{ha} with a uniform distribution. Each of these states represents intuitively a clause of the QBF formula. Then the agent $\exists$ will need to pick a literal to see if it can make the clause satisfiable.

Similar to the first phase, the choice of literals is done by agent $\exists$ changing its intention. For the cognitive mechanism, we have that $is_\exists(c_k) = is_\exists(c_k^{ha}) = 0$ and $Int_\exists(c_k) = Int_\exists(c_k^{ha}) = \{0, 1, 2, 3\}$. That is, agent $\exists$ chooses the literal l_{kx} if its intention is changed into x. Note that, we assume that agent $\exists$ can distinguish the clauses, i.e., $obs_\exists(c_k) = obs_\exists(c_k^{ha}) = c_k$. And it is able to know which clause the current literal belongs to, i.e., $obs_\exists(l_{kj}) = obs_\exists(l_{kj}^{ha}) = c_k$.

After a literal l_{kj} is chosen, the execution will need to move to obtain its value. This is done by a transition into one of the states in the first phase. Let $sign(l_{kj})$ be the sign of the literal and $var(l_{kj})$ be the value of the literal. Without loss of generality, we let $var(l_{kj}) = x_h$ and $sign(l_{kj}) = a$, then the transition from state l_{kj} will move into $x_{(h+1)a}^{ha}$, at which the atomic propositions $\{x_h, a\}$ are labelled. For the state l_{kj}^{ha}, it can only move into the state v_{ha}, at which the atomic propositions $\{x_h, a\}$ are labelled.

Formally, let $AP = \{x_i \mid i \in \{1, ..., n\}\} \cup \{l, pos, neg\}$ be a set of atomic propositions, we have $\mathcal{M}(\phi) = (Ags, S, S_{\text{init}}, \{Act_A\}_{A \in Ags}, T, L, \{O_A\}_{A \in Ags}, \{obs_A\}_{A \in Ags}, \{\Omega_A\}_{A \in Ags}, \{\pi_A\}_{A \in Ags}, \{p_A\}_{A \in Ags})$ such that

- $Ags = \{\exists, \forall\}$,
- $S = \{x_{kb}, x_{kb}^{ha} \mid 1 \le h \le n, 1 \le k \le n+1, a, b \in \{0, 1\}\} \cup \{c_k, c_k^{ha} \mid 1 \le k \le m, 1 \le h \le n, a \in \{0, 1\}\} \cup \{l_{kj}, l_{kj}^{ha} \mid 1 \le k \le m, j \in \{1, 2, 3\}, 1 \le h \le n, a \in \{0, 1\}\} \cup \{v_{ha} \mid 1 \le h \le n, a \in \{0, 1\}\}$,
- $S_{\text{init}} = \{x_{11}\}$ with $\mu_0(x_{11}) = 1$,
- $Act_A = \{\epsilon\}$ for $A \in \{\exists, \forall\}$,
- the transition function is as follows.
  - $T(x_{kb}, x_{(k+1)b}) = T(x_{kb}, x_{(k+1)b}^{kb}) = 0.5$ for $1 \le k \le n-1$, $T(x_{nb}, x_{(n+1)b}) = 1$,
  - $T(x_{kb}^{ha}, x_{(k+1)b}^{ha}) = 1$, for $1 \le k \le n$,
  - $T(x_{(n+1)b}^{ha}, c_k^{ha}) = T(x_{(n+1)b}, c_k) = 1/m$,
  - $T(l_{kj}, x_{(h+1)1}^{h1}) = 1$ for $sign(l_{kj}) = pos$ and $var(l_{kj}) = v_h$,
  - $T(l_{kj}, x_{(h+1)0}^{h0}) = 1$ for $sign(l_{kj}) = neg$ and $var(l_{kj}) = v_h$,
  - $T(l_{kj}^{ha}, v_{ha}) = 1$
- the labelling function is as follows.
  - $L(x_{kj}^{h1}) = \{x_h, pos\}, L(x_{kj}^{h0}) = \{x_h, neg\}$,
  - $L(v_{h1}) = \{x_h, pos\}, L(v_{h0}) = \{x_h, neg\}$,
  - $L(l_{kj}) = l$
- the set of possible observations is as follows.
  - $O_\exists = \{0, 1\} \cup \{c_k \mid 1 \le k \le m\} \cup \{\emptyset\}$ for player $\exists$, and
  - we ignore the definition for player $\forall$ as it is not used in the reduction.
- the partial observation functions are defined as follows for the player $\exists$.
  - $obs_\exists(x_{kb}) = obs_\exists(x_{kb}^{ha}) = b$ for $1 \le k \le n+1$ and $b \in \{0, 1\}$,
  - $obs_\exists(c_k) = obs_\exists(c_k^{ha}) = c_k$ for $1 \le k \le m$,
  - $obs_\exists(l_{kj}) = obs_\exists(l_{kj}^{ha}) = c_k$ for $1 \le k \le m$ and $j \in \{1, 2, 3\}$, and
  - $obs_\exists(v_{ha}) = \emptyset$.
- the set of intentions is as follows.
  - $Int_\exists = \{0, 1, 2, 3\}$ and $Int_\forall = \{0, 1\}$, and
  - we ignore the definition for $Goal_\exists$ and $Goal_\forall$ as they are not used in the reduction.
- the guarding mechanism is trivial.
- the preference functions are defined as follows.
  - $ip_{\exists, \exists}(x_{kb})(0) = ip_{\exists, \exists}(x_{kb})(1) = 0.5$ for $odd(k)$,
  - $ip_{\exists, \forall}(x_{kb})(0) = ip_{\exists, \forall}(x_{kb})(1) = 0.5$ for $even(k)$,
  - $ip_{\exists, \exists}t(c_k)(0) = ip_{\exists, \exists}(c_k)(1) = ip_{\exists, \exists}(c_k)(2) = ip_{\exists, \exists}(c_k)(3) = 0.25$ for $1 \le k \le m$.

In such a system, we have that ϕ is satisfiable if and only if

$$\mathcal{M}(\phi) \models \overline{\mathbb{I}_\exists} AX\ \mathbb{I}_\forall AX \ldots \mathbb{I}_\forall AX\ \overline{\mathbb{I}_\exists} (l \wedge \bigwedge_{i=1}^n ((\mathbb{B}_\exists^{\geq 1} AX(x_i \Rightarrow pos)) \vee (\mathbb{B}_\exists^{\geq 1} AX(x_i \Rightarrow neg)))).$$

Note that in above formula and below, we use AX instead of $\forall\bigcirc$ to avoid confusion with the agents $\forall$ and $\exists$. Intuitively, the prefixes $\overline{\mathbb{I}_\exists} AX$ and $\mathbb{I}_\forall AX$ represent the choosing of variables' values by the agents $\exists$ and $\forall$, respectively. After the last AX, the execution moves to the states c_k and c_k^{ha}. Then, the last $\overline{\mathbb{I}_\exists}$ is for the agent $\exists$ to choose a specific literal, and then justify that it is satisfiable. The former is represented by the presence of a literal l in the formula. The latter is guaranteed by $\bigwedge_{i=1}^n ((\mathbb{B}_\exists^{\geq 1}(x_i \Rightarrow pos)) \vee (\mathbb{B}_\exists^{\geq 1}(x_i \Rightarrow neg)))$. Without loss of generality, we let l_{ha} be the chosen literal and assume $var(l_{ha}) = x_i$. If $sign(x_i) = neg$ then the main computation will reach the state $x_{(h+1)0}^{h0}$ which is labelled with x_h and neg. By construction, we know that if the variable x_i is assigned the value neg by one of the agents then the helper computation with superscript $i0$ is explored, and the helper computation will lead to the state v_{i0} at which the propositions x_i and neg are labelled. On the other hand, the helper computation with superscript $i1$ will not be explored, which means that the state v_{i1} labelled with x_i and pos is not considered possible by the agent $\exists$. Therefore, we have that $\mathbb{B}_\exists^{\geq 1}(x_i \Rightarrow neg)$. Similar argument can be taken to show that $\mathbb{B}_\exists^{\geq 1}(x_i \Rightarrow pos)$ when $sign(x_i) = pos$.

C A DECIDABLE FRAGMENT WITH $\mathcal{U}$ AND $\square$ TEMPORAL OPERATORS

We present a fragment of the problem that is PSPACE-complete. Unlike the bounded fragment in Section B, this fragment works with the $\mathcal{U}$ and $\square$ temporal operators in specification formula ϕ. With $\mathcal{U}$ and $\square$ operators, the specification formula can express long-term properties about agent's mental attitudes in the system, and therefore this fragment complements the bounded fragment. However, the fragment is subject to other restrictions, including the following:

- The guarding mechanism is trivial. That is, the algorithm works with model checking without conducting pro-attitude synthesis first.
- The specification formula ϕ has the restrictions that
  – it works with a single agent's beliefs and trusts,
  – there is no nested beliefs or trusts,
  – beliefs and trusts cannot be in the scope of a probabilistic operator P, and
  – there is a constant number of belief or trust operators.
  We make a remark on these constraints. For the first constraint, that we choose to work with a single agent's beliefs and trusts is because in Section A we have shown that the single-agent fragment is undecidable. The second constraint is imposed to keep the algorithm simple. We conjecture that it can be relaxed. The third and fourth constraints are to keep the complexity in PSPACE. For the probabilistic operator, it requires to compute a set of paths concurrently. We conjecture that the complexity can be in EXPTIME instead of in PSPACE if relaxing this constraint. To complement this, in Section D, we study a fragment in which the probabilistic operator works with qualitative beliefs and trusts. The constraint on the number of belief or trust operators is to keep the expanded system, to be given below, in the size of a polynomial with respect to the formula.
- There is a restriction on the system about agent A's beliefs along the transitions. The restriction, whose details will be given later, is that along any path of an expanded system, the evaluations of belief or trust subformulas are the same on any two expanded states if the support of a probabilistic distribution over the set of states is the same.

Without loss of generality, we assume that it is the agent A's beliefs that the specification formula ϕ is to describe. Note that the fragment has the full expressiveness of LTL.

C.1 Upper Bound

Because the guarding mechanism is trivial, we need only handle the third step of the verification procedure.

Preprocessing Sub-Formulas in the Scope of Belief Operators. The first step of the algorithm is to pre-process subformulas in the scope of belief or trust operators. Because there is no nested beliefs, these subformulas do not contain any belief or trust operator. Therefore, they can be handled by the usual probabilistic temporal model checking techniques, with some slight adaptations for the formulas $\mathbb{G}_A\psi$, $\mathbb{I}_A\psi$ and $\mathbb{C}_A\psi$. For every such subformula ψ, we introduce a new atomic proposition p_ψ, which intuitively represents the probability of ψ being satisfied. Let $AP' = AP \bigcup AP_1$ such that AP_1 is the set of new atomic propositions. We upgrade the labelling function L into $L' : S \times AP' \to [0, 1]$, which allows us to extract the probability value "stored" in p_ψ.

- If $p \in AP$ then $L'(s, p) = 1$, if $p \in L(s)$, and $L'(s, p) = 0$, otherwise.
- If $p_\psi \in AP_1$ then $L'(s, p_\psi) = Prob(\mathcal{M}, s, \psi)$, where $Prob(\mathcal{M}, s, \psi)$ is the probability of satisfying formula ψ among all the temporal paths starting from s. We note that, because of Assumption 2, there exists a unique value for $Prob(\mathcal{M}, s, \psi)$.

We assume that L' is computed from L as part of the algorithm and therefore no change to our previous definitions (such as Definition 4.1) is necessary. Note that, after this pre-processing, all belief or trust formulas are of the form $\mathbb{B}_A^{\bowtie q}p$ or $\mathbb{DT}_{A,B}^{\bowtie q}p$ such that $p \in AP$ is an atomic proposition. This pre-processing can also be applied to subformulas containing no belief or trust formulas. Therefore, in the following, we assume that in ϕ, all subformulas contain at least one belief or trust formulas.

We consider the negated specification formula $\neg\phi$, and verify the existence of a counterexample. We assume that the formula $\neg\phi$ is in negation normal form, i.e., negation operators only appear in front of atomic propositions.

Construction of Expanded System. Let ϕ be the specification formula and $b_\phi = \{\psi_1, ..., \psi_k\}$ be the set of belief or trust subformulas of the form $\mathbb{B}_A^{\bowtie q}p$ or $\mathbb{DT}_{A,B}^{\bowtie q}p$ such that $p_1, ..., p_k \in AP$ are their atomic propositions. We need some notations. Given a set $Q \subseteq S$ of states, a probability space on Q is a triple $\alpha_Q = (S, \mathcal{P}(\mathcal{P}(S)), \Pr_Q)$ such that $\Pr_Q(\{s\}) > 0$ if and only if $s \in Q$. Given a probability space α_Q, we can evaluate belief and trust formulas as follows.

- $\Pr_Q(\mathbb{B}_A^{\bowtie q}p) = \sum_{s \in Q} \Pr_Q(s) \times L'(s, p)$.
- $\Pr_Q(\mathbb{DT}_{A,B}^{\bowtie q}p) = \sum_{s \in Q} \Pr_Q(s) \times \bigotimes_{x \in Int_B(s)} \exists s' : L'(s', p) \times (s \longrightarrow_C^{B.i.x} s')$, where $\otimes \equiv$ inf if $\bowtie \in \{\geq, >\}$, $\otimes \equiv$ sup if $\bowtie \in \{\leq, <\}$.
- the case of $\mathbb{CT}_{A,B}^{\bowtie q}p$ is analogous to $\mathbb{DT}_{A,B}^{\bowtie q}p$.

as the probability of those states satisfying formula ψ. In the following, we use the probability measure $\Pr_Q$ to represent the probability space α_Q. Given a probability distribution $\Pr_P$ and a set of states Q, we define (for a set of states P):

$$\Pr_{P, Act, Q}(t) = \frac{\sum_{s \in P}(\Pr_P(s) \times T(s, a, t))}{\sum_{t \in Q} \sum_{s \in P}(\Pr_P(s) \times T(s, a, t))},$$

as the probability measure over Q, by a temporal transition from P in one step. Moreover, on the pro-attitude dimensions, we define:

$$\Pr_{P, Goal_B(s), Q}(t) = \frac{\sum_{s \in P} \sum_{x \in Goal_B(s)(s)} (\Pr_P(s) \times gp_{A,B}(s, x) \times (s \longrightarrow_C^{B.g.x} t))}{\sum_{t \in Q} \sum_{s \in P} \sum_{x \in Goal_B(s)(s)} (\Pr_P(s) \times gp_{A,B}(s, x) \times (s \longrightarrow_C^{B.g.x} t))}$$

and

$$\Pr_{P, Int_B, Q}(t) = \frac{\sum_{s \in P} \sum_{x \in Int_B(s)} (\Pr_P(s) \times ip_{A,B}(s, x) \times (s \longrightarrow_C^{B.i.x} t))}{\sum_{t \in Q} \sum_{s \in P} \sum_{x \in Int_B(s)} (\Pr_P(s) \times ip_{A,B}(s, x) \times (s \longrightarrow_C^{B.i.x} t))}$$

for $B \in Ags$. Intuitively, they are the probability measures over Q by a goal or intentional transition of agent B from the states in P. Moreover, the transition is parameterised over the agent A's preference functions. Now we define the initial belief for agent A. Given a state s such that $\mu_0(s) > 0$, we define $obs_A(\mu_0, s) = \{t \mid obs_A(t) = obs_A(s), \mu_0(t) > 0\}$ to be the set of possible initial states whose observation is the same as s. Then, we have

$$\Pr_{obs_A(\mu_0, s)}(t) = \frac{\mu_0(t)}{\sum_{t \in obs_A(\mu_0, s)} \mu_0(t)}$$

as the initial belief.

Based on the above notations, we can construct an expanded system which maintains a probability value for each subformula in b_ϕ. The system is $\mathcal{M}_i(\phi) = (Ags, S^\dagger, S_{init}^\dagger, \{Act_A\}_{A \in Ags}, T^\dagger, L^\dagger)$ such that

- $S^\dagger = S \times \mathcal{D}(S) \times R \times \ldots \times R$, where R is the domain of real numbers. Note that, although R is infinite, we need only a finite number of real numbers. We assume that all numbers are of fixed precision and can be encoded with complexity $O(1)$.
- $S_{init}^\dagger = \{(s, \Pr_{obs_A(\mu_0, s)}, \Pr_{obs_A(\mu_0, s)}(p_1), \ldots, \Pr_{obs_A(\mu_0, s)}(p_k)) \mid \mu_0(s) > 0, AP_1 = \{p_1, \ldots, p_k\}\}$,
- the transition relation is defined as follows.
  - $T^\dagger((s, \Pr_Q, v_1, \ldots, v_k), (t, \Pr_{Q, Act, Q'}, \Pr_{Q, Act, Q'}(\psi_1), \ldots, \Pr_{Q, Act, Q'}(\psi_k))) = T(s, a, t)$ if $tp(s, t) = Act$ and $Q' = \{t' \mid \Pr_Q(s') > 0, s' \longrightarrow_T^{a'} t', obs_A(t') = obs_A(t)\}$.
  - $T^\dagger((s, \Pr_Q, v_1, \ldots, v_k), (t, \Pr_{Q, \omega_B^g(s), Q'}, \Pr_{Q, \omega_B^g(s), Q'}(\psi_1), \ldots, \Pr_{Q, \omega_B^g(s), Q'}(\psi_k))) = gp_{A,B}(s, t)$ if and only if $tp(s, t) = B.g$ and $Q' = \{t' \mid s' \in Q, y \in \omega_B^g(s)(s), s' \longrightarrow_C^{B.g.y} t', obs_A(t') = obs_A(t)\}$.
  - $T^\dagger((s, \Pr_Q, v_1, \ldots, v_k), (t, \Pr_{Q, \omega_B^i, Q'}, \Pr_{Q, \omega_B^i, Q'}(\psi_1), \ldots, \Pr_{Q, \omega_B^i, Q'}(\psi_k))) = ip_{A,B}(s, t)$ if and only if $tp(s, t) = B.i$ and $Q' = \{t' \mid s' \in Q, y \in \omega_B^i(s), s' \longrightarrow_C^{B.i.y} t', obs_A(t') = obs_A(t)\}$.
- $L^\dagger((s, \Pr_Q, v_1, \ldots, v_k)) = L(s)$.

Constraints on the Expanded System. The constraints include,

- there do not exist two states $(s, \Pr_Q, v_1, \ldots, v_k)$ and $(s, \Pr_{Q'}, v'_1, \ldots, v'_k)$ in the expanded system $\mathcal{M}^\dagger$ such that they are reachable from one to the other, $supp(\Pr_Q) = supp(\Pr_{Q'})$, and $v_j \neq v'_j$ for some $1 \leq j \leq k$, and
- for any two reachable states $(s, \Pr_Q, v_1, \ldots, v_k)$ and $(s, \Pr_{Q'}, v'_1, \ldots, v'_k)$ such that $supp(\Pr_Q) = supp(\Pr_{Q'})$, only one of them needs to be explored with $T^\dagger$.

With the above constraints, we can see that the size of the expanded system is $O(2^{|\mathcal{M}|})$. However, we do not need to explicitly construct it. The algorithm explores it on-the-fly.

Model Checking Algorithm. For the algorithm, we use function $sat(s, \Pr_Q, v_1, \ldots, v_k, \phi)$ to denote the satisfiability of the formula ϕ on the state $(s, \Pr_Q, v_1, \ldots, v_k)$. For the base cases of non-temporal operators, we have the following inductive rules:

Algorithm 1. Given a state $(s, \Pr_Q, v_1, ..., v_k) \in S^\dagger$ and a formula ϕ, we define the satisfiability relation $sat(s, \Pr_Q, v_1, ..., v_k, \phi)$ as follows:

- $sat(s, \Pr_Q, v_1, ..., v_k, p)$ for $p \in AP$ if $p \in L(s)$;
- $sat(s, \Pr_Q, v_1, ..., v_k, \neg\phi)$ if not $sat(s, \Pr_Q, v_1, ..., v_k, \phi)$;
- $sat(s, \Pr_Q, v_1, ..., v_k, \phi_1 \vee \phi_2)$ if $sat(s, \Pr_Q, v_1, ..., v_k, \phi_1)$ or $sat(s, \Pr_Q, v_1, ..., v_k, \phi_2)$;
- $sat(s, \Pr_Q, v_1, ..., v_k, \mathbb{G}_j\phi)$ if $sat(t, \Pr_{Q'}, v_1', ..., v_k', \phi)$ for all states $(t, \Pr_{Q'}, v_1', ..., v_k')$ such that $T^\dagger((s, \Pr_Q, v_1, ..., v_k), (t, \Pr_{Q'}, v_1', ..., v_k')) > 0$ and $tp(s, t) = B.g$.
- $sat(s, \Pr_Q, v_1, ..., v_k, \mathbb{I}_j\phi)$ if $sat(t, \Pr_{Q'}, v_1', ..., v_k', \phi)$ for all states $(t, \Pr_{Q'}, v_1', ..., v_k')$ such that $T^\dagger((s, \Pr_Q, v_1, ..., v_k), (t, \Pr_{Q'}, v_1', ..., v_k')) > 0$ and $tp(s, t) = B.i$.
- $sat(s, \Pr_Q, v_1, ..., v_k, \mathbb{C}_j\phi)$ if $sat(t, \Pr_{Q'}, v_1', ..., v_k', \phi)$ for all states $(t, \Pr_{Q'}, 0, ..., 0)$ such that $s \xrightarrow{B.i.x}_C t$ for some $x \in \omega_B^i(s)$, and $\Pr_{Q'}$ is any distribution such that $Q' = \{t' \mid s' \in Q, s' \xrightarrow{B.i.x}_C t', x \in \omega_B^i(s), obs_A(t') = obs_A(t)\}$. Note that, as we stated earlier, no belief formula will appear in the scope of a legal intention operator. So the probability values are not useful.
- For the computation of $sat(s, \Pr_Q, v_1, ..., v_k, \mathsf{P}^{\bowtie q} \bigcirc \phi)$ and $sat(s, \Pr_Q, v_1, ..., v_k, \mathsf{P}^{\bowtie q} \phi\mathcal{U}\phi)$, we note that, because the constraint that no belief formula can be in the scope of a probability operator, the components $\Pr_Q$ and $v_1, ..., v_k$ in the state are irrelevant. Therefore, we can compute $sat(s, \mathsf{P}^{\bowtie q} \bigcirc \phi)$ and $sat(s, \mathsf{P}^{\bowtie q} \phi\mathcal{U}\phi)$ as the usual PCTL model checking.
- $sat(s, \Pr_Q, v_1, ..., v_k, \mathbb{B}_A^{\bowtie q}p)$ if $v_j \bowtie q$, for $\mathbb{B}_A^{\bowtie q}p$ being the jth formula in the set b_ϕ.
- $sat(s, \Pr_Q, v_1, ..., v_k, \mathbb{DT}_{A,B}^{\bowtie q}p)$ if $v_u \bowtie q$, for $\mathbb{DT}_{A,B}^{\bowtie q}p$ being the uth formula in the set b_ϕ.

Note that the above cases have the same results for those distributions $\Pr_Q$ whose supports are the same. Now we discuss the case of $\exists\psi$. With the above Algorithm 1, we note that other formulas can be interpreted directly on the states. Therefore, we can introduce atomic propositions p_ψ for those ψ, and update the labelling function into $p_\psi \in L^\dagger((s, \Pr_Q, v_1, ..., v_k))$ if and only if $sat(s, \Pr_Q, v_1, ..., v_k, \psi)$. With this processing, the formula ψ is turned into an LTL formula ψ'. Then as the usual LTL model checking, we turn the formula ψ into a Buchi automaton $\mathcal{A}_\psi$. Let $\mathcal{A}_\psi = (Q, \mathcal{P}(AP), \delta, B, F)$ be the automaton such that Q is a set of states, $\delta : Q \times \mathcal{P}(AP) \to \mathcal{P}(Q)$ is a transition relation, $B \subseteq Q$ is a set of initial states, and $F \subseteq Q$ is a set of sets of acceptance states. Then we construct their product system $\mathcal{M}^\dagger \times \mathcal{A}_{\psi'} = (S^\ddagger, S_{init}^\ddagger, T^\ddagger, L^\ddagger)$ as follows.

- $S^\ddagger = S \times \mathcal{D}(S) \times R \times ... \times R \times Q$
- $(s, \Pr_Q, v_1, ..., v_k, q) \in S_{init}^\ddagger$ if $(s, \Pr_Q, v_1, ..., v_k) \in S_{init}^\dagger$ and $(q_0, L^\dagger((s, \Pr_Q, v_1, ..., v_k)), q) \in \delta$ such that $q_0 \in B$.
- $((s, \Pr_Q, v_1, ..., v_k, q), (t, \Pr_{Q'}, v_1', ..., v_k', q')) \in T^\ddagger$ if we have that $T^\dagger((s, \Pr_Q, v_1, ..., v_k), (t, \Pr_{Q'}, v_1', ..., v_k')) > 0$, $tp(s, t) = Act$, and $(q, L^\dagger((t, \Pr_{Q'}, v_1', ..., v_k')), q') \in \delta$.
- $L^\ddagger(s, \Pr_Q, v_1, ..., v_k, q) = L^\dagger((s, \Pr_Q, v_1, ..., v_k))$.

Algorithm 2. For the case where the formula is $\exists\psi$, we have that

- $sat(s, \Pr_Q, v_1, ..., v_k, \exists\psi)$ if $\mathcal{M}^\dagger[\{s, \Pr_Q, v_1, ..., v_k\}/S_{init}^\dagger] \times \mathcal{A}_\psi$ is empty;

where $\mathcal{M}^\dagger[\{s, \Pr_Q, v_1, ..., v_k\}/S_{init}^\dagger]$ is the system of $\mathcal{M}^\dagger$ by substituting the initial distribution $S_{init}^\dagger$ into $S_{init}^\dagger(\{s, \Pr_Q, v_1, ..., v_k\}) = 1$.

Then for a formula with nested branching time operators, we can use the approach of CTL* model checking. Finally, we have the following theorem to state the correctness of the above algorithm.

THEOREM C.1. $\mathcal{M} \models \phi$ iff, in $\mathcal{M}^\dagger$, $sat(s, \Pr_Q, v_1, ..., v_k, \phi)$ holds for all $(s, \Pr_Q, v_1, ..., v_k) \in S_{init}^\dagger$.

Analysis of Complexity. For the complexity, we need to remember the tuple $(s, \text{Pr}_Q, v_1, ..., v_k)$ for Algorithm 1 or the tuple $(s, \text{Pr}_Q, v_1, ..., v_k, q)$ for Algorithm 2. For Algorithm 1, we may need a polynomial number of alternations to handle the negation operators or the pro-attitudes. Therefore, by Theorem 4.2 of [9], it is in NPSPACE=PSPACE, the same complexity as that of LTL model checking. On the other hand, unlike the LTL model checking, the program complexity, i.e., the complexity measured with respect to the size of the system by assuming that the formula is fixed, is also in PSPACE.

C.2 Hardness Problem

The PSPACE-hardness of the problem can be obtained by a reduction from the problem of deciding if, for a given nondeterministic finite state automaton A over an alphabet Σ, the language $\mathcal{L}(A)$ is equivalent to the universal language Σ^*. Let $A = (Q, q_0, \delta, F)$ be an NFA such that Q is a set of states, $q_0 \in Q$ is an initial state, $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ is a transition function, and $F \subseteq Q$ is a set of final states.

We construct a system $\mathcal{M}(A)$ whose states, except for the initial state s_0, can be classified into two subsystems $\mathcal{M}_1(A)$ and $\mathcal{M}_2(A)$. Intuitively, the subsystem $\mathcal{M}_2(A)$ simulates the execution of A and upon reading each input symbol, probabilistically checks whether the current state of the automaton is a final state, while the subsystem $\mathcal{M}_1(A)$ simulates the universal language Σ^*, at the same time synchronising with $\mathcal{M}_2(A)$ on words which produce a valid run of A.

In particular, each state of $\mathcal{M}_2(A)$ is a pair of a symbol of the alphabet and an annotated automaton state, where the latter can be of the form q, q^u, q^n, q^t, or q^{tf} for $q \in Q$. The state (l, q) is a regular state corresponding to those of automaton A, representing that the state q is reached from another state by reading a symbol l. However, from a state (l_1, q_1) to another state (l_x, q_x) such that $q_x \in \delta(q_1, l_x)$, we need to go through two levels of intermediate states. In the first level, the state (l_x, q_x^u) represents a decision point from which the execution may or may not conduct a test on the current state. The purpose of the test is to see if the state q_x is a final state. With a chance of 0.5, the path will proceed to (l_x, q_x^n) which denotes that no test will be conducted. With another chance of 0.5, the path will conduct the test. If q_x is not a final state (e.g., the state q_2 of Figure 6), then the execution proceeds to the state (l_x, q_x^t), at which the atomic proposition test is labelled. If q_x is a final state (e.g., the state q_3 of Figure 6), then the execution proceeds with 0.25 chance to the state (l_x, q_x^t) and another 0.25 chance to the state (l_x, q_x^{tf}), at which both the propositions test and finished are labelled. From both the states (l_x, q_x^t) and (l_x, q_x^{tf}), the execution can only move to a self-loop.

The subsystem $\mathcal{M}_1(A)$ consists of states which are pairs of a symbol of the alphabet and one of the elements in $\{1, u, n, t\}$, and its structure mirrors that of $\mathcal{M}_2(A)$, with the exception of final states (since $\mathcal{M}_1(A)$ does not know anything about states of automaton A). The purpose of $\mathcal{M}_1(A)$ is to make sure that words which do not produce a valid run of A have a corresponding path in $\mathcal{M}(A)$. The intuition of the construction can be seen from the diagram in Figure 6 The agent A is given distinct observations t on those states (l, q^t) and (l, q^{tf}), that is, agent A can distinguish whether a test occurs at a moment. Formally, let $AP = \{\text{test}, \text{finished}\}$ be a set of atomic propositions, the system is $\mathcal{M}(A) = (Ags, S, S_{\text{init}}, \{Act_A\}_{A \in Ags}, T, L, \{O_A\}_{A \in Ags}, \{obs_A\}_{A \in Ags}, \{\Omega_A\}_{A \in Ags}, \{\pi_A\}_{A \in Ags}, \{p_A\}_{A \in Ags})$ such that

- $Ags = \{B\}$, i.e., there is only a single agent B,
- $S = \{s_0\} \cup (\Sigma \times (Q \cup Q^t \cup Q^{tf} \cup Q^u \cup Q^n \cup \{1, u, n, t\}))$, where $Q^t = \{q^t \mid q \in Q\}$, $Q^{tf} = \{q^{tf} \mid q \in Q\}$, $Q^u = \{q^u \mid q \in Q\}$ and $Q^n = \{q^n \mid q \in Q\}$,
- $S_{\text{init}} = \{s_0\}$ with $\mu_0(s_0) = 1$,
- $Act_B = \{a_\epsilon\}$, i.e., agent B only has a trivial action a_ϵ
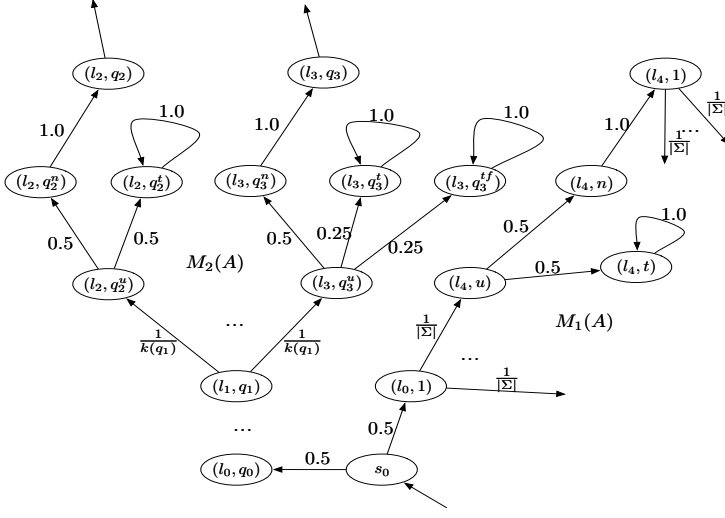
Fig. 6. Diagram for the reduction from universality problem of NFA

- the transition function is defined as follows.
  - $T(s_0, a_\epsilon, (l_0, q_0)) = T(s_0, a_\epsilon, (l_0, 1)) = 0.5$ for some $l_0 \in \Sigma$,
  - $T((l, q), a_\epsilon, (l_2, q_2^u)) = 1/k(q)$ if $q_2 \in \delta(q, l_2)$, where $k(q) = |\{(l_2, q_2) \mid q_2 \in \delta(q, l_2)\}|$,
  - $T((l, q^u), a_\epsilon, (l, q^n)) = 0.5$,
  - $T((l, q^u), a_\epsilon, (l, q^t)) = 0.5$, if $q \notin F$,
  - $T((l, q^u), a_\epsilon, (l, q^t)) = T((l, q^u), a_\epsilon, (l, q^{tf})) = 0.25$, if $q \in F$,
  - $T((l, q^n), a_\epsilon, (l, q)) = T((l, q^t), a_\epsilon, (l, q^t)) = T((l, q^{tf}), a_\epsilon, (l, q^{tf})) = 1$,
  - $T((l, 1), a_\epsilon, (l_2, u)) = 1/|\Sigma|$,
  - $T((l, u), a_\epsilon, (l, n)) = T((l, u), a_\epsilon, (l, t)) = 0.5$, and
  - $T((l, n), a_\epsilon, (l, 1)) = T((l, t), a_\epsilon, (l, t)) = 1$.
- test $\in L((l, q^t))$, test $\in L((l, q^{tf}))$, finished $\in L((l, q^{tf}))$, test $\in L((l, t))$.
- The observation of the agent B is defined as follows.
  - $obs_B((l, q)) = obs_B((l, q^u)) = obs_B((l, q^n)) = obs_B((l, 1)) = obs_B((l, u)) = obs_B((l, n)) = l$, and
  - $obs_B((l, q^t)) = obs_B((l, q^{tf})) = obs_B((l, t)) = l \wedge t$.
- other components are omitted as they are not used in this reduction.

In such a system, we have that $\mathcal{L}(A) = \Sigma^*$ if and only if

$$\mathcal{M}(A) \models \text{test}\mathcal{R}(\neg\text{test} \vee \mathbb{B}_A^{<1}(\neg\text{finished}))$$

where $\mathcal{R}$ is the release operator of LTL. To see this, we show that the language $\mathcal{L}(A)$ is not universal if and only if there exists an initialised infinite path $\rho \in \text{IPath}_T^{\mathcal{M}(A)(s_0)}$ such that $\mathcal{M}(A), \rho \models (\neg\text{test})\mathcal{U}(\text{test} \wedge \mathbb{B}_B^{\le 0}(\text{finished}))$.

($\Longrightarrow$) If the language is not universal, then there exists a word $w = l_1...l_k$ such that no execution of A on w finishes in a final state. From $l_1...l_k$, we can construct a finite path $\rho_1 = s_0(l_0, 1)(l_1, u)(l_1, n)(l_1, 1)...(l_{k-1}, u)(l_{k-1}, n)(l_{k-1}, 1)(l_k, u)(l_k, t)$ in $\mathcal{M}(A)$, but not able to find a path $\rho_2 = (l, q_0)(l_1, q_1^u)(l_1, q_1^n)(l_1, q_1)...(l_{k-1}, q_{k-1}^u)(l_{k-1}, q_{k-1}^n)(l_{k-1}, q_{k-1})(l_k, q_k^u)(l_k, q_k^{tf})$. To see this, we note that each block $(l_x, q_x^u)(l_x, q_x^n)(l_x, q_x)$ for $1 \le x \le k - 1$ represents a piece of execution that

moves from a state (l_{x-1}, q_{x-1}) to another state (l_x, q_x), such that $q_x \in \delta(q_{x-1}, l_x)$, without conducting a test. At the state (l_k, q_k^u), a decision is made to conduct a test. However, because the word w is not a word of A, the test will not move into the state (l_k, q_k^{tf}). Otherwise, it contradicts with the construction of $\mathcal{M}(A)$ that only a final state can move into a state of the form (l, q^{tf}).

Because there does not exist such a path ρ_2, we have that $\mathcal{M}(A), \rho_1 \models \mathbb{B}_B^{\leq 0}$finished. Moreover, for all the states s before (l_k, q_k^t) on the paths ρ_1, we have that $\mathcal{M}(A), s \models \neg$test, and for the state (l_k, q_k^t), we have that $\mathcal{M}, (l_k, q_k^t) \models$ test. Further, because agent B can observe the test action, the check of its beliefs will only concern those paths ρ_1' on which the test action is only taken in the last state. Therefore, the probability, under the condition of agent B's observations, will not leak into those states of the form (l, q^f), and we have $\mathcal{M}(A), \rho_1 \models \neg$test$\mathcal{U}($test $\land \mathbb{B}_B^{\leq 0}$finished$)$. Because the until operator assumes the finiteness of the path, the latter means that $\mathcal{M}(A), \rho_1\delta \models \neg$test$\mathcal{U}($test $\land \mathbb{B}_B^{\leq 0}$finished$)$ for any infinite paths δ that is consistent with the path ρ_1.

($\Longleftarrow$) If $\mathcal{M}(A), \rho \models (\neg$test$)\mathcal{U}($test $\land \mathbb{B}_B^{\leq 0}($finished$))$ for some infinite path ρ, then there must exist a finite path ρ_1 such that $\mathcal{M}(A), \rho_1 \models \mathbb{B}_B^{\leq 0}($finished$)$ and there is a first test occurs on the state $last(\rho_1)$. Now we define function $w(\rho)$ as that $w(\rho_1\rho_2) = w(\rho_1)w(\rho_2)$ and $w((l, q)) = l$. Intuitively, it abstracts the alphabets from the paths. Therefore, we have that $w(\rho_1)$ is a word of A that cannot reach a final state.

Finally, we also need to show that the system $\mathcal{M}(A)$ satisfies the constraints. Note that, the expanded system has two kinds of SCCs. The first kind of SCCs include transitions on which no test action is conducted, and the second kind of SCCs include a self loop on those states with a test action. For both the kinds of SCCs, agent B's belief about the atomic proposition finished is kept the same: for the first kind of SCCs, it is always the case that $\mathbb{B}_B^{\leq 0}($finished$)$ while for the second kind of SCCs, it can be $\mathbb{B}_B^{\leq 0}($finished$)$ or $\mathbb{B}_B^{> 0}($finished$)$, depending on whether the current path can be abstracted into a word of A, and the belief values will not change.

Not that, because the formula test$\mathcal{R}(\neg$test $\lor \mathbb{B}_B^{< 1}(\neg$finished$))$ is constant with respect to different instances of automata A, the PSPACE lower bound is also a lower bound of program complexity. Moreover, the formula complexity, which measures the complexity of problem by the size of the formula by assuming that the system is constant, is also PSPACE-complete: the lower bound comes from the LTL model checking and its upper bound can be derived directly from the combined complexity. Putting the above together, the problem is PSPACE-complete for combined complexity, program complexity, and formula complexity.

D A POLYNOMIAL TIME FRAGMENT

A stochastic automaton is a tuple $SA = (Q, A, \alpha, PI)$ where Q is a set of states, A is a set of symbols, $\alpha : Q \times A \times Q \to [0, 1]$ is a transition function such that for every $q \in Q$ we have $\sum_{a \in A} \sum_{q' \in Q} \alpha(q, a, q') = 1$, and $PI : Q \to [0, 1]$ is an initial distribution over Q such that $\sum_{q \in Q} PI(q) = 1$. Note that, the stochastic automata defined here are different with the probabilistic automata in Section A: a probabilistic automaton requires that for every $q \in Q$ and $a \in A$ we have $\sum_{q' \in Q} \mathcal{M}_a(q, q') = 1$. Given a sequence ρ of symbols, the acceptance probability of ρ in a stochastic automaton SA is

$$\sum_{q \in Q}(PI(q) \times \sum_{q' \in Q} \alpha(q, \rho, q'))$$

where $\alpha(q, \rho, q')$ is defined as follows: $\alpha(q, \rho a, q') = \sum_{q'' \in Q} \alpha(q, \rho, q'') \times \alpha(q'', a, q')$ and $\alpha(q, \epsilon, q) = 1$ for ϵ an empty sequence of symbols. Two stochastic automata are equivalent if for each string $\rho \in A^*$, they accept ρ with equal probability [49].

Given a stochastic automaton $SA = (Q, A, \alpha, PI)$, a pair $D = (Q_D, \alpha_D)$ is a component of SA if $Q_D \subseteq Q$ and for all $q, q' \in Q_D$ and $a \in A$, we have $\alpha_D(q, a, q') = \alpha(q, a, q')$. We say that D is a

strongly connected component (SCC) if for all $q, q' \in Q_D$, there exists a sequence ρ of symbols such that $\alpha_D(q, \rho, q') > 0$, where the computation of $\alpha_D(q, \rho, q')$ can be done recursively as follows: $\alpha_D(q, \rho a, q') = \sum_{q'' \in Q_D} \alpha_D(q, \rho, q'') \times \alpha_D(q'', a, q')$ and $\alpha_D(q, \epsilon, q) = 1$ for ϵ an empty sequence of symbols. An SCC is closed if for all $q \in Q_D$ we have $\sum_{a \in A} \sum_{q \in Q_D} \alpha_D(q, a, q') = 1$.

Constraints. Assume that we are given a system $\mathcal{M}$ and a formula $\Box(\psi \Rightarrow \mathsf{P}^{\rhd \lhd q} \Diamond \mathbb{B}_A^{\geq 1} \psi)$ or $\Box(\psi \Rightarrow \mathsf{P}^{\rhd \lhd q} \Diamond \mathbb{DT}_{A,B}^{\geq 1} \psi)$ such that

- ψ does not contain any belief or trust formulas,
- ψ is of CTL-style, i.e., there is no direct nesting of temporal operators, and
- the system $\mathcal{M}$ satisfies that $\mathcal{M} \models \Box(\psi \Rightarrow \Box \psi)$, i.e., once formula ψ holds, it will hold since then.

We consider their negated specification formulas $\Diamond(\psi \wedge \mathsf{P}^{\widehat{\rhd \lhd} 1-d} \Box \mathbb{B}_A^{>0} \neg \psi)$ or $\Diamond(\psi \wedge \mathsf{P}^{\widehat{\rhd \lhd} 1-d} \Box \mathbb{DT}_{A,B}^{>0} \neg \psi)$, and the algorithm, to be given below, is to determine the existence of a witness for them. We make a remark here that, the polynomial time complexity of this fragment relies on the Assumption 2 that the transition function T of the system $\mathcal{M}$ is deterministic, i.e., for every state $s \in S$ there is a unique $a \in Act$ such that $\sum_{s' \in S} T(s, a, s') = 1$. If this restriction is relaxed by allowing the possibility of more than one actions $a \in Act$ with $\sum_{s' \in S} T(s, a, s') = 1$, it is expected that the complexity could be higher.

Algorithm. Fist of all, we need to compute $\mathrm{Pr}_{\mathcal{M},s}(\neg \psi)$ or $\inf_{x \in \omega_B^i(s)} \mathrm{Pr}_{\mathcal{M},B.i(s,x)}(\neg \psi)$ for every state s and formula $\neg \psi$, depending on which operator, $\mathbb{B}_A^{>0}$ or $\mathbb{DT}_{A,B}^{>0}$, the formula $\neg \psi$ is in the scope of. Recall that $\mathrm{Pr}_{\mathcal{M},s}(\psi)$ is the probability of satisfying ψ among all the temporal paths starting from state s. Because of the restrictions on ψ, these can be computed in polynomial time, using a standard PCTL model checking procedure on DTMCs. We can define $\mathrm{Pr}_{\mathcal{M},\rho}(\psi)$ for ρ a path as $\mathrm{Pr}_{\mathcal{M},last(\rho)}(\psi)$, and have the following proposition.

PROPOSITION D.1. *For a formula ψ without any belief or trust operator, we have that $\mathrm{Pr}_{\mathcal{M},\rho}(\psi) = \mathrm{Pr}_{\mathcal{M},\rho'}(\psi)$ and $\inf_{x \in \omega_B^i(last(\rho))} \mathrm{Pr}_{\mathcal{M},\rho}(\psi) = \inf_{x \in \omega_B^i(last(\rho'))} \mathrm{Pr}_{\mathcal{M},\rho'}(\psi)$ whenever $last(\rho) = last(\rho')$.*

Because belief formulas and trust formulas can be handled in a similar way, in the following we only work with belief formulas. The technique can be adapted to work with trust formulas. From the system $\mathcal{M}$, we can construct a system $\mathcal{M}^\# = (S \times S, T^\#, S_{\mathrm{init}}^\#, L^\#)$ such that

- $S_{\mathrm{init}}^\# = \{(s_1, s_2) \mid s_1, s_2 \in S_{\mathrm{init}}, obs_A(s_1) = obs_A(s_2)\}$,
- $T^\#((s_1, s_2), (a_1, a_2), (s_1', s_2')) = (T(s_1, a_1, s_1'), T(s_2, a_2, s_2'))$ if $obs_A(s_1') = obs_A(s_2')$, $\mathrm{Pr}_{\mathcal{M},s_2'}(\psi) = 0$, $T(s_1, a_1, s_1') > 0$, and $T(s_2, a_2, s_2') > 0$, and
- $L^\#((s_1, s_2)) = (L(s_1), L(s_2))$.

Intuitively, $\mathcal{M}^\#$ is the production of two copies of the system $\mathcal{M}$ such that the two copies preserve the same observations along the transitions (by letting $obs_A(s_1) = obs_A(s_2)$ in $S_{\mathrm{init}}^\#$ and $obs_A(s_1') = obs_A(s_2')$ in $T^\#$) and the second copy always selects those states which do not satisfy the formula ψ (by letting $\mathrm{Pr}_{\mathcal{M},s_2'}(\psi) = 0$). On such a system $\mathcal{M}^\#$, we define operator L_k for $k \in \{1, 2\}$ such that $L_k(\mathcal{M}^\#) = (S \times S, T_k^\#, S_{\mathrm{init},k}^\#, L_k^\#)$ where

- $s_k \in S_{\mathrm{init},k}^\#$ if $(s_1, s_2) \in S_{\mathrm{init}}^\#$ and $k \in \{1, 2\}$.
- $T_k^\#((s_1, s_2), (a_1, a_2), (s_1', s_2'))$ is the kth component of $T^\#((s_1, s_2), (a_1, a_2), (s_1', s_2'))$, and
- $L_k^\#((s_1, s_2))$ is the kth component of $L^\#((s_1, s_2))$.

Intuitively, in $L_k(\mathcal{M}^\#)$, we only consider those initial and transition probabilities of the kth copy of the system $\mathcal{M}$.

Let $Q_D^\# \subseteq S \times S$ be a set of states. A component of $\mathcal{M}^\#$ is a pair $D^\# = (Q_D^\#, T_D^\#)$. We define $T_{k,D}^\#$ as the kth component of $T_D^\#$ for $k \in \{1, 2\}$.

Definition D.2. We say that the component $D^\#$ is an SCC if either $(Q_D^\#, T_{1,D}^\#)$ or $(Q_D^\#, T_{2,D}^\#)$ is an SCC, and is a double-closed SCC if both $(Q_D^\#, T_{1,D}^\#)$ and $(Q_D^\#, T_{2,D}^\#)$ are closed SCCs.

By the construction of $\mathcal{M}^\#$, we have

PROPOSITION D.3. $(Q_D^\#, T_{1,D}^\#)$ *is an SCC if and only if* $(Q_D^\#, T_{2,D}^\#)$ *is an SCC.*

Moreover, given $(Q_D^\#, T_{k,D}^\#)$, we can compute a stationary distribution for each of $(Q_D^\#, T_{1,D}^\#)$ and $(Q_D^\#, T_{2,D}^\#)$. Such a stationary distribution always exists and uniquely exists for a finite $(Q_D^\#, T_{k,D}^\#)$, which is a DTMC. Let $\mu_{k,D}^\#$ be the stationary distribution of $(Q_D^\#, T_{k,D}^\#)$. Then, from a component $D^\# = (Q_D^\#, T_D^\#)$, we can construct two stochastic automata $SA_1(D^\#) = (Q_D^\#, A, T_{1,D}^\#, \mu_{1,D}^\#)$ and $SA_2(D^\#) = (Q_D^\#, A, T_{2,D}^\#, \mu_{2,D}^\#)$ where $\mu_{1,D}^\#$ and $\mu_{2,D}^\#$ are their initial distributions.

Definition D.4. We say that $D^\#$ is internal equivalent when two automata $SA_1(D^\#)$ and $SA_2(D^\#)$ are equivalent. A component $D^\# = (Q_D^\#, T_D^\#)$ is formula-specific with respect to the formula ψ if $\text{Pr}_{\mathcal{M},s}(\psi) > 0$ for some $(s, t) \in Q_D^\#$ and $t \in Q$.

Recall that, two stochastic automata are equivalent if they have the same acceptance probability for all strings. Note that, in the construction of $\mathcal{M}^\#$, we require that the second copy always selects those states which do not satisfy the formula ψ. Therefore, a formula-specific component suggests that the component contains inconsistency between two copies of the system $\mathcal{M}$ with respect to the formula ψ.

Let $\text{Path}(D^\#)$ be the set of paths in $D^\#$, and $\text{IPath}(D^\#)$ and $\text{FPath}(D^\#)$ be the set of infinite and finite paths in $D^\#$, respectively. We also extend the transition relations $T_{1,D}^\#$ and $T_{2,D}^\#$ to work with paths in $\text{Path}(D^\#)$, similar as that of α and α_D in stochastic automata. We do not define $obs_A^\#$ because the two copies of the system have the same observations.

Let ρ be a finite path such that $last(\rho)$ is in an SCC $D^\#$. Given a path ρ_1 in the SCC $D^\#$, we let $obs_A(\rho_1) = \{\rho \in \text{Path}(D^\#) \mid obs_A(\rho) = obs_A(\rho_1)\}$ be the set of paths that agent A cannot distinguish. Then we write

$$g(\rho, D^\#, k, \epsilon) \equiv T_{1,D}^\#(\{\rho_1 \in \text{FPath}(D^\#) \mid \frac{T_{2,D}^\#(obs_A(\rho\rho_1))}{T_{1,D}^\#(obs_A(\rho\rho_1))} > \epsilon, |\rho_1| = k\})$$

for the probability of infinite continuations of ρ whose probability ratio between the two copies of the system is maintained (i.e., greater than ϵ). Moreover, we let $g(\rho, D^\#) = 1$, if for all number $k \in \mathbb{N}$, there exists $\epsilon > 0$ such that $g(\rho, D^\#, k_1, \epsilon) = 1$, and $g(\rho, D^\#) = 0$, otherwise. We have the following propositions.

PROPOSITION D.5. *If $D^\#$ is not internal equivalent then $g(\rho, D^\#) = 0$ for all finite paths ρ whose last state is on $D^\#$.*

PROOF. Let ρ_1 be the shortest path on which the two copies of the system have different acceptance probabilities. By the construction of $\mathcal{M}^\#$, some transitions are removed due to the restriction of $\text{Pr}_{\mathcal{M},s_2}(\psi) = 0$. Therefore, the probability of the second copy is less than the one of the first copy, and we have $\frac{T_{2,D}^\#(obs_A(\rho\rho_1))}{T_{1,D}^\#(obs_A(\rho\rho_1))} < 1$. Then we can construct a path ρ_2 which contains a large number of pieces of observations that are equivalent to ρ_1. With this path, we have that, for all $\epsilon > 0$, $\frac{T_{2,D}^\#(obs_A(\rho\rho_2))}{T_{1,D}^\#(obs_A(\rho\rho_2))} < \epsilon$. Therefore, we have that $g(\rho, D^\#) = 0$. □

PROPOSITION D.6. $g(\rho, D^\#) = 1$ if and only if $D^\#$ is double-closed and internal equivalent.

PROOF. ($\Leftarrow$) Assume that $D^\#$ is double-closed and internal equivalent. By the definition of internal equivalence, for all finite paths $\rho_1 \in \text{FPath}(D^\#)$, we have that $T_{2,D}^\#(\{\rho_2 \in \text{FPath}(D^\#) \mid obs_A(\rho_2) = obs_A(\rho_1)\}) = T_{1,D}^\#(\{\rho_2 \in \text{FPath}(D^\#) \mid obs_A(\rho_2) = obs_A(\rho_1)\})$, which means that $T_{2,D}^\#(obs_A(\rho\rho_1)) = T_{1,D}^\#(obs_A(\rho\rho_1)) > 0$. Because $D^\#$ is double-closed, we have that $g(\rho, D^\#) = 1$.

($\Rightarrow$) Assume that $g(\rho, D^\#) = 1$. We show that both conditions are indispensable. If $D^\#$ is not double-closed and it is the first copy that is not closed, then the probability of $T_{1,D}^\#$ will flow out of the SCC. Therefore, $g(\rho, D^\#, k, \epsilon) = 0$ for a sufficiently large number k. If $D^\#$ is not double-closed and it is the second copy that is not closed, then $\dfrac{T_{2,D}^\#(obs_A(\rho\rho_1))}{T_{1,D}^\#(obs_A(\rho\rho_1))} = 0$ for a sufficiently large $|\rho_1|$, which means that $g(\rho, D^\#) = 0$. Moreover, the case in which $D^\#$ is double-closed but not internal equivalent is guaranteed by Proposition D.5.   □

PROPOSITION D.7. Let ρ be a finite path such that $last(\rho)$ is on an SCC $D^\#$, and $D^\#$ is double-closed, internal equivalent, and formula-specific with respect to ψ. Then $g(\rho, D^\#) = 1$ if and only if for all $\rho_1 \in \text{Path}(D^\#)$, $(\sum_{obs_A(\rho')=obs_A(\rho\rho_1)} be_A(\rho') \times \text{Pr}_{\mathcal{M}, \rho'}(\neg\psi)) > 0$.

PROOF. This is equivalent to proving that

$$\sum_{obs_A(\rho')=obs_A(\rho\rho_1)} be_A(\rho') \times \text{Pr}_{\mathcal{M}, \rho'}(\neg\psi) > 0 \text{ iff } \frac{T_{2,D}^\#(obs_A(\rho\rho_1))}{T_{1,D}^\#(obs_A(\rho\rho_1))} > 0,$$

for $\rho_1 \in \text{Path}(D^\#)$.

Now, by the construction of $\mathcal{M}^\#$ (in which the second copy satisfies $\text{Pr}_{\mathcal{M}, s_2}(\psi) = 0$) and the three conditions of $D^\#$, we have that

$$\sum_{obs_A(\rho')=obs_A(\rho\rho_1)} be_A(\rho') \times (\text{Pr}_{\mathcal{M}, \rho'}(\neg\psi) > 0) = \frac{T_{2,D}^\#(obs_A(\rho\rho_1))}{T_{1,D}^\#(obs_A(\rho\rho_1))}.$$

The equivalence of

$$\sum_{obs_A(\rho')=obs_A(\rho\rho_1)} be_A(\rho') \times (\text{Pr}_{\mathcal{M}, \rho'}(\neg\psi) > 0) > 0$$

and

$$\sum_{obs_A(\rho')=obs_A(\rho\rho_1)} be_A(\rho') \times \text{Pr}_{\mathcal{M}, \rho'}(\neg\psi) > 0$$

can be seen by their structures.   □

With these propositions, we have the following algorithm.

Algorithm 3. After the computation of $\text{Pr}_{\mathcal{M}, s}(\neg\psi)$ or $\inf_{x \in \omega_B^i(s)} \text{Pr}_{\mathcal{M}, B.i(s,x)}(\neg\psi)$ for every state s and formula $\neg\psi$, the algorithm proceeds by the following sequential steps:

(1) compute the set of SCCs of $\mathcal{M}^\#$ such that $D^\#$ is formula-specific with respect to ψ, double-closed, and internal equivalent,
(2) for every state s satisfying ψ, we do the following:
   (a) compute the reachability probability to those SCCs from the first step; let the probability value be p, and
   (b) check whether $p \widehat{\bowtie} 1 - q$.
       If there exists a state s that can satisfy the above computation, then the specification formula does not hold, otherwise holds.

Analysis of Complexity. The complexity of the algorithm is in polynomial time. First of all, the computation of $\Pr_{\mathcal{M},s}(\neg\psi)$ or $inf_{x\in\omega_B^i(s)}\Pr_{\mathcal{M},B.i(s,x)}(\neg\psi)$ can be done in polynomial time because ψ is of CTL-style and contains no belief or trust formula. Second, for the first step of the algorithm, the computation of all SCCs of $\mathcal{M}^\#$, whose size is quadratic with respect to $\mathcal{M}$, can be done in polynomial time by Tarjan's algorithm, and the checking of the three conditions (i.e., formula-specific with respect to ψ, double-closed, and internal equivalent) can be done in polynomial time. In particular, the checking of internal-equivalent of an SCC can be done in polynomial time is a result of the existence of a polynomial time algorithm for the equivalence of stochastic automata [49]. Third, for the second step of the algorithm, the computation of reachability probability on $\mathcal{M}^\#$ can be done in polynomial time, and the comparison of values can be done in constant time.