T. Chen
G. Primiero
F. Raimondi
N. Rungta

A Computationally Grounded, Weighted Doxastic Logic

Abstract. Modelling, reasoning and verifying complex situations involving a system of agents is crucial in all phases of the development of a number of safety-critical systems. In particular, it is of fundamental importance to have tools and techniques to reason about the *doxastic* and *epistemic* states of agents, to make sure that the agents behave as intended. In this paper we introduce a computationally grounded logic called COGWED and we present two types of semantics that support a range of practical situations. We provide model checking algorithms, complexity characterisations and a prototype implementation. We validate our proposal against a case study from the avionic domain: we assess and verify the situational awareness of pilots flying an aircraft with several automated components in off-nominal conditions.

Keywords: Multi-agent systems, doxastic logic, model checking

1. Introduction

Multi-agent systems are increasingly being employed in modelling and reasoning about complex scenarios, from self-driving cars to autonomous rovers. Tasks related to such systems usually encompass design, specification, validation and verification, including certification activities when agents operate in safety-critical situations [27]. Reasoning about beliefs is a fundamental aspect of these activities, as agents typically comply with a protocol that is, essentially, prescribing a course of actions according their beliefs [29]. But reasoning about beliefs plays a crucial role also in all the verification activities that occur from design to run-time execution. Consequently, it is of utmost importance that appropriate tools and techniques are developed to support epistemic and doxastic characterisations of agents.

In this paper we propose a logic to reason about *quantified beliefs*. A standard approach to belief quantification involves the use of *probabilities*; however, a number of other approaches exist. We refer to [15] for a detailed overview. In this paper we make use of the term *degrees of belief* to abstract away from the actual mechanism employed to give a quantitative figure to beliefs. The literature employs the terms *subjective* and *objective*

Presented by **Name of Editor**; *Received* October 21, 2014

to discriminate between assignments that clearly differentiate between probabilities and beliefs in the former case, and assignments that refer to actual features in the real word (that may or may not correspond to probabilities) in the latter case.

In this paper we first take a modelling approach that can be seen as *subjective* as it does not require the definition of probability distributions but only relies on *counting*. However, as it will be clear in the following, there is a link between this approach and objective assignments. We then take an objective approach by introducing the notion of degrees of belief as a bespoke measure of reachability in probabilistic interpreted systems, an extension of interpreted systems [11] in which the temporal relation is represented as a discrete-time Markov Chain (DTMC).

More in details, our contributions are as follows:

- We provide a language called COGWED (COmputationally Grounded WEighted Doxastic logic) that extends CTLK [11] with weighted doxastic operators. These operators allow to reason about the doxastic states of one or more agents.
- We provide two types of semantics for this language. The first is based on standard interpreted systems and evaluates degrees of beliefs as ratios defined on equivalence classes of the epistemic accessibility relations. The second employs a generalised temporal relation and computes degrees of beliefs as a ratio between the probabilities of reaching epistemic equivalence classes, making use of a discounting factor for systems without perfect recall (see Section 4).
- We introduce model checking algorithms for both semantics and we characterise their complexity. We provide a prototype implementation and we assess the scalability of our approach on standard benchmark tests.
- We validate our approach against a concrete case study: we assess the situational awareness of a pilot flying in off-nominal conditions using a model provided by researchers at NASA Ames [1].

The rest of the paper is organised as follows: in Section 2 we discuss related work, in Section 3 we present the syntax of COGWED, in Section 4 we introduce various options for its semantics. We introduce model checking algorithms and a prototype implementation in Section 5. We perform an experimental evaluation and present a case study in Sections 6 and 7, and we conclude in Section 8.

2. Related Work

Formalisms to model degrees of belief have been investigated in the past by a number of authors. Dempster-Shafer belief functions [25] are among the most common approaches to assign a *mass* to beliefs and to combine belief functions. This formalism is a classical example of *subjective* assignment in which *plausibility* can be modelled differently from *probability*. We refer to [15] for other approaches to modelling degrees of belief subjectively. In all these formalisms, however, the function associating a weight to a belief needs to be externally provided, for instance by employing historical data or other means; this is a key difference with our approach, where degrees are computed as the ratio between two sets of possible worlds.

The idea of evaluating degrees of belief as the ratio between possible worlds is not new: in the formalism of *random worlds* [3] degrees of belief are computed using *proportion expressions* of the form $||\phi(x)|\psi(x)||$. These expressions denote the proportion of domain elements satisfying ϕ w.r.t. those satisfying ψ in the domain of a knowledge base. Conditional expressions are used in [3] to evaluate the weight of beliefs in knowledge bases and are shown to satisfy a set of *desiderata* for default reasoning. While our system is "computationally grounded" in the sense that degrees of belief are not provided externally, computing degrees of belief using random worlds is an undecidable problem in the general case. Moreover, there does not seem to be a tractable solution to add temporal reasoning to this formalism as we do here (as exemplified in the case of the dining cryptographers). Additionally, another key difference with our approach is that we provide a *formal language* to express degrees of belief for a *system* of agents and we are not limited to the single agent case. Along similar lines, the work in [13] introduces *plausibility measures* that are used to justify a set of axioms for default reasoning. More recently, the work in [16] addresses decision making in terms of weighted sets of probabilities by introducing an axiomatization and by providing *dynamic* decision making procedures.

Our second semantics treat reachability using Discrete Time Markov Chains. A language that combines first-order logic and probability in finite domains is introduced in [24] using *Markov Logic Networks* (MLN): similarly to [3], knowledge bases are employed as the underlying semantics, and weights are associated to formulae in the KB. In the case of finite domains, weights can be learned using a set of algorithms and the authors show that MLN can tackle real scenarios. The work in [9] presents the logic $P_F KD45$, whose syntax is very similar to COGWED. The semantics of this logic relies on externally-provided probability measures over finite bases; the authors present an axiomatization and a decision procedure for this logic but no model checking algorithm. The key differences with our work are the different semantics based on interpreted systems and the inclusion of multiple agents and temporal modalities, in addition to a dedicated model checking tool.

In the multi-agent system community there have been a number of works addressing the verification of doxastic modalities, such as the Jason tool [4] and the AIL+AJPF framework [10]. These two works address BDI architectures and are capable of verifying "standard" (i.e., non-weighted) doxastic operators. The tool MCK [14] has recently been extended to include probabilistic reasoning. In this tool probabilities are assigned to *temporal relations*; the tool is able to verify only the probability of Boolean expressions, possibly nested in an X (next-state) temporal operator. Probabilities over temporal relations are also analysed using the logic PCTL (Probabilitstic CTL) in the well known tool PRISM [19], which has recently been extended to verify rPATL (restricted Probabilistic ATL) [6, 7]. A logic to reason about probabilistic knowledge and strategies is also described in [17]: in this work probabilities are associated to temporal relations and to *observations* as well. Our key difference is again in the definition of degrees of belief in terms of *possible worlds*.

The tools PRISM and MCK and the approach in [17] employ *probabilities over temporal or epistemic transitions*. As mentioned in the introduction, we refer instead to *degrees of belief* and we allow for a choice in how degrees should be computed. In the first case, we do not use these probabilities but we rely only on ratios between equivalence classes. The relationship between the approach in which degrees of beliefs are computed as ratios and the approach in which degrees of belief arise from temporal characterisations has been investigated in [3] for a scenario very similar to ours. Similarly to this work, in our first setting all the possible worlds are equally likely and we do not model probabilities of *transitions*. Essentially, our first semantics adopts the *principle of indifference* by Bernoulli and Laplace. As described in [3], a uniform distribution for possible worlds is the one that maximizes entropy. In turn, this corresponds to the least amount of *information* about the probability distribution of epistemically equivalent worlds. In other words, our first semantics start from an *unknown* objective assignment of probabilities to transitions and we build a *subjective* assignment of degrees of belief to agents according to this unknown objective assignment; agents' degrees of belief can then be interpreted using a computationally grounded evaluation.

In our second semantics for COGWED, instead, degrees of belief are computed using reachability properties of equivalence classes. In contrast to [17], we do not require probabilities for epistemic relations to be provided externally. Instead, we compute degrees of beliefs as a reachability measure of equivalence classes. To the best of our knowledge, this is a novel approach that helps in making the proposed solution computationally grounded.

3. COGWED Syntax

In this section we introduce the syntax of COGWED. The language of COGWED includes a branching time language for temporal reasoning (CTL, [8]), epistemic operators to reason about single agent and group epistemic modalities [11], and weighted doxastic operators for one or more agents. More in detail, let Ag be a nonempty set of agents, $\varnothing \neq \Gamma \subseteq Ag$, and $\sim$ be one of the following comparison operators: $\{<, \leqslant, =, \geqslant, >\}$. The syntax of COGWED is as follows:

$$\begin{aligned} \phi \quad ::= \quad & p \mid \neg\phi \mid \phi \wedge \psi \mid EX\phi \mid EG\phi \mid E[\phi U \psi] \mid \\ & K^i\phi \mid E^\Gamma\phi \mid D^\Gamma\phi \mid C^\Gamma\phi \mid \\ & B^\Gamma_{\sim x}\phi \end{aligned}$$

Where:

- p is an atomic proposition from a set AP;
- $EX\phi, EG\phi, E[\phi U \psi]$ are standard CTL temporal operators, read respectively as "there exists a point in the next state such that", "there exists a path such that globally", and "there exists a path such that ϕ is true until ψ becomes eventually true";
- i is an index for agents, ranging from 1 to n;
- K^i is the standard epistemic operator, read as "agent i knows ϕ";
- $E^\Gamma, D^\Gamma, C^\Gamma$ are epistemic group modalities expressing the notion of "everybody knows", "distributed knowledge" and "common knowledge". We refer to [11] for further details about these operators;
- x is a real number, $0 \leqslant x \leqslant 1$; and
- $B^\Gamma_{\sim x}\phi$ is the doxastic operator and is read as "agents in group Γ believe ϕ with degree of belief $\sim x$. With slight abuse of notation we will write i for the singleton $\Gamma = \{i\}$.

An example of a COGWED formula is $B^1_{\leqslant 0.2}(p \vee q)$, which is read as "Agent 1 believes $(p \vee q)$ with a degree of belief less or equal than 0.2, while $B^2_{=0.5}(B^1_{\leqslant 0.1}(p))$ is read as "Agent 2 believes with degree exactly equal to 0.5

that Agent 1 believes with degree at most 0.1 that p". As we will see below, $B^i_{=1}\phi$ is equivalent to $K_i\phi$.

As a practical example, consider a scenario composed of two agents and a deck of N different cards, numbered from 1 to N. Suppose that the first agent draws a card from the deck, without showing it to the second agent, and that the second agent does the same. Let `agent1_has_c1` be an atomic proposition in AP denoting the fact that the first agent has card one. Then, the following is a formula encoding the fact that, if agent 1 has card 1, then agent 1 *knows* that agent 2 *believes* with degree less than $\frac{1}{(N-1)}$ that the first agent has indeed card 1:

$$\texttt{agent1\_has\_c1} \rightarrow (K^1(B^2_{<\frac{1}{(N-1)}}\texttt{agent1\_has\_c1})).$$

We will use this example in Section 6 to assess the scalability of our model checking algorithm.

4. COGWED Semantics

In this section we introduce two types of semantics for COGWED. They are both based on the formalism of Interpreted Systems from [11], which we introduce in the next subsection.

4.1. Interpreted Systems

Given a set of n agents, an Interpreted System is a tuple $IS = (G, R_t, V)$ where

- $G = \underset{1\cdots n}{\times} L_i$ is a finite set of *global* states, obtained as the cartesian product of n sets of *local* states (one set for each agent);
- $R_t \subseteq G \times G$ is a temporal relation (it is assumed that each state has at least a successor);
- $V : AP \rightarrow 2^G$ is an evaluation function for atomic propositions.

Given n agents, we define a set of n *equivalence relations* (one for each agent): let $g = (l_1, \ldots, l_n)$ and $g' = (l'_1, \ldots, l'_n)$ be two global states from G; we define gR_ig' iff $l_i = l'_i$, i.e., two global states g, g' are equivalent for agent i iff the local state of agent i is the same in g and in g' (notice that these are the standard epistemic relations used in [11] to interpret epistemic modalities). The relation R_i is obviously an equivalence relation; we define $\{g\}_{R_i}$ to be the equivalence class of the global state g with respect to R_i.

Given an interpreted system IS and a global state g, logic formulae involving CTL and epistemic operators can be interpreted as follows (we refer to [11] and references therein for additional details):

$IS, g \models p$	iff	$g \in V(p)$;
$IS, g \models \neg\phi$	iff	$IS, g \not\models \phi$;
$IS, g \models \phi \wedge \psi$	iff	$IS, g \models \phi$ and $IS, g \models \psi$;
$IS, g \models EX\phi$	iff	there exists $g' \in G$ s.t. $gR_t g'$ and $IS, g' \models \phi$;
$IS, g \models EG\phi$	iff	there exists a path $\pi = (g, g_1, \dots)$ such that, for all i, $IS, g_i \models \phi$;
$IS, g \models E[\phi U \psi]$	iff	there exists a path $\pi = (g, g_1, \dots)$ and an index j such that $IS, g_j \models \psi$ and $IS, g_i \models \phi$ for all $i < j$;
$IS, g \models K^i\phi$	iff	$gR_i g'$ implies $IS, g' \models \phi$;
$IS, g \models E^\Gamma\phi$	iff	$gR_E^\Gamma g'$ implies $IS, g' \models \phi$, where $R_E^\Gamma = \bigcup_{i\in\Gamma} R_i$;
$IS, g \models D^\Gamma\phi$	iff	$gR_D^\Gamma g'$ implies $IS, g' \models \phi$, where $R_D^\Gamma = \bigcap_{i\in\Gamma} R_i$;
$IS, g \models C^\Gamma\phi$	iff	$gR_C^\Gamma g'$ implies $IS, g' \models \phi$, where R_C^Γ is the transitive closure of R_E^Γ.

With slight abuse of notation we denote with $V(\phi)$ the set of states of an interpreted system IS in which ϕ holds. This logic is usually named CTLK and can include group epistemic modalities to reason about distributed and common knowledge. In the next section we will extend this logic with doxastic operators[1].

4.2. Counting worlds

In this section we present how COGWED formulae can be evaluated in Interpreted Systems by extending the definitions provided in the previous section with the following:

$$IS, g \models B^\Gamma_{\sim x}\phi \quad \text{iff} \quad \frac{|V(\phi) \cap \{g\}_{R_D^\Gamma}|}{|\{g\}_{R_D^\Gamma}|} \sim x$$

[1]The formalism of interpreted systems presented in [11] and employed in other model checkers such as [20, 14] also includes the notions of agents' actions and agents' protocols: to keep our presentation simple, we do not consider these here, as they play no role in the semantics for the logic presented below.

Note that, to evaluate the doxastic operator for a group of agents, we adopt the relation characterising *distributed knowledge*. This corresponds to the situation in which agents share their epistemic accessibility relation, thus reducing the overall number of alternatives. The intuition behind this characterisation is that the degree of belief that a group of agents associates to a formula ϕ in a global state g is the ratio between the number of states of $\{g\}_{R_D^\Gamma}$ (the equivalence class of g with respect to the epistemic group relation R_D^Γ) in which ϕ is true and the total number of states in $\{g\}_{R_D^\Gamma}$. Note that when Γ is a singleton for agent i, R^Γ is the epistemic relation for the individual agent R_i.

This definition of degrees of beliefs is *computationally grounded* in the sense of Wooldridge [28]: modalities are interpreted directly on the set of possible computations of a multi-agent system (equivalently: modalities are interpreted on a Kripke model that corresponds to the possible computations of a multi-agent systems), and there is no need to provide weights as part of the model. We refer to Section 2 for a comparison with other existing approaches to evaluate degrees of belief.

The following formulae for the single agent case are valid in all COGWED models implementing the semantics described above, as a result of simple arithmetic considerations:

1. $B^i_{\leqslant x}\phi \rightarrow B^i_{\leqslant y}\phi$ for all $y \geqslant x$;
2. $B^i_{\geqslant x}\phi \rightarrow B^i_{\geqslant y}\phi$ for all $y \leqslant x$;
3. $B^i_{\geqslant x}\phi \leftrightarrow B^i_{\leqslant (1-x)}\neg\phi$: this means that, if an agent believes ϕ with degree greater than x, then the agent believes the negation of ϕ with degree less than $1 - x$. The converse is also true.

Finally, it is easy to see, as we assume a finite state space, that $B^i_{=1}\phi$ is equivalent to $K_i\phi$, i.e., a degree of belief equal to 1 corresponds to the standard epistemic operator. Dually, as a result of the third formula above, it is also true that $B^i_{=0}\phi \leftrightarrow K^i(\neg\phi)$.

4.3. DTMC-based semantics

To motivate the semantics based on Discrete-Time Markov Chains, consider the following

Example 1. *Consider the scenario depicted in Figure 1. The system has two (global) states g_1 (left) and g_2 (right), in which, respectively, the propositions* working *and* broken *are true (formally: $V(g_1) = \{\text{working}\}$ and $V(g_2) =$*

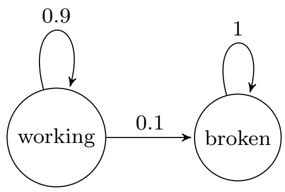

Figure 1. Example of probabilistic interpreted systems

{broken}*). In state g_1 two transitions are enabled: the first one is a loop around g_1 with probability* 0.9 *and the second is a transition to state g_2 with probability* 0.1*. Once in state g_2, the system loops there. For simplicity, we assume that there is only one agent, and the two states g_1 and g_2 are indistinguishable.*

We model the scenario above with an extension of interpreted systems with *probabilities over temporal transitions*, following the standard approach of Markov chains. We call this extension *probabilistic interpreted systems.* Technically, a probabilistic interpreted system is a tuple $PIS = (G, \mathbf{P}, V, \alpha)$ where

- G and V are as before;
- $\mathbf{P} : G \times G \to [0, 1]$ is the (temporal) *probabilistic* transition relation, such that $\sum_{g' \in G} \mathbf{P}(g, g') = 1$ for any $g \in G$, encoded by means of the matrix $\mathbf{P}$.
- $\alpha : G \to [0, 1]^n$ is an initial probability distribution (the "initial state"), such that $\sum_{g \in G} \alpha(g) = 1$.

From the probabilistic transition matrix $\mathbf{P}$ we can derive the temporal relation R_t such that, for any two global states g, g', $R_t(g, g')$ iff $\mathbf{P}(g, g') > 0$. We also introduce the standard epistemic relations R_i for each agent i, as before. As a result, all the temporal and epistemic operators can be interpreted as described in the previous section independently from the initial distribution α and the transition probabilities in $\mathbf{P}$.

One could define the semantics for the doxastic operator in probabilistic interpreted systems simply by computing the probability of staying in a certain set of states from α, i.e., the stationary distribution or the steady-state distribution (if exists). However, we argue that this approach could lead to counter-intuitive results. As an example, consider again the scenario depicted in Figure 1. The steady-state distribution is obtained as the limit of the evolution of α for an infinite number of steps [22] (i.e., $\lim_{n \to \infty} \alpha \mathbf{P}^n$).

In the case of the example, the probability of staying at state g_2 in which broken holds is 1: should a rational agent *know* that the system is in a broken state?

The key issue here is that agents are *memoryless*, and as a result it should *not* be assumed that the system has executed an infinite number of rounds when evaluating their epistemic and doxastic states. In essence, the number of rounds should be treated as an unobservable value. To this end, we consider *transient distributions* defined as:

$$\pi_n = \alpha \cdot \mathbf{P}^n.$$

This distribution π_n is a vector of values; each value represents the probability of reaching a certain global state in n steps from the initial distribution α. The distribution π_n is obtained by applying the transition relation n times to the initial distribution. Using transient distributions we define a memoryless probabilistic distribution for the states in G as:

$$\hat{\pi}_\beta = (1-\beta) \sum_{n=0}^{\infty} \beta^n \pi_n \quad .$$

where $\beta \in [0,1)$ is a *discounting factor* for future events. Intuitively, β encodes the "weight" that agents place on future events and $\hat{\pi}_\beta$ is a vector of probability values for states of G representing an estimation of the likelihood of being in each state when time is *not* known. For a given set of states $X \subseteq G$ we write $\hat{\pi}_\beta(X) = \sum_{g \in X} \hat{\pi}_\beta(g)$ to represent the probability of being in X according to distribution $\hat{\pi}_\beta$. With this notation we can finally define the semantics of $B^\Gamma_{\sim x}\phi$ in probabilistic interpreted systems as follows:

$$IS, g \models B^\Gamma_{\sim x}\phi \quad \text{iff} \quad \frac{\hat{\pi}_\beta\left(V(\phi) \cap \{g\}_{R^\Gamma_D}\right)}{\hat{\pi}_\beta\left(\{g\}_{R^\Gamma_D}\right)} \sim x$$

Similar to the counting worlds semantics, the degree of beliefs is defined as a ratio. In this case, however, we take the ratio between the "reachability" of the set of states in which ϕ is true in the equivalence class $\{g\}_{R^\Gamma_D}$ and the "reachability" of the whole equivalence class.

As a concrete example, consider again the example at the beginning of this section (cf. Figure 1). The transition matrix for this example is

$$\mathbf{P} = \begin{bmatrix} 0.9 & 0.1 \\ 0 & 1 \end{bmatrix}$$

and assume the system starts from state g_1 so that the initial distribution is $\alpha = (1, 0)$. Assuming a discounting factor β we have

$$\begin{aligned}
\hat{\pi}_\beta &= (1,0)(1-\beta)\left(I - \beta \begin{bmatrix} 0.9 & 0.1 \\ 0 & 1 \end{bmatrix}\right)^{-1} \\
&= (1,0)\frac{1-\beta}{(1-0.9\beta)(1-\beta)} \begin{bmatrix} 1-\beta & 0.1\beta \\ 0 & 1-0.9\beta \end{bmatrix} \\
&= (\frac{1-\beta}{1-0.9\beta}, \frac{0.1\beta}{1-0.9\beta})
\end{aligned}$$

(Note that the calculation exploits Propositionr̃efprop:pi in Section 5. As mentioned above, in this example the agent cannot distinguish between g_1 and g_2, and as a result $\hat{\pi}_\beta(\{g_1, g_2\}) = 1$. Proposition broken is true in g_2, and thus $V(\text{broken}) = \{g_2\}$, and consequently the degree of belief in broken is $\frac{0.1\beta}{1-0.9\beta}$. The situation in which $\beta = 0$ represents the case in which the system has not evolved and no weight is given to transient distributions. In this case the degree of belief in "broken" is zero. At the other extreme of the range, $\beta = 1$ (note that in our framework, it must be the case that $\beta < 1$; however, β can be arbitrarily close to 1) encodes the certainty that the system has run an infinite number of times. In this case the degree of belief in "broken" is one. All the other values represent intermediate situations. In the general case, the value of β needs to be chosen according to the specific scenario to be modelled and should take into account the capabilities of the agents involved and the overall structure of the system.

5. Model Checking COGWED

In this section we present model checking algorithms for the two types of semantics of COGWED. The first algorithm extends the standard labelling algorithm for CTL with epistemic operators [8, 20]. The second algorithm computes the memoryless probabilistic distribution for states of a probabilistic interpreted system, from which degrees of belief can be computed.

5.1. Counting worlds

The model checking algorithm for COGWED under the counting semantics extends the standard CTLK algorithm [20] with an additional procedure to compute the set of states in which a formula of the form $B^{\Gamma}_{\sim x}\phi$ holds. This procedure is described using a Java-like algorithm in Figure 2. The

```
1  // We are given a set of equivalence
2  // classes for group Γ:
3  Set <Set<Gstate>>> rGamma;
4
5  // This method computes the set of
6  // states in which B^Γ_{~x}φ is true
7  public Set<Gstate> satB(int i, Formula f,
8                           String op , float x) {
9    Set<Gstate> previous = SAT(f);
10   Set<Gstate> result = new Set();
11   for (Set<Gstate> eqClass: rGamma) {
12     if ( |eqClass∩previous| / |eqClass| ~ x) {
13       result.add(eqClass);
14     }
15   }
16   return result;
17 }
```

Figure 2. Java-style algorithm sketch

procedure employs the set of equivalence classes for group Γ which can be pre-computed by partitioning the set of global states.

The procedure `satB` returns the set of global states satisfying the formula $B^{\Gamma}_{\sim x}\phi$. It starts by (recursively) calling a method `SAT(`ϕ`)` that computes the set of states in which the formula ϕ is true (line 9). Then, it iterates over the equivalence classes of group Γ (line 11). In line 12 the method computes the ratio of the set in which the formula is true in a given equivalence class over the size of the actual equivalence class. If this ratio satisfies the appropriate relation $\sim$, then the method adds the *whole* equivalence class to the set of states in which the formula is true (line 13). The intersection of sets of states can be performed with standard library functions provided by Java; we refer to the source code available online at `https://sites.google.com/site/mccogwed/` for additional details about the actual implementation. The final result is returned at line 16.

As mentioned above, notice that the algorithm does not operate on individual states. Instead, once the equivalence classes are built, the algorithm works with *sets* of states.

5.1.1. Complexity considerations

Model checking CTLK formulae in an interpreted system takes time polynomial in the size of the formula and in the size of the model [11]. All the operations in the algorithm described in Figure 2 require at most polynomial time: computing the set of equivalence classes, iterating over them, and computing intersection of states. Therefore, the method described above remains in the same polynomial complexity class of the standard CTLK model checking algorithm.

In practical applications, however, the actual state space is likely to explode as a result of the number of variables employed to model a given scenario. A number of techniques are available to manage large state spaces. In particular, Ordered Binary Decision Diagrams (OBDDs) are employed in model checkers for multi-agent systems such as MCMAS [20] and MCK [14]. The algorithm `satB` of Figure 2 operates on sets of states and only performs intersections of sets: these operations can be performed on the OBDDs for the sets of states, and therefore this part of the algorithm can be executed symbolically. The computation of equivalence classes needed at line 3, however, may require in the worst case the explicit enumeration of all reachable states, if all global states are epistemically different for a given agent. This is rarely the case and, in fact, the number of equivalence classes is normally orders of magnitude smaller than the number of global states. This is indeed the case in the examples that we present in Sections 6 and 7.

5.2. DTMC-based semantics

In this section, we show how to carry out model checking COGWED under the DTMC semantics. As discussed above, the semantics for temporal and epistemic operators does not change and as a result the standard approach of [8, 20] can be employed. To evaluate the doxastic operator, we need a procedure to compute $\hat{\pi}_\beta$ as defined in Section 4. The following proposition gives an analytical solution to compute $\hat{\pi}_\beta = \sum_{n=0}^{\infty} \beta^n \pi_n$

Proposition 1.

$$\hat{\pi}_\beta = (1-\beta) \cdot \alpha (I - \beta \mathbf{P})^{-1}$$

Proof. First, observe that

$$\begin{aligned}
\sum_{n=0}^{\infty} \beta^n \pi_n &= \pi_0 + \sum_{n=1}^{\infty} \beta^n \pi_n = \pi_0 + \beta \cdot \sum_{n=1}^{\infty} \beta^{n-1} \pi_n \\
&= \pi_0 + \beta \cdot \sum_{n=0}^{\infty} \beta^n \pi_{n+1} = \pi_0 + \beta \cdot \sum_{n=0}^{\infty} \beta^n \alpha \mathbf{P}^{n+1} \\
&= \pi_0 + \beta \cdot \mathbf{P} \cdot \sum_{n=0}^{\infty} \beta^n \alpha \mathbf{P}^n = \pi_0 + \beta \cdot \mathbf{P} \cdot \sum_{n=0}^{\infty} \beta^n \pi_n
\end{aligned}$$

The matrix $I - \beta\mathbf{P}$ is invertible since $\beta \in [0, 1)$. It follows that

$$\sum_{n=0}^{\infty} \beta^n \pi_n = \pi_0 (I - \beta\mathbf{P})^{-1} = \alpha(I - \beta\mathbf{P})^{-1}$$

To conclude, $\hat{\pi}_\beta = (1-\beta)\sum_{n=0}^{\infty} \beta^n \pi_n = (1-\beta) \cdot \alpha(I - \beta\mathbf{P})^{-1}$. ■

Now, for computing the set of (global) states satisfying the formula $B^{\Gamma}_{\sim x}\phi$, we first compute the set of states "previous" in which the formula ϕ holds (as on the line 9 of the algorithm in Figure 2). Then for each equivalence class, we compute the set of states eqClass $:= \{g\}_{R_i}$ (as in line 11 of the algorithm in Figure 2) and simply check whether $\frac{\hat{\pi}_\beta(\text{previous} \cap \text{eqClass})}{\hat{\pi}_\beta(\text{eqClass})} \sim x$.

5.2.1. Complexity considerations for DTMC semantics

As in the case of counting worlds, the complexity of model checking COGWED formulas in probabilistic interpreted systems remains *polynomial* with respect to the size of the model and the formula. To see this, note that by Proposition 1 the distribution $\hat{\pi}_\beta$ can be computed in time polynomial in the size of the model, as it only requires to compute the inverse matrix which can be done in cubic time by, e.g,. Gauss elimination. In practice, one can also use an iteration method [26] which is usually more efficient.

6. Experimental Results

In this section we assess the feasibility of using COGWED by performing an initial performance evaluation on the standard example of the Dining Cryptographers [5]. In the next section we then present a practical application to an avionic scenario.

6.1. Performance evaluation for counting worlds semantics

The protocol of the dining cryptographer is a standard example from cryptography in which epistemic and doxastic logics can be used to characterise the key properties of the protocol. The protocol is normally illustrated by means of the following scenarios (wording from [5]):

> *Three cryptographers are sitting down to dinner at their favorite three-star restaurant. Their waiter informs them that arrangements have been made with the maitre d'hotel for the bill to be paid anonymously. One of the cryptographers might be paying for dinner, or it*

might have been NSA (U.S. National Security Agency). The three cryptographers respect each others right to make an anonymous payment, but they wonder if NSA is paying. They resolve their uncertainty fairly by carrying out the following protocol: Each cryptographer flips an unbiased coin behind his menu, between him and the cryptographer on his right, so that only the two of them can see the outcome. Each cryptographer then states aloud whether the two coins he can see -the one he flipped and the one his left-hand neighbour flipped- fell on the same side or on different sides. If one of the cryptographers is the payer, he states the opposite of what he sees. An odd number of differences uttered at the table indicates that a cryptographer is paying; an even number indicates that NSA is paying (assuming that dinner was paid for only once). Yet if a cryptographer is paying, neither of the other two learns anything from the utterances about which cryptographer it is"

The key property of this protocol is normally encoded as:

$$AG\left((\texttt{odd} \wedge \neg\texttt{paid}_1) \rightarrow (K^1(\bigvee_{i\in\{2,3\}} \texttt{paid}_i) \wedge (\bigwedge_{i\in\{2,3\}} \neg K^1(\texttt{paid}_i))\right)$$

which is read as: if the first cryptographer did not pay for the dinner and there is an odd number of "different" utterances, then the first cryptographer knows that either the second or the third cryptographer paid for the dinner, but he does not know who is the actual payer.

Using COGWED we can strengthen this claim and state that not only the first cryptographer does not know who the payer is, but he also considers equally likely the fact that cryptographer 2 or 3 paid. This is captured by the following formula that generalises the example to n cryptographers:

$$AG\left((\texttt{odd} \wedge \neg\texttt{paid}_1) \rightarrow \left(\bigwedge_{i=2}^{n} B^1_{(=\frac{1}{n-1})}(\texttt{paid}_i)\right)\right)$$

We have implemented the model checking algorithm described above in a tool called Mc-COGWED, available from [21] (this is an extension of the tool previously published in [23]). The tool parses a text input file describing the model and takes a COGWED formula as a parameter. In this prototype implementation we employ an explicit state representation for states but we operate on equivalence classes. The code available at [21] includes a generator for instances of the dining cryptographers with a varying number

N	$\lvert S \rvert$	$\lvert R_t \rvert$	verif. time (s)
3	$5 \cdot 10^5$	96	0.12
4	$4 \cdot 10^6$	240	0.16
5	$3 \cdot 10^8$	576	0.26
6	$2 \cdot 10^{10}$	1344	0.34
7	$2 \cdot 10^{12}$	3072	0.43
8	$1.15 \cdot 10^{15}$	6912	0.51
9	$1.50 \cdot 10^{17}$	15360	0.83
10	$1.22 \cdot 10^{19}$	33792	1.22
11	$9.85 \cdot 10^{20}$	73728	4.47
12	$7.98 \cdot 10^{22}$	159744	6.19
13	$6.46 \cdot 10^{24}$	344064	27.91
14	$5.23 \cdot 10^{26}$	737280	71.23
15	$4.24 \cdot 10^{28}$	1572864	189.23
16	$2.12 \cdot 10^{30}$	3342336	443.81

Table 1. Dining cryptographers: results

of cryptographers. Experimental results are reported in Table 1. The first column represents the number of cryptographers; the second column the number of possible global states (not all of them are reachable); the third column represents the size of the temporal relation (this is the number of pairs of *reachable* states that are connected by R_t). The last column reports the time (in seconds) for the verification of the COGWED formula reported above. All the experimental results are obtained on a 8-core Intel Xeon 2.66 GHz, 32 GB of RAM Linux machine running CentOS 6, using a maximum heap size of 25 Gb.

We consider these results extremely encouraging, as they have been obtained using a prototype, non-symbolic model checker. Nevertheless, the size of the state space that can be explored by working on equivalence classes is comparable with results obtained with more mature model checkers for multi-agent systems [20, 14], even in presence of the additional doxastic operator.

6.2. Performance evaluation for DTMC-based semantics

To assess the feasibility of verifying COGWED formulae under the DTMC-based semantics in probabilistic interpreted system we have defined a simple example that can be scaled up in the number of states. This is an extension

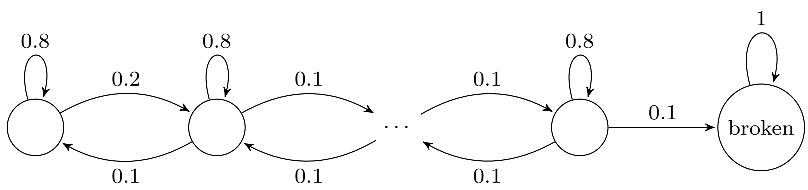

Figure 3. Example

$\|S\|$	$\|R_t\|$	verif. time (s)
100	296	0.130
200	596	0.461
300	896	0.996
400	1196	1.711
500	1496	2.737
600	1796	3.865
700	2096	5.211
800	2396	6.690
900	2696	8.554
1000	2996	10.456
2000	5996	44.615
3000	8996	103.504
4000	11996	201.702
5000	14995	290.841
10000	29996	1539.474

Table 2. Experimental results for DTMC-based semantics

of the example presented in Figure 1 and is illustrated in Figure 3. We assume that there is only one agent, that all the states are indistinguishable and that proposition "broken" is true in the final (absorbing) state. The formula we want to verify is $B^1_{\sim x}$(broken), and in particular we want to compute the value x such that $B^1_{=x}$(broken)

We have implemented the model checking algorithm described in Section 5.2 in Matlab. Experimental results are reported in Table 2 for a fixed value of $\beta = 0.99$. The first column represents the number of states N (the scaling factor); the second column represents the size of the temporal relation (according to the model, this is $3|\mathbf{S}| - 4$). The last column reports the time (in seconds) for the computation of the value x to be used inside the COGWED formula described above. All the experimental results are obtained on a 2.3 GHz Intel Core i7, 8 GB of RAM Mac machine.

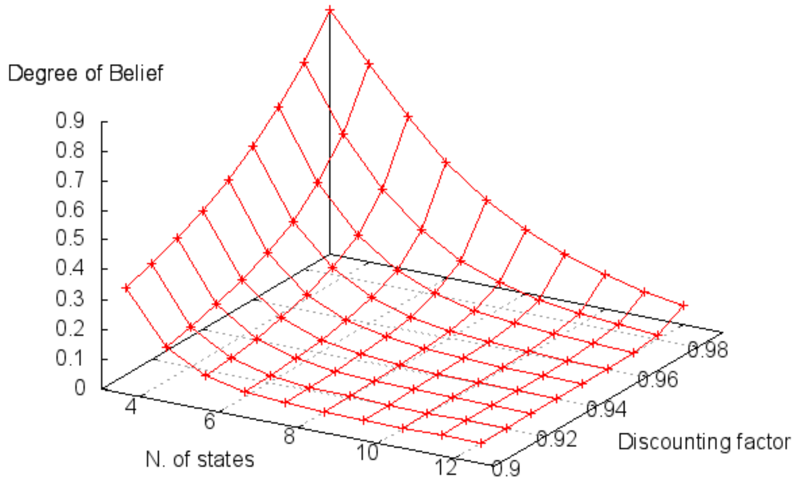

Figure 4. Degree of belief as a function of N and β

These results show that, even with a generic tool for matrix algebra, our algorithm can evaluate a substantially large example.

Figure 4 shows the variation in the degree of belief as a function of N (number of states) and β for the formula $B^1_{=x}(\text{broken})$. As expected, the degree of belief in "broken" decreases with larger values of N and increases for larger values of β.

7. Case Study: Situational Awareness

In this section we show how COGWED properties can be used to characterise and evaluate a key property in a system comprising a human and several automated components modelled as agents. In particular, we study how *situational awareness* can be assessed using COGWED. Informally, situational awareness is the ability of an agent (typically human) to determine the correct internal state of some component (or some other agent) based on his/her current beliefs. Situational awareness is a key factor for decision makers in safety-critical situations, such as airplane pilots, medical doctors, firemen, etc, and it has been investigated extensively in the past in a number or research areas, including psychology [12]. Here we focus on the aeronautic domain with a model of the Air France flight 447 from Rio de Janeiro to Paris. This is a thoroughly investigated accident involving the failure of a sensor (a set of Pitot tubes), resulting in incorrect speed readings and, through a sequence of events, to a high-altitude stall situation that failed to be correctly assessed by the pilot(s). The BAE report on the accident (`http://www.bea.aero/en/enquetes/flight.af.447/flight.af.447.php`) attributes the main cause of the accident to the inexperience of the pilot, who was not able to assess the actual speed of the airplane and, more crucially, the stall situation.

We employ a Java simulation model of the scenario taken from [1] and we modify it to generate a set of reachable states using the approach presented in [18]. The set of reachable states obtained is then encoded as a Mc-COGWED input *without* probabilities. This models the situation in which the pilots are unaware of the failure rates of the various components and as a result we adopt the counting semantics. We remark that our model does not aim at being an accurate representation of the accident; instead, our aim is to show the capabilities of COGWED in analysing *situation awareness*. In our model, a plane and its environment are characterised by:

- an actual external temperature (low, medium, high);
- an actual speed (very low, low, medium, high, very high);
- an actual vertical speed (Climbing, null, Descending);
- an actual altitude (encoded using flight levels, such as FL200, FL380 and FL450);
- an actual attitude (going up, flat, down);
- an actual thrust level (auto, 20%, 50%, TOGA, full. "TOGA" is an auto-thrust level corresponding to the thrust required for Take-Off or a Go-Around landing)

In the actual situation the pilot has access to a number of systems but he has to rely on the output of those systems to diagnose the state of the plane. We characterise the local states of the pilot by means of:

- observed temperature;
- observed speed;
- observed vertical speed;
- observed altitude;

- observed attitude.

All these values are observed by means of *sensors*, some of which may fail. When a sensor is broken, the observed value of a parameter may differ from the actual value. Additionally, a plane includes:

- an auto pilot to which the pilot has direct access, i.e., the pilot can observe whether the auto pilot is engaged or not, and we assume that the auto pilot does not fail (but the pilot may not know what caused the auto-pilot to disengage).
- a set of Pitot tubes that may be frozen when the temperature is low (but not necessarily). If the Pitot tubes are frozen, then the speed sensor is broken (but the speed sensor could be broken even when the Pitot tubes are not frozen).
- a stall warning (in the form of audio message or stick shaking, depending on the causes of the stall). Notice that *the stall warning disengages when the speed is very low* (below 60 kt), even if the plane could be actually stalling. We assume that the stall warning signal does not fail, i.e. a warning always corresponds to stalling conditions.

We model the behaviour of the pilot based on the procedures required in the various cases. For instance, if the observed speed is very high (a potentially very dangerous situation) the pilot reduces thrusts, and if the stall warning is on, the pilot modifies attitude and thrust appropriately. The Java simulation modifies the actual values of the airplane characteristics according to pilot's actions and standard physics laws, generating new states every time a value changes.

To generate the set of possible states for this scenario, we start from a situation in which the plane is flying at flight level 380 (corresponding to 38,000 feet), the thrust is 60%, the auto pilot is engaged, the stall warning is off, attitude is flat, temperature is medium and all sensors are working correctly. We then inject failures in the sensors and we generate a COGWED model covering all possible combinations reachable from the initial state. The generation is achieved by running the Java code developed in [1] and by discretising the continuous variables where required (in this case: speed, vertical speed, attitude, altitude, temperature). The number of possible discretised states is $2 \cdot 10^8$, of which approximately $1.6 \cdot 10^5$ are reachable from the initial state described above.

We can now use Mc-COGWED to evaluate the fact that the pilot is aware of a stall. In particular, we want to assess the degree of belief of a stall situation. To this end, we employ the following formula:

$$EF(\text{actualStall} \wedge B^{\text{Pilot}}_{<0.05}(\text{actualStall}))$$

This formula employs the standard EF CTL-operator and encodes the fact that there exists a state reachable from the initial state, such that the plane is actually stalling, but in that specific state the pilot believes that the stall is actually occurring with a degree of less than 5%: this formula is true in 25 states in the model. In fact, we can check that there are 5 stalling states in which the pilot believes in a stall with a degree of less than 1.5%. These are very interesting configurations that capture what may have happened on board of AF447: in these 5 states, the speed sensor is faulty (as a result of the Pitot tubes being frozen) and may report wrong measures, the attitude is UP, the speed is very low, and as a result of this low speed the stall warning remains silent. Notice that, in these specific cases, modifying the attitude to descend results in an increase in speed of the airplane, therefore re-starting the stall warning in the cabin: this is even more confusing for the pilot, as a manoeuvre that reduces the likelihood of stalling in fact generates a stall warning!

The generation of all the discretised states and its encoding as a Mc-COGWED input file require less than a minute, and Mc-COGWED can verify the formula encoding situational awareness for the stall situation in less than 8 seconds.

We argue that the doxastic pattern above can be used to characterise (the lack of) situational awareness in the general case: the formula

$$\phi \wedge B^{i}_{<\delta}\phi$$

is true in states in which ϕ holds, but agent i has a degree of belief less than δ that this is indeed the case. The parameter δ could be configured depending on the specific domain, and can be interpreted as a measure of *situational awareness*.

In the AF447 scenario, it is interesting to see how the situational awareness of a stall could be *increased*. The disengagement of the stall warning at low speed is justified by the necessity of performing low-speed operations close to the ground and to avoid spurious warnings, for instance when taking off or while landing; this, however, results in the pilot not being able to diagnose a stall at very low speed in other conditions. To address this issue, an additional visual indicator of stall warning with low speed readings could be added to the cockpit: this would be similar to ABS warnings on certain car models that remain active under 10 MPH. The additional indicator would reduce the number of possible worlds that the pilot considers possible, thereby *increasing* the minimum value of δ for which the formula above is true. This is exactly in line with the recommendations of the BAE to modify the stall management procedures on Airbuses, by re-designing the Primary Flight Display output and by adding additional training requirements in high-altitude stalling conditions.

8. Conclusion

In this paper we have presented COGWED, a logic to reason about the degrees of belief in a system of agents that is computationally grounded. We have provided two types of semantics: in one semantics degrees of belief are computed by evaluating the relative size of equivalence classes with respect to epistemic transition relations modelled in interpreted systems. In the second semantics degrees of belief are computed by evaluating the probability of reaching a set of states under the assumption that agents are memoryless and by making use of a discounting parameter in the computation of memoryless distributions. We have shown that the model checking algorithm for these two semantics remains polynomial in the size of the model and of the input formula. We have validated these complexity results by means of standard examples for both semantics. Finally, we have shown how a COGWED pattern can be used to characterise the situational awareness of a pilot flying in off-nominal conditions.

Various directions are possible for future work. We have not investigated how the belief of a group of agents could play a role in the description or verification of social interactions among agents; we plan to address this issue by exploring extensions of works such as [2]. Additionally, instead of considering distributed knowledge in the construction of the semantics of the B^Γ operator one could consider other options, such as common knowledge, thus giving rise to different forms of social interactions.

The verification of the DTMC-based semantics is currently supported only through the use of an external solver. We plan to integrate model checking algorithms currently available in PRISM [19] in the Mc-COGWED tool in the near future, thus providing a single tool for both semantics.

Acknowledgements. This paper is an extended version of material previously published in [23], with substantial new contributions: DTMC-based semantics with experimental results and implementation; new experimental evaluation to larger state spaces; new group semantics for the belief opera-

tor and corresponding new model checking algorithm; new tool release, now publicly available at `http://www.rmnd.net`.

References

[1] Agogino, A., and G. Brat, 'Statistical analysis of flight procedures', Tech. rep., NASA Ames Research Center, Moffett Field, Mountain View (CA), 2011.

[2] Alechina, Natasha, Mehdi Dastani, and Brian Logan, 'Reasoning about normative update', in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, AAAI Press, 2013, pp. 20–26.

[3] Bacchus, Fahiem, Adam J. Grove, Joseph Y. Halpern, and Daphne Koller, 'From statistical knowledge bases to degrees of belief', *Artificial Intelligence*, 87 (1996), 75–143.

[4] Bordini, Rafael H., Jomi Fred Hübner, and Michael Wooldridge, *Programming Multi-Agent Systems in AgentSpeak using Jason (Wiley Series in Agent Technology)*, John Wiley & Sons, 2007.

[5] Chaum, David, 'The dining cryptographers problem: Unconditional sender and recipient untraceability', *Journal of Cryptology*, 1 (1988), 65–75.

[6] Chen, T., V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis, 'PRISM-games: A model checker for stochastic multi-player games', in N. Piterman, and S. Smolka, (eds.), *Proc. 19th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'13)*, vol. 7795 of *LNCS*, Springer, 2013, pp. 185–191.

[7] Chen, Taolue, Vojtech Forejt, Marta Z. Kwiatkowska, David Parker, and Aistis Simaitis, 'Automatic verification of competitive stochastic systems', *Formal Methods in System Design*, 43 (2013), 1, 61–92.

[8] Clarke, Edmund M, Orna Grumberg, and Doron A Peled, *Model checking*, MIT press, 1999.

[9] de Carvalho Ferreira, Nivea, Michael Fisher, and Wiebe Van Der Hoek, 'Specifying and reasoning about uncertain agents', *International Journal of Approximate Reasoning*, 49 (2008), 1, 35–51.

[10] Dennis, Louise A, Michael Fisher, Matthew P Webster, and Rafael H Bordini, 'Model checking agent programming languages', *Automated Software Engineering*, 19 (2012), 1, 5–63.

[11] Fagin, Ronald, Joseph Y Halpern, Yoram Moses, and Moshe Y Vardi, *Reasoning about knowledge*, MIT press Cambridge, 1995.

[12] French, Han Tin, Elizabeth Clarke, Diane Pomeroy, Melanie Seymour, and C Richard Clark, 'Psycho-physiological measures of situation awareness', *Decision Making in Complex Environments*, (2007), 291.

[13] Friedman, Nir, and Joseph Y Halpern, 'Plausibility measures and default reasoning', *Journal of the ACM*, 48 (2001), 4, 648–685.

[14] Gammie, Peter, and Ron Van Der Meyden, 'MCK: Model checking the logic of knowledge', in *Computer Aided Verification*, Springer, 2004, pp. 479–483.

[15] Halpern, Joseph Y, *Reasoning about uncertainty*, The MIT Press, 2003.

[16] Halpern, Joseph Y., and Samantha Leung, 'Weighted sets of probabilities and minimaxweighted expected regret: New approaches for representing uncertainty and making decisions', in *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, August 14-18, 2012*, 2012, pp. 336–345.

[17] Huang, Xiaowei, and Cheng Luo, 'A logic of probabilistic knowledge and strategy', in *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13, Saint Paul, MN, USA, May 6-10, 2013*, 2013, pp. 845–852.

[18] Hunter, Josie, Franco Raimondi, Neha Rungta, and Richard Stocker, 'A synergistic and extensible framework for multi-agent system verification', in *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13, Saint Paul, MN, USA*, 2013, pp. 869–876.

[19] Kwiatkowska, Marta, Gethin Norman, and David Parker, 'Prism 4.0: Verification of probabilistic real-time systems', in Ganesh Gopalakrishnan, and Shaz Qadeer, (eds.), *Computer Aided Verification*, vol. 6806 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2011, pp. 585–591.

[20] Lomuscio, Alessio, Hongyang Qu, and Franco Raimondi, 'Mcmas: A model checker for the verification of multi-agent systems', in Ahmed Bouajjani, and Oded Maler, (eds.), *Computer Aided Verification*, vol. 5643 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2009, pp. 682–688.

[21] 'Mc-cogwed', `http://www.rmnd.net/software/mc-cogwed/`. Accessed October 21, 2014.

[22] Norris, J. R., *Markov Chains*, Cambridge University Press, 1998.

[23] Primiero, Giuseppe, Franco Raimondi, and Neha Rungta, 'Model checking degrees of belief in a system of agents', in *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 133–140.

[24] Richardson, Matthew, and Pedro Domingos, 'Markov logic networks', *Machine learning*, 62 (2006), 1-2, 107–136.

[25] Shafer, Glenn, *A mathematical theory of evidence*, vol. 1, Princeton university press Princeton, 1976.

[26] Varga, Richard S., *Matrix Iterative Analysis (Springer Series in Computational Mathematics)*, Springer, 2009.

[27] Webster, Matt, Neil Cameron, Michael Fisher, and Mike Jump, 'Generating certification evidence for autonomous unmanned aircraft using model checking and simulation', *Journal of Aerospace Information Systems*, 11 (2014), 5, 258–279.

[28] Wooldridge, Michael, 'Computationally grounded theories of agency', in *Proceedings of ICMAS, International Conference of Multi-Agent Systems*, IEEE Press, 2000, pp. 13–20.

[29] Wooldridge, Michael, *An introduction to multiagent systems*, John Wiley & Sons, 2009.

Taolue Chen, Giuseppe Primiero, Franco Raimondi
Department of Computer Science
Middlesex University
The Burroughs
London, UK
{T.Chen|G.Primiero|F.Raimondi}@mdx.ac.uk

Neha Rungta
NASA Ames Research Center
Moffett Field CA 94035
US
neha.s.rungta@nasa.gov