

Quantitative Group Trust: A Two-Stage Verification Approach

Jamal Bentahar
Concordia University
Montreal, QC, Canada
jamal.bentahar@concordia.ca

Nagat Drawel
Concordia University
Montreal, QC, Canada
n_drawe@encs.concordia.ca

Abdeladim Sadiki
Concordia University
Montreal, QC, Canada
abdeladim.sadiki@mail.concordia.ca

ABSTRACT

This paper is about modeling and verifying quantitative group trust. We present a formal analysis of this concept that allows us to express and reason about trust in multi-agent systems in a quantitative setting. We introduce GTL, the graded branching temporal logic that includes operators for quantitative aspects of trust within a group. A two-stage verification procedure of the logic is presented. The first stage is a transformation procedure, and the second stage is an indirect procedure that uses an existing model checking algorithm. Theoretical results about the soundness, completeness and complexity of the procedure are presented.

KEYWORDS

Quantitative Trust; Group Trust; Model Checking

ACM Reference Format:

Jamal Bentahar, Nagat Drawel, and Abdeladim Sadiki. 2022. Quantitative Group Trust: A Two-Stage Verification Approach. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022)*, Online, May 9–13, 2022, IFAAMAS, 9 pages.

1 INTRODUCTION

Evaluating trust in Multi-Agent Systems (MASs) is of prime importance for the success of agent-based applications where agents interact and operate in uncertain and dynamic environments. Although plenty of logical frameworks that handle trust in MASs have been widely analyzed in the literature [13, 19, 25, 32, 34, 38], most of these approaches treated trust in an absolute manner, i.e., we either fully trust the behavior of an agent or definitely not. However, in many contexts, it is quite difficult to determine with absolute certainty whether a proposition about the behavior of an agent is true or false. For instance, we might trust the agent to a certain degree in relation to such a proposition (i.e., we may have only 50% of trust). That is, although qualitative logical formalisms allow us to reason about various classical properties, their expressiveness is limited in representing some important aspects that deal with the way of capturing our perception of reality [21].

In this paper, we propose GTL, a temporal logic that includes, in addition to CTL operators, modalities expressing graded trust that quantifies individual, group and propagated trust. GTL expresses properties about degrees of trust involving individual agents and groups. The idea of quantifying trust has attracted the attention of several researchers in different domains. Most existing approaches consider trust as a function calculated based on multiple opinions through feedback, user ratings, or agent monitoring [6, 36, 40]. Such approaches represent and quantify the strength level in which an

agent trusts another party. Nevertheless, in dynamic MASs where agents may join for a short period of time before leaving the interaction, it might be difficult to collect sufficient data to evaluate the trustworthiness of partners. Instead of relying only on external measurable evaluations, it might be appealing to enable agents to reason about their degrees of trust to make better decisions.

We introduce a weighted logical formalism and show how the new operators contribute in expressing and reasoning about quantitative trust properties. These operators are coupled with a trust degree that counts along a run the proportion of the set of states satisfying trust formulae among all states that are reachable and accessible using the trust accessibility relation. GTL allows us to express properties such as "agent i trusts that agent j will eventually deliver the requested items in at most 75% of the cases", "In all future runs, a group of agents has 50% of trust that agent j will send the payment". Although the standard approach of trust quantification involves the use of probability mechanisms accompanied with a representation of agent's beliefs [1, 20, 30, 33], GTL uses a different approach that quantifies trust by relying only on accessibility relations. Further, we introduce an efficient technique for model checking GTL. In particular, we present a two-stage procedure that first transforms the GTL model checking to another model checking problem and then solves the latter problem by calling an efficient model checking procedure of a third language. The two-stages make the procedure more natural and easy to follow. Moreover, we use this procedure to show that the complexity of GTL model checking for concurrent programs is PSPACE-complete with respect to the size of the program's components. The two-stage procedure we introduce for model checking and its complexity provide a general methodology that can be useful for other graded logics.

2 GTL: GRADED GROUP TRUST LOGIC

DEFINITION 1 (SYNTAX OF GTL). GTL combines CTL introduced in [7] with modalities for graded trust as follows:

$$\begin{aligned} \varphi &::= \rho \mid \neg\varphi \mid \varphi \vee \varphi \mid EX\varphi \mid E(\varphi U \varphi) \mid A(\varphi U \varphi) \mid T \\ T &::= I^{\Delta k}(i, j, \varphi) \mid E^{\Delta k}(G, j, \varphi) \mid D^{\Delta k}(G, j, \varphi) \mid P^{\Delta k}(i, j, \mathcal{G}, \varphi) \end{aligned}$$

where $\rho \in AP$ is an atomic variable; E and A are the existential and universal quantifiers on paths; X and U are CTL path modal connectives standing for "next" and "until" respectively; the Boolean connectives \neg and \vee are defined and read in the usual way; the graded trust operators represented by T are to model the graded trust of agents. There are four graded trust modalities: $I^{\Delta k}$, $E^{\Delta k}$, $D^{\Delta k}$, and $P^{\Delta k}$ that represent degrees of trust for Individual, Everyone, Distributed, and Propagated trust respectively. From the syntax perspective, $I^{\Delta k}(i, j, \varphi)$ expresses that "the truster i trusts the trustee j about φ with a degree of trust Δk ", where k is a rational number in $[0, 1]$, and Δ is a relation symbol in the set $\{\leq, \geq, <, >, =\}$.

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online. © 2022 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

The formula $E^{\Delta k}(G, j, \varphi)$ means that everyone in the group G trusts the trustee j about φ with the degree Δk . Moreover, $D^{\Delta k}(G, j, \varphi)$ denotes the distributed graded trust where the group G has its trust degree Δk towards j about φ distributed among its member agents. Propagated trust $P^{\Delta k}(i, j, \mathcal{G}, \varphi)$ expresses trust about φ that propagates from i to j in a chain of propagation \mathcal{G} with a degree Δk . The propagation chain \mathcal{G} is represented as an ordered list of agents ($\mathcal{G} = [i_1, \dots, i_n]$). It is worth pointing that GTL abstracts from the ability of the trustee to bring about the trust content, which will be investigated in future work considering similar concepts in [15].

GTL formulæ are interpreted on the extended formalism of Interpreted Systems [13], which extends the original Interpreted Systems [17]. This formalism is enriched with trust function which associates to each local state $l_i \in L_i$ of each agent $i \in \text{Agt}$ in the global state $s \in S$ the trust vision of the truster towards other agents in the respective state. This vision is recorded in a data structure (a vector) as values the truster associates to the other members of the system. This data structure is part of each local state of every agent in the system. Specifically, the trust function gives rise to a binary relation between two states which in fact combines the reachability and the compatibility of the local states with respect to the recorded values. These values are used to define the individual, group and propagated trust accessibility relations $\sim_{i \rightarrow j}$, $\sim_{G \rightarrow j}^E$, $\sim_{G \rightarrow j}^D$, and $\sim_{i \rightarrow j, \mathcal{G}}^P$ in a grounded semantics setting [11].

Intuitively, the relation $\sim_{i \rightarrow j}$ relates the states that are trust-compatible from the vision of agent i with regard to agent j . This trust-compatibility is computed using the recorded values in the agent vector v . For two global states $s, s' \in S$, $s \sim_{i \rightarrow j} s'$ is obtained by comparing the value of agent j in agent i 's vector (denoted by $v^i(j)$) in the local state l_i at the global state s (denoted by $l_i(s)(v^i(j))$) with $v^i(j)$ in the local state l_i at the global state s' (i.e., $l_i(s')(v^i(j))$). Thus, the trust accessibility of agent i towards agent j (i.e., $\sim_{i \rightarrow j}$) does exist only if the element values that we have for agent j in the vector of the local states of agent i for both global states s and s' are the same, i.e., $l_i(s)(v^i(j)) = l_i(s')(v^i(j))$. The accessibility relations $\sim_{G \rightarrow j}^E$, $\sim_{G \rightarrow j}^D$, and $\sim_{i \rightarrow j, \mathcal{G}}^P$ are as follows. $\sim_{G \rightarrow j}^E = \bigcup_{i \in G} \sim_{i \rightarrow j}$ is the union of the trust accessibility relations between every agent of the group G and the agent j . $\sim_{G \rightarrow j}^D = \bigcap_{i \in G} \sim_{i \rightarrow j}$ is the intersection of the trust accessibility relations between every agent in the group G and the agent j . Finally, $s \sim_{i \rightarrow j, \mathcal{G}}^P s'$ iff there are states s_1, \dots, s_n s.t. $s \sim_{i, i_1} s_1, s_1 \sim_{i_1, i_2} s_2, \dots, s_n \sim_{i_n, j} s'$, where $\mathcal{G} = [i_1, \dots, i_n]$.

DEFINITION 2 (MODEL OF GTL). *The model of GTL is a tuple: $M_G = (S_G, I_G, R_G, \{\sim_{i \rightarrow j} \mid (i, j) \in \text{Agt}^2\}, V_G)$ where:*

- S_G is a non-empty set of reachable global states of the system. Each global state s is defined as a tuple $s = (l_1, \dots, l_n) \in (L_1 \times \dots \times L_n)$;
- $I_G \subseteq S_G$ is a set of initial global states;
- $R_G \subseteq S_G \times S_G$ is the transition relation;
- $\sim_{i \rightarrow j} \subseteq S_G \times S_G$ is the individual trust accessibility relation for each truster-trustee pair of agents $(i, j) \in \text{Agt}^2$ defined by $s \sim_{i \rightarrow j} s'$ iff: $l_i(s)(v^i(j)) = l_i(s')(v^i(j))$, and s' is reachable from s using transitions from the transition relation R ;

- $V_G : S_G \rightarrow 2^{AP}$ is a labeling function, where AP is a set of atomic propositions.

In this model, infinite sequences of states linked by transitions define paths. If π is a path, then $\pi(i)$ is the $(i + 1)^{th}$ state in π .

DEFINITION 3 (SEMANTICS OF GTL). *Given the model M_G , the satisfaction of a GTL formula φ in a global state s , denoted as $(M_G, s) \models \varphi$, is recursively defined as follows:*

- $(M_G, s) \models \rho$ iff $\rho \in V_G(s)$;
- $(M_G, s) \models \neg \varphi$ iff $(M_G, s) \not\models \varphi$;
- $(M_G, s) \models \varphi_1 \vee \varphi_2$ iff $(M_G, s) \models \varphi_1$ or $(M_G, s) \models \varphi_2$;
- $(M_G, s) \models EX \varphi$ iff there exists a path π starting at s such that $(M_G, \pi(1)) \models \varphi$;
- $(M_G, s) \models E(\varphi_1 U \varphi_2)$ iff there exists a path π starting at s such that for some $k \geq 0$, $(M_G, \pi(k)) \models \varphi_2$ and $\forall 0 \leq i < k$, $(M_G, \pi(i)) \models \varphi_1$;
- $(M_G, s) \models A(\varphi_1 U \varphi_2)$ iff for all paths π starting at s , there exists some $k \geq 0$ such that $(M_G, \pi(k)) \models \varphi_2$ and $\forall 0 \leq i < k$, $(M_G, \pi(i)) \models \varphi_1$;
- $(M_G, s) \models I^{\Delta k}(i, j, \varphi)$ iff $\exists s' \neq s$ such that $s \sim_{i \rightarrow j} s'$ and $\frac{|s \sim_{i \rightarrow j} s' : s' \neq s \ \& \ (M_G, s') \models \varphi|}{|s \sim_{i \rightarrow j} s' : s' \neq s|} \Delta k$;
- $(M_G, s) \models E^{\Delta k}(G, j, \varphi)$ iff $\exists s' \neq s$ such that $s \sim_{G \rightarrow j}^E s'$ and $\frac{|s \sim_{G \rightarrow j}^E s' : s' \neq s \ \& \ (M_G, s') \models \varphi|}{|s \sim_{G \rightarrow j}^E s' : s' \neq s|} \Delta k$;
- $(M_G, s) \models D^{\Delta k}(G, j, \varphi)$ iff $\exists s' \neq s$ such that $s \sim_{G \rightarrow j}^D s'$ and $\frac{|s \sim_{G \rightarrow j}^D s' : s' \neq s \ \& \ (M_G, s') \models \varphi|}{|s \sim_{G \rightarrow j}^D s' : s' \neq s|} \Delta k$;
- $(M_G, s) \models P^{\Delta k}(i, j, \mathcal{G}, \varphi)$ iff $\exists s' \neq s$ such that $s \sim_{i \rightarrow j, \mathcal{G}}^P s'$ and $\frac{|s \sim_{i \rightarrow j, \mathcal{G}}^P s' : s' \neq s \ \& \ (M_G, s') \models \varphi|}{|s \sim_{i \rightarrow j, \mathcal{G}}^P s' : s' \neq s|} \Delta k$.

For atomic propositions, Boolean connectives, and temporal modalities, the relation \models is defined in the standard manner (see for example [7]). The intuition behind the semantics of $I^{\Delta k}(i, j, \varphi)$ is: the degrees of trust that an agent associates to a formula φ in a global state s is the ratio between the number of states s' distinguishable and accessible from s and satisfying φ (i.e., $|s \sim_{i \rightarrow j} s' : s' \neq s \ \& \ s' \models \varphi|$), and the total number of distinguishable and accessible states from s (i.e., $|s \sim_{i \rightarrow j} s' : s' \neq s|$). The semantics of $E^{\Delta k}(G, j, \varphi)$, $D^{\Delta k}(G, j, \varphi)$, and $P^{\Delta k}(i, j, \mathcal{G}, \varphi)$ follow the same approach with respect to the appropriate accessibility relation.

Unlike the BT logic introduced in [11], $I^{\Delta k}(i_1, j, \varphi) \wedge I^{\Delta k}(i_2, j, \varphi) \Rightarrow \psi$ does not necessarily entail $D^{\Delta k}(\{i_1, i_2\}, j, \psi)$. In fact, it is counter-intuitive to have a distributed trust towards j about ψ if the two agents in the group i_1 and i_2 trust j about $\neg \psi$, which could happen in [11] if $\bigcap_{i \in \{i_1, i_2\}} \sim_{i \rightarrow j} = \emptyset$. The emptiness of this intersection also makes the formula $D_T(G, j, \perp)$ satisfiable in BT where D_T is the BT's distributed trust operator. Similarly, if one of the group members trusts the trustee about φ (i.e., $\bigvee_{i \in G} I^{\Delta k}(i, j, \varphi)$), the distributed trust about φ ($D^{\Delta k}(G, j, \varphi)$) does not necessarily hold in GTL, unlike BT, because agents might have contradicting trusts.

Because the emptiness of the set of accessible states does not allow the formula to hold in GTL makes it free from these inconsistencies.

Illustrating Examples

We consider the Breast Cancer Diagnosis and Treatment protocol that involves the following parties: patient (*PAT*), physician (*PHY*), and radiologist (*RAD*). For example, the patient will eventually trust the physician with a certain degree k about their recommendation to have a mammography (φ_{Mam}): $AF I^{\Delta k}(PAT, PHY, \varphi_{Mam})$.

Another example is when a group of physicians (*GPHY*) trusts, in a distributed manner, the radiologist about their diagnosis of recommending a breast biopsy (φ_{BB}). This trust is distributed among the physicians based on their individual experiences with the radiologist. This could happen when each individual physician has only a limited experience with the radiologist to evaluate the quality of their diagnosis, but when these experiences are put together, the physicians as a group could have a better appreciation of the radiologist diagnosis: $AF D^{\Delta k}(GPHY, RAD, \varphi_{BB})$.

A third example is when the patient trusts the radiologist through their physician about the minimally-invasive procedure (φ_{MIP}) that the radiologist would perform. In some cases, the radiologist would directly intervene through this type of procedure in a hospital or a clinic. The patient can trust the radiologist about this intervention based on the trust the patient has on the physician who recommended the radiologist: $EF P^{\Delta k}(PAT, RAD, [PYH], \varphi_{MIP})$.

3 REASONING POSTULATES

We consider in this section relevant postulates that reflect properties desirable for reasoning about graded trust in multi-agent systems. Two categories are presented: 1) specific postulates for individual, propagated, distributed and everyone trust; and 2) common postulates that are shared by all the four trust operators.

For the common postulates, we use $T^{\Delta k}$ as a common operator that replaces $I^{\Delta k}$, $D^{\Delta k}$, $E^{\Delta k}$, and $P^{\Delta k}$. Moreover, we omit the arguments that represent the participating agents as far as they are understood, so we simply write $T^{\Delta k}(\varphi)$.

3.1 Specific Postulates

Let \uplus be the concatenation operator over propagation chains.

- (1) $I^{>0}(i, j_1, I^{>0}(j_1, j_2, \varphi)) \Rightarrow P^{>0}(i, j_2, [j_1], \varphi)$.
- (2) $P^{>0}(i, j_1, \mathcal{G}, I^{>0}(j_1, j_2, \varphi)) \Rightarrow P^{>0}(i, j_2, \mathcal{G} \uplus [j_1], \varphi)$
- (3) $I^{>0}(i, j_1, P^{>0}(j_1, j_2, \mathcal{G}, \varphi)) \Rightarrow P^{>0}(i, j_2, [j_1] \uplus \mathcal{G}, \varphi)$
- (4) $P^{>0}(i, j_1, \mathcal{G}, P^{>0}(j_1, j_2, \mathcal{G}', \varphi)) \Rightarrow P^{>0}(i, j_2, \mathcal{G} \uplus [j_1] \uplus \mathcal{G}', \varphi)$
- (5) $(I^{\geq 1}(i_1, j, \varphi) \wedge I^{\geq 1}(i_2, j, \varphi \Rightarrow \psi)) \Rightarrow (D^{\geq 0}(\{i_1, i_2\}, j, \varphi) \Rightarrow D^{\geq 1}(\{i_1, i_2\}, j, \psi))$
- (6) $(\bigvee_{i \in G} I^{\geq 1}(i, j, \varphi)) \Rightarrow (D^{\geq 0}(G, j, \varphi) \Rightarrow D^{\geq 1}(G, j, \varphi))$
- (7) $\bigwedge_{i \in G} I^{>0}(i, j, \varphi) \Rightarrow E^{>0}(G, j, \varphi)$
- (8) $E^{\geq 1}(G, j, \varphi) \Leftrightarrow \bigwedge_{i \in G} I^{\geq 1}(i, j, \varphi)$

The first four postulates show how propagated trust with a certain degree is obtained through the combination of individual and propagated trust. The first postulate indicates that propagated trust is obtained from nested individual trust where the propagation chain is formed by the truster of the nested trust, which is also the trustee of the main trust. The second postulate shows that

combining propagated and individual trust yields a propagated trust where the propagation chain is expanded by the truster of the nested individual trust. The third postulate shows the dual case where nesting a propagated trust inside an individual trust yields a propagated trust with an expanded propagation chain through the chain of the nested formula. The fourth postulate follows the same pattern where propagated trusts are combined. The fifth postulate is about distributed trust about ψ that can be built if one member trusts the trustee about φ and the other member trusts the same trustee about $\varphi \Rightarrow \psi$ under the condition that a distributed trust involving the two trusters does exist. This condition avoids the inconsistency discussed in Section 2. Postulate (6) shows that distributed trust is fully established if one agent of the group trusts the trustee under the condition that this distributed trust might hold, which also avoids the previously discussed inconsistency. Finally, the last two postulates are about establishing everyone trust from the conjunction of individual trusts.

3.2 Common Postulates

$$(1) T^{\geq 1}(\varphi_1) \wedge T^{\Delta k}(\varphi_2) \Rightarrow T^{\Delta k}(\varphi_1 \wedge \varphi_2)$$

Meaning: If we have a certain trust degree on φ_1 and we trust φ_2 with a certain probability Δk , then we can infer the conjunction following the same degree Δk . For instance, if the trust in one part is impossible (≤ 0), then the impossibility of the conjunction follows.

$$(2) T^{\leq k}(\varphi_1) \Rightarrow T^{\leq k}(\varphi_1 \wedge \varphi_2)$$

Meaning: Whenever there is trust on φ_1 bounded by an upper degree k , the conjunction with any other φ_2 is also bounded by k .

The following is a direct result of this postulate:

$$(3) T^{\leq k_1}(\varphi_1) \wedge T^{\leq k_2}(\varphi_2) \Rightarrow T^{\leq \min(k_1, k_2)}(\varphi_1 \wedge \varphi_2)$$

$$(4) T^{\leq k_1}(\varphi_1) \wedge T^{\Delta k_2}(\varphi_2) \Rightarrow T^{\leq \max(k_1, k_2)}(\varphi_1 \wedge \varphi_2)$$

Meaning: If we have a trust on φ_1 bounded by a degree k_1 and a trust on φ_2 with a degree of Δk_2 , then the trust on the conjunction is bounded by the maximum of k_1 and k_2 .

$$(5) T^{\leq k_1}(\varphi_1 \wedge \varphi_2) \Rightarrow \exists k_2 \geq k_1 \text{ s.t. } T^{\leq k_2}(\varphi_1)$$

Meaning: The degree of trust to bring about a conjunction is lower than or equal to the degree to bring about its components.

$$(6) T^{\geq k}(\varphi_1 \wedge \varphi_2) \Rightarrow T^{\geq k}(\varphi_1)$$

Meaning: The degree of trust to bring about the components of a conjunction cannot be less than the degree to bring about the conjunction itself.

$$(7) T^{\geq k_1}(\varphi_1) \wedge T^{\geq k_2}(\varphi_2) \Rightarrow T^{\geq \max(k_1 + k_2 - 1, 0)}(\varphi_1 \wedge \varphi_2)$$

Meaning: The trust to bring about a conjunction where the trust about the components has lower bounds k_1 and k_2 is lower bounded by $\max(k_1 + k_2 - 1, 0)$.

$$(8) T^{\leq k}(\varphi_1 \vee \varphi_2) \Rightarrow T^{\leq k}(\varphi_1)$$

Meaning: If trust about a disjunction holds with a degree $\leq k$, then the trust about the components holds with the same degree.

$$(9) T^{\geq k}(\varphi_1) \Rightarrow T^{\geq k}(\varphi_1 \vee \varphi_2)$$

Meaning: The trust about a disjunction holds with a lower limit if the trust about one of the components holds with the same limit.

$$(10) T^{\geq k_1}(\varphi_1) \vee T^{\geq k_2}(\varphi_2) \Rightarrow T^{\geq \min(k_1, k_2)}(\varphi_1 \vee \varphi_2)$$

Meaning: The degree of trusting the disjunction cannot be more than the degree of trusting each component.

$$(11) T^{\geq k}(\varphi_1 \vee \varphi_2) \Rightarrow \exists k_1, k_2 \text{ s.t.}$$

$$T^{\geq k_1}(\varphi_1) \wedge T^{\geq k_2}(\varphi_2) \wedge k_1 + k_2 \geq k$$

$$(12) T^=k(\varphi) \Rightarrow T^{=1-k}(\neg\varphi)$$

Meaning: Trust about a content and its negation are complement.

$$(13) (T^{\Delta k}(\varphi) \wedge AG(\varphi \Rightarrow \psi)) \Rightarrow T^{\Delta k}(\psi)$$

Meaning: If there is trust with a degree Δk about φ and globally φ entails ψ , then trust about ψ holds with the same degree.

$$(14) \text{ From } T^{\geq k}(\varphi_1) \text{ and } \varphi_1 \vdash \varphi_2 \text{ infer } T^{\geq k}(\varphi_2)$$

Meaning: Trust about φ_2 with a degree $\geq k$ yields if the trust about a content from which φ_2 derives holds with the same degree.

$$(15) \text{ From } T^{\leq k_1}(\varphi_1) \text{ and } \varphi_1 \vdash \varphi_2 \text{ infer } \exists k_2 \geq k_1 \text{ s.t. } T^{\leq k_2}(\varphi_2)$$

There is consistency in the way the graded trust is captured and the possibility-necessity dual concepts in Possibilistic Logic [14]. However, the main difference with our logic is that Possibilistic Logic is a tool for reasoning about uncertainty using an ordering over the possible worlds where degrees of possibility and necessity are closely related to fuzzy sets to capture vagueness rather than counting the accessible states as in our logic.

4 MODEL CHECKING GTL

4.1 Description of the Approach

In this section, we will present a model checking technique for GTL¹. Our technique is a two-stage procedure. In the first stage, we transform the problem of model checking GTL into the problem of model checking ARCTL_τ, a new logic that we define in this paper. In the second stage, the problem of model checking this new logic is addressed by calling the model checking procedure of Action-Restricted CTL (ARCTL) introduced in [35]. We will introduce the transformation function f_1 for the first stage and the model checking algorithm MC_{τ} for the second stage. The transformation function f_1 includes rules for transforming the model and formulae from the source language GTL to the target language ARCTL_τ. Then, the model checking algorithm MC_{τ} takes an ARCTL_τ formula Φ and an ARCTL_τ model as input and computes $[[\Phi]]$, the set of states of this model satisfying the formula.

ARCTL is an extension of CTL with action formulae. We use these actions to capture the accessibility relations in the original GTL model. ARCTL_τ merges a fragment of the Counting CTL logic (CCTL) [24] with ARCTL. The reason of using ARCTL_τ as intermediate language for our transformation procedure is that CCTL counts the number of states satisfying certain sub-formulae along paths and uses this number as a constraint of the until temporal operator. This allows us to capture the number of accessible states used to define the semantics of GTL. Finally, by merging a fragment of CCTL with ARCTL, we capture the accessibility relations in the first stage through action-labeled transitions and use these transitions in the second stage toward the target language.

Before introducing ARCTL_τ, we briefly review ARCTL [35].

DEFINITION 4 (SYNTAX OF ARCTL).

$$\varphi ::= \rho \mid \neg\varphi \mid \varphi \vee \varphi \mid E_{\alpha}X\varphi \mid E_{\alpha}(\varphi U \varphi) \mid A_{\alpha}(\varphi U \varphi)$$

φ is a state formula and α is an atomic action formula ($\alpha \in AC_A$ the set of atomic actions). Instead of considering composed action formulae as in the original ARCTL logic, we only consider here atomic actions, which are enough to capture the labeled transitions in this paper. ARCTL restricts path quantifiers with an action formulae

that must be satisfied along the path (i.e., labeling each transition of the path) in order to determine the precise paths over which path formulae are evaluated.

DEFINITION 5 (MODEL OF ARCTL). *The model of ARCTL is a tuple $M_A = (S_A, AC_A, I_A, R_A, V_A)$ where S_A is a nonempty set of states; $I_A \subseteq S_A$ is a set of initial states; $R_A \subseteq S_A \times AC_A \times S_A$ is a labeled transition relation; $V_A : S_A \rightarrow 2^{AP}$ is a function labeling states with subsets of atomic propositions AP .*

A path of M_A is an infinite sequence of states and actions. $\Pi^{\alpha}(s)$ is the set of paths (called α -paths) starting at s and where all transitions are labeled with the atomic action α .

The satisfaction relation $(M_A, s) \models \varphi$ is given as follows (we omit the semantics of Boolean connectives and propositional atoms):

- $(M_A, s) \models E_{\alpha}X\varphi$ iff there exists a path $\pi \in \Pi^{\alpha}(s)$ and $(M_A, \pi(1)) \models \varphi$;
- $(M_A, s) \models E_{\alpha}(\varphi_1 U \varphi_2)$ iff there exists a path $\pi \in \Pi^{\alpha}(s)$ such that for some $k \geq 0$, $(M_A, \pi(k)) \models \varphi_2$ and $(M_A, \pi(j)) \models \varphi_1$ for all $0 \leq j < k - 1$;
- $(M_A, s) \models A_{\alpha}(\varphi_1 U \varphi_2)$ iff for all paths $\pi \in \Pi^{\alpha}(s)$ there exists some $k \geq 0$, such that $(M_A, \pi(k)) \models \varphi_2$ and $(M_A, \pi(j)) \models \varphi_1$ for all $0 \leq j < k - 1$.

DEFINITION 6 (SYNTAX OF ARCTL_τ).

$$\varphi ::= \rho \mid \neg\varphi \mid \varphi \vee \varphi \mid E_{\alpha}X\varphi \mid E_{\alpha}(\varphi U_{[c]}\varphi) \mid A_{\alpha}(\varphi U_{[c]}\varphi)$$

$$c ::= \frac{\#\varphi}{\tau} \Delta k;$$

α is an atomic action formula as in Definition 4, Δk is as in Definition 1, and c is a constraint based on counting the number of states satisfying φ ($\#\varphi$) and τ is a strictly positive natural number.

We only use a fragment of CCTL where only one formula (φ) is counted in the constraint through the division operator instead of counting different formulae (φ_i) and going through the sum of their corresponding states. $\#\varphi$ captures the number of states satisfying φ along a given prefix and τ represents the total number of states of that prefix (the formal definition will follow).

ARCTL_τ uses the standard abbreviations. For instance: $E_{\alpha}F_{[c]}\varphi = E_{\alpha}(\top U_{[c]}\varphi)$, $A_{\alpha}F_{[c]}\varphi = A_{\alpha}(\top U_{[c]}\varphi)$, $A_{\alpha}G_{[c]}\varphi = \neg E_{\alpha}F_{[c]}\neg\varphi$. The size of a formula takes into account the size of the constraint formula. For instance $|E_{\alpha}(\varphi_1 U_{[c]}\varphi_2)| = |E_{\alpha}(\varphi_1 U \varphi_2)| + |\varphi|$.

DEFINITION 7 (MODEL OF ARCTL_τ). *ARCTL_τ is interpreted over a labeled Kripke structure $M_{\tau} = (S_{\tau}, AC_{\tau}, I_{\tau}, R_{\tau}, V_{\tau})$ where S_{τ} is a nonempty set of states; AC_{τ} is a set of atomic actions; I_{τ} is a set of initial states; $R_{\tau} \subseteq S_{\tau} \times AC_{\tau} \times S_{\tau}$ is a labeled transition relation; $V_{\tau} : S_{\tau} \rightarrow 2^{AP}$ is a valuation function.*

The satisfaction relation $(M_{\tau}, s) \models \varphi$ is given as follows (we omit the semantics of propositional atoms, Boolean connectives and the next operator):

- $(M_{\tau}, s) \models E_{\alpha}(\varphi_1 U_{[c]}\varphi_2)$ iff there exists a path $\pi \in \Pi^{\alpha}(s)$ s.t. for some $i \geq 0$, $(M_{\tau}, \pi(i)) \models \varphi_2$ and $(M_{\tau}, \pi(i-1)) \models c$ and for all $0 \leq j < i$, $(M_{\tau}, \pi(j)) \models \varphi_1$;
- $(M_{\tau}, s) \models A_{\alpha}(\varphi_1 U_{[c]}\varphi_2)$ iff for all paths $\pi \in \Pi^{\alpha}(s)$, there is some $i \geq 0$, $(M_{\tau}, \pi(i)) \models \varphi_2$ and $(M_{\tau}, \pi(i-1)) \models c$ and for all $0 \leq j < i$, $(M_{\tau}, \pi(j)) \models \varphi_1$.

¹https://drive.google.com/drive/folders/11MXKPhwH1_UrG5M0LsqSxxICDfu-U0rt?usp=sharing

where $\Pi^\alpha(s)$ is the set of α paths starting at s , $\pi(i)$ is the state s_i of the path π and $\pi_{(i)}$ is the flat prefix $s_0 \dots s_i$ of π that does not contain loops. Notice that $\pi_{(-1)} = \epsilon$ is the empty flat prefix.

For every flat prefix $\pi_{(i)} = s_0 \dots s_i$, the meaning of $(M_\pm, \pi_{(i)}) \models c$ is based on the interpretation of $\frac{\#\varphi}{\tau}$ over $\pi_{(i)}$, which is the number of states among $s_0 \dots s_i$ satisfying φ over the size τ of the prefix ($\tau = i + 1$), or formally, $\frac{|\{j | 0 \leq j \leq i \wedge \pi(j) \models \varphi\}|}{i+1}$. Notice that $\tau = 1$ when the prefix is empty, which is also the case when the prefix has only one element. Under this semantics, $E_\alpha(\varphi_1 U \varphi_2)$ is equivalent to $E_\alpha(\top U \frac{\#\neg\varphi_1}{[\frac{\tau}{\tau}=0]} \varphi_2)$ and $A_\alpha(\varphi_1 U \varphi_2)$ is equivalent to $A_\alpha(\top U \frac{\#\neg\varphi_1}{[\frac{\tau}{\tau}=0]} \varphi_2)$.

4.2 Stage 1: Transformation f_1

We proceed now with the first stage that consists of the transformation function f_1 (i.e., from GTL to ARCCCTL $_{\pm 1}$). We assume that the states of the input GTL model are ordered. We can use any arbitrary order, so we simply need to assume a particular one, for instance: s_0, s_1, \dots, s_m . Assuming this order, we use the notation: $s \sim_{i \rightarrow j} (s_1, \dots, s_m)$ to denote $s \sim_{i \rightarrow j} s_1, \dots, s \sim_{i \rightarrow j} s_m$. $s \sim_{G \rightarrow j}^E (s_1, \dots, s_m)$, $s \sim_{G \rightarrow j}^D (s_1, \dots, s_m)$, and $s \sim_{i \rightarrow j, \mathcal{G}}^P (s_1, \dots, s_m)$ are defined in the same way.

The model transformation f_1 is as follows: $S_\pm = S_G \cup \{s_{new}\}$; $I_\pm = I_G$; Initialize AC_\pm with $\{\alpha_0\}$ and 1) $\forall s$ s.t. $\exists s'$ where $s \sim_{i \rightarrow j} s'$ add α_s^{ij} to AC_\pm , 2) $\forall s$ s.t. $\exists s'$ where $s \sim_{G \rightarrow j}^E s'$ add β_s^{Gj} to AC_\pm , 3) $\forall s$ s.t. $\exists s'$ where $s \sim_{G \rightarrow j}^D s'$ add γ_s^{Gj} to AC_\pm , and 4) $\forall s$ s.t. $\exists s'$ where $s \sim_{i \rightarrow j, \mathcal{G}}^P s'$ add $\delta_s^{ij\mathcal{G}}$ to AC_\pm ; $\forall s \in S_G, V_\pm(s) = V_G(s)$; $V_\pm(s_{new}) = \{\lambda\}$; $\forall (s, s') \in R_G$ add (s, α_s^{ij}, s') to R_\pm ; and if $s \sim_{i \rightarrow j} (s_1, s_2, \dots, s_m)$ then add $\{(s, \alpha_s^{ij}, s_1), (s_1, \alpha_s^{ij}, s_2), \dots, (s_{m-1}, \alpha_s^{ij}, s_m), (s_m, \alpha_s^{ij}, s_{new})\}$ to R_\pm . The same procedure applies if $s \sim_{G \rightarrow j}^E (s_1, s_2, \dots, s_m)$, $s \sim_{G \rightarrow j}^D (s_1, s_2, \dots, s_m)$, and $s \sim_{i \rightarrow j, \mathcal{G}}^P (s_1, s_2, \dots, s_m)$ where transitions labeled by β_s^{Gj} , γ_s^{Gj} , and $\delta_s^{ij\mathcal{G}}$ are added to R_\pm respectively. Figure 1 illustrates an example of model transformation f_1 .

The formula transformation f_1 is defined recursively:

- $f_1(M_G, s) \models f_1(\rho)$ iff $(M_\pm, s) \models \rho$;
- $f_1(M_G, s) \models f_1(\neg\varphi)$ iff $(M_\pm, s) \models \neg f_1(\varphi)$;
- $f_1(M_G, s) \models f_1(\varphi_1 \vee \varphi_2)$ iff $(M_\pm, s) \models f_1(\varphi_1) \vee f_1(\varphi_2)$;
- $f_1(M_G, s) \models f_1(EX\varphi)$ iff $(M_\pm, s) \models E_{\alpha_0} X f_1(\varphi)$;
- $f_1(M_G, s) \models f_1(E(\varphi_1 U \varphi_2))$ iff $(M_\pm, s) \models E_{\alpha_0} (f_1(\varphi_1) U f_1(\varphi_2))$;
- $f_1(M_G, s) \models f_1(A(\varphi_1 U \varphi_2))$ iff $(M_\pm, s) \models A_{\alpha_0} (f_1(\varphi_1) U f_1(\varphi_2))$;
- $f_1(M_G, s) \models f_1(I^{\Delta k}(i, j, \varphi))$ iff $(M_\pm, s) \models E_{\alpha_s^{ij}} X E_{\alpha_s^{ij} F} \frac{\#f_1(\varphi)}{[\frac{\tau}{\tau} \Delta k]} \lambda$;
- $f_1(M_G, s) \models f_1(E^{\Delta k}(G, j, \varphi))$ iff $(M_\pm, s) \models E_{\beta_s^{Gj}} X E_{\beta_s^{Gj} F} \frac{\#f_1(\varphi)}{[\frac{\tau}{\tau} \Delta k]} \lambda$;
- $f_1(M_G, s) \models f_1(D^{\Delta k}(G, j, \varphi))$ iff $(M_\pm, s) \models E_{\gamma_s^{Gj}} X E_{\gamma_s^{Gj} F} \frac{\#f_1(\varphi)}{[\frac{\tau}{\tau} \Delta k]} \lambda$;

- $f_1(M_G, s) \models f_1(P^{\Delta k}(i, j, \mathcal{G}, \varphi))$ iff $(M_\pm, s) \models E_{\delta_s^{ij\mathcal{G}}} X E_{\delta_s^{ij\mathcal{G}} F} \frac{\#f_1(\varphi)}{[\frac{\tau}{\tau} \Delta k]} \lambda$

All the cases are straightforward, except the trust operators $I^{\Delta k}$, $E^{\Delta k}$, $D^{\Delta k}$, and $P^{\Delta k}$ that make use of the semantics and exploit the constrained until operator of ARCCCTL $_{\pm 1}$ from the next state. This operator counts the number of states satisfying the content $f_1(\varphi)$ over the total number of states in the prefix starting from the next state to the last state preceding the state satisfying λ , which means the newly added state. This prefix corresponds to the accessible states from s , which are captured through the transitions labeled by α_s^{ij} for the operator $I^{\Delta k}$, β_s^{Gj} for the operator $E^{\Delta k}$, γ_s^{Gj} for the operator $D^{\Delta k}$, and $\delta_s^{ij\mathcal{G}}$ for the operator $P^{\Delta k}$.

THEOREM 1. $(M_G, s) \models \varphi$ iff $(M_\pm, s) \models f_1(\varphi)$

PROOF. We prove this part of the theorem by induction on the structure of the formula φ .

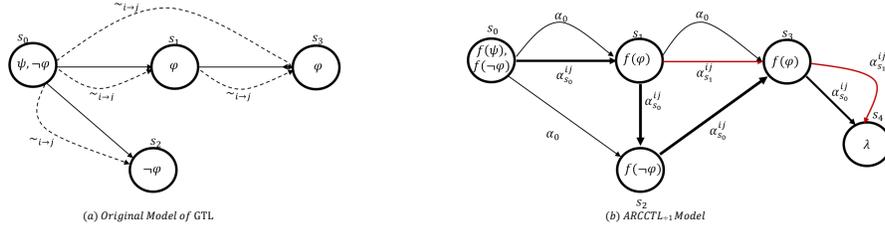
- The three first cases (atomic variables, negation and disjunction) are straightforward.

- For the formula $\varphi = EX\phi$, we have $(M_G, s) \models EX\phi$ iff there exists a path through which the immediate successor of s satisfies ϕ . From the definition of $f_1(M_G)$, all transitions obtained from M_G are labeled by α_0 in M_\pm . Consequently, using the semantics of $E_{\alpha_0} X f_1(\phi)$, we obtain $(M_\pm, s) \models E_{\alpha_0} X f_1(\phi)$.

- For the formula $\varphi = E(\phi_1 U \phi_2)$, the path through which φ holds in M_G corresponds to the same path on M_\pm labeled by α_0 . Thus, using the semantics of $E_{\alpha_0} (\phi_1 U \phi_2)$, we obtain $(M_\pm, s) \models E_{\alpha_0} (f_1(\phi_1) U f_1(\phi_2))$. The formula with the universal quantifier has the same proof.

- For the formula $\varphi = I^{\Delta k}(i, j, \phi)$, using the definition of $f_1(M_G)$, the existence of an accessible state different from the current state is captured by $E_{\alpha_s^{ij}} X$ since one of the different accessible states in M_G (precisely the first one according to the adopted ordering of accessible states) is linked to the current state by a transition labeled by α_s^{ij} in M_\pm . To capture the constraint condition in M_G , we need to have a path in M_\pm that contains all the accessible states through which we can compute the number of states satisfying ϕ . As accessible states are ordered and captured through a labeled path by α_s^{ij} and reaching the new added state labeled by λ in M_\pm , the path is captured by $E_{\alpha_s^{ij}} F \lambda$. The computation of the number of states satisfying ϕ should only consider accessible states, which corresponds to the states of the path flat prefix starting from the next state to the last state preceding the newly added state. Consequently, using the semantics of $E_{\alpha_s^{ij}} F \frac{\#f_1(\phi)}{[\frac{\tau}{\tau} \Delta k]} \lambda$, we obtain $(M_\pm, s) \models E_{\alpha_s^{ij}} X E_{\alpha_s^{ij} F} \frac{\#f_1(\phi)}{[\frac{\tau}{\tau} \Delta k]} \lambda$.

- The formulae $\varphi = E^{\Delta k}(G, j, \varphi)$, $\varphi = D^{\Delta k}(G, j, \varphi)$, and $\varphi = P^{\Delta k}(i, j, \mathcal{G}, \varphi)$ have the same proof where only the appropriate paths with their labels should be identified properly. first condition and the formula to be counted for the constraint should be updated accordingly. \square

Figure 1: f_1 model transformation from GTL to $\text{ARCCTL}_{\div 1}$

4.3 Stage 2: Model Checking MC_{\div}

For the second stage, the Function MC_{\div} operates on the $\text{ARCCTL}_{\div 1}$ formula Φ and model M_{\div} to produce the set of states satisfying Φ , i.e., $[[\Phi]]$. The function is recursive and makes use of MC_A , the model checking algorithm of ARCTL. The calls to MC_A operate on ARCTL formulae and M_{\div} . This is because $\text{ARCCTL}_{\div 1}$ models are also ARCTL models (there is a simple mapping of each element of M_{\div} to the corresponding element of M_A). For the cases of $E_{\alpha}X\varphi$ and $E_{\alpha}(\varphi_1 U_{\lfloor \frac{\# \varphi}{\tau} \Delta k \rfloor} \varphi_2)$, before calling MC_A , new fresh atomic propo-

sitions are added to the states satisfying the $\text{ARCCTL}_{\div 1}$ subformulae. Thus, for the case $E_{\alpha}X\varphi$, the Function MC_{EX} calls the model checking of ARCTL MC_A for the ARCTL formula $E_{\alpha}X\chi$ where χ is a fresh atomic proposition that holds in exactly the same states where the $\text{ARCCTL}_{\div 1}$ subformula φ holds. The formula $F1 = E_{\alpha}(\varphi_1 U_{\lfloor \frac{\# \varphi}{\tau} \Delta k \rfloor} \varphi_2)$ is model checked in the Function MC_{EU} in

the same way to compute the set A of states satisfying the ARCTL subformula $E_{\alpha}(\chi_1 U \chi_2)$ where χ_1 and χ_2 are fresh atomic propositions holding exactly in the same states where the subformulae φ_1 and φ_2 hold respectively. After building, from the set A , the finite set \mathcal{F} of flat prefixes that satisfy $\chi_1 U \chi_2$, the set of states satisfying the constraint's formula φ is computed by $MC_A(\chi, M_{\div})$. The set $Y = [[F1]]$ is updated by adding the first state of every flat prefix where the number of states satisfying φ over the size of the prefix -1 is Δk . The dual Function MC_{AU} is similar with the following changes: line 9 becomes $Y := A$; and line 14 becomes: **if** $n = 0$ **or** $\frac{|S_{\varphi}^U|}{n} \nless \Delta k$ **then** $Y := Y - \{s_0\}$. This means the first state of the flat prefix is removed from the set Y if the constraint's equation is not satisfied or if the prefix has only one state.

THEOREM 2. $(M_{\div}, s) \models \varphi$ iff $s \in MC_{\div}(\varphi, M_{\div})$

PROOF. We prove this theorem by induction over the structure of the formula φ using the soundness result of ARCTL, i.e., $(M_A, s) \models \varphi_A$ iff $s \in MC_A(\varphi_A, M_A)$ and the fact that M_A and M_C are equal.

- The atomic case is direct from the soundness of MC_A .

- For the negation case ($\varphi = \neg\phi$), the result is direct from the fact that the set of states that satisfy $\neg\phi$ is the complement of the set of states that satisfy ϕ .

- The disjunction case ($\varphi = \phi_1 \vee \phi_2$) is direct from the fact that the set of states satisfying $\phi_1 \vee \phi_2$ is the union of the sets of states satisfying ϕ_1 and ϕ_2 .

Function $MC_{\div}(\Phi: \text{ARCCTL}_{\div 1}, M_{\div}) : [[\Phi]]$

- 1: **switch** Φ :
 - 2: **case** ρ : **return** $MC_A(\rho, M_{\div})$;
 - 3: **case** $\neg\varphi$: **return** $S_{\div} - MC_{\div}(\varphi, M_{\div})$;
 - 4: **case** $\varphi_1 \vee \varphi_2$: **return** $MC_{\div}(\varphi_1, M_{\div}) \cup MC_{\div}(\varphi_2, M_{\div})$;
 - 5: **case** $E_{\alpha}X\varphi$: **return** $MC_{EX}(\alpha, \varphi, M_{\div})$;
 - 6: **case** $E_{\alpha}(\varphi_1 U_{\lfloor \frac{\# \varphi}{\tau} \Delta k \rfloor} \varphi_2)$: **return** $MC_{EU}(\alpha, \varphi_1, \varphi_2, \Delta k, M_{\div})$;
 - 7: **case** $A_{\alpha}(\varphi_1 U_{\lfloor \frac{\# \varphi}{\tau} \Delta k \rfloor} \varphi_2)$: **return** $MC_{AU}(\alpha, \varphi_1, \varphi_2, \Delta k, M_{\div})$;
 - 8: **end switch**
-

Function $MC_{EX}(\alpha, \varphi, M_{\div}) : [[E_{\alpha}X\varphi]]$

- 1: $X := MC_{\div}(\varphi, M_{\div})$;
 - 2: $\forall s \in X, V_{\div}(s) := V_{\div}(s) \cup \{\chi\}$;
 - 3: **return** $MC_A(E_{\alpha}X\chi, M_{\div})$
-

- For the case $\varphi = E_{\alpha}X\phi$, since the states satisfying ϕ are the same as the states satisfying χ , we have $s \in MC_{\div}(E_{\alpha}X\phi, M_{\div})$ iff $s \in MC_A(E_{\alpha}X\chi, M_{\div})$, so the result.

- For the formula $\varphi = E_{\alpha}(\phi_1 U_{\lfloor \frac{\# \phi}{\tau} \Delta k \rfloor} \phi_2)$, the states that satisfy

ϕ_1 and ϕ_2 are the same as the states satisfying χ_1 and χ_2 , which yields the set A contains exactly the states satisfying $E_{\alpha}(\phi_1 U \phi_2)$. Also, the set S_{φ} contains the states satisfying ϕ since the states satisfying χ are the same as the states satisfying ϕ . According to the semantics of the formula φ , s satisfies φ iff it satisfies $E_{\alpha}(\phi_1 U \phi_2)$, so it is in A , and the number of states satisfying ϕ , so in S_{φ} , and belong to the flat prefix $s_0 \dots s_n$ starting at $s = s_0$ and ending at s_n that satisfies ϕ_2 meets the condition $\frac{\# \phi}{\tau} \Delta k$, which means $\frac{|S_{\varphi}^U|}{n} \Delta k$ as S_{φ}^U contains the states in the flat prefix $s_0 \dots s_{n-1}$ that satisfy ϕ and $n = |s_0 \dots s_{n-1}|$. Thus, $s \in Y$ iff s satisfies φ , so the result.

- The proof for the formula $\varphi = A_{\alpha}(\phi_1 U_{\lfloor \frac{\# \phi}{\tau} \Delta k \rfloor} \phi_2)$ is similar. □

In the following, the subscripts C , A , \div and G are used for the models and formulae of CTL, ARCTL, $\text{ARCCTL}_{\div 1}$, and GTL resp.

5 COMPLEXITY ANALYSIS

In this section, we will show that the complexity of model checking GTL is PSPACE-complete for concurrent programs. Concurrent

Function $MC_{EU}(\alpha, \varphi_1, \varphi_2, \varphi, \Delta k, M_{\pm}) : [[F1]]$

```

1:  $X_1 := MC_{\pm}(\varphi_1, M_{\pm});$ 
2:  $X_2 := MC_{\pm}(\varphi_2, M_{\pm});$ 
3:  $X := MC_{\pm}(\varphi, M_{\pm});$ 
4:  $\forall s \in X_1, V_{\pm}(s) := V_{\pm}(s) \cup \{\chi_1\};$ 
5:  $\forall s \in X_2, V_{\pm}(s) := V_{\pm}(s) \cup \{\chi_2\};$ 
6:  $\forall s \in X, V_{\pm}(s) := V_{\pm}(s) \cup \{\chi\};$ 
7:  $A := MC_A(E_{\alpha}(\chi_1 U \chi_2), M_{\pm});$ 
8: if  $A = \emptyset$  then return  $\emptyset$ ;
9:  $Y := \emptyset$ ;
10: Build the finite set  $\mathcal{F}$  of flat prefixes  $s_0 \dots s_n$  s.t.  $\forall 0 \leq i \leq n-1, s_i \in A \& (M_{\pm}, s_i) \models \chi_1$  and  $s_n \in A \& (M_{\pm}, s_n) \models \chi_2$ ;
11:  $S_{\varphi} := MC_A(\chi, M_{\pm});$ 
12: for each flat prefix  $s_0 \dots s_n \in \mathcal{F}$ 
13:    $S_{\varphi}^U = S_{\varphi} \cap \{s_0, \dots, s_{n-1}\};$ 
14:   if  $n \neq 0$  and  $\frac{|S_{\varphi}^U|}{n} \Delta k$  then  $Y := Y \cup \{s_0\};$ 
15: end for
16: return  $Y$ 

```

programs [23] are the natural framework for MASs as they are composed of n concurrent agents, where each agent is described by a transition system where transitions are labeled by actions. A concurrent behavior of these agents is obtained by the product of the n transition systems where transition actions that appear in several agents are synchronized by common actions. In these structures, states and transitions are not listed explicitly, but having instead compact representations that still correspond to the actual system. In fact, in symbolic model checking, “the Kripke structures to which model checking is applied are often obtained by constructing the reachability graph of concurrent programs” [23]. We first start by Theorems 3, 4, and 5 that give space complexity in explicit models.

THEOREM 3. *The model checking problem of ARCTL for explicit models can be solved in space $O(|M_A| \times \log |\varphi_A|)$.*

PROOF. It is known from [37] that the model checking problem of CTL can be solved in space $O(|M_C| \times \log |\varphi_C|)$. The model checking problem of ARCTL can be solved by a transformation to the model checking of CTL. For the model, we only need to add, for each transition in the M_A model, an intermediate state to the CTL model labeled with a fresh atomic proposition representing the atomic action labeling the transition in the ARCTL model. Each transition in the input M_A model will need one additional state and one additional transition in the output M_C model. Thus, the size of the obtained M_C is linear with the size of the input M_A . For the formula, the transformation function f is defined recursively. The case of atomic propositions is direct. $\varphi \vee \psi$ will be transformed to $f(\varphi) \vee f(\psi)$. $E_{\alpha}X\varphi$ will be transformed to $EX(\alpha_p \wedge EXf(\varphi))$ and $E_{\alpha}/A_{\alpha}(\varphi U \psi)$ will be transformed to $E/A((f(\varphi) \vee \alpha_p) U f(\psi))$ where α_p is the fresh atomic proposition representing the atomic action α (the proof of the soundness of this transformation is provided after this proof). The size of the output formula is then linear with the size of the input one. Since the length of the recursion is bounded by the size of the formula and the computation of the transformation is clearly logarithmic, we are done. \square

Soundness of the Transformation in the Proof of Theorem 3. Let M_A and M_C be the ARCTL and CTL models respectively, and $f(\varphi)$ be the CTL formula obtained from the transformation of the ARCTL formula φ . We have: $(M_A, s) \models \varphi$ iff $(M_C, s) \models f(\varphi)$.

PROOF. The proof is by induction on the structure of φ .

- The three first cases (atomic variables, negation and disjunction) are straightforward since the corresponding states in the two models are the same.

- For the formula $E_{\alpha}X\varphi$, according to the semantics, φ holds in the next state through an α -path of M_A . According to the transformation of the model, α_p holds in the next (newly added) state of M_C through the corresponding path, and $f(\varphi)$ holds in the next state of this newly added state. Thus, we obtain $(M_C, s) \models EX(\alpha_p \wedge EXf(\varphi))$.

- For the formula $E_{\alpha}(\varphi U \psi)$, according to the semantics, φ holds in all the states through an α -path of M_A until the state where ψ holds. According to the model transformation, this path is transformed to a path in M_C where between each two successive states, a new state labeled by α_p is added. Thus, each state through this path in M_C will satisfy either $f(\varphi)$ if it is a state from M_A or α_p if it is an added state, e.i., $f(\varphi) \vee \alpha_p$ until reaching the state that satisfies $f(\psi)$. Consequently, $(M_C, s) \models E((f(\varphi) \vee \alpha_p) U f(\psi))$.

- For the formula $A_{\alpha}(\varphi U \psi)$, which is transformed to $A((f(\varphi) \vee \alpha_p) U f(\psi))$, the proof is similar. \square

THEOREM 4. *The explicit model checking problem of $ARCCTL_{\pm 1}$ can be solved in space $O(|M_{\pm}| \times \log |\varphi_{\pm}|)$.*

PROOF. From Theorem 2, model checking $ARCCTL_{\pm 1}$ is solved using the procedure MC_{\pm} that calls the model checking of ARCTL. MC_{\pm} is recursive and the recursion’s depth is bounded by the size of the formula. From Theorem 3, the first four cases are straightforward. For the sixth and seventh cases, the Functions MC_{EU} and MC_{AU} build and check the flat prefixes one by one without storing the whole set. The result follows then from Theorem 3. \square

THEOREM 5. *The model checking problem of GTL for explicit models can be solved in space $O(|M_G| \times \log |\varphi_G|)$.*

PROOF. Model checking GTL can be solved by calling the model checking algorithm of $ARCCTL_{\pm 1}$ (Theorem 1). The size of the obtained M_{\pm} model from the input model M_G using f_1 is clearly linear with the original M_G model. For the formula, we show that the size of the input GTL formula Φ is linear in the length of the output $ARCCTL_{\pm 1}$ formula $f_1(\Phi)$. We prove this by induction over the structure of Φ . The proposition holds for the atomic case, and we have: $|f_1(\neg\varphi)| = 1 + |f_1(\varphi)|$; $|f_1(\varphi_1 \vee \varphi_2)| = 1 + |f_1(\varphi_1)| + |f_1(\varphi_2)|$; $|f_1(EX\varphi)| = 2 + |f_1(\varphi)|$ (note that $|\alpha_0| = 1$); $|f_1(E(\varphi_1 U \varphi_2))| = 2 + |f_1(\varphi_1)| + |f_1(\varphi_2)|$; $|f_1(A(\varphi_1 U \varphi_2))| = 2 + |f_1(\varphi_1)| + |f_1(\varphi_2)|$; $|f_1(I^{\Delta k}(i, j, \varphi))| = 5 + |f_1(\varphi)|$; $|f_1(E^{\Delta k}(i, j, \varphi))| = 5 + |f_1(\varphi)|$; $|f_1(D^{\Delta k}(i, j, \varphi))| = 5 + |f_1(\varphi)|$; $|f_1(P^{\Delta k}(i, j, \varphi))| = 5 + |f_1(\varphi)|$. Thus, if the proposition holds for φ, φ_1 , and φ_2 , then it holds for $f_1(\neg\varphi), f_1(\varphi_1 \vee \varphi_2), f_1(EX\varphi), f_1(E(\varphi_1 U \varphi_2)), f_1(A(\varphi_1 U \varphi_2)), f_1(I^{\Delta k}(i, j, \varphi)), f_1(E^{\Delta k}(i, j, \varphi)), f_1(D^{\Delta k}(i, j, \varphi)),$ and $f_1(P^{\Delta k}(i, j, \varphi))$. The transformation needs logarithmic space, so the result follows from Theorem 4. \square

Let $MC(\mathcal{L})$ be the problem of model checking the language \mathcal{L} for concurrent programs.

THEOREM 6. $MC(\text{ARCCTL}_{\div 1})$ is PSPACE-complete.

PROOF. Hardness is immediate by a reduction from $MC(\text{CTL})$, proved to be PSPACE-complete [23]. Membership is direct from Function MC_{\div} since $MC(\text{ARCTL})$ is PSPACE-complete [22]. \square

THEOREM 7. The program complexity of $MC(\text{ARCCTL}_{\div 1})$ is PSPACE-complete.

PROOF. PSPACE-hardness is direct from the PSPACE-completeness of the program complexity of $MC(\text{CTL})$ [23]. The result follows then from Theorem 6 as the complexity of $MC(\text{ARCCTL}_{\div 1})$ is greater than or equal to its program complexity. \square

PROPOSITION 1. Let $\text{Mod}(\mathcal{L})$ be the model of the language \mathcal{L} and \leq_{\log} denote the log-space reduction. We have $\text{Mod}(\text{GTL}) \leq_{\log} \text{Mod}(\text{ARCCTL}_{\div 1})$.

PROOF. We show that the model reduction from GTL to $\text{ARCCTL}_{\div 1}$ presented in Section 4.2 can be computed by a deterministic Turing machine TM in space $O(\log(|\text{Mod}(\text{GTL})|))$. TM reads in the input tape a model of GTL and generates in the output tape, one by one, the same states with the same state ordering, the same state labeling function, the same transitions as the input after labeling them with α_0 and writing α_0 in the set of atomic actions AC_A , and an additional state s_{new} to which it associates the last ordering rank with a unique label λ . For each state s and each pair of agents (i, j) , TM reads the accessibility relations $\sim_{i \rightarrow j}$ from this state one by one in a sequential way in the same order of the accessible states, adds α_s^{ij} to the set AC_A and adds transitions labeled by α_s^{ij} , first from s to the first accessible state, then between each two adjacent accessible states according to their order, and finally from the last accessible state to the newly added state s_{new} . The other accessibility relations follow the same procedure. Thus, to transform the accessibility relations, we only need to record 3 states at each time: the original state s , current accessible state, and last accessible state to which a transition labeled by α_s^{ij} has been added. All these operations can be done in a logarithmic space in the size of the input model, so we are done. \square

THEOREM 8. $MC(\text{GTL})$ is PSPACE-complete.

PROOF. PSPACE-hardness is direct from the PSPACE-completeness of $MC(\text{CTL})$. The PSPACE upper bound for the formula follows from Theorem 5. Finally, since the program complexity of $MC(\text{GTL})$ is greater than or equal to the program complexity of $MC(\text{CTL})$, the result follows from Proposition 1 and Theorem 7. \square

6 RELATED WORK AND CONCLUSION

Modeling and model checking quantitative trust within a group have not been investigated yet. Indeed, there are relatively a few proposals directly related to our work. For instance, in [10], the authors proposed a logical approach for the concept of graded trust. They developed a logic by combining dynamic logic [18] with a BDI-like logic [8]. In a follow-up work, [9] defined a logical framework to represent graded trust in terms of two independent components: graded beliefs and graded regularities. Trust is reduced to graded

beliefs, so the graded trust is defined as the strength level of the trustor agent belief about the trustee agent sincerity. In another proposal, [26] have focused on analyzing the trust that can be associated with information sources. The authors have integrated graded beliefs into a logical framework that defines different kinds of trust. In [27], the quantitative aspects of trust has been considered in a dynamic epistemic logic setting, where the relationship between trust and belief change is presented using Dynamic Logic of graded Belief and Trust (DL-BT). The graded trust operator in the proposed logic is interpreted using a neighborhood semantics [31], whose model checking is yet to be developed. In [28], a logic of graded beliefs with a formal semantics grounded on the notion of belief base is presented. The authors applied the proposed language to the analysis of the concept of social influence. Moreover, a sound and complete axiomatics and a model checking algorithm are provided. However, these approaches do not deal with trust as a first-class citizen, but focus on agents with mental states where the trusting entity is normally capable of exhibiting private beliefs, desires, and intentions. Furthermore, only individual trust is addressed and no quantification over groups is considered. In [12], TCTL^G that formally presents the quantitative aspect of trust in MASs is introduced along with a preliminary model checking procedure. However, only individual trust has been considered and no complexity has been explored. In terms of reasoning about trust in groups, [11] presented a language named BT that allows reasoning about individual and group trust. The authors analyzed the satisfiability and model checking problems of this logic. However, the approach focuses solely on qualitative reasoning.

Graded modalities have been added to the semantics of CTL by many authors. For instance, in [4], the authors introduced GCTL, the extension of CTL with graded modalities on paths. GCTL counts the number of equivalence classes of paths satisfying a given formula. The authors proved that the satisfiability problem of GCTL is solvable in EXPTIME. In [5], a trust quantification framework based on Subjective Logic has been proposed. The framework interprets the behaviors of agents and assigns them a trustworthiness score. Moreover, [3] introduced a novel semantics for Strategy Logic with Knowledge in which one can specify, for each agent, the set of agents whose strategy she is informed of. The complexity of the model checking is also presented. Other proposals considered a graded logical formalism for strategic reasoning in MASs. In [29] and [2], two different graded versions of Strategy Logic (SL) are defined. For instance, the authors in [29] introduced Graded Strategy Logic (GSL), an extension of SL by graded strategy quantifiers. In [16], the authors enriched the Alternating-time Temporal Logic (ATL) with a graded modality in order to determine the value of the maximum grade for which the particular formula is true. These proposals do not consider the trust concept and their quantification approaches are different from our grading approach based on the number of accessible states instead of runs or strategies.

We introduced GTL to represent quantitative individual and group trust and presented a novel technique for its model checking. We showed that the complexity of this model checking for concurrent programs is PSPACE-complete. For future work, we plan to integrate the concept of ability to bring about something à la [15] and investigate trust and commitments [39] in the Strategy Logic.

REFERENCES

- [1] Alessandro Aldini, Gianluca Curzi, Pierluigi Graziani, and Mirko Tagliaferri. 2021. Trust Evidence Logic. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 16th European Conference, ECSQARU (Lecture Notes in Computer Science, Vol. 12897)*, Jirina Vejnárová and Nic Wilson (Eds.). Springer, 575–589.
- [2] Benjamin Aminof, Aniello Murano, and Sasha Rubin. 2018. CTL* with graded path modalities. *Inf. Comput.* 262 (2018), 1–21.
- [3] Francesco Belardinelli, Sophia Knight, Alessio Lomuscio, Bastien Maubert, Aniello Murano, and Sasha Rubin. 2021. Reasoning About Agents That May Know Other Agents' Strategies. In *IJCAI*. 1787–1793.
- [4] Alessandro Bianco, Fabio Mogavero, and Aniello Murano. 2012. Graded computation tree logic. *ACM Trans. Comput. Log.* 13, 3 (2012), 25:1–25:53.
- [5] Mingxi Cheng, Chenzhong Yin, Junyao Zhang, Shahin Nazarian, Jyotirmoy Deshmukh, and Paul Bogdan. 2021. A General Trust Framework for Multi-Agent Systems. In *AAMAS*. 332–340.
- [6] Kinzang Chhogyal, Abhaya C. Nayak, Aditya Ghose, and Hoa Khanh Dam. 2019. A Value-based Trust Assessment Model for Multi-agent Systems. In *IJCAI*. 194–200.
- [7] Edmund M Clarke, Orna Grumberg, and Doron Peled. 1999. *Model checking*. MIT press.
- [8] Philip R. Cohen and Hector J. Levesque. 1990. Intention is Choice with Commitment. *Artif. Intell.* 42, 2-3 (1990), 213–261.
- [9] Robert Demolombe. 2009. Graded trust. In *Workshop on Trust in Agent Societies*. 1–12.
- [10] Robert Demolombe and Churn-Jung Liau. 2001. A logic of graded trust and belief fusion. In *Workshop on Deception, Fraud and Trust in Agent Societies*. 13–25.
- [11] Nagat Drawel, Jamal Bentahar, Amine Laarej, and Gaïth Rjoub. 2020. Formalizing group and propagated trust in multi-agent systems. In *IJCAI*. 60–66.
- [12] Nagat Drawel, Jamal Bentahar, and Hongyang Qu. 2020. Computationally grounded quantitative trust with time. In *AAMAS*. 1837–1839.
- [13] Nagat Drawel, Hongyang Qu, Jamal Bentahar, and Elhadi M. Shakshuki. 2020. Specification and automatic verification of trust-based multi-agent systems. *Future Gener. Comput. Syst.* 107 (2020), 1047–1060.
- [14] Didier Dubois, Jérôme Lang, and Henri Prade. 1994. Possibilistic logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, D. M. Gabbay, C. J. Hogger, J. A. Robinson, and D. Nute (Eds.). Vol. 3. 439–513.
- [15] Dag Elgesem. 1997. The modal logic of agency. *Nordic Journal of Philosophical Logic* (1997).
- [16] Marco Faella, Margherita Napoli, and Mimmo Parente. 2010. Graded Alternating-Time Temporal Logic. *Fundam. Informaticae* 105, 1-2 (2010), 189–210.
- [17] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. 1995. *Reasoning about Knowledge*. MIT Press.
- [18] David Harel, Dexter Kozen, and Jerzy Tiuryn. 2000. *Dynamic Logic*. MIT press.
- [19] Andreas Herzig, Emiliano Lorini, and Frédéric Moisan. 2012. A simple logic of trust based on propositional assignments. In *The Goals of Cognition. Essays in Honour of Cristiano Castelfranchi*. College Publications, 407–419.
- [20] Xiaowei Huang and Marta Zofia Kwiatkowska. 2017. Reasoning about Cognitive Trust in Stochastic Multiagent Systems. In *AAAI*. 3768–3774.
- [21] Audun Jøsang. 2016. *Subjective logic: A Formalism for Reasoning Under Uncertainty*. Springer.
- [22] Warda El Kholy, Jamal Bentahar, Mohamed El-Menshawey, Hongyang Qu, and Rachida Dssouli. 2017. SMC4AC: A New Symbolic Model Checker for Intelligent Agent Communication. *Fundam. Informaticae* 152, 3 (2017), 223–271.
- [23] Orna Kupferman, Moshe Y. Vardi, and Pierre Wolper. 2000. An automata-theoretic approach to branching-time model checking. *J. ACM* 47, 2 (2000), 312–360.
- [24] François Laroussinie, Antoine Meyer, and Eudes Pettonnet. 2010. Counting CTL. In *International Conference on Foundations of Software Science and Computational Structures*. 206–220.
- [25] Christopher Leturc and Grégory Bonnet. 2018. A Normal Modal Logic for Trust in the Sincerity. In *AAMAS*. 175–183.
- [26] Emiliano Lorini and Robert Demolombe. 2008. From binary trust to graded trust in information sources: a logical perspective. In *Workshop on Trust in Agent Societies*. 205–225.
- [27] Emiliano Lorini, Guifei Jiang, and Laurent Perrussel. 2014. Trust-based belief change. In *ECAI*. 549–554.
- [28] Emiliano Lorini and François Schwarzentruher. 2021. A Computationally Grounded Logic of Graded Belief. In *Logics in Artificial Intelligence - 17th European Conference, JELIA (Lecture Notes in Computer Science, Vol. 12678)*, Wolfgang Faber, Gerhard Friedrich, Martin Gebser, and Michael Morak (Eds.). Springer, 245–261.
- [29] Vadim Malvone, Fabio Mogavero, Aniello Murano, and Loredana Sorrentino. 2018. Reasoning about graded strategy quantifiers. *Inf. Comput.* 259, 3 (2018), 390–411.
- [30] Karsten Martiny and Ralf Möller. 2016. PDT Logic: A Probabilistic Doxastic Temporal Logic for Reasoning about Beliefs in Multi-agent Systems. *J. Artif. Intell. Res.* 57 (2016), 39–112.
- [31] Richard Montague. 1970. Universal grammar. *Theoria* 36, 3 (1970), 373–398.
- [32] Nardine Osman and David Robertson. 2007. Dynamic verification of trust in distributed open systems. In *IJCAI*. 1440–1445.
- [33] Simon Parsons, Yuqing Tang, Elizabeth Sklar, Peter McBurney, and Kai Cai. 2011. Argumentation-based reasoning in agents with varying degrees of trust. In *AAMAS*. 879–886.
- [34] David Pearce and Levan Uridia. 2015. Trust, Belief and Honesty. In *GCAI (EPIc Series in Computing, Vol. 36)*, Georg Gottlob, Geoff Sutcliffe, and Andrei Voronkov (Eds.). 215–228.
- [35] Charles Pecheur and Franco Raimondi. 2006. Symbolic Model Checking of Logics with Actions. In *Workshop on Model Checking and Artificial Intelligence*. 113–128.
- [36] Noel Sardana, Robin Cohen, Jie Zhang, and Shuo Chen. 2018. A Bayesian Multiagent Trust Model for Social Networks. *IEEE Trans. Comput. Soc. Syst.* 5, 4 (2018), 995–1008.
- [37] Philippe Schnoebelen. 2002. The Complexity of Temporal Logic Model Checking. In *Advances in Modal Logic 4, papers from the fourth conference on "Advances in Modal Logic," held in Toulouse, France, 30 September - 2 October*, Philippe Balbiani, Nobu-Yuki Suzuki, Frank Wolter, and Michael Zakharyashev (Eds.). King's College Publications, 393–436.
- [38] Munindar P Singh. 2011. Trust as dependence: a logical approach. In *AAMAS*. 863–870.
- [39] Pankaj R. Telang, Munindar P. Singh, and Neil Yorke-Smith. 2021. Maintenance of Social Commitments in Multiagent Systems. In *AAAI*. 11369–11377.
- [40] Leonid Zeynalvand, Tie Luo, Ewa Andrejczuk, Dusit Niyato, Sin G. Teo, and Jie Zhang. 2021. A Blockchain-Enabled Quantitative Approach to Trust and Reputation Management with Sparse Evidence. In *AAMAS*. 1707–1708.