Studies in Logic, Vol. 14, No. 3 (2021): 53–69
PII: 1674-3202(2021)-03-0053-17

Reasoning about the Dynamics of Self-organizing Multi-agent Systems*

Jieting Luo Beishui Liao✉ John-Jules Meyer

Abstract. Self-organization has been introduced to multi-agent systems as an internal control process or mechanism to solve difficult problems spontaneously. When the system is deployed in an open environment, the change of participating agents might bring the system to an undesired state. Therefore, it is important to know what properties remain true and what properties become false when we change the participating agents in the system. As it is computationally expensive to verify a self-organizing multi-agent system, we need to think about how we can properly use the verification result that we get from the original system to better verify the new system. In this paper, we propose a framework to reason about the dynamics of self-organizing multi-agent systems under the change of participating agents.

1 Introduction

Self-organization is a process where a stable pattern is formed by the cooperative behavior between parts of an initially disordered system without external control or influence. It has been introduced to multi-agent systems as an internal control process or mechanism to solve difficult problems spontaneously, especially if the system is operated in an open environment thereby having no perfect and a priori design to be guaranteed. ([8, 9, 10]) When the system is deployed in an open environment, agents can enter or exit the system. It might for example be that certain properties become true and certain properties become false, possibly bringing the system to an undesired state. Because in a self-organizing multi-agent system local rules work as guidance for agents to behave thus leading to specific outcomes, we can see the set of local rules as a mechanism that we implement in the system. A mechanism is a procedure, protocol or game for generating desired outcomes. If we want to know whether we can design a set of local rules to ensure desired outcomes, we then enter the field of

Received 2021-05-30

Jieting Luo Department of Philosophy, Zhejiang University
luojieting@zju.edu.cn

Beishui Liao✉ Department of Philosophy, Zhejiang University
baiseliao@zju.edu.cn

John-Jules Meyer Department of Information and Computing Sciences,Utrecht University
J.J.C.Meyer@uu.nl

*We thank all the reviewers for their valuable comments and the financial support from the China Postdoctoral Science Fund (General Program) (505000-X92001) and the Major Program of National Social Science Foundation of China (20&ZD047).

mechanism design. ([7]) In the area of model checking, some work has been done to verify a multi-agent system, where norms are regarded as a mechanism. M. Knobbout et al. assume agents to have some preferences, which might be unknown to the system designers, and use a solution concept of Nash equilibrium for decision-making. ([4]) A formal framework is developed to verify whether a normative system implements desired outcomes no matter what preferences agents have. N. Bulling and M. Dastani formally analyze and verify whether a specific behavior can be enforced by norms and sanctions if agents follow their subjective preferences and whether a set of norms and sanctions that realize the effect exists at all. ([2]) In the present paper, agents are supposed to communicate with each other for their internals, which might be unknown to the system designers when the system is operated in an open environment, and thus it is important to know how the system behaves under the change of agents' internals. Intuitively, we can verify both the original system and the new system afresh. However, as we have proved in [6], the verification is computationally expensive. Therefore, we need to think about how we can properly use the verification result that we get from the original system to better verify the new system. In this paper, we propose a framework to reason about the dynamics of self-organizing multi-agent systems under the change of participating agents. Agents follow their local rules to communicate with each other and perform actions, which results in the structural and semantic independence between agents that are represented by the notion of full contributions to the global system behavior. We prove that the full contribution of a coalition of agents remains the same if the internal function of any agent in the coalition is unchanged. Furthermore, we prove that the properties of a coalition regarding not being semantically independent or structurally independent inherit from the original system to the new system if the internal function of any agent in the coalition is unchanged, which means that we do not need to check their semantic independence and structural independence when verifying the new system. Such results can be used to improve the verification efficiency for the new system.

The remaining part of this paper is structured as follows: In Section 2, we introduce the framework we use in this paper, setting up the context of self-organizing multi-agent systems; In Section 3, we propose the way how we change participating agents and prove what verification information we can use from the original system to efficiently verify the new system; In Section 4, we provide an example to illustrate our results; In Section 5, we conclude this paper and provide future work.

2 Framework

We use the same framework to formalize self-organizing multi-agent systems as what we introduced in [6]. The semantic structure of this paper is concurrent game structures (CGSs). It is basically a model where agents can simultaneously choose actions that collectively bring the system from the current state to a successor state. Compared to other kripke models of transaction systems, each transition in a CGS is labeled with collective actions and the agents who perform those actions. Moreover, we treat actions as first-class entities instead of using choices that are identified by their possible outcomes. Formally,

Definition 1 A concurrent game structure is a tuple $\mathcal{S} = (k, Q, \pi, \Pi, ACT, d, \delta)$ such that:

- A natural number $k \geq 1$ of agents, and the set of all agents is $\Sigma = \{1, \ldots, k\}$; we use A to denote a coalition of agents $A \subseteq \Sigma$;
- A finite set Q of states;
- A finite set Π of propositions;
- A labeling function π which maps each state $q \in Q$ to a subset of propositions which are true at q; thus, for each $q \in Q$ we have $\pi(q) \subseteq \Pi$;
- A finite set ACT of actions;
- For each agent $i \in \Sigma$ and a state $q \in Q$, $d_i(q) \subseteq ACT$ is the non-empty set of actions available to agent i in q; $D(q) = d_1(q) \times \ldots \times d_k(q)$ is the set of joint actions in q; given a state $q \in Q$, an action vector is a tuple $\langle \alpha_1, \ldots, \alpha_k \rangle$ such that $\alpha_i \in d_i(q)$;
- A function δ which maps each state $q \in Q$ and a joint action $\langle \alpha_1, \ldots, \alpha_k \rangle \in D(q)$ to another state that results from state q if each agent adopted the action in the action vector, thus for each $q \in Q$ and each $\langle \alpha_1, \ldots, \alpha_k \rangle \in D(q)$ we have $\delta(q, \langle \alpha_1, \ldots, \alpha_k \rangle) \in Q$, and we use $(q, \langle \alpha_1, \ldots, \alpha_k \rangle, q')$ where $q' = \delta(q, \langle \alpha_1, \ldots, \alpha_k \rangle)$ to denote a transition that starts from q and is labeled with $\langle \alpha_1, \ldots, \alpha_k \rangle$.

Note that the model is deterministic: the same update function adopted in the same state will always result in the same resulting state. A computation over $\mathcal{S}$ is an infinite sequence $\lambda = q_0, q_1, q_2, \ldots$ of states such that for all positions $i \geq 0$, there is a joint action $\langle \alpha_1, \ldots, \alpha_k \rangle \in D(q_i)$ such that $\delta(q_i, \langle \alpha_1, \ldots, \alpha_k \rangle) = q_{i+1}$. For a computation λ and a position $i \geq 0$, we use $\lambda[i]$ to denote the ith state of λ. More elaboration of concurrent game structures can be found in [1].

The same as what we did in [6], we can define a self-organizing multi-agent system as a concurrent game structure together with a set of local rules for agents to follow. Before defining such a type of local rules, we first define what to communicate, which is given by an internal function.

Definition 2 (Internal Functions) Given a concurrent game structure $\mathcal{S}$ and an agent i, the internal function of agent i is a function $m_i : Q \to \mathcal{L}_{prop}$ that maps a state $q \in Q$ to a propositional formula. We use a tuple $\mathcal{M} = \langle m_1, m_2, \ldots, m_k \rangle$ to denote the profile of internal functions for all the participating agents. We also use U_i to denote the set of possible internal functions for agent i in Σ and $U_{\mathcal{M}} = U_1 \times U_2 \times \ldots U_k$ to denote the universe of tuples of internal functions.

The internal function returns the information that is provided by participating agents themselves at a particular state, which is referred to as an *agent type* in this paper and thus might be different from agent to agent. A local rule is defined based on agents' communication as follows:

Definition 3 (Abstract Local Rules) Given a concurrent game structure $\mathcal{S}$, an abstract local rule for agent a is a tuple $\langle \tau_a, \gamma_a \rangle$ consisting of a function $\tau_a(q)$ that maps a state $q \in Q$ to a subset of agents, that is, $\tau_a(q) \subseteq \Sigma$, and a function $\gamma_a(M(q))$ that maps $M(q) = \{m_i(q) \mid i \in \tau_a(q)\}$ to an action available in state q to agent a, that is, $\gamma_a(M(q)) \in d_a(q)$. We denote the set of all the abstract local rules as Γ and a subset of abstract local rules as Γ_A that are designed for the coalition of agents A.

An abstract local rule consists of two parts: the first part $\tau_a(q)$ states the agents with whom agent a is supposed to communicate in state q, and the second part γ_a states the action that agent a is supposed to take given the communication with agents in $\tau_a(q)$ for their internals. We see local rules not only as constraints but also guidance on agents' behavior, namely an agent does not know what to do if he does not communicate with other agents. Therefore, we exclude the case where agents get no constraint from their respective local rules. We use *out* to denote a set of computations and notation *out*$(q, \mathcal{M}, \Gamma_A)$ is the set of computations starting from state q where agents in coalition A follow their respective local rules in Γ_A with internal functions $\mathcal{M}$. A computation $\lambda = q_0, q_1, q_2, ...$ is in *out*$(q_0, \mathcal{M}, \Gamma_A)$ if and only if it holds that for all positions $i \geq 0$ there is a move vector $\langle \alpha_1, \ldots, \alpha_k \rangle \in D(\lambda[i])$ such that $\delta(\lambda[i], \langle \alpha_1, \ldots, \alpha_k \rangle) = \lambda[i+1]$ and for all $a \in A$ it is the case that $\alpha_a = \gamma_a(M(\lambda[i]))$. Moreover, when we refer to a state q in a computation λ, which is in *out*$(q_0, \mathcal{M}, \Gamma_A)$, we will simply write $q \in$ *out*$(q_0, \mathcal{M}, \Gamma_A)$ for short in the rest of this paper. Now we are ready to define a self-organizing multi-agent system. Formally,

Definition 4 (Self-organizing Multi-agent Systems) A self-organizing multi-agent system (SOMAS) is a tuple $(\mathcal{S}, \mathcal{M}, \Gamma)$, where $\mathcal{S}$ is a concurrent game structure, $\mathcal{M}$ is the set of internal functions and Γ is a set of local rules that agents follow.

Example 5 We will use the example in [3, 11] for better understanding the above definitions. Consider a CGS scenario as Figure. 1 where there are two trains, each controlled by an agent, going through a tunnel from the opposite side. The tunnel has only room for one train, and the agents can either wait or go. Starting from state q_0, if the agents choose to go simultaneously, the trains will crash, which is state q_4; if one agent goes and the other waits, they can both successfully pass through the tunnel, which is q_3.

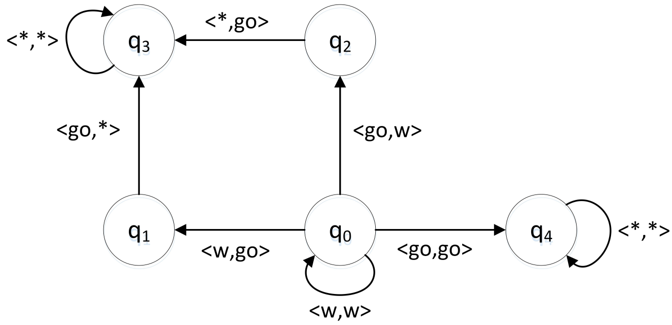

Fig. 1: A CGS example.

Local rules $\langle\tau_1, \gamma_1\rangle$ and $\langle\tau_2, \gamma_2\rangle$ are prescribed for the agents to follow: both agents communicate with each other for their urgencies u_1 and u_2 in state q_0; the one who is more urgent can go through the tunnel first; otherwise, it has to wait. Given the above local rules, if a_1 is more urgent than or as urgent as a_2 with respect to u_1 and u_2, the desired state q_3 is reached along with computation $q_0, q_2, q_3 \ldots$; if a_1 is less urgent than a_2 with respect to u_1 and u_2, the desired state q_3 is reached along with computation $q_0, q_1, q_3 \ldots$.

In order to study how a self-organizing multi-agent system behaves under the change of participating agents, we first need to characterize the independence between agents in terms of their contributions to the system behavior. In this paper, it is characterized from two perspectives: a semantic perspective given by the underlying game structure and a structural perspective derived from abstract local rules. Similar to ATL in [1], our language ATL-Γ is interpreted over a concurrent game structure $\mathcal{S}$ that has the same propositions and agents as our language. It is an extension of classical propositional logic with temporal cooperation modalities. A formula of the form $\langle A\rangle\psi$ means that coalition of agents A will bring about the subformula ψ by following their respective local rules in Γ_A, no matter what agents in $\Sigma\backslash A$ do, where ψ is a temporal formula of the form $\bigcirc\varphi$, $\Box\varphi$ or $\varphi_1\mathcal{U}\varphi_2$ (where φ, φ_1, φ_2 are again formulas in our language). Formally, the grammar of our language is defined below, where $p \in \Pi$ and $A \subseteq \Sigma$:

$$\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle A\rangle\bigcirc\varphi \mid \langle A\rangle\Box\varphi \mid \langle A\rangle\varphi_1\mathcal{U}\varphi_2$$

Given a self-organizing multi-agent system $(\mathcal{S}, \mathcal{M}, \Gamma)$, where $\mathcal{S}$ is a concurrent game structure and Γ is a set of local rules, and a state $q \in Q$, we define the semantics with respect to the satisfaction relation $\models$ inductively as follows:

- $\mathcal{S}, \mathcal{M}, \Gamma, q \models p$ iff $p \in \pi(q)$;
- $\mathcal{S}, \mathcal{M}, \Gamma, q \models \neg\varphi$ iff $\mathcal{S}, \mathcal{M}, \Gamma, q \not\models \varphi$;

- $\mathcal{S}, \mathcal{M}, \Gamma, q \models \varphi_1 \wedge \varphi_2$ iff $\mathcal{S}, \mathcal{M}, \Gamma, q \models \varphi_1$ and $\mathcal{S}, \mathcal{M}, \Gamma, q \models \varphi_2$;
- $\mathcal{S}, \mathcal{M}, \Gamma, q \models \langle A \rangle \bigcirc \varphi$ iff for all $\lambda \in$ *out*$(q, \mathcal{M}, \Gamma_A)$, we have $\mathcal{S}, \mathcal{M}, \Gamma, \lambda[1] \models \varphi$;
- $\mathcal{S}, \mathcal{M}, \Gamma, q \models \langle A \rangle \Box \varphi$ iff for all $\lambda \in$ *out*$(q, \mathcal{M}, \Gamma_A)$ and all positions $i \geq 0$ it holds that $\mathcal{S}, \mathcal{M}, \Gamma, \lambda[i] \models \varphi$;
- $\mathcal{S}, \mathcal{M}, \Gamma, q \models \langle A \rangle \varphi_1 \mathcal{U} \varphi_2$ iff for all $\lambda \in$ *out*$(q, \mathcal{M}, \Gamma_A)$ there exists a position $i \geq 0$ such that for all positions $0 \leq j \leq i$ it holds that $\mathcal{S}, \mathcal{M}, \Gamma, \lambda[j] \models \varphi_1$ and $\mathcal{S}, \mathcal{M}, \Gamma, \lambda[i] \models \varphi_2$.

Dually, we write $\langle A \rangle \Diamond \varphi$ for $\langle A \rangle \top \mathcal{U} \varphi$. We can check a formula in our language to verify whether a coalition of agents will bring about a temporal property through following its local rules, which is regarded as semantic independence in this paper. Moreover, agents in the system follow their respective local rules to communicate with other agents. Based on their communcation, we can see a coalition of agents do not get input information from agents outside the coalition, which is regarded as structural independence in this paper. This gives rise to the notion of independent components.

Definition 6 (Independent Components) Given an SOMAS $(\mathcal{S}, \mathcal{M}, \Gamma)$, a coalition of agents $A \subseteq \Sigma$ and a state q, we say that coalition A is an independent component w.r.t. q if and only if for all $a \in A$ and its abstract local rule $\langle \tau_a, \gamma_a \rangle$ it is the case that $\tau_a(q) \subseteq A$; a coalition of agents A is an independent component w.r.t. a set of computations *out* iff for all $\lambda \in$ *out* and $q \in \lambda$, A is an independent component with respect to state q.

In other words, an independent component might output information to agents in $\Sigma \backslash A$, but do not get input from agents in $\Sigma \backslash A$. As we can see, it is a structural property given by the communication between agents in a given state. We then propose the notions of *semantic independence, structural independence and full contribution* to characterize the independence between agents from different perspectives.

Definition 7 (Semantic Independence, Structural Independence and Full Contribution) Given an SOMAS $(\mathcal{S}, \mathcal{M}, \Gamma)$, a coalition of agents A and a state q,

- A is semantically independent with respect to a temporal formula ψ from q iff $\mathcal{S}, \mathcal{M}, \Gamma, q \models \langle A \rangle \psi$;
- A is structurally independent from q iff A is an independent component w.r.t. the set of computations *out*$(q, \mathcal{M}, \Gamma_A)$;
- A has full contribution to ψ in q iff A is the minimal (w.r.t. set-inclusion) coalition that is both semantically independent with respect to ψ and structurally independent from q.

In other words, coalition A has full contribution to ψ because any subset of coalition A is either semantically dependent with respect to ψ or structurally dependent. Given a self-organizing multi-agent system and a set of temporal formulas, we check with each subset of agents and each temporal formula for their semantic and structural independence in order to find which coalition has full contribution to which property. One can refer to [6] for detailed discussion about the above definitions and examples in this section.

Example 8 We continue to discuss the two-train example. Neither train, namely a_1 or a_2 as a coalition, has full contribution to the result of passing through the tunnel without crash. The reasons are listed as follows: any single train cannot ensure the result of passing through the tunnel without crash, which means that any single train is not semantically independent and can be expressed:

$$\mathcal{S}, \mathcal{M}, \Gamma, q_0 \not\models \langle a_1 \rangle \Diamond \text{passed}_1,$$

$$\mathcal{S}, \mathcal{M}, \Gamma, q_0 \not\models \langle a_2 \rangle \Diamond \text{passed}_2.$$

Moreover, both trains follow their local rules to communicate with each other in state q_0, which means that any single train is not an independent component w.r.t. state q_0 thus not being structurally independent.

Two trains have full contribution to the result of passing through the tunnel without crash, because both agents by themselves can bring about the result of passing through the tunnel without crash through following the local rules, which can be expressed:

$$\mathcal{S}, \mathcal{M}, \Gamma, q_0 \models \langle a_1, a_2 \rangle \Diamond (\text{passed}_1 \land \text{passed}_2).$$

Moreover, the coalition of two trains is obviously an independent component w.r.t. $out(q_0, \mathcal{M}, \Gamma_{\{a_1,a_2\}})$, which means that it is structurally independent, and the coalition of two trains is obviously the minimal coalition that is both semantically independent w.r.t. the result of passing the tunnel without crash and structurally independent from state q_0.

3 Changing Participating Agents

When a self-organizing multi-agent system is deployed in an open environment, different types of agents participate in the system and their internals might be unknown to system designers. Therefore, it is important to verify whether the set of local rules still generates desired outcomes, more generally how the system behaves, under the change of participating agents. Because in this paper an internal function m_i refers to an agent type and might be interpreted as agents' preferences, interests or other internal information over states that can be different from agent to agent, changing participating agents can be done by simply replacing internal functions so that the original internal function and the new internal function return different values in the same state. Here we assume that the number of agents in the system does not change so the indexes of agents in Σ remain the same. We have the following definition:

Definition 9 (Change of Participating Agents) Given an original SOMAS $(\mathcal{S}, \mathcal{M}, \Gamma)$, a new SOMAS under the change of participating agents is denoted as $(\mathcal{S}, \mathcal{M}', \Gamma)$, where $\mathcal{M}' \in U_M$ and there exists an agent $a \in \Sigma$ and a state $q \in Q$ such that $m_a(q) \neq m'_a(q)$.

As we can see, what we meant by replacing internal functions is that in the new system there exists at least an agent's internal function such that given the same state its output is different from the one in the original system. Except the internal functions, the underlying concurrent game structure and the set of local rules remain the same.

Changing participating agents in the system might cause undesired outcomes. Agents communicate with each other for their internals and make a move based on the communication results. The change of agents' internal functions might change the actions that agents are allowed to take by their local rules, and consequently the new system runs along a computation that might be different from the original system, making undesired properties hold. Therefore, local rules have to be well designed in order to ensure that desired properties remain unchanged under the change of participating agents.

Example 10 In the two-train example, suppose both trains have the same urgency, denoted as $\mathcal{M}'$, they will choose to wait because neither train is more urgent than the other one, resulting in the undesired state q_0 of deadlock. As we can see, the local rules cannot ensure that the system reaches a desired state no matter what kind of agents participate in the system.

Given a self-organizing multi-agent system $(\mathcal{S}, \mathcal{M}, \Gamma)$ and a temporal formula as an indicated property, in order to find out which coalition of agents has full contribution to a temporal formula, we need to do model-checking with each subset of Σ and that formula. Intuitively we can follow the same process to verify the new system $(\mathcal{S}, \mathcal{M}', \Gamma)$ as what we did with the original system. However, as we have proved in [6], checking whether a coalition of agents has full contribution to a temporal formula is computationally expensive, because we have to enumerate all the subsets of the coalition and check for their semantic and structural independence in order to ensure the minimality, making verifying both the original system and the new system afresh inefficient and difficult. Thus, it is important to think about whether we can use the verification result with the original system to better verify the new system. In the area of formal argumentation, Beishui Liao, et al propose a division-based method to compute the status of arguments when the argumentation system is updated. ([5]) Inspired by that, we can divide the system into two parts: the set of agents whose internal functions remain the same and the set of agents whose internal functions change. For the first part, as was in Definition 9, when a self-organizing multi-agent system switches to another under the change of participating agents, it is not necessary that the internal functions of all agents change, which means that there might exist some agents whose internal functions remain the same. Such agents are called an unchanged set. Formally:

Definition 11 (Unchanged Sets) Given two SOMASs $(\mathcal{S}, \mathcal{M}, \Gamma)$ and $(\mathcal{S}, \mathcal{M}', \Gamma)$ under the change of agents' internal functions from $\mathcal{M}$ to $\mathcal{M}'$, an unchanged set with respect to $\mathcal{M}$ and $\mathcal{M}'$ is defined as $\mathit{US}(\mathcal{M}, \mathcal{M}') = \{a \in \Sigma \mid \text{ for all } q \in Q : m_a(q) = m'_a(q)\}$.

An unchanged set is symmetric in terms of the change of agents' internal functions.

Proposition 12 Given two SOMASs $(\mathcal{S}, \mathcal{M}, \Gamma)$ and $(\mathcal{S}, \mathcal{M}', \Gamma)$, $\mathit{US}(\mathcal{M}, \mathcal{M}') = \mathit{US}(\mathcal{M}', \mathcal{M})$.

Proof We can easily prove it following Definition 11. □

A coalition of agents has full contribution to a temporal formula if and only if it is the minimal coalition that is both structurally and semantically independent. We can intuitively imagine that a coalition behaves the same if the internal function of each agent inside the coalition does not change, which means that we can reuse the verification information for the original system when verifying the new system. Thus, one important property of unchanged sets is that the full contributions of agents inside an unchanged set remain the same when the internal functions switch from $\mathcal{M}$ to $\mathcal{M}'$.

Proposition 13 Given two SOMASs $(\mathcal{S}, \mathcal{M}, \Gamma)$ and $(\mathcal{S}, \mathcal{M}', \Gamma)$ under the change of agents' internal functions from $\mathcal{M}$ to $\mathcal{M}'$, and a state q, if for any $a \in A, q' \in Q, i \in \tau_a(q')$ it is the case that $m_i(q') = m'_i(q')$, then $\mathit{out}(q, \mathcal{M}, \Gamma_A) = \mathit{out}(q, \mathcal{M}', \Gamma_A)$.

Proof Let $\mathit{out}(q, \mathcal{M}, \Gamma_A)[i]$ be a set of states, each of which is the ith state of any computation in $\mathit{out}(q, \mathcal{M}, \Gamma_A)$. That is, $\mathit{out}(q, \mathcal{M}, \Gamma_A)[i] = \{q' \in Q \mid \exists \lambda \in \mathit{out}(q, \mathcal{M}, \Gamma_A) : q' = \lambda[i]\}$. Next, we need to inductively prove that $\mathit{out}(q, \mathcal{M}', \Gamma_A)$ in the new system contains the same computations as $\mathit{out}(q, \mathcal{M}, \Gamma_A)$ in the original system. Firstly, computations from both $\mathit{out}(q, \mathcal{M}, \Gamma_A)$ and $\mathit{out}(q, \mathcal{M}', \Gamma_A)$ start from state q. Secondly, suppose $\mathit{out}(q, \mathcal{M}, \Gamma_A)[i] = \mathit{out}(q, \mathcal{M}', \Gamma_A)[i]$. If for any $a \in A, q' \in Q, i \in \tau_a(q')$ we have $m_i(q') = m'_i(q')$, then $M(q') = M'(q')$, consequently $\gamma_a(M(q')) = \gamma_a(M'(q'))$, which means that $\gamma_a(\cdot)$ returns the same action for each agent in state q' in both systems. Hence, $\mathit{out}(q, \mathcal{M}, \Gamma_A)[i+1] = \mathit{out}(q, \mathcal{M}', \Gamma_A)[i+1]$. So we can conclude that $\mathit{out}(q, \mathcal{M}, \Gamma_A) = \mathit{out}(q, \mathcal{M}', \Gamma_A)$. □

From that, we can see the properties of computation set $out(q, \mathcal{M}, \Gamma_A)$ remain the same under the change of agents' internal functions from $\mathcal{M}$ to $\mathcal{M}'$ if the internal functions of agents with which coalition A needs to communicate remain the same. This gives rise to one important property of unchanged sets: the full contributions of agents inside an unchanged set remain the same when the internal functions switch from $\mathcal{M}$ to $\mathcal{M}'$.

Theorem 14. *Given two SOMASs $(\mathcal{S}, \mathcal{M}, \Gamma)$ and $(\mathcal{S}, \mathcal{M}', \Gamma)$ under the change of agents' internal functions from $\mathcal{M}$ to $\mathcal{M}'$, for any coalition $A \subseteq US(\mathcal{M}, \mathcal{M}')$, A has full contribution to a temporal formula ψ in a state q in $(\mathcal{S}, \mathcal{M}, \Gamma)$ iff A has full contribution to ψ in q in $(\mathcal{S}, \mathcal{M}', \Gamma)$.*

Proof We first prove the forward part. Because coalition A has full contribution to ψ in state q in SOMAS $(\mathcal{S}, \mathcal{M}, \Gamma)$, we have $\mathcal{S}, \mathcal{M}, \Gamma, q \models \langle A \rangle \psi$ (semantic independence), and A is an independent component w.r.t. $out(q, \mathcal{M}, \Gamma_A)$ (structural independence), and any subset of A is either semantically dependent with respect to ψ or structurally dependent from state q in $(\mathcal{S}, \mathcal{M}, \Gamma)$. Because $A \subseteq US(\mathcal{M}, \mathcal{M}')$, the internal function of any agent in A remains the same from $\mathcal{M}$ to $\mathcal{M}'$. By Proposition 13, $out(q, \mathcal{M}, \Gamma_A) = out(q, \mathcal{M}', \Gamma_A)$. That implies the properties that hold for $out(q, \mathcal{M}, \Gamma_A)$ also hold for $out(q, \mathcal{M}', \Gamma_A)$. Thus, by Definition 7, we have $\mathcal{S}, \mathcal{M}', \Gamma, q \models \langle A \rangle \psi$ (semantic independence), and A is an independent component w.r.t. $out(q, \mathcal{M}', \Gamma_A)$ (structural independence). Next, we need to prove whether A is also the minimal coalition that is both semantically independent with respect to ψ and structurally independent from q in $(\mathcal{S}, \mathcal{M}', \Gamma)$. Suppose there exists a coalition $A' \subset A$ that is both semantically independent with respect to ψ and structurally independent from q in $(\mathcal{S}, \mathcal{M}', \Gamma)$, which means that $\mathcal{S}, \mathcal{M}', \Gamma, q \models \langle A' \rangle \psi$ and A' is an independent component w.r.t. $out(q, \mathcal{M}', \Gamma_{A'})$. Because $A' \subset A \subseteq US(\mathcal{M}, \mathcal{M}') = US(\mathcal{M}', \mathcal{M})$, the internal function of any agent in A' remains the same from $\mathcal{M}'$ to $\mathcal{M}$. By Proposition 13, $out(q, \mathcal{M}, \Gamma_{A'}) = out(q, \mathcal{M}', \Gamma_{A'})$, which implies that $\mathcal{S}, \mathcal{M}, \Gamma, q \models \langle A' \rangle \psi$ and A' is an independent component w.r.t. $out(q, \mathcal{M}, \Gamma_{A'})$ and thus contradicts with the premise that A is the minimal coalition that is both semantically independent and structurally independent. Hence, coalition A is also the minimal coalition that is semantically independent w.r.t. ψ and structurally independent from state q in the new system. Therefore, coalition A also has full contribution to ψ in state q in $(\mathcal{S}, \mathcal{M}', \Gamma)$. Because $A \subseteq US(\mathcal{M}, \mathcal{M}') = US(\mathcal{M}', \mathcal{M})$, we can prove the backward part in a similar way. □

Therefore, given that the internal functions of coalition A remain the same, if we already know that coalition A has full contribution to ψ in state q in the original system, it will remain the same in the new system; if we already know that coalition A does not have full contribution to ψ in state q in the original system, it will also remain the same in the new system. That means the verification information regarding agents in the unchanged set will remain the same under the change of agents' internal functions, which can be used to simplify the verification of the new system. To find the unchanged set with respect to $\mathcal{M}$ and $\mathcal{M}'$, we can simply check the internal functions of all the agents with all the states, which can be done in polynomial time to the number of states and the number of agents. After we obtain the unchanged set, we can copy the information about agents' full contributions within the unchanged set from the original system to the new system. Moreover, since the full contributions of agents in the unchanged set remain the same, the coalitions that we have to check with for the new system become $\mathcal{P}(\Sigma)\backslash\mathcal{P}(US(\mathcal{M},\mathcal{M}'))$ instead of $\mathcal{P}(\Sigma)$ for the original system. The propositions below will give insights to further simplify the verification of the new system.

Proposition 15 Given an SOMAS $(\mathcal{S},\mathcal{M},\Gamma)$, a coalition of agents A, a temporal formula ψ and a state q, if for any $A' \subseteq A$ it is the case that A' does not have full contribution to ψ in state q, then A' is either semantically dependent w.r.t. ψ or structurally dependent from q.

Proof Suppose there exists $A' \subseteq A$ such that it is both semantically independent w.r.t. ψ and structurally independent from q. If A' is the minimal one, A' has full contribution to ψ in state q; if A' is not the minimal one, there exists $A'' \subset A'$ such that it is the minimal coalition that is both semantically independent w.r.t. ψ and structurally independent from q. Both cases contradict with the premise that for any $A' \subseteq A$ it is the case that A' does not have full contribution to ψ in state q. Thus, A' is either semantically dependent w.r.t. ψ or structurally dependent from q. □

Theorem 16. *Given two SOMASs $(\mathcal{S},\mathcal{M},\Gamma)$ and $(\mathcal{S},\mathcal{M}',\Gamma)$ under the change of agents' internal functions from $\mathcal{M}$ to $\mathcal{M}'$, a coalition of agents $A \not\subseteq US(\mathcal{M},\mathcal{M}')$, a temporal formula ψ and a state q, A has full contribution to ψ in state q in $(\mathcal{S},\mathcal{M}',\Gamma)$ iff both of the following statements are satisfied:*

1. *for any $A' \subseteq US(\mathcal{M},\mathcal{M}') \cap A$ it is the case that A' does not have full contribution to ψ in state q in $(\mathcal{S},\mathcal{M},\Gamma)$;*
2. *A is the minimal (w.r.t. set-inclusion) coalition within $\mathcal{P}(A)\backslash\mathcal{P}(US(\mathcal{M},\mathcal{M}'))$ that is both semantically independent with respect to ψ and structurally independent from q in $(\mathcal{S},\mathcal{M}',\Gamma)$.*

Proof Since for any $A' \subseteq US(\mathcal{M},\mathcal{M}') \cap A$ it is the case that A' does not have full contribution to ψ in state q in $(\mathcal{S},\mathcal{M},\Gamma)$, we then have for any $A' \subseteq US(\mathcal{M},\mathcal{M}') \cap A$ it is the case that A' does not have full contribution to ψ in state q in $(\mathcal{S},\mathcal{M}',\Gamma)$. That also implies A' is either semantically dependent w.r.t. ψ or structurally dependent from state q in $(\mathcal{S},\mathcal{M}',\Gamma)$. Combining with the second statement, we can conclude that A has full contribution to ψ in state q in $(\mathcal{S},\mathcal{M}',\Gamma)$. □

In order to verify whether a coalition of agents A has full contribution to ψ, normally we have to check the semantic independence and structural independence of any subset of A in order to ensure the minimality, which is computationally expensive as the size of the coalition increases. While Theorem 14 shows us the case where $A \subseteq US(\mathcal{M}, \mathcal{M}')$, Theorem 16 shows us how we can further utilize the information about agents' full contribution from the original system in the new system when $A \nsubseteq US(\mathcal{M}, \mathcal{M}')$. If we have proved that any subset of the unchanged set does not have full contribution to ψ in the original system, we do not need to check their semantic independence and structural independence when verifying the new system, because they will not hold as in the original system. With the verification information from the original system, we can decrease the coalitions that we have to check from $\mathcal{P}(A)$ to $\mathcal{P}(A)\backslash\mathcal{P}(US(\mathcal{M}, \mathcal{M}') \cap A)$. We can use Theorem 16 and Theorem 14 to more efficiently verify whether a coalition of agents has full contribution to a temporal formula in the new system. In the next section, we will use an example to illustrate how we can use the verification result from the original system in the verification of the new system based on Theorems 14 and 16.

4 Example

We extend the two-train example to illustrate the above theory. Suppose we have a U-shape traffic system depicted as in Fig. 2. The top is the same as the previous example where trains a_1 and a_2, each controlled by an agent, are going through a tunnel from the opposite side. Agent a_2 exits the traffic system from Exit1 after passing through the tunnel, while agent a_1 needs to go down to exit the traffic system from Exit3 after passing through the tunnel. Below the tunnel there is a second tunnel, and on the left hand side another train, controlled by agent a_3, is also going through the tunnel to exit the system from Exit3. The whole traffic system has only room for one train to go through, and the agents can either wait or go. Since agent a_1 needs to go down to exit the traffic system after passing through the first tunnel, it will clash with agent a_3 by Exit3 if they go simultaneously. The CGS is depicted in Fig. 3.

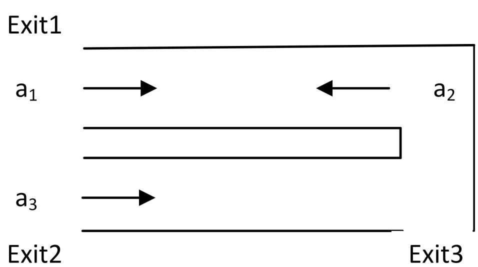

Fig. 2: A traffic system consisting of three trains.

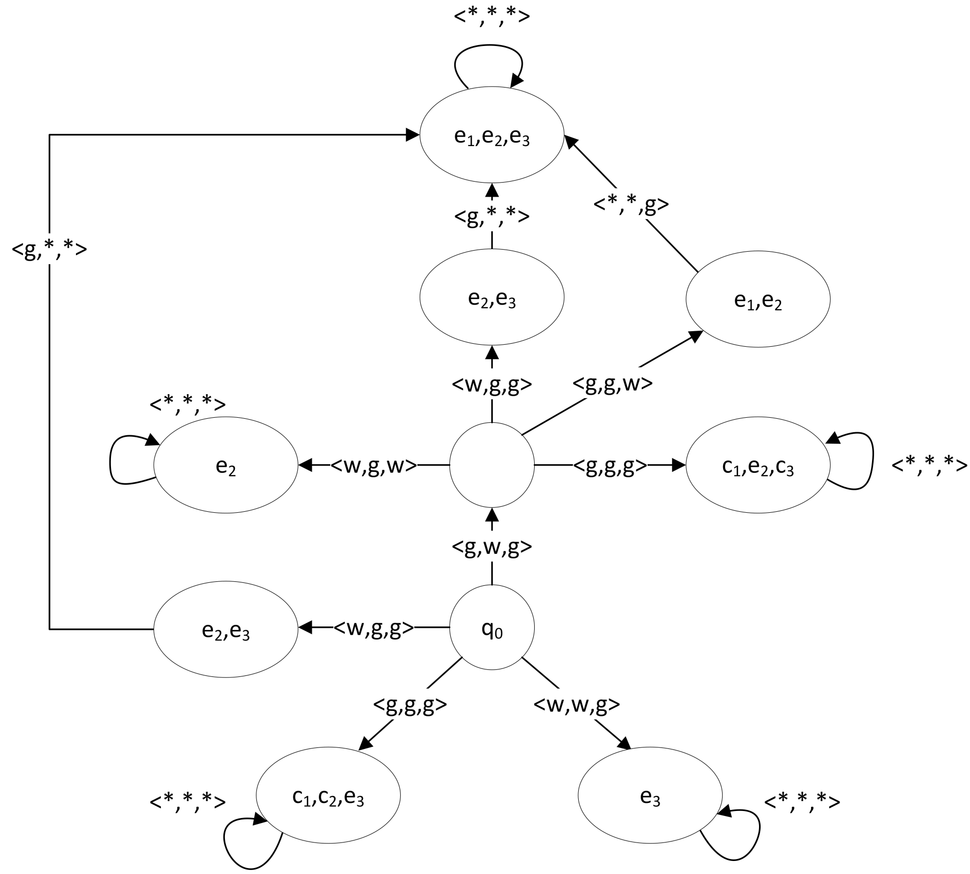

Fig. 3: CGS of the three-train example, where formula c_i means that train i clashes in the tunnel and formula e_i means that train i exits the traffic system.

The local rules for three agents are quite simple: whenever there is space limitation for two trains to go through, the one which is more urgent can go first and the other one which less urgent needs to wait. The set of temporal formulas is $\{\Diamond e_1, \Diamond e_2, \Diamond e_3\}$, each of which means that agent i exits the system. Given an SOMAS $(\mathcal{S}, \mathcal{M}, \Gamma)$, suppose agent a_2 is more urgent than agent a_1, so agent a_1 waits until agent a_2 passes through the first tunnel. Because agent a_3 goes while agent a_1 is waiting, agent a_3 can exit the system without meeting agent a_1 by the Exit3. Thus, we have the following verification information in Table 1. From Table 1 we can see that coalition $\{a_1, a_2\}$ has full contribution to the result of $\Diamond e_1$ and $\Diamond e_2$ in state q_0 in $(\mathcal{S}, \mathcal{M}, \Gamma)$ as what we had previously, and $\{a_1, a_2, a_3\}$ has full contribution to $\Diamond e_3$ in state q_0 in $(\mathcal{S}, \mathcal{M}, \Gamma)$. In particular, agent a_3 cannot bring about the result of exiting the system by itself, because if a_1 goes and a_2 waits, which means that they do not follow their local rules to behave, a_1 and a_3 will meet Exit3 and need to cooperate in order to exit the system.

	$\emptyset$	$\{a_1\}$	$\{a_2\}$	$\{a_3\}$	$\{a_1, a_2\}$	$\{a_1, a_3\}$	$\{a_2, a_3\}$	$\{a_1, a_2, a_3\}$
$\Diamond e_1$	$\times$	$\times$	$\times$	$\times$	✓	$\times$	$\times$	✓
$\Diamond e_2$	$\times$	$\times$	$\times$	$\times$	✓	$\times$	$\times$	✓
$\Diamond e_3$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	✓
SI	$\times$	$\times$	$\times$	$\times$	✓	$\times$	$\times$	✓

Table. 1: Verification information for SOMASs $(\mathcal{S}, \mathcal{M}, \Gamma)$, where "SI" stands for structural independent.

	$\emptyset$	$\{a_1\}$	$\{a_2\}$	$\{a_3\}$	$\{a_1, a_2\}$	$\{a_1, a_3\}$	$\{a_2, a_3\}$	$\{a_1, a_2, a_3\}$
$\Diamond e_1$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	✓
$\Diamond e_2$	$\times$	$\times$	$\times$	$\times$	✓	$\times$	$\times$	✓
$\Diamond e_3$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	✓
SI	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	✓

Table. 2: Verification information for SOMAS $(\mathcal{S}, \mathcal{M}', \Gamma)$, where "SI" stands for structural independent.

Now consider a new SOMAS $(\mathcal{S}, \mathcal{M}', \Gamma)$ under the change of internal functions from $\mathcal{M}$ and $\mathcal{M}'$, where the internal functions of agents a_1 and a_2 change while the internal function of agent a_3 remains the same. In this case, agent a_1 chooses to go and agent a_2 chooses to wait. We can see that a_1 and a_3 will meet by Exit3 and thus they need to communicate with each other for their urgencies. Moreover, since only the internal function of a_3 remains the same, and coalitions $\{a_3\}$ and $\emptyset$ do not have full contribution to $\Diamond e_1$, $\Diamond e_2$ or $\Diamond e_3$ in the original system $(\mathcal{S}, \mathcal{M}, \Gamma)$, by Theorem 16, we can reuse the verification information about $\{a_3\}$ from the original system by copying the columns of $\emptyset$ and $\{a_3\}$ from Table 1 to Table 2. It shows that $\{a_3\}$ and $\emptyset$ are either semantically dependent or structurally dependent. Copying this information can simplify the process of ensuring the minimality when doing model-checking. For example, for checking whether coalition $\{a_1, a_2, a_3\}$ has full contribution to $\Diamond e_3$ in state q_0 in $(\mathcal{S}, \mathcal{M}', \Gamma)$, normally we need to check both semantic and structural independence for each subset of $\{a_1, a_2, a_3\}$, which is $\mathcal{P}(\{a_1, a_2, a_3\})$; since we know from the original system that $\emptyset$ and $\{a_3\}$ are either semantically dependent w.r.t. $\Diamond e_3$ or structurally dependent, the coalitions we need to check become $\mathcal{P}(\{a_1, a_2, a_3\}) \backslash \{\emptyset, \{a_3\}\}$. We then have the following verification information in Table 2. As we can see, coalition $\{a_1, a_2, a_3\}$ is the minimal coalition that is both semantically independent (w.r.t. $\Diamond e_1$, $\Diamond e_2$ or $\Diamond e_3$) and structurally independent, so coalition $\{a_1, a_2, a_3\}$ has full contribution to $\Diamond e_1$, $\Diamond e_2$ and $\Diamond e_3$ in state q_0 in $(\mathcal{S}, \mathcal{M}', \Gamma)$.

5 Conclusion

When a self-organizing multi-agent system is deployed in an open environment, the change of participating agents might bring the system to an undesired state. As it is computationally expensive to verify a self-organizing multi-agent system, we need to think about how we can properly use the verification result that we get from the original system to better verify the new system. In this paper, we propose a framework to reason about the dynamics of self-organizing multi-agent systems under the change of participating agents. Agents in the system are divided into two parts: one whose internal functions remain the same and the other one whose internal functions change. We first proved that the full contribution of a coalition of agents remains the same if the internal function of any agent in the coalition remains the same. Furthermore, the properties of a coalition regarding not being semantically independent or structurally independent remain the same if the internal function of any agent in the coalition remains the same, which means that we do not need to check their semantic independence and structural independence when verifying the new system. Future work can be done in the direction of reasoning about the dynamics of self-organizing multi-agent systems in terms of revising agents' local rules.

References

[1] R. Alur, T. A. Henzinger and O. Kupferman, 2002, "Alternating-time temporal logic", *Journal of the ACM (JACM)*, **49(5)**: 672–713.

[2] N. Bulling and M. Dastani, 2016, "Norm-based mechanism design", *Artificial Intelligence*, **239(C)**: 97–142.

[3] M. Knobbout and M. Dastani, 2012, "Reasoning under compliance assumptions in normative multiagent systems", *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, **Vol. 1**, pp. 331–340, Richland County, SC: International Foundation for Autonomous Agents and Multiagent Systems.

[4] M. Knobbout, M. Dastani and J.-J. Meyer, 2016, "Formal frameworks for verifying normative multi-agent systems", *Theory and Practice of Formal Methods*, pp. 294–308, Cham: Springer.

[5] B. Liao, L. Jin and R. C. Koons, 2011, "Dynamics of argumentation systems: A division-based method", *Artificial Intelligence*, **175(11)**: 1790–1814.

[6] J. Luo, B. Liao and J.-J. Meyer, 2021, "A formal framework for reasoning about agents' independence in self-organizing multi-agent systems", preprint, https://arxiv.org/pdf/2105.07648.pdf.

[7] N. Nisan, 2007, "Introduction to mechanism design (for computer scientists)", in N. Nisan et al. (eds.), *Algorithmic Game Theory,* **Ch. 9**, pp. 209–242, Cambridge: Cambridge University Press.

[8] G. Picard, C. Bernon and M.-P. Gleizes, 2005, "Etto: Emergent timetabling by cooperative self-organization", *International Workshop on Engineering Self-Organising Applications*, pp. 31–45, Berlin; New York: Springer.

[9] G. Valentini, H. Hamann and M. Dorigo, 2014, "Self-organized collective decision making: The weighted voter model", *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, pp. 45–52, Richland County, SC: International Foundation for Autonomous Agents and Multiagent Systems.

[10] F. Wang, 2002, "Self-organising communities formed by middle agents", *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 3*, pp. 1333–1339, New York: ACM.

[11] M. Wooldridge and W. van der Hoek, 2005, "On obligations and normative ability: Towards a logical analysis of the social contract", *Journal of Applied Logic*, **3(3–4)**: 396–420.

自组织多主体系统动态性的推理研究

罗捷婷　　廖备水 ✉　　约翰-朱尔斯·迈尔

摘　要

自组织作为一种自发解决困难问题的内部控制过程已被应用在多主体系统中。当这类系统部署在开放环境中，参与主体的改变有可能导致系统走向不理想的状态。因此，知道改变参与主体后系统哪些特性保持不变哪些特性发生变化相当重要。由于验证自组织多主体系统的计算复杂度很高，因此需要思考如何正确使用原系统的验证信息以提高验证新系统的效率。本文提出一个理论框架，用于推理自组织多主体系统由于改变参与主体而带来的动态性。

罗捷婷　浙江大学哲学系
luojieting@zju.edu.cn

廖备水 ✉　浙江大学哲学系
baiseliao@zju.edu.cn

约翰-朱尔斯·迈尔　乌特勒支大学信息与计算科学系
J.J.C.Meyer@uu.nl