Computing and Informatics, Vol. 44, 2025, 1336–1367, doi: 10.31577/cai_2025_6_1336

PETRI NET STRUCTURAL REDUCTION FOR TEMPORAL EPISTEMIC LOGIC VERIFICATION IN MULTI-AGENT SYSTEMS

Tong Guo

School of Computer Science and Technology
Tongji University, Shanghai, 201804, China
&
Key Lab of Embedded System and Service Computing, Ministry of Education
Tongji University, Shanghai, 201804, China

Meiqin Pan

School of Business and Management
Shanghai International Studies University, Shanghai, 200083, China

Zhijun Ding*

School of Computer Science and Technology
Tongji University, Shanghai, 201804, China
&
Key Lab of Embedded System and Service Computing, Ministry of Education
Tongji University, Shanghai, 201804, China
e-mail: zhijun_ding@outlook.com

Abstract. With the increasingly widespread application of MAS (Multi-Agent System), recent years have witnessed growing research interest in the verification of MAS properties. Similar to traditional concurrent and multi-component systems, the paramount challenge in MAS verification is the state space explosion problem. One approach to mitigating this issue is to reduce the original model before gen-

* Corresponding Author

erating the state space, as reductions at the model level can often substantially alleviate state space complexity. However, for MAS modelled using Petri nets, existing methods primarily focus on reduction at the state space level. In contrast, this paper addresses reduction at the underlying model level by proposing a solution of structural reduction for Petri nets that preserves temporal epistemic logic properties. The main contributions consist of three aspects. Firstly, since existing structural reduction rules for Petri nets are not suitable for temporal epistemic logic verification, modifications and extensions to some of the rules are introduced to accommodate temporal epistemic logic, and corresponding theorems are provided to guarantee the correctness of these rules. Furthermore, given that the applicability of structural reduction rules in Petri nets is constrained by transition visibility, this paper conducts a refined visibility analysis of transitions based on semantic characteristics of epistemic logic for a subclass of Petri nets and a specific category of properties. The resulting analysis relaxes the visibility constraints when applying the rules, and its correctness is formally guaranteed by a theorem. Finally, case studies demonstrate that the proposed structural reduction rules save space overhead during verification and achieve better reduction performance under relaxed visibility constraints.

Keywords: Petri nets, multi-agent systems, temporal epistemic logic, structural reduction, transition visibility

1 INTRODUCTION

MAS [1] is a distributed computational system composed of multiple agents that interact within shared environment. This framework facilitates modular decomposition of complex systems into smaller, manageable units capable of collaborative coordination and communication, enabling the achievement of individual or collective objectives. By the idea of cooperation and competition among different agents in MAS, researchers have developed centralized computing into distributed computing, breaking down the problem to be solved into some subtasks, each agent completing its own specific task. The solving of the entire problem can be seen as the result of different agents communicating, collaborating, and competing with each other based on their own interests and requirements. Compared with centralized problem solving systems, MAS has higher adaptability and flexibility. With the improvement of computing power and the advent of the Internet of Everything era, MAS has been widely applied in various fields such as social networks [2, 3], autonomous systems [4, 5], social simulation [6, 7], security protocols [8, 9], cloud computing [10], and web services [11].

Due to the widespread application of MAS in many technical fields, there is an increasing need for tools that can design, implement, and validate the system. Especially when multi-agent systems are applied in security critical fields, it is significant to verify whether they meet design requirements. Model checking [12] is a mainstream formal verification method based on logic. In this paradigm, a system S to be verified is encoded as a transition system or model M_S programmed in a dedicated modelling language. A specification P of the system is represented as a logical formula φ_P. Verifying whether the system S satisfies the specification P is encoded as the problem of checking whether the model M_S satisfies the logical formula φ_P, formally written as $M_S \models \varphi_P$.

The model checking of MAS has been successfully applied in fields such as security protocols [8, 13], autonomous systems [4], and web services [11], playing an important role in ensuring the system properties. Among the numerous logics that express the properties of a system, temporal epistemic logic can not only depict the behavior and state evolution of the system, but also clearly and accurately describe the knowledge capabilities and cognitive situations of agents in intelligent systems. Therefore, it has received increasing attention. Common temporal epistemic logics include LTLK and CTLK, which are obtained by combining temporal logic LTL and CTL [12] with the Knowledge components, respectively. The model checking of epistemic logic was first proposed by Halpern and Vardi [14], but it did not initially incorporate temporal components. Afterwards, a comprehensive discussion on knowledge reasoning was conducted in [15], advancing an intuitive and mathematical framework for understanding and analyzing knowledge reasoning. Van der Meyden and Shilov [16] studied the complexity of the model checking problem regarding knowledge and time in infinite state systems, and pointed out that the complexity of the problem is PSPACE complete in the best case and undecidable in the worst case. Van der Hoek and Wooldridge [17] first proposed a model checking method that combines temporal logic and epistemic logic. Subsequently, various model checking methods and tools for temporal epistemic logic have emerged [13, 18, 19, 20, 21, 22, 23, 24].

Due to the state-space traversal nature of model checking, the primary challenge in MAS model checking is the state space explosion problem. One approach to mitigating this issue involves reducing the original model before generating the state space, as reductions at the model level can often substantially alleviate state space complexity. Owing to its strengths in modeling concurrency and facilitating localized behavioral analysis, Petri net based modeling has been widely adopted in the MAS domain [25, 26, 27, 28, 29, 30, 31, 32, 33, 34]. This work focuses on the most basic and general form of Petri nets, with the aim of developing a solution of structural reduction on this model to achieve better space performance. The core idea of structural reduction is to eliminate redundant parts of the net or to omit intermediate states that do not affect properties of interest by merging transition firings. The locality of Petri net transition firing semantics provides a foundation for defining reduction rules.

Berthelot [35] conducted pioneering work on structural reduction of Petri nets, proposing rules that preserve net-level properties such as boundedness, safeness, and liveness. Poitrenaud and Pradat-Peyre [36] extended structural reduction methods to temporal logic, introducing the classic pre-aggregation and post-aggregation rules, which are among the most well-known in the field. Building on this, Esparza and

Schröter [37] and Shi et al. [38] respectively proposed additional reduction rules for the temporal logic $\mathrm{LTL}_{\text{-X}}$ [12]. Subsequently, structural reduction methods were extended to colored Petri nets [39, 40] and weighted nets with inhibitor arcs [41]. In recent years, Bønneland et al. [42] defined 8 simplification rules for reachability analysis in Petri nets. Berthomieu et al. [43] introduced 6 reduction rules to facilitate efficient computation of the count of reachable markings. Thierry-Mieg [44] provided a relatively systematic summary of structural reduction, refining 22 rules for deadlock-free and safety detection in systems.

Analysis reveals that existing structural reduction methods face the following challenges in the context of temporal epistemic logic verification. Firstly, current structural reduction rules for Petri nets are not directly applicable to temporal epistemic logic. The introduction of epistemic operators imposes additional requirements on property satisfaction and alters the semantic implications of transition firings with respect to properties. Consequently, models reduced using conventional rules cannot guarantee the preservation of temporal epistemic properties. Furthermore, structural reduction rules are constrained by transition visibility, typically requiring reduced transitions to be invisible. This limits their effectiveness in nets with numerous visible transitions.

To address these limitations, this paper improves existing structural reduction methods by comprehensively incorporating epistemic logic semantics, developing a solution of structural reduction for Petri nets that preserves temporal epistemic properties. The work of this article is primarily divided into two aspects. The first aspect is to adapt structural reduction to support temporal epistemic logic verification. The second aspect is to refine the theory to achieve superior reduction performance. Specifically, the main contributions are as follows:

1. Since existing structural reduction methods for Petri nets are not applicable to temporal epistemic logic verification, this paper modifies and extends a set of structural reduction rules to adapt them to the logic. Key concepts such as transition visibility and stuttering equivalence of paths under temporal epistemic logic are introduced. Furthermore, theorems are established to guarantee the correctness of the proposed rules.
2. While existing structural reduction rules for Petri nets are constrained by transition visibility, this work focuses on a subclass of Petri nets and a specific category of properties. By incorporating semantic characteristics of epistemic logic, it is observed that changes in the truth values of atomic propositions do not necessarily affect the satisfaction of the overall property. Consequently, a refined visibility analysis of transitions is conducted, relaxing the visibility constraints when applying reduction rules. The correctness of the analysis is formally guaranteed by a theorem.
3. This paper selects two classical MAS scenarios for case studies, demonstrating that the proposed structural reduction rules are capable of saving space overhead during verification and achieve superior reduction performance after relaxing the restrictions of transition visibility.

The reminder of the article is organized as follows. Section 2 introduces some basic concepts of models and specifications, establishing the theoretical foundation. Section 3 presents the motivation for this work by analyzing limitations of existing approaches. Section 4 presents the improved structural reduction rules for Petri nets adapted to temporal epistemic logic. Section 5 analyzes how visibility constraints can be relaxed when applying these rules to a specific subclass of Petri nets and a targeted category of properties. Section 6 demonstrates the effectiveness of the proposed method through case studies. Finally, Section 7 concludes the paper.

2 PRELIMINARIES

2.1 Petri Nets

A net is a triple $N = (P, T, F)$, where P is a finite set of places, T is a finite set of transitions, and $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs. In addition, N satisfies $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$. The preset and postset of any $x \subseteq P \cup T$ are defined as ${}^\bullet x = \{y | y \in P \cup T \wedge (y, x) \in F\}$ and $x^\bullet = \{y | y \in P \cup T \wedge (x, y) \in F\}$, respectively. A Petri net is a net N with an initial marking m_0, denoted as $PN = (N, m_0)$. A marking of the net is a function $m : P \to \mathbb{N}$. It can be represented in the form of a multiset or a $|P|$ dimensional vector $\big(m(p_1), \ldots, m(p_{|P|})\big)^T$, where $m(p)$ is the token number in the place $p \in P$.

Let $PN = (P, T, F, m_0)$ be a Petri net, any transition $t \in T$ is called enabled at the marking m if and only if $\forall p \in {}^\bullet t : m(p) > 0$, denoted as $m[t >$. Firing an enabled transition t yields a new marking m' which is denoted as $m[t > m'$ such that $m'(p) = m(p) - 1$ if $p \in {}^\bullet t$, $m'(p) = m(p) + 1$ if $p \in t^\bullet$, otherwise $m'(p) = m(p)$. A sequence of transitions $\sigma = t_1 t_2 \ldots t_k$ is a firable sequence if and only if there is a marking sequence that satisfies $m[t_1 > m_1[t_2 > \cdots > m_{k-1}[t_k > m_k$, which can also be written as $m[\sigma > m_k$. At this point, m_k is reachable from m and can be obtained through the occurrence of σ from m. All reachable markings originating from marking m in net N form a reachable set, represented by $R(N, m)$. A marking $m \in R(N, m_0)$ is a deadlock if $\forall t \in T : \neg m[t >$. In addition, σ^* represents the transition sequence that occurs cyclically.

The structure of a Petri net can be represented by a matrix, which allows for introducing linear algebra methods to analyze. Let $PN = (P, T, F, m_0)$ be a Petri net. $P = \{p_1, p_2, \ldots, p_m\}$, $T = \{t_1, t_2, \ldots, t_n\}$, then net $N = (P, T, F)$ can be represented by an n-row m-column matrix $A = [a_{ij}]_{n \times m}$, where $a_{ij} = a^+_{ij} - a^-_{ij}$, $i \in \{1, 2, \ldots, n\}$, $j \in \{1, 2, \ldots, m\}$. For a^+_{ij}, if $(t_i, p_j) \in F$, then it is 1, otherwise it is 0; For a^-_{ij}, if $(p_j, t_i) \in F$, then it is 1, otherwise it is 0. A is called the incidence matrix of $PN(N)$. If there exists a non trivial n-dimensional non negative integer vector X that satisfies $A^T X = 0$, then X is called a T-invariant of net N. If X_1 is a T-invariant of net N, and any n-dimensional non negative integer vector X satisfying $X < X_1$ is not a T-invariant of net N, then X_1 is called a minimal T-invariant of net N. Let X be a T-invariant of net N. It is denoted that $\|X\| = \{t_i \in T | X(i) > 0\}$, and it is called a support of the T-invariant X. Assuming $\|X\| = T_1$, if any $T_2 \subset T_1$ is not a support of the T-invariant of net N, then T_1 is called the minimal support of the T-invariant of net N.

2.2 LTLK$_{\text{-X}}$

Let $\mathbb{PV}$ be a set of atomic propositions to be interpreted over the global states of a system, $\mathbb{A}$ be a set of agents, $p \in \mathbb{PV}$, $i \in \mathbb{A}$, and $\Gamma \subseteq \mathbb{A}$. LTLK is a logic obtained by combining epistemic operators with LTL. LTLK$_{\text{-X}}$ is the fragment of LTLK without the temporal operator "next", the syntax of which is defined by the following BNF grammar:

$$\varphi \coloneqq p|\neg\varphi|\varphi \wedge \varphi|\varphi U \varphi|K_i\varphi|E_\Gamma\varphi|D_\Gamma\varphi|C_\Gamma\varphi.$$

The temporal operator U denotes the standard "until" modality. Besides, the derived basic temporal modalities are defined as follows: $F\varphi = \mathit{true}\, U\varphi$ and $G\varphi = \neg F\neg\varphi$. Align with the standard definition, F represents the "finally" modality and G represents the "global" modality.

The epistemic operator K_i denotes knowledge of agent i: $K_i\varphi$ means "agent i knows φ". The condition that state g satisfies $K_i\varphi$ is that φ holds for all states that have epistemic accessibility relationship with g. The epistemic operators E_Γ, D_Γ, C_Γ denotes the epistemic situation of the group. Specifically, E_Γ represents "everyone in Γ knows". D_Γ represents distributed knowledge in the group Γ, which is the knowledge obtained by summarizing the knowledge of all agents in Γ and reasoning them together. And C_Γ represents common knowledge among agents in Γ.

Let $KM = (\mathbb{A}, G, G_0, TR, \{\sim_i\}_{i\in\mathbb{A}}, V)$ be a Kripke model [12] associated with a Petri net $PN = (N, m_0)$, where: G is the set of global states, which is $R(N, m_0)$. For the convenience of analysis, $l_i(g)$ represents the local state of agent i in the global state g. G_0 is the set of initial global states, which is $\{m_0\}$. $TR \subseteq G \times G$ is a transfer relationship that represents the temporal evolution of the system. $\{\sim_i\}_{i\in\mathbb{A}} \subseteq G \times G$ is a set of equivalence relationships, one for each agent, encoding epistemic accessibility relationships. For any $i \in \mathbb{A}$, the epistemic accessibility relationship is defined as $(g, g') \in \sim_i$ if and only if $l_i(g) = l_i(g')$. $V \subseteq G \times \mathbb{PV}$ is the evaluation function that labels which atomic propositions are true in a given global state. Let $\pi = g_0 a_0 g_1 \ldots$ be an infinite path of G, and $\pi(i)$ denote the state g_i. Satisfaction of a formula φ in a state g of KM is written as $(KM, g) \models \varphi$, or simplified as $g \models \varphi$. The formal semantics of LTLK$_{\text{-X}}$ is defined inductively as follows:

For $p \in \mathbb{PV}$,

- $g \models p$ iff $p \in V(g)$,
- $g \models \neg\varphi$ iff $g \not\models \varphi$,
- $g \models \varphi \wedge \psi$ iff $g \models \varphi$ and $g \models \psi$,
- $g \models \varphi U \psi$ iff $(\exists k \geq i)(g_k \models \psi \wedge (\forall i \leq j < k) g_i \models \varphi)$,

- $g \models K_i\varphi$ iff $g' \models \varphi$ for every $g' \in G$ such that $g \sim_i g'$,
- $g \models E_\Gamma\varphi$ iff $g' \models \varphi$ for every $g' \in G$ such that $g \left(\bigcup_{i\in\Gamma} \sim_i\right) g'$,
- $g \models D_\Gamma\varphi$ iff $g' \models \varphi$ for every $g' \in G$ such that $g \left(\bigcap_{i\in\Gamma} \sim_i\right) g'$,
- $g \models C_\Gamma\varphi$ iff $g' \models \varphi$ for every $g' \in G$ such that $g \left(\bigcup_{i\in\Gamma} \sim_i\right)^+ g'$, where "+" denotes the transition closure of the relation.

3 MOTIVATION

After analysis, it is believed that the existing Petri net structural reduction methods have the following problems in the context of temporal epistemic logic verification. Firstly, the existing methods are not suitable for verifying temporal epistemic logic. Due to the introduction of epistemic operators in logic, there are more requirements for property satisfaction, and the impact of transitions on properties has correspondingly changed. Therefore, the model obtained based on the original reduction rules cannot guarantee the preservation of temporal epistemic properties. In addition, the Petri net structural reduction rules are limited by the visibility of transitions, and it is typically required that the reduced transitions are invisible. Therefore, when there are massive visible transitions in the net, ideal reduction effects usually cannot be achieved. In response to the above issues, this article conducts work on two aspects accordingly. The specific analysis is as follows.

3.1 The Existing Structural Reduction Theory Not Suitable for LTLK$_{\text{-X}}$ Verification

The definition of the impact of transitions on properties is a pivotal concept in structural reduction theory. In the conventional context of temporal logic, the influence of a transition is solely manifested through its alteration of atomic proposition truth values and transitions exhibiting this characteristic are termed visible. However, with the introduction of epistemic components, the semantic implications of the effect of transitions have evolved. Based on the satisfaction semantics of epistemic operators, determining whether a property holds at a given state requires examining all global states sharing the same local state for the epistemic object with the state. Consequently, property satisfaction becomes dependent on the local state of the agent. In this context, the impact of a transition also encompasses modifications to the local state of the agent.

This semantic shift introduces a critical issue: under existing methods, certain transitions are misclassified as invisible and are thus eligible for reduction. However, models reduced under this assumption may exhibit divergent property satisfaction compared to the original model, leading to erroneous verification results. The following example illustrates this problem. Consider the MAS shown in Figure 1, where the property to be verified is $\varphi_1 = K_1(\neg p_6)$. For this property, the epistemic object is Agent 1, whose initial local state is p_0. According to the satisfaction semantics of the K operator, it is necessary to check whether $\neg p_6$ holds in all global states where the local state of Agent 1 is p_0. The constructed state space (as depicted in Figure 2 a)) reveals the existence of a marking (p_0, p_6), indicating that the original model does not satisfy φ_1.

The effect of transition t_3 on the property is discussed next. Under the conventional definition of transition visibility, t_3 would be considered invisible because it does not alter the token count in place p_6 (i.e., it does not affect truth value of the atomic proposition) and would thus be reduced. However, if t_3 is removed, the reduced model (as depicted in Figure 2 b)) loses the marking (p_0, p_6), leading the model to incorrectly satisfy φ_1. This discrepancy confirms that transitions modifying the local states of epistemic agents can indeed influence the satisfaction of system property. Therefore, the existing concept of transition visibility in temporal logic becomes inadequate in the presence of epistemic components. This insight motivates the first major aspect of this work, which is the modification and extension of existing structural reduction theory to accommodate temporal epistemic logic verification.

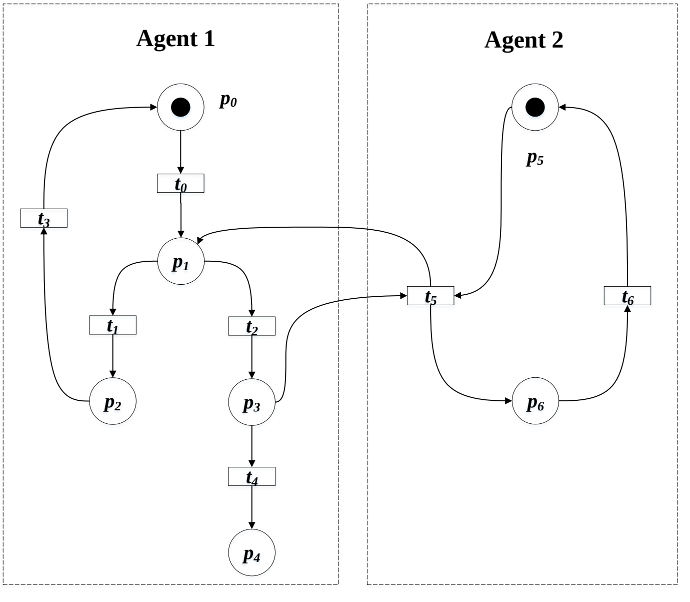


Figure 1. A MAS to illustrate the transition impact on properties in temporal epistemic logic

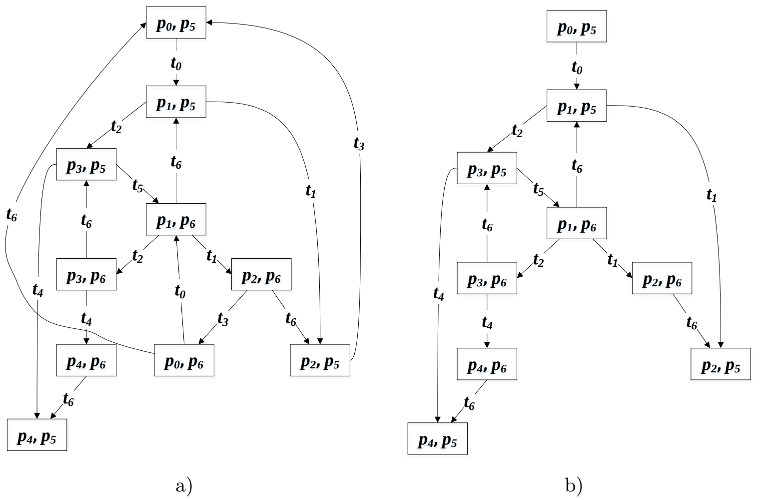


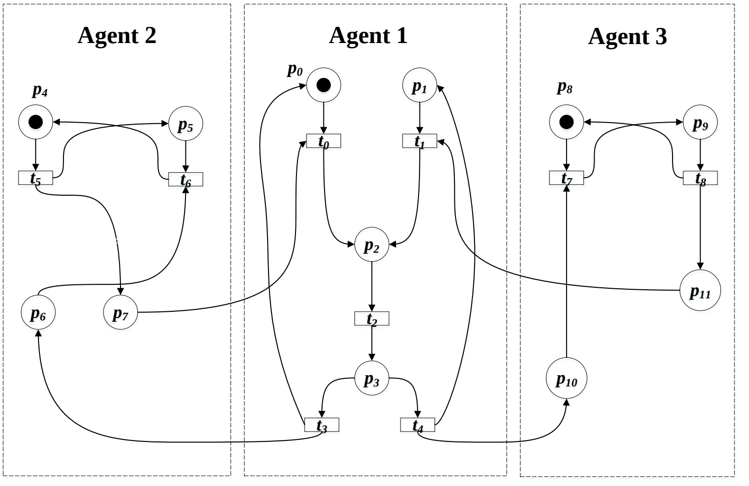
Figure 2. State spaces of the MAS in Figure 1 before and after reducing transition t_3

3.2 Restrictions of Transition Visibility on Existing Structural Reduction Rules

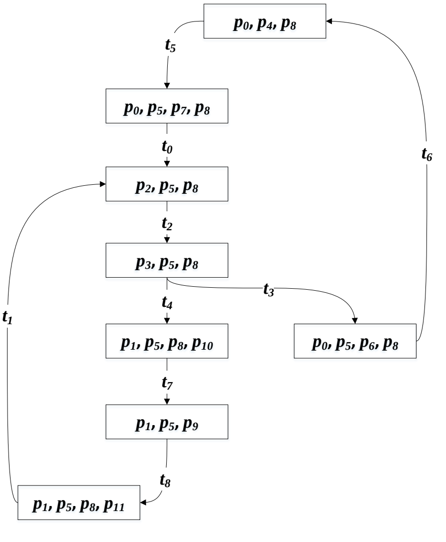
In existing structural reduction theory, transition visibility is a major factor constraining the application of reduction rules. Typically, only invisible transitions are eligible for reduction, as conventional theory relies on the equivalence of visible transition sequences along paths to determine whether two paths are equivalent before and after reduction. The sequences of visible transitions are extracted from the complete transition sequences and compared. If they are identical, the paths are considered equivalent and thus satisfy properties identically; otherwise, they are deemed not equivalent. Consequently, visible transitions cannot be reduced under the paradigm. This requirement limits the effectiveness of reduction when numerous transitions in the net are classified as visible.

In the context of temporal epistemic logic, however, it is observed that changes in the truth values of atomic propositions do not necessarily affect the satisfaction of the overall property. This insight provides a foundation for applying reduction rules even to visible transitions. The following example illustrates this point. The MAS shown in Figure 3 a) is considered, where the property to be verified is $\varphi_2 = K_1 F p_2$. The epistemic object is Agent 1, with an initial local state p_0. According to the satisfaction semantics of the K operator, it is necessary to check whether φ_2 holds in all global states where the local state of Agent 1 is p_0.

Note that t_0 and t_1 both ultimately make the atomic proposition p_2 true and would traditionally be irreducible. However, as shown in the state space in Figure 3 b), the paths corresponding to the transition sequences $(t_5 - t_0 - t_2 - t_3 - t_6)^*$ and $(t_5 - t_0 - t_2 - (t_4 - t_7 - t_8 - t_1 - t_2)^* - t_3 - t_6)^*$ are equivalent with respect to



a)



b)

Figure 3. A MAS to illustrate the occurrences of visible transitions may not necessarily affect the satisfaction of temporal epistemic properties

satisfying φ_2, although their visible transition sequences are different. Crucially, the occurrence of t_1 does not affect the satisfaction of the property, indicating that it can indeed be reduced. This demonstrates that in temporal epistemic logic, some visible transitions do not influence system properties, challenging the conventional rule that visible transitions are irreducible. This observation motivates the second major aspect of this work, which is optimizing the theory by relaxing transition visibility constraints to achieve preferable reduction performance.

4 STRUCTURAL REDUCTION RULES ADAPTED TO LTLK$_{\text{-X}}$

This section describes the first aspect of work in this article, which involves modifying and extending some existing structural reduction rules to make them applicable to temporal epistemic logic. Firstly, the reduction rules are introduced, the application conditions of the rules and the changes in the structure after reduction are provided. Then, it is proved that the paths obtained from the transition sequences before and after reduction satisfies a specific stuttering equivalence relationship, thereby ensuring the correctness of the rules.

4.1 Description of the Reduction Rules

Before describing the rules, the definition of the invisibility of transitions is given under temporal epistemic logic, as this affects the application conditions of the rules.

Definition 1. Let a Petri net $PN = (P, T, F, m_0)$, and the Kripke model $KM = (\mathbb{A}, G, G_0, TR, \{\sim_i\}_{i\in\mathbb{A}}, V)$ obtained from PN, $t \in T$, if for $\forall m, m' \in G$, there is $V(m) = V(m')$ and $m \sim_i m'$ when $m[t > m'$, then t is an E-invisible transition.

The available reduction rules are enumerated below, along with their application conditions, structural transformations after reduction, and intuitive explanations to facilitate understanding.

Rule 1. Pre-agglomeration Rule

The rule is explained through Figure 4. Let a Petri net $PN = (P, T, F, m_0)$, $p \in P$, and the application conditions of the pre-aggregation rule are as follows:

1. ${}^\bullet p \neq \emptyset$, $p^\bullet \neq \emptyset$,
2. $\forall t \in {}^\bullet p$, $t^\bullet = \{p\}$, $p \notin {}^\bullet t$ and t is E-invisible,
3. $m_0(p) = 0$.

By reduction, $PN' = (P', T', F', m_0)$ is obtained, where:

1. $P' = P \setminus \{p\}$,
2. $T' = T \setminus {}^\bullet p \cup ({}^\bullet p \times p^\bullet)$,
3. $F' = (F \cap ((P' \times T') \cup (T' \times P'))) \cup ({}^\bullet h \times (p^\bullet \cap T'))$.

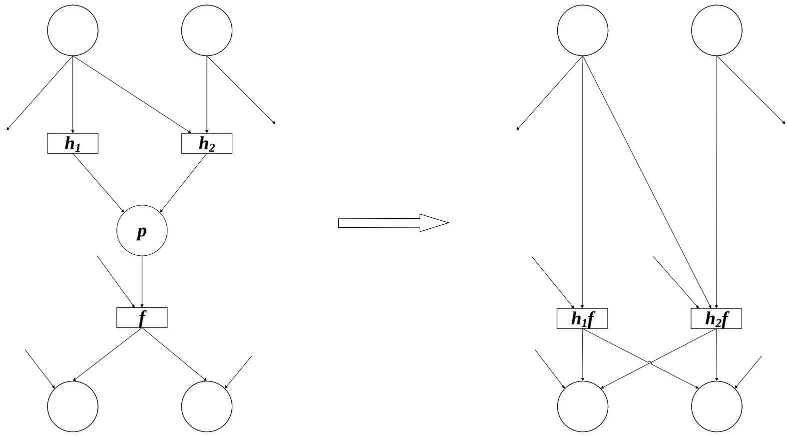


Figure 4. Pre-agglomeration rule

Intuitively, this involves delaying the firing of transitions in H until f meets all its enabling conditions except p, at which point the transitions in H are merged with f and fired together.

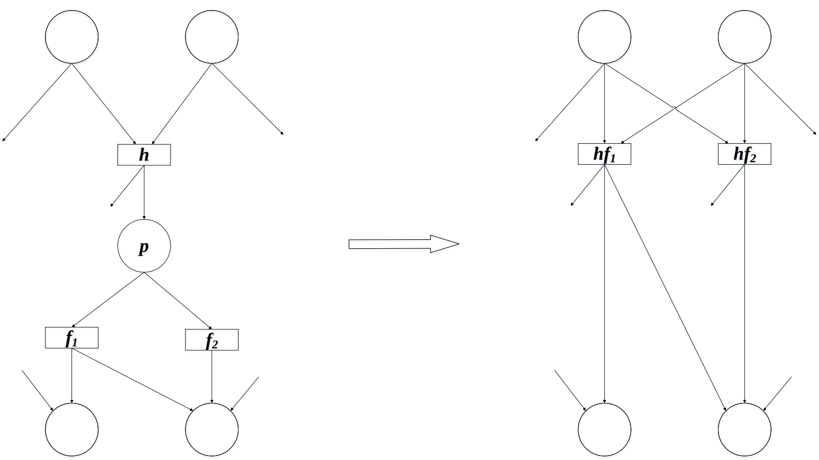


Figure 5. Post-agglomeration rule

Rule 2. Post-agglomeration Rule

The rule is explained through Figure 5. Let a Petri net $PN = (P, T, F, m_0)$, $p \in P$, and the application conditions of the post-aggregation rule are as follows:

1. ${}^{\bullet}p \neq \emptyset$, $p^{\bullet} \neq \emptyset$,

2. $\forall t \in p^\bullet$, ${}^\bullet t = \{p\}$, $p \notin t^\bullet$ and t is E-invisible,
3. $m_0(p) = 0$.

By reduction, $PN' = (P', T', F', m_0)$ is obtained, where:

1. $P' = P \setminus \{p\}$,
2. $T' = T \setminus p^\bullet \cup ({}^\bullet p \times p^\bullet)$,
3. $F' = (F \cap ((P' \times T') \cup (T' \times P'))) \cup (({}^\bullet p \cap T') \times f^\bullet)$.

Intuitively, this rule prepones the firing of transitions in F. Once h occurs, the transitions in F become inevitably enabled, and are thus fused with h and fired collectively.

Rule 3. Pre-reduction Rule

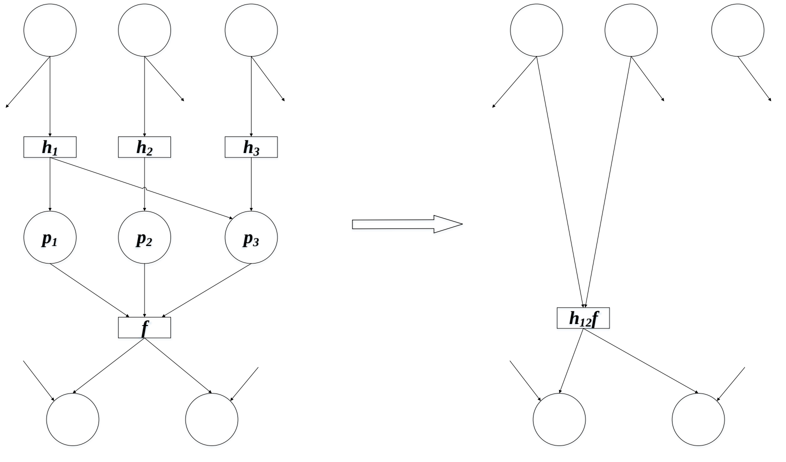


Figure 6. Pre-reduction rule

Before defining this rule, the auxiliary functions *PreTransition*, *PreCovering*, *TtoSeq*, and *SeqtoT* are introduced for calculating the transition set. *PreTransition* is a function with the mapping $T \to 2^T$, and $PreTransition(f) = \{h \in T | h^\bullet \cap {}^\bullet f \neq \emptyset\}$. *PreCovering* is a function with the mapping $T \to 2^{2^T}$, and $PreCovering(f) = \{H \subseteq T | H^\bullet = {}^\bullet f\}$, $\forall h_1, h_2 \in U$, $({}^\bullet h_1 \cap {}^\bullet h_2) \cup (h_1^\bullet \cap h_2^\bullet) = \emptyset$. *TtoSeq* is a function with the mapping $2^T \to T^*$, such that $TtoSeq(H)$ is a transition sequence where each element in H occurs exactly once. *SeqtoT* is a function with the mapping $T^* \to 2^T$, such that $SeqtoT(s)$ is the set of transitions that appear in the transition sequence s. The rule is explained through Figure 6. Let a Petri net $PN = (P, T, F, m_0)$, $p \in P$, and the application conditions of the pre-reduction rule are as follows:

1. ${}^\bullet f \cap f^\bullet = \emptyset$,

2. $\forall h \in PreTransition(f)$, $h^\bullet \subseteq {}^\bullet f$ and h is E-invisible,
3. $\forall p \in {}^\bullet f$, ${}^\bullet p \neq \emptyset$, $p^\bullet = \{f\}$, $m_0(p) = 0$.

By reduction, $PN' = (P', T', F', m_0)$ is obtained, where:

1. $P' = P \setminus {}^\bullet f$,
2. $T' = (T \setminus (PreTransition(f) \cup \{f\})) \cup D$, $D = \{st | H \in PreCovering(f), s = TtoSeq(H)\}$,
3. $F' = (F \cap ((P' \times T') \cup (T' \times P'))) \cup (\bigcup_{h \in D})({}^\bullet(SeqtoT(\text{h}) \setminus \{\text{f}\}) \times \{h\}) \cup (D \times f^\bullet)$.

Intuitively, the firing of H merely enables the firing of f. Consequently, the occurrence of transitions in H can be hidden by immediately redirecting the output tokens of H to the postset of f rather than its preset.

Rule 4. Post-reduction Rule

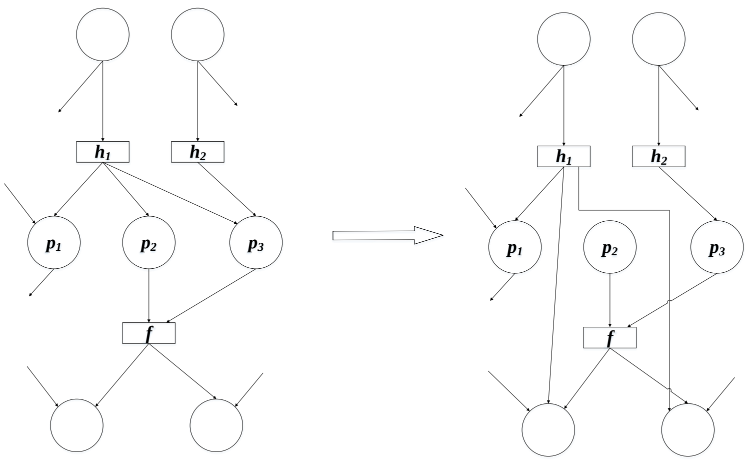


Figure 7. Post-reduction rule

The rule is explained through Figure 7. Let a Petri net $PN = (P, T, F, m_0)$, $h, f \in T$, and the application conditions of the post-reduction rule are as follows:

1. ${}^\bullet f \neq f^\bullet$, ${}^\bullet f \subseteq h^\bullet$,
2. $\forall p \in {}^\bullet t$, $p^\bullet = \{t\}$,
3. h is E-invisible.

By reduction, $PN' = (P', T', F', m_0)$ is obtained, where:

$$F' = (F \setminus (\{h\} \times {}^\bullet f)) \cup (\{h\} \times f^\bullet).$$

Intuitively, the firing of f depends exclusively on the occurrence of h_1 to provide its enabling condition. Therefore, the firing of transitions in f can be accelerated by redirecting a subset of tokens produced by h_1 immediately to the postset of f rather than its preset.

4.2 Correctness of the Reduction Rules

In this section, it is proved that paths satisfying a specific stuttering equivalence relationship are consistent with regard to the satisfaction of temporal epistemic logic. Then, the theorems are given sequentially to prove that the paths obtained from the transition sequences before and after reduction satisfy the specific stuttering equivalence relationship, thereby ensuring the correctness of the rules.

Firstly, the concept of E-stuttering equivalence between paths is introduced.

Definition 2. Given two paths π and π' and $i \in \mathbb{A}$, if there exists a partition $B_1, B_2, \ldots$ for the states on π, as well as a partition $B'_1, B'_2, \ldots$ for the states on π', such that for $\forall j > 0$, B_j and B'_j are nonempty and finite, for each state g on B_j and each state g' on B'_j, $V(g) = V(g')$ and $g \sim_i g'$ hold, then π and π' are E-stuttering equivalent.

The following theorem is given and it is proved that paths satisfying E-stuttering equivalence relationship are consistent with regard to the satisfaction of temporal epistemic logic.

Theorem 1. E-stuttering equivalent paths are indistinguishable with respect to the satisfaction of LTLK$_{\text{-X}}$ properties.

Proof. The transition sequences corresponding to the E-stuttering equivalent paths only differ in the occurrence of E-invisible transitions. Firstly, it is proved that the paths whose corresponding transition sequences differ only by one E-invisible transition satisfy the same properties. Then, the conclusion is induced.

Firstly, it is proved that assuming $\pi = g_0 a_0 g_1 a_1 \ldots$, if a_i is E-invisible, then the necessary and sufficient condition for $(M, \pi_i) \models \varphi$ is $(M, \pi_{i+1}) \models \varphi$. To this end, the structure of property φ is induced. If φ is an atomic proposition, then according to the definition of E-invisibility the conclusion can be obtained directly. The situation of $\wedge$ and $\neg$ is obvious. For the case of $\varphi = \phi U \psi$, it can be obtained through inductive assumptions based on the semantics of U. Next, the case of $\varphi = K_j \psi$ is analyzed. If $(M, \pi_i) \models \varphi$, because a_i is E-invisible, it has that $\pi(i) \sim_j \pi(i+1)$, then $(M, \pi_{i+1}) \models \varphi$. Besides, because $\sim_j$ is an equivalence relationship, it can be obtained that $(M, \pi_{i+1}) \models K_j \psi$. Similarly, from $(M, \pi_{i+1}) \models \varphi$, $(M, \pi_i) \models \varphi$ can be obtained.

By inducing the length of the transition sequence, it can be concluded that the paths of which corresponding transition sequences differ only by one E-invisible transition satisfy the same properties. That is, the paths with E-stuttering equivalence relationship satisfy the same properties, thus the theorem is proved. □

Next, proofs of the correctness of the reduction rules are provided item by item.

Theorem 2. The pre-aggregation rule preserves the satisfaction of the LTLK$_{\text{-X}}$ properties.

Proof. Assuming there exists a sequence P with the prefix $l - h - f$ in the original state space, that is, it has that $m_0[l > m_1[h > m_1'[f > m_2$. According to the application conditions of the pre-aggregation rule, there exists a reduced sequence P_r that fires a transition hf after aggregation at the position where f occurs in the original sequence, then it is obtained that $m_0[l > m_1[hf > m_2'$. Since h is E-invisible, it has that $V(m_1) = V(m_1')$ in the original sequence. The impact of the successive occurrence of h and f as well as the occurrence of aggregation transition hf are same, so there is $m_2 = m_2'$. The satisfaction of atomic proposition in the same state of the reduced state space and that of the original state space are same, thus $V(m_1) = V_r(m_1) = V(m_1')$ and $V(m_2) = V_r(m_2)$. It can be seen that the reduced sequence P_r and the original sequence P are E-stuttering equivalent, therefore the satisfaction of the LTLK$_{\text{-X}}$ property for both are same (Theorem 1). To summarize, the pre-aggregation rule preserves the satisfaction of the LTLK$_{\text{-X}}$ property. □

Theorem 3. The post-aggregation rule preserves the satisfaction of the LTLK$_{\text{-X}}$ properties.

Proof. Assuming there exists a sequence P with the prefix $l - h - f$ in the original state space, that is, it has that $m_0[l > m_1[h > m_1'[f > m_2$. According to the application conditions of the post-aggregation rule, there exists a reduced sequence P_r that fires a transition hf after aggregation at the position where h occurs in the original sequence, then it is obtained that $m_0[l > m_1[hf > m_2'$. Since f is E-invisible, it has that $V(m_1') = V(m_2)$ in the original sequence. The impact of the successive occurrence of h and f as well as the occurrence of aggregation transition hf are same, so there is $m_2 = m_2'$. The satisfaction of atomic proposition in the same state of the reduced state space and that of the original state space are same, thus $V(m_1) = V_r(m_1)$ and $V(m_1') = V(m_2) = V_r(m_2)$. It can be seen that the reduced sequence P_r and the original sequence P are E-stuttering equivalent, therefore the satisfaction of the LTLK$_{\text{-X}}$ property for both are same (Theorem 1). To summarize, the post-aggregation rule preserves the satisfaction of the LTLK$_{\text{-X}}$ property. □

Theorem 4. The pre-reduction rule preserves the satisfaction of the LTLK$_{\text{-X}}$ properties.

Proof. Firstly, prove that if there exists a sequence P with prefix $\sigma.f$ in the original state space, that is, it has that $m_0[\sigma.f > m_1$ where σ does not contain f, then there exists a sequence P' with prefix $\sigma'.h_1 \ldots h_n.f$, that is, it has that $m_0[\sigma'.h_1 \ldots h_n.f > m_1$ where σ' is the occurrence sequence obtained from σ by removing the transitions of $PreCovering(f)$ and $H = \{h_1, h_2, \ldots, h_n\} \in PreCovering(f)\ \star$. Let $\sigma = \sigma_1 - h_n - \sigma_2$, where σ_2 does not contain any transition of $PreTransition(f)$. Since $\forall h \in PreTransition(f)$, $h^\bullet \subseteq {}^\bullet f$ and $\forall p \in {}^\bullet f$, $p^\bullet = \{f\}$, it has that only f consumes tokens produced by transitions of $PreTransition(f)$. Since σ does not contain f, the transitions in σ_2 does not consume the tokens produced by h_n, it has that $m_0[\sigma_1.\sigma_2 >$

$m'[h_n.f > m_1$. By induction it follows that $m_0[\sigma'.h_1 \dots h_n.f > m_f$, where σ' is the transition sequence obtained from σ by removing transitions of *PreTransition*(f), and $H = \{h_1, h_2, \dots, h_n\}$ are the set of transitions of *PreTransition*(f) in σ. Since $\forall p \in {}^\bullet f$, $m_0(p) = 0$, f is not firable in the initial state. After the sequence $h_1 \dots h_n.f$ becomes firable, it can be obtained that ${}^\bullet f \subseteq H^\bullet$. As $\forall h \in$ *PreTransition*(f),$h^\bullet \subseteq {}^\bullet f$, it has that $H^\bullet \subseteq {}^\bullet f$ and thus $H^\bullet = {}^\bullet f$. Since $\forall h_i, h_j \in H$, $({}^\bullet h_i \cap {}^\bullet h_j) \cup (h_i^\bullet \cap h_j^\bullet) = \emptyset$, it has that $H \in$ *PreCovering*(f). Therefore $\star$ is proven. It can be seen that there exists such an original sequence P' that fires $h_1 \dots h_n$ and f successively at the position where f occurs in the original sequence, and this has the same effect as the occurrence of D after applying the pre-reduction rule. Therefore, there exists such a reduced sequence P_r that is E-stuttering equivalent to P'. And since $h_1 \dots h_n$ are E-invisible, P and P' are E-stuttering equivalent. Based on the transitivity of E-stuttering equivalence, it can be obtained that the reduced sequence P_r and the original sequence P are E-stuttering equivalent, therefore the satisfaction of the LTLK$_{\text{-X}}$ property for both are same (Theorem 1). To summarize, the pre-reduction rule preserves the satisfaction of the LTLK$_{\text{-X}}$ property. □

Theorem 5. The post-reduction rule preserves the satisfaction of the LTLK$_{\text{-X}}$ properties.

Proof. Assuming there exists a sequence P with the prefix $\sigma_1.h_1.\sigma_2.f$ in the original state space, that is, it has that $m_0[\sigma_1.h_1 > m_1[\sigma_2 > m_2[f > m_3$. Since ${}^\bullet f \subseteq h_1^\bullet$, $m_0[\sigma_1.h_1 > m_1[f > m_1'$ where $m_1' = (m_1 \setminus {}^\bullet f) \cup f^\bullet$. Since $\forall p \in {}^\bullet f$, $p^\bullet = \{f\}$, that is, only f consumes tokens in places of ${}^\bullet f$, there is no transition on σ_2 that requires the consumption of tokens in ${}^\bullet f$. Thus σ_2 remains fireable after the occurrence of f, it has that $m_0[\sigma_1.h_1 > m_1[f > m_2'[\sigma_2 > m_3$. It can be seen that there exists such an original sequence P' that fires h_1 and f successively at the position where h_1 occurs in the original sequence, and this has the same effect as the occurrence of h_1 after applying the post-reduction rule. Therefore, there exists such a reduced sequence P_r that is E-stuttering equivalent to P'. And since f is E-invisible, P and P' are E-stuttering equivalent. Based on the transitivity of E-stuttering equivalence, it can be obtained that the reduced sequence P_r and the original sequence P are E-stuttering equivalent, therefore the satisfaction of the LTLK$_{\text{-X}}$ property for both are same (Theorem 1). To summarize, the post-reduction rule preserves the satisfaction of the LTLK$_{\text{-X}}$ property. □

5 RESTRICTIONS OF TRANSITION VISIBILITY RELAXED

This section discusses the second aspect of work in this article. From the application conditions of the reduction rules mentioned earlier, it can be seen that the visibility of transitions is a major factor limiting the use of rules. In fact, according to existing methods, ideal reduction results often cannot be achieved when relatively many transitions in the net are judged as visible transitions. Therefore, this article conducts an in-depth analysis of the semantic characteristics of epistemic logic and finds that changes in the truth value of atomic propositions may not necessarily affect the satisfaction of overall properties, which provides support for applying reduction rules for visible transitions. Furthermore, this article relaxes the restrictions of transition visibility when applying rules for a Petri net subclass and a specific property subclass, and ensures the correctness of the analysis through theorem proving.

5.1 Analysis of Temporal Epistemic Logic Semantics

Temporal epistemic logic introduces epistemic operators on the basis of temporal logic. There is a significant difference in determining the satisfaction of path regarding to formulas guided by temporal operators and formulas guided by epistemic operators. Below is a detailed analysis.

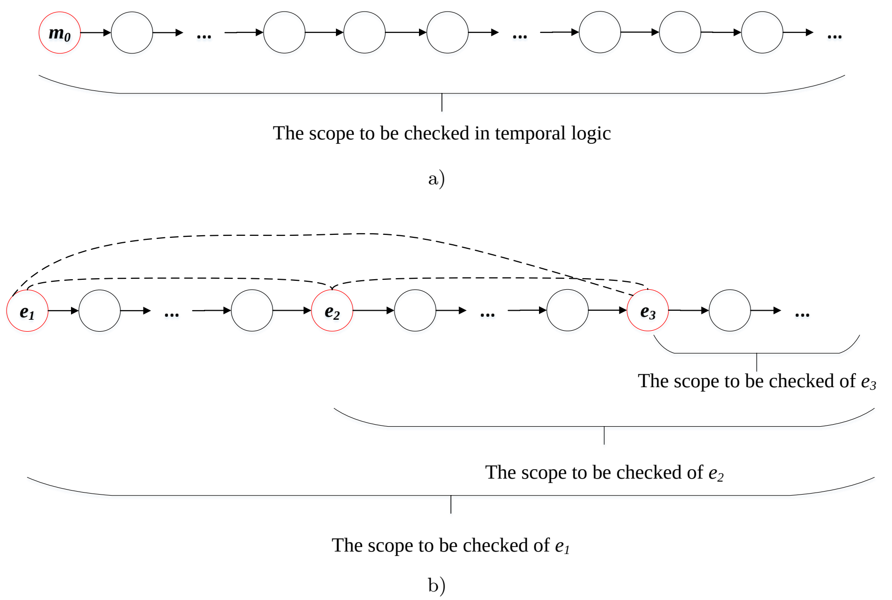


Figure 8. The different scopes to be checked in verification between formulas guided by temporal operators and formulas guided by epistemic operators

Figures 8 a) and 8 b) respectively show the different considerations when determining the satisfaction of the formulas guided by temporal operators and formulas guided by epistemic operators for a path in the state space. The solid line with arrows represents the temporal transfer relationship, while the dashed line represents the epistemic accessibility relationship. From Figure 8 a), it can be seen that for the formula guided by temporal operators, only the entire path starting from the initial state m_0 needs to be checked. For the formula guided by epistemic operators in the form of $\varphi = K_i\psi$, since $g \models K_i\psi$ if and only if for any $g' \in G$ such that $g \sim_i g'$, it has that $g' \models \psi$, thus it is necessary to check the satisfaction of the property ψ on the paths guided by all states that have a epistemic accessibility relationship with the initial state with respect to agent i. For the convenience of analysis, the concept of epistemic accessible state is given.

Definition 3. Given a state m and an agent i, any $m' \in G$ such that $m \sim_i m'$ are called the epistemic accessible states of m with regard to i.

As shown in Figure 8 b), states e_1, e_2, and e_3 are epistemic accessible states of the initial state e_1 with respect to agent i. Therefore, when verifying the formula guided by epistemic operators, the paths guided by these states need to be considered. The overall property φ is satisfied when all these paths satisfy ψ. Based on the reasoning, intuitively speaking, for a formula in the form of $\varphi = K_i\psi$, for a path in the state space, every time an epistemic accessible state appears, the satisfaction of the path guided by the state with the property ψ must be checked. So if there exists such a visible transition that makes ψ true and appears on each path guided by epistemic accessible state, then the overall property φ will be satisfied. Therefore, other visible transitions that make ψ true are not necessary, by reason that the changes in the truth value of the proposition caused by these transitions do not affect the satisfaction of the overall property. The above analysis provides support for applying reduction rules for visible transitions.

5.2 How to Relax the Restrictions of Transition Visibility

Based on previous reasoning, this section focuses on a subclass of Petri nets, which is deadlock-free Petri nets, and a specific property, which is the LTLK$_{\text{-X}}$ property with KF operator as the innermost operator. Intuitively speaking, for multiple visible transitions with the same effect on the innermost proposition that makes it true, it is possible to apply structural reduction rules to some of these transitions under certain conditions. A situation is identified that meets the reduction conditions described in the previous section. A sufficient condition is provided for epistemic properties to be satisfied, and based on this, it is analyzed that which visible transitions can be reduced, thereby relaxing the restrictions of transition visibility.

Definition 4. Given path π, state m, and agent i, let all the epistemic accessible states with regard to m and i which appear on π form a set E, and sort the states in E in the order they appear from front to back. Then, the path corresponding to every two adjacent states on π in E is called an epistemic accessible process about state m and agent i.

Lemma 1. Given path π, state m, and agent i, it is a sufficient condition for π satisfying the property of $\varphi = K_iF\psi$ that there exists a transition t that makes ψ true on each epistemic accessible process of π with respect to m and i.

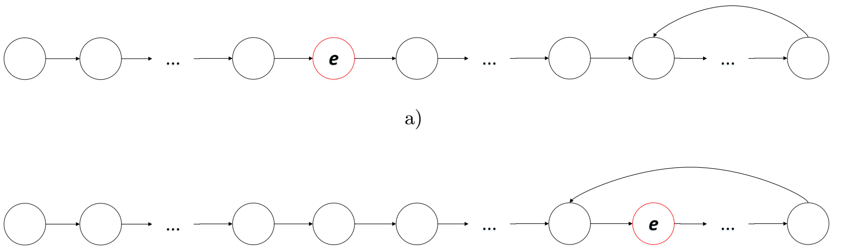
Proof. If there exists a transition t that makes ψ true on each epistemic accessible process of π with respect to m and i, then there exists at least one transition t that makes ψ true on each path guided by epistemic accessible states that appears on π with respect to m and i. Therefore, these paths all satisfy the property of $F\psi$. According to the semantics of satisfaction of the K operator, it can be inferred that π satisfies the property of $\varphi = K_i F\psi$, thus it is proven. □

On the basis of dividing transitions into visible and invisible transitions, it is possible to further classify visible transitions that have the same effect. If there are multiple transitions that make ψ true in the checked system, including transitions that satisfy the sufficient condition stated in Lemma 1, then only retaining these transitions during verification can preserve the system properties, and such transitions are called key transitions. While other transitions that make ψ true can be excluded, which are called non-key transitions.

After generating the complete state space, it is natural to identify key transitions according to the definition, but this has no meaning for space reduction. So the next discussion is on how to determine key and non-key transitions without generating the state space. Given agent i, any two epistemically accessible states about i in the state space have equal local states of i, so the epistemically accessible process between these two states definitely corresponds to a loop. In order to satisfy the sufficient condition in Lemma 1, that is, there exist key transitions that make ψ true on each epistemic accessible process, it is needed that the key transitions occur on the loop corresponding to the epistemic accessible process. In the structural analysis theory of Petri nets, T-invariants correspond to loops. It can be concluded that the key transitions are bound to occur on the same support of minimal T-invariants of the net as the transitions to reach the target epistemically accessible state. Hence, a method for determining key transitions can be provided and formalized to obtain the following theorem.

Theorem 6. Let the set consisting of transitions with the same effect on the innermost proposition that make it true be called *APT*, and the set consisting of transitions to reach the target epistemic accessible state be called *EST*. Let a deadlock-free Petri net *PN*. For any support T_1 of the minimal T-invariants of *PN* that contains *EST*, if T_1 also contains the true subset A_1 of APT, and there is no any other support of the minimal T-invariant of *PN* that contains elements from $APT \setminus A_1$ and *EST*, then A_1 is the key transition set and $APT \setminus A_1$ is the non-key transition set. After applying structural reduction rules to non-key transitions, the LTLK$_{\text{-X}}$ property with *KF* operator as the innermost operator will be preserved.

Proof. Following it is only needed to prove that for any path, the non-key transitions have no influence on the LTLK$_{\text{-X}}$ property with *KF* operator as the innermost operator, regardless of where it occurs on the path. According to Definition 4, any path is divided by its epistemic accessible states, and the part between every two adjacent epistemic accessible states is an epistemic accessible process. Every epistemic accessible process inevitably involves key transitions, so the appearance of non-key transitions does not change the satisfaction of a given property. It can



a)

b)

Figure 9. Possible scenarios for generated paths in deadlock-free Petri nets

be inferred that the occurrence of key transitions in the sequence guided by the last epistemic accessible state is the only uncertainty. Since PN is deadlock-free, any path of the state space obtained from PN will end in a loop. Below is a classification discussion based on the relationship between the last epistemic accessible state and the loop at the end (end loop for short), that is, whether the end loop includes epistemic accessible states.

1. The end loop does not include epistemic accessible states. At this point, the last epistemic accessible state is outside the end loop, as shown in Figure 9 a), where e represents the last epistemic accessible state. If there is no non-key transition in the sequence of transitions after the last epistemic accessible state, it is evident that non-key transitions have no impact on the given property. Below is the analysis of there existing non-key transitions in the sequence after the last epistemic accessible state. The path through the last epistemic accessible state could have formed a loop that includes key transitions and EST without non-key transitions, but due to entering other loops, a complete epistemic accessible process is not formed. And it is known that there must exist a key transition in the sequence between EST and non-key transition, which causes the innermost property satisfied. Therefore, the appearance of non-key transitions does not affect the given property.

2. The end loop includes epistemic accessible states. At this point, the last epistemic accessible state is within the end loop, as shown in Figure 9 b), where e represents the last epistemic accessible state. Due to the fact that the end loop inevitably includes transitions corresponding to the support of minimal T-invariants of PN that contains EST, which also contains key transitions, the satisfaction of the property is determined by the key transitions, and the occurrence of non-key transitions does not change the satisfaction of a given property.

To summarize, the theorem is proven. □

6 CASE STUDY

To evaluate the method proposed in this article, two cases are picked out for study. The first case is the online ride-hailing vehicle environment monitoring platform, through which not only the effectiveness of the method is demonstrated, but also a detailed explanation is provided of how to relax the restrictions of transition visibility based on Theorem 6. The second case is the dining cryptographer protocol [45], which has a much larger scale, and the analysis results on grand case can further enhance the persuasiveness of the proposed method.

6.1 Online Ride-Hailing Vehicle Environment Monitoring Platform

This section selects the online ride-hailing vehicle environment monitoring platform for case analysis. The MAS includes three agents: a comprehensive perception system, a safety officer end, and a passenger end. The comprehensive perception system perceives, identifies, and models the surrounding environment of the vehicle through devices such as cameras and radars. The front and rear rows of the vehicle provide software for safety officers and passengers, respectively, allowing them to view the surrounding environment. The view provided to the safety officer is more accurate, representing the target through a cube of occupied space, while the view provided to the passenger is more simplified and visually friendly, displaying the target according to category. The original Petri net model of the system is shown in Figure 10, where the meaning of the transitions is listed in Table 1. The property that needs to be guaranteed is $\varphi_3 = K_1 F p_9$, which means that the safety officer knows that the comprehensive perception system will model the environment.

Agent	**Transition**	**Meaning**
Agent 1	t_1	Monitoring vehicle status
	t_2	Providing target cube illustrations for safety officer
	t_3	Generating driving suggestions
Agent 2	t_4	Activating the perception system
	t_5	Recording the passenger end display to the main station
	t_6	Safety officer end requesting environment modeling
	t_7	Passenger end requesting environment modeling
	t_8	Identifying targets in the environment
	t_9	Sending precise environment model to safety officer end
	t_{10}	Sending simplified environment model to passenger end
Agent 3	t_{11}	Providing target category illustrations for passengers
	t_{12}	Submitting passenger experience feedback

Table 1. The meanings of transitions in the original model of the online ride-hailing vehicle environment monitoring platform

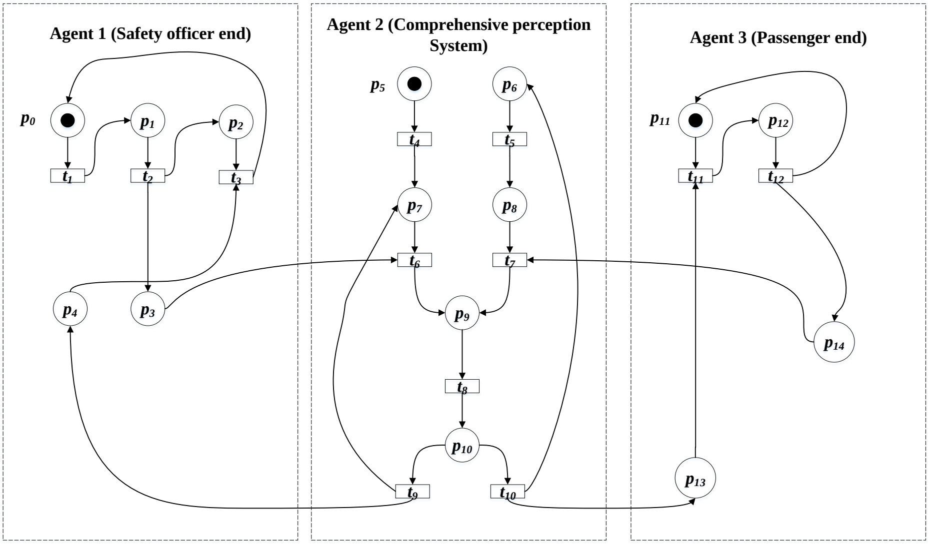


Figure 10. The original model of online ride-hailing vehicle environment monitoring platform

The reduced models obtained by applying structural reduction rules to the original model without relaxing the transition visibility constraint and with relaxing the transition visibility constraint are shown in Figures 11 and 12, respectively. Specifically, since t_7 is a visible transition, according to the previous theory, t_7 cannot be reduced. However, in fact, excluding t_7 will not affect the given property. The following describes how to use Theorem 6 to guide the reduction of visible transitions. Firstly, it is calculated that the set APT which is $\{t_6, t_7\}$ including transitions with the same effect on the innermost proposition and the transition set EST which is $\{t_3\}$ including transitions to reach the target epistemic accessible state. Then all the minimal T-invariants of the net and corresponding supports are calculated, the calculation method of which derives from [46]. It is found that there is a support $\{T_1\}$ ($\{t_1, t_2, t_3, t_6, t_8, t_9\}$) of the minimal T-invariants that contains the true subset of APT A_1 ($\{t_6\}$) and EST, and there is no any other support that contains the elements from $APT \setminus A_1$ and EST. Thus the element (t_7) in $APT \setminus A_1$ can be considered invisible, and reduction rules can be applied to it. The changes in the count of net elements and states after reduction are shown in Table 2, where Reduced model 1 and Reduced model 2 represents reduced model without and with relaxing transition visibility restrictions, respectively. It can be seen that the structural reduction method proposed in this article can simplify the model to a certain extent, reduce the state storage space, and the theory based on relaxing the restrictions of transition visibility can achieve preferable reduction results.

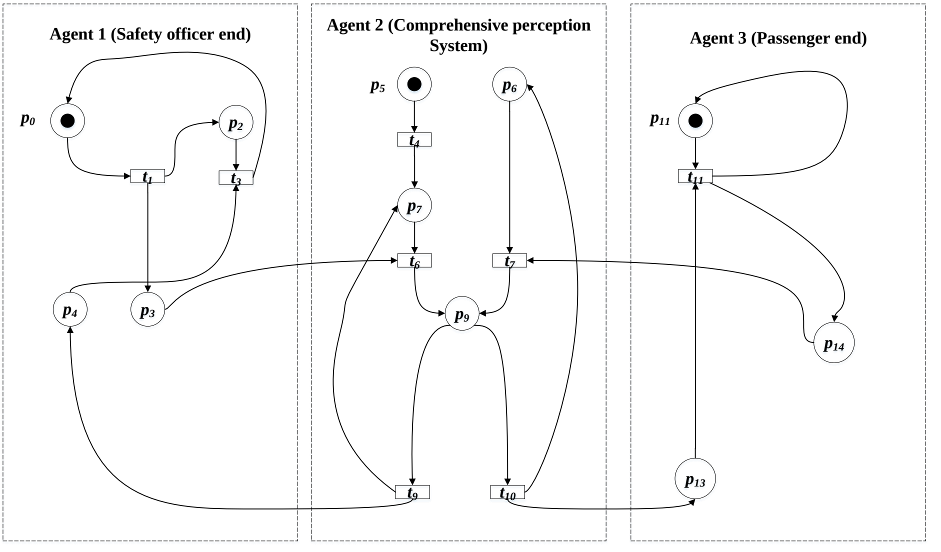


Figure 11. The reduced model without relaxing transition visibility restrictions of online ride-hailing vehicle environment monitoring platform

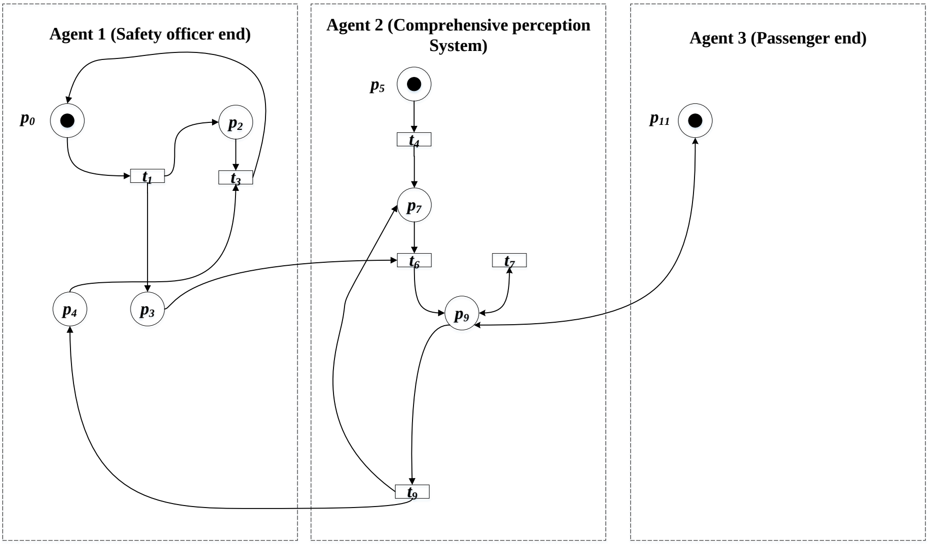


Figure 12. The reduced model with relaxing transition visibility restrictions of online ride-hailing vehicle environment monitoring platform

Indicator	**Original Model**	**Reduced Model 1**	**Reduced Model 2**
Place	15	11	8
Transition	12	8	6
Arc	32	24	18
State	15	10	8

Table 2. Changes in the count of model net elements and states of the platform before and after reduction of the online ride-hailing vehicle environment monitoring platform

6.2 Dining Cryptographer Protocol

According to the dining cryptographer protocol [45], cryptographers wonder whether one of them paid or their boss paid. It is assumed that at most one cryptographer paid for the dining. Each cryptographer tosses a coin between him and the cryptographer on his left, and only these two cryptographers can see whether the coin is positive or negative. Next, each cryptographer announces whether the side of their own coin is same as the coin' side of their right neighbor. There is an exception that if a cryptographer pays the bill, he will declare the opposite of the fact as seen. When there are an odd number of "difference" in the declaration, it indicates that one of the cryptographers paid for it. While when there are an even number of "difference", it indicates that the bill was paid by the boss. The original Petri net model of the system with 3 cryptographers is shown in Figure 13. The property to be verified is that $\varphi_4 = K_1 F p_{24}$, which means that the first cryptographer will know that the second cryptographer has made a declaration.

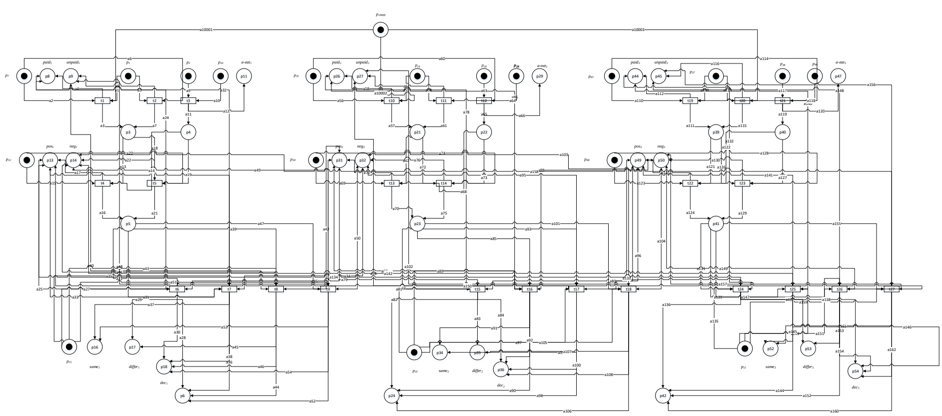


Figure 13. The original model of dining cryptographer protocol with 3 cryptographers

The reduced models obtained by applying structural reduction rules to the original model without relaxing the transition visibility constraint and with relaxing the transition visibility constraint are shown in Figures 14 and 15, respectively. The changes in the count of net elements and states after reduction are shown in Table 3.

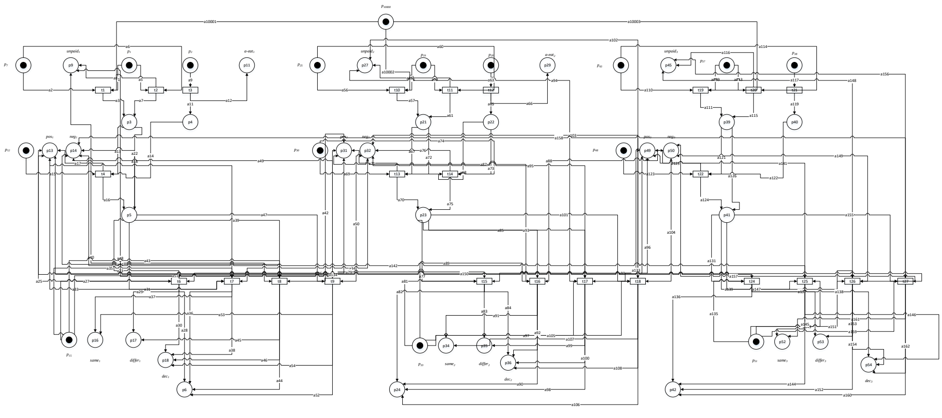


Figure 14. The reduced model without relaxing transition visibility restrictions of dining cryptographer protocol with 3 cryptographers

Specifically, the number of states was reduced by approximately 8.59 % and 15.15 %, before and after relaxing the transition visibility constraints, respectively. And the state reduction rate is increased by 6.56 % after relaxing the restrictions. It can be seen that the structural reduction method proposed in this article can simplify the model to a certain extent, reduce the state storage space, and the theory based on relaxing the restrictions of transition visibility can achieve better reduction results. The analysis results on larger examples further enhance the persuasiveness of the proposed method.

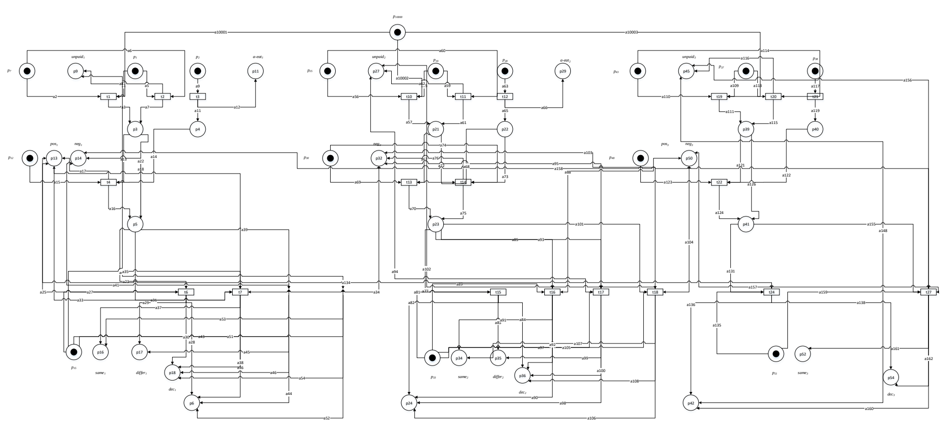


Figure 15. The reduced model with relaxing transition visibility restrictions of dining cryptographer protocol with 3 cryptographers

Indicator	Original Model	Reduced Model 1	Reduced Model 2
Place	55	48	45
Transition	27	25	21
Arc	201	185	171
State	792	724	672
Reduction rate	/	8.59 %	15.15 %

Table 3. Changes in the count of model net elements and states of the platform before and after reduction of the dining cryptographer protocol

7 CONCLUSION

To address the state explosion problem faced by existing model checking methods for MAS temporal epistemic logic, this paper proposes a Petri net structural reduction method that preserves temporal epistemic logic properties. The research is mainly conducted from two aspects. Firstly, due to the fact that the existing Petri net structural reduction methods are not suitable for verifying temporal epistemic logic, this paper modifies and extends some existing structural reduction rules to make them applicable to temporal epistemic logic, and ensures the correctness of the rules through theorem proving. Furthermore, considering that the Petri net structural reduction rules are limited by the visibility of transitions, this paper focuses on a Petri net subclass and a specific property subclass, conducts a precise analysis of transition visibility based on the semantic characteristics of epistemic logic, relaxes the restrictions of transition visibility when applying rules, and ensures the correctness of the analysis through theorem proving. Finally, it is demonstrated through case studies that the reduction rules are capable of saving space, and the reduction effect will be better after relaxing the restrictions of transition visibility.

In future research, we will continue to delve deeper into the semantic characteristics of epistemic logic, further relaxing the restrictions of transition visibility on reduction, and applying various reduction methods, including structural reduction to larger Petri net subclasses and broader properties.

Acknowledgement

This work is supported by the Special Fund of Fundamental Scientific Research Business Expense for the Higher School of Central Government under Grant No. 221202-40563.

REFERENCES

[1] Wooldridge, M.: An Introduction to Multiagent Systems. John Wiley & Sons, 2009, https://dl.acm.org/doi/abs/10.5555/1695886.

[2] Ghezelbash, E.—Yazdanpanah, M. J.—Asadpour, M.—Yaghmaei, A.: Optimal Selection of Informed Agents for Influence Opposition. IEEE Transactions on Computational Social Systems, Vol. 8, 2021, No. 1, pp. 20–32, doi: 10.1109/TCSS.2020.3043295.

[3] Min, H.—Cao, J.—Ge, J.—Liu, B.: A Multi-Agent System for Fine-Grained Opinion Dynamics Analysis in Online Social Networks. IEEE Transactions on Computational Social Systems, Vol. 11, 2024, No. 1, pp. 815–828, doi: 10.1109/TCSS.2022.3219036.

[4] Ezekiel, J.—Lomuscio, A.—Molnar, L.—Veres, S. M.—Pebody, M.: Verifying Fault Tolerance and Self-Diagnosability of an Autonomous Underwater Vehicle. Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence – Volume Two (IJCAI'11), 2011, pp. 1659–1664, doi: 10.5591/978-1-57735-516-8/IJCAI11-279.

[5] Lavaei, A.—Di Lillo, L.—Censi, A.—Frazzoli, E.: Formal Estimation of Collision Risks for Autonomous Vehicles: A Compositional Data-Driven Approach. IEEE Transactions on Control of Network Systems, Vol. 10, 2023, No. 1, pp. 407–418, doi: 10.1109/TCNS.2022.3203363.

[6] Gürsoy, F.—Badur, B.: An Agent-Based Modeling Approach to Brain Drain. IEEE Transactions on Computational Social Systems, Vol. 9, 2022, No. 2, pp. 356–365, doi: 10.1109/TCSS.2021.3066074.

[7] Trivedi, A.—Rao, S.: Agent-Based Modeling of Emergency Evacuations Considering Human Panic Behavior. IEEE Transactions on Computational Social Systems, Vol. 5, 2018, No. 1, pp. 277–288, doi: 10.1109/TCSS.2017.2783332.

[8] He, L.—Liu, G.—Zhou, M.: Petri-Net-Based Model Checking for Privacy-Critical Multiagent Systems. IEEE Transactions on Computational Social Systems, Vol. 10, 2023, No. 2, pp. 563–576, doi: 10.1109/TCSS.2022.3164052.

[9] Kacprzak, M.—Lomuscio, A.—Niewiadomski, A.—Penczek, W.—Raimondi, F.—Szreter, M.: Comparing BDD and SAT Based Techniques for Model Checking Chaum's Dining Cryptographers Protocol. Fundamenta Informaticae, Vol. 72, 2006, No. 1-3, pp. 215–234, `https://journals.sagepub.com/doi/abs/10.3233/FUN-2006-721-316`.

[10] Liu, G. P.: Coordinated Control of Networked Nonlinear Multiagent Systems Using Variable Horizon Learning Predictors via Cloud Edge Computing. IEEE Transactions on Control of Network Systems, Vol. 9, 2022, No. 4, pp. 1975–1986, doi: 10.1109/TCNS.2022.3181549.

[11] Lomuscio, A.—Qu, H.—Solanki, M.: Towards Verifying Contract Regulated Service Composition. Autonomous Agents and Multi-Agent Systems, Vol. 24, 2012, No. 3, pp. 345–373, doi: 10.1007/s10458-010-9152-3.

[12] Clarke Jr, E. M.—Grumberg, O.—Kroening, D.—Peled, D.—Veith, H.: Model Checking. MIT Press, 2018, `https://mitpress.mit.edu/9780262038836/model-checking/`.

[13] Van der Meyden, R.—Suf, K.: Symbolic Model Checking the Knowledge of the Dining Cryptographers. Proceedings. 17th IEEE Computer Security Foundations Workshop (CSFW 04), 2004, pp. 280–291, doi: 10.1109/CSFW.2004.1310747.

[14] Halpern, J. Y.—Vardi, M. Y.: Model Checking vs. Theorem Proving: A Manifesto. In: Lifschitz, V. (Ed.): Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy. Academic Press, 1991, pp. 151–176, doi: 10.1016/B978-0-12-450010-5.50015-3.

[15] Fagin, R.—Halpern, J. Y.—Moses, Y.—Vardi, M. Y.: Reasoning About Knowledge. The MIT Press, 2004, doi: 10.7551/mitpress/5803.001.0001.

[16] van der Meyden, R.—Shilov, N. V.: Model Checking Knowledge and Time in Systems with Perfect Recall. In: Rangan, C. P., Raman, V., Ramanujam, R. (Eds.): Foundations of Software Technology and Theoretical Computer Science (FSTTCS 1999). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1738, 1999, pp. 432–445, doi: 10.1007/3-540-46691-6_35.

[17] van der Hoek, W.—Wooldridge, M.: Model Checking Knowledge and Time. In: Bošnački, D., Leue, S. (Eds.): Model Checking Software (SPIN 2002). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 2318, 2002, pp. 95–111, doi: 10.1007/3-540-46017-9_9.

[18] Gammie, P.—van der Meyden, R.: MCK: Model Checking the Logic of Knowledge. In: Alur, R., Peled, D. A. (Eds.): Computer Aided Verification (CAV 2004). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3114, 2004, pp. 479–483, doi: 10.1007/978-3-540-27813-9_41.

[19] Huang, X.—Luo, C.—van der Meyden, R.: Improved Bounded Model Checking for a Fair Branching-Time Temporal Epistemic Logic. In: van der Meyden, R., Smaus, J. G. (Eds.): Model Checking and Artificial Intelligence (MoChArt 2010). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6572, 2010, pp. 95–111, doi: 10.1007/978-3-642-20674-0_7.

[20] Kacprzak, M.—Nabiałek, W.—Niewiadomski, A.—Penczek, W.—Półrola, A.—Szreter, M.—Woźna, B.—Zbrzezny, A.: VerICS 2007 – A Model Checker for Knowledge and Real-Time. Fundamenta Informaticae, Vol. 85, 2008, No. 1-4, pp. 313–328, doi: 10.3233/FUN-2008-851-422.

[21] Lomuscio, A.—Pirovano, E.: Parameterised Verification of Strategic Properties in Probabilistic Multi-Agent Systems. Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '20), 2020, pp. 762–770, `https://ifaamas.org/Proceedings/aamas2020/pdfs/p762.pdf`.

[22] Lomuscio, A.—Qu, H.—Raimondi, F.: MCMAS: An Open-Source Model Checker for the Verification of Multi-Agent Systems. International Journal on Software Tools for Technology Transfer, Vol. 19, 2017, No. 1, pp. 9–30, doi: 10.1007/s10009-015-0378-x.

[23] Męski, A.—Penczek, W.—Szreter, M.—Woźna-Szcześniak, B.—Zbrzezny, A.: BDD- Versus SAT-Based Bounded Model Checking for the Existential Fragment of Linear Temporal Logic with Knowledge: Algorithms and Their Performance. Autonomous Agents and Multi-Agent Systems, Vol. 28, 2014, No. 4, pp. 558–604, doi: 10.1007/s10458-013-9232-2.

[24] Su, K.—Sattar, A.—Luo, X.: Model Checking Temporal Logics of Knowledge via OBDDs. The Computer Journal, Vol. 50, 2007, No. 4, pp. 403–420, doi: 10.1093/comjnl/bxm009.

[25] Celaya, J. R.—Desrochers, A. A.—Graves, R. J.: Modeling and Analysis of Multi-Agent Systems Using Petri Nets. 2007 IEEE International Conference on Systems, Man and Cybernetics, 2007, pp. 1439–1444, doi: 10.1109/ICSMC.2007.4413960.

[26] Ding, Z.—Yang, R.—Cui, P.—Zhou, M.—Jiang, C.: Variable Petri Nets for Mobility. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 52, 2022, No. 8, pp. 4784–4797, doi: 10.1109/TSMC.2021.3103072.

[27] He, L.—Liu, G.: Model Checking CTLK Based on Knowledge-Oriented Petri Nets. 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/Smartcity/DSS), 2019, pp. 1139–1146, doi: 10.1109/HPCC/SmartCity/DSS.2019.00161.

[28] Kotb, Y. T.—Beauchemin, S. S.—Barron, J. L.: Petri Net-Based Cooperation in Multi-Agent Systems. Fourth Canadian Conference on Computer and Robot Vision (CRV'07), IEEE, 2007, pp. 123–130, doi: 10.1109/CRV.2007.49.

[29] Kouah, S.—Saïdouni, D. E.—Ilié, J. M.: Synchronized Petri Net: A Formal Specification Model for Multi Agent Systems. Journal of Software, Vol. 8, 2013, No. 3, pp. 587–602, doi: 10.4304/jsw.8.3.587-602.

[30] Liu, G.—Jiang, C.: Petri Net Based Model Checking for the Collaborative-Ness of Multiple Processes Systems. 2016 IEEE 13th International Conference on Networking, Sensing, and Control (ICNSC), 2016, pp. 1–6, doi: 10.1109/ICNSC.2016.7478980.

[31] Lopes, B.—Benevides, M.—Haeusler, E. H.: Reasoning About Multi-Agent Systems Using Stochastic Petri Nets. In: Bajo, J., Hernández, J. Z., Mathieu, P. et al. (Eds.): Model Checking and Artificial Intelligence (MoChArt 2010). Springer, Cham, Advances in Intelligent Systems and Computing, Vol. 372, 2015, pp. 75–86, doi: 10.1007/978-3-319-19629-9_9.

[32] Miyamoto, T.—Horiguchi, K.: Modular Reachability Analysis of Petri Nets for Multiagent Systems. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 43, 2013, No. 6, pp. 1411–1423, doi: 10.1109/TSMC.2013.2250278.

[33] Yang, R.—Ding, Z.—Guo, T.—Pan, M.—Jiang, C.: Model Checking of Variable Petri Nets by Using the Kripke Structure. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 52, 2022, No. 12, pp. 7774–7786, doi: 10.1109/TSMC.2022.3163741.

[34] Zhao, F.—Xiang, D.—Liu, G.—Jiang, C.: A New Method for Measuring the Behavioral Consistency Degree of WF-Net Systems. IEEE Transactions on Computational Social Systems, Vol. 9, 2022, No. 2, pp. 480–493, doi: 10.1109/TCSS.2021.3099475.

[35] Berthelot, G.: Checking Properties of Nets Using Transformations. In: Rozenberg, G. (Ed.): Advances in Petri Nets 1985 (APN 1985). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 222, 1986, pp. 19–40, doi: 10.1007/BFb0016204.

[36] Poitrenaud, D.—Pradat-Peyre, J. F.: Pre- and Post-Agglomerations for LTL Model Checking. In: Nielsen, M., Simpson, D. (Eds.): Application and Theory of Petri Nets 2000 (ICATPN 2000). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1825, 2000, pp. 387–408, doi: 10.1007/3-540-44988-4_22.

[37] Esparza, J.—Schröter, C.: Net Reductions for LTL Model-Checking. In: Nielsen, M., Simpson, D. (Eds.): Correct Hardware Design and Verification Methods (CHARME 2001). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 2144, 2001, pp. 310–324, doi: 10.1007/3-540-44798-9_25.

[38] Shi, Y.—Duan, Z.—Tian, C.—Yang, H.: Improved Net Reductions for LTL\X Model Checking. In: Liu, S., Duan, Z. (Eds.): Structured Object-Oriented Formal Language and Method (SOFL+MSVL 2013). Springer, Cham, Lecture Notes in Computer Science, Vol. 8332, 2013, pp. 48–61, doi: 10.1007/978-3-319-04915-1_4.

[39] Evangelista, S.—Haddad, S.—Pradat-Peyre, J. F.: Syntactical Colored Petri Nets Reductions. In: Peled, D. A., Tsay, Y. K. (Eds.): Automated Technology for Verification and Analysis (ATVA 2005). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3707, 2005, pp. 202–216, doi: 10.1007/11562948_17.

[40] Haddad, S.—Pradat-Peyre, J. F.: New Efficient Petri Nets Reductions for Parallel Programs Verification. Parallel Processing Letters, Vol. 16, 2006, No. 1, pp. 101–116, doi: 10.1142/s0129626406002502.

[41] Jensen, J. F.—Nielsen, T.—Oestergaard, L. K.—Srba, J.: TAPAAL and Reachability Analysis of P/T Nets. In: Koutny, M., Desel, J., Kleijn, J. (Eds.): Transactions on Petri Nets and Other Models of Concurrency XI. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 9930, 2016, pp. 307–318, doi: 10.1007/978-3-662-53401-4_16.

[42] Bønneland, F. M.—Dyhr, J.—Jensen, P. G.—Johannsen, M.—Srba, J.: Stubborn Versus Structural Reductions for Petri Nets. Journal of Logical and Algebraic Methods in Programming, Vol. 102, 2019, pp. 46–63, doi: 10.1016/j.jlamp.2018.09.002.

[43] Berthomieu, B.—Le Botlan, D.—Dal Zilio, S.: Counting Petri Net Markings from Reduction Equations. International Journal on Software Tools for Technology Transfer, Vol. 22, 2020, No. 2, pp. 163–181, doi: 10.1007/s10009-019-00519-1.

[44] Thierry-Mieg, Y.: Structural Reductions Revisited. In: Janicki, R., Sidorova, N., Chatain, T. (Eds.): Applications and Theory of Petri Nets and Concurrency (PETRI NETS 2020). Springer, Cham, Lecture Notes in Computer Science, Vol. 12152, 2020, pp. 303–323, doi: 10.1007/978-3-030-51831-8_15.

[45] Chaum, D.: The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability. Journal of Cryptology, Vol. 1, 1988, No. 1, pp. 65–75, doi: 10.1007/BF00206326.

[46] Lin, C.—Chanson, S. T.—Murata, T.: Petri Net Models and Efficient T-Invariant Analysis for Logic Inference of Clauses. 1996 IEEE International Conference on Systems, Man and Cybernetics, Vol. 4, 1996, pp. 3174–3179, doi: 10.1109/ICSMC.1996.561494.

Tong Guo received his B.Sc. degree in software engineering from the Shandong University of Science and Technology, Qingdao, China, in 2018. He is currently pursuing his Ph.D. degree in computer science and technology with the Department of Computer Science and Technology, Tongji University, Shanghai, China. His current research interests consist of formal engineering, Petri nets and multi-agent systems.

Meiqin Pan received her Ph.D. degree from the Shandong University of Science and Technology, Qingdao, China, in 2008. Now she is Associate Professor of the School of Business and Management, Shanghai International Studies University. Her research interests are in information systems, data mining and technology, optimization methods. She has published more than 20 papers in domestic and international academic journals and conference proceedings.

Zhijun Ding received his M.Sc. degree from the Shandong University of Science and Technology, Taian, China, in 2001, and his Ph.D. degree from the Tongji University, Shanghai, China, in 2007. Now he is Professor of the Department of Computer Science and Technology, Tongji University. He has published more than 100 papers in domestic and international academic journals and conference proceedings. His research interests are in formal engineering, Petri nets, services computing, and workflows.