









定义 3. 假定  $Agt=\{1,\dots,k\}$  为智能体集合,  $AP$  为非空原子命题集合,  $IS$  是一个多元组:

$$J \triangleq (\langle Q_i, Ac_i, \lambda_i, \delta_i, Q_i^0 \rangle_{i \in Agt}, L).$$

其中,

- $Q_i$  是智能体  $i$  的局部状态集合. 所有智能体的局部状态集合的积  $Q_1 \times Q_2 \times \dots \times Q_k$  为系统的全局状态集合, 用  $Q$  表示系统的全局状态集合. 对于任意一个全局状态  $g \in Q$ ,  $g(i)$  表示在全局状态  $g$  中, 智能体  $i$  的局部状态;
- $Ac_i$  是智能体  $i$  可选的非空有限动作集. 所有智能体动作集合的积  $Ac = Ac_1 \times \dots \times Ac_k$  被称为系统的联合动作集. 联合动作  $\alpha = \langle a_1, \dots, a_k \rangle \in Ac_1 \times \dots \times Ac_k$  表示,  $\alpha(i)$  表示智能体  $i$  的动作;
- $\lambda_i: Q_i \rightarrow 2^{Ac_i}$  是智能体  $i$  的动作规范, 给定局部状态  $q$ ,  $\lambda_i(q)$  表示智能体  $i$  在局部状态  $q$  时所有可选择的动作集, 定义  $\lambda_i(g) = \lambda_i(g(i))$ . 这表明智能体的行为是受限的, 可选动作与其当前所处的局部状态有关. 对于一个给定的全局状态  $g$ , 以  $\lambda(g)$  表示集合  $\lambda_1(g) \times \dots \times \lambda_k(g)$ ;
- $\delta_i: Q_i \times Ac \rightarrow Q_i$  是智能体  $i$  的状态转移函数, 给定局部状态  $q$  和联合动作  $\alpha$  满足  $\alpha(i) \in \lambda_i(q)$ ,  $\delta_i(q, \alpha)$  是  $i$  在下一时刻的局部状态. 定义  $\delta$  为全局状态迁移关系, 满足  $\delta(q_1, \dots, q_k, \alpha) = (\delta_1(q_1, \alpha), \dots, \delta_k(q_k, \alpha))$ ;
- $Q_i^0 \subseteq Q_i$  是智能体  $i$  的初始状态集合, 本文用  $Q_0$  表示全局初始状态集合  $Q_1^0 \times \dots \times Q_k^0$ ;
- $L: AP \rightarrow 2^{\mathcal{P}}$  是系统的标签函数, 对于任意的全局状态  $g$  和原子命题  $p, g \in L(p)$  表示  $p$  命题在全局状态  $g$  时为真.

从解释系统  $J$ , 可以推导出 CGS 系统  $(Q, Q_0, Agt, Ac', (\sim_i)_{i \in Agt}, \lambda', \delta, L)$ , 其中,

- $Q, Q_0, Agt, \delta$  和  $L$  如定义 3 所述;
- $Ac' = Ac_1 \cup \dots \cup Ac_k$ ;
- $\lambda'$  定义为  $\lambda'(g, i) = \lambda_i(g(i))$ ;
- $\sim_i$  定义为:  $g \sim_i g'$  当且仅当  $g(i) = g'(i)$ .

$J$  的路径(path)是无限全局状态序列  $\pi = g_0 g_1 g_2 \dots$ , 其中, 对任意  $m \geq 0$ , 存在联合动作  $\alpha_m \in Ac$ , 满足  $g_{m+1} = \delta(g_m, \alpha_m)$ .

给定一条路径  $\pi = g_0 g_1 g_2 \dots$  和  $j \geq 0$ , 以  $\pi(j)$  表示序列中第  $j$  处的全局状态  $g_j$ , 并用  $proj_i(\pi)$  代表智能体  $i$  在  $\pi$  中的分量, 即局部状态序列  $g_0(i) g_1(i) g_2(i) \dots$ .

有限路径  $\tau = g_0 g_1 \dots g_n$  是路径  $\pi = g_0 g_1 g_2 \dots$  的前缀. 给定有限路径  $\tau = g_0 g_1 \dots g_n$  和  $0 \leq m \leq n$ , 以  $\tau(m)$  表示序列中第  $m$  处的全局状态  $g_m$ ,  $lst(\tau)$  是  $\tau$  的最后一个状态,  $proj_i(\tau)$  代表智能体  $i$  在  $\tau$  中的分量, 即局部状态序列  $g_0(i) g_1(i) \dots g_n(i)$ .

下文将以  $Paths$  表示所有可能的路径集, 以  $Trks$  表示所有可能有限路径的集. 对于给定的全局状态  $g$ , 以  $Paths(g)$  和  $Trks(g)$  表示所有从  $g$  出发的路径和有限路径集合.

## 2.4 策略

在  $IS$  中, 智能体  $i$  在某一状态下可能存在多个可以选择的动作, 这种不确定性采用策略(strategy)  $\theta_i$  来决定. 而策略根据智能体是否可以观察到其他智能体的状态和是否依赖于历史状态有不同的定义方式. 本文采用文献[6]的定义方式, 定义 4 种不同策略类型:  $IR, Ir, iR$  和  $ir$ , 具体定义如下.

- $IR. \theta_i: Trks \rightarrow Ac_i$ : 对于任何一个智能体  $i$  以及任意一条有限路径  $\tau$ , 满足  $\theta_i(\tau) \in \lambda_i(lst(\tau))$ , 即, 策略函数  $\theta_i$  为智能体  $i$  定义了从有限路径到可选动作的映射函数;
- $Ir. \theta_i: Trks \rightarrow Ac_i$ : 对两条有限路径  $\tau_1$  和  $\tau_2$  满足  $lst(\tau_1) = lst(\tau_2)$ , 那么智能体  $i$  将选择相同的动作, 即  $\theta_i(\tau_1) = \theta_i(\tau_2)$ . 因此, 智能体  $i$  选择的动作只依赖系统当前的状态, 不依赖于历史有限路径;
- $iR. \theta_i: Trks \rightarrow Ac_i$ : 类似于  $IR$  策略, 不同的是: 对于两条有限路径  $\tau_1$  和  $\tau_2$ , 如果  $proj_i(\tau_1) = proj_i(\tau_2)$ , 则  $\theta_i(\tau_1) = \theta_i(\tau_2)$ . 直观地讲,  $iR$  策略表示智能体  $i$  在任何两条无法区分的有限路径上都选择同一个动作;
- $ir. \theta_i: Trks \rightarrow Ac_i$ :  $ir$  融合了  $iR$  和  $Ir$  的所有特点, 在  $ir$  中, 如果  $lst(\tau_1)(i) = lst(\tau_2)(i)$ , 则  $\theta_i(\tau_1) = \theta_i(\tau_2)$ . 即, 智能体  $i$  仅能根据当前自身的局部状态来判断选择一个动作.

给定智能体集合  $A$ ,  $\bar{A}$  表示其补集,  $A$  的联合  $\sigma$ -策略就是  $A$  中所有智能体的  $\sigma$ -策略构成的集合, 即  $\theta_A = \{\theta_i | i \in A\}$ , 其中,  $\sigma \in \{IR, Ir, iR, ir\}$ , 本文用  $\theta_A(i)$  表示智能体  $i$  在  $\theta_A$  中的策略.

给定智能体集  $A$ 、全局状态  $g$  以及联合  $\sigma$ -策略  $\theta_A$ , 以  $Out_{\sigma}(g, \theta_A)$  表示  $A$  中的所有智能体严格按照策略  $\theta_A$  选择动作, 从初识状态  $g$  开始, 系统所有可能路径的集合, 即: 对于任意  $\pi \in Out_{\sigma}(g, \theta_A)$  都有  $\pi(0)=g$ , 以及任意时刻  $j \geq 0$ , 存在联合动作  $\alpha \in \lambda(\pi(j))$ , 满足  $\pi(j+1)=\delta(\pi(j), \alpha)$  和对于所有智能体  $i \in A, \alpha(i)=\theta_A(i)(\pi(0) \dots \pi(j))$  成立.

## 2.5 交替时态逻辑ATL

交替时态逻辑(alternating-time temporal logic, 简称 ATL)是 CTL 的一种扩展<sup>[5]</sup>, 将存在路径量词  $\exists$  替换为联合模态操作  $\langle\langle A \rangle\rangle$ , 其中  $A$  为一个智能体集合.

定义 4. ATL 的语法如下:  $\varphi := p \mid \neg \varphi \mid \varphi_1 \vee \varphi_2 \mid \langle\langle A \rangle\rangle X \varphi \mid \langle\langle A \rangle\rangle G \varphi \mid \langle\langle A \rangle\rangle \varphi_1 U \varphi_2$ . 其中,  $A \subseteq Agt$ .

定义 5. 给定解释系统  $J = (\langle\langle Q_i, Ac_i, \lambda_i, \delta_i, Q_i^0 \rangle\rangle_{i \in Agt}, L)$ 、全局状态  $g$  和 ATL 公式  $\varphi$ , 公式的可满足性  $J, g \models_{\sigma} \varphi$  通过如下所示归纳定义.

- $J, g \models_{\sigma} p$  当且仅当  $g \in L(p)$ ;
- $J, g \models_{\sigma} \neg \varphi$  当且仅当  $J, g \not\models_{\sigma} \varphi$ ;
- $J, g \models_{\sigma} \varphi_1 \vee \varphi_2$  当且仅当  $J, g \models_{\sigma} \varphi_1$  或  $J, g \models_{\sigma} \varphi_2$ ;
- $J, g \models_{\sigma} \langle\langle A \rangle\rangle X \varphi$  当且仅当存在一个联合  $\sigma$ -策略  $\theta_A$ , 对于所有路径  $\pi \in Out_{\sigma}(g, \theta_A)$ , 都有  $J, \pi(1) \models_{\sigma} \varphi$ ;
- $J, g \models_{\sigma} \langle\langle A \rangle\rangle G \varphi$  当且仅当存在一个联合  $\sigma$ -策略  $\theta_A$ , 对于所有路径  $\pi \in Out_{\sigma}(g, \theta_A)$ , 在任何时刻  $j \geq 0$ , 都有  $J, \pi(j) \models_{\sigma} \varphi$ ;
- $J, g \models_{\sigma} \langle\langle A \rangle\rangle \varphi_1 U \varphi_2$  当且仅当存在一个联合  $\sigma$ -策略  $\theta_A$ , 对于所有路径  $\pi \in Out_{\sigma}(g, \theta_A)$ , 都存在一个时刻  $j \geq 0$ , 使得  $J, \pi(j) \models_{\sigma} \varphi_2$  成立; 并且对于所有时刻  $0 \leq i < j$ , 都有  $J, \pi(i) \models_{\sigma} \varphi_1$  成立.

解释系统  $J$  满足公式  $\varphi$  当且仅当所有  $J$  中的全局初始状态都满足  $\varphi$ . 类似的, 可以定义 ATL 在 CGS 系统上的语义, 详见文献[5].

给定公式  $\varphi$ , 在某语义下, 所有智能体采用同一个策略类型, 即称为智能体的策略类型是一致的; 反之为不一致的. 考虑公式  $\langle\langle \{1, 2, 3\} \rangle\rangle (p \ U \langle\langle \{1, 2\} \rangle\rangle X \varphi)$ , 在  $IR$  语义时是一致的; 而在其他语义时, 由于智能体 3 的策略类型在满足  $p \ U \langle\langle \{1, 2\} \rangle\rangle X \varphi$  和  $\varphi$  时可能会不同, 因此是不一致的.

计算树逻辑是 ATL 逻辑的一种特殊子类, 其中, 所有  $\langle\langle A \rangle\rangle \varphi$  公式中的  $A$  必须是空集  $\emptyset$ , 即,  $\langle\langle \emptyset \rangle\rangle$  表示所有路径量词. CTL 在 Kripke 结构的语义见文献[1].

定理 1. 给定 IS 系统  $J = (\langle\langle Q_i, Ac_i, \lambda_i, \delta_i, Q_i^0 \rangle\rangle_{i \in Agt}, L)$ 、全局状态  $g$  和 ATL 公式  $\varphi$ , 以下结论成立.

- (1)  $J, g \models_{iR} \varphi$  问题是不可判定的;
- (2)  $J, g \models_{iR} \varphi$  和  $J, g \models_{Ir} \varphi$  问题是 PTIME-完备的<sup>[5, 10]</sup>.
- (3)  $J, g \models_{iR} \varphi$  当且仅当  $J, g \models_{Ir} \varphi^{[10]}$ .
- (4)  $J, g \models_{iR} \varphi$  问题是  $P^{NP}$ -完备的<sup>[6, 8]</sup>.

证明: Dima and Tiplea 证明了 ATL 在 CGS 上的模型检查问题在  $iR$  场景下是不可判定的<sup>[7]</sup>, 其将图灵机停机问题规约到一个仅包含 3 个智能体的 CGS 系统  $G$  满足 ATL 公式  $\langle\langle \{1, 2\} \rangle\rangle G \text{ ok}$  的问题, 因此只需要将该 CGS 系统转化到一个 IS 系统  $J$ , 使得  $J, g \models_{iR} \langle\langle \{1, 2\} \rangle\rangle G \text{ ok}$  当且仅当  $G$  中的状态  $g$  在  $iR$  场景满足  $\langle\langle \{1, 2\} \rangle\rangle G \text{ ok}$ .

考虑 CGS 系统  $G = (Q, Q_0, Agt, Ac, (\sim_i)_{i \in Agt}, \lambda, \delta, L)$ , 对于任意智能体  $i$ , 假设  $Q_i = \{Q_i^1, \dots, Q_i^{n_i}\}$  为关系  $\sim_i$  在  $Q$  上的等价类集, 以  $|g|_i$  表示状态  $g$  在  $Q_i$  中的等价类. 对于公式  $\langle\langle \{1, 2\} \rangle\rangle G \text{ ok}$  而言, 智能体 3 的等价类  $Q_3$  不影响其可满足性, 因此  $\sim_3$  可以替换为最小等价关系, 即  $g \sim_3 g'$  当且仅当  $g = g'$ . 本文假设  $\sim_3$  为最小等价关系.

给定状态  $g$  和动作组  $\alpha = \langle a_1, a_2, a_3 \rangle$ , 以  $\alpha_g$  表示动作组  $\alpha = \langle a_1, a_2, (a_3, g) \rangle$ .

定义 IS 系统  $J = (\langle\langle Q_i, Ac_i, \lambda_i, \delta_i, Q_i^0 \rangle\rangle_{i \in Agt}, L)$ , 其中,

- $Ac_1 = Ac_2 = Ac, Ac_3 = Ac \times Q$ ;
- 对于任意的  $i, \delta_i(|g|_i, \alpha_g) = |g'|_i$  当且仅当  $\delta(g, \alpha) = g'$ ;
- $\lambda_i(|g'|_i) = \{\alpha_g \mid \alpha \in \lambda(g, i), g \in |g'|_i\}$ ;

- $Q_i^0 = \{ | g | | g \in Q \}$ .

对于公式  $\langle\langle\{1,2\}\rangle\rangle G ok$ , 只有智能体 1 和智能体 2 需要考虑联合  $iR$ -策略, 而智能体 3 是采用  $IR$ -策略. 根据 ATL 语义,  $J, g \models_{iR} \langle\langle\{1,2\}\rangle\rangle G ok$  当且仅当  $G$  的状态  $g$  在  $iR$  场景满足  $\langle\langle\{1,2\}\rangle\rangle G ok$ .

2.6 示 例

考虑系统模型  $J = (\langle\langle Q_i, Ac_i, \lambda_i, \delta_i, Q_i^0 \rangle\rangle_{i \in Agt}, L)$ , 其中,

- $Agt = \{A, B\}$ ;
- $Q_A = \{A_1, A_2\}, Q_B = \{B_1, B_2, B_3\}$ ;
- $Ac_A = \{a_1, a_2\}, Ac_B = \{b_1, b_2\}$ ;
- $\delta_A$  和  $\delta_B$  如图 1 所示, 节点表示局部状态, 边表示状态迁移关系, 边上的标识是智能体的联合动作;
- $\lambda_A = \{A_1: \{a_1, a_2\}, A_2: \{a_2\}\}, \lambda_B = \{B_1: \{b_1\}, B_2: \{b_1, b_2\}, B_3: \{b_1\}\}$ ;
- $Q_A^0 = \{A_1\}, Q_B^0 = \{B_1\}$ ;
- $L(eva) = \{(A_2, B_2)\}$ .

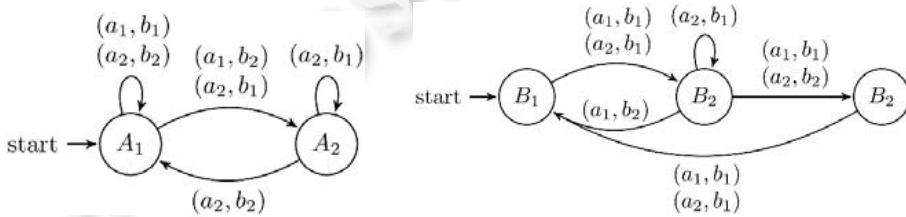


Fig.1 Local transition graphs of A and B

图 1 智能体 A 和智能体 B 的局部状态转移图

示例的全局状态转移关系如图 2 所示.

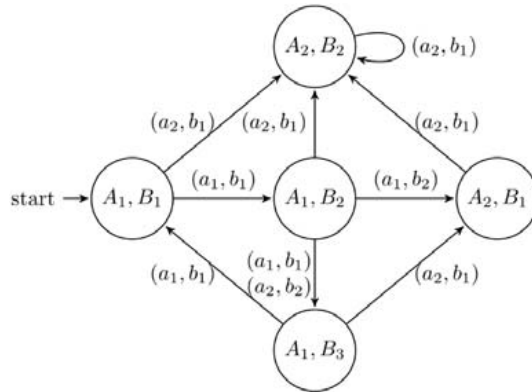


Fig.2 Transition relation of system

图 2 全局状态迁移关系

考虑 ATL 公式:  $\varphi = \langle\langle\{B\}\rangle\rangle True U eva$ . 不难发现,

- 若智能体 A 在状态  $(A_1, B_1)$  时选择动作  $a_1$ , 系统将进入状态  $(A_1, B_2)$ ;
- 在状态  $(A_1, B_2)$  时, 如果 B 选择动作  $b_1$ , A 选择动作  $a_1$ ; 如果 B 选择动作  $b_2$ , A 选择动作  $a_2$ , 系统将进入状态  $(A_1, B_3)$ ; 若 A 继续选择动作  $a_1$ , 系统重回状态  $(A_1, B_1)$ .

如此反复, 系统将永远不会到达状态  $(A_2, B_2)$ . 因此, 对于任意的  $\sigma \in \{IR, Ir, iR, ir\}, J, (A_1, B_1) \models_{\sigma} \varphi$  都不成立.

但是在实际应用中,比如在无线传感器网络中,每个节点是一个智能体,其存储资源非常有限;同时,为了降低能耗,不可能时刻跟其他智能体保持通信来获取其他智能体的局部状态.因此在上述例子中,如果将  $A$  的策略类型限制为  $ir$ ,在状态  $(A_1, B_2)$  时,  $A$  只能选择动作  $a_1$ ;如果  $B$  选择动作  $b_2$ ,系统最终将到达  $(A_2, B_2)$ .因此,智能体的策略类型对公式可满足性的结果有直接的影响.而在经典的语义下,无法对此类规约进行描述和验证;同时,CGS 和 IS 都无法对这类异构多智能体系统建模.针对这个问题,本文将在解释系统中引入智能体策略类型函数来描述智能体的策略类型,从而验证上述类型的规约.

### 3 基于策略类型的解释系统

通过讨论可知,传统的多智能体系统验证框架在语义表述方面仍然存在不足.因此提出基于策略类型的解释系统,为每个智能体引入专属的策略类型.

#### 3.1 TIS模型

**定义 6.** 基于策略类型的解释系统(typed interpreted system,简称 TIS)是二元组  $T \triangleq (J, A)$ ,其中,

- $J = (\langle Q_i, Ac_i, \lambda_i, \delta_i, Q_i^0 \rangle_{i \in Agt}, L)$  是一个解释系统;
- $A: Agt \rightarrow \{IR, Ir, iR, ir\}$  是一个映射函数,为每个智能体关联一种策略类型.

TIS 依旧沿用传统解释系统中对路径和有限路径的定义,但是需要重新定义策略.与已有工作通过语义来定义智能体的策略不同,在 TIS 模型中,智能体的策略类型由 TIS 模型给出,如,  $A(i)$  表示智能体  $i$  只能采用  $A(i)$ -策略.

给定一个智能体集合  $A$ ,  $A$  的联合策略就是  $A$  中所有智能体的策略构成的集合,即:

$$\theta_A \triangleq \{ \theta_i | i \in A, \theta_i \text{ 是 } A(i)\text{-策略} \}.$$

本文用  $\theta_A(i)$  表示智能体  $i$  在  $\theta_A$  中的策略,以  $S_A$  表示  $A$  所有可能的联合策略集合.已知智能体集合  $A$  和初始状态  $g$ ,两个联合策略  $\theta_A$  和  $\theta_{\bar{A}}$  唯一地确定了一条路径  $Path(\theta_A, \theta_{\bar{A}}) \triangleq \pi$ , 满足  $\pi(0) = g$ , 以及任意一个时刻  $j$ , 都存在联合动作  $\alpha \in \lambda(\pi(j))$ , 使得  $\pi(j+1) = \delta(\pi(j), \alpha)$ , 其中: 对于任意  $i \in A$ ,  $\alpha(i) = \theta_A(i)(\pi(0) \dots \pi(j))$ ; 对于任意  $i \in \bar{A}$ ,  $\alpha(i) = \theta_{\bar{A}}(i)(\pi(0) \dots \pi(j))$ . 给定智能体集合  $A$ 、初始状态  $g$ 、一个联合策略  $\theta_A$ , 定义:

$$Out(g, \theta_A) \triangleq \bigcup_{\theta_{\bar{A}} \in S_{\bar{A}}} Path(\theta_A, \theta_{\bar{A}}).$$

**事实 1.** 给定 TIS  $T = (J, A)$  和智能体集合  $A$ , 对于任意  $\sigma \in \{IR, Ir, iR, ir\}$ , 如果所有  $i \in A$ ,  $A(i) = \sigma$ , 则以下结论成立.

- 1)  $\theta_A$  是  $J$  中  $A$  的一个联合  $\sigma$ -策略当且仅当  $\theta_A$  是  $T$  中  $A$  的联合策略;
- 2) 对于  $T$  中  $A$  的任意一个联合策略  $\theta_A$ ,  $Out(g, \theta_A) \subseteq Out_{\sigma}(g, \theta_A)$ ;
- 3) 当所有  $i \in \bar{A}$  满足  $A(i) = IR$  时, 对于任意一个  $T$  中  $A$  的联合策略  $\theta_A$ ,  $Out(g, \theta_A) = Out_{\sigma}(g, \theta_A)$ .

证明: 根据联合  $\sigma$ -策略和联合策略的定义, 结论 1) 显而易见.

$Out(g, \theta_A) \subseteq Out_{\sigma}(g, \theta_A)$ : 对于任意路径  $\pi \in Out(g, \theta_A)$ , 证明  $\pi \in Out_{\sigma}(g, \theta_A)$  成立即可. 假设一条路径  $\pi = g_0 g_1 \dots$ , 其中,  $g_0 = g$ , 即: 对于智能体集  $\bar{A}$ , 始终存在联合策略  $\theta_{\bar{A}} = \{ \theta_i | i \in \bar{A} \} \in S_{\bar{A}}$ , 任意时刻  $m \geq 0$ ,  $g_{m+1} = \delta(g_m, \alpha_m)$  成立. 其中,  $\alpha_m$  是所有智能体的联合动作, 即: 对于智能体  $i \in A$ , 有  $\alpha_m(i) = \theta_A(i)(g_0 \dots g_m)$  成立; 对于智能体  $i \in \bar{A}$ , 有  $\alpha_m(i) = \theta_{\bar{A}}(i)(g_0 \dots g_m)$  成立. 显然, 通过以上方法找到的路径, 一定存在于  $Out_{\sigma}(g, \theta_A)$  中. 因此, 结论 2) 成立.

$Out(g, \theta_A) \supseteq Out_{\sigma}(g, \theta_A)$ : 证明对于任意路径  $\pi \in Out_{\sigma}(g, \theta_A)$ , 都有  $\pi \in Out(g, \theta_A)$  成立. 设有条路径  $\pi = g_0 g_1 \dots$ , 其中,  $g_0 = g$ , 且在任意  $m \geq 0$ ,  $g_{m+1} = \delta(g_m, \alpha_m)$  成立; 同时, 对于智能体  $i \in A$ ,  $\alpha_m(i) = \theta_A(i)(g_0 \dots g_m)$  成立. 此时, 在任意  $m \geq 0$ , 智能体  $i \in \bar{A}$ , 我们令  $\theta_{\bar{A}}(i)(g_0 \dots g_m) = \alpha_m(i)$ . 对于智能体  $i \in \bar{A}$ , 由于  $A(i) = IR$ , 则有  $\theta_{\bar{A}} \in S_{\bar{A}}$ , 使得  $\pi \in Out(g, \theta_A)$  成立. 综上, 结论 3) 成立.

**定义 7.** 给定一个 TIS  $T$ 、全局状态  $g$  和 ATL 公式  $\varphi$ , 则可满足性  $T, g \models \varphi$  通过如下归纳定义.

- $T, g \models p$  当且仅当  $g \in L(p)$ ;
- $T, g \models \neg \varphi$  当且仅当  $T, g \not\models \varphi$ ;



- $T, g \models \varphi_1 \vee \varphi_2$  当且仅当  $T, g \models \varphi_1$  或  $T, g \models \varphi_2$ ;
- $T, g \models \langle\langle A \rangle\rangle X \varphi$  当且仅当存在联合策略  $\theta_A$ , 对于任意路径  $\pi \in \text{Out}(g, \theta_A)$ , 都有  $T, \pi(1) \models \varphi$ ;
- $T, g \models \langle\langle A \rangle\rangle G \varphi$  当且仅当存在联合策略  $\theta_A$ , 对于任意路径  $\pi \in \text{Out}(g, \theta_A)$ , 在任何时刻  $j \geq 0$ , 都有  $T, \pi(j) \models \varphi$ ;
- $T, g \models \langle\langle A \rangle\rangle \varphi_1 U \varphi_2$  当且仅当存在联合策略  $\theta_A$ , 对于所有路径  $\pi \in \text{Out}(g, \theta_A)$ , 都存在一个时刻  $j \geq 0$ , 使得  $T, \pi(j) \models \varphi_2$  成立, 并且对于所有时刻  $0 \leq i < j$ , 都有  $T, \pi(i) \models \varphi_1$ .

**定理 2.** ATL 在 TIS 上的模型检查问题是不可判定的.

证明: 根据定理 1 的证明, 可以把图灵机停机问题规约到一个 IS  $J$  是否满足  $\langle\langle \{1,2\} \rangle\rangle G \text{ok}$  的问题上. 而从  $J$  模型可以构造 TIS 系统  $T=(J, A)$ , 其中,  $A$  满足  $A(1)=A(2)=ir, A(3)=IR$ , 从而将图灵机停机问题规约到 ATL 在 TIS 模型检查问题.  $\square$

由于定理 2, 只需要考虑 TIS 系统  $T=(J, A)$  满足  $A: \text{Agt} \rightarrow \{IR, Ir, ir\}$  的模型检查问题. 本文研究满足  $A: \text{Agt} \rightarrow \{IR, Ir, ir\}$  的 TIS 模型检查问题, 但是 ATL 公式中出现的智能体必须是  $Ir$  或  $ir$  类型, 即, 所有 ATL 公式中出现的  $\langle\langle A \rangle\rangle \varphi$  公式满足: 如果  $i \in A$ , 则  $A(i) \in \{Ir, ir\}$ . 对于一般情况, 目前作者还没有解决方案, 留待未来解决.

### 3.2 模型检查算法

对于任何一个智能体, 如果其策略类型为  $Ir$  或  $ir$ , 那么其策略数量是有限的. 因此, 可以通过不确定地猜测每个智能体的策略, 然后再验证猜测是否正确. 给定 TIS 系统  $T$  和 ATL 公式  $\varphi$ , 并且公式  $\varphi$  中出现的智能体是  $Ir$  或  $ir$  类型,  $\|\varphi\|_T$  表示所有满足公式  $\varphi$  的状态集合. 在给出具体算法前, 先引入一些基本定义.

给定智能体集合  $A$  和策略类型函数  $A: \text{Agt} \rightarrow \{IR, Ir, ir\}$ ,  $A_r^A$  表示智能体集合  $A$  的子集并满足:  $i \in A_r^A$  当且仅当  $A(i) \in \{Ir, ir\}$ .  $F_A^A$  表示  $f: A_r^A \times Q \rightarrow Ac$  函数集合并满足: 对任意  $i \in A_r^A$ , 任意状态  $g \in Q$ .

- 1)  $f(i, g) \in \lambda_i(g(i))$ ;
- 2) 如果  $A(i)=ir$ , 则对任意状态  $g' \in Q$ , 若  $g(i)=g'(i)$  成立, 则  $f(i, g)=f(i, g')$ .

给定两个函数  $f \in F_A^A, \bar{f} \in F_{\bar{A}}^A$  和 TIS 系统  $T, T(f, \bar{f})$  表示一个 Kripke 结构  $(Q, Q_0, \delta, L)$ , 其中,  $\delta$  定义为  $q' \in \delta(q)$  当且仅当  $q=(q_1, \dots, q_k), q'=(q'_1, \dots, q'_k)$ , 存在一个联合动作  $\alpha \in Ac$ , 满足如下条件.

- 1)  $\delta(q, \alpha)=q'$ ;
- 2) 对任意  $i \in \text{Agt}$ , 如果  $i \in A_r^A$ , 则  $f(i, q)=\alpha(i)$ ; 如果  $i \in \bar{A}_r^A$ , 则  $\bar{f}(i, q)=\alpha(i)$ .

$\|\varphi\|_T$  计算方法见算法 1, 其中假设: 对于一个给定的 Kripke 结构和 CTL 公式  $\varphi, MC(K, \varphi)$  返回满足公式  $\varphi$  的状态集合, 该问题可以在多项式时间内解决<sup>[1]</sup>. 具体来说, 输入一个 ATL 公式  $\varphi$  和一个 TIS 系统  $T$ , 通过公式的结构归纳法计算满足子公式的状态集合. 第 2 行~第 5 行显而易见, 难点在公式  $\varphi=\langle\langle A \rangle\rangle \varphi'$ . 由于  $\varphi'$  中可能有嵌套形如  $\varphi=\langle\langle A' \rangle\rangle \varphi''$  的子公式, 对于这种情况, 采用递归调用, 先计算满足这些公式的状态集, 然后把公式替换为一些新的原子命题, 并更新  $L$  函数(第 7 行~第 10 行). 根据假设, 对任意  $i \in A, A(i) \in \{Ir, ir\}$  成立, 因此,  $A$  中智能体的策略数量是有限的, 可以通过枚举法验证每一条可能的策略. 对每种  $A$  中智能体策略的策略组合  $f \in F_A^A$ , 考虑所有  $\bar{A}$  的策略. 对于  $\bar{A}$  策略类型为  $Ir$  或  $ir$  智能体, 同理枚举所有可能的策略组合. 而策略类型为  $IR$  的策略数量是无限的, 无法通过枚举进行处理. 因此, 将问题规约到 CTL 在 Kripke 结构上的模型检查问题. 其主要思想为:  $g \in \|\langle\langle A \rangle\rangle \varphi\|_T$  当且仅当存在一个联合策略  $\theta_A$  (等价于第 12 行的  $f$ ), 任意路径  $\pi \in \text{Out}(g, \theta_A)$  都满足  $\varphi$ . 即: 当且仅当存在一个联合策略  $\theta_A$ , 对任意联合策略  $\theta_{\bar{A}_r^A}$  (等价于第 13 行的  $\bar{f}$ ), 任意路径  $\pi \in \text{Out}(g, \theta_A \oplus \theta_{\bar{A}_r^A})$  都满足  $\varphi$ , 其中  $\theta_A \oplus \theta_{\bar{A}_r^A}$  是一个联合策略, 满足: 对于任意  $i \in \text{Agt}$ , 任意状态  $g \in Q$ .

$$\begin{cases} (\theta_A \oplus \theta_{\bar{A}_r^A})(i)(g) = \theta_A(i)(g), & \text{if } i \in A \\ (\theta_A \oplus \theta_{\bar{A}_r^A})(i)(g) = \theta_{\bar{A}_r^A}(i)(g), & \text{if } i \in \bar{A}_r^A \end{cases}$$

**算法 1.** 模型检查算法.

输入: ATL 公式  $\varphi$ , TIS 系统  $T = (\langle\langle Q_i, A_i, \lambda_i, \delta_i, Q_i^0 \rangle\rangle_{i \in \text{Agt}}, L, A)$ ;

输出:  $\|\varphi\|_T$ .

$MC(T, \varphi)$

1. **begin**
2. **switch**  $\varphi$  **do**
3.     **case**  $\varphi=p$ : **return**  $L(p)$ ;
4.     **case**  $\varphi=\neg\varphi'$ : **return**  $Q/MC(T, \varphi')$ ;
5.     **case**  $\varphi=\varphi_1\vee\varphi_2$ : **return**  $MC(T, \varphi_1)\cup MC(T, \varphi_2)$ ;
6.     **case**  $\varphi=\langle\langle A \rangle\rangle\varphi'$ :
7.         **for each** subformula  $\phi=\langle\langle A' \rangle\rangle\varphi''$  in  $\varphi'$  **do**
8.             Replace  $\phi$  by a fresh atomic proposition  $p_\phi$  in  $\varphi$
9.             Let  $L(p_\phi)=MC(T, \varphi)$ ;
10.         **end for**
11.          $S:=\emptyset$ ;
12.         **for each**  $f \in F_A^A$  **do**
13.              $S := S \cup \left( \bigcap_{\bar{f} \in F_{\bar{A}}^A} MC(T(f, \bar{f}), \langle\langle \emptyset \rangle\rangle\phi') \right)$ ;
14.         **end for**
15.         **return**  $S$ ;
16.     **end switch**
17. **end**

根据  $T(f, \bar{f})$  的定义可见:对于任意  $g, T(f, \bar{f})$  中的  $Paths(g)$  等价于  $Out(g, \theta_A \oplus \theta_{\bar{A}})$ .

**定理 3.** 给定一个 TIS  $T=(J, A)$  满足  $A:Ag t \rightarrow \{IR, Ir, ir\}$  时,对于所有 ATL 公式,其中出现的智能体  $i$  的策略类型  $A(i) \in \{Ir, ir\}$  满足时,模型检查问题可在 EXPTIME 内判断.

**证明:**正确性通过 ATL 结构归纳方法证明.对于  $p, \neg\phi$  和  $\varphi_1\vee\varphi_2$ ,结论显然成立,因此仅需要考虑  $\langle\langle A \rangle\rangle\phi$ .由于  $A$  中的智能体采用  $ir$  或  $Ir$  策略,因此,  $F_A^A$  中的函数一一对应到  $S_A$  中的联合策略  $\theta_A$ ;同理,  $F_{\bar{A}}^A$  中的函数一一对应到  $S_{\bar{A}}$  中的联合策略  $\theta_{\bar{A}}$ .在  $T(f, \bar{f})$  中  $Paths(g)$  与  $T$  中在给定  $\theta_A$  和  $\theta_{\bar{A}}$  时所有从  $g$  出发的路径  $Out(g, \theta_A \oplus \theta_{\bar{A}})$  等价.

$g \in \bigcap_{\bar{f} \in F_{\bar{A}}^A} MC(T(f, \bar{f}), \langle\langle \emptyset \rangle\rangle\phi)$  当且仅当所有  $\pi \in Out(g, \theta_A)$  满足公式  $\phi$ .因此正确性成立.

**复杂度:**CTL 在 Kripke 结构上的模型检查问题可在多项式时间内解决,算法 1 中的第 6 行~第 14 行内需要考虑两个递归,而  $F_A^A$  和  $F_{\bar{A}}^A$  中函数数量最多为  $|Ac|^{Q \times |Agr|}$ ,因此,算法 1 可以在 EXPTIME 内终止.

**备注 1.** 该模型检查算法同样适用于 ATL\*逻辑,只需要对  $MC(K, \varphi)$  调用 CTL\* 在 Kripke 结构的模型检查算法即可.不仅如此,该算法也可以很容易地扩展“知识”操作,如,  $K_i\varphi$  表示智能体  $i$  知道  $\varphi$  为真.

#### 4 ShTMC 模型检查工具及实验

根据算法 1,本文在开源软件 MCMAS<sup>[26,27]</sup>基础上设计实现了基于 OBDD 的符号化模型检查工具 ShTMC.MCMAS 采用解释系统编程语言(interpreted systems programming language,简称 ISPL)描述多个智能体的状态转移,支持多种包括 CTL,ATL 逻辑的基于 OBDD 的符号化模型检查.为便于系统建模,MCMAS 中可以定义一种特殊的智能体,叫环境(environment).环境中定义的状态信息可以全部或部分被其他智能体观察到,这样可以减少智能体间的同步和通信.本文在 ISPL 语言中增加智能体策略类型描述语法用于 TIS 建模,同时继承了环境智能体这一特性.

为了验证本文所提出方法的性能,以匿名协议——DCP(dining cryptographers protocol)及其变种为例,在 Cygwin 平台上进行实验.DCP 是一个基于数学不可解特性的基础安全匿名通信协议,其主要特点是通过提供匿

名信息服务来避免恶意攻击.具体描述如下:该协议中有  $k$  个密码员一起就餐,餐费是匿名者支付的,可能是其中一个密码员付的餐费,或者是他们的雇主付的餐费.密码员之间尊重彼此支付餐费的隐私,但是他们希望知道是不是雇主付了餐费.为解决这个问题,他们制定了如下的协议:每个人都抛掷一个质地均匀的硬币,抛掷结果能且仅能被自己和坐在自己右边的人知晓.然后,每个人同时大声告知所有人,自己抛掷的硬币和坐在自己左边的人抛掷的硬币正反面是否一致.如果某个密码员就是匿名支付者,那么他所告知的硬币正反面是否一致的这个信息的就要和他亲眼看到的事实相反(即两个硬币正反面一致就要说成不一致;正反面不一致就要说成一致).此时,如果有奇数个人声明“硬币正反面不一致”,那么这次晚餐是由其中一个密码员支付的,但这位匿名密码员谁,其他密码员是不清楚的;反之,如果声明“硬币正反面不一致”的人数是偶数,那么这次餐费则是由他们的雇主支付的.

在这个例子中,每个密码员可以建模为一个智能体,其局部状态信息包括该密码员是否支付餐费(即:paid, not-paid)和是否看到硬币正反面一致(即:see-equal, see-different 和初始值 empty),其是否声明看到正反面一致则用动作刻画(即:say-equal, say-different 和初始值 none).因此,密码员的局部状态数量为 6 个.另外有一个环境智能体用于建模每个密码员硬币投掷的正反面(即 0 和 1)情况,其局部状态数量为 8 个.全局状态数量为  $8 \times 6^k$ .但在这个系统中并不是所有全局状态都可达,比如在  $k=3$  时,可达的全局状态数量只有 96 个.

针对 DCP 模型,本文验证了 ATL 公式: $(agent1)F(odd \rightarrow (agent1paid \text{ or } agent2paid \text{ or } agent3paid))$ .该公式是为分析 ShTMC 工具的性能随意构造的,其中,agent1 代表密码员 1, agent1paid 表示密码员 1 支付了餐费, agent2paid 和 agent3paid 类似.Odd 表示有奇数个人声明“硬币正反面不一致”.

表 1 展示了 ShTMC 模型检查工具对 DCP 系统的验证结果,其中,第 1 列给出了密码员数量, $m(n)$  表示一共有  $m$  个密码员,其前  $n$  个密码员是 *ir* 策略类型,其他则是 *IR* 策略类型,环境智能体也采用 *IR* 策略类型;第 2 列给出了系统中可达状态数量;第 3 列是 BDD 中布尔变量数;第 4 列给出了执行时间,-表示时间超过 40 分钟;第 5 列给出了 BDD 所使用的内存.BDD 变量顺序按照 MCMAS 默认排序.

Table 1 Experimental results

表 1 实验结果

智能体数量	可达状态数量	组合策略数量	BDD 变量数	执行时间(秒)	消耗内存(兆)
3 (0)	128	1	35	<0.01	10.20
3 (1)	128	16	35	0.013	10.48
3 (2)	128	256	35	0.076	11.22
3 (3)	96	4 096	35	1.511	11.33
4 (0)	320	1	45	<0.01	10.47
4 (1)	320	16	45	0.022	11.17
4 (2)	320	256	45	0.181	11.52
4 (3)	320	4 096	45	2.706	12.45
4 (4)	240	65 536	45	73.133	13.53
5 (0)	768	1	55	0.031	10.40
5 (1)	768	16	55	0.037	10.55
5 (2)	768	256	55	0.152	11.43
5 (3)	768	4 096	55	3.599	12.97
5 (4)	768	65 536	55	103.88	15.43
5 (5)	576	1 048 576	55	2 016.15	130.85
6 (0)	1 792	1	65	0.034	10.76
6 (1)	1 792	16	65	0.041	10.80
6 (2)	1 792	256	65	0.155	11.73
6 (3)	1 792	4 096	65	5.574	17.95
6 (4)	1 792	65 536	65	151.357	33.68
6 (5)	1 792	1 048 576	65	2 270.37	220.71
6 (6)	1 344	16 777 216	65	-	-

从表 1 可见:在密码员数量为  $m$  时,如果所有密码员都采用 *ir* 策略(即  $n=m$ ),那么可达状态数量会比  $m>n>0$  时少.这是因为当所有智能体都是 *ir* 策略时,智能体将无法区分全局状态故而只能选择同一个动作,并导致了系统中的某些状态不可达.当有一个或多个智能体不是 *ir* 时,原本只能选择同一动作的智能体可以选择更多动作,此时,这些不可达的状态将可达.另一方面,在相同密码员数量时,随着采用 *ir* 策略的智能体数量增加,集合  $F_A^A$  会

变大,直接导致算法 1 需要更多的运行时间.虽然在  $m > 6$  时,系统运行时间较多,但 BDD 的变量数目不大,说明 ShTMC 可以验证规模较大的系统.

## 5 结语与展望

针对以往工作在多智能体系统中智能体策略能力建模的不足,本文提出了带策略类型的解释系统作为异构多智能体系统模型,该系统模型允许各智能体有其独立的策略能力.以 ATL 逻辑为例,本文研究了 ATL 逻辑在带策略类型的解释系统下的语义和模型检查问题.由于模型表达能力过强,ATL 模型检查算法问题不可判定.因此,本文考虑一种特殊的带策略类型的解释系统,其中,在 ATL 公式联合模态操作中出现的智能体可以是  $Ir$  或  $ir$  策略类型,而其他智能体可以是  $IR,ir$  或者  $Ir$  策略类型,提出了一个 EXPTIME 复杂度的模型检查算法,设计并实现了工具原型 ShTMC.

本文仅仅是针对多智能体系统中智能体策略能力建模的不足问题的初步工作,尚有很多问题有待后续深入研究:(1) 其他表达力更强的 ATL 逻辑的扩展,如  $ATL^*$ , Strategy Logic 的语义和模型检查问题;(2) 其他可能的策略类型,如可撤销和可删除策略类型等.

## References:

- [1] Clarke EM, Grumberg O, Peled DA. Model Checking. Cambridge: MIT Press, 1999.
- [2] Clarke EM, Emerson EA. Design and synthesis of synchronization skeletons using branching-time temporal logic. In: Proc. of the Workshop on Logics of Programs. Berlin: Springer-Verlag, 1981. 52–71. [doi: 10.1007/978-3-540-69850-0\_12]
- [3] Kupferman O, Vardi MY. Module checking. In: Proc. of the 8th Int'l Conf. on Computer Aided Verification. Berlin: Springer-Verlag, 1996. 75–86. [doi: 10.1007/3-540-61474-5\_59]
- [4] Muscettola N, Nayak PP, Pell B, Williams BC. Remote agent: To boldly go where no AI system has gone before. Artificial Intelligence, 1998,103(1-2):5–47. [doi: 10.1016/S0004-3702(98)00068-X]
- [5] Alur R, Henzinger TA, Kupferman O. Alternating-Time temporal logic. Journal of the ACM, 2002,49(5):672–713. [doi: 10.1145/585265.585270]
- [6] Schobbens P. Alternating-Time logic with imperfect recall. Electronic Notes in Theoretical Computer Science, 2004,85(2):82–93. [doi: 10.1016/S1571-0661(05)82604-0]
- [7] Dima C, Tiplea FL. Model-Checking ATL under imperfect information and perfect recall semantics is undecidable. CoRR, abs/1102.4225, 2011.
- [8] Jamroga W, Dix J. Model checking abilities under incomplete information is indeed Delta2-complete. In: Proc. of the 4th European Workshop on Multi-Agent System. CEUR-WS, 2006.
- [9] Pilecki J, Bednarczyk MA, Jamroga W. Model checking properties of multi-agent systems with imperfect information and imperfect recall. In: Proc. of the 7th Int'l Conf. on Intelligent Systems. Berlin: Springer-Verlag, 2014. 415–426. [doi: 10.1007/978-3-319-11313-5\_37]
- [10] Jamroga W. Logical Methods for Specification and Verification of Multi-Agent Systems. Warszawa: ICS PAS Publishing House, 2015.
- [11] Chatterjee K, Henzinger TA, Piterman N. Strategy logic. In: Proc. of the 18th Int'l Conf. on Concurrency Theory. Berlin: Springer-Verlag, 2007. 59–73. [doi: 10.1007/978-3-540-74407-8\_5]
- [12] Mogavero F, Murano A, Vardi MY. Reasoning about strategies. In: Proc. of the Annual Conf. on Foundations of Software Technology and Theoretical Computer Science. Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik, 2010. 133–144. [doi: 10.4230/LIPIcs.FSTTCS.2010.133]
- [13] Mogavero F, Murano A, Vardi MY. Reasoning about strategies: On the model-checking problem. ACM Trans. on Computational Logic, 2014,15(4):34:1–34:47. [doi: 10.1145/2631917]
- [14] Jamroga W. Some remarks on alternating temporal epistemic logic. In: Proc. of the Formal Approaches to Multi-Agent Systems. 2003. 133–140.

- [15] Hoek W, Wooldridge M. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 2003,75(1):125–157. [doi: 10.1023/A:1026185103185]
- [16] Lomuscio A, Raimondi F. Model checking knowledge, strategies, and games in multi-agent systems. In: *Proc. of the 5th Int'l Joint Conf. on Autonomous Agents and Multiagent Systems*. New York: ACM Press, 2006. 161–168. [doi: 10.1145/1160633.1160660]
- [17] Hoek W, Wooldridge M. Tractable multiagent planning for epistemic goals. In: *Proc. of the 1st Int'l Joint Conf. on Autonomous Agents and Multiagent Systems*. New York: ACM Press, 2002. 1167–1174. [doi: 10.1145/545056.545095]
- [18] Bulling N, Jamroga W. Alternating epistemic mu-calculus. In: *Proc. of the 22nd Int'l Joint Conf. Artificial Intelligence*. Austin: AAAI, 2011. 109–114. [doi: 10.5591/978-1-57735-516-8/IJCAI11-030]
- [19] Agotnes T, Goranko V, Jamroga W. Alternating-Time temporal logics with irrevocable strategies. In: *Proc. of the 11th Conf. on Theoretical Aspects of Rationality and Knowledge*. New York: ACM Press, 2007. 15–24. [doi: 10.1145/1324249.1324256]
- [20] Pinchinat S. A generic constructive solution for concurrent games with expressive constraints on strategies. In: *Proc. of the 5th Int'l Symp. on Automated Technology for Verification and Analysis*. Berlin: Springer-Verlag, 2007. 253–267. [doi: 10.1007/978-3-540-75596-8\_19]
- [21] Lopes ADC, Laroussinie F, Markey N. ATL with strategy contexts: Expressiveness and model checking. In: *Proc. of the IARCS Annual Conf. on Foundations of Software Technology and Theoretical Computer Science*. Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik, 2010. 120–132. [doi: 10.4230/LIPIcs.FSTTCS.2010.120]
- [22] Laroussinie F, Markey N. Augmenting ATL with strategy contexts. *Information and Computation*, 2015,245:98–123. [doi: 10.1016/j.ic.2014.12.020]
- [23] Goranko V, Kuusisto A, Rönholm R. Game-Theoretic semantics for ATL+ with applications to model checking. In: *Proc. of the 16th Conf. on Autonomous Agents and MultiAgent Systems*. New York: ACM Press, 2017. 1277–1285.
- [24] Wang F, Schewe S, Huang CH. An extension of ATL with strategy interaction. *ACM Trans. on Programming Languages and Systems*, 2015,37(3):9:1–9:41. [doi: 10.1145/2734117]
- [25] Huang CH, Schewe S, Wang F. Model-Checking iterated games. *Acta Informatica*, 2017,54(7):625–654. [doi: 10.1007/s00236-016-0277-y]
- [26] Lomuscio A, Qu H, Raimondi F. MCMAS: A model checker for the verification of multi-agent systems. In: *Proc. of the 21st Int'l Conf. on Computer Aided Verification*. Berlin: Springer-Verlag, 2009. 682–688. [doi: 10.1007/978-3-642-02658-4\_55]
- [27] Lomuscio A, Qu H, Raimondi F. MCMAS: An open-source model checker for the verification of multi-agent systems. *Int'l Journal on Software Tools for Technology Transfer*, 2017,19(1):9–30. [doi: 10.1007/s10009-015-0378-x]
- [28] Chaum D. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1988,1(1): 65–75. [doi: 10.1007/BF00206326]



张业迪(1996—),女,黑龙江五常人,硕士生,主要研究领域为模型检查.



宋富(1983—),男,博士,tenure-track 助理教授,研究员,博士生导师,CCF 专业会员,主要研究领域为模型检查,程序分析与验证,系统安全.