A Modal Logic for Joint Abilities of Structured Strategies with Bounded Complexity

Ruiqi Jin[1], **Yongmei Liu**[1*], **Liping Xiong**[2]

[1]Department of Computer Science, Sun Yat-sen University, Guangzhou 510006, China
[2]College of Electronics and Information Engineering, Wuyi University, Jiangmen 529020, China
jinrq7@mail2.sysu.edu.cn, ymliu@mail.sysu.edu.cn, xionglp@wyu.edu.cn

Abstract

Coordination and joint abilities are important topics in representation and reasoning about multi-agent systems. The modal logic JAADL proposed by Liu et al. extends ATL with joint abilities, which enables reasoning about whether a coalition of agents can coordinate and achieve a goal without communication. However, like ATL, strategic abilities in JAADL are defined in terms of combinatorial strategies, which are functions from histories or states to actions. On the other hand, there has been research on reasoning about natural strategic abilities, where a natural strategy is formalized as a sequence of condition-action pairs, making it more human-friendly than the notion of combinatorial strategies. In this work, we propose SJAADL, a variation of JAADL where strategic abilities are defined in terms of structured strategies represented with LDL (linear dynamic logic) formulas, with bounded complexity. We use nondeterministic strategies since they are more expressive, natural and succinct than determinstic ones. We present syntax and semantics of SJAADL. We show that model checking SJAADL can be done in time polynomial with the model size, exponential with the formula size, and with the complexity bound of structured strategies, exponential in the memoryless case and double exponential in the memoryful case. Finally, we introduce the problem of synthesizing norms to achieve joint abilities, and give two algorithms for it.

Introduction

Representation and reasoning about strategic abilities has been an active research field in AI and logic. A fundamental contribution in this field is Alternating-time Temporal Logic ATL/ATL* (Alur, Henzinger, and Kupferman 2002) where formula $\langle\langle A\rangle\rangle\varphi$ means coalition A has a strategy to ensure the LTL goal φ holds. Many variants and extensions of ATL have been studied in the area of multi-agent systems.

Cooperation and coordination abilities are important problems in multi-agent systems. There could be multiple collective strategies to ensure a goal, but a player may not know other agents' choices, thus a coalition might end up with a collective strategy that does not achieve the goal. Ghaderi, Levesque, and Lespérance (2007) present a formalization of joint abilities of coalitions based on the idea of iterated elimination of dominated strategies (Osborne and Rubinstein 1994). Essentially, a coalition has joint ability if after such iterated elimination, any remaining collective strategy achieves the goal. Based on their idea, Liu et al. (2020) propose JAADL (Alternating-time Dynamic Logic with Joint Abilities), an extension of ATL* with modalities $(\!(A)\!)^\infty\varphi$, meaning coalition A has joint ability to achieve φ. They show that model checking memoryless JAADL can be done in time exponential in both the model and formula size, but whether model checking memoryful JAADL is decidable remains open.

In recent years, it has been brought into attention the issue of whether to use combinatorial strategies or syntactic forms of strategies to define strategic abilities. Combinatorial strategies, defined as functions mapping histories or states to actions, are used in defining strategic abilities in ATL and most strategic logics. However, there have been arguments that the notion of combinatorial strategies is not a good representation of human strategies, as humans are not good at handling combinatorially complex objects (Jamroga, Malvone, and Murano 2019). Syntactic forms of strategies, called structured strategies, have been investigated, and regular expressions have been used to represent the internal structures of strategies (Ramanujam and Simon 2008; van Eijck 2013). Most importantly, Jamroga, Malvone, and Murano (2017; 2019) propose NatATL, a variation of ATL where strategic abilities are defined in terms of natural strategies, formalized as sequences of pairs of regular expression conditions and actions. In NatATL, the operator $\langle\langle A\rangle\rangle^{\leq k}\varphi$ means coalition A has a collective natural strategy with complexity $\leq k$ to ensure the goal φ holds.

In this paper, we propose SJAADL, an adaptation of JAADL where strategic abilities are defined in terms of structured strategies with bounded complexity. We choose to use structured strategies, since they are more general than natural strategies. Also, our structured strategies are nondeterministic, and hence more expressive, succinct, and natural than deterministic strategies. Just as natural strategies, there are mainly two advantages in considering structured strategies instead of combinatorial ones. First, structured strategies are more human friendly, as they better capture the intuitive approach a human would use when describing strategies. Thus, using structured strategies in JAADL would better describe the process of dominated strategy elimina-

*Corresponding author

tion of humans, thus better capture the decision-making by humans and human-like agents with bounded rationality. Second, structured strategies with bounded complexity are better models of behavior for agents with limited memory, compared with combinatorial strategies with bounded memory. In this paper, we use structured strategies in the form of condition-action pairs, where a condition is a formula of LDL_f, *i.e.*, Linear Dynamic Logic interpreted over finite traces (De Giacomo and Vardi 2013). Whether a finite trace satisfies an LDL_f formula can be checked by a deterministic finite-state automaton (DFA). Imposing bound on the length of the condition, and thus imposing bound on the number of states of the DFA better models agents with bounded memory, compared with imposing bound on the length of the history an agent can memorize. We prove that JAADL is not less expressive than SJAADL, but is more expressive than a variant of SJAADL using deterministic structured strategies. Finally, we show that model checking SJAADL can be done in time polynomial with the model size, exponential with the formula size, and with the complexity bound of structured strategies, exponential in the memoryless case and double exponential in the memoryful case.

Preliminaries

In this section, we introduce JAADL (Liu et al. 2020).

Let AP be a set of atoms, AC a set of actions, and $AG = \{1, \dots n\}$ a set of agents; all are finite and non-empty.

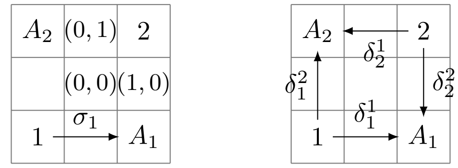
Definition 1. A concurrent game structure (CGS) is a tuple $\mathcal{G} = \langle W, L, P, \tau, w^0 \rangle$, where W is a finite non-empty set of states; $w^0 \in W$ is the initial state; $L : W \to 2^{AP}$ is a labeling function; for each agent i, $P_i : W \to 2^{AC}$ specifies her available actions at each state; τ is the transition function mapping a state w and a tuple of actions chosen by each agent $\langle a_1, \dots a_n \rangle$ (called a decision d) to a new state.

Example 1. As shown in Figure 1, two squirrels and two acorns live on a finite grid. Squirrels can move around. A squirrel can see an acorn if they are in the same place, and if the squirrel does not hold an acorn, she can try to pick up the acorn, but it only succeeds if the other squirrel is not trying to pick it up. We model such a system with a CGS where

- $AG = \{1, 2\}$, $AC = \{\mathsf{e}, \mathsf{s}, \mathsf{w}, \mathsf{n}, \mathsf{idle}, \mathsf{pickUp}\}$, $AP = \{\mathsf{seeAcorn}_i, \mathsf{hasAcorn}_i, \mathsf{posE}_i, \mathsf{posS}_i, \mathsf{posW}_i, \mathsf{posN}_i \mid i \in AG\}$, where, *e.g.*, posE_i means it is possible for i to move east.
- $W = \{(x_1, y_1, x_2, y_2, ax_1, ay_1, ax_2, ay_2, h_1, h_2) \mid -1 \leq x_j, y_j, ax_j, ay_j \leq 1, h_j = 0, 1, 2, j = 1, 2\}$, where (x_j, y_j) is the location of squirrel j, (ax_j, ay_j) is the location of acorn j, h_j means which acorn is hold by squirrel j, or she holds nothing ($h_j = 0$).
- $w^0 = (-1, -1, 1, 1, 1, -1, -1, 1, 0, 0)$.
- L, P and τ are defined intuitively.

Definition 2. A track h in a CGS $\mathcal{G}$ is a finite state-decision sequence $w_0 d_0 \dots w_n$ s.t. for all i, d_i is a decision at state w_i, and $\tau(w_i, d_i) = w_{i+1}$. We let $last(h) = w_n$.

Definition 3. A path λ in a CGS $\mathcal{G}$ is an infinite state-decision sequence $w_0 d_0 \dots$ s.t. for all i, d_i is a decision at state w_i, and $\tau(w_i, d_i) = w_{i+1}$.



(a) (b)

Figure 1: A simple squirrel world. (a) Combinatorial strategy σ_1. (b) The four structured strategies in Example 4.

We introduce combinatorial strategies and the corresponding strategy space for JAADL.

Definition 4. A combinatorial strategy σ_i of agent i from state w is a function mapping each track h beginning from w to an action $a \in P_i(last(h))$. We use $CS_i(w)$ to denote the set of combinatorial strategies for agent i from w.

Definition 5. A combinatorial strategy space cs is a function mapping each agent i to a subset of $CS_i(w)$. The full combinatorial strategy space $fcs(w)$ maps each agent i to $CS_i(w)$. cs_A is the restriction of cs to coalition A, which is a group of agents.

We also introduce memoryless combinatorial strategies.

Definition 6. A memoryless combinatorial strategy for agent i is a function mapping each state w to an action from $P_i(w)$. The full memoryless combinatorial strategy space fcs^r maps i to the set of all memoryless strategies for i.

We use σ to represent combinatorial strategies. σ_A ranges over collective strategies of coalition A, which is a collection of individual strategies for each agent in A. σ_i ranges over individual strategies of agent i. We use $-A$ to denote $AG - A$, $-i$ for $AG - \{i\}$. We will represent collective strategies with their components, e.g., (σ_1, σ_2) is a collective strategy of coalition $\{1, 2\}$, where agent 1 uses σ_1 and agent 2 uses σ_2; (σ_A, σ_{-A}) is a collective strategy of AG, etc.

Example 2. If squirrel 2 does not interfere, squirrel 1 has a memoryless strategy σ_1 (drawn in Figure 1a) to get acorn 1, formalized as follows:

- $\sigma_1(x, -1, \cdot, \cdot, 1, -1, \cdot, \cdot, 0, \cdot) = \mathsf{e}$ for $-1 \leq x < 1$;
- $\sigma_1(1, -1, \cdot, \cdot, 1, -1, \cdot, \cdot, 0, \cdot) = \mathsf{pickUp}$;
- Other states are mapped to idle.

Definition 7. A state w and a collective combinatorial strategy σ_{AG} determine a unique path $w_0 d_0 w_1 d_1 \dots$ as follows: $w_0 = w$, and for each $j \geq 0$, d_j is the decision associated to track $w_0 \dots w_j$, *i.e.*, for each agent i, $d_j(i) = \sigma_i(w_0 \dots w_j)$, $w_{j+1} = \tau(w_j, d_j)$. We use $out(w, \sigma_{AG})$ to denote this path.

We introduce syntax and semantics of JAADL. For $a \in AC$ and $i \in AG$, we have a_i to denote the atomic proposition that agent i does action a, as a type of atomic propositions outside AP. We use $\top$ for $true$ and $\bot$ for $false$.

Definition 8 (JAADL syntax)**.** JAADL uses state formula φ, path formula ψ, path expression ρ, and propositional formula ϕ. Let $p \in AP$, and $A \subseteq AG$, we have:

$$\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle\langle A\rangle\rangle\psi \mid (A)_\psi\varphi \mid (A)^\infty_\psi\varphi,$$
$$\psi ::= \varphi \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \langle\rho\rangle\psi,$$
$$\rho ::= \phi \mid \psi? \mid \rho_1 + \rho_2 \mid \rho_1; \rho_2 \mid \rho^*,$$
$$\phi ::= p \mid a_i \mid \neg\phi \mid \phi_1 \wedge \phi_2.$$

Here, in path formulas, $\langle\rho\rangle\psi$ means there exists an execution from the current state which satisfies the path expression ρ s.t. its last state satisfies ψ. We use $[\rho]\psi$ to abbreviate $\neg\langle\rho\rangle\neg\psi$. In state formulas, $\langle\langle A\rangle\rangle\psi$ means coalition A has a combinatorial strategy to achieve ψ. We write $\langle\langle i_1,\ldots,i_k\rangle\rangle$ for $\langle\langle\{i_1,\ldots,i_k\}\rangle\rangle$ where $i_1,\ldots,i_k \in AG$. The formula $\langle\langle\varnothing\rangle\rangle\psi$, where $\varnothing$ denotes the empty set, means ψ holds no matter how agents play. $(A)_\psi\varphi$ means φ holds after one step elimination of dominated strategies w.r.t. coalition A and goal ψ. $(A)^2_\psi\varphi$ is used to denote $(A)_\psi(A)_\psi\varphi$, and similarly for $(A)^n_\psi\varphi$. $(A)^\infty_\psi\varphi$ means φ holds after iterated elimination of dominated strategies w.r.t A and ψ.

We write $((A))^n\psi$ for $(A)^n_\psi\langle\langle\varnothing\rangle\rangle\psi$, and $((A))\psi$ for $((A))^1\psi$. Here $((A))^n\psi$ means after n-round elimination of dominated strategies, ψ holds no matter how the agents play. Therefore, coalition A has stage-n joint ability to achieve ψ. We write $((A))^\infty\psi$ for $(A)^\infty_\psi(\langle\langle A\rangle\rangle\psi \wedge \langle\langle\varnothing\rangle\rangle\psi)$. The reason $\langle\langle\varnothing\rangle\rangle\psi$ is conjoined with $\langle\langle A\rangle\rangle\psi$ is that the strategy space might become empty after iterated elimination.

We give semantics of JAADL. The cases of $\neg$ and $\wedge$ are trivial thus omitted for simplicity. For interpreting state formulas w.r.t a strategy space, two reduction operators on combinatorial strategy spaces are used: $R_{A,\psi,w}(cs)$ and $R^\infty_{A,\psi,w}(cs)$, meaning the reduction of cs by one step and iterated elimination of dominated strategies, respectively. The operator of iterated reduction applies the one step reduction operator iteratively on the strategy space until reaching a fixed point, *i.e.*, no strategy is dominated (and thus can be eliminated) in the resulting strategy space. Their definitions and the following definition are mutual-recursive.

Definition 9 (JAADL semantics)**.** Given a CGS $\mathcal{G}$, a state w, a decision d at w, we interpret propositional formulas as follows: $w, d \vDash p$ if $p \in L(w)$; $w, d \vDash a_i$ if $d(i) = a$.

Given a CGS $\mathcal{G}$, a state w, a combinatorial strategy space cs, and a path λ, state formulas and path formulas are interpreted inductively:

- $w, cs \vDash p$ if $p \in L(w)$.
- $w, cs \vDash \langle\langle A\rangle\rangle\psi$ if there exists a collective combinatorial strategy $\sigma_A \in cs_A$ s.t. for all strategies $\sigma_{-A} \in cs_{-A}$, we have $out(w, (\sigma_A, \sigma_{-A})), cs \vDash \psi$.
- $w, cs \vDash (A)_\psi\varphi$ if $w, R_{A,\psi,w}(cs) \vDash \varphi$.
- $w, cs \vDash (A)^\infty_\psi\varphi$ if $w, R^\infty_{A,\psi,w}(cs) \vDash \varphi$.
- $\lambda, cs \vDash \varphi$ if $w_0, cs \vDash \varphi$, where $\lambda = w_0d_0w_1\ldots$.
- $\lambda, cs \vDash \langle\phi\rangle\psi$ if $w_0, d_0 \vDash \phi$ and $\lambda', cs \vDash \psi$, where $\lambda = w_0d_0w_1\ldots$ and $\lambda' = w_1d_1\ldots$.
- $\lambda, cs \vDash \langle\psi_1?\rangle\psi_2$ if $\lambda, cs \vDash \psi_1$ and $\lambda, cs \vDash \psi_2$.
- $\lambda, cs \vDash \langle\rho_1 + \rho_2\rangle\psi$ if $\lambda, cs \vDash \langle\rho_1\rangle\psi$ or $\lambda, cs \vDash \langle\rho_2\rangle\psi$.
- $\lambda, cs \vDash \langle\rho_1; \rho_2\rangle\psi$ if $\lambda, cs \vDash \langle\rho_1\rangle\langle\rho_2\rangle\psi$.
- $\lambda, cs \vDash \langle\rho^0\rangle\psi$ if $\lambda, cs \vDash \psi$.
- $\lambda, cs \vDash \langle\rho^{m+1}\rangle\psi$ if $\lambda, cs \vDash \langle\rho^m; \rho\rangle\psi$ for $m \in \mathbb{N}$.
- $\lambda, cs \vDash \langle\rho^*\rangle\psi$ if there is $m \in \mathbb{N}$ s.t. $\lambda, cs \vDash \langle\rho^m\rangle\psi$.

A state formula φ is valid if for all CGS $\mathcal{G}$, we have $\mathcal{G} \vDash \varphi$, *i.e.*, $w^0, fcs(w^0) \vDash \varphi$.

We define the set of combinatorial strategies of $-i$ that work with σ_i to ensure ψ w.r.t. state w and combinatorial strategy space cs as follows: $M_{\psi,w,cs}(\sigma_i) = \{\sigma_{-i} \in cs_{-i} \mid out(w, (\sigma_i, \sigma_{-i})), cs \vDash \psi\}$. For σ_i, $\sigma'_i \in cs_i$, we write $\sigma_i \geq_{\psi,w,cs} \sigma'_i$ if $M_{\psi,w,cs}(\sigma_i) \supseteq M_{\psi,w,s}(\sigma'_i)$, and we say σ_i weakly dominates σ'_i. We write $\sigma_i >_{\psi,w,cs} \sigma'_i$ if $M_{\psi,w,cs}(\sigma_i) \supset M_{\psi,w,s}(\sigma'_i)$, and we say σ_i dominates σ'_i. We say that σ_i and σ'_i are incomparable if neither $\sigma_i \geq_{\psi,w,cs} \sigma'_i$ nor $\sigma'_i \geq_{\psi,w,cs} \sigma_i$.

For a combinatorial strategy space cs, we define the reduction of cs w.r.t. coalition A, goal ψ and state w as follows: $R_{A,\psi,w}(cs) = cs'$ s.t. if $i \notin A$, $cs'_i = cs_i$; otherwise $cs'_i = \{\sigma_i \in cs_i \mid \neg\exists\sigma'_i \in cs_i.\sigma'_i >_{\psi,w,cs} \sigma_i\}$. The iterative reduction of cs is defined as $R^\infty_{A,\psi,w}(cs) = cs'$ s.t. for $i \in AG$, $cs'_i = \bigcap_{n=0}^\infty R^n_{A,\psi,w}(cs)_i$.

Syntax and Semantics of SJAADL

In this section, we propose SJAADL, a variation of JAADL where strategic abilities are defined in terms of structured strategies of bounded complexity. We first introduce structured strategies. After presenting the syntax and semantics, we analyze properties of the logic.

Structured Strategies

To define structured strategies, we use a fragment of path formulas of SJAADL/JAADL, where the operators $\langle\langle A\rangle\rangle$, $(A)_\psi$, and $(A)^\infty_\psi$ are excluded from the definition of state formulas, and the action atoms a_i are excluded from the definition of propositional formulas. This fragment is equivalent to LDL_f, which has the same syntax as LDL. The semantics of LDL_f differs from that of LDL in the evaluation of $\langle\phi\rangle\psi$, where $h \vDash \langle\phi\rangle\psi$ iff $h = w_0d_0w_1\ldots$ has length at least 2, $w_0 \vDash \psi$ and $h' \vDash \psi$ where $h' = w_1d_1\ldots$.

We also introduce a simple action language AL, the propositional language over action atoms. Let α be an AL formula, and a an action. We say that a satisfies α, written $a \models \alpha$ if the assignment where a is the only atom assigned true satisfies α. Thus, an AL formula defines a set of actions.

Intuitively, a structured strategy is a set of condition-action pairs. When applying the strategy at a history, the action formulas corresponding to the satisfied conditions give the set of actions that can be performed.

Definition 10. A structured strategy δ_i of agent i is a set of pairs $\{(\psi_1, \alpha_1), \ldots, (\psi_n, \alpha_n)\}$, also written $\{\psi_1 \to \alpha_1, \ldots, \psi_n \to \alpha_n\}$ or $(\psi_j, \alpha_j)^n_{j=1}$, where ψ_j is a formula of LDL_f, and α_j is a formula of AL. We say that δ_i is memoryless if each ψ_j is propositional.

Definition 11. The application of a structured strategy $\delta_i = (\psi_j, \alpha_j)^n_{j=1}$ on a track h is a set of actions $\delta_i(h) \subseteq P_i(last(h))$ s.t. $a \in \delta_i(h)$ iff $a \vDash \alpha_j$ for some j s.t. $h \vDash \psi_j$. When δ_i is memoryless, we replace $h \vDash \psi_j$ by $last(h) \vDash \psi_j$.

Thus, unlike combinatorial strategies, which are deterministic as a history corresponds to a unique action, structured strategies are non-deterministic: Different actions might be performed at a history. In the rest of the paper, we assume a structured strategy to be well-defined: It gives at least one executable action for any given history. We use δ to range over structured strategies, $r\delta$ and $R\delta$ to denote memoryless and memoryful structured strategies, respectively.

Example 3. We first give an example of a memoryless structured strategy, and then a memoryful example. The combinatorial strategy of squirrel 1 in Example 2 can be represented with a structured strategy δ_1^1:

$$\{(\mathsf{seeAcorn}_1, \mathsf{pickUp}_1), (\mathsf{hasAcorn}_1, \top), (\neg\mathsf{seeAcorn}_1, \mathsf{e}_1)\}.$$

This strategy is non-deterministic: when she has an acorn, she can do any possible action.

Squirrel 1 can also have a passive strategy to get an acorn: she wanders around if she does not see an acorn or has one; She tries to pick up an acorn if seeing one, but gives up if squirrel 2 is also trying to pick it up, causing the action to fail. The strategy can be formalized as a memoryful structured strategy as follows:

$$\begin{aligned}\delta_1^p = \{&\langle \top^*; \mathsf{hasAcorn}_1 \vee \neg\mathsf{seeAcorn}_1\rangle Last \to \top,\\ &\langle \top^*; \mathsf{seeAcorn}_1; \neg\mathsf{hasAcorn}_1\rangle Last \to \neg\mathsf{pickUp}_1,\\ &\langle \top^*; \neg\mathsf{seeAcorn}_1; \mathsf{seeAcorn}_1\rangle Last \to \mathsf{pickUp}_1\}.\end{aligned}$$

We write $Last$ for $[\top]\bot$, meaning there is no next state.

We now define outcomes of collective structured strategies of a coalition. Since structured strategies are non-deterministic, the outcome of a collective structured strategy at a state is a set of paths.

Definition 12. A state w and a collective structured strategy δ_{AG} determine a set of paths $Out(w, \delta_{AG})$ as follows: For $w_0 d_0 \cdots \in Out(w, \delta_{AG})$, we have $w_0 = w$, and for each $j \geq 0$, d_j is a decision associated to track $w_0 \dots w_j$, *i.e.*, for each agent i, $d_j(i) \in \delta_i(w_0 \dots w_j)$, $w_{j+1} = \tau(w_j, d_j)$.

Definition 13. The length of an LDL_f formula ψ is the number of symbols in the formula where an atom is treated as a single symbol. The complexity of a structured strategy δ_i is the sum of length of all conditions in the strategy.

For example, δ_1^1 given in Example 3 has complexity 4.

Definition 14. A structured strategy space ss is a function mapping each agent i to a set of i's possible structured strategies. The full structured strategy space fss maps each agent i to the set of i's all possible structured strategies. A structured strategy space with bounded complexity $ss^{\leq k}$ maps i to a set of i's possible structured strategies with complexity $\leq k$. ss_A is the restriction of ss to coalition A.

It is easy to see that the number of different k-bounded structured strategies, memoryless or memoryful, is $\mathcal{O}(2^k)$, while the number of different combinatorial strategies is $\mathcal{O}(2^n)$, where n is the size of CGS. Thus structured strategy space is much smaller than combinatorial strategy space. This is a clear advantage of considering structured strategies.

Syntax and Semantics

Definition 15 (SJAADL syntax)**.** SJAADL modifies JAADL's state formula φ as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle\langle A\rangle\rangle^{\leq k}\psi \mid (A)_{\psi}^{\leq k}\varphi \mid (A)_{\psi}^{\leq k,\infty}\varphi,$$

where $k \in \mathbb{N}$, and operators appearing in the scopes of $\langle\langle A\rangle\rangle^{\leq k}, (A)_{\psi}^{\leq k}, (A)_{\psi}^{\leq k,\infty}$ should have the same k-superscripts. Other formulas are the same as in JAADL.

Intuitively, these modal operators have meanings similar to those of JAADL, but coalition A now uses k-bounded structured strategies. $((A))^{\leq k,n}\psi$, $((A))^{\leq k}\psi$ and $((A))^{\leq k,\infty}\psi$ are defined similarly to JAADL. If $((A))^{\leq k,n}\psi$ holds, we say coalition A has stage-n k-bounded joint ability to achieve ψ.

Definition 16 (SJAADL semantics)**.** Given a CGS $\mathcal{G}$, a state w, a structured strategy space ss, and a path λ, we interpret state formulas and path formulas as follows (we omit the cases similar to those of JAADL):

- $w, ss \models \langle\langle A\rangle\rangle^{\leq k}\psi$ if there exists a collective structured strategy $\delta_A \in ss_A^{\leq k}$ s.t. for all strategies $\delta_{-A} \in ss_{-A}^{\leq k}$, and for all $\lambda \in Out(w, (\delta_A, \delta_{-A}))$, $\lambda, ss^{\leq k} \models \psi$.
- $w, ss \models (A)_{\psi}^{\leq k}\varphi$ if $w, R_{A,\psi,w}^{\leq k}(ss^{\leq k}) \models \varphi$.
- $w, ss \models (A)_{\psi}^{\leq k,\infty}\varphi$ if $w, R_{A,\psi,w}^{\leq k,\infty}(ss^{\leq k}) \models \varphi$.

Note that our definition of the semantics of the strategic modality $\langle\langle A\rangle\rangle^{\leq k}$ is similar to that of JAADL, in which each agent gets a strategy assigned to her in the strategy space. In contrast, in NatATL, only agents in the coalition are assigned strategies, while other agents are allowed to perform arbitrary actions.

Definition 17. The set of structured strategies of $-i$ that work with δ_i to ensure ψ w.r.t. state w and structured strategy space $ss^{\leq k}$, denoted by $M_{\psi,w,ss^{\leq k}}(\delta_i)$, is defined as the set of $\delta_{-i} \in ss_{-i}^{\leq k}$ s.t. for all $\lambda \in Out(w, (\delta_i, \delta_{-i}))$, we have $\lambda, ss^{\leq k} \models \psi$.

The concept of strategy domination and the reduction operators can be similarly defined as in JAADL.

Example 4. Consider the squirrel world from Example 1. We check if given SJAADL formulas hold in w^0 and $fss^{\leq 4}$.

First, $\langle\langle 1\rangle\rangle^{\leq 4}\langle\top^*\rangle\mathsf{seeAcorn}_1$ holds, meaning squirrel 1 has a 4-bounded strategy to let her see an acorn, an example of which is δ_1^1 from Example 3. Secondly, $\langle\langle 1,2\rangle\rangle^{\leq 4}\langle\top^*\rangle(\mathsf{hasAcorn}_1 \wedge \mathsf{hasAcorn}_2)$ holds. As shown in Figure 1b, there are two achieving collective strategies. One is combining δ_1^1 with a strategy δ_2^1 for 2:

$$\{(\mathsf{seeAcorn}_2, \mathsf{pickUp}_2), (\mathsf{hasAcorn}_2, \top), (\neg\mathsf{seeAcorn}_2, \mathsf{w}_2)\}.$$

Here, 1 goes east and gets acorn 1, 2 goes west and gets acorn 2. The other collective strategy is δ_1^2 and δ_2^2, where 1 goes north and 2 goes south.

However, $((1,2))^{\leq 4,\infty}\langle\top^*\rangle(\mathsf{hasAcorn}_1 \wedge \mathsf{hasAcorn}_2)$ does not hold, as none of the strategies can be eliminated, since the compatible sets of δ_1^1 and δ_1^2 are $\{\delta_2^1\}$ and $\{\delta_2^2\}$, respectively, which are incomparable.

The above example inspires the following definition:

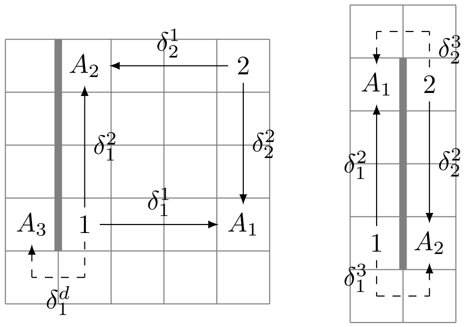


Figure 2: Walled squirrel worlds

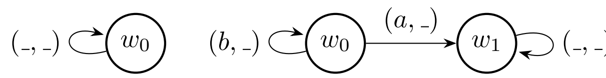


Figure 3: CGS $\mathcal{G}_c$ (left) and $\mathcal{G}'_c$ (right). All states are labeled by p. Actions of agents 1 and 2 are labeled on the edges.

Definition 18. We say that two strategies δ_i^1 and δ_i^2 of an agent i form an incoordination core if they are incomparable.

It is easy to prove the following:

Proposition 1. *If a coalition has strategic ability after iterated reduction but no joint ability, then there exists an incoordination core in the reduced strategy space.*

Proof. As there is strategic ability after iterated reduction, there is at least one strategy for each agent in the reduced strategy space. Assume that all agents in the coalition only have equivalent strategies. Then there is joint ability. Thus, there is an agent with two incomparable strategies. □

Properties of SJAADL

Liu *et al.* (2020) analyze valid formulas in JAADL (Propositions 1-4), and give sufficient/necessary conditions for joint abilities (Theorems 1-4). The SJAADL variants of these results still hold, and the proofs are the same.

We now present two simple properties unique to SJAADL, where $\overline{A}$ denotes the complement of A. Intuitively, item 1 means k-bounded strategic ability implies strategic ability with a bigger bound. Item 2 means if other agents have a strategy to spoil the goal, then a coalition does not have k-bounded joint ability for any k. Note that $k_2 \geq k_1$ is not required, since the strategic power of $\overline{A}$ to act according to their k_1-bounded spoiling strategy is retained by the simple nondeterministic strategy $\{\top \to \top\}$.

Proposition 2. *The following are valid in SJAADL:*

1. $\langle\langle A\rangle\rangle^{\leq k_1}\psi \to \langle\langle A\rangle\rangle^{\leq k_2}\psi$*, where* $k_2 \geq k_1$.
2. $\langle\langle \overline{A}\rangle\rangle^{\leq k_1}\neg\psi \to \neg(\!(A)\!)^{\leq k_2,\infty}\psi$ *for all* k_1 *and* k_2.

An interesting question to explore is whether the formula $(\!(A)\!)^{\leq k_1,n}\psi \to (\!(A)\!)^{\leq k_2,n}\psi$ is valid when $k_2 \leq k_1$ or $k_2 \geq k_1$. Intuitively, $(\!(A)\!)^{\leq k_1,n}\psi \wedge \neg(\!(A)\!)^{\leq k_2,n}\psi$ might hold for $k_2 < k_1$ in two situations: first, there is not k_2-bounded strategic ability, but as we increase the bound and include more strategies we get k_1-bounded joint ability; second, there is a k_2-bounded incoordination core, but as we increase the bound and get a strongly dominating strategy, the core disappears. Similarly, $(\!(A)\!)^{\leq k_1,n}\psi \wedge \neg(\!(A)\!)^{\leq k_2,n}\psi$ might hold for $k_2 > k_1$ because a new core can appear with increased bound. We illustrate this with concrete examples.

Example 5. Consider two modified versions of the squirrel world in Example 1, shown in Figure 2. There is a wall limiting squirrels' moving action, represented by a solid bold line. We are concerned with the problem whether $\varphi = (\!(1,2)\!)^{\leq k}\langle\top;\top;\top;\top\rangle(\mathsf{hasAcorn}_1 \wedge \mathsf{hasAcorn}_2)$ holds for some k in the starting position.

In the left world, φ does not hold for $k = 4$, as δ_2^1 and δ_2^2 let δ_1^1 and δ_1^2 form an incoordination core. But for $k = 8$, φ holds, as a new strategy δ_1^d dominating both δ_1^1 and δ_1^2 lets squirrel 1 take a detour and get acorn 3: $\{\mathsf{seeAcorn}_1 \to \mathsf{pickUp}_1, \mathsf{hasAcorn}_1 \to \top, \neg\mathsf{seeAcorn}_1 \to \mathsf{w}_1 \vee \mathsf{s}_1, \neg(\mathsf{posW}_1 \vee \mathsf{posS}_1) \to \mathsf{n}_1\}$. The incoordination core is destroyed by this strategy.

In the right world, φ holds for $k = 4$, for only δ_1^2, δ_2^2 are available. But for $k = 8$, φ does not hold, because there are two new strategies δ_1^3 and δ_2^3 available, and the two strategies form an incoordination core.

Expressivity of Logics for Joint Abilities

In this section, we compare expressive power of JAADL and its variants, including SJAADL and a variant of it with deterministic structured strategies called SJAADL^d. It has the same syntax as SJAADL, but uses deterministic strategy space dss in its semantics.

We use classic definitions of distinguishing power and expressivity.

Definition 19. Let L_1 and L_2 be two logics interpreted over the same classes of models $\mathcal{M}$. We say that L_2 is at least as distinguishing as L_1, written $L_1 \preceq_d L_2$, if for every pair of models $M, M' \in \mathcal{M}$, we have that if there exists a formula $\varphi_1 \in L_1$ s.t. $M \models \varphi_1$ and $M' \not\models \varphi_1$, then there is also $\varphi_2 \in L_2$ s.t. $M \models \varphi_2$ and $M' \not\models \varphi_2$; We say that L_2 is at least as expressive as L_1, written $L_1 \preceq_e L_2$, if for every formula $\varphi_1 \in L_1$, there exists a formula $\varphi_2 \in L_2$ s.t. for every model $M \in \mathcal{M}$, we have $M \models \varphi_1$ iff $M \models \varphi_2$.

Clearly, $L_1 \preceq_e L_2$ implies $L_1 \preceq_d L_2$.

First, we prove that JAADL $\not\preceq_d$ SJAADL, by constructing two models that are distinguishable by JAADL but not by SJAADL. The main reason is that a structured strategy cannot react differently to two different paths with the same state labeling, thus an agent may have different combinatorial strategic abilities, but the same structural strategic abilities in different models.

Theorem 3. *JAADL* $\not\preceq_d$ *(therefore* $\not\preceq_e$*) SJAADL in both memoryless and memoryful semantics.*

Proof. Consider the models $\mathcal{G}_c$ and $\mathcal{G}'_c$ in Figure 3. There are $AP = \{p\}$, $AG = \{1, 2\}$, $AC = \{a, b\}$ available for both agents in every state.

We can prove no SJAADL formula can distinguish states $(\mathcal{G}_c, w_0)$, $(\mathcal{G}'_c, w_0)$ and $(\mathcal{G}'_c, w_1)$ via structural induction.

	σ_1^b	$\sigma_1^a(1)$	$\sigma_1^a(2)$	$\sigma_1^a(3)$	$\dots$
σ_2^b					
$\sigma_2^a(1)$					
$\sigma_2^a(2)$		✓			
$\sigma_2^a(3)$			✓		
$\sigma_2^a(4)$				✓	
$\vdots$					$\ddots$

	σ_1^b	σ_1^a
σ_2^b		
σ_2^r		✓
$\vdots$	$\vdots$	$\vdots$

Table 1: Memoryful combinatorial strategy spaces and compatible matrices in $\mathcal{G}_c$ (top) and $\mathcal{G}'_c$ (bottom).

The case of atoms is straightforward, as there is only one atom p, which takes the same value in all states. The cases of negations and disjunctions are trivial.

To prove the case of $\langle\langle A\rangle\rangle^{\leq k}\psi$ where no state subformula in ψ can distinguish the three states, note that in every state, the agents have the same set of actions, thus both agents have the same set of structured strategies available in all three states. Moreover, using the same structured strategy δ_{AG} in these states will give outcomes of equivalent paths. For example, for a path in $Out(w_0, \delta_{AG})$ of $\mathcal{G}'_c$, we can replace all states to w_0 in $\mathcal{G}_c$ to get a path in $Out(w_0, \delta_{AG})$ of $\mathcal{G}_c$. In addition, ψ takes the same value on equivalent paths. Therefore, $\langle\langle A\rangle\rangle^{\leq k}\psi$ takes the same value on states $(\mathcal{G}_c, w_0)$, $(\mathcal{G}'_c, w_0)$ and $(\mathcal{G}'_c, w_1)$.

To prove the case of $(A)^{\leq k}_{\psi}\varphi$ (and $(A)^{\leq k,\infty}_{\psi}\varphi$) where no state subformula in ψ and φ can distinguish the three states given the same strategy space, note that the reduction of strategy space is based on compatible sets, which are based on outcomes of strategies. As we have already shown, the same strategy results in outcomes of equivalent paths, and ψ evaluates the same on equivalent paths, thus results of reductions will be the same. Therefore, given the same structured strategy space, formula $(A)^{\leq k}_{\psi}\varphi$ (and $(A)^{\leq k,\infty}_{\psi}\varphi$) takes the same value on states $(\mathcal{G}_c, w_0)$, $(\mathcal{G}'_c, w_0)$ and $(\mathcal{G}'_c, w_1)$.

To prove JAADL can distinguish these two CGSs, we give a formula that takes different values on their initial states.

For memoryless JAADL, $\langle\langle 1\rangle\rangle\langle a_1; b_1\rangle p$ is true in $\mathcal{G}'_c$ but false in $\mathcal{G}_c$. The formula means agent 1 has a strategy to perform a then b and achieve p. There exists such memoryless strategy in $\mathcal{G}'_c$ but not in $\mathcal{G}_c$, as in $\mathcal{G}'_c$ agent 1 can choose different actions in different states.

For memoryful JAADL, consider the formula

$$((1,2))^{\infty}\langle (b_1 \wedge b_2)^*; a_1 \wedge b_2; a_2\rangle p,$$

where the goal means that agent 2's first a action must be performed one step after agent 1's first a action, e.g., if agent 1's first a is done at step t, then agent 2 should do its first a at step $t+1$. The whole formula means the two agents have joint ability w.r.t. the goal.

The strategy spaces and compatible matrices are shown in Table 1, where the check mark means combining two strategies can achieve the goal. In the tables, σ_i^b denotes the strategy that do b only, σ_i^a denotes strategies that do a at some step, $\sigma_i^a(t)$ denotes strategies that do a at step t. In $\mathcal{G}'_c$, agent 2 can choose to do a only after transition to w_1, which can only happen if 1 has done a. Such a strategy σ_2^r of doing a iff 1 did a in the last step will work with all σ_1^a, but not with σ_1^b. However, σ_1^b works with no strategies of 2 and is eliminated in the first round. Thus joint ability holds. Yet, such coordination cannot be done in $\mathcal{G}_c$, since σ_2^r does not exist as 2 cannot tell if 1 has done a. The only strategies that can be eliminated are σ_1^b, σ_2^b and $\sigma_2^a(1)$. The resulting strategy space has an incoordination core. □

Next, we prove SJAADL$^d \preceq_e$ JAADL. The main idea is that we can use a JAADL path formula to represent a deterministic structured strategy, by explicitly specifying that on the path, the agent always does the action allowed by the strategy. Hence, by using the strategy elimination operator of JAADL, we can restrict our attention to deterministic structured strategies of bounded complexity.

Theorem 4. *SJAADL$^d \preceq_e$ JAADL in both memoryless and memoryful semantics.*

Proof. We first show that the strategy space $dss^{\leq k}$ can be expressed in JAADL with a strategy space in which strategies that are not equivalent to one of the strategies in $dss^{\leq k}$ are eliminated.

Strategies in $dss_i^{\leq k}$ are in the form of $\delta_i^d = (\psi_j, \alpha_j)_{j=1}^n$ that satisfy the following conditions: First, in the memoryless case, no state will satisfy two different ψ_j; In the memoryful case, no history will satisfy two different ψ_j. Second, every α_j is a single action atom.

For a propositional formula ϕ and an action formula α of a pair (ϕ, α) in a deterministic memoryless structured strategy, we write their corresponding path formula $trP_r(\phi, \alpha) = \neg\langle \top^*; \phi \wedge \neg\alpha\rangle\top$. The formula means there does not exist a path in which ϕ is satisfied somewhere but the corresponding action does not satisfy α.

For an LDL$_f$ formula ψ and an action atom α of a pair (ψ, α) in a deterministic memoryful structured strategy, we can also write such a corresponding path formula. To do so, we first translate ψ into a corresponding RE$_f$ formula $\varrho ::= \phi \mid \varrho_1 + \varrho_2 \mid \varrho_1; \varrho_2 \mid \varrho^*$ using method in (De Giacomo and Vardi 2013), where ϱ^* means ϱ is repeated for zero or more times. We then translate the formula into regular expressions under an alternative syntax $\varrho ::= \phi \mid \varrho_1 + \varrho_2 \mid \varrho_1; \varrho_2 \mid \varrho^+$, where ϱ^+ means ϱ is repeated for one or more times. This translation can be done by rewritting each ϱ^* into $\epsilon + \varrho^+$, where ϵ means an empty path, and reorganizing the formula to move all ϵ into top level. In the progress, all ϵ appearing in $\varrho_1; \epsilon$ or $\epsilon; \varrho_2$ can be removed. Finally, if the top level expression is in the form of $\epsilon + \varrho$, ϵ can also be removed since a strategy doesn't apply to an empty path.

Finally, we write the corresponding path expression of this RE$_f$ formula with alternative syntax ϱ and action formula α, denoted $trP_R(\varrho, \alpha)$, recursively as follows:

- $trP_R(\phi, \alpha) = \phi \wedge \neg\alpha$,

- $trP_R(\varrho_1 + \varrho_2, \alpha) = trP_R(\varrho_1, \alpha) + trP_R(\varrho_2, \alpha)$,
- $trP_R(\varrho_1; \varrho_2, \alpha) = \varrho_1; trP_R(\varrho_2, \alpha)$,
- $trP_R(\varrho^+, \alpha) = \varrho^*; trP_R(\varrho, \alpha)$.

The corresponding path formula of the original pair (ψ, α) is written as $trP_R(\psi, \alpha) = \neg\langle trP_R(\varrho, \alpha)\rangle\top$, where ψ is rewritten as the RE$_f$ ϱ with alternative syntax. The formula means there does not exist a path in which ψ is satisfied somewhere but the corresponding action does not satisfy α.

For a deterministic structured strategy $\delta_i^d = (\psi_j, \alpha_j)_{j=1}^n$, we write its corresponding path formula as $\psi_{\delta_i^d} = \bigwedge_{j=1}^n trP(\psi_j, \alpha_j)$. Here we use trP_r (resp. trP_R) if δ_i is a memoryless (resp. memoryful) strategy. The formula is made by conjoining all corresponding formulas of condition-action pairs of δ_i^d. It describes the set of paths where i's strategy is equivalent to δ_i^d.

For the deterministic structured strategy space $dss_i^{\leq k}$, we write its corresponding path formula as $\psi_i^{\leq k} = \bigvee_{\delta_i^d \in dss_i^{\leq k}} \psi_{\delta_i^d}$. The formula, constructed by disjoining all such formulas of strategies in $dss_i^{\leq k}$, describes the set of paths where i's strategy is equivalent to one of the strategies in $dss_i^{\leq k}$. This is possible as $dss_i^{\leq k}$ is finite.

For the collective deterministic structured strategy space $dss_A^{\leq k}$, we write its corresponding path formula as $\psi_A^{\leq k} = \bigwedge_{i \in A} \psi_i^{\leq k}$. The formula is made by conjoining all such formulas of agents in coalition A. This is the formula we need: It is a path formula with no strategic modalities, meaning that on this path, coalition A's strategy is equivalent to one of the strategies in $dss_A^{\leq k}$.

Finally, for each SJAADL^d formula φ, we translate it into an equivalent JAADL formula $tr(\varphi)$ recursively, by reducing the combinatorial strategy space before considering strategic abilities and joint abilities.

- $tr(p) = p, tr(\neg\varphi) = \neg tr(\varphi)$,
- $tr(\varphi_1 \wedge \varphi_2) = tr(\varphi_1) \wedge tr(\varphi_2)$,
- $tr(\langle\langle A\rangle\rangle^{\leq k}\psi) = (AG)_{\psi_{AG}^{\leq k}} \langle\langle A\rangle\rangle\psi$,
- $tr((A)_\psi^{\leq k}\varphi) = (AG)_{\psi_{AG}^{\leq k}} (A)_\psi\varphi$,
- $tr((A)_\psi^{\leq k,\infty}\varphi) = (AG)_{\psi_{AG}^{\leq k}} (A)_\psi^\infty\varphi$.

Intuitively, this means that in the translated JAADL formulas, when strategic abilities are considered, we consider it in the reduced strategy space where each strategy is equivalent to a deterministic structured strategy with complexity $\leq k$.

The translation is exponential in k in the memoryless case, as it enumerates all strategies in $dss^{\leq k}$, which has size exponential in k. It is double exponential in k in the memoryful case, as LDL$_f$ to RE$_f$ translation is double exponential. □

However, the proof of Theorem 4 does not work with SJAADL as nondeterministic structured strategies cannot be described using linear-time path formulas as in the proof. Further, it is unclear how to simulate the reduction of nondeterministic strategies with that of deterministic ones in JAADL. Therefore, it remains open whether SJAADL is less expressive than JAADL.

Algorithm 1: Labeling State Formulas

function $Label(\mathcal{G}, ss, \varphi)$:
if $\varphi = p$ **then** $[\varphi]_{ss} \leftarrow \{w \in W \mid p \in L(w)\}$
else if $\varphi = \neg\varphi_1$ **then** $[\varphi]_{ss} \leftarrow W - [\varphi_1]_{ss}$
else if $\varphi = \varphi_1 \wedge \varphi_2$ **then** $[\varphi]_{ss} \leftarrow [\varphi_1]_{ss} \cap [\varphi_2]_{ss}$
else if $\varphi = \langle\langle A\rangle\rangle^{\leq k}\psi$ **then**
  $(\mathcal{G}', \psi') \leftarrow ReduceP(\mathcal{G}, ss^{\leq k}, \psi)$;
  $[\varphi]_{ss} \leftarrow \{w \mid \exists\delta_A \in ss_A^{\leq k}.\forall\delta_{-A} \in ss_{-A}^{\leq k}.$
    $PathF(\mathcal{G}', w, (\delta_A, \delta_{-A}), \psi')\}$
else if $\varphi = (A)_\psi^{\leq k}\varphi_1$ **then**
  $(\mathcal{G}', \psi') \leftarrow ReduceP(\mathcal{G}, ss^{\leq k}, \psi)$;
  $[\varphi]_{ss} \leftarrow \{w \mid w \in [\varphi_1]_{StrS(\mathcal{G}', A, \psi', w, ss^{\leq k})}\}$
else if $\varphi = (A)_\psi^{\leq k,\infty}\varphi_1$: **then**
  $(\mathcal{G}', \psi') \leftarrow ReduceP(\mathcal{G}, ss^{\leq k}, \psi)$;
  $[\varphi]_{ss} \leftarrow \{w \mid w \in [\varphi_1]_{StrS^\infty(\mathcal{G}', A, \psi', w, ss^{\leq k})}\}$
return $[\varphi]_{ss}$

Algorithm 2: Reducing Path Formulas

function $ReduceP(\mathcal{G}, ss, \psi)$:
Max(ψ) ← the set of maximal state-subformulas in ψ;
for each $\varphi \in$ Max(ψ) **do**
  define a fresh atom p_φ s.t. $p_\varphi \in L(w)$ for $w \in [\varphi]_{ss}$;
replace $\varphi \in$ Max(ψ) in ψ by p_φ to get LDL formula ψ';
return $\mathcal{G}', \psi'$

Model Checking

Liu *et al.* (2020) propose a model-checking algorithm for memoryless JAADL with time exponential in both the model size and formula size. But whether model checking memoryful JAADL is decidable remains open. In this section, we give model-checking algorithms for both memoryful and memoryless SJAADL. We show that the complexity of both algorithms is exponential in the formula size and polynomial in the model size, and with the complexity bound k of structured strategies, exponential in the memoryless case and double exponential in the memoryful case.

We first state the model-checking problem for SJAADL. We say that an SJAADL formula φ has complexity bound b, if the largest k-superscript in φ is b.

Definition 20 (SJAADL model checking)**.** Given a CGS $\mathcal{G}$, an SJAADL formula φ with complexity bound k, a full structured strategy space with bounded complexity $fss^{\leq k}$, the problem is to find states w s.t. $w, fss^{\leq k} \vDash \varphi$.

In this section, we use n for the size of $\mathcal{G}$, l for the size of φ, and k for the complexity bound of φ. We denote k-bounded memoryless structured strategy space by $ss^{\leq k,r}$, and the memoryful one by $ss^{\leq k,R}$.

We first give Algorithm 1 for labeling state formulas. Given a CGS $\mathcal{G} = \langle W, L, P, \tau, w^0\rangle$, a structured strategy space ss and a SJAADL state formula φ, Algorithm 1 returns $[\varphi]_{ss} = \{w \in W \mid w, ss \vDash \varphi\}$ by recursively labeling the subformulas. In the cases of $\langle\langle A\rangle\rangle^{\leq k}\psi$, $(A)_\psi^{\leq k}\varphi_1$ and $(A)_\psi^{\leq k,\infty}\varphi_1$, Algorithm 1 first calls Algorithm 2 to reduce

Algorithm 3: Calculating Reduced Strategy Space

function $StrS(\mathcal{G}, A, \psi, w, ss)$:
for each $i \in A$ **do**
  for each $\delta_i \in ss_i$ **do**
    for each $\delta_{-i} \in ss_{-i}$ **do**
      if $PathF(\mathcal{G}, w, (\delta_i, \delta_{-i}), \psi)$ **then**
        add δ_{-i} to $M_{\psi,w,ss}(\delta_i)$
  for each $\delta_i, \delta_i' \in ss_i$ **do**
    if $M_{\psi,w,ss}(\delta_i) \supset M_{\psi,w,ss}(\delta_i')$ **then**
      $ss_i \leftarrow ss_i - \{\delta_i'\}$
return ss

Algorithm 4: Model Checking Path Formulas in the r Case

function $PathF(\mathcal{G}, w, r\delta_{AG}, \psi)$:
Prune $\mathcal{G}$ with $r\delta_{AG}$ to get a Kripke structure $\mathcal{K}$
return whether $w \vDash_{ldl} \psi$ in $\mathcal{K}$

the model checking of the path formula ψ w.r.t. $\mathcal{G}$ to that of a pure LDL formula ψ' w.r.t. an extended CGS $\mathcal{G}'$. The idea is to first label each maximal state-subformula φ of ψ, call Algorithm 1 to compute $[\varphi]_{ss}$, introduce a fresh atom p_φ to represent φ, and so get $\mathcal{G}'$ and ψ'.

Algorithm 3 $StrS(A, \psi, w, ss)$ reduces ss w.r.t. coalition A, goal ψ and state w. $StrS^\infty$ is calculated by calling $StrS$ repeatedly until reaching a fixed point.

Both Algorithms 1 and 3 call $PathF$ to model-check a path formula. In JAADL, when checking a path formula ψ w.r.t. a state w and a collective strategy σ_{AG}, first use the idea in Algorithm 2 to get a pure LDL formula ψ_{ldl}. The infinite path $out(w, \sigma_{AG})$ can be viewed as a finite Kripke model with initial state w, denoted as $Kripke(w, \sigma_{AG})$. To verify whether $Kripke(w, \sigma_{AG}) \models_{ldl} \psi_{ldl}$, call the LDL model-checking algorithm (Faymonville and Zimmermann 2017), which is in time linear in the model size and exponential in the formula size. We now show that for SJAADL, $PathF$ can also be reduced to LDL model checking.

Memoryless SJAADL

In memoryless SJAADL, as decisions of a coalition are based only on the current state, given a state w and a collective structured strategy $r\delta_{AG}$, we can construct a Kripke structure from a CGS $\mathcal{G}$ by keeping the necessary transitions.

Lemma 5. *$PathF$ for memoryless SJAADL can be reduced to model checking an LDL formula with size $\mathcal{O}(l)$ on a Kripke structure with size $\mathcal{O}(n)$.*

Proof. We first construct a pure LDL formula ψ_{ldl}. Given a memoryless strategy $r\delta_{AG}$ and a state w, we can construct a Kripke structure $\mathcal{K}$ as follows: at any state w_0, for an agent i, her possible actions $r\delta_i(w_0) \subseteq P_i(w_0)$, and thus we keep only those necessary transitions. $\mathcal{K}$ has size at most n and we have $PathF(\mathcal{G}, w, r\delta_{AG}, ss^{\leq k,r}, \psi)$ iff $\mathcal{K} \vDash_{ldl} \psi_{ldl}$. □

Memoryless $PathF$ algorithm is given as Algorithm 4.

Theorem 6. *Model checking memoryless SJAADL can be done in time exponential in k and l, and polynomial in n.*

Proof. We prove that $Label$ takes $\mathcal{O}(n^l 2^k 2^l)$ time by induction on l. We only show the two most complex cases: processing $\langle\langle A\rangle\rangle^{\leq k}\psi$ and $(A)^{\leq k}_{\psi}\varphi$, both requiring calling $ReduceP$, enumerating strategies and calling $PathF$. Let $l_1 = |\psi|$, and $l_2 = |\varphi|$. Assume that processing a subformula φ' takes $\mathcal{O}(n^{l'} 2^k 2^{l'})$ time, where $l' = |\varphi'|$. Then calling $ReduceP$ takes $\mathcal{O}(n^{l_1} 2^k 2^{l_1})$ time. There are $\mathcal{O}(2^k)$ structured strategies, and calling $PathF$ takes $\mathcal{O}(n 2^{l_1})$ time by Lemma 5, hence calling $StrS$ takes $\mathcal{O}(n 2^k 2^{l_1})$ time. Processing $(A)^{\leq k}_{\psi}\varphi_1$ requires calling $StrS$ and $Label$ for each state, which takes $\mathcal{O}(n(n2^k 2^{l_1} + n^{l_2} 2^k 2^{l_2})) = \mathcal{O}(n^l 2^k 2^l)$ time. Hence processing each case takes $\mathcal{O}(n^l 2^k 2^l)$ time. □

Note that the model-checking algorithm for memoryless JAADL is exponential in n, while our model-checking algorithm for memoryless SJAADL is polynomial in n. The main reason is that the size of the strategy space for memoryless JAADL is exponential in n, while for memoryless SJAADL the size is exponential in k. Thus, our model-checking algorithm for memoryless SJAADL has better complexity than that for memoryless JAADL.

Memoryful SJAADL

In memoryful SJAADL, as decisions of a coalition are based on the whole history, given a state w and a collective structured strategy $R\delta_{AG}$, when we construct a Kripke structure from a CGS $\mathcal{G}$, we have to use extra states to memorize the history. For an agent i, δ_i has at most k LDL_f conditions. Each LDL_f condition can be transformed to an equivalent finite automaton. So for agent i, she can use the states of at most k automata to memorize the history.

Lemma 7. *$PathF$ for memoryful SJAADL can be reduced to model checking an LDL formula with size $\mathcal{O}(l)$ on a Kripke structure with size $\mathcal{O}(n2^{2^k})$.*

Proof. We first construct a pure LDL formula ψ_{ldl}. Each LDL_f formula ψ in $R\delta_{AG}$ can be converted to an equivalent alternating finite automata on words (AFW) $\mathcal{A}_\psi$ with at most $2|\psi|$ states (De Giacomo and Vardi 2013), the exact state set is the Fisher-Ladner closure (Harel, Pnueli, and Stavi 1983) of ψ. An AFW with m states can be converted to an equivalent DFA with at most 2^{2^m} states (Chandra, Kozen, and Stockmeyer 1981). Given a memoryful strategy $R\delta_{AG}$ and a state w, we can construct a Kripke structure $\mathcal{K}$ as follows: at any history h, for an agent i, after input h, her possible

Algorithm 5: Model Checking Path Formulas in the R Case

function $PathF(\mathcal{G}, w, R\delta_{AG}, \psi)$:
for each ψ^i_j in $R\delta_{AG}$ **do**
  construct AFW $\mathcal{A}^i_j$ from ψ^i_j;
  convert $\mathcal{A}^i_j$ to DFA $\mathcal{D}^i_j$;
  $W^i_j \leftarrow$ the set of states of $\mathcal{D}^i_j$
$W \leftarrow$ the set of states of $\mathcal{G}$;
construct a Kripke structure $\mathcal{K}$ by joining $\mathcal{G}$ and all $\mathcal{D}^i_j$;
return whether $w \vDash_{ldl} \psi$ in $\mathcal{K}$

actions depend only on the current state of each DFA. Thus, we represent $\mathcal{K}$'s states by states of $\mathcal{G}$ and all DFAs. The initial state s_0 of $\mathcal{K}$ is composed of the initial state w_0 of $\mathcal{G}$, and for each DFA, the w_0-successor (*i.e.*, successor via edge w_0) of the initial state. Let s be a state of $\mathcal{K}$ composed from state w of $\mathcal{G}$ and state $w_{\mathcal{D}}$ for each DFA $\mathcal{D}$. Let d be a possible decision at w. Let s' be composed from w', the d-successor of w, and for each DFA $\mathcal{D}$, the w'-successor of $w_{\mathcal{D}}$. Then s' is a succesor of s in $\mathcal{K}$. $R\delta_A$ needs $\mathcal{O}(k)$ DFAs with $\mathcal{O}(2^{2^k})$ states each, therefore $\mathcal{K}$ has $\mathcal{O}(n2^{2^k})$ states, and we have $PathF(\mathcal{G}, w, R\delta_{AG}, ss^{\leq k,r}, \psi)$ iff $\mathcal{K} \vDash_{ldl} \psi_{ldl}$. □

We call the Kripke structure constructed in the proof the joining of $\mathcal{G}$ and all DFAs. Memoryful $PathF$ algorithm is given as Algorithm 5.

Theorem 8. *Model checking memoryful SJAADL can be done in time double exponential in k, exponential in l, and polynomial in n.*

Proof. We prove $Label$ takes $\mathcal{O}(n^l 2^{2^k} 2^l)$ time by induction on l. The proof is similar to the memoryless case, except that calling $PathF$ takes $\mathcal{O}(n2^{2^k}2^l)$ time by Lemma 7. □

Norm Synthesis

In this section, we introduce the problem of norm synthesis, and solve it with SJAADL model checking.

We have presented cases in which a coalition has strategic ability, but has no joint ability despite agents' coordinating intention. In some cases, adding a rule to enforce the coalition follows the rule can ensure joint ability. We call the rule a social norm, and the problem of finding it norm synthesis. In this paper, we consider the memoryless case. The memoryful case is more complex and is left for future work. We first state it formally for SJAADL:

Definition 21 (norm synthesis)**.** Given a CGS $\mathcal{G}$, a state w, an SJAADL formula φ in the form of $\langle\!\langle A\rangle\!\rangle^{\leq k,\infty}\langle\top^*\rangle\psi$ or $\langle\!\langle A\rangle\!\rangle^{\leq k,\infty}[\top^*]\psi$, a k-bounded memoryless strategy space ss s.t. $w, ss \nvDash \varphi$, the problem is to find a propositional formula ϕ s.t. for the corresponding formula φ' in the form of $\langle\!\langle A\rangle\!\rangle^{\leq k,\infty}\langle\phi^*\rangle\psi$ or $\langle\!\langle A\rangle\!\rangle^{\leq k,\infty}[\phi^*]\psi$, we have $w, ss \vDash \varphi'$.

Then, we give two algorithms for the case $\varphi = \langle\!\langle A\rangle\!\rangle^{\leq k,\infty}\langle\top^*\rangle\psi$. The other case is dealt similarly.

The first algorithm (Algorithm 6) enumerates all possible ϕ with a fixed size bound, and checks whether each of them fulfills the requirement with SJAADL model checking.

Proposition 9. *Algorithm 6 is sound and complete if there is a solution of the given bound, and runs in time exponential w.r.t. the bound.*

The second algorithm (Algorithm 7) is based on Proposition 1 that if a coalition has strategic ability but no joint ability, there will be an incoordination core. The algorithm uses $StrSM^\infty$, which is the same as $StrS^\infty$ except it also returns the compatible sets of each remaining strategy (written as M). For $\delta_i = (\phi_j, \alpha_j)_{j=1}^n$, let $\phi_d = \neg(\phi_p \to \alpha_p)$ for some $p \leq n$, then ϕ_d disables a condition-action pair of δ_i, we call ϕ_d a disabler of δ_i. Although the complexity is not improved, the exhaustive search in Algorithm 6 is avoided.

Algorithm 6: Exhaustive Search for Norm Synthesis

```
function Search(A, φ, w, ss, G, bound):
for each ϕ that |ϕ| ≤ bound do
    if w ∈ Label(G, ss, ⟨⟨A⟩⟩^{≤k,∞}⟨ϕ*⟩ψ) then
        return ϕ
return ⊥
```

Algorithm 7: Local Search for Norm Synthesis

```
function LSearch(A, φ, w, ss, G, bound):
ϕ ← ⊤
while |ϕ| ≤ bound do
    ss, M ← StrSM^∞(A, ⟨ϕ*⟩ψ, w, ss);
    return ϕ if there is joint ability;
    return ⊥ if there is no strategic ability;
    choose a core (δ_i, δ'_i);
    construct a disabler ϕ_d of δ_i;
    ϕ ← ϕ ∧ ϕ_d
return ⊥
```

Proposition 10. *Algorithm 7 is sound, and runs in time exponential w.r.t. the bound.*

Example 6. Consider the scenario in Example 4, where $\langle\!\langle 1,2\rangle\!\rangle^{\leq 4,\infty}\langle\top^*\rangle(\mathsf{hasAcorn}_1 \wedge \mathsf{hasAcorn}_2)$ does not hold. Here δ_1^1 and δ_1^2 are in an incoordination core. Algorithm 7 lets ϕ be $\neg(\neg\mathsf{seeAcorn}_1 \to \mathsf{e}_1)$, which disables δ_1^1 and destroys the core, enabling 4-bounded joint ability. The new formula $\langle\!\langle 1,2\rangle\!\rangle^{\leq 4,\infty}\langle(\neg(\neg\mathsf{seeAcorn}_1 \to \mathsf{e}_1))^*\rangle(\mathsf{hasAcorn}_1 \wedge \mathsf{hasAcorn}_2)$ holds.

Conclusions

In this paper, we have proposed SJAADL, a modal logic for joint abilities of structured strategies with bounded complexity. Compared with JAADL, SJAADL has the following advantages. First, in contrast to combinatorial or deterministic strategies, the notion of our structured strategies is a better representation of human strategies, especially in the memoryful case where a strategy is combinatorially more complex. Secondly, our algorithms for model-checking SJAADL have complexity polynomial in the model size, and thus excel over the model-checking algorithm for memoryless JAADL. Thirdly, in SJAADL, we can formalize the problem of synthesizing norms to achieve joint ability, and give preliminary algorithms for it. In the future, we are interested in solving the expressivity problem in the nondeterministic case, finding non-trivial lowerbounds for model-checking both memoryless and memoryful SJAADL, formalizing more natural and restricted ways of reasoning about coordination, as well as further exploration of norm synthesis and its implementation.

Acknowledgments

We thank the anonymous reviewers for helpful comments. We acknowledge support from the Natural Science Foundation of China under Grant No. 62076261.

References

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

Chandra, A. K.; Kozen, D. C.; and Stockmeyer, L. J. 1981. Alternation. *J. ACM*, 28(1): 114–133.

De Giacomo, G.; and Vardi, M. Y. 2013. Linear Temporal Logic and Linear Dynamic Logic on Finite Traces. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, 854–860. AAAI Press. ISBN 9781577356332.

Faymonville, P.; and Zimmermann, M. 2017. Parametric linear dynamic logic. *Information and Computation*, 253: 237–256.

Ghaderi, H.; Levesque, H.; and Lespérance, Y. 2007. Towards a logical theory of coordination and joint ability. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, 1–3.

Harel, D.; Pnueli, A.; and Stavi, J. 1983. Propositional dynamic logic of nonregular programs. *Journal of Computer and System Sciences*, 26(2): 222–243.

Jamroga, W.; Malvone, V.; and Murano, A. 2017. Reasoning about natural strategic ability. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, 714–722.

Jamroga, W.; Malvone, V.; and Murano, A. 2019. Natural strategic ability. *Artificial Intelligence*, 277: 103170.

Liu, Z.; Xiong, L.; Liu, Y.; Lespérance, Y.; Xu, R.; and Shi, H. 2020. A Modal Logic for Joint Abilities under Strategy Commitments. In *IJCAI*, 1805–1812.

Osborne, M. J.; and Rubinstein, A. 1994. *A Course in Game Theory*. The MIT Press. London, England: MIT Press.

Ramanujam, R.; and Simon, S. E. 2008. Dynamic Logic on Games with Structured Strategies. In *KR*, 49–58.

van Eijck, J. 2013. PDL as a multi-agent strategy logic. In *Proceedings of the 14. Conference on Theoretical Aspects of Rationality and Knowledge*, 206–215.