A Logic for Reasoning About Game Strategies

Dongmo Zhang
The University of Western Sydney
Australia
d.zhang@uws.edu.au

Michael Thielscher
The University of New South Wales
Australia
mit@cse.unsw.edu.au

Abstract

This paper introduces a modal logic for reasoning about game strategies. The logic is based on a variant of the well-known game description language for describing game rules and further extends it with two modalities for reasoning about actions and strategies. We develop an axiomatic system and prove its soundness and completeness with respect to a specific semantics based on the state transition model of games. Interestingly, the completeness proof makes use of forgetting techniques that have been widely used in the KR&R literature. We demonstrate how general game-playing systems can apply the logic to develop game strategies.

Introduction

General Game Playing is concerned with replicating the ability of humans to learn to play new games solely by being given the rules. The most challenging task of a general game-playing system is to generate strategies for game playing. Several approaches have been developed in general game playing, including automatically generated heuristics (Clune 2007; Schiffel and Thielscher 2007), symbolic search (Kissmann and Edelkamp 2008), and Monte Carlo-based search (Finnsson and Björnsson 2008); for an overview, see (Genesereth and Björnsson 2013; Genesereth and Thielscher 2014). However, for most games these approaches are too general to be as effective as a simple game-specific strategy. The great challenge is how to automatically generate specific strategies for specific games based on a players' understanding of the game rules. Ideally, a player should be able to think logically about game strategies based on the provided logical descriptions of the game rules.

As a quasi-standard in general game playing, game rules are specified in the *Game Description Language* (GDL) (Genesereth *et al.* 2005). The language is a fragment of first-order logic, which is essentially propositional but expressive enough for describing any finite combinatorial games with complete information. GDL has been designed solely for describing game rules but not for describing game strategies, nor for reasoning about game strategies.

There exist a number of logical frameworks proposed for reasoning about strategic abilities of players. Most of them were developed based on *Alternating-time Temporal Logic* (ATL) (Alur *et al.* 2002; van der Hoek *et al.* 2005; Chatterjee *et al.* 2010; Walther *et al.* 2007). However, these logics "*tend to have existential quantifiers saying that 'players have a strategy' for achieving some purpose, while descriptions of these strategies themselves are not part of the logical language*" (van Benthem 2008). In fact, reasoning about strategic abilities of players is different from reasoning about strategies themselves. The former takes a global view of a group of players and does not consider how individual players generate strategies.

In this paper we introduce a modal logic for representing and reasoning about game strategies. We extend the game description language with three modalities for representing and reasoning about actions and game strategies. We establish an axiomatic system for the logic and prove its soundness and completeness using variable forgetting technique. We demonstrate how to the logic can be used for creating strategies and verifying properties of strategies.

The paper is arranged as follows. The 2^{nd} section introduces the model we use for describing a game. The 3^{rd}-5^{th} sections present the syntax, semantics and axiomatic system of the logic, followed by a sketch of completeness proof. The last few sections demonstrate how to use the logic for strategy representation and reasoning before conclude the work.

State Transition Model for Games

All the games we consider in this paper are assumed to be played in a multi-agent environment. We use N to represent the set of agents. We assume that each agent in N can perform a finite but non-empty number of actions. Let A_i be the set of actions agent $i \in N$ can perform. We assume that actions are different if they are performed by different agents even though they may have same pre-conditions and post-conditions. Formally, we call $(N, \mathcal{A})$ a *multi-agent frame* where

- N is a non-empty, finite set of agents;
- $\mathcal{A} = \bigcup_{i \in N} A^i$, where A^i is a non-empty, finite set of *actions* for player $i \in N$ and $A_i \cap A_j = \varnothing$ if $i \neq j$.

Most strategic games we play in the real-world, such as board games, are finite and asynchronous. These games can be easily specified with the state transition model.

Definition 1 *Given a multi-agent frame $\mathcal{F} = (N, \mathcal{A})$, a* state transition game*, or shortly* game*, is a tuple* (W, I, T, U, L)*, where*

- *W is a non-empty set of states (possible worlds);*
- *$I \subseteq W$ is the* initial *states;*
- *$T \subseteq W$ is the* terminal *states.*
- *$U : W \times \mathcal{A} \mapsto W \backslash I$ is the* update function*, which maps each pair of state and action to a non-initial state.*
- *$L \subseteq W \times \mathcal{A}$ is a* legality relation *(describing the allowed actions in a state);*

As mentioned above, we only consider asynchronous games, in the sense that only one action can be taken at one time. This can be seen from the definition of update function. If we want to specify synchronous games, the update function has to be extended to take a vector of actions from all the players in each state as shown in (Genesereth *et al.* 2005). We leave that option for future investigations.

Slightly different from game models in the GGP literature, we do not assume that initial states are unique. We also leave the $wins(.)$ operator to users to define. These simplicities make our axioms look neat. For convenience, we will write $u_a(w) =_{def} U(w, a)$.

For any $w \in W$ and $a \in \mathcal{A}$, we call (w, a) a *move*. It is a *legal move* if $(w, a) \in L$. It is a *move of player i* if $a \in A^i$.

Any set, S, of legal moves is called a *strategy*, which specifies which action is taken at some states. We call a strategy S *player i's strategy* if it contains only player i's moves.

We say that a strategy S of player i is *complete* if for each state $w \in W$, there is a legal move $(w, a^i) \in S$ unless $l^i(w) = \emptyset$, where $l^i(w) = \{a \in A^i : (w, a) \in L\}$. Roughly speaking, a complete strategy of a player provides the player with a "complete" guideline so that she can move whenever she has a legal move. A strategy S of player i is *deterministic* if for any $(w, a) \in S$ and $(w, a') \in S$, $a, a' \in A^i$ implies $a = a'$. A strategy is *functional* if it is complete and deterministic.

The Syntax

Describing a game with the state transition model is possible but not practical in all cases. We may encode a game in a form that is more compact. Game Description Language (GDL) is a logical language for describing game rules, proposed as the official language for General Game Playing Competition (Genesereth *et al.* 2005). The original GDL is a variant of Datalog, which is a fragment of first-order language. For the purpose of this paper, we reformat it in propositional modal logic.

Definition 2 Given a multi-agent frame $\mathcal{F} = (N, \mathcal{A})$, let $\mathcal{L}^{GDL}$ be a propositional modal language consisting of

- a countable set, Φ, of propositional variables;
- logical connectives $\neg$ and $\wedge$ (from which $\vee$, $\rightarrow$, $\leftrightarrow$, $\top$, $\bot$ can be defined as usual);
- pre-defined propositions: *initial*, *terminal*, $legal(a)$ and $does(a)$, for each $a \in \mathcal{A}$;
- action execution modality $[\,.\,]$.

The following BNF rule generates all formulas in $\mathcal{L}^{GDL}$:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid initial \mid terminal \mid does(a) \mid legal(a) \mid [a]\varphi \mid$$

where $p \in \Phi$ and $a \in \mathcal{A}$.

As in GDL, *initial* and *terminal* are specific propositional variables, representing the initial states and the terminal states of a game, respectively. For any action $a \in \mathcal{A}$, $does(a)$ is a propositional variable stands that the action a is to be taken at the current state while $legal(a)$ means that it is legal to perform a at the current state[1]. Different from the standard GDL, instead of using the temporal operator $\bigcirc$ to specify effects of actions, we introduce the PDL-like action modality $[.]$ and use it to define the next operator:

$$\bigcirc\varphi =_{def} \bigvee_{a \in \mathcal{A}} (does(a) \wedge [a]\varphi) \tag{1}$$

Note that $[a]\varphi$ means if action a is executed at the current state, φ will be true in the next state.

Note that the language $\mathcal{L}^{GDL}$ reliefs on the multi-agent frame $\mathcal{F}$. For simplicity without losing generality, in the rest of the paper we will fix a multi-agent frame $\mathcal{F} = (N, \mathcal{A})$ and all concepts will be based on the same multi-agent frame. Now we extend the language with two additional modalities to represent choice of actions and effects of strategies.

Definition 3 Let $\mathcal{L}^{GDL^+}$ be the extension of $\mathcal{L}^{GDL}$ augmented by the action selection operator $\lfloor \cdot \rfloor$ and strategy modality $\wr \cdot \int$. The formulas of $\mathcal{L}^{GDL^+}$ are generated by the following BNF rule:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid initial \mid terminal \mid does(a) \mid legal(a) \mid [a]\varphi \mid \lfloor a \rfloor\varphi \mid \wr\varphi\int\varphi \mid$$

where $p \in \Phi$ and $a \in \mathcal{A}$.

Intuitively, $\lfloor a \rfloor\varphi$ means that if action a were chosen (but not yet executed), then φ would be true. $\wr\alpha\int\varphi$ means that if player i were to play according to strategy α, then φ would be true.

Interestingly, the action selection modality allows us to define the following two binary modal operators:

- Prioritised Disjunction:

$$\alpha \mathbin{\bigtriangledown} \beta =_{def} \alpha \vee (\beta \wedge \bigwedge_{c \in \mathcal{A}} \lfloor c \rfloor \neg\alpha) \tag{2}$$

- Prioritised Conjunction:

$$\alpha \mathbin{\bigtriangleup} \beta =_{def} \alpha \wedge ((\bigvee_{c \in \mathcal{A}} \lfloor c \rfloor(\alpha \wedge \beta)) \rightarrow \beta) \tag{3}$$

We will show that if α and β are two strategies for a player, $\alpha \mathbin{\bigtriangledown} \beta$ and $\alpha \mathbin{\bigtriangleup} \beta$ are also strategies of the player. $\alpha \mathbin{\bigtriangledown} \beta$ means "*apply strategy α if it is applicable; otherwise, apply β*". $\alpha \mathbin{\bigtriangleup} \beta$ means "*apply α and β if both applicable; otherwise, apply α only*". As we will demonstrate later, these connectives play important roles in strategy composition.

To show how to use the language to represent game rules and related strategies, we consider the following simple game (van den Herik *et al.* 2002).

[1] Since we assume that different players take different actions, it is clear who takes the action.

Example 1 *(mk-**Game**) An mk-game is a board game in which two players take turns in marking either a nought 'o' or a cross 'x' on a board with m × m grid. The player who first gets k consecutive marks of her own symbol in a row (horizontally, vertically, or diagonally), will win the game. It is assumed that player 'x' makes the first move.*

It is easy to see that Tic-Tac-Toe is a 3, 3-game and the standard Gomoku game is a 15, 5-game. Now let us describe the *mk*-games in our language.

Given a player $t \in \{\mathsf{x}, \mathsf{o}\}$, let $p^t_{i,j}$ denote that grid (i, j) is filled with player t's symbol, and $a^t_{i,j}$ denote the action that player t marks grid (i, j), where $1 \leqslant i, j \leqslant m$. The game can then be expressed with the following domain axioms:

1. $initial \to turn(\mathsf{x}) \wedge \neg turn(\mathsf{o}) \wedge \bigwedge_{i,j=1}^{m} \neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j})$
2. $wins(t) \leftrightarrow (\bigvee_{i=1}^{m} \bigvee_{j=1}^{m-k+1} \bigwedge_{l=0}^{k-1} p^t_{i,j+l}) \vee (\bigvee_{i=1}^{m-k+1} \bigvee_{j=1}^{m} \bigwedge_{l=0}^{k-1} p^t_{i+l,j})$
   $\vee (\bigvee_{i=1}^{m-k+1} \bigvee_{j=1}^{m-k+1} \bigwedge_{l=0}^{k-1} p^t_{i+l,j+l}) \vee (\bigvee_{i=1}^{m-k+1} \bigvee_{j=k}^{m} \bigwedge_{l=0}^{k-1} p^t_{i+l,j-l})$
3. $teminal \leftrightarrow wins(\mathsf{x}) \vee wins(\mathsf{o}) \vee \bigwedge_{i,j=1}^{m} (p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j})$
4. $legal(a^t_{i,j}) \leftrightarrow \neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j}) \wedge turn(t) \wedge \neg terminal$
5. $\bigcirc p^t_{i,j} \leftrightarrow p^t_{i,j} \vee does(a^t_{i,j})$
6. $turn(t) \to \bigcirc\neg turn(t) \wedge \bigcirc turn(-t)$, where $-t$ represents t's opponent.

We let $\Sigma^{m,k}$ be the set of the above axioms. Rule (1) says that player x has the first turn and all grids are empty initially. Rules (2) and (3) describe the winning conditions and termination conditions, respectively. (4) specifies the preconditions of each action (legality). (5) is the combination of the frame axioms and effect axioms *a grid is marked with a player's symbol in the next state if the player takes the respective action in the current state or the grid has been marked before*. The last rule specifies turn-taking.

We will see that a strategy can be also represented by a formula in $\mathcal{L}^{GDL^+}$. For instance, the following are examples of strategies for player t in m, k-game:

- Mark anywhere available:
  $markany^t = \bigvee_{i,j=1}^{m} (\neg p^{\mathsf{x}}_{i,j} \wedge \neg p^{\mathsf{o}}_{i,j} \wedge does(a^t_{i,j}))$
- Mark the center ($c = \lfloor m/2 \rfloor$):
  $markcenter^t = \neg p^{\mathsf{x}}_{c,c} \wedge \neg p^{\mathsf{o}}_{c,c} \wedge does(a^t_{c,c})$
- Check if I can win:
  $check^t = \bigvee_{i,j=1}^{m} (\neg p^{\mathsf{x}}_{i,j} \wedge \neg p^{\mathsf{o}}_{i,j} \wedge does(a^t_{i,j}) \wedge \bigcirc wins(t))$
- Prevent immediate loss:
  $defence^t = \bigvee_{i,j=1}^{m} (\bigcirc(does(a^{-t}_{i,j}) \wedge \bigcirc wints(\text{-}t)) \to does(a^t_{i,j}))$
- Combined strategies:
  $combined^t = markcentre^t \,\triangledown\, check^t \,\triangledown\, defence^t \,\triangledown\, markany^t$

We will further explain these strategies in later sections.

The Semantics

We now provide the semantics for the language. Let $G = (W, I, T, U, L)$ be a state transition game within the multi-agent framework $(N, \mathcal{A})$. V is a valuation function for Φ. We call $M = (G, V)$ is a *state transition model*. We will assess if a formula φ is true in each state with respect to a state transition model. As a logic for reasoning about actions, the truth value of a formula relies on not only which state it is but also which action is to take in the state. For instance, to satisfy the formula $markcenter^t$, the center grid of the board should be empty and player t is to take the action at the underlying state. Therefore we will define the satisfiability relation as $M \models_{(w,a)} \varphi$ to check if formula φ is satisfied when action a is to be taken in state w of M.

Definition 4 *Let M be a state transition model within the multi-agent framework $(N, \mathcal{A})$. The* satisfiabilty *of a formula φ wrt. M and a move (w, a) is defined as follows, where $p \in \Phi$; $a, b \in \mathcal{A}$; and $\varphi, \alpha \in \mathcal{L}^{GDL^+}$:*

$M \models_{(w,a)} p$ iff $p \in V(w)$
$M \models_{(w,a)} \neg\varphi$ iff $M \not\models_{(w,a)} \varphi$
$M \models_{(w,a)} \varphi_1 \wedge \varphi_2$ iff $M \models_{(w,a)} \varphi_1$ and $M \models_{(w,a)} \varphi_2$
$M \models_{(w,a)} does(b)$ iff $a = b$
$M \models_{(w,a)} legal(b)$ iff $(w, b) \in L$
$M \models_{(w,a)} initial$ iff $w \in I$
$M \models_{(w,a)} terminal$ iff $w \in T$
$M \models_{(w,a)} [b]\varphi$ iff $M \models_{(u_b(w),c)} \varphi$ for all $c \in \mathcal{A}$
$M \models_{(w,a)} \lfloor b \rfloor\varphi$ iff $M \models_{(w,b)} \varphi$
$M \models_{(w,a)} \wr\alpha\int\varphi$ iff $M|_\alpha \models_{(w,a)} \varphi$.

where $M|_\alpha$ is the same as M except for replacing the legality relation in M by $L|_\alpha = \{(s, b) \in L : M \models_{(s,b)} \alpha\}$.

The definition is standard except for the modalities. $[b]\varphi$ requires that φ holds at the next state no matter which action to be taken at that state. $[b]\varphi$ being true at (w, a) means that φ holds at w when b is taken instead of a. To understand the semantics of $\wr . \int$, let

$$\|\alpha\|^M = \{(s, b) \in W \times \mathcal{A} : M \models_{(s,b)} \alpha\}. \quad (4)$$

i.e., the set of all the moves that make α true in M. Note that $L|_\alpha = L \cap \|\alpha\|^M$, which means that $L|_\alpha$ is in fact a strategy (see the definition of strategy in the 2^{nd} section), which contains all the moves that make α true in M. Thus φ being true under strategy α in M (i.e., $M \models_{(w,a)} \wr\alpha\int\varphi$) means φ holds in the restricted model where all legal moves are restricted to the strategy $L|_\alpha$.

As in any modal logic, $\varphi \in \mathcal{L}^{GDL^+}$ is said to be *valid* in M, written as $M \models \varphi$, if $M \models_{(w,a)} \varphi$ for all $w \in W$ and $a \in \mathcal{A}$. We write $G \models \varphi$ to mean that $M \models \varphi$ for all state transition models M of G. Similarly, by $\models \varphi$, we mean φ is valid in all state transition models within $\mathcal{F}$.

The Axioms

GDL was designed for describing game rules rather than for reasoning about games because there is no inference mechanism associated with it[2]. From this section, we will develop a

[2]Reasoning can be done in the semantical level but it is not essential to the language.

proof theory for the logic we just defined. Firstly we present the axioms and inference rules:

1. Basic axioms:

   A1 all axioms for propositional calculus.

   A2 $\vdash \neg(does(a) \wedge does(b))$ if $a \neq b$.

   A3 $\vdash \bigvee_{a \in \mathcal{A}} does(a)$

   A4 $\vdash \neg[a]initial$

   A5 $\vdash terminal \rightarrow \neg legal(a)$

2. Axioms on the action execution modality:

   B1 $\vdash [a](\varphi \wedge \psi) \leftrightarrow [a]\varphi \wedge [a]\psi$

   B2 $\vdash \neg[a]\varphi \leftrightarrow [a] \bigvee_{b \in \mathcal{A}} \lfloor b \rfloor \neg\varphi$

3. Axioms on the action selection modality:

   C1 $\vdash \lfloor b \rfloor p \leftrightarrow p$, where $p \in \Phi \cup \{initial, terminal\}$

   C2 $\vdash \lfloor b \rfloor legal(a) \leftrightarrow legal(a)$

   C3 $\vdash \lfloor b \rfloor does(a) \leftrightarrow a = b$

   C4 $\vdash \lfloor b \rfloor \neg\varphi \leftrightarrow \neg\lfloor b \rfloor\varphi$

   C5 $\vdash \lfloor b \rfloor(\varphi \wedge \psi) \leftrightarrow \lfloor b \rfloor\varphi \wedge \lfloor b \rfloor\psi$

   C6 $\vdash \lfloor a \rfloor\lfloor b \rfloor\varphi \leftrightarrow \lfloor b \rfloor\varphi$

   C7 $\vdash \lfloor a \rfloor[b]\varphi \leftrightarrow [b]\varphi$

4. Axioms on the strategy modality:

   D1 $\vdash \wr\alpha\int\ p \leftrightarrow p$, where $p \in \Phi \cup \{initial, terminal\}$

   D2 $\vdash \wr\alpha\int legal(a) \leftrightarrow (legal(a) \wedge \lfloor a \rfloor\alpha)$

   D3 $\vdash \wr\alpha\int does(a) \leftrightarrow does(a)$

   D4 $\vdash \wr\alpha\int \neg\varphi \leftrightarrow \neg\wr\alpha\int\varphi$

   D5 $\vdash \wr\alpha\int(\varphi \wedge \psi) \leftrightarrow (\wr\alpha\int\varphi \wedge \wr\alpha\int\psi)$

   D6 $\vdash \wr\alpha\int\lfloor b \rfloor\varphi \leftrightarrow \lfloor b \rfloor\wr\alpha\int\varphi$

   D7 $\vdash \wr\alpha\int[a]\varphi \leftrightarrow [a]\wr\alpha\int\varphi$

   D8 $\vdash \wr\alpha\int\wr\beta\int\varphi \leftrightarrow \wr\alpha \wedge \wr\alpha\int\beta\int\varphi$

5. Inference rules:

   (MP) If $\vdash \varphi$, $\vdash \varphi \rightarrow \psi$, then $\vdash \psi$.

   (GEN A) If $\vdash \varphi$, then $\vdash [a]\varphi$.

   (GEN C) If $\vdash \varphi$, then $\vdash \lfloor a \rfloor\varphi$.

   (GEN S) If $\vdash \varphi$, then $\vdash \wr a\int\varphi$.

where $a, b \in \mathcal{A}$ and $\varphi, \psi, \alpha \in \mathcal{L}^{GDL^+}$. We will call the logical system *Strategic Game Logic* (SGL).

The first set of axioms specify the basic properties of the logic. A2&3 say that one action and only one action can be performed in a state. A4&5 specifies the generic properties of initial states and terminal states. The axiom sets B,C and D specify the properties of each modality, respectively. Interestingly, the axioms for strategy modality $\wr.\int$ is similar to the axioms for public announcement logic (van Ditmarsch *et al.* 2007). Note that α is not a premise of $\wr\alpha\int\varphi$.

A formula $\varphi \in \mathcal{L}^{GDL^+}$ that can be derived from the above axioms and inference rules is denoted by $\vdash \varphi$. For any set of formulas Γ and a formula φ, $\Gamma \vdash \varphi$ means that there are $\varphi_1, \cdots, \varphi_m \in \Gamma$, such that $\vdash (\varphi_1 \wedge \cdots \wedge \varphi_m) \rightarrow \varphi$.

It is straightforward to verify that all the axioms and inference rules are valid under the given semantics.

Theorem 1 (Soundness) *For any $\varphi \in \mathcal{L}^{GDL^+}$, if $\vdash \varphi$, then $\models \varphi$.*

However, due to the non-standard semantics of the logic, the proof of completeness is much harder.

Completeness

The standard technique of completeness proof for a modal logic is to use the canonical model to verify if a formula is satisfiable or not. In other words, we need to establish a truth lemma, something like:

$$M^\Lambda \models_\Gamma \varphi \text{ iff } \varphi \in \Gamma$$

where M^Λ is the canonical model of the logic under consideration and Γ is a maximal consistent set. Unfortunately, this approach is not applicable here because we have to consider which action is taken at each state, i.e, to assess $M^\Lambda \models_{(\Gamma,a)} \varphi$. The question is how to assess $\varphi \in \Gamma$ accordingly to gain a truth lemma as above. In other words, how to separate action information from φ and Γ? This leads to a new technique of action forgetting we introduced in this paper. Due to space limitation, we are unable to present the full proof of completeness but will outline the major ideas in this section.

Disjunctive normal form

In order to introduce the concept of action forgetting, we first show a technique to convert a formula in $\mathcal{L}^{GDL^+}$ into a disjunctive normal form.

Consider a sublanguage of $\mathcal{L}^{GDL^+}$ by removing all *does*(.) variables. Formally, a formula is a *state-wise formula* if it can be generated by the following BNF rule:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid initial \mid terminal \mid legal(a) \mid$$

$$\lfloor a \rfloor\varphi \mid [a]\varphi \mid \wr\varphi\int\varphi$$

The resultant language is denoted by $\mathcal{L}^{GDL^*}$.

The following lemma shows why a formula in $\mathcal{L}^{GDL^*}$ is state-wise, i.e. its true value is irrelevant to which action being chosen.

Lemma 1 *For any $\chi \in \mathcal{L}^{GDL^*}$ and an action $a \in \mathcal{A}$,*

$$\vdash \chi \leftrightarrow \lfloor a \rfloor\chi$$

The following theorem shows that any formula in $\mathcal{L}^{GDL^+}$ can be transformed into a disjunctive normal form.

Theorem 2 (Disjunctive Normal Form) *Given a formula $\varphi \in \mathcal{L}^{GDL^+}$, for each $a \in \mathcal{A}$ there is $\chi_a \in \mathcal{L}^{GDL^*}$ such that*

$$\vdash \varphi \leftrightarrow \bigvee_{a \in \mathcal{A}} (\chi_a \wedge does(a))$$

Moreover, the normal form is unique in the sense that if $\vdash \varphi \leftrightarrow \bigvee_{a \in \mathcal{A}} (\chi'_a \wedge does(a))$, then $\vdash \chi_a \leftrightarrow \chi'_a$ for all $a \in \mathcal{A}$.

We let $Norm(\varphi)$ represent the set of all normal forms of φ.

Action forgetting

With normal forms, we can introduce the concept of action forgetting, which is inspired by the idea of variable forgetting in (Lin 2001). Given a formula $\varphi \in \mathcal{L}^{GDL^+}$, we let

$$forget(\varphi; does) = \{\bigvee_{a\in\mathcal{A}} \chi_a : \bigvee_{a\in\mathcal{A}}(\chi_a \wedge does(a)) \in Norm(\varphi)\} \quad (5)$$

It is easy to see that $forget(\varphi; does)$ contains the formulas resulting from forgetting all the $does(.)$ variables for each normal form of φ[3].

Furthermore, for any set, Γ, of formulas in $\mathcal{L}^{GDL^+}$, we let

$$forget(\Gamma; does) = \bigcup_{\varphi\in\Gamma} forget(\varphi; does) \quad (6)$$

Canonical model

Let $\Gamma \subseteq \mathcal{L}^{GDL^+}$. Γ is *maximal consistent* iff

1. Γ is consistent: $\Gamma \nvdash \bot$
2. Γ is maximal: there is no $\Gamma' \subseteq \mathcal{L}^{GDL^+}$ such that $\Gamma \subset \Gamma'$ and $\Gamma' \nvdash \bot$

The set of all maximal consistent sets of formulas in $\mathcal{L}^{GDL^+}$ is denoted by Ω. For any Γ and Γ' in Ω, we say Γ and Γ' are *equivalent*, denoted by $\Gamma \sim \Gamma'$, if

$$forget(\Gamma; does) = forget(\Gamma'; does) \quad (7)$$

Obviously $\sim$ defines an equivalent relation over Ω. For any Γ in Ω, we let

$$|\Gamma| = \{\Gamma' \in \Omega : \Gamma \sim \Gamma'\} \quad (8)$$

Furthermore, for each $\Gamma \in \Omega$ and $a \in \mathcal{A}$, we write

$$|\Gamma|^a = \{\Gamma' \in |\Gamma| : does(a) \in \Gamma'\} \quad (9)$$

It is easy to see that $|\Gamma| = \bigcup_{a\in\mathcal{A}} |\Gamma|^a$. For simplicity, we write $\varphi \in |\Gamma|$ to mean $\varphi \in \Gamma'$ for any $\Gamma' \in |\Gamma|$. Similar abuse also applies to $|\Gamma|^a$.

The following lemma shows that the forgetting operator gives us what we need, which separates action information from state information.

Lemma 2 *For any* $\Gamma \in \Omega$, $forget(\Gamma; does) = \Gamma \cap \mathcal{L}^{GDL^*}$.

Now we are ready to define the canonical model for a given multi-agent framework.

Definition 5 For the given multi-agent framework $(N, \mathcal{A})$, we define the *canonical model* $M^\Lambda = ((N, \mathcal{A}, W^\Lambda, I^\Lambda, T^\Lambda, U^c, L^\Lambda), V^\Lambda)$ as follows:

1. $W^\Lambda = \{|\Gamma| : \Gamma$ is maximal consistent subset of $\mathcal{L}^{GDL^+}\}$.
2. $I^\Lambda = \{|\Gamma| : initial \in \Gamma\}$
3. $T^\Lambda = \{|\Gamma| : terminal \in \Gamma\}$.
4. $U^\Lambda : W^\Lambda \times \mathcal{A} \mapsto W^\Lambda \backslash I^\Lambda$ such that for each $a \in \mathcal{A}$ and $w \in W^\Lambda$, $\{\varphi : [a]\varphi \in w\} \subseteq U^\Lambda(w, a)$.
5. $L^\Lambda = \{(w, a) \in W^\Lambda \times \mathcal{A} : legal(a) \in w\}$.
6. $V^\Lambda : \Phi \mapsto 2^{W^\Lambda}$ such that $V^\Lambda(p) = \{w \in W^\Lambda : p \in w\}$.

The following two lemmas guarantee the canonical model is a state transition model.

Lemma 3 *For any* $a \in \mathcal{A}$ *and any* $w \in W^\Lambda$, *there is a unique* $w' \in W^\Lambda$ *such that* $\{\varphi : [a]\varphi \in w\} \subseteq w'$.

Lemma 4

1. *For all* $(w, a) \in W^\Lambda \times \mathcal{A}$, $U^\Lambda(w, a) \notin I^\Lambda$.
2. *For all* $(w, a) \in L^\Lambda$, $w \notin T^\Lambda$.

Truth Lemma

Now we are ready to show the completeness, which is immediately followed by the following Truth Lemma.

Lemma 5 *Let* $\varphi \in \mathcal{L}^{GDL^+}$. *For any* $(|\Gamma|, a) \in W^\Lambda \times \mathcal{A}$,

$$M^\Lambda \models_{(|\Gamma|,a)} \varphi \text{ iff } \varphi \in |\Gamma|^a$$

Proof: We prove it by induction on φ.

Assume that $\varphi = p$ where $p \in \Phi$. By the construction of M^Λ, $|\Gamma| \in V^\Lambda(p)$ iff $p \in |\Gamma|$ iff $p \in |\Gamma|^a$.

Assume that $\varphi = initial$. Then $M^\Lambda \models_{(|\Gamma|,a)} initial$ iff $|\Gamma| \in I^\Lambda$ iff $initial \in \Gamma$ iff $initial \in |\Gamma|^a$. $\varphi = terminal$ is similar.

Assume that $\varphi = does(b)$ where $b \in \mathcal{A}$. By the construction of M^Λ, $M^\Lambda \models_{(|\Gamma|,a)} does(b)$ iff $a = b$. Note that $does(b) \in |\Gamma|^a$ iff $a = b$, because $\vdash \neg(does(a) \wedge does(b))$ if $a \neq b$. Therefore $M^\Lambda \models_{(|\Gamma|,a)} does(b)$ iff $does(b) \in |\Gamma|^a$.

Assume that $\varphi = legal(b)$ where $b \in \mathcal{A}$. Then $M^\Lambda \models_{(|\Gamma|,a)} legal(b)$ iff $(|\Gamma|, a) \in L^\Lambda$ iff $legal(b) \in |\Gamma|$ iff $legal(b) \in |\Gamma|^a$, as desired.

Assume that $\varphi = \neg\psi$. If $M^\Lambda \models_{(w,a)} \neg\psi$, we have $M^\Lambda \not\models_{(w,a)} \psi$. By inductive assumption, $\psi \notin |\Gamma|^a$, which means that there is a maximal consistent set $\Gamma' \in |\Gamma|^a$ such that $\psi \notin \Gamma'$. Since Γ' is maximal consistent, we have $\neg\psi \in \Gamma'$. Assume that $\psi = \bigvee_{b\in\mathcal{A}}(does(b) \wedge \chi_b)$. Thus $\vdash \neg\psi \leftrightarrow \neg\bigvee_{b\in\mathcal{A}}(does(b) \wedge \chi_b) \leftrightarrow \bigwedge_{b\in\mathcal{A}}(does(b) \rightarrow \neg\chi_b)$. By $\neg\psi \in \Gamma'$, we have $does(b) \rightarrow \neg\chi_b \in \Gamma'$ for all $b \in \mathcal{A}$. Specifically, we have $does(a) \rightarrow \neg\chi_a \in \Gamma'$. Note that $does(a) \in \Gamma'$, we yield $\neg\chi_a \in \Gamma'$. Since $\neg\chi_a \in \mathcal{L}^{GDL^*}$, we have $\neg\chi_a \in |\Gamma|^a$. On the other hand, $\neg does(b) \in |\Gamma|^a$ for any $b \neq a$. Combining the two cases together, we then have $\neg does(c) \vee \neg\chi_c \in |\Gamma|^a$ for any $c \in \mathcal{A}$, which implies $\neg\psi \in |\Gamma|^a$, as desired.

Assume that $\varphi = \lfloor b \rfloor \psi$ where $b \in \mathcal{A}$. By the construction of M^Λ and the assumption of induction, $M^\Lambda \models_{(|\Gamma|,a)} \lfloor b \rfloor \psi$ iff $M^\Lambda \models_{(|\Gamma|,b)} \psi$ iff $\psi \in |\Gamma|^b$. Let $\psi = \bigvee_{c\in\mathcal{A}}(\chi_c \wedge does(c))$. Then $\bigvee_{c\in\mathcal{A}}(\chi_c \wedge does(c)) \in |\Gamma|^b$. Note that $does(b) \in |\Gamma|^b$. Thus $\bigvee_{c\in\mathcal{A}}(\chi_c \wedge does(c) \wedge does(b)) \in |\Gamma|^b$, which implies $\chi_b \in |\Gamma|^b$. Since $\chi_b \in \mathcal{L}^{GDL^*}$, we have $\chi_b \in |\Gamma|$. Note that $\vdash \lfloor b \rfloor \psi \leftrightarrow \lfloor b \rfloor \bigvee_{c\in\mathcal{A}}(\chi_c \wedge does(c)) \leftrightarrow \chi_b$. Therefore $\lfloor b \rfloor \psi \in |\Gamma|$, which implies $\lfloor b \rfloor \psi \in |\Gamma|^a$.

Assume that $\varphi = [b]\psi$. If $M^\Lambda \models_{(|\Gamma|,a)} [b]\psi$, we have $M^\Lambda \models_{(U_b^\Lambda(|\Gamma|),c)} \psi$ for all $c \in \mathcal{A}$. By the inductive assumption, $\psi \in |\Gamma'|^c$ for any $c \in \mathcal{A}$, where $|\Gamma'| = U_b^\Lambda(|\Gamma|)$. Let $\psi = \bigvee_{d\in\mathcal{A}}(\chi_d \wedge does(d))$. For each $c \in \mathcal{A}$, we have $\bigvee_{d\in\mathcal{A}}(\chi_d \wedge$

[3] Note that there are significant differences between our concept and Lin's, despite of some similarities. We apply the forgetting process to the normal forms of a formula rather than to its original form. For instance, let $\varphi = \lfloor b \rfloor does(a)$ where $a \neq b$. $forget(\varphi; does) = \{\psi \in \mathcal{L}^{GDL^*} : \vdash \bot \leftrightarrow \psi\}$ because $\vdash \varphi \leftrightarrow \bot$. However, $\varphi(does(a)/\top) \vee \varphi(does(a)/\bot) = \top$.

$does(d) \wedge does(c)) \in |\Gamma'|^c$, that is, $\chi_c \in |\Gamma'|^c$. Therefore $\chi_c \in |\Gamma'|$ for any $c \in \mathcal{A}$, that is $\bigwedge_{c\in\mathcal{A}} \chi_c \in |\Gamma'|$. On the other hand, if $[b]\psi \notin |\Gamma|$, there is $\Gamma'' \in |\Gamma|$ such that $\neg[b]\psi \in \Gamma''$. It turns out that $[b] \bigvee_{c\in\mathcal{A}} \lfloor c \rfloor \neg\psi \in \Gamma''$, or $[b] \bigvee_{c\in\mathcal{A}} \lfloor c \rfloor \bigwedge_{d\in\mathcal{A}} (does(d) \to \neg\chi_d) \in \Gamma''$. We then have $[b] \bigvee_{c\in\mathcal{A}} \neg\chi_c \in \Gamma'$. Since $[b] \bigvee_{c\in\mathcal{A}} \neg\chi_c \in \mathcal{L}^{GDL*}$, we have $[b] \bigvee_{c\in\mathcal{A}} \neg\chi_c \in |\Gamma|$. It follows that $\bigvee_{c\in\mathcal{A}} \neg\chi_c \in |\Gamma'|$, which contradicts the fact $\bigwedge_{c\in\mathcal{A}} \chi_c \in |\Gamma'|$. This means that assumption $[b]\psi \notin |\Gamma|$ is incorrect. Therefore $[b]\psi \in |\Gamma|$, or $[b]\psi \in |\Gamma|^a$, as desired.

If $\varphi = \wr\alpha\int\psi$, we apply Axioms (D1-D8) recursively to eliminate the occurrences of $\wr.\int$ operator. □

Theorem 3 **(Completeness)** *For any* $\varphi \in \mathcal{L}^{GDL^+}$, *if* $\models \varphi$, *then* $\vdash \varphi$.

Reasoning About Strategies

We have established a proof theory for the logic we introduced. In this section, we demonstrate how to use this logic for reasoning about game strategies. In fact, we are exploring a proof-theoretical approach that can be used for general game players to create, combine and verify their strategies.

Inference with game-specific axioms

The axioms and inference rules of SGL are generic for any game. To verify a game specific property, we must utilise the game specific axioms as we have shown for the *mk*-games.

Let Σ be a set of game-specific axioms for a game G. We write $\vdash^{\Sigma} \varphi$ to mean that φ can be derived from the generic axioms of SGL and the game-specific axioms in Σ by the inference rules of SGL.

For instance, the following can be derived in any *mk*-game:

$\vdash^{\Sigma^{m,k}} turn(\mathsf{x}) \to \bigcirc turn(\mathsf{o}) \wedge \bigcirc\bigcirc turn(\mathsf{x})$

We say Σ is an *axiomatisation* of a game G if for any state transition model M of G and any $\varphi \in \mathcal{L}^{GDL^+}$, $\vdash^{\Sigma} \varphi$ iff $M \models \varphi$. Note that not all games can be axiomatised.

Strategy rules of a player

As defined in the 2^{nd} section, a strategy in a game is a set of legal moves. In this sense, any formula can be viewed as a strategy because it uniquely determines a set of legal moves. Formally, given a game $G = (N, \mathcal{A}, W, I, T, U, L)$, for any $\alpha \in \mathcal{L}^{GDL^+}$, let

$$S^G(\alpha) = \{(w, a) \in L : M \models_{(w,a)} \alpha \text{ for any } M \text{ of } G\}$$

We say α is a strategy rule of the strategy $S^G(\alpha)$. However, when we talk about a strategy, we normally mean a strategy of a player. The following defines strategies for specific players.

Definition 6 *A formula* $\alpha \in \mathcal{L}^{GDL^+}$ *is called a* strategy rule, or simply strategy, of player *i if each normal form of* α *can be expressed in the form* $\bigvee_{a\in A^i} (\chi_a \wedge does(a))$.

From the uniqueness of normal forms, it is easy to show that player i's strategies specify only player i's moves.

Lemma 6 *If* α *is a strategy of player i in game G,* $S^G(\alpha) \subseteq W \times A_i$.

One may have noticed that our concept of strategies is significantly different from the one in the context of ATL (Alur *et al.* 2002; van der Hoek *et al.* 2005; Walther *et al.* 2007). In ATL, a strategy is a function that maps each state (or a sequence of states) to an action. In other words, a strategy specifies which action has to do exactly in each state.

The following theorem provides us with a syntactical approach to verify if a strategy rule represents a functional strategy in a game.

Proposition 1 *Let* Σ *be an axiomatisation of a game G. For any strategy rule* α *of player i,*

1. *α of player i is complete in G if*

$$\vdash^{\Sigma} \bigvee_{a\in A^i} legal(a) \to \wr\alpha\int \bigvee_{a\in A^i} legal(a)$$

2. *α of player i is deterministic in G if*

$$\vdash^{\Sigma} \wr\alpha\int \bigwedge_{a,b\in A^i \,\&\, a\neq b} \neg(legal(a) \wedge legal(b))$$

It is not hard to prove that the strategy $markany^t$ defined in the 3^{rd} section is complete for player t and $markcenter^t$ is deterministic for t with respect to any *mk*-game.

Strategy composition

One way of creating strategies is that we start with some "rough ideas" of strategies and combine them in certain way to form more complicated, if needed, functional strategies. This can be done by using the prioritised strategy connectives defined by Equations (2) and (3) in the 3^{rd} section. The following observation gives the semantics of these connectives:

Proposition 2 *For any state transition model M,*

- $M \models_{(w,a)} \alpha \triangledown \beta$ *iff either* $M \models_{(w,a)} \alpha$, *or* $M \models_{(w,a)} \beta$ *but* $M \models_{(w,c)} \neg\alpha$ *for all* $c \in \mathcal{A}$.
- $M \models_{(w,a)} \alpha \triangle \beta$ *iff* $M \models_{(w,a)} \alpha$, *and if* $M \models_{(w,c)} \alpha \wedge \beta$ *for some* $c \in \mathcal{A}$, *then* $M \models_{(w,a)} \beta$.

One may wonder if α and β are strategy rules for a player, whether their prioritised disjunction and conjunction are still strategy rules for the same player. The following proposition answers this question.

Proposition 3 *If* α *and* β *are strategy rules for player i, so are* $\alpha \triangledown \beta$ *and* $\alpha \triangle \beta$.

The following properties will help us to create functional strategies:

Proposition 4 *Let* α *and* β *are strategy rules for player i.*

1. *If* α *or* β *is complete, so is* $\alpha \triangledown \beta$.
2. *If* α *is complete, so is* $\alpha \triangle \beta$.
3. *If* α *and* β *are deterministic, so is* $\alpha \triangledown \beta$.
4. *If* α *is deterministic, so is* $\alpha \triangle \beta$.

With this result, we can easily know that $combined^t$ is complete because *markany* is complete.

Strategy implementation

One target of the current research is to develop a syntactical approach for a game player to generate strategies for game playing. We have shown how to use logical formulas to write simple strategies and how to combine them into more complicated strategies. We have also demonstrate how to verify if a strategy satisfies some desired properties such as completeness and determinicity.

One may even want to prove if a strategy can guarantee to win or not to lose. The following introduces a syntactical way to verify if a strategy can bring a game from an initial state to a final state in order to achieve a property.

Let Σ be an axiomatisation of a game. For any $\alpha, \varphi \in \mathcal{L}^{GDL^+}$, α *implements* φ in Σ, denoted by, $\alpha \Vdash^{\Sigma} \varphi$, if

- $\vdash^{\Sigma} initial \rightarrow \wr\alpha\wr \bigvee_{a\in\mathcal{A}} legal(a)$
- $\vdash^{\Sigma} \neg terminal \rightarrow$
  $\bigwedge_{a\in\mathcal{A}} \wr\alpha\wr (legal(a) \rightarrow [a](terminal \vee \bigvee_{c\in\mathcal{A}} legal(c)))$
- $\vdash^{\Sigma} terminal \wedge \alpha \rightarrow \varphi$.

Intuitively, α implements φ if there is a complete path that leads the game from an initial state to a terminal in which φ is true. Note that if you want to verify one of your strategies, you must have a guess of another player's strategy and combine them into a formula α. This will introduce significant complexity, especially if you target for a winning strategy.

Conclusion and Related Work

We have established a modal logic with a non-standard Kripke semantics, and we have presented a sound and complete axiomatic system for representing, combining, and reasoning about game strategies. It is the first logical system for GDL that is sound and complete, and one of a few existing formal systems for strategic reasoning.

In terms of future work, our logic provides the foundations for gaining new insights into the mechanism of strategic reasoning and likely leads to a new approach for automatically generating game strategies for game-playing systems, including general game players. Automatically generating good strategies for games in general game playing is a largely unsolved problem (Genesereth and Björnsson 2013). But we believe that our logic can help general game players to systematically generate, and automatically evaluate, strategies from the logical elements of a given game description, especially if the aim is to find compact, practically useful strategies rather than optimal ones. It is a widely held belief that automatically generating smart strategies for complicated games is hard. But our own, recent experiments indicate that it might be practically feasible nonetheless. Specifically, we found that standard Minimax with alpha-beta heuristics can easily generate the following strategy rules for m,k-games (ordered by decreasing priority): (1) Fill the center; (2) Fill a cell if this wins the game; (3) Fill a cell to prevent an immediate threat by the opponent; (4) Make a threat; (5) Fill any cell, but give priority to central positions. Combining these strategies with the prioritized connectives provides provably winning/loss-preventing strategies for simple m,k-games and reasonably good strategies for larger ones. In general, if we do not aim to find prefect winning strategies for any game, then automatically discovering strategies with certain desirable properties seems always possible.

References

R. Alur, T. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002.

K. Chatterjee, T. Henzinger, and N. Piterman. Strategy logic. *Information and Computation*, 208(6):677 – 693, 2010.

J. Clune. Heuristic evaluation function for general game playing. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pages 1134–1139. 2007.

P. Kissmann and S. Edelkamp. Symbolic classification of general multi-player games. In *Proceedings of the 18th European Conference on Artificial Intelligence*, pages 905–906. 2008.

H. Finnsson and Y. Björnsson. Simulation-based approach to general game playing. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 259–264. 2008.

M. Genesereth and M. Thielscher. *General Game Playing*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2014.

M. Genesereth, N. Love, and B. Pell. General game playing: Overview of the AAAI competition. *AI Magazine*, 26(2):62–72, 2005.

M. Genesereth and Y. Björnsson. The international general game playing competition. *AI Magazine*, 34(2):107–111, 2013.

F. Lin. On strongest necessary and weakest sufficient conditions. *Artificial Intelligence*, 128(1-2):143 – 159, 2001.

S. Schiffel and M. Thielscher. Fluxplayer: a successful general game player. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pages 1191–1196. 2007.

J. van Benthem. In praise of strategies. In Jan Van Eijck and Rineke Verbrugge, editors, *Games, Actions, and Social Software*, ILLC scientific publications. University of Amsterdam, 2008.

H. van den Herik, J. Uiterwijk, and J. van Rijswijck. Games solved: Now and in the future. *Artificial Intelligence*, 134(1-2):277 – 311, 2002.

W. van der Hoek, W. Jamroga, and M. Wooldridge. A logic for strategic reasoning. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 157–164. 2005.

H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic epistemic logic*. Springer, 2007.

D. Walther, W. van der Hoek, and M. Wooldridge. Alternating-time temporal logic with explicit strategies. In *Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 269–278. 2007.