Strategic and Epistemic Reasoning for the Game Description Language GDL-II

Ji Ruan and Michael Thielscher

Abstract. The game description language GDL has been developed as a logic-based formalism for representing the rules of arbitrary games in general game playing. A recent language extension called GDL-II allows to describe nondeterministic games with any number of players who may have incomplete, asymmetric information. In this paper, we show how the well-known Alternating-time Temporal Epistemic Logic (ATEL) can be adapted for strategic and epistemic reasoning about general games described in GDL-II. We provide a semantic characterisation of GDL-II descriptions in terms of ATEL models. We also provide a syntactic translation of GDL-II descriptions into ATEL formulas, and we prove that these two characterisations are equivalent. We show that model checking in this setting is decidable by giving an algorithm, and we demonstrate how our results can be used to verify strategic and epistemic properties of games described in GDL-II.

1 Introduction

The general game description language GDL, which has been established as input language for general game-playing systems [5, 8], has recently been extended to GDL-II to incorporate games with nondeterministic actions and where players have incomplete/imperfect information [15]. We have previously analysed the epistemic logic behind GDL-II and in particular shown that the situation at any stage of a game can be characterised by a multi-agent epistemic (i.e., S5-) model [10]. However, this result merely provides a static characterisation of what players know (and don’t know) at a certain stage. As such it cannot be used to reason about how players’ knowledge evolves as the game progresses, nor does it allow to reason about the strategic ability of players to reach a desired state (possibly in cooperation with other players), etc. All these aspects presuppose the use of an underlying logic that goes beyond standard epistemic logic and that combines both strategic and epistemic reasoning. Alternating-time Temporal Epistemic Logic (ATEL) [16], an extension to Alternating-time Temporal Logic (ATL) [1], with incomplete information, is such a formalism. For strategic reasoning alone, it has been shown that ATL can be applied to reason about complete-information games described in the original GDL using model checking methods [12]. Also, model checking for GDL is known to be EXPTIME-complete [12]. But unfortunately, the addition of incomplete information (and perfect recall) in GDL-II renders the model checking problem of ATL/ATEL undecidable [3].

In this paper we provide an adaption of ATEL for strategic and epistemic reasoning about general games described in GDL-II. Our main results are a characterisation of GDL-II descriptions in ATEL and a decidability result for the model checking problem. Specifically, we provide a semantic characterisation of GDL-II descriptions in terms of ATEL models and a syntactic translation of GDL-II descriptions into ATEL formulas, and we prove that these two characterisations are equivalent. We show that model checking in this setting is decidable and demonstrate how our results allow to verify strategic and epistemic properties of games described in GDL-II.

The paper is organised as follows. Section 2 gives preliminaries on GDL-II and ATEL. Section 3 presents a semantic characterisation of GDL-II using ATEL models and a syntactic mapping from GDL-II to ATEL formulas, along with the main equivalence result. Section 4 shows the decidability of model checking problem by giving an algorithm and discusses the strategic and epistemic properties that can be used for reasoning about GDL-II games. We conclude with a discussion on related work. Full proofs of our results can be found in an accompanying technical report [11].

2 Preliminaries

2.1 Game Description Language GDL-II

A complete game description consists of the names of (one or more) players, a specification of the initial position, the legal moves and how they affect the position, and the terminating and winning criteria. The emphasis of game description languages is on high-level, declarative game rules that are easy to understand and maintain. At the same time, GDL and its successor GDL-II have a precise semantics and are fully machine-processable. Moreover, background knowledge is not required—a set of rules is all a player needs to know to be able to play a hitherto unknown game. The description language GDL-II uses these keywords:

<table>
<thead>
<tr>
<th>Role</th>
<th>Syntax</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>role(τr)</td>
<td>τr is a player</td>
<td></td>
</tr>
<tr>
<td>init(τf)</td>
<td>τf holds in the initial position</td>
<td></td>
</tr>
<tr>
<td>true(τf)</td>
<td>τf holds in the current position</td>
<td></td>
</tr>
<tr>
<td>legal(τr, ?m)</td>
<td>τr can do move ?m</td>
<td></td>
</tr>
<tr>
<td>does(τr, ?m)</td>
<td>player τr does move ?m</td>
<td></td>
</tr>
<tr>
<td>next(τf)</td>
<td>τf holds in the next position</td>
<td></td>
</tr>
<tr>
<td>terminal</td>
<td>the current position is terminal</td>
<td></td>
</tr>
<tr>
<td>goal(τr, ?v)</td>
<td>goal value for role τr is ?v</td>
<td></td>
</tr>
<tr>
<td>sees(τr, ?p)</td>
<td>τr perceives ?p in the next position</td>
<td></td>
</tr>
<tr>
<td>random</td>
<td>the random player</td>
<td></td>
</tr>
</tbody>
</table>

GDL (without sees and random) is suitable for describing finite, synchronous, and deterministic n-player games with complete information about the game state [8]. The extended game description language GDL-II allows to specify games with randomness and imperfect/incomplete information [15]. Valid game descriptions must satisfy certain syntactic restrictions, which ensure that all deductions

1 School of Computer Science and Engineering, The University of New South Wales, Australia, email: {jiruan, mit}@cse.unsw.edu.au
“...” used in the following definition are finite and decidable; for details we have to refer to [8] for space reasons.

A unique game model can be obtained from a valid GDL-II game description by using the notion of the stable models of logic programs with negation [4]. The syntactic restrictions in GDL-II ensure that all logic programs we consider have a unique and finite stable model [8, 15]. Hence, the game model for GDL-II has a finite set of players, finite states, and finitely many legal moves in each state [15]. By $G \vdash p$ we denote that ground atom $p$ is contained in the unique stable model, denoted as $\text{SM}(G)$, for a stratified set of clauses $G$.

In the following definition of a game model for GDL-II, states are identified with the set of atoms that are true in them.

**Definition 1 (GDL-II Game Semantics).** Let $G$ be a valid GDL-II specification. The semantics of $G$ is the state transition system $(R, s_0, t, l, u, T, g)$ given by

- **roles** $R = \{r \mid \text{role}(r) \in \text{SM}(G)\}$;
- **initial position** $s_0 = \text{SM}(G \cup \{\text{true}(f) \mid \text{init}(f) \in \text{SM}(G)\})$;
- **terminal positions** $t = \{s \mid \text{terminal} \in s\}$;
- **legal moves** $l = \{(r, m, s) \mid \text{legal}(r, m) \in s\}$;
- **state update function** $u(M, s) = \text{SM}(G \cup \{\text{true}(f) \mid \text{next}(f) \in \text{SM}(G \cup s \cup M^{\text{does}})\})$, for all joint legal moves $M$ (i.e., where each role in $R$ takes one legal move) and states $s$;
- **information relation** $I = \{(r, M, s, p) \mid \text{sees}(r, p) \in \text{SM}(G \cup s \cup M^{\text{does}})\}$;
- **goal relation** $g = \{(r, n, s) \mid \text{goal}(r, n, s) \in s\}$.

Note that a state $s$ contains all ground atoms that are true in the state, which includes the “fluent atoms” $\text{true}(f)$ in, respectively, $\{\text{true}(f) \mid \text{init}(f) \in \text{SM}(G)\}$ (for the initial state) and $\{\text{true}(f) \mid \text{next}(f) \in \text{SM}(G \cup s \cup M^{\text{does}})\}$ (for the successor state of $s$ and $M$), and all other atoms that can be derived from $G$ and these fluent atoms.

Different runs of a game can be described by developments, which are sequences of states and moves by each player up to a certain round. A player cannot distinguish two developments if he makes the same moves and perceptions in the two.

**Definition 2.** [15] Let $(R, s_0, t, l, u, T, g)$ be the semantics of a GDL-II description $G$, then a **development** $\delta$ is a finite sequence $(s_0, M_1, s_1, \ldots, s_d, M_d, s_d)$ such that for all $i \in \{1, \ldots, d\} (d \geq 0)$, $M_i$ is a joint move and $s_i = u(M_{i-1}, s_{i-1})$.

The length of a development $\delta$, denoted as $\text{len}(\delta)$, is the number of states in $\delta$. By $M(j)$ we denote agent $j$’s move in the joint move $M$.

A player $j \in R \setminus \{\text{random}\}$ cannot distinguish two developments $\delta = (s_0, M_1, s_1, \ldots, s_d)$ and $\delta' = (s_0', M_1', s_1', \ldots)$ (written as $\delta \sim \delta'$ iff $\text{len}(\delta') = \text{len}(\delta)$ and for all $i \in [\text{len}(\delta) - 1]$:

- $\{p \mid (j, M_i, s_i, p) \in T\} = \{p \mid (j, M_i', s_i', p) \in T\}$,
- $M_i(j) = M_i'(j)$, and $\delta_i \sim \delta'_i$.

As an example, Figure 1 provides a GDL-II description of a card trading game adapted from [6]. In this game, player Bob (abbreviated as $b$) plays against the random player (line 1). The deck consists of Ace, King and Queen $(A, K, Q)$; it is assumed that $A$ beats $K$.

For a joint move $M$ where players $r_1, \ldots, r_k$ take moves $m_1, \ldots, m_k$, we define $M^{\text{does}} \triangleq \{\text{does}(r_1, m_1), \ldots, \text{does}(r_k, m_k)\}$.

For a joint move $M$ where players $r_1, \ldots, r_k$ take moves $m_1, \ldots, m_k$, we define $M^{\text{does}} \triangleq \{\text{does}(r_1, m_1), \ldots, \text{does}(r_k, m_k)\}$.

$K$ beats $Q$, but $Q$ beats $A$ (lines 2–4). The game starts at round 0 (line 5). Lines 7–11 describe the possible moves for the players in different rounds: First, $\text{random}$ deals one card to $b$ and one card to itself. Then $b$ decides whether to trade his card for the one in the deck or to keep the current one. The player with the better card wins the game. Lines 13–20 define what will be true in the next state. For example, if $\text{random}$ does $\text{deal}(\text{round}(1))$, then $b$ will hold $A$ in the next state (line 15). Lines 22–24 specify what $b$ can see in the next state: if $\text{random}$ does $\text{deal}(\text{round}(1))$, then $b$ will see that it holds card $A$ in the next state (line 22), and if $\text{random}$ holds $A$ in round 1, then $b$ sees this in the next state (i.e., in round 2) (lines 23–24). Line 26 specifies when the game is terminal, and lines 27–31 give the goal values of player $b$ (where $\text{goal}(b, 100)$ implies a win for $b$). The corresponding game model is sketched in Figure 2. Due to limited space, the $\text{noop}$ actions are not shown in the joint actions; e.g., for $\langle \text{noop}, \text{deal}(A, K) \rangle$ we only show $\text{deal}(A, K)$. States that player $b$ cannot distinguish are connected by a dashed line.

**2.2 ATEL with Finite Computations**

**Definition 3 (Language of ATEL).** [16]. The language of ATEL (with respect to a set of agents $A_g$ and a set of atomic propositions $\Phi$), is given by the following grammar:

$$
\omega ::= p \mid \neg p \mid p \lor q \mid p \land q \mid \langle X \rangle \mid \langle X \rangle \lor \varphi \mid \langle X \rangle \land \varphi \mid \langle X \rangle \varphi U \varphi \mid K_p \varphi \mid C_X \varphi
$$

where $p \in \Phi$ is an atomic proposition and $X \subseteq A_g$ is a set of agents. Other logic constants and connectives $\top, \bot, \varnothing, \rightarrow$ are defined as usual.
Intuitively, $\langle X \rangle \bigcirc \varphi$ means a coalition of agents $X$ can ensure that $\varphi$ will hold in the next state; $\langle X \rangle \bigcirc \phi$ means a coalition of agents $X$ can ensure that $\varphi$ will always hold; $\langle X \rangle \bigcirc \psi$ means a coalition of agents $X$ can ensure that $\varphi$ will hold until $\psi$ holds; $K_i \varphi$ means agent $i$ knows $\varphi$; and $C_X \varphi$ means $\varphi$ is common knowledge among the agents in $X$. For example, “agent $i$ knows that agent $j$ knows $p$” can be expressed as $K_i K_j p$.

We will interpret ATEL formulas on finite tree-like structures (game trees) derived from GDL-II games (such as shown in Figure 2). A game is well-formed if it terminates after finite steps and if all players have at least one legal move in non-terminal states [8]. In addition, the setup of general game playing competitions is such that all agents are aware of the time progressing (Synchronicity) and remember what they have seen and done in the past (Perfect Recall) [15]. For this reason, we adapt the ATEL models from [16, 6] by adding an explicit set of terminal states. These terminal states correspond to those games states in which terminal holds. 

**Definition 4 (Model of ATEL).** An Action-based Alternating Epistemic Transition System (AAETS) is a tuple

$$ Q, s_0, T, Ag, \{Ac_i \mid i \in Ag\}, \rho, \tau, \{\sim_i \mid i \in Ag\}, \Phi, V $$

where

- $Q$ is a finite, non-empty set of possible states;
- $s_0 \in Q$ is the initial state;
- $T \subseteq Q$ is a non-empty set of terminal states;
- $Ag$ is a finite, non-empty set of agents;
- $Ac_i$ is a finite, non-empty set of actions for each $i \in Ag$;
- $\rho : Ac_i \times \omega \rightarrow 2^Q$ is an action precondition function, which for each action $a \in Ac_i(s \cup \cup_a \text{Ac}_j)$ defines the set of states $\rho(a)$ from which a move can be executed (for any action $a$ it is assumed that $T \cap \rho(a) = \emptyset$, which means that no actions are possible in terminal states);
- $\tau : Ac_i \times \cdots \times Ac_n \times Q \rightarrow Q$ is a partial system transition function, which defines the system state $\tau(M, s)$ that would result from agents’ actions $M = (a_1, \ldots, a_n)$ on state $s$, given that $s \in \rho(M(i))$ for all agent $i \in Ag$ (i.e., agent $i$’s action $M(i) = a_i$ can be executed at state $s$);
- $\Phi$ is a finite, non-empty set of atomic propositions;
- each $\sim_i \subseteq Q \times Q$ is an equivalence relation (called the accessibility relation) for agent $i$;
- $V : Q \rightarrow 2^\mathcal{P}$ is a valuation function that assigns each state a set of atomic propositions (said to be true in that state).

A computation of an AAETS is a finite sequence of states $s = s_0, s_1, \ldots, s_k \in Q^+$ such that for each $0 < u < k$, there is a joint action $M'$ such that $s_u = \tau(M', s_{u-1})$. A computation starting in state $s$ is referred to as an $s$-computation. If $0 \leq u < |s|$ (the size of $s$), then we denote by $\lambda(u)$ the $u$-th state in $s$ by $\lambda(0, u)$ the finite prefix $s_0 \ldots s_u$ of $\lambda$, and by $\lambda(s)$ the last state of $\lambda$.

In addition to finiteness, we stipulate the following three properties for our AAETS in accordance with the general game playing setting as discussed above.

**Definition 5 (Tree, Synchronicity and Perfect Recall).** An AAETS $A$ has tree property iff any state $s$ is reached from initial state $s_0$ via a unique computation. We denote such a computation by $\lambda(s_0, s)$.

An AAETS $A$ has synchronicity iff for all $s, t \in A$ and agents $i, s \sim_i t$ implies that the computations from the initial state $s_0$ to $s$ (i.e., $\lambda(s_0, s)$) and from $s_0$ to $t$ (i.e., $\lambda(s_0, t)$) have the same length, i.e., $|\lambda(s_0, s)| = |\lambda(s_0, t)|$.

An AAETS $A$ has perfect recall iff for all finite computations $\lambda, X \subseteq Q^+$ and agents $i, \lambda \sim_i X$ implies that last$(\lambda) \sim_i$ last$(X)$ and $|\lambda[0, |\lambda| - 2]| \sim_i |\lambda[0, |\lambda| - 2]|$.

Given an agent $i \in Ag$ and a state $s \in Q$, we denote the options available to $i$ in $s$—the actions that $i$ may perform in $s$—by $\text{options}(i, s) = \{m \mid m \in Ac_i, \rho(m)\}$.

We define a perfect recall strategy for an agent $i \in Ag$ to be a function $\sigma_i : Q^+ \rightarrow Ac_i$ that must satisfy

- the legality constraint that $\sigma_i(\lambda) \in \text{options}(i, \text{last}(\lambda))$ for all finite computations $\lambda \in Q^+$, and
- the uniformity constraint that for any two finite computations $\lambda_1, \lambda_2 \in Q^+$, if $\lambda_1 \sim \lambda_2$ then $\sigma_i(\lambda_1) = \sigma_i(\lambda_2)$.

A perfect recall strategy $\sigma_X$ for some coalition $X$, and a state $s \in Q$, we define $\text{comp}(\sigma_X, s)$ to be the set of all finite computations that may occur if every agent $i \in X$ follows the corresponding strategy $\sigma_i$, starting when the system is in state $s \in Q$ and ending with a terminal state $T : \text{comp}(\sigma_X, s) = \{\lambda \mid \lambda(0) = s, \text{last}(\lambda) \in T \text{and } \forall 0 \leq u < |\lambda| - 1 : \lambda(u + 1) \in \text{out}(\sigma_X, \lambda(0, u))\}$.

Note that herein lies the major difference between our models and the models defined in other ATEL/ATEL papers [1, 16, 6]. In our case, all the computations in $\text{comp}(\sigma_X, s)$ are finite in accordance with well-formed GDL-II games, whereas they are infinite in other papers. This results in the following modified semantics.

**Definition 6 (Semantics of ATEL).** For a finite AAETS $A$ and a state $s$, let $\lambda(s_0, s)$ denote the finite computation starting from $s_0$ and ending with $s$. The truth of ATEL formulas is inductively defined as follows:

- $A, s \models p$ iff $p \in V(s)$ (where $p \in \Phi$);
- $A, s \models \neg \varphi$ iff $A, s \not\models \varphi$;
- $A, s \models \varphi \lor \psi$ iff $A, s \models \varphi$ or $A, s \models \psi$;
- $A, s \models \langle X \rangle \bigcirc \varphi$ iff $\exists X, s \models \varphi$ and $\forall X, s \models \psi$;
- $A, s \models \langle X \rangle \bigcirc \varphi$ and $\forall X, s \models \psi$;
- $A, s \models \langle X \rangle \bigcirc \varphi$ iff $\forall X, s \models \varphi$ for all $0 \leq u < |\lambda|$;
- $A, s \models \langle X \rangle \bigcirc \varphi$ iff $\exists X, s \models \varphi$ and $\forall X, s \models \psi$;
- $A, s \models \langle X \rangle \bigcirc \varphi$ iff $\forall s \lambda(s_0, s) \sim_i \lambda(s_0, s')$, $A, s' \models \varphi$.

where $\sim_i$ is the transitive and reflexive closure of $\cup_{i \in X} \sim_i$.

3 Mapping GDL-II into ATEL

GDL-II serves as a language to describe games, while ATEL is for reasoning about such games. We build two links between GDL-II and ATEL.

- On the semantic level, every GDL-II description $G$ induces an AAETS model $A_G$.
On the syntactic level, every GDL-II description $G$ can be translated into an ATEL theory $G_{ATEL}$ (defined as $T_{syn}(G)$ below).

Thus, we are able to interpret ATEL formulas $\varphi$ over $G$ either via ATEL semantics, i.e., define $G \models \varphi$ as $\mathcal{A}_G \models \varphi$. Or, we can use the syntactic characterisation, i.e., define $G \models \varphi$ as $\models \varphi \in \mathcal{A}_G \models T_{syn}(G) \rightarrow \varphi$. As our main result, we will prove that these two characterisations are equivalent. The following diagram depicts the main idea, and we are now going to present it in detail.

```
G \quad T_{syn} \quad \mathcal{A}_G
has a derived AAETS model
has isomorphic AAETS models
```

Since ATEL does not support first-order predicates, we follow [13, 12] in applying a pre-processing step to GDL-II descriptions by replacing all predicates with variables, such as isdeal(?X,?Y), by all relevant instances (also called ground agents), such as isdeal(A, K). This maps an arbitrary GDL-II description into an equivalent variable-free specification. We then translate such ground atoms to atomic propositions in ATEL.

**Definition 7** (Translation $t$ and $t_{pre}$). Let $A_{GDII}$ be the set of ground atoms in GDL-II, and $A_{ATEL}$ be the set of atomic propositions in ATEL. The translation $t$ maps every GDL-II formula to an ATEL formula as follows.

*Base case:*

$$t(p) = p \quad \text{for all } p \in A_{GDII}$$

*Extended cases:*

$$t(\neg p) = \neg t(p)$$

$$t((p_1, \ldots, p_k)) = \{t(p_1), \ldots, t(p_k)\} \text{ for all literals } p_i.$$

Note that a literal is either $p$ or $\neg p$ for all $p \in A_{GDII}$. Furthermore, let $t_{pre}$ be defined by $t_{pre}(p) = t(p)_{pre}$ for the base case (and similar to $t$ for the extended cases). An atom $p_{pre}$ will represent the value of atom $p$ in the previous state (in ATEL). For convenience, we abbreviate $\text{does}(i, m)_{pre}$ by $\text{done}'(i,m)$.

We define how to induce an AAETS from a GDL-II description.

**Definition 8** (Semantic interpretation of GDL-II in ATEL). Given a GDL-II description $G$ with semantics $\langle \mathcal{R}, s'_0, t, l, u, T, g \rangle$, an AAETS for $G$ (denoted as $\mathcal{A}_G$) is a tuple

$$(Q, s_0, T, A_g, \{ A_{ci} \mid i \in A_g \}, \rho, \tau, \{ \sim_i \mid i \in A_g \}, \Phi, V)$$

where

- $Q$ is the set of states of $G$;
- $s_0 \in Q$ is the initial game state $s'_0$;
- $T$ is the set of terminal states of $G$;
- $A_g$ is the set of roles $R \setminus \{ \text{random} \}$ (assume n agents);
- $A_{ci} = \{ m \mid (i, m, s) \in T, s \in Q \}$ is the set of moves of agent $i$;
- $\tau : A_{ci} \times \cdots \times A_{ci} \times Q \rightarrow Q$ is a partial function that maps a set of action and a state to another state such that $\tau(M, s) = u(M, s)$;
- $\sim_i \subseteq Q \times Q$ is the accessibility relation for agent $i \in A_g$ given by $(s, s') \in \sim_i$ (also written as $s \sim_i s'$) iff role $i$ cannot distinguish any two developments $\delta$ and $\delta'$ such that $\delta = \langle s_0, \ldots, s \rangle$ and $\delta' = \langle s_0, \ldots, s' \rangle$ (cf. Definition 2);
- $\Phi \subseteq A_{ATEL}$ is a set of atomic propositions translated by $t$ and $t_{pre}$ from the atoms in $A_{GDII}$;
- $V : Q \rightarrow 2^\Phi$ is an interpretation function which associates with each state $s$ the set of atoms that satisfies the following requirements: if $p \in s$ then $t(p) \in V(s)$; if $s = u(M, s')$ then for any agent $i$ we have $\text{done}'(i, M(i)) \in V(s)$, and for any $p \in \sigma$ we have $t_{pre}(p) \in V(s)$; and $p_{pre} \in V(s_0)$.

The interpretation function requires that if there is a transition from $s'$ to $s$, then the moves that were made on $s'$ are recorded in $V(s)$ and the atoms true in $s'$ are also recorded in $s$ using the corresponding atoms labelled with $pre$. The accessibility relation of states is given according to the developments that end with such states. This ensures that $\mathcal{A}_G$ always has synchronicity and perfect recall.

Next, we give the full syntactic translation of GDL-II descriptions into ATEL formulas.

**Definition 9** (Syntactic Translation $T_{syn}$). Given a game description $G$, we define its ATEL theory $G_{ATEL} = T_{syn}(G)$ as a conjunction:

$$\Gamma_{ini} \land \Gamma_{norm} \land \Gamma_{log} \land \Gamma_{act} \land \Gamma_{mem} \land \Gamma_{next} \land \Gamma_{sees}.$$  

These conjuncts are described in detail as follows.

- $\Gamma_{ini}$. The initial state is captured by a conjunction of: all the fluent atoms that are initially true plus the extra atomic proposition $p_{init}$ (which is only true in $s_0$). Formally, $T_{syn} = \bigwedge_{i \in \mathcal{I}} (\text{true}(f_i) \land (p_{ini} \land (\langle i \rangle \land \langle i \rangle) \land \neg p_{init})$.

- $\Gamma_{norm}$. For normal rules (i.e., rules without any of the keywords role, init, next or sees in the head), we group those by heads:

$$r_1 : \quad p \Leftarrow bd_1 \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad r_k : \quad p \Leftarrow bd_k$$

Such rules decide whether $p$ is true in the current state. Let $R_{bd} = \{r_1, \ldots, r_k\}$, $\mathcal{H}$ be the heads of all such rules, and $AH$ be the set of atoms that do not appear in the heads of any rules and are not “does” atoms, then:

$$\Gamma_{norm} = \langle i \rangle \land (\forall p \in \mathcal{H} (CP(R_{bd}) \land LF(R_{bd}) \land \bigwedge_{p \in AH} \neg t(p)))$$

where $CP(R_{bd}) = t(p) \leftrightarrow (\forall i \in \mathcal{I} \land (\langle i \rangle \land (t(bd_i))))$, and $LF(R_{bd})$ is a loop formula (see below). Notice that $bd_i$ (a set of literals) is the body of rule $i$. Specially, if $bd_i$ is empty, then $\bigwedge (t(bd_i)) = T$, and if $p$ does not appear in the head of any rules and is not a “does” atom, then it must be false, which is captured by $\neg t(p)$. Formula $CP(R_{bd})$ applies Clark’s completion [2] to a given set of GDL-II rules $R_{bd}$. An example from the card trading game is:

$CP(R_{bd}) = \text{win} \leftrightarrow \text{goal}(b, 100)$. But in general the semantics of the completion of a (stratified) logic program is too weak to fully characterise the standard model in the presence of redundant rules like $p \Leftarrow p$. The standard model remains the same when such “superfluous” clauses are added, but Clark’s completion is weakened by them [7]. This issue is solved by a propositional formula denoted as $LF(R_{bd})$ (which is also called a loop formula); we refer to [7] for a detailed algorithm to compute such a formula.
• Γ_{eq}. In all non-terminal states, each agent must make one legal move. This means that if legal(i, m) is true in the current state, then agent i can enforce done(i, m) to be true in a next state and on the other hand if done(i, m) is true in the current state, then legal(i, m), must be true in the previous state, i.e., legal(i, m)_{prev} is true in the current state. This is captured by the following:

\[ \Gamma_{eq} = \{ \Box \lnot \text{terminal} \rightarrow \bigwedge_{i \in Ag} \bigwedge_{m \in Ac} (\text{legal}(i, m) \leftrightarrow \langle i \rangle \bigcirc \text{done}(i, m)) \land \\bigwedge_{i \in Ag} \bigwedge_{m \in Ac} (\text{done}(i, m) \rightarrow \text{legal}(i, m)_{prev}) \} \]

• Γ_{act}. For all non-initial states, each agent should have done exactly one action in the previous state, and agents always know what they did:

\[ \Gamma_{act} = \{ \Box \lnot \text{init} \rightarrow \bigwedge_{i \in Ag} \text{XOR}_{m \in Ac} \text{done}(i, m) \land \bigwedge_{i \in Ag} \bigwedge_{m \in Ac} (\text{done}(i, m) \rightarrow \text{legal}(i, m)_{prev}) \} \]

where XOR is the exclusive OR operator.

• Γ_{mem}. For non-terminal states, we use the atom t_{prev}(p) to record the truth-value of (p) for it to be used in the next states. Let At be At_{GDL-II} but without “done” atoms, then:

\[ \Gamma_{mem} = \{ \Box \lnot \text{terminal} \rightarrow \bigwedge_{i \in Ag} \text{XOR}_{m \in Ac} \text{done}(i, m) \land \bigwedge_{i \in Ag} \bigwedge_{m \in Ac} (\text{done}(i, m) \rightarrow \text{legal}(i, m)_{prev}) \} \]

• Γ_{next}. Suppose these are all the rules with head next(f):

\[ r_1 : \text{next}(f) \leftarrow \text{bd}_1 \]
\[ \ldots \]
\[ r_k : \text{next}(f) \leftarrow \text{bd}_k \]

There are two alternatives for translating these rules into ATEL formulas. From the perspective of a current state, the truth of f in the next state is determined by the truth of the propositions in the bodies of rules \( r_1, \ldots, r_k \) in the current state and the actions that are chosen by the agents for the transition. From the perspective of a next state, the truth of f in this state is determined by the previous truth of the propositions in the bodies of rules \( r_1, \ldots, r_k \) and the actions that have just been done. We adopt the second perspective. Let \( HN \) be the heads of all such rules, and define:

\[ \Gamma_{next} = \{ \Box \lnot \text{terminal} \rightarrow \bigwedge_{i \in Ag} \bigwedge_{\tau} (\text{true}(f) \leftrightarrow \bigwedge_{j \in [1..k]} (\Box \text{t}_{prev}(\text{bd}_j))) \} \]

• Γ_{sees}. The rules with “sees” are similar to those with “next”, but instead of defining what will be true they specify what will be seen by the agents next. Suppose these are all the rules with head sees(i, x):

\[ r_1 : \text{sees}(i, x) \leftarrow \text{bd}_1 \]
\[ \ldots \]
\[ r_k : \text{sees}(i, x) \leftarrow \text{bd}_k \]

Again we adopt the perspective of a next state. Let \( HS \) be the heads of all “sees” rules, and define:

\[ \Gamma_{sees} = \{ \Box \lnot \text{init} \rightarrow \bigwedge_{\text{sees}(i, x) \in HS} ((K_i \text{sees}(i, x) \leftrightarrow \bigwedge_{j \in [1..k]} (\Box \text{t}_{prev}(\text{bd}_j)))) \land (K_i \lnot \text{sees}(i, x) \leftrightarrow \lnot \bigwedge_{j \in [1..k]} (\Box \text{t}_{prev}(\text{bd}_j))) \} \]

Note that the size of the ATEL theory is polynomial in the size of a variable-free GDL-II description. Our translation is correct in the sense that the resulting ATEL formula is satisfiable in the ATEL model derived from the same game description.

**Proposition 1 (Correctness).** For a GDL-II game description G, \( A_G \models G_{ATEL} \).

To show that \( T_{syn} \) is an adequate syntactic characterisation of GDL-II descriptions, we define an equivalence relation on AAETS.

**Definition 10 (AAETS Isomorphism).** Let \( A = \langle Q, \delta, T, Ag, \{Ac_i | i \in Ag\}, r, \tau_i, \rho_i, \gamma_i \rangle \) and \( A' = \langle Q', \delta', T', Ag, \{Ac_i' | i \in Ag\}, r', \tau_i', \rho_i', \gamma_i' \rangle \) be two AAETSs. Then they are isomorphic (denoted as \( A \cong A' \)) iff there is a function \( f \) such that:

- \( f \) maps every state in \( Q \) to a state in \( Q' \) and it is a bijection; in particular, \( f(\delta_0) = \delta'_0 \), and for all \( s \in T, f(s) \in T' \);
- \( f \) maps every action in \( Ac \) to an action in \( Ac' \) and it is a bijection;
- for every state s and action \( m, s \in \rho(m) \iff f(s) \in \rho'(f(m)) \);
- for every state \( s, s' \) and joint action \( M, s = \tau(M, s') \iff f(s) = \tau'(f(M), f(s')) \);
- for every state s and agent i, \( s \sim_i s' \iff f(s) \sim_i f(s') \);
- for every proposition \( p \) and state \( s \in Q, p \in V(s) \iff p \in V'(f(s)) \).

The existence of an isomorphism between two AAETSs implies that they satisfy the same formulas.

**Proposition 2.** Given two AAETSs \( A \) and \( A' \) along with an arbitrary ATEL formula \( \varphi \),

\( A \cong A' \implies (A \models \varphi \iff A' \models \varphi) \).

**Proposition 3.** Let G be a game description and \( \varphi \) an ATEL formula, then the following holds

\[ \models G_{ATEL} \rightarrow \varphi \iff A_G \models \varphi \).

**Proof.** (Sketch) The direction from left to right follows from Proposition 1. The direction from right to left is proved by showing that for any AAETS \( A \) with synchronicity and perfect recall such that \( A \models G_{ATEL} \), there is an isomorphism between \( A \) and \( A_G \); the result then follows from Proposition 2. See [11] for details.

This is a main result in this paper. It shows that \( G_{ATEL} \) completely characterises G in ATEL in the sense that it entails any formula that is satisfied in the AAETS derived from G directly, and vice versa.

### 4 Model Checking Strategic and Epistemic Properties

Our main result in this paper allows us to consider the following model checking problem: given a game represented by GDL-II, and a property represented by an ATEL formula, decide whether the property is true for the game description. If the agents have incomplete information and perfect recall, the model checking problem for ATEL in traditional semantics is undecidable (see [3]). Hence, had we used the standard semantics, the above problem would also be undecidable since we can reduce the problem by deriving an ATEL model from a GDL-II description and then perform the ATEL model checking. However, with our new ATEL semantics—over AAETS models with finite computations—the model checking problem becomes decidable as we can give an algorithm for it.

We only sketch the algorithm below for the case of \( (X) \bigcirc \varphi \) due to space limitations. This algorithm terminates because only a finite number of strategies and computations needs to be checked.
How complex is the problem? We know for sure that the problem is at least 2EXPSPACE-hard due to [9], which shows that the outcome problem of the Private-PEEK game is complete in double exponential time (2EXPSPACE-complete). The Private-PEEK game can be rather straightforwardly specified in GDL-II and the outcome problem can be equivalently expressed as deciding whether $\langle 1 \rangle \diamond \mathsf{win}$, is true in the initial state of the game. Finding an upper bound for the complexity, however, is left for future work.

In [12], ATL is used to characterise some interesting playability properties for the original GDL games. With ATL being the language to express properties for GDL-II games, we can not only express the above properties but also a new class of properties that are not expressible in ATL. We discuss two kinds of such properties.

**Coherence Knowledge Properties.** There are some properties $\varphi$ that involve pure knowledge, i.e., where no temporal modalities occur in $\varphi$. For such $\varphi$, we call $\langle \langle \rangle \rangle \varphi$ a coherence knowledge property.

We know that $A, s_0 \models \langle \langle \rangle \rangle \varphi$ iff $\varphi$ is true in all reachable states from $s_0$.

In GDL-II, agents may not always know their legal moves. In order to check this, we can express the property that “if a move is legal for an agent then the agent knows it” as a formula:

$$\langle \langle \rangle \rangle \bigwedge_{k \in A, m \in Ag} (\mathsf{legal}(i, m) \rightarrow K_i \mathsf{legal}(i, m))$$

The following is not necessarily true for a GDL-II game: if the game has terminated, then this is common knowledge,

$$\langle \langle \rangle \rangle (\mathsf{terminal} \rightarrow C_A \mathsf{terminal})$$

If we want to ensure that a GDL-II description $G$ has this property, then we can either verify $A_G, s_0 \models \langle \langle \rangle \rangle (\mathsf{terminal} \rightarrow C_A \mathsf{terminal})$, or prove

$$\models G \mathsf{TEL} \rightarrow \langle \langle \rangle \rangle (\mathsf{terminal} \rightarrow C_A \mathsf{terminal}).$$

**Properties with knowledge and strategic power.** This class of properties mixes knowledge and coalition modalities, allowing us to talk about the agent’s knowledge and power simultaneously.

The following property says that if $i$ has a winning strategy, then he knows it:

$$\langle \langle \rangle \rangle \mathsf{win}_i \rightarrow K_i \langle \langle \rangle \rangle \mathsf{win}_i.$$ Suppose that in the current state $A, s \models K_i \langle \langle \rangle \rangle \mathsf{win}_i$. This does not give a winning strategy for agent $i$ explicitly. But agent $i$ can check the following on any state that he cannot distinguish from $s$ (i.e., $s \sim_i s'$):

$$A, s' \models \langle \langle \rangle \rangle (\mathsf{done}(i, m) \land \langle \langle \rangle \rangle \mathsf{win}_i).$$

If the above holds, then agent $i$ can safely choose $\mathsf{does}(i, m)$, and it still guarantees him a winning position in the next state.

We conclude with the following strategic property, which says that if $i$ knows $\varphi$, then he can ensure that agent $j$ knows $\psi$ next:

$$K_i \varphi \rightarrow \langle \langle \rangle \rangle \odot K_j \psi.$$  

**5 Related Work and Conclusion**

There are just a few papers on reasoning about games in GDL and its extension GDL-II. In [12], a method based on ATL is given to verify properties of general games, but this is restricted to original GDL and hence to games where players can maintain complete state information. Our paper extends this approach to GDL-II and uses a version of ATEL for this purpose. Our characterisation formula is inspired by the one given in [12], but we make these improvements: (1) we can deal with imperfect-information games; (2) we show that the models that satisfy this formula are isomorphic to $A_G$, rather than a weaker relation as alternating bisimulation given in [12]; (3) we do not require an extra “sink” state as there is no need to make computations infinite with our new semantics.

In [14], it is shown how GDL-II can be formally translated into the Situation Calculus as a first-order axiomatisation that allows players to reason about their perceptions and what they know about the legality and effects of moves based on the game description. In [10], the epistemic structure and expressiveness of GDL-II is analysed in terms of epistemic modal logic. It was shown that the operational semantics of GDL-II entails that the situation at any stage of a game can be characterised by a multi-agent epistemic (i.e., $S_5$-) model and GDL-II is sufficiently expressive to model any situation that can be described by a (finite) multi-agent epistemic model. Our work extends the static epistemic model into a dynamic ATEAs, and therefore a larger class of strategic and epistemic properties can be addressed by our approach.

**Acknowledgements** We thank the anonymous reviewers for their helpful comments. This research was supported under Australian Research Council’s (ARC) Discovery Projects funding scheme (project DP120102023). Michael Thielscher is the recipient of an ARC Future Fellowship (project FT0991348). He is also affiliated with the University of Western Sydney.

**REFERENCES**


