A Path in the Jungle of Logics for Multi-agent System: On the Relation between General Game-playing Logics and Seeing-to-it-that Logics

Emiliano Lorini
IRIT-CNRS, Toulouse University, France
Emiliano.Lorini@irit.fr

François Schwarzentruber
ENS Rennes, France
francois.schwarzentruber@ens-rennes.fr

ABSTRACT

In the recent years, several concurrent logical systems for reasoning about agency and social interaction and for representing game properties have been proposed. The aim of the present paper is to put some order in this 'jungle' of logics by studying the relationship between the dynamic logic of agency DLA and the game description language GDL. The former has been proposed as a variant of the logic of agency STIT in which agents' action are named, while the latter has been introduced in AI as a formal language for reasoning about general game-playing. The paper provides complexity results for the satisfiability problems of both DLA and GDL as well as a polynomial embedding of GDL into DLA.

Keywords

modal logic, STIT logic, general game playing, dynamic logic

1. INTRODUCTION

Several logics for modelling interaction in the context of multi-agent systems (MAS) and for the formal representation and specification of games have been proposed in the AI literature. Among them, we should mention multi-agent variants of propositional dynamic logics by [1] and [11], coalition logic by [22], alternating-time temporal logic by [2], coalition logic of propositional control by [25] and strategy logic by [21]. These logics aim to formally represent the concept of *capability*, that is, the consequences of the *potential* action of either an agent or a coalition of agents.[1] More recently, two additional systems have been added to the picture of logics for MAS: the family of seeing-to-it-that logics, whose most representative member is STIT by [4], and GDL by [28, 29].

On the one hand, STIT has been introduced in philosophical logic as a formal system for reasoning about (i) what an agent or a coalition *does* as a result of her/its individual/collective choice (*actuality*), and (ii) what an agent or a coalition *can do* as a result of her/its individual/collective choice (*potentiality*). It has been shown to be a valuable formal language for modelling socio-cognitive concepts such as emotion [18], responsibility [17] and intention [6]. One central feature of Belnap et al.'s STIT is that agents' actions are not named. Thus, this logic allows to represent the *result* of an agent's choice but does not allow to represent the *chosen action* that produces the result. For example, in this logic one can represent the fact that an agent 1 kills another agent 2 (agent 1 sees to it that agent 2 is dead) but one cannot represent the fact that agent 1 kills agent 2 *by* deciding to shoot. To overcome this limitation, DLA has been introduced by [10], as a variant of STIT in which agents' choices are explicitly named. A deterministic variant of DLA called DDLA has been proposed by [15]. While DDLA assumes that the outcome of the collective choice of all agents is uniquely determined, DLA drops this assumption.

On the other hand, GDL has been introduced in AI as a minimal formal system for representing game properties and strategies in the context of general game-playing. Similarly to seeing-to-it-that logics, GDL supports reasoning about both actual consequences and potential consequences of an agent's action. Moreover, similarly to DLA, in GDL, actions of agents are explicitly named. Thus, *at a conceptual level*, GDL and DLA are closely connected.

In this paper, we (i) explore the connection between DLA and GDL *at a formal level*, and (ii) thus solve some open issues about the computational properties of these two logics (see Table 1). For instance, we prove that the satisfiability problem of GDL is in NP by translating a GDL-formula into a DDLA-formula without any occurrence of the temporal operator "next" (X). This result highlights that GDL does not allow to express a property about the "next" state without mentioning the action leading to it. We also show that the satisfiability problems for DLA and for the non-deterministic variant of GDL become PSPACE-hard. Moreover, we prove that the complexity of the satisfiability problem for DDLA increases from 'in NP' to PSPACE-hard when the temporal operator "next" is given as a primitive operator in the logical language.

The paper is organized as follows. Section 2 is devoted to DLA and DDLA We first recall the syntax and the semantics of these two logics. Then, we investigate the complexity of the satisfiability problems of both DLA and DDLA, an issue that has not been explored up to now. Then, in Section 3, we move to GDL. The original GDL semantics for actions being deterministic, we present a new variant of GDL called IGDL, dropping the assumption that outcome resulting from the execution of an action is unique. We investigate the complexity of the satisfiability problems of both GDL and IGDL. Finally, in Section 4, we provide a polynomial embedding of GDL into DDLA and of IGDL into DLA helping proving upper bound of complexity for the satisfiability problems of GDL and IGDL. The paper concludes in Section 5 with a discussion of perspectives of future research.

2. DYNAMIC LOGIC OF AGENCY

We recall the syntax and the two semantics of the dynamic logic of agency: the non-deterministic one [10] called DLA and the deterministic one [15] called DDLA.

[1] Cf. [9] for a recent survey of existing logics for MAS.

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

NP-complete:	DDLA without X-modalities	$\leftrightsquigarrow$ GDL
PSPACE-complete:	DDLA	
	DLA	$\leftrightsquigarrow$ IGDL

Table 1: Logics with new complexity results for the satisfiability problem (symbols $\leftrightsquigarrow$ denote upper bound results obtained via embedding.)

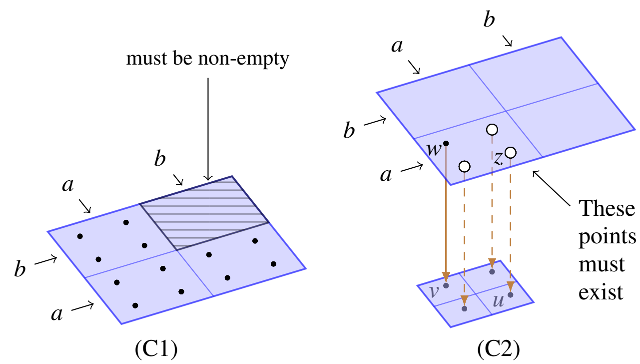


Figure 1: Illustrations for constraints C1 and C2

2.1 Syntax

Let *Atm* be a countable set of atomic propositions denoted by $p, q, \ldots$, let *Agt* be a finite set of agents denoted by $1, \ldots, n$ and let *Act* be a finite set of action names denoted by $a, b, \ldots$ The set of joint action names is defined to be $JAct = Act^n$. Elements of *JAct* are denoted by $\delta, \delta', \ldots$ For every $\delta \in JAct$, $\delta(i)$ denotes the element in δ corresponding to agent i. The language $\mathcal{L}_{\mathrm{DLA}}(Atm, Agt, Act)$ of DLA is defined by the following grammar:

$$\varphi, \psi \quad ::= \quad p \mid \neg\varphi \mid (\varphi \wedge \psi) \mid \Box\varphi \mid [\![\delta]\!]\varphi \mid \mathrm{X}\varphi$$

where p ranges over *Atm* and δ ranges over *JAct*.

The formula $\Box\varphi$ is read "φ is true for all histories passing through the current moment" or, more shortly, "φ is necessarily true". The formula $[\![\delta]\!]\varphi$ is read "if the joint action δ is chosen by the agents, then φ is true afterwards". The formula $\mathrm{X}\varphi$ is read "φ holds in the next moment". A formula is said to be *X-free* if it does not contain any occurrence of the temporal operator "next" X.

We provide the following abbreviations:

$$\Diamond\varphi \stackrel{\mathrm{def}}{=} \neg\Box\neg\varphi$$

$$\langle\!\langle\delta\rangle\!\rangle\varphi \stackrel{\mathrm{def}}{=} \neg[\![\delta]\!]\neg\varphi$$

$$\langle\!\langle i{:}a\rangle\!\rangle\varphi \stackrel{\mathrm{def}}{=} \bigvee_{\delta \in JAct : \delta(i)=a} \langle\!\langle\delta\rangle\!\rangle\varphi$$

that read respectively "φ is true for at least one history passing through the current moment", "the joint action δ is chosen by the agents and φ is true afterwards" and "action a is chosen by agent i and φ is true afterwards".

2.2 Semantics

The semantics of DLA is based on DLA models. A DLA model is a structure that provides information about agents' actual choices and available choices as well as about the consequences of the agents' choices. A DLA model can also be seen as a variant of a STIT model, as defined by [4], in which time is assumed to be discrete and agents' choices are explicitly named.[2]

Definition 1. *A DLA model is a tuple $M = (W, \equiv, (C)_{i \in Agt}, S, V)$ where:*

- *W is a non-empty set of worlds,*
- *$\equiv$ is an equivalence relation on W,*
- *$C_i : W \longrightarrow Act$ is the choice function for agent i,*
- *$S : W \longrightarrow W$ is a successor state function,*
- *$V : W \longrightarrow 2^{Atm}$ is a valuation function,*

such that for all $w, v, u \in W$ and $\delta \in \Delta$:

(C1) *if $\delta(i) \in CS_i(w)$ for all $i \in Agt$ then $\delta \in CS(w)$,*

(C2) *if $w \Rightarrow_\delta v$ and $v \equiv u$ then there exists $z \in W$ such that $w \equiv z$ and $z \Rightarrow_\delta u$,*

where:

- *$CS_i(w) = \{a \in Act : \exists v \in W \mid w{\equiv}v$ and $C_i(v) = a\}$;*
- *$CS(w) = \{\delta \in JAct : \exists v \in W \mid w{\equiv}v$ and $C(v) = \delta\}$;*
- *$\Rightarrow_\delta$ is the relation $\{(w, v) : S(w) = v$ and $C(w) = \delta\}$;*
- *$C(v) = (C_i(w))_{i \in Agt}$.*

In Definition 1, a world corresponds to a time point. Equivalence classes generated by the equivalence relation $\equiv$ are called *moments*. At every world w, every agent i chooses exactly one action denoted by $C_i(w)$. Let $\equiv(w) = \{v \in W : w \equiv v\}$ be the moment including world w. Agent i has a set of individual choices available at this moment denoted by $CS_i(w)$. Moreover, the agents have a set of collective choices available at this moment denoted by $CS(w)$. Every world w has exactly one successor denoted by $S(w)$: $S(w)$ is the world resulting from the agents' collective choice at world w. Function S can be associated with a serial and deterministic relation $\Rightarrow$ on W such $w \Rightarrow v$ iff $S(w) = v$. Let F be the transitive closure of $\Rightarrow$. $F(w) = \{(w, v) : wFv\}$ denotes the *history* starting in w, i.e., the sequence of worlds in the future of w.[3]

Constraint C1 says that if a_1 is a possible choice for agent 1,..., a_n is a possible choice for agent n then $(a_1, \ldots, a_n)$ is a possible collective choice. More intuitively, this means that agents can never be deprived of choices due to the choices made by other agents. Figure 1 illustrates constraint C1 in Definition 1: as it is possible for agent 1 to play b and for agent 2 to play b, there must be a world in which both agents play b. This is called *independence of choices* according to the STIT terminology. Constraint C2 expresses a basic relation between action and time: if v is in the future of w and u and v are in the same moment, then there exists an alternative z in the collective choice of all agents at w such that u is in the future of z. Figure 1 illustrates Constraint C2 in Definition 1. This is called *no choice between undivided histories* according to the STIT terminology. It captures the idea that if two histories come together in some future moment then, in the present, each agent does not have a choice between these two histories. This implies that if an agent can choose between two histories at a later stage, then she does not have a choice between them in the present.

[2] Belnap et al. call BT+ACs (Branching Time + Agent Choices) the structures over which the STIT language is interpreted.

[3] As shown by [10], one can safely assume that, for all $w, v \in W$, if wFv then $w \not\equiv v$, as this property is not modally characterizable in DLA. Since $\equiv$ is reflexive, the latter implies that $F(w)$ can be safely assumed to be a linearly ordered set.

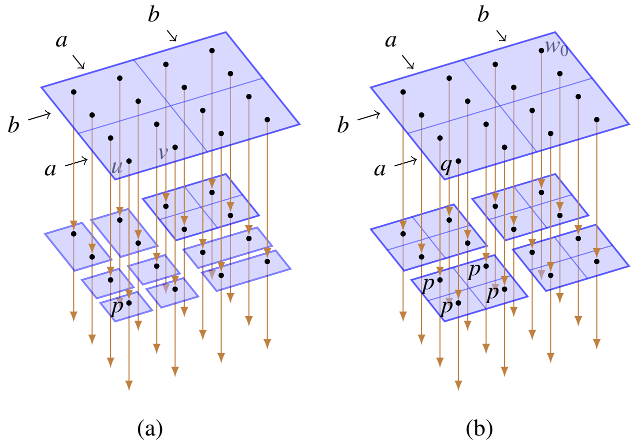


Figure 2: Structure of a (a) DLA model and (b) DDLA model

Truth conditions of DLA-formulas are given with respect to a world of a DLA model.

Definition 2 (Truth conditions). *Let $M = (W, \equiv, (C)_{i \in Agt}, S, V)$ be a* DLA *model and let $w \in W$. Then:*

$$\begin{aligned}
M, w \models p &\iff p \in V(w)\\
M, w \models \neg\varphi &\iff M, w \not\models \varphi\\
M, w \models (\varphi \wedge \psi) &\iff M, w \models \varphi \text{ and } M, w \models \psi\\
M, w \models \Box\varphi &\iff \forall v \in W: \text{ if } w \equiv v \text{ then } M, v \models \varphi\\
M, w \models [\![\delta]\!]\varphi &\iff \text{if } C(w) = \delta \text{ then } M, S(w) \models \varphi\\
M, w \models \mathsf{X}\varphi &\iff M, S(w) \models \varphi
\end{aligned}$$

As usual, a formula φ is DLA-*valid* iff φ is true for every DLA model M and every world w in M. A formula φ is DLA-*satisfiable* iff $\neg\varphi$ is not DLA valid.

The deterministic variant of DLA (DDLA) has the same language as DLA. It is interpreted with respect to deterministic DLA models that are defined as follows:

Definition 3. *A deterministic* DLA *model (*DDLA *model) is a* DLA *model such that for all $w, v \in W$:*

(C3) *if $w \equiv v$ and $C(w) = C(v)$ then $S(w) \equiv S(v)$.*

Constraint C3 says that the collective choice of all agents leads to a unique next moment. Notions of validity and satisfiability for DDLA with respect to DDLA models are defined in the usual way.

Let us illustrate the DLA and DDLA-semantics.

Example 1. *The left side of Figure 2 represents a* DLA *model that illustrates non-determinism: the collective choice (b, a) leads to two moments, the collective choice (a, a) leads to four moments, etc. The right side represents a* DDLA *model: for instance the collective choice (b, a) leads to a unique moment. Let us consider world w_0 in the model on the right. We have $C(w_0) = (b, b)$. Formula $\Box(\langle\!\langle (a, a)\rangle\!\rangle\top \rightarrow \mathsf{X}p)$ holds at w_0. Indeed, at moment $\equiv(w_0)$, if the joint action (a, a) is chosen (for instance at the unique q-world of the moment $\equiv(w_0)$ but also at the three other worlds where (a, a) is chosen), then p will be necessarily true in the next world.*

Example 2. *Formula $\mathsf{X}\varphi \rightarrow [\![\delta]\!]\varphi$ is* DLA*-valid (thus also* DDLA*-valid): if φ is true in the next state, then, in particular, if δ is actually played, then φ is true in the next state.*

Example 3. *Formula $(\langle\!\langle (a, a)\rangle\!\rangle\top \wedge \mathsf{X}\Box p) \rightarrow \Box[\![(a, a)]\!]p$ is a* DDLA*-valid formula. Indeed, suppose (a, a) is played and p is necessarily true in the next state. Then, necessarily if (a, a) is played, then p is true in the next state since the model is deterministic. But the formula is not* DLA*-valid. Indeed, $(\langle\!\langle (a, a)\rangle\!\rangle\top \wedge \mathsf{X}\Box p)$ holds in world u of the* DLA*-model shown in the left side of Figure 2, but $\Box[\![(a, a)]\!]p$ is false ($[\![(a, a)]\!]p$ does not hold at world v).*

2.3 Related work

The interesting aspect of DLA is its connection with STIT, the logic of "seeing to it that" by [4] and [13] one of the most prominent logical theories of agency proposed in the recent years.

Specifically, in [10], an embedding of the 'group STIT' by [13] with discrete time and bounded choices into DLA is given.[4] "Bounded choices" means that, at every moment in time, an agent can choose among at most n actions.

STIT logic, as defined by [4] and [13], is non-deterministic, as the consequence of the joint action of a coalition is not uniquely determined. This distinguishes STIT from the family of logics of coalitional power in games of which ATL (Alternating-time Temporal Logic) [2] and CL (Coalition Logic) [22] are the most representative ones, CL being the one-step fragment of ATL [7]. The standard semantics of ATL and CL is based on the concept of concurrent game structure (CGS) according to which, once every agent has made her choice at a given state, the successor state is uniquely determined.[5] In [15], an embedding of CL into DDLA is given.

The CL language includes the classical boolean constructions *plus* modal operators of coalition capability of the form $\langle\!\![J]\!\!\rangle$, where $\langle\!\![J]\!\!\rangle\varphi$ has to be read "the coalition of agents J *can* see to it that φ is true in the *next* state, regardless of what the agents outside J choose". The translation tr of the CL operator $\langle\!\![J]\!\!\rangle$ into the DLA language proposed by [15] is the following:

$$tr(\langle\!\![J]\!\!\rangle\varphi) = \bigvee_{\delta \in JAct} (\Diamond\langle\!\langle \delta_J\rangle\!\rangle\top \wedge \Box(\langle\!\langle \delta_J\rangle\!\rangle\top \rightarrow \mathsf{X}\varphi)).$$

2.4 Complexity of DLA and DDLA: Lower bound

For a given logic L, the satisfiability problem of L is denoted by L-sat.

Theorem 1. *If $|Agt| \geq 2$,* DDLA*-sat is PSPACE-hard.*

Proof. We give a polynomial-time reduction from the (PSPACE-hard) satisfiability problem for modal logic B defined by the class of infinite binary trees (cf. [5] and the use of this logic to prove a PSPACE-hardness result in [19]) to the satisfiability problem of DDLA. We write $\blacksquare$ for the operator of that logic B. We define $f(\blacksquare\varphi) = \Box\mathsf{X}tr(\varphi)$. We define the formula

$$twoActions(T) := \bigwedge_{t=0}^{T} \Box X^t[\Diamond\langle\!\langle \delta\rangle\!\rangle\top \wedge \Diamond\langle\!\langle \delta'\rangle\!\rangle\top \wedge \Box(\langle\!\langle \delta\rangle\!\rangle\top \vee \langle\!\langle \delta'\rangle\!\rangle\top)]$$

where $\delta = (a, a, \ldots a)$ and $\delta' = (b, a, \ldots, a)$ with $a \neq b$. Informally, $twoActions(T)$ means that until time T, agent 1 has exactly two actions a and b while the other players only can play action a.

The reduction is $tr(\varphi) := f(\varphi) \wedge twoActions(T)$ where T is the modal depth of φ. Formula φ is B-satisfiable iff $tr(\varphi)$ is DDLA-satisfiable. $\square$

[4]In the literature on STIT logic (cf. [3, 16, 12]), it is common to distinguish 'individual STIT' from 'group STIT'. The former only considers the consequences of the choices of individual agents, while the latter also considers the consequences of the collective choices of coalitions.

[5]Alternative semantics for CL and ATL based on the concept of effectivity function are presented respectively by [22] and [8].

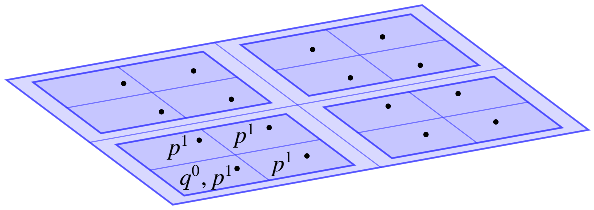


Figure 3: Fresh propositions in the flattened version of a DDLA model

Theorem 2. DDLA-*sat restricted to* X*-free formulas is NP-hard.*

Proof. The logic DDLA restricted to X-free formulas is a conservative extension of classical propositional logic. □

Theorem 3. DLA-*sat (even for* X*-free formulas) is PSPACE-hard.*

Proof. We polynomially reduce the (PSPACE-hard) satisfiability problem for modal logic K to it. We write $\blacksquare$ for the operator of K. Let δ be a fixed joint action. We define $tr(\blacksquare\varphi) := \Box[\![\delta]\!]tr(\varphi)$. Formula φ is K-satisfiable iff $tr(\varphi)$ is DLA-satisfiable. □

2.5 Complexity of DLA and DDLA: Upper bound

For giving upper bounds for satisfiability problems of DLA and DDLA, the idea is to flatten DLA models so that histories are represented by single worlds. As in [24], we introduce fresh propositions p^t saying that "proposition p is true at time t". Furthermore, we introduce propositions δ_i^t meaning that "δ_i is played at time t" and we write δ^t for $\bigwedge_{i=1}^n \delta_i^t$, meaning that the joint action δ is played at time t. These notations are used in the proofs of the following Theorems 4 and 5.

Theorem 4. DDLA-*sat restricted to* X*-free formulas is in NP.*

Proof. We polynomially reduce the satisfiability problem of DDLA restricted to X-free formulas to the satisfiability problem of S5, which is the modal logic where the accessibility relation is an equivalence relation. We recall that the satisfiability problem of S5 is in NP. The modal operator of S5 is denote by $\blacksquare$. The idea is that S5 models are flattened versions of DDLA models and we use modality $\blacksquare$ to simulate the modality $\Box$ and to guard the quantification to the current moment.

In the sequel, we will define a reduction tr. The idea of the translation $tr(\varphi)$ of φ is to simulate the semantics of DDLA in S5-models that look like the model shown in Figure 3.

In the world on the bottom left of the picture, q^0 and p_1 holds. That world corresponds to a history where q holds at time 0 and p holds at time 1. The big rectangle corresponds to the moment at time 0 and we use modality $\blacksquare$ to quantify over all worlds in that moment at time 0. The sub-rectangles corresponds to moment at time $t > 0$ (in picture, at time $t = 1$). Modality $\Box$ interpreted in the moment at time 1 after joint action δ at time 0 is simulated by the guarded construction $\blacksquare(\delta^0 \rightarrow ...)$. Modality $\Box$ interpreted in the moment at time 2 after joint action δ at time 0 and δ' at time 1 is simulated by the guarded construction $\blacksquare((\delta^0 \wedge \delta'^1) \rightarrow ...)$.

More generally, a moment at time t is actually identified with a sequence L of joint actions. That is why, we first define mappings tr_L from the language $\mathcal{L}_{\mathrm{DLA}}(Atm, Agt, Act)$ into the language of S5-formulas, indexed by a sequence of joint actions L. In the sequel, we write L for $\delta^{0^0} \wedge \cdots \wedge \delta^{t^t}$ if $L = (\delta^0, \ldots, \delta^t)$ (δ^{j^j} meaning that joint action δ^j is played at time j). We write ϵ for the empty sequence, $L{::}\delta$ for the sequence L where we added δ at the end of it, and $|L|$ for denoting the length of L.

We define a translation function tr_L by:

- $tr_L(p) = p^{|L|}$;
- $tr_L(\Box\varphi) = \blacksquare(L \rightarrow tr_L(\varphi))$;
- $tr_L([\![\delta]\!]\varphi) = (\delta^{|L|} \rightarrow tr_{L::\delta}(\varphi))$.

In the clauses $tr_L(p)$ and $tr_L([\![\delta]\!]\varphi)$, superscripts $|L|$ refer to time $|L|$. In the clause $tr_L(\Box\varphi)$, the guard $L \rightarrow ...$ ensures that we are in the current moment, uniquely determined by sequence L because the determinism of DDLA.

Example 4. *Let* $\varphi := \Box(p \rightarrow [\![\delta]\!]\Box q)$. *We have* $tr_\epsilon(\varphi) = \blacksquare(p^0 \rightarrow (\delta^0 \rightarrow \blacksquare(\delta^0 \rightarrow q^1)))$.

Let $\mathcal{S}(\varphi)$ be the set of sequences L such that $L = (\delta^0, \ldots, \delta^t)$ and $[\delta^0], \ldots, [\delta^t]$ is a list of nested modalities in that order in φ, of formulas. Note that the cardinality of $\mathcal{S}(\varphi)$ is polynomial in $|\varphi|$.

We define $C1(\varphi)$ to be the formula

$$\bigwedge_{L \in \mathcal{S}(\varphi)} \bigwedge_{\delta \in JAct} \left(\bigwedge_{i=1}^{n} \neg\blacksquare\neg(L \wedge \delta_i^{|L|}) \right) \rightarrow \neg\blacksquare\neg(L \wedge \delta^{|L|})$$

Intuitively, $C1(\varphi)$ expresses that the Constraint C1 of Definition 1 is true in all moments that the evaluation φ can reach, that is, in all moments reachable from a sequence L in $\mathcal{S}(\varphi)$. The guard $L \wedge ...$ ensures that we impose Constraint C1 in the moment uniquely determined by sequence L. We define $actions(T)$ to be the formula:

$$\bigwedge_{t=0}^{T} \blacksquare[(\bigvee_{\delta \in JAct} \delta^t) \wedge (\bigwedge_{\delta,\delta' \in JAct \delta \neq \delta'} \delta^t \rightarrow \neg\delta^t)]$$

Intuitively, formula $actions(T)$ expresses that there exists a unique joint action δ that is performed at each time t up to depth T.

The reduction is $tr(\varphi) = tr_\epsilon(\varphi) \wedge C1(\varphi) \wedge actions(T)$ where ϵ denotes the empty sequence and T the $[\delta]$-modal depth of φ. $tr(\varphi)$ can be computed in polynomial time in the size of φ. We have φ is DDLA-satisfiable iff $tr(\varphi)$ is S5-satisfiable. □

The rest of the section is devoted to prove:

Theorem 5. *Both* DLA-*sat and* DDLA-*sat are in PSPACE.*

For both DLA and DDLA, we adapt a polynomial-space non-deterministic algorithm given in [24].[6] Thus, by Savitch's theorem (NPSPACE = PSPACE) [23], Theorem 5 is proven. Let us define the multi-modal language, called $\mathcal{L}'$, whose modality are $[t]$ for all integer t denoting the time t. A construction $[t]\varphi$ means that φ holds for all histories passing through the moment at time t.

We first translate a DLA-formula φ into a $\mathcal{L}'$-formula $tr_0(\varphi)$ where tr_t, for any integer t, is defined by:

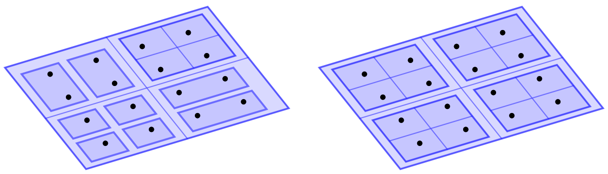
- $tr_t(p) = p^t$;
- $tr_t(\Box\varphi) = [t]tr_t(\varphi)$;
- $tr_t([\![\delta]\!]\varphi) = (\delta^t \rightarrow tr_{t+1}(\varphi))$;
- $tr_t(\mathrm{X}\varphi) = tr_{t+1}\varphi$.

Whereas $tr_L(\varphi)$ in the proof of Theorem 4 is written with a single type of modality, $tr_t(\varphi)$ contains several types of modality $[t]$ for each time t.

Example 5. *Let* $\varphi := \Box(p \rightarrow [\![\delta]\!](\Box q \wedge \mathrm{X}r))$. *We have* $tr_0(\varphi) = [0](p^0 \rightarrow (\delta^0 \rightarrow ([1]q^1 \wedge r^2)))$.

```
function satℒ′(t, ϕ, Σ : set of Hintikka sets)
  if there is a [t′] modality with t′ ≥ t in ϕ then
    Guess W a t-witness set of ϕ such that Σ ⊆ W
                              and |W| ≤ max(n_t(Σ) + |Σ|, |JAct|)
    Fail if the guess was not possible.
    for C : cells in W
      satℒ′(t + 1, ϕ, C)
```

Figure 4: Non-deterministic subroutine $\mathsf{sat}\mathcal{L}'$ used for the satisfiability problem of DDLA



(a) (b)

Figure 5: Flattened version of DLA **models and** DDLA **models**

We claim that a formula φ is DLA-satisfiable iff $tr_0(\varphi)$ is satisfiable in a Kripke structure where:

- K1 $[t]$-modality is interpreted by an equivalence relation R_t;
- K2 R_{t+1} is finer than R_t;
- K3 all points in an R_{t+1}-equivalence class satisfy the same joint action δ^t;
- K4 an adaptation of Constraint C1 of Definition 1 holds in each R_t-equivalence class with respect to propositions δ^t.

Such a Kripke structure can be depicted as in Figure 5(a). The big rectangle is the moment at time 0 (an R_0-equivalence class), sub-rectangles are sets of worlds where players choose the same joint actions at time 0, sub-sub-rectangles in the picture above are moment at time 1 (R_1-equivalence classes), etc. Contrary to the proof of Theorem 4 where we quantified over worlds in a moment at time t by guarded constructions, here we quantify over worlds in a moment at time t with modality $[t]$.

We moreover claim that a formula φ is DDLA-satisfiable iff $tr_0(\varphi)$ is satisfiable in a Kripke structure satisfying K1-K4 and:

- K5 an R_{t+1}-equivalence class consists exactly of those points where the same joint action δ^t is played.

Such a Kripke structure can be depicted as in Figure 5(b). The big rectangle is again the moment at time 0 (an R_0-equivalence class). But now, thanks to constraint K5, sub-rectangles and sub-sub rectangles coincide. Sets of worlds where agents choose the same joint actions at time 0 are exactly moments at time 1.

Let $\phi := tr_0(\varphi)$. The set $cl(\phi)$ is the smallest set containing ϕ, closed under subformulas, negations and such that if $[t]\psi \in cl(\phi)$ then $[t']\psi \in cl(\phi)$ for $t < t'$ and t' smaller than any t'' in φ. The size of $cl(\phi)$ is polynomial in $|\varphi|$, where $|\varphi|$ is the length of φ.

Now, we concentrate on the satisfiability problem of DDLA. We will now explain step by step the pseudo-code given in Figure 4. The algorithm takes as an input a time t, the formula $\phi := tr_0(\varphi)$ and a finite set Σ of Hintikka sets of ϕ defined as follows.

[6]It is the algorithm for the satisfiability problem of atemporal STIT for coalitions of the form $\{1, \dots, i\}$.

Definition 4. *A* Hintikka set *H of ϕ is a subset of $cl(\phi)$ such that:*

- *if $(\psi_1 \wedge \psi_2) \in H$ then $\psi_1 \in H$ and $\psi_2 \in H$;*
- *if $\neg(\psi_1 \wedge \psi_2) \in H$ then $\neg\psi_1 \in H$ and $\neg\psi_2 \in H$;*
- *if $\neg\neg\psi \in H$, then $\psi \in H$;*
- *$\psi \notin H$ or $\neg\psi \notin H$;*
- *for all t, there exists δ such that $\delta^t \in H$;*
- *for all t, for all $\delta \neq \delta'$, $\delta^t \notin H$ or $\delta'^t \notin H$;*
- *if $[t]\psi \in H$ then $\psi \in H$ and $[t']\psi \in H$ if $t' > t$.*

Intuitively, such a Hintikka set represents a world in a Kripke structure, that is, a history in a DDLA model.

The base case in the algorithm is when there is no $[t']$-modality with $t' \geq t$ in ψ: it means that we have considered enough steps in time. Otherwise, in the inductive case, we guess a t-witness set containing Σ. The role of a t-witness set is to represent the current moment at time t, that is, an R_t-equivalence class. Formally:

Definition 5. *A t-witness set W of ϕ is a set of Hintikka sets such that*

- *if for all $i \in \{1, \dots, n\}$, there exists $H_i \in W$ such that $\delta_i^t \in H_i$, then there exists $H \in W$ such that for all $i \in \{1, \dots, n\}$, $\delta_i^t \in H$;*
- *if $[t]\psi \in H \in W$ then for all $H' \in W$, $[t]\psi \in H'$;*
- *if $\neg[t]\psi \in H \in W$ then there exists $H' \in W$, $\neg\psi \in H'$;*
- *for all $t' < t$, for all δ if $\delta^{t'} \in H \in W$, then for all $H' \in W$, $\delta^{t'} \in H'$.*

The first item in Definition 5 is a reformulation of Constraint C1 of Definition 1. The second and third items correspond to the semantics of the $[t]$-operators. The last item means that previous joint actions (at time $t' < t$) are the same in all Hintikka sets. We impose such a t-witness to contain at most $\max(n_t(\Sigma) + |\Sigma|, |JAct|)$ Hintikka sets where $n_t(\Sigma)$ is the number of $[t]$-modalities in Σ. The bound $|JAct|$ guarantees that we have a sufficient number of worlds for Constraint C1 of Definition 1. The bound $n_t(\Sigma) + |\Sigma|$ is imposed because we do not need to create more worlds than the number of subformulas of the form $[t]\psi$ in ϕ that should be false.

We then perform all the recursive calls to explore consistency of next moments. Given a t-witness set W, a *cell* in W is the subset of Hintikka sets in W that all contain the same δ^t for a given δ. The fact that recursive calls of $\mathsf{sat}\mathcal{L}'$ are performed on cells corresponds to the constraint of determinism (C3) of Definition 3.

Proposition 1. *The two following statements are equivalent:*

- *$H(t, \phi, \Sigma)$: there exists a Kripke structure $\mathcal{M}$ satisfying $(K1-5)$ and a mapping f from Σ into a R_t-equivalence class such that for all $H \in \Sigma$, all formulas that do not contain any $[t']$-modality with $t' < t$ in H are true in $\mathcal{M}$ at point $f(\Sigma)$;*
- *$\mathsf{sat}\mathcal{L}'(t, \phi, \Sigma)$ does not fail.*

Proof. By induction on t. The base case corresponds to t such that there is no $[t']$-modality with $t' \geq t$ in ϕ. The result is true since we only consider Boolean formulas in Hintikka sets of Σ.

Inductive case. ($\Downarrow$) Let $\mathcal{M}$ be as in the first statement. We consider the execution of $\mathsf{sat}\mathcal{L}'(t, \phi, \Sigma)$ that guesses a t-witness set W

that is satisfied in $\mathcal{M}$, that is such that the mapping f can be naturally extended from W into the R_t-equivalence class such that for all $H \in W$, all formulas that do not contain any $[t']$-modality with $t' < t$ in W are true. Such a W exists since:

- all $\neg[t]\psi$ formulas that are true implies the existence of a world in the R_t-equivalence class such that $\neg\psi$ holds;
- and the number of such $\neg[t]\psi$ is bounded by $n_t(\phi)$. in $\mathcal{M}$ at point $f(\Sigma)$.

Hence, for all cells C in W, we have $H(t+1, \phi, C)$. By induction, we have that all calls $\mathsf{sat}\mathcal{L}'(t+1, \phi, C)$ do not fail. Hence, call $\mathsf{sat}\mathcal{L}'(t, \phi, \Sigma)$ does not fail.

($\Uparrow$) Conversely, suppose that $\mathsf{sat}\mathcal{L}'(t, \phi, \Sigma)$ does not fail. Let W be the t-witness set. We have that for all cells C in W, calls $\mathsf{sat}\mathcal{L}'(t+1, \varphi, C)$ do not fail. By induction, $H(t+1, \phi, C)$: there are $\mathcal{M}_C$ models corresponding to each cell C. We construct a model $\mathcal{M}$ by *gluing* models $\mathcal{M}_C$ and assigning the corresponding values to propositions p_t and δ_t: models $\mathcal{M}_C$ corresponds to 'sub-rectangles' at time $t+1$ and we juxtapose them to obtain a 'rectangle' at time t. We prove that we obtain a Kripke structure $\mathcal{M}$ satisfying $(K1-5)$ and such that there exists a mapping f from W into the R_t-equivalence class as required in $H(t, \phi, \Sigma)$. In particular, the restriction of f to Σ makes $H(t, \phi, \Sigma)$ true. □

Proposition 2. *Let H_0 be a Hintikka set containing ϕ. The call $\mathsf{sat}\mathcal{L}'(0, \phi, \{H_0\})$ requires a polynomial amount of memory in the size of ϕ.*

Proof. The number of nested recursive calls is clearly bounded by $O(|\phi|)$. We only require to store at most $O(|\phi|)$ parameters Σ and local variables. It suffices to convince ourselves that $|\Sigma|$ is polynomial in $|\phi|$. Let n_k be the maximal size of a set Σ at depth k in the tree of computation of $\mathsf{sat}\mathcal{L}'(0, \phi, \{H_0\})$. We have $n_0 = 1$. We have $n_k \leq max(n_t(\Sigma) + n_{k-1}, |JAct|) \leq max(|\phi| + n_{k-1}, |JAct|)$. So n_k is polynomial in $|\phi|$ for all k. □

The algorithm for the DDLA-sat is: compute $\phi := tr_0(\varphi)$, guess a Hintikka set H_0, call $\mathsf{sat}\mathcal{L}'(0, \phi, \{H_0\})$. The algorithm for DLA-sat is similar except that '**for** C : cell in $\mathcal{M}$' is replaced by '**for** C : singleton $\{H\}$ such that $H \in W$' in the algorithm $\mathsf{sat}\mathcal{L}'$ given in Figure 4. In other words, the Constraint C3 is dropped.

3. GAME DESCRIPTION LANGUAGE

In this section we shortly present the syntax and the semantics of a modal variant of GDL recently introduced in the AI community by [28, 29] as a general logic for reasoning about strategies in general game playing. We abstract away from Zhangh and Thielscher (Z&T)'s modality $\wr\alpha\wr$. At the end of the paper, we will introduce this modality when presenting perspectives of our future research.

GDL as defined by Z&T is deterministic. We also define a non-deterministic variant of GDL (IGDL) that relates with DLA.

3.1 Syntax

In GDL it is assumed that each agent i in the finite set of agents Agt is associated with a finite set of actions A_i defining i's action repertoire and that if $i \neq j$ then $A_i \cap A_j = \emptyset$. The set $A = \bigcup_{i \in Agt} A_i$ defines the set of all actions. The language $\mathcal{L}_{\mathrm{GDL}}(Prop, Agt, A)$ of GDL is defined by:

$$\begin{array}{rcl}\varphi, \psi & ::= & p \mid \neg\varphi \mid (\varphi \wedge \psi) \mid initial \mid terminal \mid \\ & & does(a) \mid legal(a) \mid [a]\varphi \mid \lfloor a \rfloor \varphi\end{array}$$

where p ranges over the set of atomic propositions Atm and a ranges over A.

Intuitively, formula $[a]\varphi$ means that "if action a is executed at the current state, φ is going to be true in the next state". Formula $\lfloor a \rfloor \varphi$ means that "if action a were chosen (but not yet executed), then φ would be true".

Constants *initial*, *terminal*, *does*(a) and *legal*(a) denote, respectively, the fact that the current state is an initial state, that the current state is a terminal state, that action a is executed and that action a is legal.

3.2 Semantics

The semantics of GDL is based on *state transition models*.

Definition 6. *A state transition model is a tuple $M = (X, I, T, U, L, V)$ such that:*

- *X is a non-empty set of worlds or states,*[7]
- *$I \subseteq X$ is the initial states,*
- *$T \subseteq X$ is the terminal states,*
- *$U : X \times A \longrightarrow X \setminus I$ is an update function,*
- *$L \subseteq X \times A$ is a legality relation, and*
- *$V : X \longrightarrow 2^{Atm}$ is a valuation function.*

Truth conditions of GDL formulas are given in the following definition.

Definition 7 (Truth conditions). *Let $M = (X, I, T, U, L, V)$ be a state transition model and let $(x, a) \in X \times A$. Then:*

$$\begin{array}{rcl}
M, (x,a) \models p & \Longleftrightarrow & p \in V(x) \\
M, (x,a) \models \neg\varphi & \Longleftrightarrow & M, (x,a) \not\models \varphi \\
M, (x,a) \models (\varphi \wedge \psi) & \Longleftrightarrow & M, (x,a) \models \varphi \text{ and } M, (x,a) \models \psi \\
M, (x,a) \models initial & \Longleftrightarrow & x \in I \\
M, (x,a) \models terminal & \Longleftrightarrow & x \in T \\
M, (x,a) \models does(b) & \Longleftrightarrow & a = b \\
M, (x,a) \models legal(b) & \Longleftrightarrow & (x,b) \in L \\
M, (x,a) \models [b]\varphi & \Longleftrightarrow & \forall c \in A : M, (U(x,b), c) \models \varphi \\
M, (x,a) \models \lfloor b \rfloor \varphi & \Longleftrightarrow & M, (x,b) \models \varphi
\end{array}$$

As usual, a formula φ is GDL valid iff φ is true for all state transition models $M = (X, I, T, U, L, V)$ and all $(x, a) \in X \times A$. A formula φ is GDL satisfiable iff $\neg\varphi$ is not GDL valid.

We here define a new non-deterministic variant of GDL (IGDL) that has the same language as GDL and that is interpreted with respect to *non-deterministic* state transition models.

Definition 8. *A* non-deterministic *state transition model is a tuple $M = (X, I, T, U, L, V)$ where X, I, T, L, V are as in the definition of state transition model and:*

- *$U : X \times A \longrightarrow 2^{X \setminus I}$ is a non-deterministic state transition function.*

The only difference about the interpretation of formulas between GDL and IGDL is relative to the modality $[b]$ that in IGDL is interpreted as follows:

[7] Z&T use the symbol W to denote the set of states. We use the symbol X to avoid confusion with the set of worlds in the semantics of DLA.

Definition 9 (Truth conditions (cont.)).
Let $M = (X, I, T, U, L, V)$ be a non-deterministic state transition model and let $(x, a) \in X \times A$. Then:

$$M, (x, a) \models [b]\varphi \quad \Leftrightarrow \forall c \in A \text{ and } \forall y \in U(x, b) : M, (y, c) \models \varphi$$

Notions of validity and satisfiability for IGDL with respect to non-deterministic state transition models are defined in the expected way.

4. RELATIONSHIP BETWEEN GDL AND DLA

In this section we provide a polynomial translation of the GDL language into the DLA language and we show that our translation preserves validity.

4.1 Embedding of GDL into DLA

Let us suppose that the cardinality of *JAct* (the set of joint actions in DLA) is equal to or bigger than the cardinality of A (the set of all actions in GDL). Hence there exists an injective function f from A to *JAct*: $f(a) \neq f(b)$ implies $a \neq b$. Moreover, let

$$Atm^+ = Atm \cup \{initial, terminal, legal\}.$$

Then, our translation is a function:

$$tr : \mathcal{L}_{\mathrm{GDL}}(Atm, Agt, A) \longrightarrow \mathcal{L}_{\mathrm{DLA}}(Atm^+, Agt, Act)$$

such that for all $\alpha \in Atm \cup \{initial, terminal\}$ and $a \in A$:

$$\begin{aligned}
tr(\alpha) &= \alpha \\
tr(\neg\varphi) &= \neg tr(\varphi) \\
tr(\varphi \wedge \psi) &= tr(\varphi) \wedge tr(\psi) \\
tr(does(b)) &= \langle\!\langle f(b) \rangle\!\rangle \top \\
tr(legal(b)) &= \Box(\langle\!\langle f(b) \rangle\!\rangle \top \rightarrow legal) \\
tr([b]\varphi) &= \Box[\![f(b)]\!]\Box tr(\varphi) \\
tr(\lfloor b \rfloor \varphi) &= \Box(\langle\!\langle f(b) \rangle\!\rangle \top \rightarrow \varphi)
\end{aligned}$$

Before proving that the previous translation preserves satisfiability, let us introduce the following concept of *φ-theory* expressed in terms of the DLA language. Let φ be a GDL formula. Moreover, let $Atm(\varphi)$ denote the set of atoms occurring in formula φ, let $depth(\varphi)$ denote the modal depth of φ and let $|\varphi|$ denote the length of φ. We define:

$$\begin{aligned}
\Sigma_\varphi = \; & \{\Box\alpha \vee \Box\neg\alpha : \alpha \in Atm(\varphi) \cup \{initial, terminal\}\} \cup \\
& \{\Box legal(\delta) \vee \Box\neg legal(\delta) : \delta \in JAct\} \cup \\
& \{\Box[\![\delta]\!]\neg initial : \delta \in JAct\} \cup \\
& \{\Diamond\langle\!\langle \delta \rangle\!\rangle \top : \delta \in JAct\}
\end{aligned}$$

where $legal(\delta) \overset{\text{def}}{=} (\langle\!\langle \delta \rangle\!\rangle \top \wedge legal)$. The φ-theory could have been defined by

$$\tau(\varphi) := \bigwedge_{0 \leq i \leq k} \Box \mathrm{X}^k \bigwedge_{\chi \in \Sigma_\varphi} \chi$$

where $k = depth(\varphi)$. But we can actually avoid using X-modalities by taking $\tau(\varphi)$ to be:

$$\tau(\varphi) = \bigwedge_{\vec{b} \in Seq(\varphi)} \Box[\vec{b}] \bigwedge_{\chi \in \Sigma_\varphi} \chi$$

where $\vec{b}$ denotes a sequence $b_1, \ldots, b_\ell$, $[\vec{b}]$ is $[b_1][b_2]\ldots[b_\ell]$ and $Seq(\varphi)$ is the set of sequences of actions as they appear in φ in the nesting order. Note that $Seq(\varphi)$ is polynomial in $|\varphi|$ therefore $|\tau(\varphi)|$ is polynomial in $|\varphi|$. The new definition $\tau(\varphi)$ clearly shows that we can express properties about the future by mentioning explicitly actions in the language without any occurrence of the temporal operator X.

Example 6. *If $\varphi = [a][b]p \wedge [c]q$ then $Seq(\varphi) = \{\epsilon, (a), (a, b), (c)\}$.*

The two following theorems highlight that the previous translation provides both an embedding of GDL into DDLA and an embedding of IGDL into DLA. They are proven by transforming a GDL model (IGDL model) into a corresponding DDLA model (DLA model) and vice versa.

Theorem 6. *Let φ be in $\mathcal{L}_{\mathrm{GDL}}(Atm, Agt, A)$. Then, φ is* GDL-*satisfiable iff $\tau(\varphi) \wedge tr(\varphi)$ is* DDLA-*satisfiable.*

Proof. ($\Rightarrow$) Let $M = (X, I, T, U, L, V)$ be a state transition model, let $x \in X$ and let $a \in A$ such that $M, (x, a) \models \varphi$.

For every world $x \in X$, let H_x denote the set of histories starting in state x, a history being an infinite sequence of state-action pairs ordered by the update function. That is:

$$\begin{aligned}
H_x = \; & \{x_0, a_0, x_1, a_1, \ldots : x_0 = x \text{ and} \\
& \forall k \geq 0 : U(x_k, a_k) = x_{k+1}\}
\end{aligned}$$

$H = \bigcup_{x \in X} H_x$ denotes the set of all histories. Histories are denoted by symbols $h, h', \ldots$ For every $h \in H$ and for every $k \geq 0$, $h_{state}(k)$ denotes the state in the k-th position in h, while $h_{action}(k)$ denotes the action in the k-th position in h.

For every history $h = x_0, a_0, x_1, a_1, \ldots$, $h_{\geq 1} = x_1, a_1, \ldots$ denotes the longest proper suffix of h.

We build a corresponding structure $M' = (W, \equiv, (C_i)_{i \in Agt}, S, V')$ as follows:

- $W = H$,
- for every $h, h' \in W$, $h \equiv h'$ iff $h_{state}(0) = h'_{state}(0)$,
- for every $i \in Agt$ and for every $h \in W$, $C_i(h) = f(h_{action}(0))$,
- for every $h \in W$, $S(h) = h_{\geq 1}$,
- every $p \in Atm$ and for every $h \in W$, $p \in V'(h)$ iff $p \in V(h'_{state}(0))$.

It is routine to check that M' is a DLA model.

We have that there exists $h \in H$ such that $h_{state}(0) = x$ and $h_{action}(0) = a$. Moreover, by induction on the structure of φ, we can prove that $M', h \models \tau(\varphi) \wedge tr(\varphi)$ for every $h \in H$ such that $h_{state}(0) = x$ and $h_{action}(0) = a$.

($\Leftarrow$) Let $M = (W, \equiv, (C)_{i \in Agt}, S, V)$ be a DLA model and let $w \in W$ such that $M, w \models tr(\varphi)$.

We call the quotient set $W/\equiv$ the set of moments and denote it by *Mom*. Elements of *Mom* are denoted by $m, m', \ldots$

We build a corresponding structure $M' = (X, I, T, U, L, V')$ as follows:

- $X = Mom$,
- $I = \{m \in X : \exists w \in m \text{ such that } M, w \models initial\}$,
- $T = \{m \in X : \exists w \in m \text{ such that } M, w \models terminal\}$,
- for every $m \in Mom$ and for every $a \in A$, $U(m, a) = m'$ iff $\exists w \in m$ such that $C(w) = f(a)$ and $S(w) \in m'$,
- $L = \{(m, a) \in X \times A : \exists w \in m \text{ such that } M, w \models legal(a)\}$,
- for every $p \in Atm$ and for every $m \in X$, $p \in V'(m)$ iff $\exists w \in m$ such that $p \in V(w)$.

It is routine to check that M' is a well-defined GDL model.

Moreover, by induction on the structure of φ, we can prove that $M', (m, f^{-}(C(w))) \models \varphi$ where f^{-} is the inverse function of f and $w \in m$. $\square$

Theorem 7. *Let φ be in $\mathcal{L}_{\text{GDL}}(Atm, Agt, A)$. Then, φ is* IGDL*-satisfiable iff $\tau(\varphi) \wedge tr(\varphi)$ is* DLA*-satisfiable.*

Proof. The proof is similar to proof of Theorem 6. As for the $\Rightarrow$-direction, we only need to change the definition of H_x as follows:

$$\begin{aligned} H_x &= \{x_0, a_0, x_1, a_1, \ldots : x_0 = x \text{ and} \\ &\quad \forall k \geq 0 : x_{k+1} \in U(x_k, a_k)\} \end{aligned}$$

As for the $\Leftarrow$-direction, we only need to change the definition of the function U as follows: for every $m \in Mom$ and for every $a \in A$, $m' \in U(m, a)$ iff $\exists w \in m$ such that $C(w) = f(a)$ and $S(w) \in m'$. $\square$

4.2 Complexities of GDL and IGDL

Although a sound and complete axiomatization for GDL has been given by [28], the complexity of its satisfiability problem remains an open problem.

Theorem 8. GDL*-sat is NP-complete.*

Proof. The NP-hardness comes from the fact that GDL is a conservative extension of classical propositional logic. The NP-membership comes from Theorem 4 and the polynomial-time reduction given in Theorem 6. $\square$

Theorem 9. IGDL*-sat is PSPACE-complete.*

Proof. The PSPACE-hardness is proven by providing the following polynomial reduction tr of the satisfiability problem of modal logic K into the satisfiability problem of IGDL: $tr(p) = p$, $tr(\Box\varphi) := [a]tr(\varphi)$. The PSPACE membership comes from Theorem 5 and the polynomial-time reduction given in Theorem 7. $\square$

Note that Theorem 8 strongly relies on the fact that $\tau(\varphi) \wedge tr(\varphi)$ does not contain any temporal operator X. The definition of the modal variant proposed by [28, 29] does not have this temporal operator in the language. Note that if we extend the language of GDL by the temporal operator X, and we call GDL_X the obtained language, we could reduce the satisfiability of GDL_X into DDLA and obtain a PSPACE-upper bound. As for DDLA, the satisfiability problem of GDL_X would have been PSPACE-hard.

Concerning Theorem 9, using a reduction involving the temporal operator X $\tau(\varphi) := \bigwedge_{0 \leq i \leq k} \Box \text{X}^k \bigwedge_{\chi \in \Sigma_\varphi} \chi$ would also have given an PSPACE-upper bound since Theorem 5 does not impose a restriction to X-free formulas.

5. PERSPECTIVES

The variant of GDL presented in [28] is equipped with an additional modal operator $\wr\psi\wr$ that we have not considered so far. It is a model-update operator in the sense of dynamic epistemic logic [27]. Formula $\wr\psi\wr\varphi$ is read "φ holds, after the declaration that only the states in which ψ is true are legal". It is interpreted as follows:

$$M, (x, a) \models \wr\psi\wr\varphi \iff M^\psi, (x, a) \models \varphi$$

where the state transition model M^ψ is the same as M except for replacing the legality relation L by the following relation: $L^\psi = \{(x, a) \in X \times A : M, (x, a) \models \psi\}$. Z&T provide reduction principles [28, Axioms D1-D8] that guarantee the existence of a reduction procedure such that, for every GDL formula φ, it returns an equivalent GDL formula φ' with no occurrence of the dynamic operators $\wr\psi\wr$. It is routine to verify that the reduction principles given by Z&T are also valid in the context of IGDL. Let us denote by *dyn*GDL (*dyn*IGDL) the GDL (IGDL) logic extended with $\wr\psi\wr$-modalities. Thus, because of Theorems 8 and 9, both *dyn*GDL-sat and *dyn*IGDL-sat are decidable.

Actually, in order to simulate $\wr\psi\wr\varphi$-modality in DLA, we extend DLA (and DDLA) with modal-update operators of type $[(p, \psi)]$, where the event (p, ψ) corresponds to a propositional assignment in the sense of [26]. The extensions are respectively denoted by *dyn*DLA and *dyn*DDLA. Formula $[(p, \psi)]\varphi$ is read "φ holds, after the declaration that atom p should be true only in the worlds in which ψ holds". It is interpreted relative to a DLA (or DDLA) model M and world w in M as follows:

$$M, w \models [(p, \psi)]\varphi \iff M^{(p,\psi)}, w \models \varphi$$

where $M^{(p,\psi)}$ is the same as M except for replacing the valuation function V by the following function $V^{(p,\psi)}$:

$$\begin{aligned} V^{(p,\psi)}(w) &= V(w) \cup \{p\} \text{ if } M, w \models \psi \\ V^{(p,\psi)}(w) &= V(w) \setminus \{p\} \text{ if } M, w \models \neg\psi \end{aligned}$$

It is routine to verify that the following formulas are both *dyn*DLA valid and *dyn*DDLA valid:

$$[(p, \psi)]p \leftrightarrow \psi \tag{1}$$

$$[(p, \psi)]q \leftrightarrow q \text{ if } p \neq q \tag{2}$$

$$[(p, \psi)]\neg\varphi \leftrightarrow \neg[(p, \psi)]\varphi \tag{3}$$

$$[(p, \psi)](\varphi_1 \wedge \varphi_2) \leftrightarrow ([(p, \psi)]\varphi_1 \wedge [(p, \psi)]\varphi_1) \tag{4}$$

$$[(p, \psi)]\blacksquare\varphi \leftrightarrow \blacksquare[(p, \psi)]\varphi \tag{5}$$

with $\blacksquare \in \{\Box, [\![\delta]\!], \text{X}\}$. Since the rule of replacement of equivalents preserves both *dyn*DLA and *dyn*DDLA validity:

$$\frac{\psi_1 \leftrightarrow \psi_2}{\varphi \leftrightarrow \varphi[\psi_1/\psi_2]} \tag{6}$$

by the preceding validities (1)-(5), we can find a reduction procedure such that, for every formula φ of the language $\mathcal{L}_{\text{DLA}}(Atm, Agt, Act)$ extended by the formulas $[(p, \psi)]\varphi$, it returns an equivalent formula φ' without any dynamic operator $[(p, \psi)]$. Because of Theorem 5, this reduction guarantees that *dyn*DLA-sat and *dyn*DDLA-sat are decidable.

Finally, *dyn*GDL $\wr\psi\wr$-constructions are translated in *dyn*DLA as follows: $tr(\wr\psi\wr\varphi) = [(legal, tr(\psi))]tr(\varphi)$. Theorems 6 and 7 generalize straightforwardly to this new case.

An interesting perspective of future research is to study the complexity of of *dyn*GDL-sat, *dyn*IGDL-sat, *dyn*DLA-sat and *dyn*DDLA-sat. To this aim, we plan to adapt the technique proposed by [20] to study the complexity of public announcement logic (PAL).

Another perspective is to go beyond games with perfect information and to study the relation between the epistemic extension of DLA given in [10] and the epistemic extension of GDL proposed by [14].

We also plan to consider a variant of GDL in which actions of agents are executed in parallel and with modal operators of type $[\delta]$, where $[\delta]\varphi$ has to be read "if the joint action δ is executed at the current state, φ is going to be true in the next state". We believe that this better corresponds to the way actions are represented in game theory. We plan to study its relationship with DLA and DDLA.

Finally, we plan to show how DLA and DDLA can be concretely used to model scenarios of general game playing and to highlight their advantages compared to IGDL and GDL.

REFERENCES

[1] T. Ågotnes and N. Alechina. Reasoning about joint action and coalitional ability in K n with intersection. In *Proceedings of 12th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA XII)*, volume 6814 of *LNCS*, pages 139–156. Springer-Verlag, 2011.

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

[3] P. Balbiani, A. Herzig, and N. Troquard. Alternative axiomatics and complexity of deliberative STIT theories. *Journal of Philosophical Logic*, 37(4):387–406, 2008.

[4] N. Belnap, M. Perloff, and M. Xu. *Facing the future: agents and choices in our indeterminist world*. Oxford University Press, New York, 2001.

[5] P. Blackburn, M. de Rijke, and Y. Venema. Modal logic, volume 53 of. *Cambridge Tracts in Theoretical Computer Science*, page 29, 2001.

[6] J. Broersen. Making a start with the *stit* logic analysis of intentional action. *Journal of Philosophical Logic*, 40(4):499–530, 2011.

[7] V. Goranko. Coalition games and alternating temporal logics. In *Proceedings of the 8th Conference on Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, pages 259–272. Morgan Kaufmann, 2001.

[8] V. Goranko and W. Jamroga. State and path coalition effectivity models of concurrent multi-player games. *Autonomous Agents and Multi-Agent Systems*, 30(3):446–485, 2016.

[9] A. Herzig. Logics of knowledge and action: critical analysis and challenges. *Autonomous Agents and Multi-Agent Systems*, 29(5):719–753, 2015.

[10] A. Herzig and E. Lorini. A dynamic logic of agency I: STIT, abilities and powers. *Journal of Logic, Language and Information*, 19(1):89–121, 2010.

[11] A. Herzig, E. Lorini, F. Moisan, and N. Troquard. A dynamic logic of normative systems. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 228–233. AAAI Press, 2011.

[12] A. Herzig and F. Schwarzentruber. Properties of logics of individual and group agency. In *Proceedings of the Seventh conference on Advances in Modal Logic (AIML 2008)*, pages 133–149. College Publications, 2008.

[13] J. F. Horty. *Agency and Deontic Logic*. Oxford Univ. Press, Oxford, 2001.

[14] G. Jiang, D. Zhang, L. Perrussel, and H. Zhang. Epistemic GDL: A logic for representing and reasoning about imperfect information games. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI 2016)*, pages 1138–1144. AAAI press, 2016.

[15] E. Lorini. A dynamic logic of agency II: deterministic DLA, Coalition Logic, and game theory. *Journal of Logic, Language and Information*, 19(3):327–351, 2010.

[16] E. Lorini. Temporal STIT logic and its application to normative reasoning. *Journal of Applied Non-Classical Logics*, 23(4):372–399, 2013.

[17] E. Lorini, D. Longin, and E. Mayor. A logical analysis of responsibility attribution: emotions, individuals and collectives. *Journal of Logic and Computation*, 24(6):1313–1339, 2014.

[18] E. Lorini and F. Schwarzentruber. A logic for reasoning about counterfactual emotions. *Artificial Intelligence*, 175(3-4):814–847, 2011.

[19] E. Lorini, F. Schwarzentruber, and A. Herzig. Epistemic games in modal logic: Joint actions, knowledge and preferences all together. In *Proceedings of the Second International Workshop on Logic, Rationality, and Interaction (LORI 2009)*, volume 5834 of *LNCS*, pages 212–226. Springer-Verlag, 2009.

[20] C. Lutz. Complexity and succinctness of public announcement logic. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)*, pages 137–143. ACM, 2006.

[21] F. Mogavero, A. Murano, and M. Y. Vardi. Reasoning about strategies. In *Proceedings of the Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2010)*, volume 8 of *LIPIcs*, pages 133–144, 2010.

[22] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.

[23] W. J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.

[24] F. Schwarzentruber. Complexity results of STIT fragments. *Studia Logica*, 100(5):1001–1045, 2012.

[25] W. van der Hoek and M. Wooldridge. On the logic of cooperation and propositional control. *Artificial Intelligence*, 164(1-2):81–119, 2005.

[26] H. van Ditmarsch, W. van der Hoek, and B. Kooi. Dynamic epistemic logic with assignment. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005)*, pages 141–148. ACM, 2005.

[27] H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Springer, 2008.

[28] D. Zhang and M. Thielscher. A logic for reasoning about game strategies. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2015)*, pages 1671–1677. AAAI Press, 2015.

[29] D. Zhang and M. Thielscher. Representing and reasoning about game strategies. *Journal of Philosophical Logic*, 44(2):203–236, 2015.