Game Description Language and Dynamic Epistemic Logic Compared

Thorsten Engesser[1], **Robert Mattmüller**[1], **Bernhard Nebel**[1], **Michael Thielscher**[2]
[1] Faculty of Engineering, University of Freiburg, Germany
[2] School of Computer Science and Engineering, University of New South Wales, Australia
{engesser,mattmuel,nebel}@cs.uni-freiburg.de, mit@unsw.edu.au

Abstract

Several different frameworks have been proposed to model and reason about knowledge in dynamic multi-agent settings, among them the logic-programming-based game description language GDL-III, and dynamic epistemic logic (DEL), based on possible-worlds semantics. GDL-III and DEL have complementary strengths and weaknesses in terms of ease of modeling and simplicity of semantics. In this paper, we formally study the expressiveness of GDL-III vs. DEL. We clarify the commonalities and differences between those languages, demonstrate how to bridge the differences where possible, and identify large fragments of GDL-III and DEL that are equivalent in the sense that they can be used to encode games or planning tasks that admit the same legal action sequences. We prove the latter by providing compilations between those fragments of GDL-III and DEL.

1 Introduction

Modeling and reasoning about knowledge in dynamic multi-agent settings has gained increasing interest in recent years [Baral *et al.*, 2017]. Several frameworks for this task have been proposed: In logic, dynamic epistemic logic (*DEL*) [van Ditmarsch *et al.*, 2007] has emerged, with an application to epistemic planning [Bolander and Andersen, 2011]. In the area of general game playing [Genesereth *et al.*, 2005], the game description language with imperfect information and introspection (*GDL-III*) [Thielscher, 2017] has recently been developed. Moreover, various other proposals of how to handle knowledge have been made in planning [Kominis and Geffner, 2015; Muise *et al.*, 2015].

Despite some superficial differences in their originally intended application areas, in their syntax, semantics, and execution frameworks, DEL and GDL-III can often be used as alternative modeling languages for the same problems, whenever knowledge must be made explicit within the model of a dynamic multi-agent system, for example in preconditions or goal specifications. This is required whenever multiple agents ought to gain, share, or hide knowledge [Cooper *et al.*, 2016].

However, both languages are far from perfect. In particular, DEL is based on possible-worlds semantics [van Ditmarsch *et al.*, 2007; Bolander and Andersen, 2011], which means that both the initial situation and all actions have to be specified as Kripke structures with explicitly given indistinguishability relations between worlds (or events) for all agents. Moreover, the description of an action may change depending on whether another agent observes its execution [Baral *et al.*, 2017]. This makes DEL relatively complicated to use for a non-expert modeler.

GDL-III is easier to use because indistinguishability between states, i. e., the knowledge of the agents, follows implicitly via *observation tokens* and action histories. This easier syntax, however, comes at a price: First, specification and modeling of *nested* knowledge is not as straightforward in GDL-III as it is in DEL. Second, it requires a more involved, inductive semantics that a GDL-III based general game player has to adhere to, which makes the analysis of the language more difficult. Hence, while it has been shown that possible worlds can be used to characterize the knowledge of players for the predecessor language GDL-II [Ruan and Thielscher, 2011], it is unclear whether the expressiveness of GDL-III matches that of other general languages for multi-agent epistemic reasoning and in particular DEL [Thielscher, 2017].

In order to determine whether the use of observation tokens in GDL-III and DEL event models are equally expressive, we investigate the commonalities and differences between the two languages. We demonstrate how to bridge the differences, and we formally show identical expressiveness of large fragments by providing compilations between GDL-III and DEL, along with soundness and completeness results.

Our results show that the use of observation tokens provides an equally expressive alternative to the use of Kripke structures for specifying actions in multi-agent epistemic domains. Problems can often be described in GDL-III more compactly, and more naturally, than in DEL. Hence, our analysis paves the way for the development of a description language for epistemic domains that combines the best of both worlds: the simple syntax of GDL-III, including observation tokens to model knowledge and its evolution, and the clean and simple semantics of DEL and DEL-based planning, including perspective shifts between the agents [Engesser *et al.*, 2017]. In addition, our results should help to transfer theoretical complexity results and solution concepts, e.g. the notion of implicitly coordinated plans in DEL-based planning [Engesser *et al.*, 2017], from one to the other formalism.

2 Background

We begin by recapitulating the foundations of DEL and GDL-III. To formally compare the two languages, we have to be able to talk about their states in a common language.

Let P be a finite set of atomic propositions and $\mathcal{A}$ be a finite set of agents. Then the *epistemic language* $\mathcal{L}_{KC}(P, \mathcal{A})$ is given by the BNF

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \phi \mid K_i\phi \mid C\phi, \text{ where } p \in P \text{ and } i \in \mathcal{A}.$$

The formula $K_i\phi$ reads *"Agent i knows that ϕ"* and $C\phi$ stands for *"ϕ is common knowledge between all agents"*. We will also use the abbreviations $\phi \vee \psi$, $\phi \rightarrow \psi$, $\phi \leftrightarrow \psi$, $\bot$ and $\top$, which are defined as usual in propositional logic.

2.1 Dynamic Epistemic Logic (DEL)

The term Dynamic Epistemic Logic refers to several modal logics of knowledge and the change thereof [van Ditmarsch *et al.*, 2007]. As base variant of DEL, we use the action model logic as described by Bolander and Andersen [2011]. For the compilation from GDL-III to DEL, we will introduce some extensions that allow us to keep the translation of actions as simple as possible. While it is possible to translate these action models back to our base version of DEL, it comes with a worst-case exponential increase in the size of the models.

Description of Static Facts and Knowledge

In DEL, epistemic formulas are evaluated on *epistemic models*. An epistemic model is a tuple $\langle W, (R_i)_{i\in\mathcal{A}}, V\rangle$ with a non-empty, finite set of *worlds* W, an *indistinguishability relation* $R_i \subseteq W \times W$ for each agent $i \in \mathcal{A}$, and a *valuation function* $V : P \rightarrow \mathcal{P}(W)$, mapping each proposition to the set of worlds in which they are true. Since we talk about *knowledge* in this paper (e.g., in contrast to *belief*), we require our indistinguishability relations to be equivalence relations. The idea is that agents cannot distinguish between worlds that belong to the same equivalence class of their respective indistinguishability relations. We visualize epistemic models as undirected graphs, where the set of nodes is W and the edges between two worlds w and w' are labeled with all agents i such that wR_iw' (or left out, if there is no such i). Nodes are additionally labeled with the world name and the propositions that are true in that world. Note that for improved readability we leave out all reflexive edges and sometimes also edges that are implied by transitivity. Consider the following example:

$$\mathcal{M} = \underset{w_1 : p, q}{\bullet} \overset{1}{\text{———}} \underset{w_2 : p}{\bullet} \overset{2}{\text{———}} \underset{w_3 :}{\bullet}$$

Let us assume that w_1 is the *actual* world. Then agent 1 knows that p is true, since it is true in both worlds which he considers possible (w_1 and w_2). However, he does not know whether or not q is true. Trivially, agent 2 knows that both p and q are true. But agent 1 does not know that agent 2 knows this, since if w_2 was the actual world, agent 2 would not be able to rule out being in w_3 where p (and also q) is false.

Formally, the evaluation semantics is given as follows, where the propositional cases are standard and hence left out:

$$\begin{array}{lll} \mathcal{M}, w \models p & \text{iff} & w \in V(p) \\ \mathcal{M}, w \models K_i\phi & \text{iff} & \mathcal{M}, w' \models \phi \text{ for all } wR_iw' \\ \mathcal{M}, w \models C\phi & \text{iff} & \mathcal{M}, w' \models \phi \text{ for all } wR^*w' \end{array}$$

where R^* is the transitive closure of $\bigcup_{i\in\mathcal{A}} R_i$.

Description of Factual and Knowledge Change

An event model in the style of Bolander and Andersen [2011] is a tuple $\mathcal{E} = \langle E, (Q_i)_{i\in\mathcal{A}}, \mathrm{pre}, \mathrm{eff}\rangle$ with a non-empty, finite set of *events* E and for each agent $i \in \mathcal{A}$ an indistinguishability relation $Q_i \subseteq E \times E$ (again – since we talk about knowledge – with all Q_i being equivalence relations). Furthermore, it contains functions $\mathrm{pre} : E \rightarrow \mathcal{L}_{KC}(P, \mathcal{A})$ and $\mathrm{eff} : E \rightarrow \mathcal{L}_{KC}(P, \mathcal{A})$ assigning preconditions (arbitrary formulas) and effects (conjunctions of propositional literals) to each event. We depict event models similar to epistemic models, with the events being the nodes in a graph. As with epistemic states, the edges between the events are annotated with the agents for which the events are indistinguishable. Additionally, each node is annotated with the event name e followed by $\langle \mathrm{pre}(e), \mathrm{eff}(e)\rangle$. As an example, consider the following event model that describes the action of agent 1 simultaneously switching and possibly (in the case that p is true) sensing the value of proposition q:

$$\mathcal{E} = \begin{array}{ccc} e_1 : \langle p \wedge q, \neg q\rangle & & e_2 : \langle p \wedge \neg q, q\rangle \\ \bullet & \overset{2}{\text{———}} & \bullet \\ \Big|2 & & \Big|2 \\ \bullet & \overset{1,2}{\text{———}} & \bullet \\ e_3 : \langle \neg p \wedge q, \neg q\rangle & & e_4 : \langle \neg p \wedge \neg q, q\rangle \end{array}$$

Our event model contains one event for all possible valuations over $\{p, q\}$. All events are indistinguishable for agent 2. Agent 1's indistinguishability relation consists of the equivalence classes $\{e_1\}, \{e_2\}$, and $\{e_3, e_4\}$, corresponding to the observations that q was true, false, or that it cannot be sensed (which implies that p must be false).

To compute the update of a state model and an event model, each world w is paired up with each event e for which the precondition is satisfied, meaning $\mathcal{M}, w \models \mathrm{pre}(e)$. Resulting worlds (w, e) and (w', e') are indistinguishable for some agent i if both the predecessor worlds and the events were indistinguishable for that agent (wR_iw' and eQ_ie'). The set of true propositions in a world (w, e) is the set of true propositions in w plus the positive and minus the negative literals occurring in $\mathrm{eff}(e)$. E.g., by applying our event model $\mathcal{E}$ to $\mathcal{M}$, we obtain the following model:

$$\mathcal{M} \otimes \mathcal{E} = \begin{array}{ccc} \bullet & & \bullet \overset{2}{\text{———}} \bullet \\ (w_1, e_1) : p & & (w_2, e_2) : p, q \quad (w_3, e_4) : q \end{array}$$

We can see that if the actual world was w_1 initially, since the only applicable event is e_1, in the successor model $\mathcal{M}\otimes\mathcal{E}$, we end up in world (w_1, e_1), where it is common knowledge between the agents that now p is true and q is false. If the actual world was w_2 or w_3 initially, we end up in world (w_2, e_2), or (w_3, e_4) respectively, where it is common knowledge that q is true and that agent 1 (but not agent 2) knows the value of p. We will write $\mathcal{M} \otimes \mathcal{E}^n$ for the n-fold update of $\mathcal{M}$ with $\mathcal{E}$.

Epistemic Actions and States

While an event model specifies all events that any agent might consider possible, for defining a single *action* we additionally have to designate the events that the acting agent actually seeks to perform. Sometimes we need the outcome of an action to be chosen nondeterministically. For that purpose we

employ *multi-pointed* event models. E.g., for an event model $\mathcal{E}$ with two possible events e_1 and e_2 we can define three reasonable actions: $(\mathcal{E}, \{e_1\})$, $(\mathcal{E}, \{e_2\})$ and $(\mathcal{E}, \{e_1, e_2\})$. One example where we need nondeterminism is sensing actions. E.g., for our example action, all four events of the event model have to be designated. This is because agent 1 cannot know the sensing result in advance, yet the action is supposed to be applicable in any case, even if p is false.

Formally, an *epistemic action* $(\mathcal{E}, E_d)$ is an event model $\mathcal{E}$ together with a set of designated events E_d, which is a subset of the set of all events of $\mathcal{E}$. An *epistemic state* $(\mathcal{M}, w)$ is a model $\mathcal{M}$ together with a designated world w of the model An action $(\mathcal{E}, E_d)$ is said to be *applicable* in $(\mathcal{M}, w)$ if there exists a designated event $e \in E_d$ such that $\mathcal{M}, w \models \mathrm{pre}(e)$. The application of $(\mathcal{E}, E_d)$ in $(\mathcal{M}, w)$ results in a successor state $(\mathcal{M} \otimes \mathcal{E}, (w, e))$ such that $e \in E_d$ and $\mathcal{M}, w \models \mathrm{pre}(e)$.

2.2 Game Description Language GDL-III

The Game Description Language (GDL) was developed for the purpose of specifying the rules of arbitrary games to general game-playing systems [Genesereth *et al.*, 2005]. The most recent version is GDL-III, which can express both incomplete information and *epistemic* game rules, i.e., which refer to the knowledge of players [Thielscher, 2017].

Unlike DEL, the game description language is based on the syntax of logic programs with negation [Lloyd, 1987]. Different elements of a game description are specified by using *reserved keywords* [Genesereth and Thielscher, 2014].

Description of Agents, States and Actions

Agents are defined by a finite set of ground facts using the keyword `role`. A special agent is the predefined role `random`, which always uniformly chooses one of its legal actions randomly. An (actual) *initial* state is specified with the help of the reserved predicate `init`, for example:

```
role(ag1).
role(ag2).
init(p).
```

The principle of negation-as-failure [Clark, 1978] applies here and elsewhere in GDL. For example, since the given rules do not imply `init(q)`, proposition `q` is initially false.

Preconditions for an action a taken by agent i are defined by rules with the keyword $\texttt{legal}(i, a)$ in the head. The body of these rules can use any of the following three keywords: $\texttt{true}(f)$, saying that f must hold in the current state; $\texttt{knows}(i, f)$, to require that agent i knows f; and $\texttt{knows}(f)$, which means that f is common knowledge among all agents.

The *state update* effected by an action a is described with the keyword $\texttt{next}(f)$ to define the propositions, also known as *fluents*, that are true after the execution of a. The rule body can include keywords `true` and `knows` along with the keyword $\texttt{does}(i, a)$, which is true when agent i takes action a. For example, we can model the action of agent 1 from Section 2.1 in GDL-III as follows:

```
legal(ag1,a).
next(q) :- does(ag1,a), not true(q).
next(p) :- does(ag1,a), true(p).
```

According to these rules, action `a` by `ag1` is always possible. Its effect is to switch the state of `q` (note that when `true(q)` holds, `next(q)` cannot be derived and hence `q` is false in the next state), while `p` remains unchanged.

Description of Observations

Effect axioms are accompanied by *observation* rules with the reserved keyword $\texttt{sees}(i, t)$ in the head. They define the conditions under which agent i receives an *observation token* t. These can be used for very succinct specifications of epistemic actions. For example, the epistemic effect of the action from above, namely that agent 1 senses the value of proposition q under the condition that p holds, can be expressed in GDL-III as follows:

```
sees(ag1,qT) :- does(ag1,a),true(p),true(q).
sees(ag1,qF) :- does(ag1,a),
                true(p), not true(q).
```

Observation tokens are what make representations of multi-agent knowledge and epistemic actions so concise in GDL-III. Only objective rules about what players can see need to be specified; rules about how actions and percepts affect the knowledge of players are not required. For example, from the absence of an observation token for agent 2, it follows implicitly from the semantics of GDL-III that all events corresponding to action a are indistinguishable for this agent.

There are also keywords for defining conditions for a game to end and how a game is won; however, these are not relevant for the purpose of this paper and will therefore be ignored.

Semantics

The semantics of GDL-III combines the standard semantics of logic programs with negation[1] with an inductive definition of how a knowledge state evolves. The initial state of a game described by rules G consists of all f for which $\texttt{init}(f)$ can be derived from G.

The rules for `legal`, `next`, and `sees` determine a knowledge state transition system as follows: Let (S, K) be an arbitrary *knowledge state*, where S is a set of ground `true`-instances and K a set of ground `knows`-instances. An action a of agent i is *legal* in (S, K) if $\texttt{legal}(i, a)$ can be derived from $G \cup S \cup K$. A *joint action* M is a set of $\texttt{does}(i, a)$-instances, one action for every agent i. The *resulting state* when M is executed in (S, K) consists of all $\texttt{true}(f)$ such that $\texttt{next}(f)$ can be derived from $G \cup S \cup K \cup M$. Agent i *observes* token t when M is executed in (S, K) iff $\texttt{sees}(i, t)$ can be derived from $G \cup S \cup K \cup M$.

By definition, the initial state is common knowledge among the agents. A *legal play sequence* is a sequence of joint actions, beginning in the initial knowledge state, in which all agents always select a legal action. Legal play sequences δ and δ' are *indistinguishable* by agent i, written $\delta \sim_i \delta'$, iff i's actions and observations are the same in δ and δ'. Agent i *knows* a property ϕ after a legal play sequence δ iff ϕ is true in all δ' that i cannot distinguish from δ. Finally, ϕ is *common knowledge* after δ if it holds after all δ' in the transitive closure $\sim^C$ of $\bigcup_i \sim_i$.

[1] Syntactic restrictions [Genesereth *et al.*, 2005] ensure that GDL descriptions always have a *finite grounding* and admit the so-called *standard model* [Apt *et al.*, 1987], which is the same as their unique *answer set* [Gelfond, 2008].

Given a game description G, we can determine a set of propositional variables P that contains exactly the grounded propositions from G [Schiffel and Thielscher, 2009]. Furthermore, we can assume a set $\mathcal{A}$ containing exactly the r which occur in a fact $\texttt{role}(r)$. With this, we can evaluate arbitrary formulas $\phi \in \mathcal{L}_{KC}(P, \mathcal{A})$ over arbitrary GDL-states (G, δ). For propositional formulas, truth is defined inductively based on the `init` and `next` rules. For knowledge formulas, we have $G, \delta \models K_i\phi$ iff $G, \delta' \models \phi$ for all δ' with $\delta \sim_i \delta'$ and $G, \delta \models C\phi$ iff $G, \delta' \models \phi$ for all δ' with $\delta \sim^C \delta'$.

3 Differences between GDL-III and DEL

Before translating between GDL-III and DEL, we briefly discuss some differences between them and why these are unproblematic for the translations. Where necessary, we point out how they can be dealt with by either restricting the more expressive or enhancing the less expressive formalism. The differences can be grouped into syntactic and semantic differences, and differences pertaining to the execution model.

Syntax. GDL-III is a *first-order* language while DEL is *propositional*. However, syntactic restrictions ensure that GDL-III game descriptions have a finite grounding [Genesereth *et al.*, 2005]. Furthermore, unlike GDL-III, DEL does not know *derived fluents*. We deal with this difference by disallowing cyclic definitions and unrolling all definitions in GDL-III first. Finally, in GDL-III, observations can be *state-dependent*. In other words, the same action can yield different observations to the same agent depending on the state in which it occurs. The same cannot be expressed as easily and compactly in standard DEL. To overcome this difference, we will introduce a variant of DEL that allows a form of *edge-conditioned event models* similar to those of Bolander [2014].

Semantics. GDL-III uses *successor-state axioms* to derive the next state, whereas DEL uses *state updates*. To address this mismatch, we assume that the GDL-III game description gives us a precise condition $\varphi_a(f)$ under which action a makes fluent f true, which can be obtained following the solution of the inferential frame problem in GDL of Romero *et al.* [2014]. We can then use a variant of DEL with conditional effects [van Benthem *et al.*, 2006] of the form $f \mapsto \varphi_a(f)$.

Execution model. In GDL-III, moves are associated with players allowed to make them, whereas action ownership is not part of DEL itself. Related to that, GDL-III assumes *joint moves* where all players move simultaneously in each step. While we could define DEL planning domains assigning DEL actions to each *action profile* (containing the decision of each agent), our solution is to assume that GDL-III games are sequentialized in the standard way [Rasmusen, 2007]. We make the assumption that as a result of the sequentialization, after each play sequence the GDL-III description allows exactly one agent to freely choose between actions. All other agents have only one action available *waiting* for the active agent. We assume these waiting actions to be identifiable from the GDL-III description. Most importantly, they are not allowed to have any effects on the successor state or on the received observation tokens (in particular, they should not occur in any of the `does`-instances of the GDL description).

Since GDL-III does not require knowledge of preconditions [Moses, 2015], we do not assume this knowledge of the action's owner in DEL, either. Finally, whereas GDL-III is most commonly viewed as a language for encoding *competitive* settings, DEL planning is more about *cooperation* towards achieving a joint goal. Since the expressive power of the formalization languages does not depend on the goal conditions, we ignore them in this paper.

4 An Extension of DEL

In Section 2.2, we exemplified how an action with epistemic effects can be compactly described in GDL, whereas corresponding DEL event models require an exponential number of events in the worst case, both in the number of observation tokens and in the number of propositions that can change conditionally. In the following, we propose a modified version of event models, which allow us to model GDL-III actions in a more succinct way while not making the computation of the product update harder. Most importantly, it will simplify our compilation from GDL to DEL, making each action representable by one single event in a larger underlying event model which is common to all actions.

A More Succinct Representation of Events

Our first change is that we replace the indistinguishability relation by a partial *edge-condition function* $Q_i : E \times E \nrightarrow \mathcal{P}(\mathcal{L}_{KC}(P, \mathcal{A}) \times \mathcal{L}_{KC}(P, \mathcal{A}))$, mapping event pairs to a finite set of *edge conditions* for each agent $i \in \mathcal{A}$. The idea behind these edge conditions is the following: Determining whether two worlds (w, e) and (w', e') will be indistinguishable should be dependent on additional information (e.g., observation tokens) received by the agents, which can be conditionalized both on the actual predecessor world (w or w') and on the event that actually occurred (e or e'). We say that an edge condition $(\phi, \psi) \in Q_i(e, e')$ is satisfied between w and w' if $\mathcal{M}, w \models \phi$ iff $\mathcal{M}, w' \models \psi$. If all edge conditions are satisfied, this means the additional information received by the agent is identical in w and w' and that i thus cannot distinguish (w, e) from (w', e'). We leave $Q_i(e, e')$ undefined if we want e and e' to be always distinguishable. If $Q_i(e, e')$ is defined, we also write eQ_ie'. In spirit, our edge-conditioned event models work very similarly to the event models used in generalized arrow update logic [Kooi and Renne, 2011].[2]

We additionally adopt a more expressive version of effects in the style of van Benthem *et al.* [2006], which resemble the effect axioms of GDL much more closely. We define our *conditional effects* by a function $\text{eff} : E \to \{P \to \mathcal{L}_{KC}(P, \mathcal{A})\}$, which specifies for each proposition a formula that is to be evaluated in the parent worlds to determine the new value for that proposition. In our depiction of event models, we sometimes omit effects of the form $p \mapsto p$. Using these extensions, we can express the action from our initial example as

$$\mathcal{E}' = \quad \begin{array}{c} 1 : \{(p \wedge q, p \wedge q), (p \wedge \neg q, p \wedge \neg q)\}, 2 : \emptyset \\ \circlearrowleft \\ \bullet \\ e_1 : \langle \top, \{q \mapsto \neg q\} \rangle. \end{array}$$

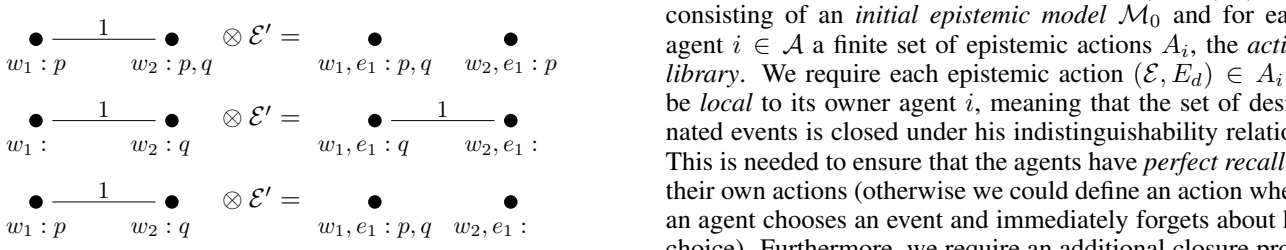
[2] The slightly differing semantics (we have universally quantified *iff*-conditions instead of existentially quantified *and*-conditions) is vital to the succinctness of our translation, as we will see later.

The event model can be interpreted as follows: There is only one possible event which always switches the value of q. After the event, situations which were previously indistinguishable, but which differed in terms of $p \wedge q$ or $p \wedge \neg q$ (the conditions from the GDL version of the action for obtaining observation tokens `qT` and `qF`) will additionally be distinguished.

Formally, we define the *product update* of an epistemic model with such an event model as $\langle W, (R_i)_{i\in\mathcal{A}}, V\rangle \otimes \langle E, (Q_i)_{i\in\mathcal{A}}, \mathrm{pre}, \mathrm{eff}\rangle = \langle W', (R'_i)_{i\in\mathcal{A}}, V'\rangle$ where

- $W' = \{(w, e) \in W \times E \mid \mathcal{M}, w \models \mathrm{pre}(e)\}$
- $R'_i = \{((w, e), (w', e')) \in W' \times W' \mid wR_iw', eQ_ie', \text{ and f.a. } (\phi, \psi) \in Q_i(e, e')\text{: } \mathcal{M}, w \models \phi \text{ iff } \mathcal{M}, w' \models \psi\}$
- $V'(p) = \{(w, e) \in W' \mid \mathcal{M}, w \models \mathrm{eff}(e)(p)\}$

One can now verify that taking the product update produces an equivalent model as before (but with e_2 and e_4 replaced by e_1). Instead, we want to show what happens for three much simpler examples. In each example, agent 1 initially does not know the value of q. We're not interested in agent 2. Furthermore, in the first example, p is true in all worlds and in the second example p is false in all worlds. In the third example, agent 1 does not know whether or not p is true.



We can see that the product update works as expected. If p is true in both worlds, the agent will be able to distinguish the successor worlds. This corresponds to the GDL action, where the agent would have received one of the observation tokens `qT` or `qF`. If p is false in both worlds, the agent will not be able to distinguish the successor worlds, which corresponds to the GDL case, where no tokens are received. Finally, if p is true in one and false in the other world, the successor worlds will be distinguishable. This is also analogous to our GDL specification, where the agent is able to distinguish between receiving the token `qT` (or `qF`) and receiving no tokens at all.

Note that our edge-conditioned event models are a non-straightforward generalization of the edge-conditioned event models introduced by Bolander [2014]. We can express one of their edge-conditions ϕ as $\{(\phi, \top)\}$. It is important to note that defining sensing actions like $\mathcal{E}'$ with only a single event is not possible using the edge-conditions of Bolander [2014]. Using the arrow update semantics from Kooi and Renne [2011] this is possible, but only with a number of edge conditions that is worst-case exponential in the number of independent sensing outcomes (i.e., observation tokens). Similarly, conditional effects can only be compiled away with a worst-case exponential blowup in the number of events.

To ensure that the product update is closed epistemically (meaning indistinguishability relations of updated models are always again equivalence relations), additional requirements have to be imposed. Our compilation satisfies the following sufficient conditions on Q_i for all agents $i \in \mathcal{A}$ and events $e, e', e'' \in E$: (1) eQ_ie and if $(\phi, \psi) \in Q_i(e, e)$, then $\phi = \psi$, (2) If $(\phi, \psi) \in Q_i(e, e')$, then $(\psi, \phi) \in Q_i(e', e)$, and (3) If $(\phi, \psi) \in Q_i(e, e'')$ and eQ_ie' or $e'Q_ie''$, then there is a $\chi \in \mathcal{L}_{KC}(P, \mathcal{A})$ such that $(\phi, \chi) \in Q_i(e, e')$ and $(\chi, \psi) \in Q_i(e', e'')$. Condition (1) enforces reflexivity, (2) enforces symmetry and (3) enforces transitivity.

There is a straight-forward compilation from event models with edge conditions into basic event models: Edge conditions can be removed one by one. Specifically, for an edge condition $(\phi, \psi) \in Q_i(e, e')$, we replace e by two events e_ϕ and $e_{\neg\phi}$, and we replace e' by two events e'_ψ and $e'_{\neg\psi}$. While the effects are inherited from e and e', we have $\mathrm{pre}(e_\phi) = \mathrm{pre}(e) \wedge \phi$ and $\mathrm{pre}(e_{\neg\phi}) = \mathrm{pre}(e) \wedge \neg\phi$, as well as $\mathrm{pre}(e'_\psi) = \mathrm{pre}(e') \wedge \psi$ and $\mathrm{pre}(e'_{\neg\psi}) = \mathrm{pre}(e') \wedge \neg\psi$. Agent i can distinguish between the equivalence classes $\{e_\phi, e'_\psi\}$ and $\{e_{\neg\phi}, e'_{\neg\psi}\}$. For all the other agents, all four events are indistinguishable. Also, the indistinguishability between the duplicates of e, e', and events other than e, e' is inherited from the indistinguishability between the original events.

DEL Domain Descriptions

A *DEL domain description* is a tuple $\Delta = \langle \mathcal{M}_0, (A_i)_{i\in\mathcal{A}}\rangle$, consisting of an *initial epistemic model* $\mathcal{M}_0$ and for each agent $i \in \mathcal{A}$ a finite set of epistemic actions A_i, the *action library*. We require each epistemic action $(\mathcal{E}, E_d) \in A_i$ to be *local* to its owner agent i, meaning that the set of designated events is closed under his indistinguishability relation. This is needed to ensure that the agents have *perfect recall* of their own actions (otherwise we could define an action where an agent chooses an event and immediately forgets about his choice). Furthermore, we require an additional closure property for the action library: If $(\mathcal{E}, E_d) \in A_i$ for some $i \in \mathcal{A}$ and e is an event in E, then there exists a set E'_d with $e \in E'_d$ and $(\mathcal{E}, E'_d) \in \bigcup_{j\in\mathcal{A}} A_j$. This ensures that the uncertainty of the agents about an occurred event is reflected by the uncertainty about the actions that can actually be performed.

5 Compilation from GDL-III to DEL

We now define a translation from GDL-III to DEL under the assumptions made in Section 3, i.e., that GDL-III descriptions are sequentialized, grounded, and without derived predicates. We assume that the agents' waiting actions are all named `wait`. Then the set of action names for some agent i can then be extracted from the GDL-III description as $A^i = \{a \mid$ `legal(`i, a`) :- body.` $\in G$ for some arbitrary `body`$\} \setminus \{$`wait`$\}$. Similarly, the set of names for the observation tokens which agent i can receive can be extracted as $T^i = \{t \mid$ `sees(`i, t`) :- body.` $\in G$ for some arbitrary `body`$\}$.

For each action $a \in A^i$ of an agent i, we can then extract the precondition as DEL-formula $\mathit{legal}(i, a)$, as well as the action's effect $\mathit{next}(i, a, p)$ on a given proposition p and the condition $\mathit{sees}(j, t, i, a)$ under which an agent j sees observation token $t \in T^j$ after the action was applied:

$$\mathit{legal}(i, a) = \bigvee_{\texttt{legal(i,a) :- body.} \in G} \sigma(\texttt{body}, i, a)$$
$$\mathit{next}(i, a, p) = \bigvee_{\texttt{next(p) :- body.} \in G} \sigma(\texttt{body}, i, a)$$
$$\mathit{sees}(j, t, i, a) = \bigvee_{\texttt{sees(j,t) :- body.} \in G} \sigma(\texttt{body}, i, a)$$

We use the substitution $\sigma(\texttt{body}, i, a)$ to obtain a proper DEL formula from the body of a relevant GDL clause. It replaces occurrences of $\texttt{true}(f)$ by f and occurrences of $\texttt{knows}(i, p)$ and $\texttt{knows}(p)$ by $K_i\sigma(x)$ and $C\sigma(x)$ where x is obtained from the clause `p :- x.` Moreover, the GDL connectives `not`, `,`, and `;` are replaced by the DEL connectives $\neg$, $\wedge$, and $\vee$. Finally, occurrences of $\texttt{does}(j, b)$ are replaced by $\top$ if $i = j$ and $a = b$, and by $\bot$ otherwise. For simplification, we assume that facts in G are written as rules of the form `p :-` $\top$`.`

In our section about edge-conditioned event models, we have already shown an example of how our compilation of GDL actions works in the case where there is only one action. We now generalize this to arbitrarily many actions. The idea is to construct a single *composite event model* that contains one event $i{:}a$ for all actions $a \in A^i$ of all agents $i \in \mathcal{A}$. We can directly extract the preconditions, effects, and edge conditions using the definitions above.

Definition 1. *Let G be a GDL-III description. Then the* composite event model *of G is $\mathcal{E} = \langle E, (Q_i)_{i\in\mathcal{A}}, \mathrm{pre}, \mathrm{eff}\rangle$, where $E = \{i{:}a \mid i \in \mathcal{A}, a \in A^i\}$ and for each $i, j, k \in \mathcal{A}$ and $a \in A^i$ and $b \in A^j$, we have*

- $\mathrm{pre}(i{:}a) = \mathit{legal}(i, a)$,
- $\mathrm{eff}(i{:}a) = \{p \mapsto \mathit{next}(i, a, p) \mid p \in P\}$, *and*
- $Q_k(i{:}a, j{:}b) =$ *undefined if* $k \in \{i, j\}$ *and* $i{:}a \neq j{:}b$, *or* $\{(\mathit{sees}(k, t, i, a), \mathit{sees}(k, t, j, b)) \mid t \in T^k\}$ *otherwise.*

In the following, analogously to our event names, we also use $i{:}a$ to denote the move in G where agent i performs the action a. Note that for an arbitrary play sequence δ and two such moves $i{:}a$ and $j{:}b$ in G, we have $\delta, i{:}a \sim_k \delta, j{:}b$ iff $i{:}a\, Q_k\, j{:}b$ and all the edge conditions in $Q_k(i{:}a, j{:}b)$ are satisfied in $\mathcal{E}$. Note also that Q_k satisfies our sufficient conditions for epistemic closedness. The reflexivity and symmetry properties follow trivially from the definition of Q_k. For transitivity, we consider the formulas $\phi^a_i = \mathit{sees}(l, t, i, a)$ that are used for the edge conditions between different actions, for a fixed observation token t and agent l. We can see that for three events $e = i{:}a$, $e' = j{:}b$, and $e'' = k{:}c$ with eQ_le'', we have either both or none of eQ_le' and $e'Q_le''$. If we have both, the introduced edge conditions are $(\phi^a_i, \phi^b_j) \in Q_l(e, e')$, $(\phi^b_j, \phi^c_k) \in Q_l(e', e'')$, and $(\phi^a_i, \phi^c_k) \in Q_l(e, e'')$.

We can now define the compilation from GDL-III to DEL domain descriptions. Note that, while non-random agents have one action in their action library for each of their actions as defined in G, the random agent has only a single action.

Definition 2. *Let G be a GDL-III description and $\mathcal{E}$ be its composite event model. Then the* DEL compilation *of G is a DEL domain description with the initial model $\mathcal{M}_0 =$ $\bullet_{w_0 : \{p \mid \texttt{init}(p)\texttt{.} \in G\}}$ and the action libraries $A_i = \{(\mathcal{E}, \{i{:}a\}) \mid a \in A^i\}$ for each agent $i \in \mathcal{A} \setminus \{\mathit{random}\}$ and $A_{\mathit{random}} = \{(\mathcal{E}, \{\mathit{random}{:}a \mid a \in A^{\mathit{random}}\})\}$.*

Both the GDL specification and its DEL translation induce transition systems over epistemic states. We now show that they are isomorphic and that identified states share the same set of satisfied formulas.

Theorem 1. *Let G be a GDL-III description and Δ be its DEL-compilation with initial state $\mathcal{M}_0$. Then a play sequence $\delta = i_1{:}a_1, \ldots, i_n{:}a_n$ is legal in G if and only if the action sequence $(\mathcal{E}, i_1{:}a_1), \ldots, (\mathcal{E}, i_n{:}a_n)$ is successively applicable in $(\mathcal{M}_0, w_0)$. Furthermore, for $\mathcal{M}_n = \mathcal{M}_0 \otimes \mathcal{E}^n$, and $w_n = (w_0, i_1{:}a_1, \ldots, i_n{:}a_n)$, and an arbitrary formula $\phi \in \mathcal{L}_{KC}(P, \mathcal{A})$, we have $\mathcal{M}_n, w_n \models \phi$ iff $G, \delta \models \phi$.*

Proof sketch. We can prove by induction over the length of play/action sequences that there is a one-to-one correspondence between all play sequences δ of length n and worlds (w, δ) in $\mathcal{M}_0 \otimes \mathcal{E}^n$. Both valuations and indistinguishability coincides for worlds and play sequences, meaning in both cases the same formulas are satisfied and the same actions are legal/applicable. □

Example 1. *Consider the following GDL description. Initially, the random agent decides whether or not to set p true. Afterwards, p will never change its value.*

```
role(random). role(ag1). role(ag2).
init(turnrandom).
legal(random, ptrue) :- true(turnrandom).
legal(random, pfalse) :- true(turnrandom).
next(p) :- does(random, ptrue).
next(p) :- true(p).
```

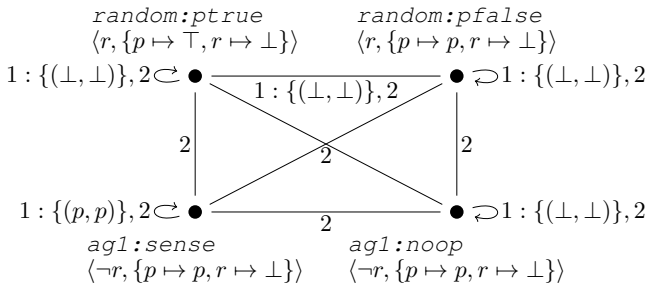
Now, agent `ag1` can decide whether he wants to sense the value of p or do nothing. Note that `noop` is a deliberate action the agent can chose to execute instead of `sense`, rather than a waiting action for someone else's move.

```
legal(ag1, sense) :- not true(turnrandom).
legal(ag1, noop) :- not true(turnrandom).
sees(ag1, ptrue) :- does(ag1,sense), true(p).
```

For our GDL game to be sequentialized, we have to enforce that agents have to wait while it is another agent's turn. In our example, `ag2` is a passive bystander who always waits.

```
legal(random, wait) :- not true(turnrandom).
legal(ag1, wait) :- true(turnrandom).
legal(ag2, wait).
```

Let us now have a look at the composite event model $\mathcal{E}$ of our GDL-III description. We use 1 and 2 for agents `ag1` and `ag2`, as well as r as shorthand for `turnrandom`. The precondition and edge condition formulas are already simplified.



Finally, the DEL-compilation of our GDL description is given as $\Delta = \langle \mathcal{M}_0, (A_1, A_2, A_{\mathit{random}})\rangle$ with $\mathcal{M}_0 = \bullet_{w_0 : r}$, $A_1 = \{(\mathcal{E}, \{\texttt{ag1:sense}\}), (\mathcal{E}, \{\texttt{ag1:noop}\})\}$, $A_2 = \emptyset$, and $A_{\mathit{random}} = \{(\mathcal{E}, \{\texttt{random:ptrue}, \texttt{random:pfalse}\})\}$. Computing the updated models $\mathcal{M}_0 \otimes \mathcal{E}$ and $\mathcal{M}_0 \otimes \mathcal{E}^2$, one can verify the correspondence between worlds and play sequences in terms of valuation and indistinguishability.

6 Compilation from DEL to GDL-III

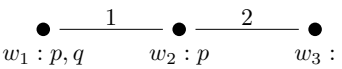
In order to show how DEL can be embedded into GDL-III, we first deal with the difference that agents in DEL planning problems can have incomplete knowledge of the initial state whereas the initial state in a GDL-III game is always common knowledge. To bridge this gap, we use an initial action by the special `random` agent to generate an initial epistemic model.

Definition 3. *For a DEL domain description over $(P, \mathcal{A})$ with initial model $\mathcal{M}_0 = \langle W, (R_i)_{i\in\mathcal{A}}, V\rangle$, the* GDL-III initialization rules *are*

```
role(i).  ∀i ∈ A ∪ {random}
init(ran). init(0).
legal(random,w) :- true(0).  ∀w ∈ W
next(p) :- does(random,w).  ∀p ∈ P, w ∈ V(p)
sees(i,[w]_i) :- does(random,w).  ∀i ∈ A, w ∈ W
```

By this translation, `random` chooses any of the possible initial worlds. Each agent then receives an observation token indicating the equivalence class to which this world belongs according to that agent's indistinguishability relation. The special fluent `ran` is used to denote that it is `random`'s turn, and fluent `0` is true only in the initial state.

Example 2. *Recall the epistemic model from Section 2.1:*



The GDL-III translation is as follows:

```
role(random). role(1). role(2).
init(ran). init(0).
legal(random, w1) :- true(0).
legal(random, w2) :- true(0).
legal(random, w3) :- true(0).
next(p) :- does(random, w1).
next(p) :- does(random, w2).
next(q) :- does(random, w1).
sees(1, w12) :- does(random, w1).
sees(1, w12) :- does(random, w2).
sees(1, w3) :- does(random, w3).
sees(2, w1) :- does(random, w1).
sees(2, w23) :- does(random, w2).
sees(2, w23) :- does(random, w3).
```

It is easy to see that the resulting states from the three possible moves by `random` *satisfy the same propositions as in the epistemic model. Moreover, from their observation tokens, agent 1 cannot distinguish* `does(random, w1)` *from* `does(random, w2)` *while agent 2 cannot distinguish* `does(random, w2)` *from* `does(random, w3)`.

The example illustrates the use of observation tokens to obtain a given epistemic model. The key to a successful embedding of DEL into GDL-III is the proper use of these tokens to encode actions defined by arbitrary event models. We represent the execution of a DEL-action by two steps in GDL-III: First, an agent chooses an applicable action, that is, an event model along with a set of designated events. Thereafter, and similar to the above, `random` selects one of the applicable designated events, and the state changes according to the effects of that event. Each agent will receive an observation token that indicates their own *equivalence class* to which the actual event belongs according to the event model.

In the following, we assume event models with conditional effects but without edge conditions, which can be compiled away as described in Section 2.1. Every DEL-action is encoded in the GDL-III compilation by a unique constant a, which is used both as the name of an action and a fluent that becomes temporarily true when an agent has selected this action. The latter is used to let `random` choose an appropriate event in the subsequent step. For the sake of simplicity, we use DEL-formulas in the body of clauses; to obtain proper GDL, these need to rewritten by substituting every fluent f by `true`(f); every $K_i\phi$ and $C\phi$ by `knows`(i, p) and `knows`(p), respectively, along with a clause p `:-` ϕ; and by encoding propositional formulas in clause bodies by sets of logic program rules in the standard way [Lloyd, 1987].

Definition 4. *The* GDL-III compilation *of an action library $(A_i)_{i\in\mathcal{A}}$ over $(P, \mathcal{A})$ consists of the following rules for every action $a = (\mathcal{E}, E_d) \in A_i$ with $\mathcal{E} = (E, (Q_j), \mathrm{pre}, \mathrm{eff})$:*

```
legal(j,noop).  ∀j ∈ A
legal(i,a) :- pre(e) ∧ ¬true(ran).  ∀e ∈ E_d
legal(random,e) :- true(a) ∧ pre(e).  ∀e ∈ E_d
legal(random,noop) :- ¬true(ran).
next(a) :- does(i,a).
next(ran) :- does(i,a).
next(X) :- true(X) ∧ does(i,a).
next(p) :- does(random,e) ∧ eff(e,p).  ∀p ∈ P, e ∈ E_d
sees(j,[e]_j) :- does(random,e) ∧ eff(e,p). ∀p∈P,e∈E_d
```

In the theorem below that states the correctness of the translation, we use $i_a : a$ to indicate a joint move that consists of action a chosen by its owner and `noop` by all other roles.

Theorem 2. *Let Δ be a DEL domain description with initial model $\mathcal{M}_0$ and G be its GDL-III compilation consisting of the action and initialization rules for Δ. Then $(\mathcal{M}_n, w_n) = (\mathcal{M}_0 \otimes \mathcal{E}_1 \otimes \ldots \otimes \mathcal{E}_n, (w, e_1, \ldots, e_n))$ is a successor state of a sequence of applicable actions $a_1, \ldots, a_n$ in Δ iff $\delta =$ `random`$:w, i_{a_1}{:}a_1,$ `random`$:e_1, \ldots, i_{a_n}{:}a_n,$ `random`$:e_n$ is a legal play sequence in G, where w is a possible world in $\mathcal{M}_0$ and, for all $k \in \{1, \ldots, n\}$, e_k is a designated event in action a_k. Furthermore, $(\mathcal{M}_n, w_n) \models \phi$ iff $G, \delta \models \phi$, for arbitrary formulas $\phi \in \mathcal{L}_{KC}(P, \mathcal{A})$.*

Proof sketch. For the base case $n = 0$, the rules in Definition 3 ensure that the only legal play sequences of length 1 are of the form `random`$:w$. Using the rules in Definition 4 we can then prove inductively the correspondence between the applicability of a DEL-action and its legality in GDL-III as well as the correspondence between the ensuing action by `random` and the effects of the selected designated event. □

7 Conclusion

We have formally investigated the commonalities and differences between two expressive languages for modeling and reasoning about knowledge in dynamic multi-agent settings, GDL-III and DEL. We have demonstrated identical expressiveness of large fragments by providing provably correct compilations between GDL-III and DEL. To this end, and as an additional benefit, we have developed a powerful extension to DEL that allows us to compactly define actions whose effects and observability by other agents can depend on the

current state. The fact that such actions were not definable in DEL before has been one of the criticisms (see the discussion about *action types* vs. *action tokens* in Baral *et al.* [2017]).

Our compilations allow for investigating the translation of theoretical results (e.g., [Bolander *et al.*, 2015; Charrier *et al.*, 2016]) from one language to the other. Also, the application of existing epistemic planning systems (e.g., [Muise *et al.*, 2015; Huang *et al.*, 2017; Le *et al.*, 2018]) to GDL problems is conceivable. Finally, our compilations may pave the way for developing a combined description language for epistemic domains integrating the best features of GDL and DEL.

Acknowledgments

This research was supported under the Australia-Germany Joint Research Cooperation Scheme 2017–18.

References

[Apt *et al.*, 1987] Krzysztof Apt, H. Blair, and A. Walker. Towards a theory of declarative knowledge. In *Foundations of Deductive Databases and Logic Programming*, chapter 2, pages 89–148. 1987.

[Baral *et al.*, 2017] Chitta Baral, Thomas Bolander, Hans van Ditmarsch, and Sheila A. McIlraith. Epistemic planning (Dagstuhl seminar 17231). *Dagstuhl Reports*, 7(6):1–47, 2017.

[Bolander and Andersen, 2011] Thomas Bolander and Mikkel Birkegaard Andersen. Epistemic planning for single and multi-agent systems. *Journal of Applied Non-Classical Logics*, 21(1):9–34, 2011.

[Bolander *et al.*, 2015] Thomas Bolander, Martin Holm Jensen, and François Schwarzentruber. Complexity results in epistemic planning. In *Proceedings of IJCAI*, pages 2791–2797, 2015.

[Bolander, 2014] Thomas Bolander. Seeing is believing: Formalising false-belief tasks in dynamic epistemic logic. In *Proceedings of ECSI*, pages 87–107, 2014.

[Charrier *et al.*, 2016] Tristan Charrier, Bastien Maubert, and François Schwarzentruber. On the impact of modal depth in epistemic planning. In *Proceedings of IJCAI*, pages 1030–1036, 2016.

[Clark, 1978] Keith Clark. Negation as failure. In *Logic and Data Bases*, pages 293–322. 1978.

[Cooper *et al.*, 2016] Martin C. Cooper, Andreas Herzig, Faustine Maffre, Frédéric Maris, and Pierre Régnier. A simple account of multi-agent epistemic planning. In *Proceedings of ECAI*, pages 193–201, 2016.

[Engesser *et al.*, 2017] Thorsten Engesser, Thomas Bolander, Robert Mattmüller, and Bernhard Nebel. Cooperative epistemic multi-agent planning for implicit coordination. In *Proceedings of M4M*, pages 75–90, 2017.

[Gelfond, 2008] Michael Gelfond. Answer sets. In *Handbook of Knowledge Representation*, pages 285–316, 2008.

[Genesereth and Thielscher, 2014] Michael Genesereth and Michael Thielscher. *General Game Playing*. Synthesis Lectures on AI and Machine Learning. 2014.

[Genesereth *et al.*, 2005] Michael Genesereth, Nathaniel Love, and Barney Pell. General game playing: Overview of the AAAI competition. *AI Magazine*, 26(2):62–72, 2005.

[Huang *et al.*, 2017] Xiao Huang, Biqing Fang, Hai Wan, and Yongmei Liu. A general multi-agent epistemic planner based on higher-order belief change. In *Proceedings of IJCAI*, pages 1093–1101, 2017.

[Kominis and Geffner, 2015] Filippos Kominis and Hector Geffner. Beliefs in multiagent planning: From one agent to many. In *Proceedings of ICAPS*, pages 147–155, 2015.

[Kooi and Renne, 2011] Barteld P. Kooi and Bryan Renne. Generalized arrow update logic. In *Proceedings of TARK*, pages 205–211, 2011.

[Le *et al.*, 2018] Tiep Le, Francesco Fabiano, Tran Cao Son, and Enrico Pontelli. EFP and PG-EFP: Epistemic forward search planners in multi-agent domains. In *Proceedings of ICAPS*, 2018.

[Lloyd, 1987] John Lloyd. *Foundations of Logic Programming*. Series Symbolic Computation. 1987.

[Moses, 2015] Yoram Moses. Relating knowledge and coordinated action: The knowledge of preconditions principle. In *Proceedings of TARK*, pages 231–245, 2015.

[Muise *et al.*, 2015] Christian Muise, Vaishak Belle, Paolo Felli, Sheila McIlraith, et al. Planning over multi-agent epistemic states: A classical planning approach. In *Proceedings of AAAI*, pages 3327–3334, 2015.

[Rasmusen, 2007] Eric Rasmusen. *Games and Information: an Introduction to Game Theory*. 4th edition edition, 2007.

[Romero *et al.*, 2014] Javier Romero, Abdallah Saffidine, and Michael Thielscher. Solving the inferential frame problem in the general game description language. In *Proceedings of AAAI*, pages 515–521, 2014.

[Ruan and Thielscher, 2011] Ji Ruan and Michael Thielscher. The epistemic logic behind the game description language. In *Proceedings of AAAI*, pages 840–845, 2011.

[Schiffel and Thielscher, 2009] Stephan Schiffel and Michael Thielscher. Automated theorem proving for general game playing. In *Proceedings of IJCAI*, pages 911–916, 2009.

[Thielscher, 2017] Michael Thielscher. GDL-III: A description language for epistemic general game playing. In *Proceedings of IJCAI*, pages 1276–1282, 2017.

[van Benthem *et al.*, 2006] Johan van Benthem, Jan van Eijck, and Barteld Kooi. Logics of communication and change. *Information and Computation*, 204(11):1620–1662, 2006.

[van Ditmarsch *et al.*, 2007] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*. 2007.