Modelling Simultaneous Games in Concurrent Dynamic Logic

Johan van Benthem[*], Sujata Ghosh[†] and Fenrong Liu[‡]

Abstract: We make a proposal for formalizing simultaneous games at the abstraction level of player's powers, combining ideas from dynamic logic of sequential games and concurrent dynamic logic. We prove completeness for a new system of 'concurrent game logic' $CDGL$ with respect to finite non-determined games. We also show how this system raises new mathematical issues, and throws light on branching quantifiers and independence-friendly evaluation games for first-order logic.

1 Introduction: parallelism in games

Games are very much a part of our daily life. Even our dialogues, arguments, or other interactions can be viewed naturally as games involving goals and strategies. Games also play an important role in economics, logic, linguistics and computer science. In all these settings, the following distinction makes sense. We can talk about a single game, but also about several games played in sequence, or in parallel. While sequential play has been studied a lot recently by logicians, the focus of this paper is simultaneous play, or in other words, parallel games.

In game theory, evergreens like 'Prisoner's Dilemma', or 'Rock, Paper and Scissors' can be thought of as simultaneous single move games for two players. Moreover, playing in this way allows for interesting strategic manoeuvres, such as 'strategy stealing'. In another setting, computer scientists have used games to model concurrent interactive processes allowing for switching between games, where 'copy-cat strategies' (cf. the game semantics of [AJ94] for linear logic) are of the essence. Finally, in linguistics, parallel games have entered in the area of 'branching quantifiers'

$$\begin{array}{lcl} \forall x \exists y & & \\ & \searrow & \\ & & Rxyzu \\ & \nearrow & \\ \forall z \exists u & & \end{array}$$

These have been modelled by means of 'IF games' of imperfect information by [HS97]. E.g., with the preceding quantifier pattern, the challenge player Abelard chooses an assignment for x following which the response player Eloise chooses one for y in one game. In another part of the game without informational access to the first, Abelard chooses for z following which Eloise chooses for u. We may view these games as played simultaneously, and after this phase, we have a 'test' game checking whether $Rxyzu$ holds.

How can we model these games in logic? The situation with sequential games is relatively well-understood in terms of modal logic. [Par85] used analogies with propositional dynamic logic PDL to define a Dynamic Game Logic (DGL) of sequential game constructions, representing players' global powers for determining final outcomes. [Ben01] showed how dynamic logics enhanced with epistemic and preference modalities can also describe the move-by-move local structure of extensive sequential games. But what about parallel games?

In computer science, the challenge of describing concurrency has led to a host of new formal systems, including Process Algebra and Game Semantics. It has proved much harder to extend dynamic logic to a

[*] University of Amsterdam and Stanford University.

[†] Visva-Bharati and University of Amsterdam. Acknowledges a **Rubicon** grant of the NWO (680-50-0504) for her visit to University of Amsterdam in the academic year 2006/07.

[‡] University of Amsterdam and Chinese Academy of Social Sciences. Acknowledges her University of Amsterdam doctoral grant, as well as support from the INIGMA project, NWO.

theory of concurrency, and likewise, so far, no version of Parikh's game logic has been proposed which deals with parallel games. Likewise, no elegant abstract game logic is known for *IF* evaluation games. But there is hope. [Gol92] proves completeness for *Concurrent PDL*, first proposed by [Pel87], whose key program construct is $\alpha \cap \beta$, meaning "α and β are executed in parallel". Against this background, the purpose of our paper is this. We want to stay with the modal abstraction level for describing players' powers in games, for its familiarity and elegance. And to do so, we seek the missing link in the following diagram:

$$\begin{tikzcd}
PDL(sequential\ programs) \arrow[r, dash] \arrow[d, dash] & DGL(sequential\ game\ logic) \arrow[d, dash] \\
CPDL(extension\ to\ concurrency) \arrow[r, squiggly] & CDGL?(concurrent\ game\ logic)
\end{tikzcd}$$

After explaining the less familiar parts of this diagram, we propose a new *concurrent dynamic game logic CDGL*, prove some of its key properties, including completeness – and relates all this to logical evaluation games and game algebra. We finally show how *CDGL* generates further interesting questions.

Before doing all this, here is one further point to be made. Parikh's *DGL* is a logic of determined games, where one of the two players has a winning strategy. However, as we shall see, this mathematically convenient but still extreme simplification no longer works when we have parallel games, so we will allow *non-determined* games throughout.

2 From PDL to DGL

2.1 Powers of players in complex games

Propositional dynamic logic of programs PDL is well-known and we assume that the reader is familiar with it ([HKT00], [BdRV01]). Its ideas have spread to many other areas, including the dynamic logic DGL of sequential games. We start with an informal, though hopefully useful analysis of players' powers in extensive games. Following that, we briefly review the basics of DGL, first proposed in [Par85], further developed by [Pau01], [PP03], [Ben03] and others. DGL works over 'game boards' rather than real games, but to motivate this abstraction step, we will first analyze the latter.

Let us introduce so-called *forcing relations* describing the powers each player has to end an extensive game in a set of final states, starting from a single initial state.

> $s\rho^i_G X$: player i has a strategy for playing game *G* from state s onwards, whose resulting states are always in the set X, whatever the other players choose to do.

As an example, consider the following simple extensive game tree:

$$\begin{tikzcd}
 & & & E \arrow[dll] \arrow[drr] & & & \\
 & A \arrow[dl] \arrow[dr] & & & & A \arrow[dl] \arrow[dr] & \\
1 & & 2 & & 3 & & 4
\end{tikzcd}$$

In this game, player *E* has two strategies, forcing the sets of end states $\{1, 2\}$, $\{3, 4\}$, while player *A* has four strategies, forcing one of the sets of states $\{1, 3\}$, $\{1, 4\}$, $\{2, 3\}$, $\{2, 4\}$.

These forcing relations satisfy the following two conditions ([Ben03]):

(C1) Monotonicity: If $s\rho^i_G X$ and $X \subseteq X'$, then $s\rho^i_G X'$.

(C2) Consistency: If $s\rho^E_G Y$ and $s\rho^A_G Z$, then Y and Z overlap.

Here, Monotonicity is a convenience, making for a smoother mathematical theory - but it also tends to obscure some intuitive features of the structure of players' powers. In this paper, we will assume Monotonicity for the most part, for ease of exposition. But at some strategic places, we also give (in our view) more informative formulations of players' powers that work without it.

Now, consider the following intuitive constructs which form games out of given ones. We do not provide their precise definitions as mathematical operations on trees, which are available in many places, but their meaning should be clear: *choice ($G \cup G'$), dual (G^d), and sequential composition ($G; G'$).* Some game logics also have a construct of *iteration* for repeated play, but we will disregard such infinitary notions in this paper - interesting though they are.

The following observation shows how players' powers have an elegant recursive structure in complex games ([Ben99]). Note that we are not assuming determinacy, which explains our deviation from the more usual presentations of DGL. We need to describe the powers of both players independently, without the winner taking all the center stage in the account. We will use 'E' and 'A' to denote the two players, and 'i' for either player when the statement is the same for both. Again, the statement that follows does not assume that games are determined, and hence we must describe the powers of both players at the same time:

Fact 2.1 *Forcing relations for players in complex games satisfy the following equivalences:*

$$
\begin{array}{lll}
s\rho^E_{G\cup G'} X & \textit{iff} & s\rho^E_G X \textit{ or } s\rho^E_{G'} X \\
s\rho^A_{G\cup G'} X & \textit{iff} & s\rho^A_G X \textit{ and } s\rho^A_{G'} X \\
s\rho^E_{G^d} X & \textit{iff} & s\rho^A_G X \\
s\rho^A_{G^d} X & \textit{iff} & s\rho^E_G X \\
s\rho^i_{G;G'} X & \textit{iff} & \exists Z : \rho^i_G Z \textit{ and for all } z \in Z,\ z\rho^i_{G'} X
\end{array}
$$

Next, we give a reformulation which does not presuppose upward monotonicity of powers. Some readers may find this closer to the actual computation of powers in concrete examples, while it also points the way towards our discussion of concurrent games later on.

Fact 2.2 *Forcing relations for players in complex games when we are not assuming monotonicity as well, are described below:*

$$
\begin{array}{lll}
s\rho^E_{G\cup G'} X & \textit{iff} & s\rho^E_G X \textit{ or } s\rho^E_{G'} X \\
s\rho^A_{G\cup G'} X & \textit{iff} & \exists Z : s\rho^A_G Z,\ \exists Z' : s\rho^A_{G'} Z' \textit{ and } X = Z \cup Z' \\
s\rho^E_{G^d} X & \textit{iff} & s\rho^A_G X \\
s\rho^A_{G^d} X & \textit{iff} & s\rho^E_G X \\
s\rho^i_{G;G'} X & \textit{iff} & \exists Z : s\rho^i_G Z \textit{ and for each } z \in Z,\ \exists X_z : z\rho^i_{G'} X_z \textit{ and } X = \bigcup\{X_z : z \in Z\}
\end{array}
$$

2.2 A dynamic logic of game boards

Now we leave the arena of concrete games G, moving towards a more abstract level of 'generic games', which can be played starting from any state s on game boards. Here are the basic models serving as our 'game boards' which have a set of states plus some 'hard-wired' forcing relations:

Definition 2.3 *A game model is a structure $\mathcal{M} = (S, \{\rho^i_g \mid g \in \Gamma\}, V)$, where S is a set of states, V is a valuation assigning truth values to atomic propositions in states, and for each $g \in \Gamma$, $\rho^i_g \subseteq S \times \mathcal{P}(S)$. We assume that for each g, the relations are upward closed under supersets (the earlier Monotonicity), while also, the earlier Consistency condition holds for the forcing relations of the players A, E.*

Note that the forcing relation ρ, unlike the state-to-state transition relations for programs in PDL, runs from states to *sets of states*.

The language of DGL (used here without game iteration) is defined as follows:

Definition 2.4 *Given a set of atomic games Γ and a set of atomic propositions Φ, game terms γ and formulas ϕ are defined inductively:*

$$
\begin{array}{l}
\gamma := g \mid \phi? \mid \gamma;\gamma \mid \gamma \cup \gamma \mid \gamma^d \\
\phi := \bot \mid p \mid \neg\phi \mid \phi \vee \phi \mid \langle \gamma, i\rangle\phi,
\end{array}
$$

where $p \in \Phi$, $g \in \Gamma$ and $i \in \{A, E\}$.

Note the separate modalities for each player, introducing an explicit, though modest notion of agency.

The truth definition for formulas ϕ in a model $\mathcal{M}$ at a state s is standard, except for the modality $\langle\gamma, i\rangle\phi$, which is interpreted as follows:

$\mathcal{M}, s \models \langle\gamma, i\rangle\phi$ iff there exists $X : s\rho^i_\gamma X$ and $\forall x \in X : \mathcal{M}, x \models \phi$.

Here is the basic result about this logic, for which we provide a proof which is not based on determinacy, combining the suggestions in [Ben99] with the method of [Pau01] for determined games.

Theorem 2.5 *DGL is complete and its validities are axiomatized by*

a) all propositional tautologies and inference rules

b) if $\vdash \phi \to \psi$ *then* $\vdash \langle\gamma, i\rangle\phi \to \langle\gamma, i\rangle\psi$

c) reduction axioms:

$\langle\alpha \cup \beta, E\rangle\phi \leftrightarrow \langle\alpha, E\rangle\phi \vee \langle\beta, E\rangle\phi$
$\langle\alpha \cup \beta, A\rangle\phi \leftrightarrow \langle\alpha, A\rangle\phi \wedge \langle\beta, A\rangle\phi$
$\langle\gamma^d, E\rangle\phi \leftrightarrow \langle\gamma, A\rangle\phi$
$\langle\gamma^d, A\rangle\phi \leftrightarrow \langle\gamma, E\rangle\phi$
$\langle\alpha; \beta, i\rangle\phi \leftrightarrow \langle\alpha, i\rangle\langle\beta, i\rangle\phi$
$\langle\delta?, i\rangle\phi \leftrightarrow (\delta \wedge \phi)$

Proof. Soundness of these axioms should be clear from the earlier definition of the forcing relations. It is totally obvious for the monotonic version, but the reader may want to check that all the above axioms would also hold for our version of players' powers without monotonicity. The case of A's powers in the game starting with a choice for the other player E would be a good example.

For the completeness direction, we perform a canonical model construction. For all $Y \subseteq S$, we define $Q^i_g(Y) = \{s : s\rho^i_g Y\}$. Next a function Q^i_γ for complex formulas is defined recursively as follows:

1. $Q^E_{\alpha\cup\beta}(Y) = Q^E_\alpha(Y) \bigcup E^E_\beta(Y)$
2. $Q^A_{\alpha\cup\beta}(Y) = Q^A_\alpha(Y) \bigcap E^A_\beta(Y)$
3. $Q^A_{\alpha^d}(Y) = Q^E_\alpha(Y)$
4. $Q^E_{\alpha^d}(Y) = Q^A_\alpha(Y)$
5. $Q^i_{\alpha;\beta}(Y) = Q^i_\alpha(Q^i_\beta(Y))$
6. $Q^i_{\alpha?}(Y) = \phi^{\mathcal{M}} \bigcap Y$ where $\phi^{\mathcal{M}} = \{s : \mathcal{M}, s \models \phi\}$.

Now we construct the canonical model $\mathcal{C} = (S^c, \{Q^i_g \mid g \in \Gamma\}, V^c)$, where S^c is the set of all maximally consistent sets of formulae. Let $\hat{\phi} = \{s^c \in S^c : \phi \in s^c\}$. We set:

$s^c \in V^c(p)$	iff	$p \in s^c$
$s^c Q^i_g X$	iff	$\exists\hat{\phi} \subseteq X : \langle g, i\rangle\phi \in s^c$

What we need to prove then is the following lemma:

Lemma 2.6 *For any maximally consistent set* $s^c \in S^c$ *and any formula* ϕ*:* $\mathcal{C}, s^c \models \phi$ *iff* $\phi \in s^c$.

To do so, we prove the following two claims by simultaneous induction on ϕ and γ:

$$(1)\ \phi^{\mathcal{C}} = \hat{\phi}, \text{ and } (2)\ \forall\psi : Q^i_\gamma(\hat{\psi}) = \widehat{\langle\gamma, i\rangle\psi}$$

The base of both claims hold by definition. The boolean inductive steps for (1) are standard. For $\langle\gamma, i\rangle\phi$, suppose $s^c \in (\langle\gamma, i\rangle\phi)^{\mathcal{C}}$ where $\phi^{\mathcal{C}} = \hat{\phi}$ by induction hypothesis. Then the claim follows from (2). Now we prove (2) for complex game terms γ. Here we just consider the following sample cases:

If $s^c Q^i_\alpha Q^i_\beta(\hat{\psi})$, then by the induction hypothesis, $Q^i_\beta = \langle\beta, i\rangle\psi$, and again by the induction hypothesis, $\langle\alpha, i\rangle\langle\beta, i\rangle\psi \in s^c$. The argument is analogous for the converse.

If $s^c Q^A_{\alpha\cup\beta}(\hat{\psi})$, then $Q^A_\alpha(\hat{\psi}) \bigcap Q^A_\beta(\hat{\psi}) \in s^c$ by definition, and hence $\langle\alpha, A\rangle\psi \in s^c$ and $\langle\beta, A\rangle\psi \in s^c$ by the induction hypothesis. Therefore, $\langle\alpha \cup \hat{\beta}, A\rangle\psi \in s^c$ by the relevant reduction axiom. The argument is analogous for the converse.

If $s^c Q^A_{\alpha^d}(\hat{\psi})$, then $s^c Q^E_\alpha(\hat{\psi})$ by definition, and so $\langle\alpha, E\rangle(\hat{\psi}) \in s^c$ by the induction hypothesis. Thus, $\langle\alpha^d, A\rangle(\hat{\psi}) \in s^c$, by the relevant reduction axiom. All further cases are analogous.

The completeness now follows from the Lemma; with an added routine check that forcing relations in our canonical game board satisfy the Monotonicity and Consistency conditions.

■

One quick way of summarizing this completeness proof is as a combination of two insights. First, the reduction axioms for players' powers turn every formula into an equivalent one involving only modalities for atomic game terms. Next, the latter really form just a poly-modal logic interpreted over neighborhood semantics - with just two additional constraints, viz. the monotonicity, and the consistency between paired modalities $\langle\alpha, A\rangle$ and $\langle\alpha, E\rangle$.

The logic DGL presented here is also decidable, since the preceding completeness proof can be carried out wholly in a finite universe of subformulas for the initial formulas to be refuted. But we will not discuss issues of computational complexity in this paper.

2.3 Intermezzo: determinacy and non-determinacy

We have just proved the completeness of game logic for the non-determined case. To simplify things, game logic as in [Par85] and [PP03] works with determined games. While convenient, this suppresses the essential roles of the players. The additional condition can be stated as follows, with S the total set of states:

(C3) Determinacy: If it is not the case that $s\rho^E_G X$, then, $s\rho^A_G S\text{-}X$, and the same for A vis-a-vis E.

In determined games, the powers of one player completely fix those of the other – making these activities somewhat strange as a paradigm for interaction. [Par85] showed that the dual-free logic of determined games with iteration is sound and complete w.r.t the class of all board models. [Pau01] showed that iteration-free logic of determined games with dual is also sound and complete w.r.t the class of all game models. Our completeness proof for the non-determined case includes dual but not iteration - something which we must leave here as an open problem.

But in the present setting we cannot take this easy road! In parallel play, even when the constituent games are determined, non-determinacy arises. Consider the following two games:

$$\begin{tikzcd}[column sep=small] & E \arrow[dl] \arrow[dr] & \\ 1 & \mathbf{G} & 2 \end{tikzcd} \qquad\qquad \begin{tikzcd}[column sep=small] & A \arrow[dl] \arrow[dr] & \\ 3 & \mathbf{H} & 4 \end{tikzcd}$$

The powers of E and A in the game **G** are $\{1\}$, $\{2\}$ and $\{1, 2\}$, respectively and in the game **H**, they are $\{3, 4\}$ and $\{3\}$, $\{4\}$. Let us take a simple first stab at player's powers in a product game, as those which they have in both games when we consider the space of all possible outcomes. Then the powers of E and A in the product $G \times H$, say, the simultaneous play found in a one-step matrix game, become $\{1, 3, 4\}$, $\{2, 3, 4\}$ and $\{1, 2, 3\}$, $\{1, 2, 4\}$, respectively. But then, evidently, neither is $\{2, 3\}$ a power of E, nor $\{1, 4\}$ a power of A, and hence the product game is non-determined in the sense of ([Ben99]).

These observations suggest the need for a modal logic which can deal with non-determined games and a 'product' construct. In the next section, we will discuss a modal process logic which leads the way.

2.4 Game boards and real games

DGL is about generic games played on game boards. But there is another, and richer, tradition: process algebra and game semantics, which gives mathematical formulations for communication and general interaction between agents and systems. It works with real game trees representing players and their moves. The basic question here is how the two traditions are connected. One immediate connection here is that we can represent given game boards as still coming from real game trees, using the following two representation results from [Ben01]:

Proposition 2.7 *Any two families F_1 and F_2 of subsets of some set S satisfying the three earlier conditions (C1), (C2), and (C3) are the powers of players at the root of some two-step game.*

Proposition 2.8 *Any two families F_1 and F_2 of subsets of some set S satisfying just the conditions (C1), (C2) can be realized as the powers in a two-step imperfect information game.*

There are deeper connections to pursue here (cf. Section 6). But we will stay at the board level.

3 From Concurrent PDL to Concurrent DGL

3.1 Concurrent PDL

The system of concurrent dynamic logic due to [Pel87] extends regular dynamic logic by introducing the new operator $\alpha \cap \beta$ of program α and β, interpreted as "α and β executed in parallel". In [Gol92] two modalities $\langle\alpha\rangle$ and $[\alpha]$ are introduced for describing the effects of this, which are no longer interdefinable by $\neg$. [Gol92] proved that concurrent PDL with these two modalities is finitely axiomatizable and decidable. Since the modality of $\langle\alpha\rangle$ is more relevant to us in this context and since we are not considering iteration, we will only review a fragment of $CPDL$ in the following.

The language of iteration-free, necessity-free $CPDL$ is defined as follows:

Definition 3.1 *Given a set of atomic programs Γ and a set of atomic propositions Φ, program expressions γ and formulas ϕ are defined inductively:*

$$\gamma := g \mid \phi? \mid \gamma;\gamma \mid \gamma \cup \gamma \mid \gamma \cap \gamma$$
$$\phi := \bot \mid p \mid \neg\phi \mid \phi \vee \phi \mid \langle\gamma\rangle\phi$$

where $p \in \Phi$ and $g \in \Gamma$.

Definition 3.2 *A CPDL-model is a structure $\mathcal{M} = (S, \{R_g \mid g \in \Gamma\}, V)$ where S is a set of states, V is a valuation assigning truth values to atomic propositions in states, and for each $g \in \Gamma$, $R_g \subseteq S \times$ $\mathcal{P}$(S), interpreted as a relation of simultaneous reachability. The truth of a formula ϕ in $\mathcal{M}$ at a state s is defined in the standard way, except for $\langle\gamma\rangle\phi$, which is as follows:*

$\mathcal{M}, s \models \langle\gamma\rangle\phi$ *iff there exists $T \subseteq S$ with sR_gT and $T \subseteq \phi^{\mathcal{M}}$*

Again, the relation R_g, unlike those for programs in PDL, runs from 'states to sets of states'. But this time, unlike with our earlier forcing relations for sequential games, the interpretation of the sets is 'conjunctive' rather than 'disjunctive': a distributed program can run from one state to a set of states as its output. This point should be kept in mind when checking soundness of the axioms and other assertions in what follows.

We now define the intended meanings of the key program constructs in $CPDL$ in terms of reachability:

Composition: $s(R \cdot Q)T$ iff $\exists U \subseteq S$ with sRU, and a collection $\{T_u : u \in U\}$ of subsets of T with uQT_u for all $u \in U$, such that $T = \bigcup\{T_u : u \in U\}$.

Parallel combination: $R \otimes Q = \{(s, T \cup W) : sRT \text{ and } sQW\}$.

Definition 3.3 *A CPDL-model is standard if it satisfies the following properties:*

$$
\begin{aligned}
R_{\alpha;\beta} &= R_\alpha \cdot R_\beta \\
R_{\alpha\cup\beta} &= R_\alpha \cup R_\beta \\
R_{\alpha\cap\beta} &= R_\alpha \otimes R_\beta \\
R_{\phi?} &= \{(s, \{s\}) : \mathcal{M}, s \models \phi\}
\end{aligned}
$$

For this, and even for the full version of $CPDL$, [Gol92] proved the following result:

Theorem 3.4 *CPDL is complete with respect to standard CPDL-models, and it is decidable.*

There are further interesting aspects to $CPDL$. E.g., [BES94] provide a bisimulation analysis for this language, and show how both the sequential and the parallel operations are 'safe for bisimulation'.

Now our task is to combine the ideas presented here with those for the earlier sequential game logic.

3.2 Forcing relations for product games

To introduce forcing relations for players in product games, we must reconcile the two earlier perspectives. Games can produce complex outcome states now, denoted by sets read 'conjunctively' as in $CPDL$, but players also have choices leading to sets of these read disjunctively as in the semantics of DGL. Here is our proposal for dealing with this - and it is the essential new feature of this paper:

Definition 3.5 *Forcing relations for composite games are defined as follows.*

$$
\begin{array}{rlc}
s\rho^E_{G\cup G'}X & \textit{iff} & s\rho^E_G X \textit{ or } s\rho^E_{G'}X \\
s\rho^A_{G\cup G'}X & \textit{iff} & s\rho^A_G X \textit{ and } s\rho^A_{G'}X \\
s\rho^E_{G^d}X & \textit{iff} & s\rho^A_G X \\
s\rho^A_{G^d}X & \textit{iff} & s\rho^E_G X \\
s\rho^i_{G;G'}X & \textit{iff} & \exists U : s\rho^i_G U \textit{ and for each } u \in \bigcup U,\ u\rho^i_{G'}X \\
s\rho^i_{G\times G'}X & \textit{iff} & \exists T, \exists W : s\rho^i_G T \textit{ and } s\rho^i_{G'}W \textit{ and } X = \{t \cup w : t \in T \textit{ and } w \in W\}
\end{array}
$$

As an illustration, we show how this format fits our earlier intuitive example of parallel games.

$$
\begin{array}{ccccccc}
 & E & & \qquad & & A & \\
\swarrow & & \searrow & & \swarrow & & \searrow \\
1 & \mathbf{G} & 2 & & 3 & \mathbf{H} & 4
\end{array}
$$

The powers of E and A in the game **G** are $\{\{1\}, \{2\}\}$ and $\{\{1, 2\}\}$, respectively and in the game **H**, they are $\{\{3, 4\}\}$ and $\{\{3\}, \{4\}\}$ respectively. The powers of E and A in the product game $\mathbf{G} \times \mathbf{H}$ are then $\{\{1, 3, 4\}\}, \{2, 3, 4\}\}$ and $\{$ $\{1, 2, 3\}\}, \{1, 2, 4\}\}$, respectively. This seems to fit our intuitions.

3.3 Concurrent DGL

3.3.1 Modal language and forcing models

The language of Concurrent DGL is a simple combination of all we had so far:

Definition 3.6 *Given a set of atomic games Γ and a set of atomic propositions Φ, game terms γ and formulas ϕ are defined inductively as follows:*

$$
\begin{array}{l}
\gamma := g \mid \phi? \mid \gamma;\gamma \mid \gamma \cup \gamma \mid \gamma \times \gamma \\
\phi := \bot \mid p \mid \neg\phi \mid \phi \vee \phi \mid \langle \gamma, i \rangle \phi
\end{array}
$$

where $p \in \Phi$, $g \in \Gamma$ and $i \in \{A, E\}$.

The intended meaning of the new game construct $\alpha \times \beta$ is that the games α and β are played in parallel.

Definition 3.7 *A game model is a structure $\mathcal{M} = (S, \{\rho^i_g \mid g \in \Gamma\}, V)$, where S is a set of states, V is a valuation assigning truth values to atomic propositions in states, and with basic relations $\rho^i_g \subseteq S \times \mathcal{P}(\mathcal{P}(S))$ assigned to basic game expressions g.*

For the semantics of our language, we define the truth of a formula ϕ in $\mathcal{M}$ at a state s in the obvious manner, with the following key clause for parallel game product:

- $\mathcal{M}, s \models \langle \alpha, i \rangle \phi$ iff $\exists X : s\rho^i_\alpha X$ and $\forall x \in \bigcup X : \mathcal{M}, x \models \phi$.

Note that this takes together the outcomes of all the separate games. Validities of this logic include all those we had before for DGL, as is easy to see. But in addition, the logic now also encodes facts about parallel games. Here are some of them, pointing the way toward a game algebra of parallel games:

- $\langle \alpha \times \beta, i \rangle \phi \leftrightarrow \langle \beta \times \alpha, i \rangle \phi$
- $\langle (\alpha \times \beta)^d, i \rangle \phi \leftrightarrow \langle \alpha^d \times \beta^d, i \rangle \phi$
- $\langle \alpha \times \beta, i \rangle \phi \rightarrow \langle \alpha \times \beta, i \rangle (\phi \vee \psi)$

A crucial further principle will be found just below.

3.3.2 Axioms and completeness

Here is the essential reduction axiom for game product:

- $\langle \alpha \times \beta, i \rangle \phi \leftrightarrow \langle \alpha, i \rangle \phi \wedge \langle \beta, i \rangle \phi$.

Theorem 3.8 *The axiom $\langle \alpha \times \beta, i \rangle \phi \leftrightarrow \langle \alpha, i \rangle \phi \wedge \langle \beta, i \rangle \phi$ is valid in CDGL.*

Proof. $\mathcal{M}, s \models \langle \alpha \times \beta, i \rangle \phi$
iff $\exists X$: $s\rho^i_{\alpha\times\beta} X$ and $\forall x \in \bigcup X : \mathcal{M}, x \models \phi$
iff $\exists X$: $\exists T$, $\exists W : s\rho^i_\alpha T$ and $s\rho^i_\beta W$ and $X = \{t \cup w : t \in T \text{ and } w \in W\}$ and $\forall x \in \bigcup X, \mathcal{M}, x \models \phi$
iff $\exists X$: ($\exists T$: $s\rho^i_\alpha T$) and ($\exists W$: $s\rho^i_\beta W$) and $X = \{t \cup w : t \in T \text{ and } w \in W\}$ and $\forall x \in \bigcup X, \mathcal{M}, x \models \phi$
iff $\exists T$: $s\rho^i_\alpha T$ and $\forall y \in \bigcup T : \mathcal{M}, y \models \phi$ and $\exists W : s\rho^i_\beta W$ and $\forall z \in \bigcup W : \mathcal{M}, z \models \phi$
iff $\mathcal{M}, s \models \langle \alpha, i \rangle \phi$ and $\mathcal{M}, s \models \langle \beta, i \rangle \phi$
iff $\mathcal{M}, s \models \langle \alpha, i \rangle \phi \wedge \langle \beta, i \rangle \phi$. ∎

Next, using our earlier methods again, we claim that $CDGL$ is complete. We refer to the full paper for details, but the argument is an easy combination of the following two facts.

Fact 3.9 *Due to the reduction axioms, all game formulas can be reduced to equivalent ones involving only atomic modalities $\langle g, i \rangle \phi$.*

Moreover, our generalized semantics still has the original board models for DGL as a special case. Here we use the fact that the map f: $S \rightarrow \mathcal{P}(S)$, given by $f(s) = \{s\}$ can easily be lifted to an injective map from $\mathcal{P}(S)$ to $\mathcal{P}(\mathcal{P}(S))$, and any $X \subseteq S$ can be represented as $f(X) \subseteq \mathcal{P}(S)$.

Fact 3.10 *There is a faithful embedding from DGL models into $CDGL$ models making any DGL-satisfiable formula $CDGL$-satisfiable as well.*

Soundness of all our principles on $CDGL$ models is clear. Next, suppose ϕ is not derivable in $CDGL$. Then because of Fact 3.9, its atomic equivalent ϕ' is not derivable in DGL. Hence, ϕ' has a counter-model in DGL. Then, by Fact 3.10 we get a counter-model in $CDGL$. So we have the following result:

Theorem 3.11 *CDGL is complete. It can be axiomatized by the reduction axioms in DGL together with the new reduction axiom for product.*

Again, decidability can also be shown as before.

4 Connections with logical evaluation games

DGL is a logic of general game structure, but even so, it has strong analogies with specific games used by logicians for evaluation of first-order formulas ϕ in models $\mathcal{M}$. We quickly recall some basics. Verifier V and Falsifier F dispute the truth of a formula ϕ in some model $\mathcal{M}$. The game starts from a given assignment s sending variables to objects in the domain of some given model. Verifier claims that the formula is true in $\mathcal{M}$, Falsifier claims it is false. The rules of this game **eval**$(\phi, \mathcal{M}, s)$ are defined as follows:

- *If ϕ is an atom, V wins if the atom is true, and F wins if it is false.*
- *For formulas $\phi \vee \psi$, V chooses a disjunct to continue with.*
- *For formulas $\phi \wedge \psi$, F chooses a conjunct to continue with.*
- *With negation $\neg\phi$, the two players switch roles.*
- *For existential quantifiers $\exists x\psi$, V chooses an object d in $\mathcal{M}$, and play continues with ϕ and $s[x := d]$.*
- *For universal quantifiers $\forall x\psi$, F chooses some d in $\mathcal{M}$, and play continues with ϕ and $s[x := d]$.*

This analogy has been high-lighted more formally in [Ben03], who shows that evaluation games for first-order logic involve a combination of extensive game trees **eval**$(\phi, \mathcal{M}, s)$ where s is a variable assignment at which the game starts, plus a game board consisting of all variable assignments over the model $\mathcal{M}$. Players' powers make sense as well. E.g., 'Hintikka's Lemma' says that Verifier has a winning strategy iff ϕ is true in $(\mathcal{M}, s)$, while it is false iff Falsifier has a winning strategy.

Many of the central features of DGL as explained earlier already occur in the proof for this simple assertion, and this is no coincidence. The Boolean constants of propositional logic match the game constructs of choice and dual, whereas quantification involves sequential game composition, not per se, but in the way a quantifier $\exists x$ first shifts the current value of the variable x, 'and then' moves on to evaluate the matrix formula following it. Notice that this set-up has the same *generic* character as Parikh's game expressions: a formula can start an evaluation game at any model and assignment, and moreover, the set of available concrete moves is not encoded in the formula: what objects can be assigned depends on the model $\mathcal{M}$.

More precisely, in this way, first-order evaluation games are a special case of DGL, where the atomic games are of two specific sorts: (a) tests for truth and falsity of atomic formulas, and (b) variable-to-value reassignment for quantifiers by themselves. On top of these, one then has the same sequential game constructs as in DGL. This is a non-standard view of first-order logic, as consisting of a decidable game algebra over specific atomic games, where the undecidability of the logic comes from the mathematical structure of the actions (in particular, full assignment spaces), rather than the compositional repertoire by itself. As for the latter, [Ben03] proves that no generality is lost, through a converse representation result.

Theorem 4.1 *There is an effective translation τ from DGL-formulas ϕ to first-order formulas, and from DGL game expressions to first-order formula operators, and there is also a transformation taking any game board $\mathcal{M}$ for DGL to a first-order model $\mathcal{M}^*$ such that the following two equivalences hold:*

(a) $\mathcal{M}, s \models \phi$ iff $\mathcal{M}^*, s \models \tau(\phi)$.

(b)$s\rho_G^{i,\mathcal{M}} X$ iff $s\rho_{\tau(G)}^{i,\mathcal{M}} X$.

The idea is, as in the results in Section 2.4, to replace abstract atomic games g by evaluation games for quantifier combinations $\exists\forall$. Though the main result of the cited paper is about the 'game algebra' of DGL (cf. Section 5 below), this representation also has the following consequence ([Ben03], Section 3.3):

Corollary 4.2 *There is a faithful embedding of DGL into the game logic of first-order evaluation games.*

Thus, logic games (of which there are many more than just evaluation games), though very specialized scenarios, are complete for general game logics of sequential constructions in a precise sense.

Now this also suggests an extension to parallel games. As we observed in our introduction, parallel game structures occur in the extension of first-order logic to 'independence-friendly' IF logic proposed by [HS97] as a procedural analogue of Henkin's 'branching quantifiers'. Indeed, [Ben03] considered this option, and observed how various *basic principles* of IF logic seem to be basic game laws for parallel games. Here is a typical example. Consider again the pattern from the introduction:

$$\begin{array}{lll} \forall x \exists y & & \\ & \searrow & \\ & & Rxyzu \\ & \nearrow & \\ \forall z \exists u & & \end{array}$$

Clearly, this can be described as a game construction of the following sort:

$$(G \times H); K$$

where G, H are the two quantifier prefix games, and K is the subsequent test game. To make this fit our formal framework, we would think of G, H as producing separate assignments (one, s, to x, y, the other, t, to z, u), and the result of $G \times H$ can be viewed as the set $\{s, t\}$, just as in the semantics of $CPDL$.

Another nice example of this way of thinking is another valid IF law. Consider the following formula in *IF* logic: $\forall x \exists y/x Rxy$. Here the slash indicates that Verifier has no access to the value chosen by Falsifier for x. There has been some discussion about correct equivalents for this, with some people claiming that it is just $\exists y \forall x Rxy$. But [Ben06] shows that the correct transformation is to the formula $\exists y \forall x/y Rxy$, which gives players the same powers under any concrete interpretation. This mysterious inversion of quantifiers seems to go against all received wisdom in first-order logic. But viewed in terms of our game algebra, it demystifies simply to the earlier commutativity of parallel product!

$$\begin{array}{lll} \forall x & & \\ & \searrow & \\ & & Rxy \\ & \nearrow & \\ \exists y & & \end{array} \qquad\qquad \begin{array}{lll} \exists y & & \\ & \searrow & \\ & & Rxy \\ & \nearrow & \\ \forall x & & \end{array}$$

Thus, we are really seeking, once more, to fill a corner in a diagram:

$$\begin{tikzcd} DGL \arrow[r, dash] \arrow[d, dash] & FOL \arrow[d, dash] \\ CDGL \arrow[r, rightsquigarrow] & Branching\ quantifier\ game\ logic \end{tikzcd}$$

[Ben03] makes a proposal to this effect, defining powers of players in parallel games through the following stipulation in terms of constituent games:

$s\rho^i_{G\times H} X$ iff $\exists U$: $s\rho^i_G U, \exists V$: $s\rho^i_H V$: $U \times V \subseteq X$.

But we have found it hard to make sense of this proposal, and our system $CDGL$ is a hopefully more sophisticated attempt at supplying the 'missing corner'. Clearly, all valid laws of $CDGL$ can be instantiated as valid principles of branching quantification, or more general IF logic. Thus, we have found a *decidable core logic* inside what is a rather gruesome and complex higher-order system, whose combinatorial nature is still somewhat ill-understood.

Even so, issues remain. For instance:

Question Can we extend Theorem 4.1 to a representation of $CDGL$ models in terms of branching or IF evaluation games of imperfect information?

And there may be genuine objections to the proposal as well. Some people delve into the syntactic jungle of IF-logic, never to see daylight again. Our game expressions do not quite match this. But we see this distance as a virtue. Compare what [Bla72] did for Lorenzen's 'dialogue games', famous for their blend of attractive analysis of validity and obnoxious details. Inside dense thickets of 'procedural conventions', he saw a compositional game structure, which was the dawn of linear logic. Indeed, [Abr06] has taken a similar look at IF logic, and proposed a linear logic-based analysis. While we are not claiming equal status for the above (we have only one game product, whereas linear approaches have a subtler repertoire), we work in the same spirit. Likewise, some people may object to the disappearance of explicit epistemic character of IF logic in our analysis. While we do appreciate epistemic analysis (cf. Section 6 below), we see $CDGL$ as a way of doing some epistemics implicitly, by keeping games 'incommunicado'.

5 Game algebra

5.1 Game algebra for DGL

The forcing relations in models for DGL validate a basic game algebra. Consider a language of game expressions with variables and the operations $\vee$ (choice for E), $-$(dual) and $\diamond$ (sequential composition). We say that the two game expressions G and H are identical if their interpretations in any game board model give the same forcing relations for both players.

The following game identities are valid:

$$
\begin{array}{ccr}
x \vee x \approx x & x \wedge x \approx x & (G1) \\
x \vee y \approx y \vee x & x \wedge y \approx y \wedge x & (G2) \\
x \vee (y \vee z) \approx (x \vee y) \vee z & x \wedge (y \wedge z) \approx (x \wedge y) \wedge z & (G3) \\
x \vee (y \wedge z) \approx x & x \wedge (y \vee z) \approx x & (G4) \\
x \vee (y \wedge z) \approx (x \vee y) \wedge (x \vee z) & x \wedge (y \vee z) \approx (x \wedge y) \vee (x \wedge z) & (G5) \\
- - x \approx x & & (G6) \\
-(x \vee y) \approx -x \wedge -y & -(x \wedge y) \approx -x \vee -y & (G7) \\
(x \diamond y) \diamond z \approx x \diamond (y \diamond z) & & (G8) \\
(x \vee y) \diamond z \approx (x \diamond z) \vee (y \diamond z) & (x \wedge y) \diamond z \approx (x \diamond z) \wedge (y \diamond z) & (G9) \\
-x \diamond -y \approx -(x \diamond y) & & (G10) \\
y \preceq z \rightarrow x \diamond y \preceq x \diamond z & & (G11)
\end{array}
$$

Here $s \preceq t$ is an abbreviation of the equation $s \vee t \approx t$, and $\wedge$ denotes the dual game of $\vee$.

Conditions $(G1)$ - $(G7)$ gives us a De Morgan algebra. By the '$\wedge$' game, we mean the dual of '$\vee$', in the sense that the choice is now for player A, not E. Typically, the condition for right-distribution of composition over choice is not valid here. Consider:

$$G; (H \cup K) = (G; H) \cup (G; K)$$

On the left, player E has a choice for playing the games H or K after playing G, but not so on the right. Here, she has to decide in advance, the composition game she wants to play. The above set of equations was first conjectured by [Ben99], and later on, it has been proved complete by [Gor03] and [Ven03].

5.2 Discussion: game algebra for CDGL

In a similar manner, our new logic $CDGL$ contains a game algebra, this time also for a parallel operation $\times$. It is still unknown what this algebra would look like - though we have assembled quite a few principles. We merely present a brief discussion here, showing some of the issues.

$$
\begin{array}{ccr}
x \times x \approx x & & (G12) \\
x \times y \approx y \times x & & (G13) \\
(x \times y) \times z \approx x \times (y \times z) & & (G14) \\
x \times (y \vee z) \approx (x \times y) \vee (x \times z) & x \times (y \wedge z) \approx (x \times y) \wedge (x \times z) & (G15) \\
-(x \times y) \approx -x \times -y & & (G16) \\
(x \times y) \diamond (u \times v) = (x \diamond u) \times (y \diamond v) & & (G17)
\end{array}
$$

The reader may want to check that (G13), (G14), (G15) and (G16) are valid with the proposed interpretation of $\times$, whereas (G17) is not.

Interestingly, whether (G12) is valid or not much depends on whether we assume the monotonicity condition for the players' powers in the new setting. For example, consider

$$
\begin{tikzcd}
 & E \arrow[dl] \arrow[dr] & \\
1 & \mathbf{G} & 2
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
 & E \arrow[dl] \arrow[dr] & \\
1 & \mathbf{G} & 2
\end{tikzcd}
$$

The powers of E and A in the game **G** are $\{\{1\}, \{2\}\}$ and $\{\{1, 2\}\}$, respectively. The powers of E and A in the product game $\mathbf{G} \times \mathbf{G}$ are $\{\{1\}, \{2\}, \{1,2\}\}$ and $\{\{1, 2\}\}$, respectively. If we assume the monotonicity condition, the power of E in the game **G** can be considered as $\{\{1\}, \{2\}, \{1,2\}\}$, and thus the equation (G12) becomes valid after all in this case.

Clearly, not the last word has been said on the issue of monotonicity here. Indeed, the semantics for $CDGL$ allows for two sorts of monotonicity. One is at the outside level of sets of sets, allowing for arbitrary further sets as members. The other is at the inside level of the sets representing outcome states, using the natural set inclusion there. We will not pursue these options here.

6 Conclusions and intentions

This paper proposes a merge between dynamic logic of games and modal logics of concurrency. The resulting system $CDGL$ has a semantics on generalized game boards allowing for parallel play and compositional analysis of players' powers for determining outcomes. Our main new result says that the set of validities is axiomatizable, and indeed decidable by reduction to a poly-modal base logic over generalized neighborhood models. $CDGL$ also suggests a number of new algebraic and model-theoretic questions, of which we mention axiomatizing the complete game algebra validated by $CDGL$, and indeed, even the complete program algebra validated by $CPDL$, which seems unknown as well. We also think that the model theory of concurrent game bisimulation is worth exploring, including issues of expressive power and bisimulation safety for game constructs. But we have also tried to show that $CDGL$ is natural in other ways, as a means of throwing new light on vexed issues such as the proper interpretation of evaluation games for first-order logic which allow for independence, or imperfect information.

We feel the framework proposed here is still a sort of bare mathematical minimum, and various sorts of structure can, and should, be added to get closer to real interactive games. Here are three particular avenues of investigation which we have in mind.

Process algebra and linear game semantics As we discussed briefly, parallel processes and games are studied at the level of extensive trees in other traditions - even though they, too, seem to focus on players powers for winning the games. Connections between our global modal analysis over game boards and process logics over extensive games remain to be developed in detail. In particular, linear game semantics has a richer repertoire of game operations involving powers of switching between games, and hence powerful strategic manoeuvres like 'Copy cat'. This reflects the *interleaving* nature of its products, whereas our direct products allow no interaction between the subgames at all. Switching and stealing may be essentially beyond our modal framework, but the jury is still out.

Explicit logics of strategies DGL and its ilk quantify over the existence of strategies, but they do not display these strategies as part of their formalism. By contrast, [Ben02] has used dynamic logic to explicitly define strategies of players in given extensive games. In subsequent work, we intend to combine the two perspectives and do a 'double gamification' for propositional dynamic logic, once as a logic of game structure, and once as a logic of strategies within these games. Of course, this would also have to deal with strategies in our parallel game products.

Explicit knowledge Informal discussions of parallel games soon involve knowledge and communication between subgames. Indeed, IF logic was cast initially in terms of games of imperfect information, where players need not know each others' moves. [Ben06] even shows how to add an explicit epistemic language to bring this out. Likewise, we could add knowledge operators to DGL and $CDGL$, giving us means of saying $\langle\gamma, i\rangle K_i\phi$: player i has a strategy in game γ, after playing which she would know ϕ, or converting the operators to $K_i\langle\gamma, i\rangle\phi$: that player i knows that her strategy will have the effect ϕ. This gets even more interesting when knowledge of other players is added. Epistemic versions of our game logics remain to be explored. But they also raise further challenges. It has often been observed in game theory that games with imperfect information no longer allow for simple compositional analysis. Our parallel product gets away with compositionality after all, but it is limited in scope. Should we perhaps 'epistemize' the operations of game algebra to obtain richer calculi of epistemic game combination?

References

[Abr06] S. Abramsky. Socially responsive, environmentally friendly logic. In A. Tuomo and A-V Pietarinen, editors, *Truth and Games: Essays in Honour of Gabriel Sandu*. Acta Philosophica Fennica, Helsinki, 2006.

[AJ94] S. Abramsky and R. Jagadeesan. Games and full completeness for multiplicative linear logic. *Journal of symbolic logic*, 59(2):543–574, 1994.

[BdRV01] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. CUP, 2001.

[Ben99] J. van Benthem. Logic in games. Lecture Notes, Amsterdam and Stanford, 1999.

[Ben01] J. van Benthem. Games in dynamic-epistemic logic. *Bulletin of Economic Research*, 53(4):219–248, 2001.

[Ben02] J. van Benthem. Extensive games as process models. *Journal of Logic, Language and Information*, 11:289–313, 2002.

[Ben03] J. van Benthem. Logic games are complete for game logics. *Studia Logica*, 75:183–203, 2003.

[Ben06] J. van Benthem. The epistemic logic of IF games. In R. E. Auxier and L. E. Hahn, editors, *Philosophy of Jaakko Hintikka*. Open Court, 2006.

[BES94] J. van Benthem, J. van Eijck, and V. Stebletsova. Modal logic, transition systems and processes. *Journal of logic computation*, 4(5):1–50, 1994.

[Bla72] A. Blass. Degrees of indeterminacy of games. *Fund. Math.*, 77:151–166, 1972.

[Gol92] R. Goldblatt. Parallel action: Concurrent dynamic logic with independent modalities. *Studia Logica*, 51(3-4):551–578, 1992.

[Gor03] V. Goranko. Basic algebra of game equivalences. *Studia Logica*, 75:221–238, 2003.

[HKT00] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. The MIT Press, 2000.

[HS97] J. Hintikka and G. Sandu. Game-theoretical semantics. In J. van Benthem & A. ter Meulen, editor, *Handbook of Logic and Language*, pages 361–410. MIT Press, 1997.

[Par85] R. Parikh. The logic of games and its applications. In *Selected papers of the international conference on "foundations of computation theory" on Topics in the theory of computation*, pages 111–139, New York, NY, USA, 1985. Elsevier North-Holland, Inc.

[Pau01] M. Pauly. *Logics for Social Software*. PhD thesis, University of Amsterdam, 2001.

[Pel87] D. Peleg. Concurrent dynamic logic. *J.A.C.M*, 34:450–479, 1987.

[PP03] M. Pauly and R. Parikh. Game logic - an overview. *Studia Logica*, 75:165–182, 2003.

[Ven03] Y. Venema. Representation of game algebras. *Studia Logica*, 75:239–257, 2003.