Synthese (2008) 165:247–268
Knowledge, Rationality & Action 211–232
DOI 10.1007/s11229-008-9390-y

Modelling simultaneous games in dynamic logic

Johan van Benthem · Sujata Ghosh · Fenrong Liu

Received: 15 January 2008 / Accepted: 8 August 2008 / Published online: 26 September 2008

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.

Keywords Simultaneous games · Parallelism · Game logic · Evaluation games · IF logic · Branching quantifiers

1 Introduction: parallelism in games

Games are very much a part of our daily lives. Even our dialogues, arguments, or other interactions can be viewed naturally as games involving goals and strategies. Games

J. van Benthem · F. Liu
Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam,
The Netherlands

J. van Benthem
Department of Philosophy, Stanford University, Standord, CA, USA

S. Ghosh (✉)
Department of Mathematics, Visva-Bharati, Santiniketan, India
e-mail: sujata@science.uva.nl

S. Ghosh
Center for Soft Computing Research, Indian Statistical Institute, Kolkata, India

F. Liu
Department of Philosophy, School of Humanities and Social Sciences, Tsinghua University,
Beijing, China

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 games have been studied a lot recently by logicians, the focus of this paper is simultaneous play in parallel games.

In game theory, typical matrix games like ‘Prisoner’s Dilemma’ involve simultaneous moves for two players: each chooses independently from the other, and the outcome may be viewed as the set of both moves. In another setting, computer scientists use parallel games with simultaneous moves to model concurrent processes. And finally, in linguistics, parallel play has entered the area of ‘branching quantifiers’

$$\begin{tikzcd}[column sep=small]
\forall x\exists y \arrow[dr, dash] & \\
& Rxyzu \\
\forall z\exists u \arrow[ur, dash] &
\end{tikzcd}$$

These have been modelled by means of ‘*IF* games’ of imperfect information by Hintikka and Sandu (1997). 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 structure of sequential games, and players’ responses to observed moves by others is relatively well-understood in terms of modal logic. Parikh (1985) used analogies with Propositional Dynamic Logic (*PDL*) to define a Dynamic Game Logic (*DGL*) of sequential game constructions, representing players’ global powers for determining the final outcomes of the game. Moreover, van Benthem (2001) showed how dynamic logics enhanced with epistemic and preference modalities can describe a richer move-by-move local structure of extensive sequential games, and in particular the interplay between players’s moves and their uncertainties. But what about parallel games? So far, no version of *DGL* has been proposed which deals with parallel games. Likewise, no elegant game logic is known for the above *IF* evaluation games.

Finding such a logic is not one question, but a whole family, depending on how we cast parallel play. In computer science, the challenge of concurrency has led to such systems as Process Algebra and Game Semantics. These systems vary in what *communication* they allow between the different components of a parallel process. Baeten and Verhoef (1995) gives a nice exposition of different types of parallel operators, viz. free merge, left merge as well as communication merge. Modelling ‘open systems’ that interact with their environment gave rise to the concept of ‘concurrent game structures’ Alur et al. (2002), both synchronous and asynchronous (interleaving).

Against this background, the present paper has a modest aim. We will model simultaneous play of parallel games in terms of players’ abstract powers, without allowing for communication. Using ideas from dynamic logic for concurrency, we will propose a system where players' powers in a parallel game can be *reduced to* their powers in the constituent games. This is by no means the only perspective that makes sense, and we will discuss some alternatives in our Sect. 4 on game bisimulation, game algebra, and richer modal languages.

Our starting point is this. Goldblatt (1992) proves completeness for *concurrent PDL* over abstract state models, proposed by Peleg (1987), whose key program construct is $\alpha \cap \beta$, meaning "α and β are executed in parallel". We will work at this same modal abstraction level for describing players' powers in games, for its familiarity and elegance. And thus, we seek the missing link in the following diagram:

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

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 relate all this to logical evaluation games and game algebra. We finally show how *CDGL* suggests further interesting questions, as well as generalizations to richer modal logics of parallel games.

Before doing all this, here are two more points. One is that we say 'games' where a game theorist would say 'game forms', since we disregard *preferences* or *utilities*. The latter structure is important, but beyond the scope of this paper. This is in line with common usage in logic and computer science, though it may sound strange at first to readers from other communities. The other, more crucial point is this. Parikh's *DGL* is a logic of *determined games*, where one of the two players has a winning strategy. But as we shall see, this mathematically convenient simplification no longer works with parallel games, so we will allow *non-determined* games throughout, whose sequential logic will be denoted by $DGL^{\bullet}$ in later sections.

2 From PDL to DGL

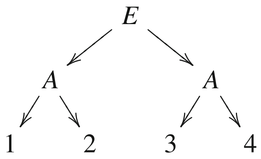
2.1 Powers of players in complex games

Propositional dynamic logic is well-known and we assume that the reader is familiar with it (Harel et al. 2000; Blackburn et al. 2001). Its ideas have spread to many other areas, including the dynamic logic of sequential games. We start with an informal analysis of the players' powers in extensive games. Following that, we briefly review the basics of *DGL*, first proposed in Parikh (1985), and further developed by Pauly (2001), Pauly and Parikh (2003), van Benthem (2003) and several others. *DGL* works over 'game boards' rather than real games, but to motivate this abstraction, we will first analyze the latter.

Consider *forcing relations* describing each player's powers to end a game in a set of final states, starting from an initial state:

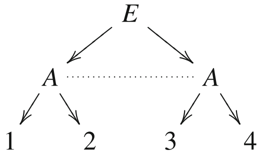
$s\rho^i_G X$: player i has a *strategy* for playing game G from state s onwards, whose final states are always in the set X, whatever the other players choose to do.

As an example, consider the following simple extensive game tree describing a game of perfect information for two players:



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

This is a determined game. Non-determinacy arises, for instance, when we allow imperfect information about other players' moves. In the following imperfect information game, the dotted line indicates that player A cannot observe E's move:



Players' powers are now given by *uniform strategies* which prescribe the same move at epistemically indistinguishable nodes. Thus, E's powers are the same as above, whereas A can now just force the game to end in $\{1, 3\}$ ('always play left') and $\{2, 4\}$ ('always play right').

Similar non-determined phenomena occur in strategic matrix games. In Prisoners' Dilemma, players simultaneously choose one of {co-operate}, {defect}, and the power for any choice consists of two outcomes, sharing the chosen move, but differing in what the other does.

While the examples discussed above are of finite games, 'forcing relations' apply equally well to infinite games as well, viewing their complete histories as the outcomes. In particular, even sequential games of perfect information can be non-determined with infinite play. The notions and results in our paper apply to all these settings, illustrating the power of abstraction.

Forcing relations satisfy two simple conditions van Benthem (2003):

(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 makes for smoother theory—though it may obscure some intuitive features of players' powers. In this paper, we will assume Monotonicity for the most part, for ease of exposition. But at some strategic places in our exposition, we also give more informative formulations of players' powers that work without it. For a deeper logical study of non-monotonic versions of powers in a general neighbourhood setting, we refer to Hansen et al. (2007).

For *determined games*, a much stronger third condition holds, which says that, if one player lacks a set power X, the other player has the complement power $-X$. This property typically failed in our earlier games with imperfect information, and with simultaneous moves. We will discuss this issue once again later.

Next, 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 game constructions in this paper—interesting though they are.

The following observation shows how players' powers have an elegant recursive structure in complex games (van Benthem 1999). 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.

Fact 2.1 *Forcing relations for players in complex sequential two-person games satisfy the following equivalences:*

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

Some of these equivalences can be easily simplified using Monotonicity. We gave this version as it seems closer to computation of powers in concrete games, both sequential, and later on, also parallel.

2.2 Test games

In what follows, we also need a stipulation of powers for the auxiliary 'test games', which have been a topic of some discussion in the literature. The standard *DGL* stipulation is given by:

$$\begin{array}{rl}
s\rho^E_{\delta?}X & \text{iff } s \in [[\delta]]\ \ \textbf{and}\ \ s \in X. \\
s\rho^A_{\delta?}X & \text{iff } s \notin [[\delta]]\ \textbf{or}\ \ s \in X.
\end{array}$$

This makes test games determined, but the roles of the two players look asymmetric. In our view, the following more symmetric stipulation is preferable, which we are going to adopt for *DGL*$^\bullet$:

$$s\rho^{A}_{\delta?}X \quad \text{iff } s \notin [[\delta]] \ \textbf{and} \ s \in X.$$

This makes test games one more example of indeterminacy: A does not get any powers if $s \in [[\delta]]$, but this does not mean that E gets every power. Also, on both stipulations, players never get the empty set as a power: Non-Triviality holds.

One reason why taking a stand on test games is confusing are the three notions involved: a test game $\delta?$, its dual game $(\delta?)^d$, and the test game $(\neg\delta)?$. We will not attempt to sort all this out here, as it seems orthogonal to the main points that we want to make in what follows.

2.3 A dynamic logic of game boards

Now we leave the arena of concrete games G, moving towards 'generic games' that can be played starting from any state s on 'game boards'. In this generic calculus, atomic games can be arbitrary: finite, infinite, with perfect or imperfect information. The basic 'game boards' are models with a set of states plus some 'hard-wired' forcing relations, defined in the following:

Definition 2.2 A game model $\mathcal{M} = (S, \{\rho^i_g \mid g \in \Gamma\}, V)$ has a set S of states, a valuation V assigning truth values to atomic propositions in states, and for each $g \in \Gamma$, $\rho^i_g \subseteq$ S $\times \mathcal{P}(S)$. For each g, the relations satisfy *Monotonicity* and *Consistency*, and we also require *Non-Triviality*: no player can force the empty set.

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 $DGL^\bullet$ (without game iteration) is defined as follows:

Definition 2.3 Given a set of atomic games Γ and a set of atomic propositions Φ, game terms γ and formulas ϕ are defined inductively:

$$\gamma := g \mid \phi? \mid \gamma;\gamma \mid \gamma \cup \gamma \mid \gamma^d$$
$$\phi := \bot \mid p \mid \neg\phi \mid \phi \vee \phi \mid \langle \gamma, i\rangle\phi,$$

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

The player modalities introduce an explicit notion of agency.

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

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

In this semantics, we assume the earlier recursive definitions of forcing relations for complex game terms. One can check by induction that these preserve Monotonicity, Consistency, and Non-Triviality.

Here is the basic result for this logic. Our proof follows the method of Pauly (2001) for determined games, using the suggestions in van Benthem (1999) to circumvent the assumption of determinacy.

Theorem 2.4 *DGL$^\bullet$ is complete w.r.t. game models and its validities are axiomatized by the following axioms:*

(*a*) *all propositional tautologies and inference rules*
(*b*) *if* $\vdash \phi \to \psi$ *then* $\vdash \langle g, i\rangle\phi \to \langle g, i\rangle\psi$
(*c*) $\langle g, E\rangle\phi \to \neg\langle g, A\rangle\neg\phi$
(*d*) $\neg\langle \gamma, i\rangle\bot$
(*e*) *reduction axioms:*

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

Proof Soundness of these axioms should be clear from the definition of forcing relations for the complex games given earlier. We stated the consistency axiom just for one player, as the other one follows.

For the completeness, we use a canonical model construction. The model is $\mathcal{C} = (S^c, \{Q^i_g \mid g \in \Gamma\}, V^c)$, with S^c the set of all maximally consistent sets of formulae. Let $\widehat{\phi} = \{s^c \in S^c : \phi \in s^c\}$. We set:

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

For all $Y \subseteq S^c$, we define $Q^i_g(Y) = \{s^c : s^c Q^i_g Y\}$. Next a function Q^i_γ for complex formulas is defined recursively as follows:

1. $Q^E_{\alpha\cup\beta}(Y) = Q^E_\alpha(Y) \bigcup Q^E_\beta(Y)$
2. $Q^A_{\alpha\cup\beta}(Y) = Q^A_\alpha(Y) \bigcap Q^A_\beta(Y)$
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. $Q^E_{\delta?}(Y) = \delta^{\mathcal{C}} \bigcap Y$ where $\delta^{\mathcal{C}} = \{s^c : \mathcal{C}, s^c \models \delta\}$
7. $Q^A_{\delta?}(Y) = \neg\delta^{\mathcal{C}} \bigcap Y$ where $\neg\delta^{\mathcal{C}} = \{s^c : \mathcal{C}, s^c \not\models \delta\}$

What we need to prove then is the following lemma:

Lemma 2.5 *For any maximally consistent set $s^c \in S^c$ and any formula ϕ, the following holds: $\mathcal{C}, s^c \models \phi$ iff $\phi \in s^c$.*

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

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

The base of both claims hold by definition. The boolean inductive steps for (1) are standard. For the game modality ($\langle \gamma, i\rangle\phi$), (1) can be proved easily using (2). We now prove (2) for complex game terms γ. Here are the typical cases:

Suppose $s^c Q^i_{\alpha;\beta}\widehat{\psi}$. Then, by definition $s^c Q^i_\alpha Q^i_\beta \widehat{\psi}$. Now by the induction hypothesis, $Q^i_\beta(\widehat{\psi}) = \widehat{\langle\beta, i\rangle\psi}$, and so, $Q^i_\alpha(Q^i_\beta(\widehat{\psi})) = Q^i_\alpha(\widehat{\langle\beta, i\rangle\psi})$. Thus, $s^c \in Q^i_\alpha(\widehat{\langle\beta, i\rangle\psi}) = \widehat{\langle\alpha, i\rangle\langle\beta, i\rangle\psi}$, which implies that $\langle\alpha, i\rangle\langle\beta, i\rangle\psi \in s^c$. Then, by the relevant reduction axiom, $\langle\alpha; \beta, i\rangle\psi \in s^c$. The argument is analogous for the converse.

Suppose $s^c Q^A_{\alpha\cup\beta}\widehat{\psi}$. Then $s^c \in Q^A_\alpha(\widehat{\psi}) \bigcap Q^A_\beta(\widehat{\psi})$ 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, $\langle\alpha\cup\beta, A\rangle\psi \in s^c$ by the relevant reduction axiom. The argument is analogous for the converse.

Suppose $s^c Q^A_{\alpha^d}\widehat{\psi}$. Then $s^c Q^E_\alpha \widehat{\psi}$ by definition, and so we have $\langle\alpha, E\rangle\psi \in s^c$ by the induction hypothesis. Thus, $\langle\alpha^d, A\rangle\psi \in s^c$ by the relevant reduction axiom. All further cases are analogous.

The completeness result now follows from the Lemma; with an added routine check that the forcing relations on our canonical game board satisfy the conditions of Monotonicity, Consistency, and Non-Triviality. □

One 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 three additional constraints, viz. the monotonicity, the non-triviality, and the consistency between paired modalities $\langle\alpha, A\rangle$ and $\langle\alpha, E\rangle$.

It is to be noted that the logic $DGL^\bullet$ 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.

2.4 Intermezzo: determinacy and non-determinacy

We have proved the completeness of game logic for the non-determined case. To simplify things, game logic as given in Parikh (1985) and Pauly and Parikh (2003) works with determined games. While convenient, this also suppresses the essentially intertwined roles of the players. The condition can be stated as follows, with S denoting the total set of states:

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

We summarize once more why non-determinacy is natural: cf. also van Benthem (1999). Consider our examples in Sect. 2.1. In the imperfect information game given there, the set $\{2, 3\}$ is not a power for player E, but neither is its complement $\{1, 4\}$ a power for A. Essentially the same held with simultaneous play in matrix games, and we will elaborate that example in Sect. 3.2, using an appropriate account of compound outcome states. Finally, as we pointed out already, our test games, too, violate condition $(C3)$ for understandable reasons.

In determined games, the powers of one player completely fix those of the other. Parikh (1985) showed that the dual-free logic of determined games with iteration is sound and complete with respect to the class of all board models. Pauly (2001) showed that iteration-free logic of determined games with dual is sound and complete on the class of all game models. Our completeness proof for the non-determined case includes dual but not iteration. Since the first version of this paper appeared in van Benthem et al. (2007), van Eijck and Verbrugge (2008) have used our framework to analyze the full sequential language with iteration, announcing a completeness proof for $DGL^{\bullet}$ with iteration.

Against this background, our general setting including parallel play suggests the need for a richer logic which can deal with non-determined games as well as a non-sequential 'product' construct for games. In the next section, we will discuss a modal process logic which leads the way.

2.5 Game boards and real games

$DGL^{\bullet}$ and DGL are about generic games played on game boards. But as stated in Sect. 1, there is another, and richer, tradition: process algebra and game semantics give mathematical models for communication and general interaction between agents and systems Bergstra et al. (2001) with real game trees for players and their moves. We only note one simple connection here. One can represent game boards as coming from real game trees, either determined or not, using the following two representation results from van Benthem (2001):

Proposition 2.6 *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 extensive game.*

Proposition 2.7 *Any two families F_1 and F_2 of subsets of some set S satisfying just the conditions $(C1)$, $(C2)$ can be realized as players' powers in a two-step imperfect information game.*

For further discussion, we refer to Sect. 6. As for the present, however, 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 Peleg (1987) extends regular dynamic logic by introducing a new program operator $\alpha \times \beta$, interpreted as "α and β executed in parallel". In Goldblatt (1992) two modalities $\langle\alpha\rangle$ and $[\alpha]$ are introduced for describing the effects of this, which are no longer interdefinable by $\neg$. Concurrent *PDL* with these two modalities is finitely axiomatizable and decidable. Given our interests, we will only review a fragment of *CPDL* in the following.

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

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

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

where we take $p \in \Phi$ 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 at states, and for each $g \in \Gamma$, $R_g \subseteq S \times \mathcal{P}(S)$ is a relation of simultaneous reachability. Truth of a formula ϕ in $\mathcal{M}$ at state s is defined as usual, except for $\langle\gamma\rangle\phi$, which reads:

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

Again, the relation R_g, unlike those for programs in *PDL*, runs from 'states to sets of states'. But this time, unlike with our forcing relations for sequential games, the view of the sets is 'conjunctive' rather than 'disjunctive': a distributed program runs from one state to a set of states as its 'collective output'. This point should be kept clearly in mind when checking our 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 family $\{T_u : u \in U\}$ 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

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

For this formalism, and even for the full language of *CPDL*, the following result has been proved Goldblatt (1992):

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

There are also interesting model-theoretic aspects to *CPDL*. For example, van Benthem et al. (1994) give a bisimulation analysis, and show how both sequential and parallel operations are 'safe for bisimulation'. Later on, we will state similar results for our game operations.

Now our task is to combine the model for concurrency presented here with that for the earlier sequential game logic.

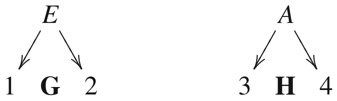
3.2 Forcing relations for product games

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 such sets, still read disjunctively at this second level as we did with *DGL*. Here is our proposal—and it is the essential new feature of this paper. X, U, T, W range over sets of sets of states, t, w range over sets of states, and s, u range over states:

Definition 3.5 Forcing relations for composite games are these:

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

As an illustration, we show in the following how this format for computation of players' powers fits an intuitive example of parallel games, for instance, simultaneous move selection in a matrix game:



To make things comparable, we now change earlier single outcomes s to singleton states s. The powers of E in the game **G** are given by $\{\{1\}\},\{\{2\}\}$ and that of A by $\{\{1\}, \{2\}\}$. Similarly, in the game **H**, the powers of E and A are $\{\{3\}, \{4\}\}$ and $\{\{3\}\},\{\{4\}\}$, respectively. The powers of E and A in the product game **G** × **H** are then formed by taking unions: $\{\{1, 3\}, \{1, 4\}\},\{\{2, 3\}, \{2, 4\}\}$ and $\{\{1, 3\}, \{2, 3\}\}$, $\{\{1, 4\}, \{2, 4\}\}$, respectively. Reading the inner brackets as conjunctive, and the outer ones as disjunctive, this seems to fit our intuitions.

3.3 Concurrent DGL

3.3.1 Modal language and forcing models

The language of our new game logic of Concurrent DGL is a simple combination of all ingredients we had so far:

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

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

where we take $p \in \Phi$, $g \in \Gamma$ 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, without communication.

Definition 3.7 A conjunctive 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 $\rho^i_g \subseteq \mathrm{S} \times \mathcal{P}(\mathcal{P}(S))$ assigned to basic game expressions g, satisfying the conditions of Monotonicity, Consistency and Non-Triviality.

Here, we assume the same structural conditions on forcing relations as before. In 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 squashes together the outcomes of all separate games, making only local assertions at single states. An alternative option would be to evaluate formulas at 'collective states', being sets of the original states. We will return to this in Sect. 4.2.

Validities of this logic include all those we had before for $DGL^\bullet$. But the logic now also encodes facts about parallel games. Here are two, pointing toward an algebra of parallel games lying encoded here:

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

A crucial further principle will be found just below.

3.3.2 Axioms and completeness

Here is the reduction axiom for game product ('Product Axiom'):

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

Theorem 3.8 *The Product Axiom 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 $X = \{t \cup w : t \in T$ and $w \in W\}$ and $\forall x \in \bigcup X : \mathcal{M}, x \models \phi$
iff (by some simple calculation; crucially using Non-Triviality forbidding the forcing of an empty set of outcome states) $\exists T : s\rho^i_\alpha T$ 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. The argument is a combination of the following two facts.

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

Also, our generalized semantics still has the original board models for $DGL^\bullet$ as a special case. For, the map $f : S \to \mathcal{P}(S)$ with $f(s) = \{s\}$ is easily lifted to an injective map from $\mathcal{P}(S)$ to $\mathcal{P}(\mathcal{P}(S))$, and so, any $X \subseteq S$ can be represented as $f(X) \subseteq \mathcal{P}(S)$.

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

Soundness of our axioms on *CDGL* models is easy to check. Next, suppose ϕ is not derivable in *CDGL*. By Fact 3.9, its atomic equivalent ϕ' is not derivable in *DGL*$^\bullet$. So, ϕ' has a counter-model in *DGL*$^\bullet$. Then, by Fact 3.10 we get a counter-model in *CDGL*. This proves

Theorem 3.11 *CDGL is completely axiomatized by the reduction axioms in DGL$^\bullet$ plus the new reduction axiom for product.*

Again, decidability can be shown as before, by a simple adjustment.

While this result may seem a victory for *PDL*-style ‘reductionism’, our product axiom also reflect the expressive poverty of *CDGL* (and indeed, also the program logic *CPDL*) as an account of parallelism. We have no means of stating truly collective properties of conjunctive states. We return to this point in Sect. 4.

3.4 Monotonicity: an excursion

Our forcing relations in parallel games still satisfy the original monotonicity and consistency conditions, as may be proved by induction:

($C1'$) 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.

But really, in *CDGL*, *two* set-theoretic levels are relevant to players’ powers: the ‘conjunctive’ output sets inside, and the ‘disjunctive’ sets of these at the outside. Accordingly, there is ‘outer monotonicity’ adding new sets, and ‘inner monotonicity’ enlarging sets inside a power. The latter has no general appeal for us, since it would represent stronger outcomes of a process or game. If anything, our weak language stating universal properties of individual states inside conjunctive sets might suggest ‘downward monotonicity’ at the inside level.

4 Game models and game algebra

4.1 Game models

One of the more basic questions of model theory is: “When are two models equivalent?” The notion of modal bisimulation can be adapted to express equivalences between game models: detailed discussion can be found in van Benthem et al. (1994) regarding *CPDL*, and van Benthem (2002) for general games. For *CDGL*, the relevant concept can be formulated as follows.

Definition 4.1 Let $\mathcal{M} = (S, \{\rho^i_g \mid g \in \Gamma\}, V)$ and $\mathcal{M}' = (S', \{\rho'^i_g \mid g \in \Gamma\}, V')$ be two conjunctive game models. A relation $\underline{\leftrightarrow} \subseteq S \times S'$ is a *CDGL-bisimulation* between $\mathcal{M}$ and $\mathcal{M}'$, if for any $s \,\underline{\leftrightarrow}\, s'$, we have

1. $s \in V(p)$ iff $s' \in V'(p)$, for all $p \in \Phi(\textit{the atomic propositions})$,
2. For all $X \subseteq \mathcal{P}(S)$, and $g \in \Gamma(\textit{the atomic games})$, if $s\rho^i_g X$, then $\exists X' \subseteq \mathcal{P}(S')$ with $s'\rho_g X'$ and $\forall x' \in \cup X', \exists x \in \cup X : x \,\underline{\leftrightarrow}\, x'$.
3. For all $X' \subseteq \mathcal{P}(S')$, and $g \in \Gamma$, if $s'\rho'^i_g X'$, then $\exists X \subseteq \mathcal{P}(S)$, such that $s\rho^i_g X$, and $\forall x \in \cup X, \exists x' \in \cup X' : x \,\underline{\leftrightarrow}\, x'$.

Analogously to the results in Pauly (2001) for *DGL*, the following theorem shows that *CDGL* formulas are invariant for bisimulation, and the *CDGL* game constructions are safe for bisimulation:

Theorem 4.2 *Let $\mathcal{M} = (S, \{\rho^i_g \mid g \in \Gamma\}, V)$ and $\mathcal{M}' = (S', \{\rho'^i_g \mid g \in \Gamma\}, V')$ be two conjunctive game models, while $s \leftrightarrows s'$ with $s \in S$ and $s' \in S'$. Then*

1. *For all formulae ϕ: $\mathcal{M}, s \models \phi$ iff $\mathcal{M}', s' \models \phi$.*
2. *For all games γ: if $s\rho^i_\gamma X$, then $\exists X' \subseteq \mathcal{P}(S')$ with $s'\rho'^i_\gamma X'$, and $\forall x' \in \cup X', \exists x \in \cup X : x \leftrightarrows x'$.*
3. *For all games γ: if $s'\rho'^i_\gamma X'$, then $\exists X \subseteq \mathcal{P}(S)$ with $s\rho^i_\gamma X$, and $\forall x \in \cup X, \exists x' \in \cup X' : x \leftrightarrows x'$.*

The proof is a simultaneous induction on the complexity of formulas and games. The only interesting case is checking the safety of the 'product' construction, where once more, we need the structural property of Non-Triviality.

Our notion of bisimulation is very 'rough', squashing the structure of sets of sets in the forcing relations. This again reflects the poverty of our modal language. Let us now try to remedy this.

4.2 Intermezzo: a richer 'collective' modal language

Can we create a richer modal language, a finer notion of bisimulation for parallel games and truly collective action? While details of this are beyond the aims of this paper, we state one possible track, which also throws some light on our actual system.

Let us reinterpret our modal language over sets of states as the basic entities, including the original states as singletons. This requires a valuation assigning sets of sets to proposition letters, but otherwise, we can use the same models as before:

Definition 4.3 A collective model is a structure $\mathcal{M} = (S, \{\rho^i_g \mid g \in \Gamma\}, V)$, with S a set of states, V a valuation assigning truth values to atomic propositions in sets of states, with the basic relations $\rho^i_g \subseteq \mathcal{P}(S) \times \mathcal{P}(\mathcal{P}(S))$ assigned to basic game expressions g as before, satisfying again Consistency, Monotonicity, and Non-Triviality.

The sets in $\mathcal{P}(S)$ are collective states, and the language can now express properties of these, rather than just individual states. In particular, while some properties of collectives may be reducible to those of their constituent points (or, atomic singleton states), others need not. This 'set lift' by itself would also make sense for *CPDL*, to make it a more realistic account of collective action, and we think it might even simplify the semantics of sequential program composition. The set lift also fits with the semantics of collective predicates in natural language.

We can now interpret our modal language in collective models $\mathcal{M}$ over sets of states X as the basic indices. The truth definition for formulas ϕ is standard with the modality $\langle\gamma, i\rangle\phi$ lifted as follows:

- $\mathcal{M}, X \models \langle\gamma, i\rangle\phi$ iff $\exists Y\colon X\rho^i_\gamma Y$ and for all $Z \in Y, \mathcal{M}, Z \models \phi$.

This pretty much recreates *CDGL* as a logic for sets. But the interesting feature is now that we can also access other structure, in particular the inclusion structure among base sets. For this purpose, borrowing an idea from temporal logic, one can introduce modal operators $[\subseteq]\phi$ for downward inclusion which say that

- $\mathcal{M}, X \models [\subseteq]\phi$ iff ϕ holds at each subset of the current set X.

'Distributive properties' like the ones in *CDGL* hold if and only if each atomic subset has them. These imply their downward modalized versions, but other properties may, too. Other well-known modal operators that make sense here are an *upward inclusion modality*, as well as a binary *sum modality* $\langle+\rangle\phi\psi$ saying that the current set is a union of one satisfying and one satisfying ψ. This seems close to what we want to say about the output of parallel games as defined above.

We can translate our earlier logics into this extended formalism by forcing every proposition letter p to become distributive. This can be done using a suitable modal prefix like $[\subseteq]\langle\subseteq\rangle p$, which says that every subset includes a subset where p holds.

It would be of interest to axiomatize richer modal languages of joint action like this, and develop the bisimulation-based model theory of parallel play together with the collective state structure which it produces. We do not expect obvious reduction principles like the conjunction axiom of *CDGL*, since the properties of unions of conjunctive states from the separate games may have only a tenuous relationship to those of their components.

4.3 Game algebra for *DGL*$^\bullet$ (*DGL*) and *CDGL*

Now we take another mathematical perspective, following van Benthem (2003). The forcing relations in models for logic *DGL*$^\bullet$ (*DGL*) validate a basic *game algebra*. Consider a language of game expressions with variables and the operations $\vee$ (choice for E), - (dual) and $\circ$ (sequential composition). We call two game expressions G and H identical if they give the same forcing relations for both players in each board model.

The valid game identities form a 'De Morgan Algebra': a Boolean Algebra without special laws for 0 and 1. On top of that, further principles describe interactions between the choice operators and sequential composition. We give a few typical examples of the valid identities, where the '$\wedge$' game is the dual of '$\vee$', with the first choice now for player A, not E:

$$(x \circ y) \circ z \approx x \circ (y \circ z)$$

$$(x \vee y) \circ z \approx (x \circ z) \vee (y \circ z)$$
$$(x \wedge y) \circ z \approx (x \circ z) \wedge (y \circ z)$$

$$-x \circ -y \approx -(x \circ y)$$

$$y \preceq z \rightarrow x \circ y \preceq x \circ z$$

Here $s \preceq$ abbreviates $s \vee t \approx t$. Typically, right-distribution of composition over choice is invalid here:

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

On the left, E has a choice for playing game H or K after playing game G; but on the right, she must decide in advance which of the composed games she has to play. A set of game equations was conjectured by van Benthem (1999) and was proved complete by Goranko (2003), Venema (2003).

In a similar manner, our new logic *CDGL* contains a game algebra, this time also for a parallel operation ×. It is still unknown what this algebra looks like—though we have assembled some valid principles:

$$x \times y \approx y \times x$$

$$(x \times y) \times z \approx x \times (y \times z)$$

$$x \times (y \vee z) \approx (x \times y) \vee (x \times z)$$

$$x \times (y \wedge z) \approx (x \times y) \wedge (x \times z)$$

$$-(x \times y) \approx -x \times -y$$

As a matter of fact, one could have different interpretations of × and consequently different sets of algebraic equations. For example, for some other notions of concurrent game *Idempotence* $x \times x \approx x$ might be considered a desideratum.

This setting suggests a representation result for *CDGL* game algebra in terms of a concrete algebra of consistent pairs of monotone relations over some game board. This was done in Venema (2003) for $DGL^{\bullet}$ (DGL) game algebra, but now, we need to incorporate a product operator—which might give the algebra a ‘quantale’-like structure. We leave this for future investigations.

5 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 $\mathcal{M}$. Verifier claims the formula is true in $\mathcal{M}$, Falsifier claims it is false. The rules of the game $\textbf{eval}(\phi, \mathcal{M}, s)$ are as follows:

- If ϕ is an atom, V wins if ϕ 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 some object d in $\mathcal{M}$, and play continues with ϕ and $s[x := d]$.
- For universal quantifiers $\forall x\psi$, F chooses some object d in $\mathcal{M}$, and play continues with ϕ and $s[x := d]$.

This analogy has been highlighted more formally in van Benthem (2003), 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, together with a game board consisting of all variable assignments over the model $\mathcal{M}$. Players' powers make sense as well in such situations. 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 central features of DGL that were explained earlier occur in the proof for this simple lemma, and this is no coincidence. The Booleans of propositional logic match the game constructs of choice and dual, and quantification involves sequential game composition, 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. Notice that this set-up has the same *generic* character as Parikh's logic: a formula can start an evaluation game at any model and assignment, and moreover, the available concrete moves are not encoded in the formula: what objects can be assigned depends on the model $\mathcal{M}$.

Viewed in this way, first-order evaluation games are a special case of *DGL*, with atomic games of two 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. Still, van Benthem (2003) proves that no generality is lost, through a converse representation:

Theorem 5.1 *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*

(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 Sect. 2.4, to replace abstract atomic games g by evaluation games for quantifier combinations $\exists\forall$. Though the cited paper high-lights the 'game algebra' of *DGL* (Sect. 4.3), this representation also implies (van Benthem (2003), Sect. 3.3):

Corollary 5.2 *There is a faithful embedding of DGL into the complete 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.

Caveat. These results do not say that general games as in $(C)DGL$ should be understood as *test games* ϵ? for propositions ϵ in our earlier sense! The quantifiers are understood here as real moves changing the current state, and as such, they represent arbitrary game moves, or at least, arbitrary changes through forced sets.

The above translation involves converting atomic game modalities $\langle g\rangle\phi$ into a relational modality plus one for 'set membership', following the representation result for powers of players in perfect information games proved in van Benthem (2001). This is also related to our extended languages in Sect. 4.2, where the inclusion modalities access substructure of sets as collective states.

Next, as we observed in our introduction, parallel games occur in the extension of first-order logic to 'independence-friendly' *IF* logic proposed by Hintikka and Sandu (1997) as a procedural analogue of Henkin's 'branching quantifiers'. Indeed, van Benthem (2003) 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 the following pattern from the introduction:

$$\begin{tikzcd}\forall x\exists y \arrow[dr, dash] & \\ & Rxyzu \\ \forall z\exists u \arrow[ur, dash] & \end{tikzcd}$$

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

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

with G, H the two quantifier prefix games, and K the subsequent test game. To make this fit our formal framework, we 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 then be viewed as the set $\{s, t\}$, just as in the semantics of *CPDL*.

In general, collective states will now be sets of partial assignments, whose merge is a new total output state. This introduces a new feature not encountered before. Sometimes, these assignments are 'inconsistent', their union assigns different values to the same variable. Thus, we need a *CPDL*-style semantics where some output sets are consistent, and others are not. But this is easily accommodated. What seems harder to accommodate is the role of atomic predicate Rxy whose variables may refer to different partial assignments. These express truly collective properties of states, more in line with the extended modal framework of Sect. 4.2. We leave this issue open here.

Another nice example of an algebraic way of thinking is the following basic *IF* law. Consider the *IF* formula $\forall x\exists y/x Rxy$, where the slash indicates that Verifier has no access to the value chosen by Falsifier for x. There has been some discussion about equivalents for this, with some people claiming that it is just $\exists y\forall x Rxy$. But van Benthem (2006) shows that the correct transform is the formula $\exists y\forall x/y Rxy$, which gives players the same powers under any concrete interpretation. This inversion of quantifiers seems to go against the received wisdom in first-order logic. But when viewed in terms of our game algebra, it demystifies simply to the earlier commutativity of parallel product!

$$\begin{tikzcd}
\forall x \arrow[dr, dash] & & & \exists y \arrow[dr, dash] & \\
& Rxy & & & Rxy \\
\exists y \arrow[ur, dash] & & & \forall x \arrow[ur, dash] &
\end{tikzcd}$$

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

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

In this connection, van Benthem (2003) has proposed to define powers of players in parallel games through the following stipulation:

$$s\rho^{i}_{G\times H}X \quad \text{iff } \exists U\colon s\rho^{i}_{G}U, \exists V\colon s\rho^{i}_{H}V\colon U \times V \subseteq X.$$

But we have found it hard to make sense of this, and our system *CDGL* is our new attempt at supplying the 'missing corner'.

Clearly, all valid laws of *CDGL* can be instantiated as valid principles of branching quantification, or general *IF* logic. Thus, we have found a *decidable core logic* inside a rather gruesome higher-order system, whose combinatorial nature is still somewhat ill-understood. Even so, open problems abound. For instance,

Question. Can we extend Theorem 5.1 to a representation of *CDGL* models in terms of *IF* evaluation games of imperfect information?

We could try to use the representation theorem for powers in imperfect information games of van Benthem (2001) for this purpose, introducing suitable moves and uniform strategies over given game boards. Another possible result might arise by sticking to a modal language, but using an *IF*-variant for it, as in Sevenster (2006).

Finally, given our discussion, we do not see our proposal as a complete analysis of *IF* logic. But this distance is also a virtue. Compare what Blass (1972) did for Lorenzen's 'dialogue games', famous for their blend of attractive insights and obnoxious details. Inside dense thickets of 'procedural conventions', he saw a compositional game structure, which was the dawn of linear logic. Indeed, Abramsky (2006) has taken a similar look at *IF* logic, and proposed a linear logic-based analysis. While our analysis is more crude than his, it is in the same spirit.

6 Conclusions and intentions

This paper merges dynamic logic of games with 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 decidable by reduction to a poly-modal base logic over generalized neighborhood models. *CDGL* also suggests new algebraic and model-theoretic questions, such as axiomatizing its complete game algebra (or *CPDL* itself), or exploring its model theory based on our new notion of bisimulations. Finally, we showed that *CDGL* may throw some new light on *IF* evaluation games for first-order logic which allow for interpreting independence or imperfect information.

The framework proposed here is a bare minimum, and its reduction of powers to component games, while convenient, makes it a poor theory of parallel play. We have indicated a richer modal language of parallel games in Sect. 4.2, which would allow for genuine collective predicates. In addition, we list some further avenues of investigation in the following.

Modal fixed-point logics. Our system of game logic left out the infinitary game iteration of the original *DGL*, since our main concern was orthogonal: a lift to parallel games. Nevertheless, various infinitary game iterations make sense in game theory and process semantics of linear logic. In particular, van Eijck and Verbrugge (2008) have extended essentially our system to include Parikh's *PDL*-style Kleene iteration. All this raises the issue of how *DGL*s relate to modal fixed-point logics generally. As is well-known, modal logic over neighbourhood models can be translated into bimodal logics over standard relational models, and Parikh (1985) uses this to embed *DGL* into the modal mu-calculus, while the same '$\langle\rangle[]$-trick' drives the representation theorem in van Benthem (2003). Still, neighbourhood semantics has tricky features like this: fixed-point equations for common knowledge are no longer equivalent to *PDL*-style countable iterations: cf. van Benthem and Sarenac (2004). Thus, a translation of *PDL* over neighbourhood models must end up, not in bimodal *PDL* but the mu-calculus. So, what is the natural fixed-point structure of operations on games, when interpreted over modal-style game boards?

Extensive games and communication. Parallel processes and games are studied at the richer level of extensive game trees in Process Algebra and Game Semantics. Connections with our modal analysis over game boards remain to be developed. In particular, linear game semantics has a richer repertoire of game operations with *communication*, including powers of switching between games, and strategic manoeuvres like 'Copy cat': for details, see Abramsky and Jagadeesan (1994). This reflects the *interleaving* nature of its products, whereas our direct products allow no interaction between subgames at all.

Logics of strategies. *DGL* and its ilk state the existence of strategies, but do not display any strategies in their formalism. By contrast, game semantics, and indeed game theory itself has many interesting strategic manoeuvres between stages of one sequential game, or components of parallel games, such as 'strategy stealing'. van Benthem (2002) has used dynamic logic to explicitly define strategies of players in extensive games. In subsequent work, we intend to combine the two perspectives, and do a ‘double gamification’—so to speak—for *PDL*, once as a logic of game structure, and once as a logic of strategies within these games. See Netchitailov (2000), van Benthem (2007) and Ghosh (2008) for some first steps.

Explicit knowledge. Intuitively, parallel games involve knowledge and communication between players. Indeed, *IF* logic involves games of imperfect information, where players need not know each others’ moves, while one of our key examples of non-determinacy was a game of the same sort. van Benthem (2006) has an explicit epistemic language bringing this out. It is not hard to add knowledge modalities to *DGL* and *CDGL*, to state explicitly what players know about their powers—and doing this would be more appropriate for some of our motivating examples. But getting to a complete game logic in this setting is not routine. Game operations may ‘package’ some epistemic information (our product said implicitly that no communication take place). But in general, explicit epistemic structure seems at odds with the compositionality that we have used throughout, in line with the observation in game theory that imperfect information games have no good notion of subgame, and hence no easy game algebra. The experience with compositionality in *IF* logic may be relevant there: see discussions in Hodges (1997).

Goals and preferences. Real game theory is, as we noted in our Introduction, about players pursuing goals driven by their evaluation of outcomes, or their preferences between them. Does our abstract modal approach still make sense in this setting, where raw powers might have to give way to more interactive notions of *equilibrium*?

Acknowledgements The authors thank the anonymous referees for their helpful comments.

References

Abramsky, S. (2006). Socially responsive, environmentally friendly logic. In A. Tuomo & A.-V. Pietarinen (Eds.), *Truth and games: Essays in honour of Gabriel Sandu. Acta Philosophica Fennica*. Helsinki.

Abramsky, S., & Jagadeesan, R. (1994). Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic, 59*, 543–574.

Alur, R., Henzinger, T. A., & Kupferman, O. (2002). Alternating-time temporal logic. *Journal of the ACM, 49*, 672–713.

Baeten, J. C. M., & Verhoef, C. (1995). Concrete process algebra. In S. Abramsky, D. M. Gabbay, & T. S. E. Maibaum (Eds.), *Handbook of logic in computer science* (Vol. 4, pp. 149–268). London: Semantic Modelling, Clarendon Press.

Bergstra, J. A., Ponse, A., & Smolka S. A. (Eds.) (2001). *Handbook of process algebra*. North-Holland, Amsterdam.

Blackburn, P., de Rijke, M., & Venema, Y. (2001). *Modal logic*. CUP.

Blass, A. (1972). Degrees of indeterminacy of games. *Fundamenta Mathematicae, 77*, 151–166.

Ghosh, S. (2008). Strategies made explicit in dynamic game logic. In J. van Benthem & E. Pacuit (Eds.), *Proceedings of the Workshop on Logic and Intelligent Interaction, ESSLLI 2008*, pp. 74–81.

Goldblatt, R. (1992). Parallel action: Concurrent dynamic logic with independent modalities. *Studia Logica, 51*, 551–578.

Goranko, V. (2003). Basic algebra of game equivalences. *Studia Logica, 75*, 221–238.

Hansen, H., Kupke, C., & Pacuit, E. (2007). Bisimulation for neighbourhood structures. In M. Haveraaen, U. Montanari, & T. Mossakowski (Eds.), *Proceedings of 2nd Conference on Algebra and Coalgebra in Computer Science (CALCO 2007)* (pp. 279–293). LNCS 4624.

Harel, D., Kozen, D., & Tiuryn, J. (2000). *Dynamic logic*. The MIT Press.

Hintikka, J., & Sandu, G. (1997). Game-theoretical semantics. In J. van Benthem & A. ter Meulen (Eds.), *Handbook of logic and language* (pp. 361–410). MIT Press.

Hodges, W. (1997). Compositional semantics for a language of imperfect information. *Journal of the Interest Group in Pure and Applied Logics, 5*, 539–563.

Netchitailov, I. (2000). *An extension of game logic with parallel operators*. Master's thesis, University of Amsterdam.

Parikh, R. (1985). The logic of games and its applications. *Annals of Discrete Mathematics, 24*, 111–140.

Pauly, M. (2001). *Logics for social software*. Ph.D. Dissertation, University of Amsterdam.

Pauly, M., & Parikh, R. (2003). Game logic—An overview. *Studia Logica, 75*, 165–182.

Peleg, D. (1987). Concurrent dynamic logic. *Journal of the ACM, 34*, 450–479.

Sevenster, M. (2006). *Branching of imperfect information: Logic, games and computation*. Ph.D. dissertation, University of Amsterdam.

van Benthem, J. (1999). *Logic in games*. Lecture Notes, Amsterdam and Stanford.

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

van Benthem, J. (2002). Extensive games as process models. *Journal of Logic, Language and Information, 11*, 289–313.

van Benthem, J. (2003). Logic games are complete for game logics. *Studia Logica, 75*, 183–203.

van Benthem, J. (2006). The epistemic logic of IF games. In R. E. Auxier & L. E. Hahn (Eds.), *Philosophy of Jaakko Hintikka*. Open Court.

van Benthem, J. (2007). In praise of strategies, 2007. In J. van Eijck & R. Verbrugge (Eds.), *Games, actions and social software* (to appear).

van Benthem, J., Ghosh, S., & Liu, F. (2007). Modelling simutaneous games with concurrent dynamic logic. In J. van Benthem, S. Ju, & F.Veltman (Eds.), *A meeting of the minds—Proceedings of the workshop on logic, rationality and interaction* (pp. 243–258). London: King's College Publications.

van Benthem, J., & Sarenac, D. (2004). The geometry of knowledge. In J.-Y. Béziau, A. Costa Leite, & A. Facchini (Eds.), *Aspects of universal logic* (pp. 1–31). Centre de Recherches Sémiologiques, Université de Neuchatel.

van Benthem, J., van Eijck, J., & Stebletsova, V. (1994). Modal logic, transition systems and processes. *Journal of Logic Computation, 4*, 1–50.

van Eijck, J., & Verbrugge, R. (2008). *A fresh look at propositional dynamic logic and game logic*. CWI Amsterdam and Department of AI, University of Groningen.

Venema, Y. (2003). Representation of game algebras. *Studia Logica, 75*, 239–257.