Johan van Benthem

Logic games are complete for game logics

Abstract. Game logics describe general games through powers of players for forcing outcomes. In particular, they encode an algebra of sequential game operations such as choice, dual and composition. Logic games are special games for specific purposes such as proof or semantical evaluation for first-order or modal languages. We show that the general algebra of game operations coincides with that over just logical evaluation games, whence the latter are quite general after all. The main tool in proving this is a representation of arbitrary games as modal or first-order evaluation games. We probe how far our analysis extends to product operations on games. We also discuss some more general consequences of this new perspective for standard logic.

Keywords: logic game, powers, dynamic logic, game algebra

1. Logical evaluation games

Many logical notions can be cast very naturally as two-player games. Examples are argumentation between a defender and critic of a claim (Lorenzen games), model comparison between people disputing an analogy (Ehrenfeucht-Fraïssé games), and perhaps most basically of all, semantical evaluation of assertions made with respect to some given situation [13]. In this paper, we concentrate on the latter games, extracting their general thrust. The notion of game in what follows is standard. *Games in extensive form* are trees whose nodes are possible states of play, with labeled arrows from a node to its daughters indicating the available moves. Nodes are either 'in play', marked for the player whose turn it is, or end nodes, where the game has ended successfully. But a game may also end 'unsuccessfully', with a scheduled turn for a player without available moves. Game nodes may carry information about further properties, such as players' having won or lost in end nodes, or markings for more finely-grained utilities. Moves encode local actions for players in game states. More global patterns of behaviour are defined as follows. A *strategy* for player $\boldsymbol{i}$ is a subtree of the full game tree in which each turn for $\boldsymbol{i}$ has a unique outgoing move, while at all other nodes, the subtree retains all outgoing arrows from the original game tree.

1.1. First-order evaluation games

Two players dispute the truth of a formula ϕ in some model $\mathbf{M}$, starting from a given assignment s sending variables to objects in the domain. *Verifier*

(V) claims that the formula is true in **M**, s, *Falsifier* **(F)** claims that it is false. The rules of this game **eval(**ϕ**, M,** s**)** are usually stated informally:

a) If ϕ is an atom, **V** wins if the atom is true, and **F** wins if it is false,

b1) For formulas $\phi \vee \psi$, **V** chooses a disjunct to continue with

b2) For formulas $\phi \wedge \psi$, **F** chooses a conjunct to continue with

c) With negations $\neg\phi$, the two players switch roles

d1) For an existential quantifier $\exists x\psi$,**V** chooses an object d in **M**, and play continues w.r.t. ψ and the new assignment $s[x{:=}d]$

d2) For a universal quantifier $\forall x\psi$, **F** chooses an object d in **M**, and play continues w.r.t. ψ and the new assignment $s[x{:=}d]$.

To be more precise, one can define game trees for these evaluation games in an inductive manner. Nodes are tuples of the form

$$(t,\ \psi)$$

where t is the current assignment, and ψ the remaining formula to be played.

a) **eval(**Px**, M,** s**)** has top node $(t,\ Px)$: a turn for **V**. If **M**, $s \models Px$ the game moves to a single end node $(t, \text{-})$ which is a win for **V**. Otherwise, the game stops in the top node with 'dead-lock', which is a loss for **V**.

b) **eval(**$\phi\vee\psi$**, M,** s**)** consists of disjoint copies of **eval(**ϕ**, M,** s**)**, **eval(**ψ**, M,** s**)**, put under one initial node $(s,\ \phi \vee \psi)$, which is a turn for **V**, with actions 'left' and 'right' to the topnodes of the component games. The game **eval(**$\phi\wedge\psi$**, M,** s**)** is similar, but starting with a turn for **F**.

c) **eval(**$\neg\psi$**, M,** s**)** is the *dual* of **eval(**ψ**, M,** s**)** reversing turn indications and win-lose markings for the two players. Also, formulas in game nodes are syntactically dualized, interchanging conjunctions/disjunctions, existential/universal quantifiers, and the polarity of atomic formulas.

d) **eval(**$\exists x\psi$**, M,** s**)** starts with a top node which is **V**'s turn, followed by possible moves to the top nodes of all games **eval(**ϕ**, M,** $s[x{:=}d]$**)**, where d runs over all objects in **M**. The game **eval(**$\forall x\psi$**, M,** s**)** is completely similar, but now starting with a turn for **F**.

This is just one possible definition, with some peculiarities. E.g., the clause for atomic games has its rather technical format for later convenience. Also, our simple clause for negation is different in character from the others, as dual erases earlier information, so that **game** $(\neg\psi$, **M**, $s)$ is not a precise record of what has been played so far. Other options are possible, but the present ones suffice for us.

Independently from such details of format, first-order evaluation games have general game-theoretic features. E.g., they have a fixed finite depth for their longest runs, bounded by the operator depth of the initial formula. This makes them subject to *Zermelo's Theorem* stating that

> Each two-person zero-sum game of perfect information with finite tree depth is *determined*: i.e., one of the two players has a winning strategy.

Thus, either **V** or **F** must have a winning strategy in an evaluation game. This observation is one instance of a more general fact. Classical truth of first-order formulas in a model amounts to Verifier's having a guaranteed win for the associated game:

Proposition The following two assertions are equivalent:

a) **V** has a winning strategy in **eval**(ϕ, **M**, s)
b) **M**, $s \models \phi$

The proof is a straightforward induction on formulas - most easily, by including the dual assertion for **F** . E.g., **V** has a winning strategy in the dual game $\neg\psi$ iff **F** has a winning strategy in ψ. As another typical illustration, **V** has a winning strategy in $\phi \vee \psi$ iff she has one in either subgame, whereas **F** only has one if he can win both subgames. Through the Proposition, logical laws now come to express game-theoretic facts. E.g., determinacy shows in the validity of *Excluded Middle* $\phi \vee \neg\phi$ for these games. This correspondence will return in what follows.

1.2. Modal games

Evaluation games also work for extensions of first-order logic with second-order quantifiers, or fixed-point operators (Note 1). They also fit weaker languages, such as modal propositional logic, which will be used below, with

> proposition letters p, q, ..., Boolean operators, and modalities $\Diamond, \Box$

As above, game states are pairs (s, ϕ), with s a state in the relevant modal model **M**, and ϕ the current modal formula.

The new rules concern the modalities:

a) At $(s, \Diamond\phi)$, **V** must pick an R-successor t of the current state s, and the game continues w.r.t. (t, ϕ). If there are no R-successors, then **V** loses the game right at this stage.

b) At $(s, \Box\phi)$, **F** must pick an R-successor t of the current state s, and the game continues w.r.t. (t, ϕ). If there are no R-successors, then **F** loses the game right at this stage.

Modalities are like *bounded quantifiers* $\exists y(Rxy\wedge$ and $\forall y(Rxy\rightarrow$. Their moves differ from those for quantifiers in first-order games, which can always be performed. By contrast, modal games may have a player's turn where no strategy for that player can assign a move. Consider the model $\{1, 2\}$ with the relation $\{(1, 1), (1, 2)\}$. Evidently, in state 1, the modal formula $\Diamond\Diamond\top$ holds. Here is the game tree for this formula, indicating the player to move:

$$\begin{tikzcd}[column sep=small]
& (1, \Diamond\Diamond\top), \mathbf{V} \arrow[dl, no head] \arrow[dr, no head] & \\
(1, \Diamond\top), \mathbf{V} \arrow[d, no head] & & (2, \Diamond\top), \mathbf{V} : \text{dead-lock, } \textit{loss for } \mathbf{V} \\
(1, \top) : \textit{win for } \mathbf{V} & &
\end{tikzcd}$$

V has a winning strategy, even though she can never move in state $(2, \Diamond\top)$. The main Proposition remains as for first-order evaluation games. (Note 2.)

2. From logic games to game logics

Logic games, though serving specific aims, show a lot of general game structure. Pursuing this leads to much more general *game logics*. Henceforth, we use **1, 2** for players of any game, with $\boldsymbol{i}$, $\boldsymbol{j}$ as variables for different players.

2.1. Actions and powers

Extensive game trees record all possible moves. But the Proposition made a more global statement about strategies for players guaranteeing certain effects - such as winning outcomes. Consider any strategy σ for a player $\boldsymbol{i}$, defined as a subtree of the full game. This strategy enables player $\boldsymbol{i}$ to make sure, against every possible counterplay by the opponent, that the game ends in a leaf of that subtree, and hence in a particular set O_σ of outcomes for the total game. The family of these outcome sets for all $\boldsymbol{i}$'s strategies gives a player's 'powers' in the game:

$\rho^{i}_{G}\ s, X$ player $\boldsymbol{i}$ has a strategy for playing game $\boldsymbol{G}$ from state s onward whose outcome set is contained in the set X

These *power relations* are generalized transition relations in an interactive two-agent process, relating states to sets of states - rather than just states.

These state-to-set relations may also be defined more generally. The key Proposition for evaluation games connected players' powers inside a game with ordinary assertions about states s on the model **M**. The latter serves as an external *game-board*. Moreover, we had an obvious projection map F from internal game states (s, ψ) to external board states s (Note 3). But then, with any such map we can also define players' powers on a game board:

$\rho^{i}_{\mathbf{M}}\ \mathrm{F}(s), A$ if $\rho^{i}_{G}\ s, X$ for some set of game states X with $\mathrm{F}[X] \subseteq A$

Fact Players' powers, defined either way, satisfy the following properties:

C1)	if $\rho^{i}_{G}\ s, Y$ and $Y \subseteq Z$, then $\rho^{i}_{G}\ s, Z$	Monotonicity
C2)	if $\rho^{i}_{G}\ s, Y$ and $\rho^{j}_{G}\ s, Z$, then Y, Z overlap	Consistency
C3)	if not $\rho^{i}_{G}\ s, Y$, then $\rho^{j}_{G}\ s, -Y$	Determinacy

Proof C1 expresses that larger sets represent weaker powers. C2 says that, if both players follow some strategy of theirs, an outcome must result. C3 expresses the special property of determinacy in abstract set-theoretic form. □

2.2. Game operations and algebra

Evaluation games bring out several completely general operations on games. These include dynamic counterparts of the Boolean operations:

Choice Choice for some specified player. For instance, for player **1**, $G \cup H$ is the result of putting two disjoint copies of G, H under a new root, giving **1** the choice which one to play. $G \cap H$ denotes the analogous construction for player **2**.

Dual The dual G^{d} reverses all turns and win/lose markings in G.

But there is a third general game operation operating in first-order evaluation games. This is not the earlier object picking for quantifiers, which is just a specific move in a special semantic setting. It is rather the glue sticking a quantifier to its matrix:

Composition Composition $G;H$: first play game G, and then H, starting from the states where the first game ended successfully.

E.g., a modal formula ◊□p composes three games: '**V** picks successor' **;** '**F** picks successor' **;** 'atomic test'. We define this more precisely later on.

These operations support a natural algebra of games. E.g., it is easy to see the intuitive validity of De Morgan laws on the game interpretation:

$$G \cap H = (G^{\boldsymbol{d}} \cup H^{\boldsymbol{d}})^{\boldsymbol{d}}$$

This is just one of many laws of game algebra, to be made more precise below. On the present analysis, predicate logic is a general theory of sequential game constructions, over two specific base games, viz. object picking and fact testing. We will see some surprising consequences of this view later. (Note 4.)

2.3. Game language

To study the above situation more generally, we adopt a perspective due to [16], on the pattern of propositional dynamic logic. The language has both propositions P and game expressions G, according to the following syntax:

P	$\mathrm{At}P \mid \vee \mid \wedge \mid \neg \mid \{\boldsymbol{G},\ \boldsymbol{i}\}P$
G	$\mathrm{At}G \mid \cup \mid \cap \mid {}^{\boldsymbol{d}} \mid \boldsymbol{;} \mid P\boldsymbol{?}$

That is, formulas are formed from atomic ones p, q, ... with Boolean operations, plus a game modality $\{\boldsymbol{G},\ \boldsymbol{i}\}P$ expressing players' powers to achieve a certain type of outcome state. Game expressions are formed from atomic games g, h, ... using the above operations, plus an operation **?** taking propositions to 'test games'. A first semantics for this language works as follows. Game models are structures

$$\mathbf{M} = (S,\ \mathbf{game},\ V)$$

with a set of states S, a function $\mathbf{game}(g,\ s)$ assigning concrete games to every basic game expression at every state in S, and V a valuation for atomic propositions at states. The semantics interprets propositions in tandem with a lift of **game** to arbitrary game expressions, following the given operations:

$$\mathbf{M},\ s \models \phi \qquad\qquad \mathbf{game}(G,\ s,\ \mathbf{M})$$

The only non-routine clauses here are the following:

a)	$\mathbf{M},\, s \models \{G, \boldsymbol{i}\}\phi$	iff $\mathbf{game}(G,\, s,\, \mathbf{M})$ has a strategy for player $\boldsymbol{i}$ achieving a set of outcomes x s.t. $\mathbf{M},\, x \models \phi$
b)	$\mathbf{game}(\phi\mathbf{?},\, s,\, \mathbf{M})$	is a move by $\mathbf{1}$ to an end state if $\mathbf{M},\, s \models \phi$, and a dead-lock for $\mathbf{1}$ otherwise.

(Note 5). This modeling is still too detailed for many purposes. A more convenient second semantics just looks at power relations over the game-board, defined as earlier using a map F sending game states to board states:

$$\rho_{\boldsymbol{G}}^{\mathbf{M},\boldsymbol{i}}\ s,\ X \text{ if, for some set of game states } A,\ \rho^{\boldsymbol{i}}_{\mathbf{game}(G,s,\mathbf{M})}\ s,\ A \wedge \mathrm{F}[A] \subseteq X$$

<u>Fact</u> These power relations satisfy the following inductive clauses

$$\begin{array}{lll}
\rho^{\mathbf{1}}_{\boldsymbol{G}\cup\boldsymbol{H}}\ s,\ Y & \text{iff} & \rho^{\mathbf{1}}_{\boldsymbol{G}}\ s,\ Y \ \textit{or}\ \ \rho^{\mathbf{1}}_{\boldsymbol{H}}\ s,\ Y \\
\rho^{\mathbf{2}}_{\boldsymbol{G}\cup\boldsymbol{H}}\ s,\ Y & \text{iff} & \rho^{\mathbf{2}}_{\boldsymbol{G}}\ s,\ Y \ \textit{and}\ \rho^{\mathbf{2}}_{\boldsymbol{H}}\ s,\ Y \\
\rho^{\mathbf{1}}_{\boldsymbol{G}^d}\ s,\ Y & \text{iff} & \rho^{\mathbf{2}}_{\boldsymbol{G}}\ s,\ Y \\
\rho^{\mathbf{2}}_{\boldsymbol{G}^d}\ s,\ Y & \text{iff} & \rho^{\mathbf{1}}_{\boldsymbol{G}}\ s,\ Y \\
\rho^{\mathbf{1}}_{\boldsymbol{G};\boldsymbol{H}}\ s,\ Y & \text{iff} & \exists Z: \rho^{\mathbf{1}}_{\boldsymbol{G}}\ s,\ Z \wedge \forall z{\in}Z\colon \rho^{\mathbf{1}}_{\boldsymbol{H}}\ z,\ Y \\
\rho^{\mathbf{2}}_{\boldsymbol{G};\boldsymbol{H}}\ s,\ Y & \text{iff} & \exists Z: \rho^{\mathbf{2}}_{\boldsymbol{G}}\ s,\ Z \wedge \forall z{\in}Z\colon \rho^{\mathbf{2}}_{\boldsymbol{H}}\ z,\ Y
\end{array}$$

<u>Proof</u> These inductive clauses largely speak for themselves. Here is the case of composition. Suppose that $\rho^{\mathbf{1}}_{\boldsymbol{G};\boldsymbol{H}}\ s,\ Y$. This means that in $\mathbf{game}(G;H,\, s,\, \mathbf{M})$, player $\mathbf{1}$ can force a set of outcomes A with $\mathrm{F}[A] \subseteq Y$ - say, via a strategy σ. Now, the restriction $\sigma|G$ of σ to G played from s forces a set of end positions U. From each of these, the remaining strategy $\sigma|H$ forces a subset of A. But then, $\mathrm{F}[U]$ is the required set Z, using monotonicity to get $\rho^{\mathbf{1}}_{\boldsymbol{H}}\ z,\ Y$ for each state $z{\in}Z$. Next, let $\exists Z: \rho^{\mathbf{1}}_{\boldsymbol{G}}\ s,\ Z \wedge \forall\ z{\in}Z\colon \rho^{\mathbf{1}}_{\boldsymbol{H}}\ z,\ Y$. Then $\mathbf{1}$ has a strategy σ for playing $\mathbf{game}(G,\, s,\, \mathbf{M})$ forcing an outcome set A with $\mathrm{F}[A] \subseteq Z$. By the second conjunct, $\mathbf{1}$ has strategies τ_z in each $\mathbf{game}(H,\, z,\, \mathbf{M})$ forcing sets B_z with $\mathrm{F}[B_z] \subseteq Y$. The composition of σ and all τ_z is a strategy for playing G **;** H forcing a set of outcomes mapped by F into Y. □

For determined games, it suffices to state the powers of player $\mathbf{1}$ only:

$$\begin{array}{lll}
\rho_{\boldsymbol{G}\cup\boldsymbol{H}}\ s,\ Y & \text{iff} & \rho_{\boldsymbol{G}}\ s,\ Y\ \textit{or}\ \rho_{\boldsymbol{H}}\ s,\ Y \\
\rho_{\boldsymbol{G}^d}\ s,\ Y & \text{iff} & \neg\rho_{\boldsymbol{G}}\ s,\ S\text{-}Y \\
\rho_{\boldsymbol{G};\boldsymbol{H}}\ s,\ Y & \text{iff} & \exists Z: \rho_{\boldsymbol{G}}\ s,\ Z \wedge \forall z{\in}Z: \rho_{\boldsymbol{H}}\ z,\ Y
\end{array}$$

2.4. Logic over game boards

We now define our eventual game models as structures

$$\mathbf{M} = (S,\ \{\rho_g |\ g \in \mathrm{BG}\},\ V)$$

where S is a set of states, V is a valuation for proposition letters, and BG is a set of basic games whose power relations ρ_g are hard-wired into the game board. Of the three conditions in Section 2.1, we impose one: *upward monotonicity* (C1) - the other two will follow automatically for determined games. Moreover, we now *define* power relations for all games using the above inductive clauses. The semantics then becomes like that for dynamic logic, with two special clauses:

a)	$\mathbf{M},\ s \models \{G,\ \boldsymbol{i}\}\phi$	iff	$\exists X : \rho_G^{\boldsymbol{i},\ \mathbf{M}}\ s,\ X$ and $\forall x \in X$: $\mathbf{M},\ x \models \phi$
b)	$\rho_{\phi?}^{\boldsymbol{i},\mathbf{M}}\ s,\ Y$	iff	$\mathbf{M},\ s \models \phi$ and $s \in Y$

We suppress the $\boldsymbol{i}$ henceforth, recording just one player in determined games.

Evaluation games provide concrete illustrations. In the first-order setting, the game board consists of the variable assignments s into the given domain of individuals. The power relations for Verifier in the two basic games are:

$\rho_{Px}^{\boldsymbol{V}}\ s,\ X$	iff	$P^{\mathbf{M}}\ (s(x))$ and $s \in X$
$\rho_{\exists x}^{\boldsymbol{V}}\ s,\ X$	iff	for some d in $\lvert\mathbf{M}\rvert$, $s[x{:=}d] \in X$

The earlier inductive clauses take care of all complex formulas. Modal games have the modal model itself as a board, with basic power relations:

$\rho_p^{\boldsymbol{V}}\ s,\ X$	iff	$V(p,\ s) = 1$ and $s \in X$
$\rho_{\Diamond}^{\boldsymbol{V}}\ s,\ X$	iff	for some t with Rst, $t \in X$

The system resulting from the above general semantics is *Dynamic Game Logic* (DGL) studied by [16] and [17]. Here is the basic result:

<u>Theorem</u> DGL is decidable, and its validities are axiomatized by

a) all valid principles of propositional logic: both axioms and rules

b) monotonicity: if $\phi \to \psi$ is provable, then so is $\{G\}\phi \to \{G\}\psi$

c) reduction laws for existence of strategies in compound games:

$$\begin{array}{rcc} \{G;H\}\phi & \leftrightarrow & \{G\}\{H\}\phi \\ \{G \cup H\}\phi & \leftrightarrow & \{G\}\phi \vee \{H\}\phi \\ \{G^d\}\phi & \leftrightarrow & \neg\{G\}\neg\phi \\ \{P?\}\phi & \leftrightarrow & P \wedge \phi \end{array}$$

These axioms encode basic reasoning about powers in games. (Note 6.) For instance, formalize the proof of the Proposition for evaluation games in Section 1. Inductive steps are provided by the above axioms, with the base step just the definition of the *win* predicate for players in atomic games.

When games are not assumed to be determined, one can use a similar set of DGL axioms, but now with a pair of modalities $\{G, \mathbf{1}\}\phi$, $\{G, \mathbf{2}\}\phi$ following the earlier simultaneous decomposition of power relations for both players.

2.5. Game algebra

DGL can also express equivalence of two game expressions G, H by means of validity of the assertion $\{G\}p \leftrightarrow \{H\}p$. This says that

> the power relations $\rho_G^{i,\mathbf{M}}$ of G, H for both players as defined above are the same on every game board $\mathbf{M}$

This notion of equivalence can also be argued for on independent grounds. It validates an algebra of game equivalence whose equational axiomatization was first conjectured in [4]. The following result is from [12, 19], omitting test games for convenience:

Theorem Game Algebra is axiomatized completely by means of

1) De Morgan algebra for choices and dual (Note 7)

$$\begin{array}{llcll} 2) & G \,;\, (G';G'') & = & (G;G') \,;\, G'' & \text{associativity} \\ & (G \cup G') \,;\, G'' & = & (G;G'') \cup (G';G'') & \text{left-distribution} \\ & (G;G')^d & = & G^d \,;\, G'^d & \text{dualization} \\ 3) & G \leq G' & \to & H \,;\, G \leq H \,;\, G' & \text{right-monotonicity} \end{array}$$

Typically though, right-distribution of composition over choice is invalid:

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

1's choice for H or K on the left, but not on the right, may use the result of G. As for the connection with the preceding game logic, all these algebraic axioms may be derived from the given axioms of DGL.

3. From powers to evaluation games

In Section 2, game boards and power relations were presented as a generalization of evaluation games, culminating in general game logic and algebra. Now, we go the other way, showing how all such general structures already live inside evaluation games.

3.1. A simple representation

Though not strictly necessary for what follows, the following helps understand our modus operandi. Determined two-player games assign families of powers P_i to their players $\boldsymbol{i}$, subject to the constraints of Section 2.1:

C1) if $Y \in P_i$ and $Y \subseteq Z$, then $Z \in P_i$
C2) if $Y \in P_i$ and $Z \in P_j$, then Y, Z overlap
C3) if $Y \notin P_i$, then $S\text{-}Y \in P_j$

Now, these conditions are also all that must hold, witness the following

Proposition Any two families P_1, P_2 of subsets of some set S satisfying the three conditions C1, C2, C3 are the root powers in some two-step game.

Proof Start with player **1** and let her choose between the sets in P_1. At these nodes, player **2** gets to move, and can pick any member of that set. Clearly, **1** has the powers specified in P_1. Now for player **2**. In the game just defined, she can force any set of outcomes that *overlaps with each set in* P_1. But by C2, C3, these are precisely the sets in her initial family P_2. E.g., if a set of outcomes A overlaps with all sets in P_1, its complement $S\text{-}A$ cannot be in that family, and so the set A itself is in P_2, by C3. □

This representation allows the same outcome at different end nodes of the game tree. Again there is some game-board flavour here, with an identification map F . If one wants the outcomes unique on each branch of the game, then the following strengthening of C2 is necessary and sufficient:

C2$^+$) if $Y \in P_i$ and $Z \in P_j$, then Y, Z overlap in just one point

3.2. Modal representation

In terms of powers, general games are still close to logical evaluation games. As a result, basic game algebra coincides with the algebra of evaluation games for ordinary predicate logic, or even propositional modal logic. The following analysis, a stream-lined version of [6], was inspired by [16]. Consider an arbitrary game board $\mathbf{M} = (S, \{\rho_g |\ g{\in}\mathrm{BG}\},\ V)$. Now, define an associated standard modal model as follows:

$$\mathbf{M^*} = (S \cup \mathrm{POW}(S), \{R_g |\ g{\in}\mathrm{BG}\},\ V)$$

whose states are all states in $\mathbf{M}$ plus all sets of such states. The binary relations R_g hold only between state objects s and set objects X such that ρ_g s, X held in $\mathbf{M}$. The binary relation $\in$ is standard set membership. Finally, the valuation V makes proposition letters true only at those states s where they were true in $\mathbf{M}$. Think of $\mathbf{M^*}$ as a two-sorted first-order version of $\mathbf{M}$.

Next, we define a translation $\mathbf{t}$ taking (a) DGL formulas ϕ to modal formulas $\mathbf{t}(\phi)$, and (b) DGL game expressions G to game expressions $\mathbf{t}(G)$ whose atoms are modal evaluation games. The latter may be viewed as generalized evaluation games. In what follows, the modality $\boxed{\ni}$ refers to all elements of the current object:

$$
\begin{array}{lcl}
\mathbf{t}(\mathrm{p}) & = & \mathrm{p} \\
\mathbf{t}(\neg\phi) & = & \neg\mathbf{t}(\phi) \\
\mathbf{t}(\phi \vee \psi) & = & \mathbf{t}(\phi) \vee \mathbf{t}(\psi) \\
\mathbf{t}(\{\mathrm{g}\}\phi) & = & \langle \mathrm{g} \rangle \boxed{\ni}\mathbf{t}(\phi) \\
\mathbf{t}(\{G{\cup}H\}\phi) & = & \mathbf{t}(\{G\}\phi) \vee \mathbf{t}(\{H\}\phi) \\
\mathbf{t}(\{G^{\boldsymbol{d}}\}\phi) & = & \neg\mathbf{t}(\{G\}\neg\phi) \\
\mathbf{t}(\{G;H\}\phi) & = & \mathbf{t}(\{G\}\{H\}\phi) \\
\mathbf{t}(\mathrm{g}) & = & \langle \mathrm{g} \rangle \boxed{\ni} \\
\mathbf{t}(G{\cup}H) & = & \mathbf{t}(G){\cup}\mathbf{t}(H) \\
\mathbf{t}(G^{\boldsymbol{d}}) & = & \mathbf{t}(G)^{\boldsymbol{d}} \\
\mathbf{t}(G;H) & = & \mathbf{t}(G);\mathbf{t}(H)
\end{array}
$$

A straightforward induction shows how this works:

<u>Proposition</u> The following pairs of assertions are equivalent, one for all DGL formulas ϕ, and one for all game expressions G:

a)	$\mathbf{M}, s \models \phi$	a$'$)	$\mathbf{M^*}, s \models \mathbf{t}(\phi)$
b)	$\rho_G^{i,\mathbf{M}}\ s,\ X$	b$'$)	$\rho_{\mathbf{t}(G)}^{i,\mathbf{M}}\ s,\ X$

<u>Proof</u> The heart of the matter is a fact which reflects the representation of Section 3.1. The power relation for a game is exactly the same as that for the associated evaluation game. If $\rho_g^{i,\mathbf{M}}$ s, X, then Verifier can choose X in

the game for $\langle g\rangle[\ni]$, after which every move by Falsifier gives an object in the set X. Thus, X is a power for $\mathbf{V}$ in the evaluation game. Conversely, if X is such a power for Verifier, then there is an R_g-successor of s all of whose elements are in X. But then also $\rho_g^{i,\mathbf{M}}\ s,\ X$, by monotonicity. $\square$

Instead of the modal prefix $\langle g\rangle[\ni]$, the complete modal formula $\langle g\rangle[\ni]\top$ will work just as well in games $\mathbf{t}(G)$ - where $\top$ is the game that always succeeds. Here is some further information on this modal reduction.

Proposition For all game expressions G and modal power formulas ϕ, when interpreted in the obvious way in the model $\mathbf{M^*}$ at any state s:

$$\mathbf{eval}(\mathbf{t}(\{G\})\phi) = \mathbf{t}(G)\ ;\ \mathbf{eval}(\mathbf{t}(\phi))$$

Proof The proof is again an obvious induction. We do two illustrative cases.

$$\begin{array}{lll} \text{a)}\ \mathbf{eval}(\mathrm{t}(\{\mathrm{g}\}\phi) & = & \mathbf{eval}(\langle g\rangle[\ni]\mathbf{t}(\phi)) \\ & = & \mathbf{t}(\mathrm{g})\ ;\ \mathrm{eval}(\mathbf{t}(\phi)) \end{array}$$

$$\begin{array}{lll} \text{b)}\ \mathbf{eval}(\mathrm{t}(\{G;H\}\phi) & = & \mathbf{eval}(\mathbf{t}(\{G\}\{H\}\phi) \\ & = & \mathbf{t}(G)\ ;\ \mathbf{eval}(\mathbf{t}(\{H\}\phi)) \\ & = & \mathbf{t}(G)\ ;\ (\mathbf{t}(H);\mathbf{eval}(\mathbf{t}(\phi))) \\ & = & (\mathbf{t}(G);\mathbf{t}(H))\ ;\ \mathbf{eval}(\mathbf{t}(\phi)) \\ & = & \mathbf{t}(G;H)\ ;\ \mathbf{eval}(\mathbf{t}(\phi)) \end{array}$$ $\square$

Finally, here is another way of thinking about the latter game. Confusing formulas and evaluation games in a harmless manner:

Proposition $\mathbf{t}(G)\ ;\ (\mathbf{t}(\phi))$ is the modal evaluation game for the formula obtained by tagging on the formula ϕ behind all 'final occurrences' of an atomic game $\langle g\rangle[\ni]$ in the game expression $\mathbf{t}(G)$.

Proof This is justified by the following set of validities in Game Algebra:

$$\begin{array}{lll} (G\cup H)\ ;\ K & = & (G;K) \cup (H;K) \\ (G;H)\ ;\ K & = & G\ ;\ (H;K) \\ (G^d)\ ;\ H & = & (G\ ;\ H^d)^d \end{array}$$ $\square$

3.3. General game algebra is algebra of evaluation games

The preceding reductions show how models for game logic and game algebra can also be viewed as models for a standard modal language and its evaluation games. Here is one striking consequence of this observation, and the main result of this paper - inspired by the analysis of dynamic predicate logic in [20] (cf. Note 8).

Theorem The following are equivalent for any two game expressions G, H:

a) $G = H$ is valid in general Game Algebra
b) $G = H$ is valid in the algebra of modal evaluation games.

Proof From (a) to (b) is obvious. From (b) to (a), suppose the interpretations of G, H have different forcing relations on a game board $\mathbf{M}$. Then, they differ in powers at some state s, say: $\rho_G^{\mathbf{M}}\ s,\ X$ but not $\rho_H^{\mathbf{M}}\ s,\ X$. Now take some new proposition letter p and make it true only in the states of X. With this valuation, $\mathbf{M}, s \models \{G\}p$, but not $\mathbf{M}, s \models \{H\}p$. Then, as in Section 3.2, there are two modal evaluation games having exactly the same operational structure as G,H - with suitable modalities and proposition letters plugged in for atoms - that differ on the game board for the model $\mathbf{M^*}$. □

The point here is that game algebra of a very special class of logic games is rich enough to give all the structure of general game algebra. This justifies a certain pride of place for logical evaluation games in general game logic.

Similar observations may be made about a modal embedding of game logic.

Fact Every modal power statement which is falsifiable for abstract games can also be falsified on a game board for first-order evaluation games.

The reason is that the above reduction works just as well on models for game logic - witness the connection with the μ-calculus made in [16].

3.4. The link with first-order games

In an earlier version of this paper ([6]), a complex construction linked failures of equations in game algebra to failures in first-order evaluation games. We now analyze this in a modular way. The modal representation of Section 3.3 needs to be lifted to a first-order one. Now modal propositional formulas ϕ can be translated into formulas in a first-order language quantifying over states, with modalities becoming *bounded* quantifiers:

$\Diamond p$ goes to $\exists y \cdot Rxy\colon\ Py$

$\Box p$ goes to $\forall y \cdot Rx\colon\ Py$

This translation works at the level of truth. But the two sorts of evaluation games are different (cf. Section 1). Moves for modalities may result in deadlock, whereas those for quantifiers always succeed. The difference shows in the games for $\Diamond p$ and $\exists y(Rxy \wedge Py)$. (Note 9.) But in terms of power relations from any state s, there is no difference. In the modal game for $\Diamond p$, Verifier can force those sets X containing some R-successor t of s which satisfies p. Analogously, in the first-order game for $\exists y(Rxy \wedge Py)$, Verifier can force those sets of variable assignments that contain some variant $s[y{:=}d]$ of s such that both $Rs(x)d$ and Pd hold. This suggests the following formal reduction. We translate the above modal counter-examples on models $\mathbf{M^*}$ to first-order evaluation games over a game board whose states are

assignments of S^*-objects to the two variables x, y

Then atomic games are replaced, not by $\langle g \rangle [\ni]$, but by

$\exists y$; Rxy ; $\forall x$; $x \in y$

This makes all the state-change actions take place in the x-argument.

<u>Corollary</u> First-order evaluation games are complete for Game Algebra.

(Notes 10, 11.)

4. Standard logic as game logic

The game connection does not just reinterpret existing logical systems, it also suggests a fresh look at their architecture. Evaluation games are an alternative semantics for first-order logic, providing much richer denotations than standard truth values, or sets of assignments - which may be of use for more intensional theories of propositions. (Note 12.) This suggests a different philosophical look at first-order logic itself. With formulas viewed as evaluation games, the key ingredient of Tarski semantics are its dynamic *procedures*. These start from fact testing for atomic assertions and object picking for quantifiers. Further game operations create complex procedures out of these, via *choice* and *dual* (the Boolean structure) and *sequential composition* (the hidden structure encoded in the gaps behind a quantifier symbol). This changes our view of logical validity, as standard first-order logic now becomes a mix of two things: some specific semantic operations on particular models, plus general game structure that would make sense over many other base repertoires of semantic actions.

More technically, in standard predicate logic, we take the meanings of the usual logical constants (propositional connectives and quantifiers) as fixed, and let the denotations of the other expressions vary over individual domains and predicates. But in the new light, quantifier symbols are also variable parts, that can be replaced by any game expression. E.g., the first-order distribution law

$$\exists x\ (Px \vee Qx) \leftrightarrow \exists x\ Px\ \vee\ \exists x\ Qx$$

is not valid in this stronger sense, - as replacing '$\exists x$' by '$\forall x$' gives a counter-example. Thus, the standard laws of first-order logic now fall into at least three different levels, providing a fine-structure not normally observed:

a) *General laws of game algebra*

These hold under any substitution of concrete games for quantifier symbols and of concrete statements for atomic subformulas. This may be viewed as the game-theoretic core of first-order logic, which is *decidable* - because its parent system of dynamic game logic is.

b) *Special laws still exhibiting some general game-theoretic point*

An example is the just-mentioned *right-distribution* of the existential quantifier over disjunction. This is invalid in general game algebra - but it does hold for all games in a broad natural class, viz. the 'distributive games' where Verifier can force singleton sets of outcomes.

c) *Idiosyncracies of atomic games*

An example is the *idempotence* of both quantifier and atomic fact games: satisfying the equation $G;G = G$. It is the interplay of these three levels which causes the undecidability of standard first-order logic - and a game-theoretic 'deconstruction' may help us understand this phenomenon better.

5. Extensions

With basic sequential operations, logical evaluation games are fully general for game logic - at least, at the global level of players' powers (Note 13). How far does this observation generalize? Straightforward extensions include the use of 'idle games' $\top$**?**, or arbitrary test games. More challenging extensions come in different kinds. First, there are stronger sequential operations, found in evaluation games for fixed-point languages like the modal

μ-calculus. One example is the unbounded *game iteration* of [16, 17], where one of the players may open up to a finite number of new copies of the game. Its complete game algebra, over the above repertoire, is still open. [11] shows how arbitrary μ-calculus evaluation games can be encoded in dynamic game logic, which suggests that basic game algebra with analogues of Kleene iteration is a very powerful format. Fixed-point games crucially have *infinite runs* as bona fide outcomes, which is another challenge to our analysis.

Another extension are *parallel game compositions*, allowing simultaneous play of games. These have no evident counterpart in modal or first-order evaluation games, but they may for more complex logical languages. Again, some of these operations typically live in infinite games (cf. [1, 15]). In particular, the connection remains to be understood between the dynamic logic perspective of this paper and games for linear logic (but cf. [2]).

Finally, realistic game theory is replete with *non-determined games*, such as card games, or real-life decision problems - whose players need not know exactly where they are in an extensive game tree. Our general definitions extend to this case, but what about more specific logic games of this sort? A typical source of non-determinacy is *imperfect information* ([7]). This is precisely the point of the *IF-games* of [14]. Extending our result would also motivate the latter logic games as a 'universal format' for basic game theory allowing imperfect information. We include one foray in this direction as an Appendix, not to end with just an empty litany of good intentions.

6. Coda on product games

We present one foray into parallel game operations - cf. [1, 2] for a systematic take via linear logic. The best-known structures in much of game theory are not the extensive trees of our paper, but strategic matrix games

		player **1**	
		a	b
player **2**	c	1	2
	d	3	4

These represent two players moving in parallel, with four possible outcomes. Occasionally, similar parallel phenomena occur in logic. An example is so-called 'branching quantification'. E.g., a two-dimensional pattern like

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

lets choices for prefixes take place independently - bringing together the results only at the end to evaluate the matrix assertion *Rxyzu*. Such games involve a mild form of imperfect information: ignorance of others' moves played at the same time. We define this game operation more generally as a

$$\textit{product}\ \ G \times H$$

whose runs are pairs of separate runs for G, H with the product of their end states as the total end state. In terms of powers, this works out as follows:

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

This equivalence fails if players have access to earlier moves in both games - which gives a much richer space of strategies in the product game. (Note 14.) Players' powers in games $G\times H$ are no longer determined, but they still satisfy conditions C1, C2 of Section 2.1. There is even an analogue of 3.1:

Fact Monotonicity and Consistency characterize powers in product games.

Proof [7] shows that these conditions are necessary and sufficient for players' powers in two-move imperfect information games. But in fact, the representation in the cited paper produces product games of the above sort! □

If outcomes are to be unique, as in matrix games, Consistency must be strengthened to ensure singleton intersections between powers for the players. One can think of dropping Determinacy as providing much greater flexibility in modelling behaviour of partially interacting agents. (Note 15.)

Next, as to *game algebra* over power models, we note some valid laws.

$$\begin{array}{lcl} A\times(B\cup C) & = & (A\times B)\cup(A\times C) \\ (A\cup B)\times C & = & (A\times C)\cup(B\times C) \\ (A\times B)^{d} & = & A^{d}\times B^{d} \end{array}$$

These laws compute all powers for players in direct products of finite games. There seem to be no significant interaction laws between the product $\times$ and sequential composition **;** - which reflects the complexity of iterated games.

Open question Axiomatize the complete game algebra with product.

Again, this game algebra lives inside a richer language with a game forcing modality. To state the minimum of product structure, one needs to assume some kind of combination predicate Cx, yz on states: 'x is a pair of y, z'. This supports some auxiliary product modalities for pairs of states:

$$\begin{array}{lll} \mathbf{M},\, s \models \Diamond_{\cdot}\boldsymbol{\phi\psi} & \text{iff} & \exists y,\, z\text{:}\ C\ s,\, yz \wedge \mathbf{M},\, y \models \boldsymbol{\phi} \wedge \mathbf{M},\, z \models \boldsymbol{\psi} \\ \mathbf{M},\, s \models \Diamond_{l}\boldsymbol{\phi} & \text{iff} & \exists y,\, z\text{:}\ C\ y,\, sz \wedge \mathbf{M},\, y \models \boldsymbol{\phi} \wedge \mathbf{M},\, z \models \top \\ \mathbf{M},\, s \models \Diamond_{r}\boldsymbol{\phi} & \text{iff} & \exists y,\, z\text{:}\ C\ y,\, zs \wedge \mathbf{M},\, y \models \boldsymbol{\phi} \wedge \mathbf{M},\, z \models \top \end{array}$$

Lacking determinacy, dynamic game logic needs modal operators $\{G,\, \boldsymbol{i}\}\boldsymbol{\phi}$ with explicit player marking. Its valid principles will include, e.g.,

$$\begin{array}{lll} \{G,\boldsymbol{i}\}\boldsymbol{\phi} \wedge \{H,\, \boldsymbol{i}\}\boldsymbol{\psi} & \rightarrow & \{G\times H,\, \boldsymbol{i}\}\Diamond_{\cdot}\boldsymbol{\phi\psi} \\ \{G\times H,\, \boldsymbol{i}\}\boldsymbol{\phi} & \rightarrow & \{G,\, \boldsymbol{i}\}\Diamond_{l}\boldsymbol{\phi} \wedge \{H,\, \boldsymbol{i}\}\Diamond_{r}\boldsymbol{\phi} \end{array}$$

This is not an automatic 'reduction' of the product modality, like in earlier cases. We do not have an equivalence like $\{G\times H\}\boldsymbol{\phi} \leftrightarrow \{G\}\Diamond_{l}\boldsymbol{\phi}\wedge\{H\}\Diamond_{r}\boldsymbol{\phi}$. Derivations of our algebraic laws are not obvious, nor is completeness.

Finally, consider logic games as a candidate for representing general product games. What would it mean to play say an evaluation game $\boldsymbol{\phi} \times \boldsymbol{\psi}$? Consider the above branching quantifier game. IF logic ([14]) is a generalized first-order logic allowing for this type of meaning, via a 'slash formula':

$$\forall x \exists y \forall z\ /\{\boldsymbol{x},\, \boldsymbol{y}\}\ \exists u\ /\{\boldsymbol{x},\, \boldsymbol{y}\}\ Rxyzu$$

which suppresses all information flow between the two prefixes. (Note 16.) In our game-algebraic language, this would be written as follows:

$$((\forall x;\exists y)\times(\forall z;\exists u))\ ;\ Rxyzu\mathbf{?}$$

Like standard first-order logic, IF logic is a mixture of general game algebra and special facts about semantic procedures. Game-algebraic laws have IF-instances that allow one to manipulate quantifier prefixes, such as

$$(\forall x;\exists y)\times((\forall z;\forall u)\cup(\exists v;\exists u)) = ((\forall x;\exists y)\times(\forall z;\forall u)) \cup ((\forall x;\exists y)\times(\exists v;\exists u)$$

Also, valid principles of IF logic may be seen to be algebraic validities. E.g., the following nice quantifier exchange law is valid ([5]):

$$\forall x \exists y/\boldsymbol{x}\ Rxy \leftrightarrow \exists y \forall x/\boldsymbol{y}\ Rxy$$

In game-algebraic terms, this says that

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

This principle did not occur in our earlier list. Its core $G \times H = H \times G$ amounts to commutativity of state product: the order of composition in pairs is irrelevant. But also, IF logic can detect invalid algebraic principles. Here is an example refutable in general games:

$$(A \times B) \; ; \; C = (A;C) \times (B;C)$$

An IF-counterexample is $\exists x \forall y/\boldsymbol{x} \; Rxy$, whose evaluation game is not equivalent to that for $\exists x \; Rxy \times \forall y \; Rxy$. In all, product games can be analyzed like the sequential ones we had before - but a complete extension of the previous completeness and representation results is by no means obvious.

7. Notes

1) Evaluation games for fixed-point languages may involve infinite runs. This happens with games for the modal $\boldsymbol{\mu}$-calculus that decompose fixed-point operators (cf. [18]). Likewise, Ehrenfeucht-Fraïssé games can have infinite runs, with bisimulations or potential isomorphisms encoding winning strategies for the analogy player. This requires analysis with runs, rather than end states, as outcomes of a game.

2) There are subtle differences here. The computational complexity of first-order model checking is PSPACE in the size of the model and the input formula. But the same task for modal logic takes only PTIME. This has to do with the size of the game trees. Nodes in modal games do not involve assignments s that can grow with the number of variables, and hence the size of the formula.

3) Another illustration of this dual perspective are *graph games*, where players alternate in moving a pebble along the edges of some graph.

4) There are also natural non-sequential *parallel game operations*, of which we will consider one example at the end. These arise only in 'non-standard' first-order logics, as in Hintikka & Sandu's game-theoretical semantics (cf. Section 6).

5) Test games have some peculiarities. E.g., the general test game $(\neg\boldsymbol{\phi})\mathbf{?}$ is not the same as the game dual $(\boldsymbol{\phi}\mathbf{?})^{\boldsymbol{d}}$. If $\boldsymbol{\phi}$ is true, in the former game, **V** is to move, and ends in the first state (and dead-locks) - while in the latter, **F** is to move. This difference carries over to the associated forcing relations. The success condition in the former game for **V** is $\neg\boldsymbol{\phi} \wedge x \in Y$, whereas in the latter, it is $\neg\boldsymbol{\phi} \vee x \in Y$. We will use test games only sparingly.

6) The dynamic game language also has a characteristic game bisimulation for object-to-set transition relations, explaining when two game boards support the same game assertions ([4, 9, 17]).

7) De Morgan algebra consists of the standard axioms for a distributive lattice plus an idempotent negation ($x^{dd} = x$):

$$
\begin{array}{rcl@{\qquad\qquad}rcl}
x\cup x & = & x & x\cap x & = & x \\
x\cup y & = & y\cup x & x\cap y & = & y\cap x \\
x\cup(y\cup z) & = & (x\cup y)\cup z & x\cap(y\cap z) & = & (x\cap y)\cap z \\
x\cup(y\cap z) & = & (x\cup y)\cap(x\cup z) & x\cap(y\cup z) & = & (x\cap y)\cup(x\cap z) \\
(x\cup y)^d & = & x^d\cap y^d & (x\cap y)^d & = & x^d\cup y^d
\end{array}
$$

This is the working part of Boolean Algebra, without any special laws for **0**, **1**.

8) Dynamic predicate logic is an assignment-change semantics for first-order logic based on dynamic logic of programs. Cf. [3] for a technical exposition, including a short proof of the mentioned result from [20]. Game-theoretical semantics for first-order logic is similar in spirit, but with transitions from input assignments to output sets of assignments. The precise connection might run along the lines of [10], which lifts dynamic semantics to a set change version.

9) This suggests genuine extra expressive power of bounded quantifiers in game semantics - which remains hidden in standard logic.

10) One can also view this final step as an independent representation of game models **M**. We take a new model whose states are pairs of objects $(s,\ X)$, setting $\boldsymbol{\rho}_g\ (s,\ X)\ A$ (with A a family of such pairs) iff the original forcing relation $\boldsymbol{\rho}_g$ holds in **M** between s and the set of left-projections of all pairs in A. The result is a game-bisimulation between the two models.

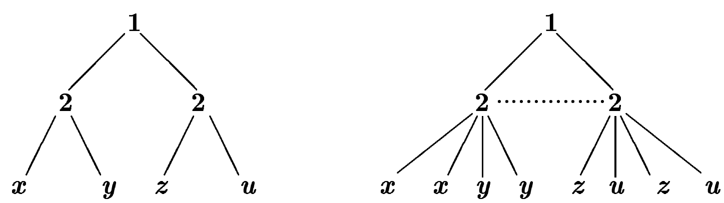
11) To mimick modal evaluation operator by operator, another trick is needed. One relates modal games over models **M** to first-order games over the same model, with states $\{(x,\ s),\ (y,\ t)\}$, mapping the latter onto their x-value s in the modal model. One then simulates a modality $\Diamond$ by means of $\exists y$; Rxy ; $\exists x$; $x{=}y$; $\exists y$; $\top$.

12) Games also look differently at standard first-order *syntax*. For, the natural class of expressions to interpret is larger than the usual 'well-formed formulas'. It also includes the latter's combinations with game operations, plus free-standing operators. Thus, the following is a perfectly fine game expression: Px ; $\exists x$; $(Rxy\vee Py)$; $(\forall x\wedge\exists z)$. This provides independent denotations for a much larger class of discourse expressions. It also suggests that logical deduction might work with other expressions than just well-formed formulas. Similar points occur in dynamic semantics of natural language under the slogan of 'emancipation of syntax'.

13) There may yet be alternative roads to validating the title of this paper. At the level of local actions, arbitrary finite games work directly as evaluation games! The relevant recipe works up the game tree. Translate outcomes into unique proposition letters. Write modalities $\langle a \rangle$ for available moves a to nodes already described by a formula - and put the disjunction of all these if player **1** is to move, and a conjunction, otherwise. The evaluation game for the resulting formula on the given game tree is essentially that game tree itself.

14) [9] has more details on product games that do allow interaction - including connections with unbounded repetition games like infinite Prisoner's Dilemma, which arise as DGL game iterations of products $G \times H$.

15) The result also says that perfect information games can be modeled as product games, at least qua powers. Here is one, with player **2** having four actions:



16) IF syntax is much richer in general. E.g., a slash formula $\forall x \exists y \forall z\ \exists u/\boldsymbol{x}\ Rxyzu$ allows Verifier in the second prefix $\forall z \exists u$ access to what she has played in the first. Its evaluation would be more like a product game allowing limited interaction.

References

[1] S. Abramsky, 1996, 'Semantics of Interaction: an Introduction to Game Semantics', in P. Dybjer & A. Pitts, eds., *Proceedings 1996 CLiCS Summer School, Isaac Newton Institute*, Cambridge University Press, Cambridge, 1-31.

[2] A. Abramsky, 1997, *Games in the Semantics of Programming Languages*, invited lecture, 11th Amsterdam Colloquium, Amsterdam.

[3] J. van Benthem, 1996, *Exploring Logical Dynamics*, CSLI Publications, Stanford.

[4] J. van Benthem, 1999 onward, *Logic in Games*, electronic lecture notes, ILLC Amsterdam & CSLI Stanford (printed version 2001).

[5] J. van Benthem, 2000A, 'Hintikka Self-Applied', to appear in L. Hahn, ed., *Jaakko Hintikka*, Library of Living Philosophers, Southern Illinois University.

[6] J. van Benthem, 2000B, 'Logical Evaluation Games are Complete for the Game Algebra of Forcing Relations', ILLC Amsterdam, the original version of this paper.

[7] J. van Benthem, 2001, 'Games in Dynamic Epistemic Logic', *Bulletin of Economic Research* 53:4, 219-248.

[8] J. van Benthem, 2002A, 'Extensive Games as Process Models', *Journal of Logic, Language and Information*, to appear.

[9] J. van Benthem, 2002B, 'Notes on Product Games', manuscript, ILLC Amsterdam.

[10] M. van den Berg, 1996, *The Dynamics of Nominal Anaphora*, dissertation DS 1996-03, ILLC Amsterdam.

[11] D. Berwanger & E. Grädel, 2002, 'The Variable Hierarchy of the μ-Calculus', Department of Informatics, RWTH Aachen.

[12] V. Goranko, 2000, 'The Basic Algebra of Game Equivalences', Preprint 2000-12, ILLC Amsterdam.

[13] J. Hintikka, 1973, *Logic, Language Games, and Information*, Clarendon Press, Oxford.

[14] J. Hintikka & G. Sandu, 1997, 'Game-Theoretical Semantics', in J. van Benthem & A. ter Meulen, eds., *Handbook of Logic and Language*, Elsevier, Amsterdam, 361-410.

[15] W. Hodges, 1998, *An Invitation to Logical Games*, lecture notes, department of mathematics, Queen Mary's College, London.

[16] R. Parikh, 1985, 'The Logic of Games and its Applications', *Annals of Discrete Mathematics 24*, 111-140.

[17] M. Pauly, 2001, *Logic for Social Software*, dissertation DS 2001-10, ILLC Amsterdam.

[18] C. Stirling, 1999, 'Bisimulation, Modal Logic, and Model Checking Games', *Logic Journal of the IGPL 7*, 103-124.

[19] Y. Venema, 2001, 'Representation of Game Algebras', manuscript, ILLC Amsterdam. To appear in *Studia Logica*.

[20] A. Visser, 1995, 'Relational Validity and Dynamic Predicate Logic', preprint 144, Logic Group, Department of Philosophy, Utrecht University.

Johan van Benthem
University of Amsterdam & Stanford University)
Plantage Muidergracht 24
1018 TV Amsterdam
The Netherlands
johan@science.uva.nl