Journal of Logic, Language and Information **11:** 289–313, 2002.

Extensive Games as Process Models

JOHAN VAN BENTHEM
ILLC, University of Amsterdam, Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands
E-mail: johan@science.uva.nl

(Received 24 January 2002)

Abstract. We analyze extensive games as interactive process models, using modal languages plus matching notions of bisimulation as varieties of game equivalences. Our technical results show how to fit existing modal notions into this new setting.

Key words: Bisimulation, extensive games, modal logic

1. Games as Interactive Processes: Actions and Outcomes

1.1. Games as Process Models

An extensive game is a mathematical tree decorated as follows:

M = (NODES, MOVES, PLAYERS, **turn**, **end**, VAL),

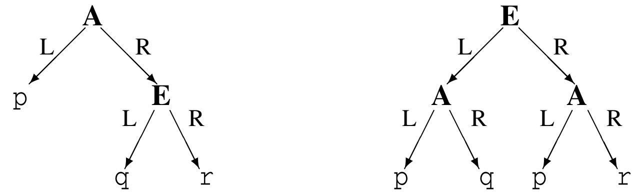
where non-final nodes are turns for unique players, with outgoing transitions as moves for that player. Final nodes have no outgoing moves. Technically, such structures are models for a poly-modal logic, with the nodes as states, moves as binary transition relations, and special proposition letters $\mathbf{turn}_i$ marking turns for player $\boldsymbol{i}$, and **end** marking final points. The valuation VAL may also interpret other relevant game-internal predicates at nodes, such as utility values for players or more external properties of game states. Thus, games are processes with two or more interacting agents – and analogies are worth exploring between game theory and process logics. In this paper, we discuss issues of "description level." What are appropriate formal languages for games, and in tandem with this: what are appropriate semantic simulations – answering the fundamental question

When are two games the same?

In looking at these issues, we shall mostly deal with *finite two-player games only*. But we are more liberal in another sense. Most results that follow hold for arbitrary graphs, not just trees. This makes sense. There are two intuitive ways of interpreting the usual game diagrams. One is as a *tree of all possible histories* for the game, the other as an abstract *state automaton* telling us what states and transitions are possible. On the latter view, we can liberalize the definition of a game, allowing cycles that generate infinite runs in the tree of all real plays.

1.2. GAME EQUIVALENCE: ACTION AND POWER LEVELS

As a warm-up example, consider the following two games:



Are these the same? The answer depends on our level of interest:

(a) *If we focus on turns and moves, then the games are not equivalent.*

For they differ in "protocol" (who gets to play first) and in choice structure. This is indeed a natural level for looking at game, involving local *actions* and choices. Later on, modal *bisimulations* will define this comparison more precisely.

But one might also want to call these games equivalent in another sense – if only, because they represent evaluation games for the two sides of the logical law

$$p \wedge (q \vee r) \leftrightarrow (p \wedge q) \vee (p \wedge r). \qquad \text{Distribution}$$

The proper sense of equivalence then looks at achievable outcomes only:

(b) *If we focus on outcomes only, then the games are equivalent.*

The reason is players can force the same sets of outcomes across these games:

A can force the game to end in the sets $\{p\}, \{q, r\}$.

E can force the game to end in the sets $\{p, q\}, \{p, r\}$.

Here "forcing" refers to sets of outcomes ("powers") guaranteed by strategies. In the left-hand tree, **A** has 2 strategies, and so does **E**, yielding the listed sets. In the right-hand tree, **E** has again 2 strategies, while **A** has 4: LL, LR, RL and RR. Of these, LL yields the outcome set $\{p\}$, and RR yields $\{q, r\}$. But LR, RL guarantee only supersets $\{p, r\}, \{q, p\}$ of $\{p\}$: i.e., weaker powers. Thus the same "control" results in both games. More generally, at an input-output level, Distribution switches the scheduling of a game without affecting players' powers.

Thus, game equivalences come in varieties depending on one's level of interest: coarser or finer – illustrating that universal philosophical insight first enunciated clearly during the Lewinsky hearings:

Clinton's Principle It all depends on what you mean by "*is*."

1.3. PROCESS LOGICS AND PROCESS EQUIVALENCE

The same ladder of perspectives is known from process theories, running from purely observational equivalence to more internal simulations. Here is a well-known example. Consider the following machine pictures – or if you wish, two rather simple games with just one player:

Do these machines represent the same process?

Both produce the same observable finite traces: {*ab*, *ac*}, even though the first machine starts deterministically, and the other with an internal choice. If one cares for input-output behaviour only, then, the two machines are the same under *finite trace equivalence*. But normally, one is also interested in some internal workings of a machine, including the choices "en route." This is measured by a finer structural comparison called *bisimulation* – and indeed, the two machines are not bisimilar.

Thus there is a hierarchy of process equivalences, from coarser ones like finite trace equivalence to finer ones such as bisimulation, to be defined below. Moreover, the ladder has a syntactic counterpart: the finer the structural equivalence, the more expressive the associated logical language. We will address such definability issues for games in this paper, disregarding logical issues of axiomatization or complexity.

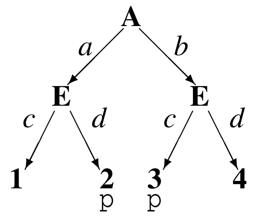
There is much more to games and processes than this (van Benthem, 2000). Many process calculi coexist in computer science, including Hoare–Dijkstra calculus, modal and dynamic logic, process algebra, temporal logic, and linear logic. More toward AI, logics for multi-agent systems introduce knowledge, belief, intentions and desires of agents. All this additional structure is relevant to extensive games, where players deliberate, plan and interact over time. We will discuss two more realistic topics of this kind as they affect language design and game equivalence: *imperfect information*, and *preferences/expectations*. These involve enrichments of the above models to combined modal logics of various sorts. One challenging further test for the approach would be *infinite games*, which suggest a switch from a modal to a temporal logic making branches of game trees primary objects in their own right.

2. The Action Level: Modal and Dynamic Logic

At the action level, extensive games can be described by standard modal languages, making modal bisimulation the preferred game equivalence. Thus, modal and dynamic logics as they stand are already interesting calculi of games and strategies! We will demonstrate this role with a sequence of examples.

2.1. Modal-Dynamic Properties of Games

Consider a simple 2-step game like the following, with players **A**, **E**.



Player **E** clearly has a strategy making sure that a state is reached where p holds. This can be expressed by the following modal formula which is true at the root:

$[a \cup b]\langle c \cup d\rangle \mathtt{p}$,

indicating that for every execution of the choice action $a \cup b$, there exists an execution of the choice action $c \cup d$ ending in a p-state. Thus, modal operator sequences can state the existence of strategies. But the language can also deal with strategies more explicitly. The latter may be viewed as partial transition functions defined on players' turns, given via a bunch of conditional instructions of the form

"*if she plays this, then I play that*."

More generally, strategies are partial transition relations. The latter involve basic moves, tests for conditions, plus some compounding – and hence they are definable in a dynamic logic over these models which starts with the basic moves, and adds the relational operations of

composition **;**, choice $\cup$, iteration *****, and test (ϕ)**?**

Thus, a strategy is like a program, which has to act at turn nodes for the relevant player. For instance, in this language, we can state that the final state in a 2-player game reached by playing strategies σ, τ against each other has some property p:

$[((\mathrm{turn}_{\mathbf{E}})\mathbf{?}\ \mathbf{;}\ \sigma) \cup (\mathrm{turn}_{\mathbf{A}})\mathbf{?}\ \mathbf{;}\ \tau))^*]\ (\mathbf{end} \rightarrow \mathtt{p})$.

In other words, dynamic logic encodes explicit reasoning about strategies and their final effects. This can even be extended. First, we can equally well describe effects at intermediate nodes of strategies, dropping the antecedent **end** →. Next, the relational setting is a plus. A standard game-theoretic strategy was a function giving a unique response at every turn for its player. But in actual deliberation about actions, total recipes are scarce. We usually have some vaguer plan, which may present us at our turns with more alternatives, placing only some constraint on our actions. Realistic opponents play such plans against each other, resulting in a *set of* possible outcome states: restricted, but not uniquely defined by their plans. This situation is covered automatically by the above formulas and their logic.

2.2. Dynamic Logic as Strategy Calculus

Dynamic modal logic can also talk about strategies running over only part of the game tree, and their *combination*. Thus one gets a *calculus of strategies* for free! The following modal operator describes the effect of partial strategy σ for player **E** running until the first game states where it is no longer defined:

$$\{\sigma, \mathbf{E}\}\phi \quad [((\text{turn}_{\mathbf{E}})\,\mathbf{?}\,\mathbf{;}\,\sigma) \cup (\text{turn}_{\mathbf{A}})\,\mathbf{?}\,\mathbf{;}\,A))^{*}]\phi.$$

Here A is the union of all available moves for player $\boldsymbol{j}$. Likewise, in what follows, E is the union of all moves for player $\boldsymbol{i}$.

A basic operation on strategies is *union*, allowing all possible moves according to both. Union plays two roles. On the one hand, it merges two "plans" constraining players' moves into a common weakening. Then we can do plan calculus like program calculus in dynamic logic, including reasoning about effects.

Here are two examples:

$$\{\sigma, \mathbf{E}\}\phi \wedge \{\sigma, \mathbf{E}\}\psi \;\rightarrow\; \{\sigma, \mathbf{E}\}(\phi \wedge \psi) \text{ is valid,}$$

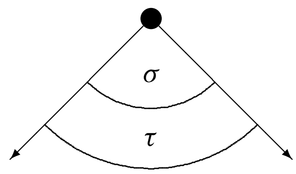
$$\{\sigma, \mathbf{E}\}\phi \wedge \{\tau, \mathbf{E}\}\varphi \;\rightarrow\; \{\sigma \cup \tau, \mathbf{E}\}\phi \quad \text{ is invalid.}$$

But union also has another guise: as sequential *composition* of successive strategies – at least, if we make the assumption that

σ, τ are *disjoint*: never defined at the same turns.

Then the following principle of strategy calculus becomes valid:

$$\{\sigma, \mathbf{E}\}\{\tau, \mathbf{E}\}\phi \rightarrow \{\sigma \cup \tau, \mathbf{E}\}\phi.$$



2.3. Fixed-Point Languages for Equilibrium Notions

A good test for proposed languages checks their expressive power for rendering proofs of significant results. Recall the following basic result:

ZERMELO'S THEOREM. *Every finite zero-sum game is* determined.

I.e., one of the two players in such a finite game always has a winning strategy. Here is the heart of the matter. Starting from atomic predicates $\mathbf{win}_i$ at end nodes indicating which player has won, we define predicates WIN_i ("player i has a winning strategy at the current node") through the following recursion:

$$\text{WIN}_i \leftrightarrow (\mathbf{end} \wedge \mathbf{win}_i) \vee (\mathbf{turn}_i \wedge \langle E\rangle \text{WIN}_i) \vee (\mathbf{turn}_j \wedge [A]\text{WIN}_i).$$

Note that the given schema amounts to an inductive definition for the predicate WIN_i, obtained by a *smallest fixed-point* schema

$$\text{WIN}_i = \mu \mathtt{p}\bullet (\mathbf{end} \wedge \mathbf{win}_i) \vee (\mathbf{turn}_i \wedge \langle E\rangle \mathtt{p}) \vee (\mathbf{turn}_j \wedge [A]\mathtt{p}).$$

The right-hand side is not a formula of dynamic logic proper, but it does belong to the modal *μ-calculus* which allows *fixed-point definitions* – provided the defining schema only has syntactically positive occurrences of the atomic predicate p. Thus, the μ-calculus is a good game logic, too. Using it, one can vary on the above recursive schema. E.g.,

$$\{\boldsymbol{i}\}\phi = \mu \mathtt{p}\bullet (\mathbf{end} \wedge \phi) \vee (\mathbf{turn}_i \wedge \langle E\rangle \mathtt{p}) \vee (\mathbf{turn}_j \wedge [A]\mathtt{p})$$

defines the existence of a strategy for ***i*** guaranteeing a set of outcomes where the proposition ϕ holds. And the following recursion

$$\text{COOP}\ \phi \leftrightarrow \mu \mathtt{p}\bullet (\mathbf{end} \wedge \phi) \vee (\mathbf{turn}_i \wedge \langle E\rangle \mathtt{p}) \vee (\mathbf{turn}_j \wedge \langle A\rangle \mathtt{p})$$

would define the existence of a cooperative outcome ϕ. The latter can still be defined explicitly inside dynamic logic, using the formula

$$\langle ((\text{turn}_i)\mathbf{?}\ ;\ E) \cup (\text{turn}_j)\mathbf{?}\ ;\ A))^*\rangle\ (\mathbf{end} \wedge \phi).$$

Modal fixed-point definitions reflect the *equilibrium* character of many game-theoretic notions, reached through some process of iteration.

Thus the basic principles of dynamic logic provide a formalization of an elementary part of game theory, namely, game forms, actions and strategies.

2.4. Bisimulation as Game Equivalence

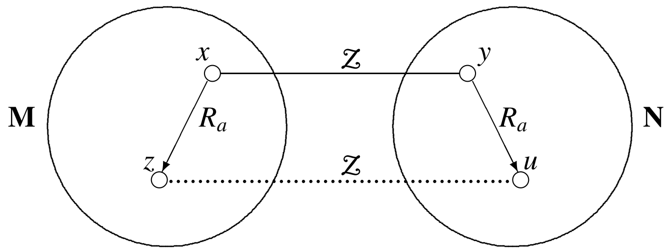
A language represents a level of detail for talking about properties of structures. Modal languages discuss games locally, at the level of individual actions and progression through successive states. But one can also look at such levels in more structural terms, looking for a notion of invariance between different presentations of the same game - just as different "machines" can be implementations of the same process. The crucial equivalence for the modal language is as follows:

DEFINITION. A *bisimulation* is any binary relation $\mathcal{Z}$ between states of two graphs **M**, **N** with labeled edges (i.e., binary transition relations R_a), such that, whenever $x\mathcal{Z}y$, then we have (1) atomic harmony, and (2) zigzag clauses for all a:

(1) x, y verify the same proposition letters,

(2a) if xR_az, then there exists u in **N** s.t. yR_au and $z\mathcal{Z}u$,

(2b) vice versa.



Bisimulation is the typical "action-similarity" for processes taking care of equivalent local properties as well as options available at any stage. It can be used to *contract* given machines to smallest ones implementing the same process, but also to *unravel* a diagram to the full tree of possible execution sequences.

Bisimulation preserves the truth of modal and dynamic formulas, up to the full μ-calculus – and there are converses, too. Three key results establish the link (van Benthem, 1996, 1997; Barwise et al., 1997):

THEOREM. *For graphs **M**, **N** with nodes s, t, condition* (a) *implies condition* (b)*:*

(a) *there is a bisimulation $\mathcal{Z}$ between **M**, **N** with $s\mathcal{Z}t$,*
(b) ***M**, s and **N**, t satisfy the same formulas of the μ-calculus.*

In particular, modal or dynamic formulas are invariant under bisimulations.

THEOREM. *For finite graphs **M**, **N** and nodes s,t, the following are equivalent:*

(a) ***M**, s and **N**, t satisfy the same modal formulas,*
(b) *there is a bisimulation $\mathcal{Z}$ between **M**, **N** with $s\mathcal{Z}t$.*

This partial converse says that the basic modal language and the similarity relation match on finite models. The third result says that at the same level of description, the dynamic language even provides complete descriptions for any finite graph:

THEOREM. *For each finite graph **M** and state s, there exists a dynamic logic formula $\beta_{\mathbf{M},s}$ such that the following are equivalent for all graphs **N**, t:*

(a) *N, t* $\models \beta_{M,s}$,
(b) ***N**, t is bisimilar to **M**, s.*

Proof. Any finite model **M**, *s* falls into maximal zones of states satisfying the same modal formulas in our base language. The resulting partition is definable:

CLAIM 1. There exists a finite set of modal formulas ϕ_i ($1 \leq \boldsymbol{i} \leq$k) s.t.

(a) each state satisfies one and only one of them,
(b) states satisfying the same formula ϕ_i agree on all modal formulas.

To see this, take any state *s* in the model. For any state *t* which does not satisfy the same modal formulas take a "difference formula" $\delta^{s,t}$ true in *s* and false in *t*. The conjunction of all $\delta^{s,t}$ is a formula ϕ_i true only in *s* and states with the same modal theory. Without loss of generality, we assume the ϕ_i list all information about the *proposition letters* that are true and false throughout their partition zone. We also make a quick useful observation about action links between these zones:

\# If any state satisfying ϕ_i is R_a-linked to a state satisfying ϕ_j,
then all states satisfying ϕ_i also satisfy $\langle a \rangle \phi_j$

Next take the following description $\beta_{\mathbf{M},s}$ of **M**, *s*:

(a) all propositional literals true at *s* plus the unique ϕ_i true at **M**, *s*,
(b) the "universal modality" $[(E \cup A)^*]$ prefixed to the conjunction of

(b1) the disjunction of all ϕ_i, plus all $\neg(\phi_i \wedge \phi_j)$ ($\boldsymbol{i \neq j}$),
(b2) all implications $\phi_i \rightarrow \langle a \rangle \phi_j$ for which situation # occurs,
(b3) all implications $\phi_i \rightarrow [a] \vee \phi_j$ where the disjunction runs over all situations listed in the previous clause.

CLAIM 2. **M**, $s \models \beta_{\mathbf{M},s}$.

CLAIM 3. If **N**, $t \models \beta_{\mathbf{M},s}$, then there is a bisimulation between **N**, *t* and **M**, *s*.

Let **N**, *t* be any model for $\beta_{\mathbf{M},s}$. The ϕ_i partition **N** into disjoint zones of states satisfying them. Link all states in such a zone to all states satisfying ϕ_i in **M**. In particular, *t* gets connected to *s*. We check that this is a bisimulation. The atomic clause is clear. The zigzag clauses follow from the given description. (a) Any R_a-successor step in **M** has been encoded in a formula $\phi_i \rightarrow \langle a \rangle \phi_j$ which holds everywhere in **N**, producing the required successor there. (b) Conversely, if there is no R_a-successor in **M**, this shows up in the limitative formula $\phi_i \rightarrow [a] \vee \phi_j$, which also holds in **N**, preventing "excess" successors there. □

The second and third theorems hold for arbitrary graphs provided one increases the expressive power of the basic modal language drastically, allowing the formation of arbitrary *infinite* conjunctions and disjunctions in its construction rules.

On trees, bisimulations identify very little, and are close to *graph isomorphism*. But our analysis works just as well for broader games as graph automata.

2.5. Further Modal Languages

The preceding results are typical of a genre. Similar definability and expressiveness theorems can be proved for many notions of process equivalence with richer modal languages (van Benthem, 1996) – and the same is true for game equivalences. Thus, game structure can be described either via suitable semantic equivalences between representations, or more syntactically, in terms of defining "game formulas" in corresponding formal languages (Bonanno, 1993; van Benthem, 1999).

In particular, it is useful to extend modal languages with *past* operators, looking back at the history of the game prior to the current state. Most generally, take the

relational converse $A^{\cup}$

on all complex actions A so far, and go back in the tree by inverted moves. Iterating, this yields modal operators with universal access to any node in the tree, reachable from the current state by backing up and then moving down. This allows for counterfactual reasoning about nodes that never occurred. Bisimulations for this language simply add zigzag clauses for *incoming* moves at a state to the earlier ones for outgoing moves. The above results go through with the same proofs. Here is an additional observation about expressive power (Rodenhäuser, 2001):

Fact. All strategies are definable in a deterministic finite game tree using the dynamic modal language with converse operators on actions.

Proof. If moves are deterministic partial functions, this language can define every node s in a game tree uniquely by a formula δ_s enumerating all moves in its unique history from the root. Then, every transition in the tree of type R_a from a node s to a node t can be defined as follows in the dynamic language:

$(\delta_s)\mathbf{?}\ \mathbf{;}\ a$

Any relation is definable by some finite union of these. In particular, then, any strategy for players becomes definable in our language. □

Defining strategies via backward-looking node definitions, makes them dependent only on the history of the game so far. But the earlier "backward induction" strategies defined their moves by looking at the future of the current node. This takes us back to the preceding sections. Consider the *bisimulation contraction* of a finite game model, i.e., the smallest graph bisimilar to it. Our original forward-looking language is fully expressive for strategies in such game models:

Fact. In finite bisimulation contractions of a game, every strategy can be defined in the dynamic modal language as it stands.

Proof. It is well known that every state in such a contraction is uniquely definable by some standard modal formula (Blackburn et al., 2001). Now, just enumerate all state transitions in the strategy, using a finite union of modal action expressions

$$(\phi)\mathbf{?}\ \mathbf{;}\ a\ \mathbf{;}\ (\psi)\mathbf{?}$$

where ϕ identifies the start of the move, and ψ its end point. □

Adding past operators to our language is a more ambitious way of talking about the same games. But the above approach also works when we change the games themselves. A case in point would be the modal study of games with *simultaneous moves* with turns defining active *groups* of players.

3. The Power Level: Games as Input-Output Relations

Much coarser levels of game description can be important, too. E.g., *strategic forms* in game theory just list all possible strategies and outcomes for strategy pairs in a matrix. For the two games of Section 1.2, e.g., these matrices are as follows:

E \ A	L	R
L	p	q
R	p	r

E \ A	LL	LR	RL	RR
L	p	p	q	q
R	p	r	p	r

This only records global actions and outcomes. A more economic description looks directly at players' powers for forcing the game to end in a certain set of outcomes. We will define this, find some basic properties, and show how generalized modal languages can deal with this game level, including an appropriate bisimulation.

3.1. FORCING AND POWERS

We define the following forcing relations:

$\rho^{i}_{G}s, X$ *player* ***i*** *has a strategy for playing game G from state s onward whose resulting states are always in the set X.*

Forcing relations at the root of a game encode players' powers for determining the outcomes against arbitrary counterplay by their partners. Mathematically, they are generalized accessibility relations in an interactive process, relating states to sets of states: leaving the state-to-state transition relations of the above game models. We will work with this input-output view of game structure henceforth. Powers forced by strategies are easy to compute, witness the example in Section 1.2.

3.2. A Simple Representation of Powers

Though not strictly necessary for what follows, the following will help understand the current game level. For simplicity, consider games between two players **A**, **E**. Here are some constraints. Power relations are evidently *closed under supersets*:

C1 *if* ρ_G^i *s, Y and* $Y \subseteq Z$*, then* ρ_G^i *s, Z.*

Another obvious constraint is *consistency*. Players cannot force the game into disjoint sets of outcomes, or a contradiction would result:

C2 *if* ρ_G^A *s, Y and* ρ_G^E *s, Z, then Y, Z overlap*.

Moreover, Zermelo's Theorem said that all *finite* 2-player games are *determined*: for any winning convention, one of the two players must have a winning strategy. This is really a condition of *completeness*. Let *S* be the total set of outcome states:

C3 *if not* ρ_G^A *s, Y, then* ρ_G^E *s, S-Z;*
and the same holds for **E** *vis-à-vis* **A**.

Conversely, these conditions are also *all* that must hold, witness this result:

PROPOSITION. *Any two families* F_1, F_2 *of subsets of a set S satisfying conditions* C1*,* C2*,* C3 *are the root powers in some two-step game.*

Proof. Start with player **A** and let him choose between successors corresponding to the sets in F_1. At these nodes, player **E** gets to move, and can pick any member of that set. Clearly, player **A** has the powers specified in F_1. Now for player **E**. In the game just defined, she can force any set of outcomes that *overlaps with each set in* F_1. But by C2, C3, these are precisely the sets in her initial family F_2. E.g., if outcome set *U* overlaps with all sets in F_1, its complement *S-U* cannot be in that family, and so *U* itself must have been in F_2 by Completeness. □

With these constraints, in a two-player game, powers for one player automatically determine those for the other by C2, C3. Our result gives an outcome-level *normal form* for games, related (though not identical) to their usual strategic form. It has two moves, and it does not matter which player begins. This is like the distributive normal form of standard propositional logic. Indeed, the major Boolean operations form a *logical calculus of game equivalence* (van Benthem, 2001c) allowing us to reverse the order of players, and suppress repeated moves by them.

The above conditions on two families of sets collapse into the standard set-theoretic notion of an *ultrafilter* when we identify the two families, and have just one player. This limit behaviour shows we have a natural notion. The game-theoretic setting is an arena for systematic generalization of classical logical notions. One is now free to make separate stipulations for players, and relate these in various ways.

3.3. A Modal Forcing Language

There still is a matching modal language for games at this level, with proposition letters, Boolean operators, and this time, modal operators:

$$\mathbf{M}, s \models \{G, \boldsymbol{i}\}\phi \text{ iff } \textit{there exists a set X with } \rho_G^{\boldsymbol{i}}\ s, X \textit{ and } \forall s \in X\, \boldsymbol{M},\, s \models \phi.$$

In the context of a fixed game, we need not encode the first argument in the modality – writing just $\{\boldsymbol{i}\}\phi$. This is the so-called *neighbourhood semantics* for modal logic, taking states to sets of states. Its universal validities are all principles of the minimal modal logic except for distribution of $\{\boldsymbol{i}\}\phi$ over disjunctions. In particular, the above C1, C2, C3 return as principles of the logic:

if $\models \phi \to \psi$, then $\models \{G, \boldsymbol{i}\}\phi \to \{G, \boldsymbol{i}\}\psi$	upward monotonicity
$\{G, \mathbf{A}\}\phi \leftrightarrow \neg\{G, \mathbf{E}\}\neg\phi$	consistency + determinacy

Failure of distribution means that the following is not valid:

$$\{G, \boldsymbol{i}\}(\phi \vee \psi) \to \{G, \boldsymbol{i}\}\phi \vee \{G, \boldsymbol{i}\}\psi.$$

This is precisely the point of forcing. Other players' powers may keep us from determining outcomes uniquely. E.g., in the game of Section 1.2, player **A** can force {q, r}, but neither {q} nor {r} – as **E** has the decisive say in this.

3.4. Bisimulation, Invariance and Definability

Consider any game model **M** plus forcing relations as defined above. Here is the appropriate generalization of standard modal bisimulation.

DEFINITION. A *power bisimulation* between game models **M**, **N** is a relation $\mathcal{Z}$ between game states satisfying the following two conditions:

(1) if $x\mathcal{Z}y$, then x, y satisfy the same proposition letters,
(2a) for each $\boldsymbol{i}$, if $x\mathcal{Z}y$ and $\rho_{\mathbf{M}}^{\boldsymbol{i}}\ x, U$, then there exists a set V with $\rho_{\mathbf{N}}^{\boldsymbol{i}}\ y, V$ and $\forall v \in V\ \exists u \in U$: $u\mathcal{Z}v$,
(2b) vice versa.

This notion is natural. It has been proposed independently in concurrent dynamic logic (van Benthem et al., 1994), topological modal logics (Aiello et al., 2001), game logics for players' strategic powers (Parikh, 1985; Pauly, 2001), and co-algebra (Baltag, 2000). All results of Section 2 generalize:

THEOREM. *Formulas in the modal forcing language are invariant for power bisimulation.*

Proof. The inductive step for the forcing operator explains the above zigzag clauses. Consider two game models **M**, **N**. Suppose **M**, $s \models \{\mathbf{i}\}\phi$ and $s\ \mathcal{Z}t$. By the truth definition, there is a set U with $\rho^{i}_{\mathbf{M}}\ s, U$ and for all $u \in U$: **M**, $u \models \phi$. Now by the zigzag clause (2), there is a set V in **N** with $\rho^{i}_{\mathbf{N}}\ t, V$ and $\forall v \in V\ \exists u \in U$: $u\mathcal{Z}v$. Thus, every $v \in V$ is $\mathcal{Z}$-related to some $u \in U$, and by the inductive hypothesis: **N**, $v \models \phi$. But then, **N**, $t \models \{\boldsymbol{i}\}\phi$. □

THEOREM. *Finite models **M**, x and **N**, y satisfying the same forcing formulas have a power bisimulation $\mathcal{Z}$ between them with $x\mathcal{Z}y$.*

Proof. Define a relation $\mathcal{Z}$ between states in the models as follows:

$u\mathcal{Z}v$ iff **M**, u and **N**, v satisfy the same modal forcing formulas.

CLAIM 4. $\mathcal{Z}$ is a power bisimulation.

The atomic clause is clear from the definition. Now, suppose that sRt, while also, for some subset U of **M**, $\rho^{i}_{\mathbf{M}}\ s, U$. We need to find V with

$$\rho^{i}_{\mathbf{N}} t, V \quad \text{and} \quad \forall v \in V\ \exists u \in U : u\mathcal{Z}v.$$

Suppose that no such set exists. That is, for every set V in **N** with $\rho^{i}_{\mathbf{N}}\ t, V$, there is a state $v^{V} \in V$ which is not $\mathcal{Z}$-related to any $u \in U$. Let us analyze the latter statement. By the definition of the relation $\mathcal{Z}$, for each $u \in U$, v^{V} disagrees with u on some forcing formula ψ^{u}: say, it is true in u, and false in v^{V}. But then, the *disjunction* Ψ^{V} of all these formulas is true in every member of U, and still false in v^{V}. Now let Ψ be the *conjunction* of all the latter formulas, where V runs over the sets satisfying $\rho^{i}_{\mathbf{N}}\ t, V$. Evidently, we have

$$\mathbf{M}, s \models \Psi \quad \text{for each } u \in U$$

and hence

$$\mathbf{M}, t \models \{\boldsymbol{i}\}\Psi.$$

But then, by the above definition of $\mathcal{Z}$, also **N**, $t \models \{\boldsymbol{i}\}\Psi$. This means there is a set V with $\rho^{i}_{\mathbf{N}}\ t, V$ all of whose members satisfy formula Ψ. This contradicts the construction of Ψ, as v^{V} certainly does not satisfy its conjunct Ψ^{V}. □

We can even find true "strategy invariants" for any game, namely, *infinitary* forcing formulas defining the class of all games that have a forcing bisimulation with them. These results also hold over general process models, not necessarily game trees.

THEOREM. *For each finite graph* $\boldsymbol{M}$*, s, there is a modal forcing formula* $\beta_{\boldsymbol{M},s}$ *such that the following are equivalent for all graphs* $\boldsymbol{N}$*, t:*

(a) $\boldsymbol{N}, t \models \beta_{\boldsymbol{M},s}$,
(b) $\boldsymbol{N}, t$ *is power similar to* $\boldsymbol{M}$*, s.*

Proof. We only indicate the change needed in the proof of the similar result that was already given in Section 2. This time, whenever $\rho^{\boldsymbol{i}}_{\mathbf{M}}\ t, U$ holds in the model, in the global description formula $\beta_{\mathbf{M},s}$, we put a conjunct of the form

$$\phi_t \to \{\boldsymbol{i}\} \bigvee_{u \in U} \phi_u$$

whereas, if $\rho^{\boldsymbol{i}}_{\mathbf{M}}\ t, U$ does not hold, we put

$$\phi_t \to \neg\{\boldsymbol{i}\} \bigvee_{u \in U} \phi_u.$$

The argument that truth of the resulting formula guarantees a power bisimulation with the original model is essentially similar to the one given in Section 2.4. □

3.5. Connections with Modal Action Languages

The above forcing modality was already shown definable inside the modal μ-calculus in Section 2. The same implication holds at the level of simulations:

Fact. If there exists an action bisimulation between two models, linking s to t, then there is also a power bisimulation doing the same.

Action and power perspectives on games co-exist. One example are first-order *evaluation games* as treated in van Benthem (2001a: ch. 7), where truth of a first-order sentence ϕ (a modal action assertion about a model viewed as a “game-board’) amounts to the existence of a winning strategy for Verifier: i.e., a power for a player in the ϕ-game on that board. Another example are *graph games*, with players picking successors according to some protocol. Here winning strategies are associated with modal action properties of the graph itself. Another connection is a much more speculative issue of game language design.*

* *Two-sorted modal decomposition*
One can also use standard modal languages to re-design the forcing language. The forcing modality $\{\boldsymbol{i}\}\phi$ combines two logical quantifiers, being of the form

there exists a strategy such that *all* its outcomes have property ϕ.

This is a two-sorted modal combination $\langle \mathbf{1} \rangle [\mathbf{2}]$, with an existential modality $\langle 1 \rangle$ ranging over *available strategies* at the current state, and the universal $[2]$ over the *reachable outcome states* of the

4. Intermediate Levels of Game Structure

Modal action bisimulation and power bisimulation are two extremes of a spectrum. For games, there are also attractive intermediate possibilities that do not correspond directly to known process equivalences. We discuss two ways to go.

4.1. FORCING INTERMEDIATE POSITIONS

Powers tell us what players can achieve in the end. But sometimes we want to describe their powers at intermediate stages. E.g., the local dynamics of the two power-equivalent Distribution games in Section 1.2 is quite different. On the left, player **A** can hand player **E** a choice between achieving q and r – but this is impossible on the right. This property might be expressed in a simple notation:

$$\{\{\mathbf{A}\}\}(\{\{\mathbf{E}\}\}\mathtt{q} \wedge \{\{\mathbf{E}\}\}\mathtt{r}),$$

true in the left root, false in the right one. This is a new forcing modality

$\{\{\mathbf{E}\}\}\phi$ Player **E** has the power to take the game to a set of states, final *or intermediate*, all of which verify proposition ϕ.

This new level in the simulation/language spectrum does not keep track of specific actions (as bisimulation does), but it does care about the internal dynamics. Again, the new modality also has a simple recursive definition in our action language:

$$\{\{\mathbf{E}\}\}\phi \leftrightarrow \phi \vee (\mathbf{turn}_E \wedge \langle E\rangle\{\{\mathbf{E}\}\}\phi) \vee (\mathbf{turn}_A \wedge [A]\{\{\mathbf{E}\}\}\phi.$$

Given this new language, one can look for a corresponding bisimulation. In this case, a simple variant on that of Section 3 suffices:

just drop the requirement that all sets forced consist of end nodes.

Invariance and definability results remain as before. Universal validities are also similar to forcing logic. But one gets the following two interesting new properties:

$\{\{\mathbf{E}\}\}\{\{\mathbf{E}\}\}\phi \rightarrow \{\{\mathbf{E}\}\}\phi$	successive strategies compose,
$[A]\{\{\mathbf{E}\}\}\phi \rightarrow \{\{\mathbf{E}\}\}\phi$	a bunch of strategies answering any move by the opponent can be patched to one whole strategy.

But not all is routine generalization here. One interesting open question is whether intermediate forcing bisimulation can also be defined in a way that is closer to ordinary modal bisimulation. One can reformulate it in terms of *barriers* through a game that can be forced by players, asking for similar barriers on the other side. This notion is natural in a graph-theoretic perspective (Wagner, 1970). Here is

former. This requires two-sorted models with a domain of states, and one of strategies – which have their own two-sorted bisimulations. This is more elegant, as we can state properties of strategies per se, and reason about the available strategies of a game in a less implicit way.

another attempt, closer to standard bisimulation at the action level – in the form of a

DIGRESSION (Player-dependent bisimulations). Ordinary bisimulation does not care about players' turns. But what if we use the latter to make a new distinction to get the following refined notion? Call a relation $\mathcal{Z}$ a *game bisimulation* if it satisfies the following zigzag clause:

(a) whenever $x\mathcal{Z}y$, and the same player is to move in both of these states, require the usual zigzag clauses;
(b) but when, say, **E** is to move at x and **A** at y, then, whenever $x\ R_a z$, then $z\mathcal{Z}u$ *for all* successors of y – and vice versa in the other direction.

Thus, when **E** can move at x starting some strategy, she has a similar response after any move made by **A** on the other side. This does not quite match intermediate power bisimulation, witness our Distribution example of Section 1.2. On the left, **A** can move to a state where **E** can force either `q` or `r`: but there is no such state on the right, so any recipe for linking successor states must go wrong. But the above notion seems interesting in its own right. In particular, it can be shown it matches a modal language allowing only modalities of the forms

$$(\mathbf{turn}_E \wedge \langle E\rangle\phi) \vee (\mathbf{turn}_A \wedge [A]\phi)$$

and similarly for the other player. We are not totally happy with this, but leave improvements to the reader.

4.2. Alternating Bisimulation

Another intermediate game simulation starts from standard process equivalences, mixing output views (finite trace equivalence) with choice awareness (bisimulation). Combining both again takes advantage of the fact that we can make independent stipulations for the two players. Intuitively, we do not care about the choices one player makes inside her own "zone of control," but we do when control switches from one player to another. This shows in the common idea that, without loss of generality, extensive games should have an *alternating schedule* where players keep switching. Consider an example adapted from Section 1:

In terms of what can be achieved, options for player **E** are the same in both games, even though there is no modal bisimulation, only finite trace equivalence. But if player **A** gets to choose the second move, strategic effects are quite different!

E controls the outcome to the right, while **A** does on the left. The two game pairs may be distinguished as follows. First, define a *maximal move* for a player ***i*** as

> a finite sequence of moves along successive turns for ***i***, ending in either an end point, or a state where it is the other player's turn.

Now, define an *alternating bisimulation* like an ordinary modal bisimulation, but

> with zigzag conditions only with respect to *maximal moves*.

Again, this game comparison is more discerning than that of Section 3:

Fact. Games with an alternating bisimulation also have a power bisimulation.

Proof. This follows by tracing steps on both sides. Any strategy can be described in chunks that match those in the above definition: playing maximal finite sequences of one's moves, against such sequences coming from one's opponent. □

The converse is not true, witness – as ever – the two non-alternating-bisimilar Distribution games in Section 1.2, which do have equal powers for both players.

This is a converse case of Section 4.1. We start with a plausible game simulation – and look for a characteristic language. The answer is again a fragment of the full modal action language. Let **switch** be the generic name of compound actions, and MOVE_i the union of all moves for player ***i***, and consider:

> ((**turn**$_i$)**?** **;** MOVE_i)* **;** (**end**)**?**
>
> ((**turn**$_i$)**?** **;** MOVE_i)* **;** (**turn**$_j$)**?** with ***j*** the other player.

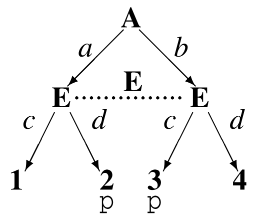
Using modalities ⟨**switch**⟩ with respect to just these "switch relations," we can match alternating bisimulation qua expressive power, obtaining results like before.

5. Imperfect Information

All results so far concern just action-outcome structure of games ("game forms"). What happens to this approach when confronted with richer, more realistic features of real games? The following sections address two such challenges. First, consider games with *imperfect information*, whose players need not know exactly where they are in a game tree. This happens in card games, electronic communication, and real life involving bounds on our memory, or powers of observation. Games like this are a good test for our framework (van Benthem, 2001a).

5.1. ACTIONS AND KNOWLEDGE IN DYNAMIC-EPISTEMIC LOGIC

A typical game tree in this sense extends the example of Section 2.1 with a dotted line indicating player **E**'s uncertainty about her position when her turn comes. Thus, she does not know the move played by player **A**, for whatever reason:



Structures like this are game models of the earlier kind, but now with additional *uncertainty relations*

$$\backsim_i$$

for each player ***i***. In the standard sense of epistemic logic, we then have at any game state *s*,

$K_i\phi$ player ***i*** knows those assertions ϕ that are true at every state $\backsim_i$-indistinguishable from *s*

Thus, we have models for a combined dynamic-epistemic language. E.g., after **A** plays move *c* in the root, in both middle states, **E** knows that playing *a* or *b* will give her p – as the disjunction $\langle a\rangle \mathtt{p} \vee \langle b\rangle \mathtt{p}$ is true at both middle states:

$$K_{\mathbf{E}}(\langle a\rangle \mathtt{p} \vee \langle b\rangle \mathtt{p})$$

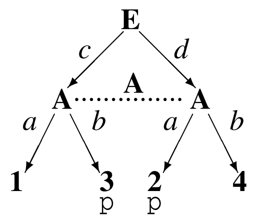
On the other hand, there is no *specific* move of which **E** knows that it guarantees a p-outcome – which shows in the truth of the formula

$$\neg K_{\mathbf{E}}\langle a\rangle \mathtt{p} \wedge \neg K_{\mathbf{E}}\langle b\rangle \mathtt{p}$$

Thus, **E** knows *de dicto* that she has a strategy which guarantees p, but she does not know, *de re*, of any specific strategy that it guarantees p. Such fine distinctions are typical for a language with both actions and knowledge for agents. Another new aspect of the above game is its *non-determinacy*. **E**'s playing "the opposite direction from that chosen by player **A**" was a strategy guaranteeing outcome p in the original game – but it is unusable now. For, **E** cannot tell if the condition holds! Game theorists only accept *uniform strategies* here, prescribing the same move at indistinguishable nodes. But then no player has a winning strategy, when we interpret "p" as the statement that player **E** wins (and hence ¬p as a win for player **A**). Player **A** did not have one to begin with, while **E** has now lost hers.

5.2. Powers, Knowledge, and Uniform Strategies

One can also look at these games more coarsely, from an outcome point of view. For instance, in terms of available uniform strategies, our game is outcome-equivalent to one that interchanges the two players plus some outcomes:



The right language for the action level is the dynamic-epistemic one. As it is a sum of two standard modal languages, its characteristic bisimulation is predictable:

> one just adds the zigzag conditions for both types of relation:
> action moves, and uncertainty jumps.

With powers the situation is more subtle (van Benthem, 2001a). One can let forcing modalities refer to them via uniform strategies, and get results like those of Section 3. But there are further interesting options. E.g., one would like to say that uniform strategies are those ordinary strategies whose players *know* that they have the outcomes they have. One very natural formalism would combine the ordinary strategy modalities of Section 3 with epistemic operators, allowing us to express

$K_i\{\boldsymbol{i}\}\phi$ ***i*** knows that she has a strategy guaranteeing ϕ,

$\{\boldsymbol{i}\}K_i\phi$ ***i*** has a strategy making her know that ϕ.

At this level, we can express special epistemic varieties of game playing. A nice example is the following version of *Perfect Recall*:

$K_i\{\boldsymbol{i}\}\phi \rightarrow \{\boldsymbol{i}\}K_i\phi$ If I know I have a strategy achieving ϕ, then I have a strategy making me know that ϕ.

Good bisimulations for this language must mix ideas from Sections 3 for powers and Section 2 for actions, viz. "uncertainty jumps." Their formulation is obvious, and so are invariance and expressiveness results w.r.t. the preceding language.

5.3. Defining Uniform Strategies

Here is an interesting epistemic feature of uniform strategies. Intuitively, down at the action level, they are recipes for moves that should work despite players' ignorance of their precise position. In terms of plans, this involves a change in syntax for conditions of actions. The following instruction is no longer good enough:

If condition C holds, then play a.

But the following *is* usable even to players with uncertainties:

If you *know* that C holds, then play a

as we are never in doubt about our knowledge or ignorance – at least in the usual S5-type epistemic game models. Call conditions of this special sort *epistemically secure*. Here is a counterpart to the earlier definability of strategies in finite games, provided we work with a two-sided modal language also looking at the past:

Fact. The uniform strategies in a finite imperfect information game are precisely those that can be defined in a dynamic-epistemic language with past operators, using only epistemically secure conditions.

Proof. Any strategy was definable with this language in such finite games. Now just note that, for uniform strategies, a uniform move "at" some maximal set X of $\backsim_i$-related nodes involves a condition of the form $K_i\ \delta_X$, with δ_X the disjunction of the unique definitions for all nodes in X. As in Section 2.5, a similar analysis works for the pure modal language alone in bisimulation contractions, taken this time with respect to both actions and uncertainty jumps. □

Thus, imperfect information does not invalidate modal logic and bisimulation perspectives – but it does raise additional issues of its own.

6. Preferences and Expectations

The real drama of game theory only enters when players attach values to outcomes, and strive for maximal utility, or other things related to *evaluation* of game states. To some extent, the debate is still on how to best model this richer setting in logic (Osborne et al., 1994; Battigalli et al., 1999; Board, 2001). The simplest way is adding a bunch of atomic propositions for value assertions at end points, encoding everything in the modal-dynamic language of Section 2. But this bleak approach does not high-light the doings of intelligent agents, who have *preferences* between states, and *expectations* about the future course of the game.

6.1. Plausibility and Expectation, First Version

Perhaps the more striking item are the expectations – which play a decisive role in understanding behaviour of players. Pay-offs and values are important in the background, as rational players will have their expectations in harmony with assumptions about maximizing values, in competitive or cooperative mode. Expectations are coded by binary relations of relative plausibility for each player $\boldsymbol{i}$:

$s \leq_i t$ $\quad$ $\boldsymbol{i}$ considers state t at least as plausible as state s.

In general counterfactual logic, these relations are ternary, with relative plausibility depending on the current state as one's vantage point of comparison x: $s \leq_{i,x} t$. One can then define operators like *conditional* belief of a proposition ϕ given that ψ:

$\mathbf{M}, x \models \mathbf{B}_i(\phi|\psi)$ $\quad$ ϕ is true in all most $\leq_{i,x}$-plausible ψ-states.

Note that this is a *maximality* principle, reflecting the typical use of maximizing utilities in game theory. For illustrative purposes, we consider a simpler case here, of expectations about the next move. In that case, the relation $\leq_i$ orders the daughters of the current node, and we can say that $\boldsymbol{i}$ believes that ϕ if

$\mathbf{M}, x \models \mathbf{B}_i\phi$ $\quad$ ϕ is true in all most $\leq_{i,x}$-plausible successors of x.

Beliefs about the further future can be described using iterations of this one-step belief modality. More in detail, de Bruin (2000) shows how one can define Backward Induction paths in a framework like this, derived from common belief of rationality. Thus, game-theoretical models of this sort naturally support languages of belief and conditionals, whose properties are reasonably well understood.

This new language still admits of *bisimulation* analysis. Consider the existential belief modality with respect to next moves (given by, say, a relation NEXT):

$\mathbf{M}, s \models \langle \boldsymbol{i} \rangle \phi$ $\quad$ there is a t with s NEXT t and $\mathbf{M}, s \models \phi$
while for no u with $t <_i u$, $\mathbf{M}, u \models \phi$.

This is a two-quantifier condition – which shows in the zigzag clause of the matching bisimulation:

* Start from any link $x\mathbb{Z}y$. Given a NEXT successor z of x, there must be a matching one u for y which is *maximal*. That is, still in the same simulation step – any more $\boldsymbol{i}$-plausible state u' than u must be $\mathbb{Z}$-linked to some state z' more $\boldsymbol{i}$-plausible than z.

Alternatively, we can separate the two steps, and take a slightly richer language dealing with the successor steps and “plausibility jumps” via separate modalities

⟨NEXT⟩ a standard action modality,
$\langle <_i \rangle$ a modality for plausibility: “at some better state.”

Thus, simple notions of plausibility and belief can be handled by combining two unary modal languages and their bisimulations as before.

6.2. PLAUSIBILITY AND EXPECTATION, LANGUAGE REDESIGN

Things get more complicated with truly *ternary* relations of relative plausibility. E.g., take an existential modality of the counterfactual sort:

$\mathbf{M}, s \models \text{<>}(\phi, \psi)$ there is a t with $\mathbf{M}, t \models \phi$ and $\mathbf{M}, t \models \psi$ and for no u with $u <_s t$, $\mathbf{M}, u \models \phi$.

Finding characteristic bisimulations for languages with maximality operators like this is not routine, as we are mixing binary and ternary relations. An analogous challenge were bisimulations for temporal languages with an UNTIL operator (Kurtonina et al., 1997). These things are often best handled by redesigning one’s language to fit more standard bisimulations. One possible approach is a *two-dimensional* one, using bisimulations that relate single states, but also pairs of states. See van Benthem (1996) on the analogous case of richer, temporal languages, and finite-variable fragments of first-order logic. But one can also pick a more workable binary modality. E.g., the following slight variant on the preceding one behaves reasonably well:

$\mathbf{M}, s \models \mathbf{U}(\phi, \psi)$ there is a t with $\mathbf{M}, t \models \phi$ and for every u with $u <_s t$, $\mathbf{M}, u \models \psi$.

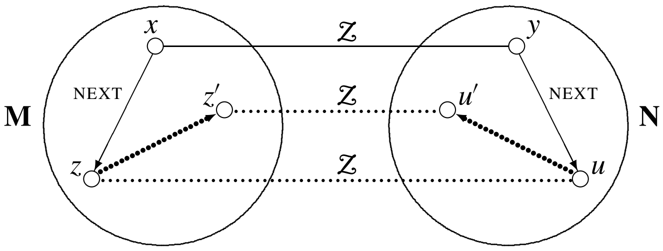
This modality has pleasant features, such as distributivity in the left-hand argument, and monotonicity in the right-hand argument:

$$\mathbf{U}(\phi_1 \vee \phi_2,\ \psi) \leftrightarrow \mathbf{U}(\phi_1,\ \psi) \vee \mathbf{U}(\phi_2,\ \psi)$$
$$\mathbf{U}(\phi,\ \psi) \rightarrow \mathbf{U}(\phi,\ \psi \vee \xi).$$

It defines the above ⟨ ⟩ via $\mathbf{U}(\phi \wedge \psi, \neg\phi)$. Notice also that this language can define a global existential modality:

$\mathbf{U}(\phi, \mathbf{T})$ ϕ is true at some point.

The notion of bisimulation for this language reads exactly like (*) in Section 6.1, but now choosing the more plausible successors in the second stage as seen from the original vantage point.



One can reason with such notions as before:

PROPOSITION. *The following are equivalent for finite models* ***M****, x and* ***N****, y:*

(a) ***M****, x and* ***N****, y satisfy the same formulas,*
(b) *there is a bisimulation between* ***M*** *and* ***N*** *linking x to y.*

Proof. (b) ⇒ (a) is a routine induction on modal formulas. For (a) ⇒ (b), as usual, the required bisimulation is the relation of modal equivalence w.r.t. the language. Consider any link xZy of the latter sort. Let z be any point in **M**. Suppose there is no matching point u in **N** satisfying our two-stage zigzag condition. This can be, for any u in **N**, for one of two reasons:

(a) z, u do not satisfy the same modal formulas,
(b) there is a point v with $v<_y u$ in **N** with no matching point w with $w<_x z$ in **M**.

In case (a), there is a formula ϕ_u true in z but false in u. Thus, the conjunction ϕ of all these formulas holds in z. In case (b), every w with $w<_x z$ in **M** satisfies some formula ψ_w which is false in v. Thus, the *disjunction* Ψ_V of the latter formulas holds in all w with $w<_x z$ in **M**. Let Ψ be the conjunction of all these formulas for all v exemplifying an occurrence of case (b). Putting these two things together, the following formula of our language is true at x in **M**:

$$\mathbf{U}(\Phi, \Psi).$$

Since y satisfies the same formulas as x, $\mathbf{U}(\Phi, \Psi)$ also holds in **N**, y. Thus, there is some u in **N** satisfying Φ such that all $v<_y u$ satisfy Ψ. But this is a contradiction! First, such a point u cannot exemplify case (a): otherwise, its difference formula with z would have been on the Φ-list. But it cannot exemplify the second case either, as its more plausible worlds all satisfy Ψ – and this excludes a modal difference of type (b). □

Thus preference structure in games, and the maximality principles of game theory based upon it, do make for more complex modal logics – but bisimulation analysis still applies, especially when we allow ourselves some freedom of language design.

7. Conclusion

Our analysis suggests two main claims. First, extensive games are natural process models, which support many familiar modal logics without the need for exotic new formalisms. Moreover, bisimulation analysis, the hall-mark of logical process theories, applies to many varieties of game structure, and seems to provide just the right tool for studying the varieties of game equivalence. We have shown this for finite two-player games here – but infinite games are within reach (van Benthem, 2000, 2001b). Having come this far, one may draw contemporary game theory and process logics together in other ways as well. Our account has been mainly restricted to game forms with moves and strategies, with some propositional annotation of game states. But many further aspects of the interaction between rational agents make equal sense in both fields, too – including such features as coalitions (Pauly, 2001), mechanism design, and the evolution of behaviour.

Acknowledgements

I would like to thank Boudewijn de Bruin, Paul Harrenstein, Hanno Hildmann, Marc Pauly, Ben Rodenhäuser, and two anonymous JoLLI referees of this paper, for help of various kinds.

References

Aiello, M. and van Benthem, J., 2001, “Logical patterns in space,” in *Logic, Language and Diagrams*, P. Scotto di Luzio et al., eds., Stanford, CA: CSLI Publications.

Baltag, A., 2000, “Co-algebra and modal logic,” Manuscript, Amsterdam: CWI.

Barwise, J. and Moss, L., 1997, *Vicious Circles*, Stanford, CA: CSLI Publications.

Battigalli, P. and Bonanno, G., 1999, “Synchronic information, knowledge and common knowledge in extensive games,” *Research in Economics* **53**, 77–99.

Blackburn, P., de Rijke, M., and Venema, Y., 2001, *Modal Logic*, Cambridge: Cambridge University Press.

Board, O., 2001, “Dynamic reasoning about knowledge and belief,” Paper presented at Italian Meeting on Game Theory and Its Applications, Ischia.

Bonanno, G., 1993, “The logical representation of extensive games,” *International Journal of Game Theory* **22**, 153–169.

de Bruin, B., 2000, “Modeling knowledge in games. Topology and logic,” MA Thesis, Amsterdam: ILLC.

Goranko, V., 2001, “Coalition games and alternating temporal logics,” pp. 259–272 in *Proceedings TARK Siena*, J. van Benthem, ed.

Halpern, J., 2001, “A computer scientist looks at game theory,” Paper presented at *Siena Summer School on Cognitive Processes and Rationality in Economics*.

Harrenstein, P., 2000, “A modal interpretation of Nash equilibria and some related concepts,” in *Proceedings LOFT-4*, Torino, G. Bonanno, ed.

Kurtonina, N. and de Rijke, M., 1997, “Bisimulations for temporal logic,” *Journal of Logic, Language and Information* **6**, 403–425.

Osborne, M. and Rubinstein, A., 1994, *A Course in Game Theory*, Cambridge, MA: MIT Press.

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

Pauly, M., 2001, "Logic for social software," Ph.D. Thesis, Amsterdam: ILLC.
Rodenhäuser, B., 2001, "Updating epistemic uncertainty," MA Thesis MoL-2001-07, Amsterdam: ILLC.
Stalnaker, R., 1999, "Extensive and strategic form: Games and models for games," *Research in Economics* **53**, 293–291.
Stirling, C., 1999, "Bisimulation, modal logic, and model checking games," *Logic Journal of the IGPL* **7**, 103–124.
van Benthem, J., 1996, *Exploring Logical Dynamics*, Stanford, CA: CSLI Publications.
van Benthem, J., 1997, "Dynamic bits and pieces," Report LP-97-01, Amsterdam: ILLC.
van Benthem, J., 1999, "When are two games the same?," Amsterdam: ILLC.
van Benthem, J., 2000, *Logic in Games*, Electronic Lecture Notes, Amsterdam: ILLC.
van Benthem, J., 2001a, "Games in dynamic-epistemic logic," *Bulletin of Economic Research* **53**, 219–248.
van Benthem, J., 2001b, "Games over time," Amsterdam: ILLC.
van Benthem, J., 2001c, "Logic games are complete for game logics," *Studia Logica*, to appear.
van Benthem, J., van Eijck, J., and Stebletsova, V., 1994, "Modal logic, transition systems and processes," *Journal of Logic and Computation* **4**, 811–855.
Wagner, K., 1970, *Graphentheorie*, Mannheim: Bibliographisches Institut.