institute *for* logic, language *and* computation

p u b l i c a t i o n

research report

Universiteit van Amsterdam

institute *for* logic, language *and* computation

JOHAN VAN BENTHEM

When are Two Games the Same?

PP-1999-22, *received*: October 1999

ILLC Scientific Publications
Series editor: Dick de Jongh

Prepublication (PP) Series, ISSN: 1389-3033

Institute for Logic, Language and Computation (ILLC)
University of Amsterdam
Plantage Muidergracht 24
NL-1018 TV Amsterdam
The Netherlands
e-mail: illc@wins.uva.nl

When are Two Games the Same?

June 1999

Johan van Benthem, Amsterdam & Stanford

1 Logic and games: historical encounters

Logic and games have had several encounters in history. One major factor in the emergence of logic in Antiquity was a systematic interest in the rules of argumentation, which have a clear game-like structure. Games were mathematized for the first time, one might say, in the17th century: when Pascal and Huygens based their idea of probability on betting odds. The logic connection for this came later, e.g., with the Dutch Book theorems of the 1950s. But that same decade also saw some basic results inspired by Game Theory in the modern sense, such as the Gale-Stewart theorem stating the determinacy of closed infinite games (extending an old result of Zermelo's), that became important in descriptive set theory. Such connections have multiplied in recent decades, witness the variety of topics at the bi-annual TARK and LOFT conferences. This time, the focus is rather on the role of *knowledge in action*, obviously a shared concern between logic and game theory, including deep issues of rationality, counterfactual reasoning, communication, and cooperation. These contacts between logic and game theory do not all fit the same mold. There are uses of logic as a formalism for analysing and sharpening up game-theoretical argumentation, and there is logic as a general meta-theory of games. But there is also logic as a 'producer' of specific games for logical purposes (argumentation and proof, evaluation of sentences, model comparison, model construction), which then makes it a 'consumer' of game theory! Such different, and changing relationships have nothing strange, of course, in real life: the same human being can be both in our care, and our best advisor.

2 Our purpose: the logical structure of games

We want to step back from any particular detailed concern, and look at games with the eyes of a logician. *What is a game?* This broad question translates into several more detailed ones, witness our section headings below. First, what kind of analysis would one expect? For a start, it seems obvious that games are a special kind of *processes*, which involve both concrete action and cognitive action. Thus, the world can change, but there is also information flow. But then, the fundamental question is: what kind of processes? A typical foundational approach should ask here for the *invariances* defining the field of enquiry.

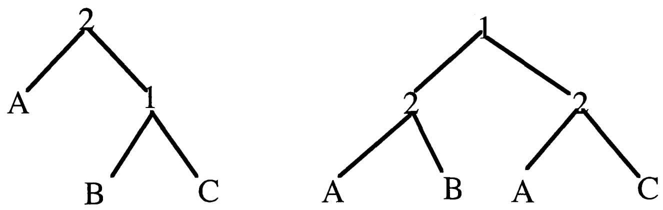
What transformations between structures leave 'the same game'? Of course, this only makes sense if we first settle on some *representation* format for games. Next comes mathematics: where there are transformations of structures, there is *invariance* for certain notions across transformed ones. And 'then there is logic': invariants suggest *definability* in suitable languages, whose valid principles encode *logical laws* of the structures described. We will apply this abstract perspective to games, using their analogy with processes, for which latter succesful invariance approaches already exist. Our discussion is example-based, using logic games as a more concrete 'laboratory setting' for exploring options and results. We leave formal elaboration to further publications. Our contribution is a way of thinking, which will hopefully be congenial to game theorists. We start with perfect information games (most logic games are) – sketching afterwards how imperfect games fit as well.

3 Logic games

Logic games exist for many purposes. Here is one example, due to Hintikka in the 1960s, that demonstrates some basic ideas. We give the games, and progressively present some of the basic logical issues that they raise. In this paper, we pursue only some, not all of these – but the example may be of broader interest by itself.

3.1 Evaluation games for propositional logic

Consider propositional formulas formed using proposition letters p, q, .. and Boolean operators $\neg$, $\wedge$, $\vee$. Each such formula ϕ defines a game between two players, Verifier (**V**, who claims it is true) and Falsifier (**F**, who claims it is false), as follows. Suppose that we have some fixed valuation VAL for the proposition letters. If the formula is a disjunction, **V** must choose, and play continues with respect to the chosen disjunct. If it is a conjunction, **F** chooses, and play continues with respect to the chosen conjunct. If it is a negation, players interchange roles, while the winning convention also reverses. Finally, winning or losing occurs with an atom ϕ: if it is true according to VAL , **V** wins, otherwise, **F** wins. Thus, the game is zero-sum in a direct sense. Note the eminently game-theoretic explication of the logical operations. The standard Booleans now denote *choices*, and *role switch.* These moves return in many logic games, and indeed in games in general, where choice nodes for one player can be labeled with $\wedge$ and for the other with $\vee$. For instance, the two finite game trees in the next figure correspond to the propositional formulas $A\wedge(B\vee C)$ and $(A\wedge B)\vee(A\wedge C)$, respectively. Likewise, many game trees in textbooks are propositional formulas. Notice these two Boolean formulas are logically equivalent: so are the pictured games. But in which sense...? We will return to this issue later on.



3.2 Winning strategies , truth, and logical validities

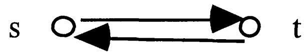
Hintikka observed the following key fact, which is easily proved by induction on formulas:

> *Formula ϕ is true in VAL in the classical sense iff*
> *Verifier has a* winning strategy *in the game for ϕ.*

This simple observation has far-reaching consequences. In particular, *logical laws now express game-theoretic facts*. An example is the law of Excluded Middle $\phi \vee \neg\phi$ .This is true in every valuation, and hence Verifier has a winning strategy for it in all these cases. Given the definition of the Booleans, this game consists of a choice by Verifier whether to play the game for ϕ as **V**, or that for $\neg\phi$: i.e., the game for ϕ once again, but now as **F** . The assertion about a winning strategy says that the game for every formula is *determined.* This is the reflection inside logic of an old game-theoretic result by Zermelo, stating that every two-player zero-sum game of finite depth is determined. (To some extent, Excluded Middle for stronger logical languages can be seen as an equivalent of Zermelo's Theorem.) Likewise, valid principles of Boolean Algebra express game-theoretic facts, such as the above case of propositional distribution: the corresponding role switch for players changes the prima facie representation of the games, but it does not change them essentially.

3.3 Games for predicate-logical evaluation, and other purposes

These evaluation games extend to first-order predicate logic. 'Game states' now involve some model **M** , plus an assignment s of objects s(x) to variables x , as in standard first-order semantics. This allows for direct interpretation of atomic predications, as above. The Boolean operations retain their game-theoretic role. Next, an existential quantification $\exists x\phi$ corresponds to a choice by **V** of some object d in the domain, after which play continues for the formula ϕ , with respect to the new assignment s[x:=d] . Universal quantification is the same, but now the move is **F**'s. Here is a concrete illustration. Let a model **M** have two objects s, t , with a binary relation R holding only of s, t and t, s in that order:



Here is the game of perfect information for the formula $\forall x\ \exists y\ Rxy$:

			F				
	x:= s				x:= t		
	V				**V**		
y:= s		y:= t		y:= s		y:= t	
win_F		win_V		win_V		win_F	

More radically, we can deviate from standard first-order syntax in this game setting, and reinterpret quantifiers as follows. First, there is an atomic move $\exists x$, which is a game by itself: "pick an object d , and set x:=d" . This succeeds for **V** , whenever some object is available. A universal quantifier $\forall x$ is the same game, but now for **F** . Atomic picking games are comparable to the earlier one-step test games for atoms. Next, play continues w.r.t the formula ϕ . The game construction here is *sequential composition.* From a game-theoretic standpoint, then, standard predicate logic is essentially a calculus of complex games formed out of object-picking and fact-testing moves by means of choices, role switches, and compositions. Indeed, that operational structure is completely general. What makes them model *evaluation* games is the choice of the specific atomic repertoire.

Other logic games perform more complicated tasks. There are games that carry out model *construction* for given formulas, and hence test *satisfiability* (Hodges, Hodkinson). There are also games whose winning strategies are in fact *proofs* of some proposition defended by one player against 'concessions' from the other (Lorenzen). The latter have evolved further into so-called *interaction games* for linear logic (Blass, Abramsky). Semantically, such more formal games cannot use information about any specific propositional valuation or predicate-logical model, so that their strategies are *uniform.* Of course, the latter may be much harder to find, as satisfiability and provability problems are undecidable (at least for predicate logic). Finally, there are widely used logic games for many other tasks, including in particular, *model comparison.* For the point of that, see below.

3.4 Logical operations as game constructions

Logical calculi handle formal syntax. Logical formulas ϕ can be interpreted as finite effective representations of games. (What happens then in standard semantics is really this. As soon as a concrete model **M** is supplied, these abstract forms turn into concrete games game(ϕ, **M**) of 'type' ϕ .) Now these formulas have their usual logical operations, which we can interpret in game-theoretic ways. But what do the games want themselves? What are natural operations? Certainly, the above ¬, ∧, ∨ plus now also sequential composition • are a very natural set. Note that these operations suffice for building all finite game trees. But there may be other operations, and indeed, the logical literature is developing these.

3.5 Two levels: dynamics = games and statics = assertions about games

Now that we have interpreted standard logical formulas as games, we seem to have lost the traditional useful role of logical formulas as declarative statements. But this is not true, since the latter re-appear. Consider Hintikka's Lemma about the equivalence of truth with the existence of a winning strategy. Note this involves a significant assertion *about a game*, namely that it has a winning strategy. If we want to study such assertions systematically, we really need a *two-level language*, with both 'dynamic' game expressions G and 'static' propositional assertions p (not interpreted as games!), so that we can say things like

> <G>p : player **V** has a strategy for playing G
> that will always result in final states satisfying p .

Languages like this (proposed by Parikh) are close to 'dynamic logic' in computer science. One step further, we might also want to respresent the *strategies* themselves directly. For first-order logic, we then get 'Skolem functions'. We postpone this issue until Section 6.

3.6 From perfect to imperfect information

Most logic games are based on perfect information: players always know where they are. But Hintikka has also proposed versions of his game semantics for predicate logic that must have *imperfect information.* For instance, in a 'slashed' quantifier pattern **∀x ∃y/x** , the idea is that **V** has to make her choice independently from the preceding 'challenge' by **F** . She may have forgotten what it was, or there may be other reasons (one interpretation is of **V** as a *team*, whose members may not always know what all members of the opposing team are doing). Games like this no longer satisfy the standard logical laws. E.g., Excluded Middle fails. The game tree for ∀x ∃y/x Rxy is *non-determined* in the model of Section 3.3:

F

x:= s .... x:= t

V **V**

y:= s y:= t y:= s y:= t

win_F win_V win_V win_F

This is about the simplest example of a non-determined game tree. Logic games with incomplete infornation are a very new trend, but they do show how the field is expanding.

This completes our brief tour of logic games, as an introduction to some basic notions, and the style of thinking in this paper. Now, we turn to logical analysis of games in general.

4 Invariance and definability: Helmholtz' program

4.1 Transformations and invariants

In the mid-19th century, Helmholtz introduced a powerful idea into the foundations of geometry, which has become a dominant mathematical theme since. What he wanted to explain, rather than merely assume, was the standard repertoire of geometrical primitives. Nature around us is full of invariances. For instance, our natural movements are translations and rotations, and their accompanying shifts in 'perspective' leave certain relations between points invariant, such as betweenness. An appropriate scientific account of space then starts with a language expressing these natural invariants. This approach admits of many variations. Euclidean geometry is about the invariants of the classical transformations, topology arose by choosing a much larger transformation group (the homeomorphisms). More generally, this methodogy applies to any subject matter, whether in physical or in conceptual space. Usually, things are given to us in some kind of representation – but to fix the abstraction level, we must determine what are the transformations, or more general *simulation relations* that determine *when two representations are the same*. Once we have these simulation relations, we get invariant notions inside our structures, and we can design apprropriate languages expressing these. With games, of course, this leads to two questions: (a) what are the natural representations, and (b) what are the natural invariances? We will approach these issues via a discussion of related process theories in Section 5.

4.2 Potential isomorphism and first-order definability

A good example of invariance thinking is first-order predicate logic. Representations are the usual semantic models, and the invariance relation is often taken to be 'isomorphism', i.e., a bijection preserving all relations and operations. But a better notion eventually, showing more of the relevant 'dynamics' when we are comparing two models, is *potential isomorphism.* This is rather a family PI of finite partial isomorphisms between two models **M**, **N** satisfying two dynamic 'back-and-forth properties' which say that given any partial isomorphism $f \in PI$ and any object $a \in |\mathbf{M}|$, there exists some corresponding object $b \in |\mathbf{N}|$ with $f \cup \{(a, b)\} \in PI$ – and vice versa.

The idea is that objects on either side can be *simulated* by objects on the other in arbitrarily large finite environments. Potentially isomorphic models satisfy the same first-order sentences, by an easy induction. There is a converse, too, though a bit more complicated: two models that satisfy the same first-order sentences in a first-order logic with arbitrary *infinite* conjunctions and disjunctions are potentially isomorphic (Karp's Theorem). These results establish a close-fit between first-order logic and its invariance relations. Incidentally, the original history of these results was the other way around: the language existed first, and the corresponding simulation (in this case, potential isomorphism) was discovered only afterwards. Both patterns occur in the literature: we can listen to nature (going "from simulations to languages"), or we can listen to language ("from languages to simulations") – and of course, the two directions can also interact.

4.3 Testing invariance by model comparison games

The preceding style of analysis concerns the whole language at once. What about a focus on more specific individual assertions? Here we need some *fine-structure* of simulations, which is provided by: *games*! Ehrenfeucht-Fraïssé games test finite 'degrees' of potential isomorphism, as follows. There are two players, Spoiler (**S**) and Duplicator (**D**). **S** claims the models are different, **D** that they are similar. They agree to play over some finite number of k rounds. Each round proceeds as follows. **S** chooses one of the models, and picks an object d in its domain. **D** then chooses an object e in the other model, and the pair (d, e) is added to the current list of matched objects. At the end of the k rounds, the total matching obtained is inspected. If it is a *partial isomorphism*, **D** wins; otherwise, **S** wins the game. (A partial isomorphism is simply an injective partial function (often finite) between subsets of the domains of two models, which is an isomorphism as far as its own domain and range are concerned.) Here are two examples, illustrating winning strategies.

I 3-arrow cycle versus 4-arrow cycle

Round 1 S chooses 1 in **M** — D chooses i in **N**

Round 2 S chooses 2 in **M** — D chooses j in **N**

Round 3 S chooses 3 in **M** — D chooses k in **N**

S has won, because this match is not a partial isomorphism. But S can do better: there is a winning stategy in 2 rounds, starting with i in **N**, and then choosing k in the next round. No such pattern occurs in **M**, so D is bound to lose.

II **Z** (integers with <) versus **Q** (rational numbers with <)
These two linear orders have different properties: one is dense, the other discrete.

…..	-1	0	+1	…	**Z**
--					**Q**
		0	1/3		
		↑ 1/5			

D has a winning strategy here for the game over 2 rounds. But S can win this comparison game in 3 rounds. Here is a typical play:

Round 1 S chooses 0 in **Z** — D chooses 0 in **Q**

Round 2 S chooses 1 in **Z** — D chooses 1/3 in **Q**

Round 3 S chooses 1/5 in **Q** — D chooses any object: and loses.

These games are determined again, being of finite depth. Of course, players may play badly and lose, even if they have a winning strategy. Spoiler wins by exploiting some 'definable difference'. (E.g., in **II**, this is the first-order formula for density of < : $\forall x \forall y\ (x<y \rightarrow \exists z\ (x<z\ \&\ z<y))$.) The key fact about the role and use of Ehrenfeucht-Fraïssé games is the following Adequacy Theorem (using the *quantifier depth* of a first-order formula, measuring the longest nesting of quantifiers occurring inside it: e.g., density had depth 3):

*For all k, **M**, **N**, the following are equivalent:*

*(a) **D** has a winning strategy in the k-round game between **M** and **N***

*(b) **M**, **N** agree on all first-order sentences up to quantifier depth k .*

The proof of this is by induction, and establishes something extra, stated most easily for Spoiler. There exists an effective correspondence between winning strategies for **S** in the k-round game and first-order sentences ϕ of quantifier depth k with $\mathbf{M} \models \phi$, *not* $\mathbf{N} \models \phi$.

Finally, the finite limit can be pushed upwards. We can play the above game *forever*, saying that **D** wins an *infinite run* if she has maintained a partial isomorphism all along the way. In this way, we get a game in which winning strategies for **D** correspond precisely to the earlier *potential isomorphisms* between the two models involved.

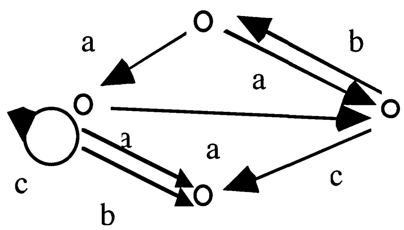
Model comparison games are the most widely used logic games. Applied in our setting, they will also illustrate a very strange, and sometimes disconcerting, logician's habit. When games become our structures, and their invariance relations our target, comparison games may be a good way of measuring how much one game is like another. Thus, we play games to compare games! This happy circularity is not a prerogative of logicians, of course. Computer scientists would be happy to let programs test the equivalence of programs.

5 Modal process theory

Games are processes, in which actions are performed, that have effects in the outside world. They are not just any process, of course, as internal interactions between players with goals and roles are essential – but even so, we can profit from the analogy. Logical theories of processes are much studied in computer science. Originally, these were geared toward computation stricto sensu, but the resulting insights are really about any kind of process.

5.1 Representation: process graphs

A wide-spread representation in this area are *process graphs* $\mathbf{M} = \langle S, \{R_a\}_{a \in A}, V \rangle$ with labeled transition arrows $\rightarrow_a$ standing for succesful executions (input state, output state) of the action a . (In computer science, such **M** are called 'labeled transition systems'). Here, S consists of the states of a computer, or any physical or man-made process (such as a game...). The binary relations R_a encode transitions between states corresponding to actions a . The valuation V indicates for each atomic proposition whether or not it is true, at any state. This is about the simplest logical model of action, as in the following picture:



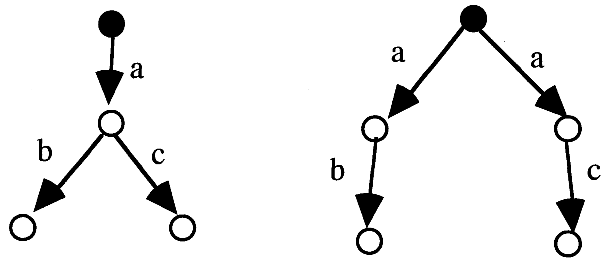
The four small circles are states – one of which might be considered the 'starting state'. There are three types of action: a, b, c . In the western state, actions b, c are deterministic:

one possible transition. (c 'loops', and does not change the state.) a is non-deterministic: two possible transitions. The southern state is an 'endpoint': no action possible. Actions can have mutual relationships: e.g., above right, b 'undoes' the effect of a . Also, atomic actions combine to form new compounds. The sequential action a•b ('a followed by b') gives a new transition from 'west' to 'north'. The picture may also indicate 'local properties', e.g., some proposition letter "cold" holding in west and north, but not in east and south.

In concrete applications, states get more structure. E.g., when evaluating predicate-logical formulas, as in our evaluation games of Section 3, or for simple programming languages, states are usually all assignments taking variables to objects into some fixed model of 'data values', atomic transitions are tests, or shifts in values for a variable (think of an instruction x:=t), and the valuation says which atomic predicates hold for every relevant assignment.

5.2 A ladder of process equivalences; bisimulation

Next, we must specify an invariance telling us when two representations stand for the same process. There is no unique best choice for this – it depends on the purpose. If observable behaviour is your only concern, compare as follows. Look at all finite sequences of actions that can occur from the initial state ('root'), and call two process graphs the same if they are *finite-trace equivalent.* This is the choice in formal language theory: two finite automata are equivalent if they recognize, or produce the same language, viewed as a set of strings. It is also used by newspapers comparing politicians by their voting records, regardless of internal differences (one's decisions may have been fast, another's may have cost sleepless nights, full of dilemma's). Here are two trace-equivalent process graphs:

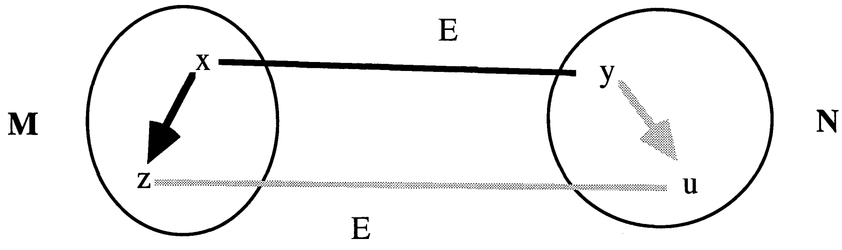


But for other purposes, you do want to encode the internal structure (e.g., when considering marrying a politician, wondering how much sleep you are going to get). Most computer scientists wish to take internal choice structure into account, and the notion of choice for this is *bisimulation.* A bisimulation is any binary relation E between states of two process graphs **M**, **N** such that, if x E y , then we have atomic harmony, plus zigzag clauses:

(1) x, y verify the same proposition letters

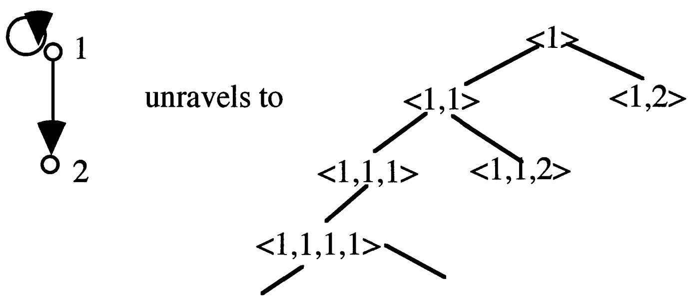
(2a) if x E z, then there exists u in **N** s.t. y R u and z E u

(2b) vice versa.



Example. The following two graphs have a bisimulation, indicated by the shaded lines:

But the above two finite-trace-equivalent graphs have no bisimulation between their roots. Bisimulations occur quite often. An important example is *tree unraveling*, a construction that will be mentioned below. Every process graph **M**, x is bisimilar with a rooted tree-like model constructed as follows: the states are all finite paths through **M** starting with x and passing on to R-successors at each step. One path has another path accessible if the second is one step longer. The valuation on paths is copied from that on their last nodes.



A tree unraveling is much like a game tree, with the branches encoding all possible runs. Finally, there exist even finer process invariances than bisimulation, such as *isomorphism of generated graphs*. Thus, there is a whole spectrum of legitimate process theories.

5.3 Definability in modal languages

As with potential isomorphism (of which bisimulation is a well-chosen weaker variant), there exists an adequate description language defining those properties of process graphs that are invariant for bisimulation. This is a *modal action language* with proposition letters for local properties, Boolean operations, and two modal operators <a>φ, [a]φ expressing "there exists some a-successor where φ holds" and "for all a-successors, φ holds", resp. This language is interpreted in the usual Kripke-style, at states x in process graphs **M** . Here is the key result motivating bisimulation:

> *For any bisimulation* E *between* $\boldsymbol{M}$ *and* $\boldsymbol{N}$ *with* $x\,E\,y$,
> $\boldsymbol{M}, x \models \phi$ *iff* $\boldsymbol{N}, y \models \phi$ *for all modal formulas* ϕ.

The proof is an easy induction on modal formulas. We say *modal formulas are invariant for bisimulation.* There are also a number of converse results, tying up the close fit, such as:

> *Let the worlds* x, y *satisfy the same modal formuals in the finite models* $\boldsymbol{M}, \boldsymbol{N}$.
> *Then there exists a bisimulation between* $\boldsymbol{M}, \boldsymbol{N}$ *connecting* x *to* y.

For arbitrary models, this converse does not hold. But the equivalence is valid if you use an 'infinitary' modal language allowing arbitrary *infinite conjunctions and disjunctions.* And something even stronger is true: each process graph has a complete bisimulation-invariant:

> *For each* $\boldsymbol{M}, x$, *there exists an infinitary modal formula* ϕ *such that*
> *for all* $\boldsymbol{N}, y$, $\boldsymbol{N}, y \models \phi$ *iff* $\boldsymbol{N}, y$ *is bisimilar with* $\boldsymbol{M}, x$.

This really matches bisimulation with its modal description language (modulo some infinite hocus pocus). As before, one can go back and forth between a structural process invariance and definability in some process language. Bisimulation was proposed in computer science in the 1980s, and the modal connection was noted only afterwards. But its first discovery was in modal logic in the 1970s, as an invariant to the already existing modal language. Once again, we note that this is just one result on a long ladder. Coarser or finer process equivalences than bisimulation will match other, weaker or stronger, modal languages.

Finally, there is fine-structure again, via finite *bisimulation games*, where Spoiler's winning strategies correspond effectively to modally expressible differences. E.g., Spoiler wins in 2 rounds in the above non-bisimilar model pair (starting from the roots), using the difference formula <a>(<b>T & <c>T) of modal operator depth 2 to guide his moves.

6 When are two games the same?

Now that we have a panorama of invariances and matching description languages for processes in general, we can think about the case of 'game processes' in particular.

6.1 Representation formats: process graphs versus game trees

There are many options here, which we will not all explore. One can distinguish 'formal games', such as the formulas of Section 3, and 'material games', such as their associated evaluation-game trees in concrete models. A choice depends on how much structure (internal and external) we wish to represent. Technically, we will mainly use *process graphs* – enriched with suitable propositional constants for game structure.

> $\mathbf{turn_i}$ indicates that player i is to move at a state where it holds,
> $\mathbf{win_i}$ marks states as winning for player i .

One can make process graphs satisfy extra axioms, stating that turn and win predicates are disjoint, that each node except final ones has one turn predicate true, while win predicates hold at all and only final nodes. Modal formulas then describe possibilities in further play as seen from the current node. For some purposes, though, *game trees* of possible *runs* (histories) are preferable. E.g., in many infinite games, there is no special interest to final nodes (they are often considered losses for the player who cannot move), while we *are* interested in the long-term behaviour of infinite runs to determine who has won. (E.g., in interaction games, we let one player win if the total infinite run is 'fair', or satisfies other nice properties. In the infinite Ehrenfeucht game, Duplicator won provided she managed to maintain a partial isomorphism all along.) In these cases, a *temporal* language, or sometimes a modal-temporal language, seems more appropriate than a purely modal one – but, except for a few passing remarks, we forego analysis of the latter format in this paper. Anyway, game trees are closely related to process graphs, via the earlier tree unraveling.

In what follows, we just look at the pure action structure. No preferences are encoded yet. This seems a serious restriction from the standpoint of game theory – but preference considerations can be added naturally to the analysis proposed here.

6.2 Game equivalence 1: bisimulation cum trace equivalence

True game equivalence sits in between two levels of the earlier process hierarchy, Finite Trace Equivalence and Bisimulation. Consider an example adapted from an earlier picture:

$$\begin{array}{ccc} & \mathbf{1} & \\ & a\downarrow & \\ & \mathbf{1} & \\ b\downarrow & & c\downarrow \\ p & & q \end{array} \qquad \text{versus} \qquad \begin{array}{ccc} & \mathbf{1} & \\ a\downarrow & & a\downarrow \\ \mathbf{1} & & \mathbf{1} \\ b\downarrow & & c\downarrow \\ p & & q \end{array}$$

In terms of what happens and what can be achieved, options for the player **1** are the same on both sides, even though there is no modal bisimulation, only finite trace equivalence. But if player **2** gets to chose the second move, the strategic effects are quite different!

$$\begin{array}{ccc} & \mathbf{1} & \\ & a\downarrow & \\ & \mathbf{2} & \\ b\downarrow & & c\downarrow \\ p & & q \end{array} \qquad \text{versus} \qquad \begin{array}{ccc} & \mathbf{1} & \\ a\downarrow & & a\downarrow \\ \mathbf{2} & & \mathbf{2} \\ b\downarrow & & c\downarrow \\ p & & q \end{array}$$

The latter two games are not the same. **1** controls the outcome to the right, **2** on the left. (In algebraic terms, one can also contrast associated 'propositional formulas': $a•(b \vee c)$ vs $(a•b) \vee (a•c)$, and $a•(b \wedge c)$ vs $(a•b) \vee (a•c)$.) Now, we are not interested in one player's internal choices per se, only insofar as they affect alternation with *others*. Thus, we want FTE for one player's moves, but Bisimulation for the other's. This may be achieved as follows:

> we only require the zigzag conditions of bisimulation with respect to *extended moves* that consist of a finite action sequences by one player ending in a state that is a move for the other player, or an endpoint.

Thus, one quite natural game equivalence is what may be called *alternating bisimulation*. This notion has the additional virtue of implying something which game theorists want, viz. the availability of 'the same strategies' (a phrase requiring elucidation!) for both players:

> *Alternating bisimilar games have the same strategies forcing sets of end nodes.*

The reason is simply this: the zigzag conditions allow both players to copy move 'across'. But the converse is not true, witness the first example in this subsection, where the strategies seem "the same" on both sides, even though there is no bisimulation. Indeed, we need something coarser here. Alternating bisimulations focus on action structure: they are most concerned with what the players *do*. Our next way of looking at games focuses only on what the players *get* or *achieve*. Incidentally, both are valid perspectives. Some games are played only for their gains, others less for that than the pleasurable activities involved.

6.2 Game equivalence 2: forcing bisimulation

An interesting purely result-oriented notion of game equivalence underlies an abstract game logic devised in the 1980s by Rohit Parikh, using an analogy with modal logic of programs. The idea is this. In general, players in a game may not be able to force unique outcomes. But they can force certain *sets* of outcome states (regardless of how the other plays), satisfying certain properties. Thus, we say that

$\rho_G{}^{\mathbf{i}}\ sX$	player **i** has a strategy for playing game G whose resulting states are always in the set X

For instance, consider the earlier case of propositional distribution, pictured as two games in Figure 1. There is no bisimulation, and not even an alternating bisimulation, starting from the two top nodes. But the forcing relations for the two players **V**, **F** between the top node s and sets of final nodes (outcomes) are the same in both games:

for **F**	{A}, {B, C}
for **V**	{A, B}, {A, C}

Given any game tree, we can compute these forcing relations in an obvious inductive manner. If player **i** is to move at node s, then he can force exactly those sets of outcomes that are forceable from at least one of his successor nodes. The other players can only force *unions* $\bigcup_x Y_x$ of sets which they can force at these successor nodes x. But even though we will usually think of forcing sets of final nodes, this is not necessary. The notion makes sense for any state of a game, and any subsequent set of states. (I might force visiting certain 'intermediate zones' in a chess game, perhaps regardless of my winning or losing.)

To define the corresponding notion of game equivalence, we assume that these relations $\rho_G{}^{\mathbf{i}}$ are given somehow. Now we define our new result-oriented game equivalence: viz. *forcing bisimulation* between two game representations (viewed as process graphs again). This proposal generalizes the notion of a modal bisimulation. A forcing bisimulation is

> a binary relation E between states in models G, G' satisfying atomic harmony as before (E-related states satisfy the same propositional letters) as well as two zigzag clauses, that must hold for each player **i**: if x E y and $\rho_G{}^{\mathbf{i}}$ x, U, then there exists a set V with $\rho_{G'}{}^{\mathbf{i}}$ y, V and $\forall v \in V\ \exists u \in U$ u E v – and vice versa.

The rationale for this shows in 'getting the same outcomes', plus good generalizations of earlier results for modal logic. This time, the appropriate modalities are like the strategic statements about games first encountered Section 3.5. Consider a modal language formed with proposition letters, Boolean operations, and modalities $\langle i \rangle\phi$ interpreted as follows:

$$\mathbf{M}, x \models \langle i \rangle\phi \quad \text{iff} \quad \text{there exists a set } Y \text{ with } \rho^{i}\ x, Y \text{ and } \forall y{\in}Y\ \mathbf{M}, y \models \phi$$

Here we refer to forcing relations given in our model **M** . (Parikh's system is a bit different, in that different games can co-occur: but we defer this extra feature until Section 8 below.) The logic for this language is the usual minimal modal logic, except for the fact that there is *no distributive law* for $\langle\rangle$ over $\vee$. Evidently, a player may be able to force some set of outcomes which is a union of two others, without being able to force either smaller set. Forcing bisimulations between these models are as above. Now we get the usual results – of which we formulate a sample:

(1) *modal formulas in this language are invariant for forcing bisimulation*
(2) conversely, *if finite models* ***M****, x and* ***N****, y satisfy the same modal formulas, then there exists a forcing bisimulation between them* .

We can even find true 'strategy invariants' for any game, namely, infinitary modal formulas (cf. Section 5.3) defining the class of all games that have a forcing bisimulation with them.

Remark. That this notion is natural shows in the fact that it also occurs independently in the semantics of 'concurrent dynamic logic', and in that for 'topological modal logics'. There is a link with topological views of games in the latter case – which we will not spell out here.

6.3 Further aspects: explicit strategies, trees of runs, game equivalence of propositions

We now have two proposals for natural game equivalences, one more action-oriented, the other more result-oriented. In line with the earlier perspective in process analysis, though, there may well be others. One way in which the analysis can be refined would bring in *strategies* explicitly. Strategies are implicit in the modal game logic, but they are 'quantified away' in the above truth condition for $\langle i \rangle\phi$. There are various ways of bringing strategies into a logical picture, witness the literature on interaction games and linear logic. Most simply, perhaps, we could think of a strategy as a *subgame* in which one player has uniquely restricted moves. In that case, one very simple formalization would use just some *new proposition letters* that hold in those subgames. We will see what this does below.

Even at the level of analysis that we have proposed, concerning just actions and outcomes there is another possible mind-set. Game trees have a modal character, with branchings indicating all possible moves. But in reality, only one selection of moves is made, and the history of what actually happened is linear. Should not one distinguish between *what can take place* and *what will take place*? This intuitive distinction is familiar from *branching temporal logic*. There we evaluate statements on branches in a tree of possible histories, which enables us to distinguish between "ϕ will happen" (on the current branch) and "ϕ can happen" (in the future on some branch, perhaps forking off from the actual one). One can also try to find game descriptions in these temporal, or modal-temporal terms, identifying a game with the tree of its possible runs. Simulation thinking in the above style is certainly possible in this context too – and it may even become imperative once we take games with infinite runs seriously. But note that, since we do not know the actual branch that we are on, the very distinction that we are making is to some extent wishful thinking...

One test of our analysis takes the two proposals back to logic games – always a good sanity check. In that setting, alternating and forcing bisimulation are two answers to the question: "*when are two Boolean propositions the same?*" This may sound strange: is not this settled by standard propositional logic? To the contrary, there is a continuing debate about the proper 'grain size' of propositions. Our proposals give two new answers. Consider forcing bisimulation (leaving alternating bisimulation open). This validates most basic laws of Boolean Algebra (associativity, commutativity, idempotence, distributivity) in a general sense, where we take atomic expressions any way we like – as atomic tests, or further games to be played. This takes care of the Boolean mechanics of transforming propositions. By contrast, the remaining principles of Excluded Middle and Non-Contradiction did not express general laws about games, but the special assumption of their being *determined.*

6.4 Games for comparing games

As with process bisimulation, there are natural game versions of the above equivalences. Again these involve Spoiler and Duplicator, playing over some finite number k of rounds. E.g., for forcing bisimulation, with a current match between states $x - y$, Spoiler chooses some set U with $\rho^i\, x, U$ and Duplicator has to respond with a set V such that $\rho^i\, y, V$. Still in that round, Spoiler chooses a state $v \in V$, and Duplicator must respond with a state $u \in U$. The pair $u–v$ becomes the new current match. The Adequacy Theorem ties this up with game-logical differences of operator depth k. Thus, we can play games to answer a more finely-structured question than the one in our title: *how similar are two given games?*

7 Game languages

7.1 From game structure to modal languages

What languages provide natural game descriptions? One can approach this issue in many ways. Most practically, one can analyze existing game-theoretical notions and arguments, and determine some appropriate formalization. This could be in modal languages, or newly developed formalisms, or just standard first-order logic (no need to invent exotic things, when standard ones will do). As a matter of fact, modal languages offer a convenient simple representation for many arguments, including the existence of strategies, which typically involve alternating operator patterns $[a]\langle b\rangle[c]\langle d\rangle\, \phi$. (But to deal with strategies explicitly, we would have to change the modal set-up considerably.) Note that this is still just about action structure and outcomes. As is well-known, much more modal structure comes in when we consider *preferences* between outcomes (but there is a modal preference logic) and *incomplete information* (the well-established uses of epistemic logic; see below).

7.2 Illustration: backward induction and modal fixed-point languages

The uses to which game theory puts modal languages can be unexpected, and involve additional sophisticated logical machinery. We give an illustration, to make our discussion more concrete. The usual *backwards induction arguments* proving Zermelo's Theorem, and driving the location of Nash Equilibria use the basic modal language in a new manner. What happens is this. We are trying to define a strategy inside a given game tree – which may be coded as some new proposition letter s – by analyzing its behaviour inductively, looking at successor states, and working upwards in the game tree. Here is a simple case: viz. a typical modal equation defining a *winning strategy* for player **1**

$$s \quad \leftrightarrow \quad \mathbf{win_1} \vee (\mathbf{move_1} \,\&\, \langle\rangle s) \vee (\mathbf{move_2} \,\&\, []s)$$

The two modalities $\langle\rangle$, $[]$ refer to the union of all available moves. Now note this is not an explicit definition of the usual kind, but a recursive or *inductive* one, as s calls s itself. Such definitions are treated systematically in modal languages with *fixed-point operators*. These include propositions $\mu s \bullet \phi(s)$ interpreted semantically – in arbitrary models $\mathbf{M}$ – as the smallest proposition (viewed as a set of states) S for which $S = \{s \mid \mathbf{M}, s \models \phi$, with the proposition letter s set to denote the states in $S\}$. There is a syntactic restriction here: s should only occur *positively* in the formula ϕ – which is indeed the case in backward induction proofs. Then, we may think of S as being computed in *approximation stages*. Starting from the empty set of states, one plugs each current approximation into ϕ as its interpretation for the proposition letter s , until some first stable state is reached. The successive levels of this approximation, well-known in the theory of fixed-point logics, are precisely the successive levels in computing strategic equilibria! Unwinding inductive definitions may be an infinite task: explicit definitions for fixed-point propositions may involve *infinite disjunctions* of all approximations traversed. This may seem strange; until we recall how infinitary modal languages came up naturally with bisimulation invariants in Section 5.3 (whose proof indeed involves fixed-point computation of game descriptions).

7.3 Which languages fit our game equivalences precisely?

In line with the general philosophy of Section 5, we would now have to determine the precise game languages corresponding to our proposed simulations. This is in fact feasible.

What stands to forcing bisimulation as modal logic stands to plain bisimulation?

The answer is the *modal game logic* with the strategy-existence operators $\langle i\rangle\phi$ described above – and the results cited earlier provide the backing. So it remains to determine

What stands to alternating bisimulation as modal logic stands to bisimulation?

This time, the answer is not some pre-existing language – but it is not hard to find either. We can modify the standard modal proofs to see it is a modal language with two operators

$\langle \mathbf{i}, \mathbf{end}\rangle\ \phi$ there exists a finite chain of moves for player **i** (whose turn it is all the time) ending in some final node where ϕ holds

$\langle \mathbf{i}, \mathbf{switch}\rangle\phi$ there exists a finite chain of moves for player **i** (whose turn it is all the time) ending in some node where another player gets the turn and ϕ holds

One independent way of seeing whether these two restricted languages are natural is to determine how many natural game-theoretic arguments can be formalized in this way.

8 What are natural game combinations?

Simulation invariance and definability are not the only route to logical structure of games. For instance, we can also ask what are natural operations forming new games out of old, and what are their laws? An answer will tell us something about the compositional structure of games at the same time. This approach fits in with what we did so far, but it also raises some further issues of its own, which we mention to put the preceding in perspective.

8.1 From internal description to external combination

The logic games in Section 3 suggested a view of logical constants as operations on games. E.g., negation dualizes a game, while disjunction takes two games, and creates a new one by putting them both under a common starting node where it is Verifier's turn to choose. Thus, the perspective changes. We are not primarily interested in the internal structure of single games, but in a calculus for combining them. In linguistic terms, then, we move from *internal* languages of games to *external* ones. For instance, in logical evaluation games, propositional or first-order formulas interpreted as games are such an external language, with the standard logical operations providing game structure. This perspective is quite general, also in process theories (cf. Section 5). Instead of developing modal languages for describing what goes on inside single process graphs, one can also design languages that talk about combining process graphs, such as the major formalisms of 'Process Algebra'. To be sure, the process equivalences of Section 5 still play a crucial role here. Usually, one only wants to consider operations that *respect bisimulation*, in the sense that their values are well-defined even when we replace some arguments by bisimilar ones. There is some interesting model theory on this, which would also be relevant, mutatis mutandis, to games. For instance, which external game operations respect the above forcing bisimulations? Admittedly, the distinction internal/external is fluid. One single game naturally contains many subgames arranged in compositional patterns. Thus, by understanding internal structure, we also understand game combinations, and vice versa. But the thrust differs...

8.2 Game algebra: proliferation of logical operations

As we have seen with our Boolean algebra example in Section 3.1, the usual laws of logic contain an algebra of equivalent games. Of course, the exact calculus that emerges will depend on two decisions. First, we must define our conception of games in terms of some underlying invariance. Next, we must determine *which operations* we wish to consider. The logic games of Section 3.4 had *choice*, *dual*, and *sequential composition.* But these are certainly not the complete natural repertoire for games. For instance, consider conjunction. One way of playing this was Boolean choice: **F** chooses at the start, and afterwards only one of the two games is played. If we want to play both games we need rather composition: first one, then the other (notice the order dependence). But even this may not be right. Many academics play *sequential* composition (first 'career', then 'happy family'), whereas they later wish they had played a *parallel game composition*: making progress in both.

Parallel compositions of various kinds exist in process theories of concurrency, and also in the literature on logic games, e.g. for *linear logic*. One good example is parallel disjunction of games **G+G'**, where one plays episodes from each game, resuming where one left off. What is essential here is not (just) the initial choice between G and G', but rather, which player is allowed the initiative in *switching* between these subgames. The laws of such additional operations can be very interesting, even when they do not correspond quite to any item in standard logic. A popular example from the literature on logical games is this. *"If I can play a chess game once as Black and once as White, copying the moves of my opponent, I can produce two identical runs of both games, one of which will be a win for me, or both are draws. This way I can play against Kasparov, and never come out a loser."* Therefore, for parallel (as opposed to Boolean) game disjunction, Excluded Middle holds, even when the underlying games are not determined! On the other hand, parallel disjunction does not satisfy all classical properties of Boolean $\vee$. In particular, it is not idempotent. E.g., G+G is not the same game as G, because I can use your moves against you in the first, but not in the second. Parallel game operations like this are studied in the current literature on *linear logic*, which provides complete calculi for a natural repertoire of them. But the field seems much richer than what has been explored so far, even when we stick with just logic games. E.g., the Ehrenfeucht-Fraïssé games of Section 4.3 can also be analysed with some profit as combinations of evaluation games in the two models being compared. But this kind of game combination has not yet been studied systematically at all.

8.3 Example: operations in dynamic game logic

For greater concreteness, we return to forcing bisimulation and dynamic game logic. Parikh's language is actually a bit different from what we presented, because it allows reference to many games at the same time. Thus we really have a two-level language, with both dynamic game expressions and static assertions. Their interplay shows in the truth condition for the above strategy modalities:

$\langle G, \mathbf{i}\rangle\phi$ player **i** has a strategy in game G for forcing a set of outcome states all of which satisfy the assertion ϕ

To interpret this, we now think of models, not as single games, but as *game boards*, where many games can be played at the same time. In the simplest case, these are sets of states W on which we have a number of atomic forcing relations $\rho_G{}^{\mathbf{i}}\, x, Y$ for each player and game. Then the above explanation amounts to the following technical clause – involving both forcing relations for games G and truth values for assertions ϕ :

$\mathbf{M}, x \models \langle G, \mathbf{i}\rangle\phi$ iff there exists a set Y with $\rho_G{}^{\mathbf{i}}\ x, Y$ & $\forall y \in Y\ \mathbf{M}, y \models \phi$

The most interesting aspect, however, are the stipulations that compute the forcing relations for compound games – as these fix their meaning, in this rather rough input-output format. We assume that these relations are closed under supersets: if $\rho_G{}^{\mathbf{i}}\ x, Y$ and Y' contains Y, then $\rho_G{}^{\mathbf{i}}\ x, Y'$. Also, we think of two players, to demonstrate how the recursion works:

$\rho^1{}_{G\vee G'}\ x, Y$	iff	$\rho^1{}_G\ x, Y$ *or* $\rho^1{}_{G'}\ x, Y$
$\rho^2{}_{G\vee G'}\ x, Y$	iff	$\exists Z, Z'$: $\rho^2{}_G\ x, Z$ *and* $\rho^2{}_{G'}\ x, Z'$ and $Y = Z \cup Z'$
$\rho^1{}_{\neg G}\ x, Y$	iff	$\rho^2{}_G\ x, W-Y$
$\rho^2{}_{\neg G}\ x, Y$	iff	$\rho^1{}_G\ x, W-Y$
$\rho^1{}_{G\bullet G'}\ x, Y$	iff	$\exists Z$: $\rho^1{}_G\ x, Z$ & $\forall z \in Z\ \rho^1{}_{G'}\ z, Y$
$\rho^2{}_{G\bullet G'}\ x, Y$	iff	$\exists Z$: $\rho^2{}_G\ x, Z$ & $\forall z \in Z\ \rho^2{}_{G'}\ z, Y$

In case all games are *determined*, as with Parikh, we can simply define forcing relations for player **1**, simplifying the dual clause to $\rho^1{}_{\neg G}\ x, Y \Leftrightarrow$ *not* $\rho^1{}_G\ x, W-Y$.

Remark As noted before, forcing relations do not give the underlying strategies explicitly. We could do a richer semantics assigning to any game the set of its associated strategies. Then we get explicit versions of game logics, bringing out the fact that a valid principle of dynamic game logic like $\langle G\bullet G'\rangle\phi \leftrightarrow \langle G\rangle\langle G'\rangle\phi$ really rests on *composition* of strategies.

8.4 Game algebra

Calculi of valid principles tend to show the behaviour of a game semantics most vividly. We state some simple observations, extending earlier remarks. Identifying games under forcing bisimulation makes all laws of *Boolean Algebra* valid except Excluded Middle and Non-Contradiction. If we also look at composition, we find the following additional laws:

$$G \bullet (G' \bullet G'') = (G \bullet G') \bullet G''$$
$$(G \vee G') \bullet G'' = (G \bullet G'') \vee (G' \bullet G'')$$
$$\neg\neg G = G$$
$$\neg (G \bullet G') = \neg G \bullet \neg G'$$

The first of these are reminiscent of Relational Algebra (reading • as composition), though game negation over composition certainly behaves very differently from Boolean negation! But importantly, we lack a converse distributive law

$$G \bullet (G' \vee G'') = (G \bullet G') \vee (G \bullet G'')$$

which cannot hold in general. (A quick counter-example in logic games: substitute some universal quantifier game for G , and get a clearly invalid first-order game equivalence.) The latter failure is reminiscent of current process algebras, and our conjecture is that game logic based on forcing bisimulation is precisely *basic process algebra* plus the stated negation laws. Many other laws emerge once we add further (parallel) game operations.

9 So: what is basic game logic?

The answer is: 'it all depends'. If we describe games with simulations and invariance, there is a spectrum of plausible candidates. Hence there is a corresponding spectrum of internal languages for these invariances, whose complete sets of validities will differ in form and intent – including modal logic, temporal logic, dynamic game logic, and first-order logic. If we describe games via their compositional structure, there is now a choice of external languages, including algebraic equations in different vocabulaires, dynamic game logic once again, linear logic, or category theory. In a sense, then, 'the basic logic of games' is as non-existent as 'the basic logic of space'. We are dealing with a rich phenomenon that can be studied at various grain sizes, with many formalisms whose laws make different points.

10 Incomplete information

10.1 Knowledge and ignorance in games

The above analysis was all about games with perfect information, as are most logic games to date. But actual games often involve *ignorance*: players may lack important knowledge when deciding upon a move. This ignorance can take various forms. Even in a game whose players know where they are at each stage, there is 'forward uncertainty' about the future. We do not discuss this, noting merely that this phenomenon fits very well into our earlier modal-dynamic description languages (or modal-temporal ones) – and hence it does not threaten any points made in the body of this paper so far. But in *imperfect information games*, there is also 'backwards' or 'sideways' ignorance. Players may not even know where they are! The ubiquity of uncertainty in games has given rise to an independent invention of epistemic logic in game theory. Even so, it is not always clear how games with imperfect information are to be interpreted. As it happens, such games have also emerged inside logic itself (cf. Section 3.6), as a means of interpreting languages with quantifiers that can be informationally independent from each other. This provides a clear-cut case for analysis – though the subject is still controversial. In any case, the brief discussion of epistemic action logic in this Section is just meant to illustrate how the concerns of our paper extend systematically to more general games where knowledge and ignorance are essential.

10.2 Information-friendly evaluation games

Recall that in the evaluation game for a formula $\forall x\ \exists y/x\ Rxy$, player **V** had to make her choice independently from **F**'s opening move. Consider the game tree for the 2-element model of Section 3.6, which was non-determined. The dotted line indicates the uncertainty relation for **V** – apart from this players have no uncertainty at any state.

	F		
x:= s	....	x:= t	
V		**V**	
y:= s	y:= t	y:= s	y:= t
win_F	win_V	win_V	win_F

These game trees with arrows and dotted lines are precisely right for a combination of the earlier *modal logics* (having action modalities [a]) with *epistemic logic* (having knowledge modalities K_x). A player knows ϕ if this formula is true at all states connected to the current one by a dotted line (indexed for x if we wish to be precise – and more general). In our example, **F** knows exactly what is true at each state, but – rather subtly:

> **V** knows *that* she have a winning move 'de dicto': $K(\langle y{:=}s\rangle \underline{win} \vee \langle y{:=}t\rangle \underline{win})$
> but she lacks the know-how *which one* ('de re'): $\neg K\langle y{:=}s\rangle \underline{win}\ \&\ \neg K\langle y{:=}t\rangle \underline{win}$

Insisting that players have a strategy which they know to be winning amounts to having a *uniform strategy* in Hintikka's terms. Here is another example. People often think that the evaluation game for $\forall x\ \exists y/x\ Rxy$ is equivalent to the standard game for $\exists y\ \forall x\ Rxy$, since **V** has the same uniform strategies in both. We picture the latter game tree:

	V		
x:= s		x:= t	
F		**F**	
y:= s	y:= t	y:= s	y:= t
win_F	win_V	win_V	win_F

The two trees are not equivalent. The latter is determined, unlike the former – and this difference shows in **F**'s knowledge about actions. Reasoning about (non-)equivalence of such game trees occurs in the game-theoretic literature as well. The advantage of the logic games is their more constrained setting for checking intuitions. We can also apply the combined language to deal with other phenomena in imperfect information games, such as *signalling*.

10.3 Epistemic action logic

More technically then, we have an epistemic action logic with the following semantics:

(a) $\mathbf{M}, s \models [a]\phi$ iff ϕ *holds in all* R_a*-successors of* s (the succesful output states of completed executions for action a starting from the input state s),
(b) $\mathbf{M}, s \models K_x\phi$ iff ϕ *holds in every* $\sim_x$*-successor state of* s (i.e., in all states that x considers epistemically indistinguishable from s).

For simplicity, we think of process graphs here with additional indistinguishability relations $\sim_x$ for each player x (the latter are equivalence relations). Similar systems were proposed in AI in the 1980s in the analysis of *planning*. (Think of the delicate combination of changing information and physical action when a professional burglar breaks a safe.) In a language like this, one can make interesting combined assertions such as

$K_i[a]B$ player i knows that doing a will bring about that B
$[a]K_iB$ doing a will make player i know that B

This leads naturally to the question whether there are significant valid interactions here. Consider the following potential axiom, of a type familiar from the modal literature:

$K_i[a]B \rightarrow [a]K_iB$ which says that, if i knows that doing a will bring about B, then doing a will make her know that B

On the above models it says (by some standard modal correspondence analysis) that

$$\forall xyz: ((y \sim_i z \ \& \ x R_a y) \rightarrow \exists u: ((x \sim_i u \ \& \ u R_a z) \qquad \#$$

10.4 Epistemic game bisimulation

Interestingly, condition # can be pictured as one half of a *bisimulation diagram*:

$\sim_i$ is a bisimulation w.r.t. the converse action relation $R_a{}^{\vee}$

What are appropriate bisimulations for incomplete information games? We will not pursue this very far, but mention some points. For a start, principle # does not seem reasonable.

E.g., it fails in the above game tree: **V**'s indistinguishable states have different pasts. Vice versa, we can analyze constraints on games from the literature, and look for bisimulations. E.g., in games with imperfect recall, genuinely indistinguishable states must not allow players to find out where they are in an indirect manner – say, by inspecting their available actions. Thus, any move for player i at x must also be available at any $y \sim_i x$. Moreover, we might want to say that further inspection should not reveal any differences either. Analyzing this recursively, we again get bisimulation, but now for the forward action R_a :

$$\forall xyz: ((x \sim_i y \ \&\ y\ R_a\ z) \rightarrow \exists u: ((x\ R_a u\ \ \&\ u \sim_i z)$$

This corresponds to a converse axiom $[a]K_iB \rightarrow K_i[a]B$. But again this seems too strong. There is merely an intuition in some signalling arguments that indistinguishable states should have the same atomic actions available. This would be a rather weak form of bisimulation, not requiring a new bisimulation link between the results of those actions.

Remark. That bisimulation comes up here again shows the ubiquity of this semantic notion. But note that we are now talking 'internal bisimulation' inside one incomplete information game – rather than 'external bisimulation' (as in Section 6) between two different games. As for the latter, one of the interesting notions of game equivalence to be investigated now would measure equivalence in the epistemic language over the game logic of Section 6.2.

10.5 Case study: the Thompson transformations

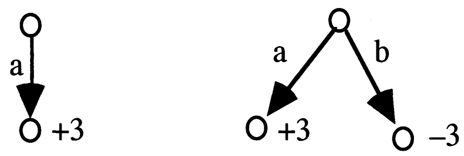
In game theory, there are well-known sets of transformations that are supposed to transform incomplete information games to equivalent 'normal forms'. These provide an interesting illustration of most themes in this paper. A glance at the example in Osborne & Rubinstein, pp. 210/211, will reveal the following high-lights – when stated in the present terms.

(a) Some transformations have nothing to do with knowledge and ignorance. E.g., finite trace equivalence underlies 'co-alescing moves' (with some re-encoding of the action type).
(b) 'Interchanging Moves' is almost literally a simple variant of the earlier propositional Boolean distribution law in game-theoretic guise.
(c) Some transformations do involve knowledge essentially, but it is very unclear what statements of epistemic action logic are supposed to be preserved across them.
(d) The eventual equivalence between the first game tree and the last can be validated via a forcing bisimulation, showing that both players have the same outcome sets before and after the transformations have been applied.

Of course, this is not a serious discussion: just a pointer at what needs to be investigated, if we are to make a significant linkage between game theory and logic at this interface.

11 From game theory to logic

The analysis in this paper has not touched what some may view as the most characteristic aspects of games – such as players' preferences, cooperative equilibria, and probabilistic aspects. All of these are relevant to our analysis, and indeed, we think that vice versa, logic stands to gain a lot from game-theoretic connections here. For instance, reasoning is a social activity, which is goal-driven by preferences of participants. Standard logical models just do not capture such finer aspects. As for preferences, we do think our analysis extends in a straightforward manner. Epistemic action logic can easily support preference structure. But as before, one must also make up one's mind. Consider the following two games:



In one, the player can perform action a only, with an outcome of 3 guilders, in the other, she can also perform action b , and lose 3 guilders. Are these 'the same game'? "Yes", if we insist on rational behaviour, but "no" if we see the games as activities, in which players have a chance of displaying non-calculating behaviour, such as choosing 'noble tragedy'.

12 Conclusion

This paper has the following main line. We have surveyed some essentials of *logic games*, as these seem an interesting 'pure laboratory case' for the study of general game structure. This allowed us to bring out some basic phenomena, such as game equivalence, operational structure, and calculi of game laws. This special domain also showed how ubiquitous games can be. Thus, we advocated the use of games for comparing games – a circularity which logicians know and love, but which may yet have to be appreciated by others! Following that, we have developed an analogy between *games and processes*, viewing game theory as a more refined version of current process theories. This involved looking for a spectrum of natural game simulations, especially, appropriate versions of bisimulation, definability in matching internal languages, and a foray into game algebra of various sorts.

Finally, we have indicated how some of these recommendations fare when *richer game structure* is taken into account, in particular, knowledge & ignorance. In all this, we have not proposed any fixed solution – just a style of thinking about the issues.

Even at this preliminary stage, we feel that these are interesting times. The contemporary confluences between game theory and logic point at a *third mathematization of games*. After the early, probabilistic phase , and the work of van Neuman & Morgenstern and Nash, we now have 'information flow in action' as a focus for mathematization – which seems an obvious shared concern of logicians and game theorists. But, the new mathematization will involve much more than the semantic issues raised in this paper. There is an equally important issue of computational power. If games are the cognitive activity par excellence, what is their 'problem solving power'? It will take a Turing of the field to answer *that*.

13 References

J. Barwise & L. Moss, 1997, *Vicious Circles*, CSLI Publications, Stanford.

A modern source on modal description of structures up to bisimulation.

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

An exploration of general process theories for action and cognition.

J. van Benthem, 1998–99, Lo*gic and Games*, www.turing.wins.uva.nl/~johan/

Lecture notes on all topics discussed in this paper, plus a lot more.

K. Doets, 1996, *Basic Model Theory* , CSLI Publications, Stanford.

A thorough introduction to model comparison games.

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

The original thoughts from one of the great pioneers.

W. Hodges, 1998, *An Invitation to Logical Games*, Queen Mary's College, London.

Another set of lecture notes on logic and games, with different music.

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

The text-book on game theory that logicians love at first sight.

institute *for* logic, language *and* computation

ILLC Scientific Publications

Coding for Reports and Dissertations: *Series-Year-Number*, with CT = Computation and Complexity Theory (up to 1998); LP = Logic, Philosophy and Linguistics (up to 1998); ML = Mathematical Logic and Foundations (up to 1998); PP = Prepublication Series (since 1999); X = Technical Notes; MoL = Master of Logic Thesis; DS = Dissertations.
All previous ILLC-publications are available from the ILLC bureau. For prepublications before 1994, contact the bureau.

CT-1997-01 Carl H. Smith, Rūsiņš Freivalds *Catagory, Measure, Inductive Inference: A Triality Theorem and its Applications*

CT-1997-02 Peter van Emde Boas *Resistance is Futile; Formal Linguistic Observations on Design Patterns*

CT-1997-03 Harry Buhrman, Dieter van Melkebeek *Complete Sets under Non-Adaptive Reductions are Scarce*

CT-1997-04 Andrei Muchnik, Andrei Romashchenko, Alexander Shen, Nikolai Vereshagin *Upper Semi-Lattice of Binary Strings with the Relation "x is simple conditional to y"*

CT-1998-01 Hans de Nivelle *Resolution Decides the Guarded Fragment*

CT-1998-02 Renata Wassermann *On Structured Belief Bases - Preliminary Report*

CT-1998-03 Johan van Benthem *Temporal Patterns and Modal Structure*

CT-1998-04 Ghica van Emde Boas-Lubsen, Peter van Emde Boas *Compiling Horn-Clause Rules in IBM's Business System 12 - an Early Experiment in Declarativeness*

CT-1998-05 Carlos Areces, Maarten de Rijke *Expressiveness Revisited*

CT-1998-06 Jan van Eijck *Programming with Dynamic Predicate Logic*

CT-1998-07 Frans Voorbraak *Optimising and Satisficing under Partial Ignorance*

LP-1997-01 Johan van Benthem *Dynamic Bits and Pieces*

LP-1997-02 Paul Dekker *On Denoting Descriptions*

LP-1997-03 Paul Dekker *On First Order Information Exchange*

LP-1997-04 Jelle Gerbrandy *Dynamic Epistemic Logic*

LP-1997-05 Jelle Gerbrandy *Bisimulation and Bounded Bisimulation*

LP-1997-06 Jan van Eijck *Typed Logic With States*

LP-1997-07 Michiel van Lambalgen, Jaap van der Does *A Logic of Vision (expansion of LP-1996-14)*

LP-1997-08 Johan van Benthem *Wider Still and Wider... Resetting the Bounds of Logic*

LP-1997-09 Frans Voorbraak *A Nonmonotonic Observation Logic*

LP-1997-10 Jan van Eijck *Dynamic Reasoning Without Variables*

LP-1998-01 Hans Rott, Maurice Pagnucco *Severe Withdrawal (and Recovery)*

LP-1998-02 Jaap van der Does, Helen de Hoop *Type-shifting and Scrambled Definites*

LP-1998-03 Renate Bartsch *The Role of Consciousness and Intentionality in Perception, Semantics, Representations and Rules*

LP-1998-04 Renata Wassermann *Resource Bounded Belief Revision*

LP-1998-05 Johan van Benthem *Linguistic Grammar as Dynamic Logic*

LP-1998-06 Renate Bartsch *The Formal Relationship between Dynamic Conceptual Semantics and Connectionist Neural Network Modelling*

LP-1998-07 Jan van Eijck *Axiomatising Dynamic Logics for Anaphora*

LP-1998-08 Jan van Eijck *Incremental Dynamics*

LP-1998-09 Carlos Areces, Verónica Becher *Iterable AGM Functions*

LP-1998-10 Christof Monz, Maarten de Rijke *A Tableaux Calculus for Ambiguous Quantification*

LP-1998-11 Christof Monz, Maarten de Rijke *A Resolution Calculus for Dynamic Semantics*

LP-1998-12 Dick de Jongh, Paul van Ulsen *Beth's Nonclassical Valuations*

LP-1998-13 Hans Rott *"Just Because": Taking Belief Bases Seriously*

LP-1998-14 Hans Rott *Words in Contexts: Fregean Elucidations*

LP-1998-15 Paul Dekker *Speaker's Reference, Descriptions and Information Structure*

LP-1998-16 Yde Venema *Temporal Logic*

LP-1998-17 Kazushige Terui *Anaphoric Linking at Run Time: A Type-Logical Account of Discourse Representation*

ML-1997-01 Dick de Jongh, Giorgi Japaridze *The Logic of Provability*

ML-1997-02 Maarten Marx *Complexity of Modal Logics of Relations*

ML-1997-03 Giovanna D'Agostino *The Loś-Tarski and Lyndon Theorem for the μ-logic*

ML-1997-04 Ian Hodkinson, Szabolcs Mikulás *Non-finitely axiomatizable, union-free reducts of algebras of relations*

ML-1997-05 Johan van Benthem *The Range of Modal Logic: an Essay in Memory of George Gargov*

ML-1997-06 Johan van Benthem *Modality, Bisimulation and Interpolation in Infinitary Logic*

ML-1997-07 Sebastiaan A. Terwijn, Domenico Zambella *Algorithmic Randomness and Lowness*

ML-1997-08 Antonín Kučera, Sebastiaan A. Terwijn *Lowness for the Class of Random Sets*

ML-1998-01 A.S. Troelstra *Marginalia on Sequent Calculi*

ML-1998-02 A.S. Troelstra *Concepts and Axioms*

ML-1998-03 Hans de Nivelle *Decoding the E^+-Class by an A Posteriori, Liftable Order*

ML-1998-04 Yde Venema *Points, Lines and Diamonds: a Two-Sorted Modal Logic for Projective Planes*

ML-1998-05 Steven Givant and Yde Venema *The Preservation of Sahlqvist Equations in Completions of Boolean Algebras with Operators*

ML-1998-06 Victor N. Krivtsov *A Negationless Interpretation of Intuitionistic Axiomatic Theories: Arithmetic and Analysis*

ML-1998-07 Victor N. Krivtsov *A Negationless Interpretation of Intuitionistic Axiomatic Theories: Higher-Order Arithmetic*

ML-1998-08 Johan van Benthem *Dynamic Odds & Ends*

ML-1998-09 Rosalie Iemhoff *A Modal Analysis of Some Principles of the Provability Logic of Heyting Arithmetic*

ML-1998-10 Carlos Areces, Verónica Becher, Sebastián Ferro *Characterization Results for d-Horn Formulas*

ML-1998-11 Carlos Areces, Dick de Jongh, Eva Hoogland *The Interpolation Theorem for IL and ILP*

ML-1998-12 Johan van Benthem *Modal Logic in Two Gestalts*

ML-1998-13 Yde Venema *Canonical Pseudo-Correspondence*

ML-1998-14 Yde Venema, Vera Stebletsova *Undecidable theories of Lyndon algebras*

PP-1999-01 Maarten Marx *Relation Algebras Can Tile*

PP-1999-02 Maarten Marx *Relativized Relation Algebras*

PP-1999-03 Maarten Marx, Carlos Areces *Failure of Interpolation in Combined Modal Logics*

PP-1999-04 Kazushige Terui *Labelled Tableau Calculi Generating Simple Models for Substructural Logics*

PP-1999-05 Hans Rott *Coherence and Conservation in the Dynamics of Belief, Part I: Finding the Right Framework*

PP-1999-06 Maarten Marx *Amalgamation in Finite Dimensional Cylindric Algebras*

PP-1999-07 Carlos Areces, Patrick Blackburn, Maarten Marx *Hybrid Logics. Characterization, Interpolation and Complexity*

PP-1999-08 Rosalie Iemhoff *On the Admissible Rules of Intuitionistic Propositional Logic*

PP-1999-09 Jaap Kamps, László Pólos *Reducing Uncertainty: A Formal Theory of Organizations in Action*

PP-1999-10 Eva Hoogland *Algebraic Characterizations of Various Beth Definability Properties*

PP-1999-11 Paul Dekker *The Semantics of Dynamic Conjunction*

PP-1999-12 Paul Dekker *Coreference and Representationalism*

PP-1999-13 Pieter W. Adriaans *Learning Shallow Context-Free Languages under Simple Distributions*

PP-1999-14 Christof Monz, Maarten de Rijke *A Tableau Calculus for Pronoun Resolution*

PP-1999-15 Maarten Marx, Stefan Schlobach, Szabolcs Mikulás *Labelled Deduction for the Guarded Fragment*

PP-1999-16 Carlos Areces, Hans de Nivelle, Maarten de Rijke *Prefixed Resolution: A Resolution Method for Modal and Description Logics*

PP-1999-17 Renata Wassermann, Sven Ove Hansson *Local Change*

PP-1999-18 Renata Wassermann, Eduardo Fermé *A Note on Prototype Revision*

PP-1999-19 Jeroen Groenendijk *The Logic of Interrogation (classical version)*

PP-1999-20 Eva Hoogland, Maarten Marx, Martin Otto *Beth Definability for the Guarded Fragment*

PP-1999-21 Nikos Massios, Frans Voorbraak *Hierarchical Decision-Theoretic Robotic Surveillance*

PP-1999-22 Johan van Benthem *When are Two Games the Same?*

PP-1999-23 J.A.G. Groenendijk *Questions*

PP-1999-24 Marco Aiello, Johan van Benthem *Logical Patterns in Space*

X-1997-01 Paul Dekker, David Beaver *Report on ECDS: An Interactive Course on the Internet*

X-1997-02 Dimiter Ivanov Vakarelov *Applied Modal Logic: Modal Logics in Information Science*

X-1998-01 Ghica van Emde Boas-Lubsen *Feature Analysis of Business System 12*

X-1998-02 Paul Harrenstein *From a Modal Point of View: a Logical Investigation into Modalities De Dicto and De Re*

X-1998-03 Peter van Emde Boas *Formalizing UML; Mission Impossible?*

X-1999-01 Boudewijn P. de Bruin *Game Transformations and Game Equivalence*

MoL-1997-01 Dimitris Dimitriadis *Identity and Identification*

MoL-1997-02 Brian Semmes *The Raisonnier-Shelah Construction of a Non-Measurable Set*

MoL-1997-03 Marc Pauly *Transforming Predicates or Updating States? Total Correctness in Dynamic Logic and Structured Programming*

MoL-1998-01 Rosella Gennari *Temporal Reasoning and Constraint Programming*

MoL-1998-02 Paulo E. Santos *Formalising the Common Sense of a Mobile Robot*

MoL-1998-03 Eyal Hurvitz *Towards a Uniform Strategy of Plural Quantification? Iterative Cumulativity and the Witness Condition*

MoL-1999-01 Alessandra Palmigiano *Normal Gentzen Deductions in the Classical Case*

DS-1997-01 Ronald Cramer *Modular Design of Secure yet Practical Cryptographic Protocols*

DS-1997-02 Nataša Rakić *Common Sense Time and Special Relativity*

DS-1997-03 Arthur Nieuwendijk *On Logic. Inquiries into the Justification of Deduction*

DS-1997-04 Atocha Aliseda-LLera *Seeking Explanations: Abduction in Logic, Philosophy of Science and Artificial Intelligence*

DS-1997-05 Harry Stein *The Fiber and the Fabric: An Inquiry into Wittgenstein's Views on Rule-Following and Linguistic Normativity*

DS-1997-06 Leonie Bosveld - de Smet *On Mass and Plural Quantification. The Case of French 'des'/'du'-NP's.*

DS-1998-01 Sebastiaan A. Terwijn *Computability and Measure*

DS-1998-02 Sjoerd D. Zwart *Approach to the Truth: Verisimilitude and Truthlikeness*

DS-1998-03 Peter Grunwald *The Minimum Description Length Principle and Reasoning under Uncertainty*

DS-1998-04 Giovanna d'Agostino *Modal Logic and Non-Well-Founded Set Theory: Translation, Bisimulation, Interpolation*

DS-1998-05 Mehdi Dastani *Languages of Perception*

DS-1999-01 Jelle Gerbrandy *Bisimulations on Planet Kripke*

DS-1999-02 Khalil Sima'an *Learning efficient disambiguation*

DS-1999-03 Jaap Maat *Philosophical Languages in the Seventeenth Century: Dalgarno, Wilkins, Leibniz*