What Logic Games are Trying to Tell Us

December 2002, Johan van Benthem, Amsterdam & Stanford

Abstract We present logic games as a research topic in its own right, providing attractive models for dynamic interaction between agents. First, we survey the most basic logic games. Then we show how their common properties raise interesting general issues of possible structures for games and corresponding 'game logics'. Next, we review logic games in the light of this general game logic. Finally, we discuss what happens when we import more 'realistic' themes from game theory into logic games, including players' preferences, and imperfect information.

Contents

1 From products to activities: logic in games

1.1 Logical Dynamics Logic is usually taken to be about propositions, proofs, truth, and consequence: i.e., abstract objects in Heaven and their Platonic properties. But the discipline arose in Greek Antiquity by studying *activities* that take place on Earth: claims, dialogue, argumentation, and debate. Indeed, the very terms of logic still have a double meaning. A 'statement' is both a dynamic activity and the static product of that activity, a 'proof' is both a dynamic procedure of establishing some claim and the static formal record of that procedure, and so on. These activities are usually kept in the background, as part of the didactical motivation of the field, but not part of its eventual scientific content. Taking the dynamic action aspect more seriously as an essential feature of logical theory has led to so-called 'logical dynamics' – cf. the programmatic exposition in van Benthem 1996.

1.2 Logic games The best example inside modern logic of dynamic activities with a clear formal structure are *logic games*. These have been around at least since the Middle Ages, with the stylized 'Obligatio' tasks described by medieval authors like Walter Burleigh. More mathematical logic games were proposed in the 1950s by Lorenzen, Ehrenfeucht/Fraïssé, and Hintikka – not accidentally in conjunction with the new wave in game theory for which Harsanyi, Nash, and Selten received their Nobel prize in 1993. Today, there are various two-person games for performing basic logical tasks, such as evaluation of propositions in a given model, comparing given models, building models for given assertions, or constructing proofs for given claims. And new varieties are still appearing. For a recent survey including many references, cf. the lecture notes van Benthem 1999 – 2002. This paper is a revision of van Benthem 1999, a panorama of logic games in a dynamic perspective.

The aim of this paper is to give an impression of what happens when we take logic games seriously in their own right as dynamic activities, which deserve a key place in logic as such. We refer to a number of recent publications, trying to convey an over-all perspective. To set the scene, let us first look at a bunch of logic games.

2 The basic logic games

2.1 Gamification Current logic games are diverse. In principle, any logical task can be 'gamified', provided we find a way of pulling it apart into opposing roles for two players whose dynamic interaction tests the notion involved. This works as soon as we can discern a quantifier pattern, with the 'Attacker' being the universal quantifier ***A*** ('All', 'Adam', 'Alter') and the 'Defender' being ***E*** ('Exists', 'Eve', ''Ego').

For instance, here is how Leibniz already explained the meaning of the sometimes tricky quantifier forms $\forall\varepsilon\,\exists\delta$ occurring with continuity of mathematical functions:

> A challenger $\forall$ chooses some number ε at his discretion,
> and the defender $\exists$ has to produce a suitable response δ.

In what follows, we give some brief sketches of major logic games, referring to the literature for more detailed exposition and references. The list may look a bit enumerative and sketchy, but our main purpose is to make the reader aware of the ubiquity of the phenomenon, and have some examples at hand for the more systematic considerations later on in this paper. For a much more comprehensive account of logic games and their connections, cf. again van Benthem 1999 – 2002.

2.2 Argumentation Perhaps the oldest, and still widely appealing, example of a logic game is *argumentation*, where one makes a claim against an opponent, upholding it in the face of various objections. We all experience its game-like character of having to say the right thing at the right time, and also, the bitter taste of defeat when we have talked ourselves into a corner, contradicting our own earlier statements. The latter situations are the typical losing stages in argumentation – being at the same time wins for the other player. Precise dialogue games for argumentation, in the style of Lorenzen, may be found in Rahman & Rueckert 2001.

Here is the key game-theoretic feature of argumentation – which probably led to the discovery of logical patterns of reasoning in the first place. Roughly speaking,

> *Logically valid propositions* ϕ will be precisely those
> for which their proponent Eve has a *winning strategy*:

that is, a way of choosing her conversational moves against the opponent Adam which guarantees that Eve never reaches a losing situation – no matter with which moves and in which order Adam attacks her claim. More concretely, Eve's winning strategy are the dynamic counterpart of a logical *proof* for the proposition ϕ Of course, there may be more than one correct proof for a claim, which reflects the dynamic diversity of behaviour: players may have more than one strategy to win a given game. This general style of analysis holds across a wide range of logic games.

Incidentally, not all propositions are valid, so there must also be games of this sort where Eve is wrong, and it is rather the opponent Adam who has a winning strategy, i.e., a guaranteed method for involving Eve in self-contradictions. What do the latter argumentative strategies look like, viewed on their own merits? Section 2.4 below takes up this matter. Players are on a par in games, and there is no need to glorify Eve over Adam. It is their interaction which drives the logical analysis.

2.3 Obligatio Much conversation is not about proof, but about consistency. We tend to believe things people tell us – even implausible ones like "I love you" – as long as they maintain consistency. Only under very special circumstances will we be called upon to prove the assertions we make. This happens, say, in a juridical procedure, or when teaching mathematics. But just maintaining consistency is itself a major logical task! Medieval training disputations often had a form like this:

> Eve has to maintain consistency when Adam confronts her with successive assertions ϕ, each of which she has to accept or reject. In the former case, ϕ is added to her cumulative store of commitments, otherwise $\neg\phi$ is added. Eve loses if at any stage, her commitments become inconsistent.

One may ask whether this sort of logic exam is fair. And indeed, in principle,

> Eve always has a *winning strategy*, being a string of YES/NO answers to any sequence of propositions, keeping her commitments consistent.

More precisely, Eve's strategies for passing the exam may be correlated with *logical models*: situations that make all her commitments true. Of course, these strategic insights are also the logical skills that help people when misinforming and lying. For details of medieval disputations involving Eve's initial background knowledge – and an option of giving a third response of 'doubt' – cf. Dutilh-Novaes 2002.

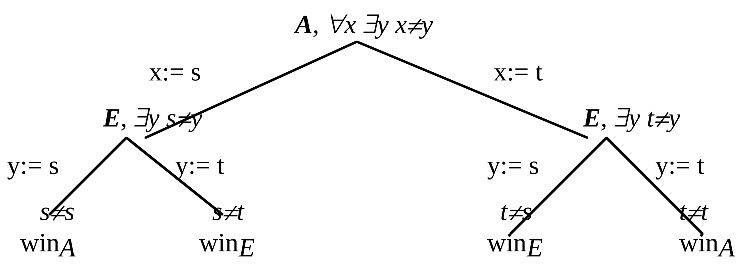
2.4 Model checking But arguably, the most basic logic game occurs when an assertion ϕ is made about a given first-order model ***M*** and variable assignment *s*. A game-theoretic account of the semantic evaluation procedure lets Eve act as a 'Verifier' claiming the truth of the assertion ϕ, and Adam as a 'Falsifier' defending its falsity. The resulting game *game*(ϕ, ***M***, *s*) works as follows.

> With atomic formulas ϕ , one *checks* who is right about ***M***, *s*, and that player wins. Disjunctions are a *choice* for Verifier, and then play continues with the disjunct chosen. Conjunctions are an initial choice for Falsifier. Negations lead to a *role switch*, with all turns interchanged between players. Existential quantifiers $\exists x\phi$ start with a move for Verifier who picks a witness object *d* in ***M***, and the game continues with the formula $\phi(d)$. Universal quantifiers are analogous, starting with the choice of a challenge object by Falsifier.

For instance, the evaluation game for the first-order formula

$$\forall x\ \exists y\ x \neq y$$

in a first-order model with two distinct objects *s*, *t* looks lie this:



Note that ***V*** has a winning strategy here. A good account of first-order evaluation games is Hintikka & Sandu 1995, while more sophisticated variants are used in computer science (cf. Stirling 1999). Again, the central logical notion corresponds to the existence of a winning strategy. The following is easy to prove (Hintikka 1973):

Fact For all ***M***, *s*, ϕ, the following assertions are equivalent:

(a) $\boldsymbol{M}, s \models \phi$

(b) Eve has a winning strategy in *game*(ϕ, ***M***, *s*).

Likewise, it is Adam who has a winning strategy in case the formula is false in the given model. This time, different winning strategies correspond to what might be called different *reasons* for the truth or falsity of the assertion ϕ. 'Reasons' of this sort are not standard logical objects – but e.g., with quantifier combinations such as

$$\forall x\ \exists y\ \forall z\ \exists u\ \phi(x, y, z, u)$$

one can think of them as (bunches of) *Skolem functions f(x), g(x, z)* of the right arities providing Eve with her winning response in ϕ to Adam's successive choices.

2.5 Model construction The evaluation task of checking if $\boldsymbol{M}, s \models \phi$ starts from a given model and formula. In a logical satisfiability task of checking consistency, however, only assertions are given, with the question if one can find a model for these. This suggests a *model construction game* between a 'Builder' Eve who tries to create such a model, making some initial assertions true and others false, and a 'Critic' Adam who raises objections, making sure that every building task gets scheduled. In particular, Critic can force Builder to choose when a disjunct is to be made true, and also, he can keep calling new instances of initial universal quantifiers as Builder puts new objects into the model under construction. Builder loses at any stage if her current schedule tells her to make the same formula both true and false. A precise format for the game rules may be read off from the standard satisfiability method of *semantic tableaus* (cf. van Benthem 2001B), with decomposition rules for logical operators as game moves. The result of all this is the following equivalence:

Fact The following assertions are equivalent in tableau construction games:

(a) The set of first-order formulas Σ has a model

(b) Builder has a winning strategy in the construction game for Σ.

Unlike first-order evaluation games, whose depth is bounded by the operator depth of the initial formula, these games can have infinite runs. The reason is that some first-order formulas have only infinite models, requiring that Builder keeps going forever. Again, more precisely, there will be a correlation between Builder's different winning strategies and different *models* (if any) for the given assertions. Games for model construction have been used extensively in Hodges 1985.

If Builder's winning strategies are like models, what about Critic's? The latter amount to guaranteed ways of blocking any construction attempt. It can be shown that these are like *proofs* of the negation of the initial assertion. Thus, as is only natural, the construction game is like an argumentation game. Reversing the perspective, Critic tries to prove some initial assertion, while Builder is looking for a *counter-example*. This also answers our question in Section 2.2 about what the Opponent Adam was doing. His winning strategies are correlated with developing counter-examples to the initial claim put forward by Proponent. Thus, argumentation and model construction games are really two takes on the same logical process.

2.6 Model comparison In addition to model checking and satisfiability testing, there are other basic tasks for a logic system. E.g., we can measure expressive power of such a system by seeing which models can be distinguished by our language. Probably the most widely used logic game performs just this task. *Ehrenfeucht-Fraïssé games* have Eve playing as 'Duplicator', who claims that two given models ***M***, ***N*** are similar, against Adam playing as 'Spoiler', claiming that they are different. Each round of the game starts with a match ***M, a – N, b***, where ***a, b*** are tuples of objects already chosen according to the following stepwise procedure:

> In each round, Spoiler chooses a model ***M*** or ***N***, and an object *x* in it,
> Duplicator then chooses a corresponding object in the other model;
> and the link *x*–*y* is then added to the current match ***a – b***

Duplicator loses whenever the obvious function defined by the current match of objects is no longer a partial isomorphism between the two models. She wins those runs of the game where this failure never occurs. We can play such games over a fixed finite number of rounds, or forever. An excellent standard textbook on Ehrenfeucht-Fraïssé games is Doets 1996. Here is the key result about the method:

Fact For all models ***M***, ***N***, the following are equivalent:

(a) Duplicator has a winning strategy in a *k*-round comparison game

(b) ***M***, ***N*** satisfy the same first-order sentences up to quantifier depth *k*.

In versions of the game without a finite bound, Duplicator's winning strategies correspond to so-called *potential isomorphisms* between the two models. Thus, these strategies have to do with various notions of structural similarity for models. But as above, the viewpoint of the other player is of independent interest. E.g., in the *k*-round game, a winning strategy for Spoiler is a first-order formula ϕ of depth $\leq k$ which is true in ***M*** and false in ***N***. This is closer to the original goal of testing the expressive power of the given language. In another view of the game, then, Spoiler is the player with a positive intention, claiming that the language is rich, whereas his opponent Duplicator maintains that it is poor.

2.7 Other logic games Variant games emerge all the time. A nice example is Hirsch & Hodkinson 2002, which studies the following situation. An abstract relational algebra ***A*** is given, and we want to know if it is representable as an algebra ***S*** of binary set relations over some set of individual objects *U*. Thus, we want to build a model *(U,* ***S****)* standing in some isomorphism-like relationship *E* to the given model ***A*** – mixing ideas from model construction and model comparison games. For this purpose, one lets players create 'networks', which are stages of a representation in progress, with Builder responding to challenges made by Critic. In particular, Critic can force Builder to accept or reject any 'label' ***a*** taken from the algebra for any ordered pair *(s, t)* in the current network. Moreover, in case ***a*** is a composition of two relations ***b*** *;* ***c***, Builder can be forced to create a new object *u* generating new ordered pairs *(s, u)* labelled with ***b*** and *(u, t)* labelled with ***c***. The game is lost for Builder when the current network has a manifest inconsistency in its labelling. The authors show that an abstract relational algebra ***A*** is set-theoretically representable iff Builder has a winning strategy in this representation game over ***A***. They use this to find a perspicuous complete axiomatization of the class of representable relational algebras.

This concludes our survey of logic games. We now turn to their general structure.

3 The unifying role of strategies, and its consequences

3.1 Strategies as a unifying notion Our games all had an Adequacy Theorem stating that some standard notion holds (truth, satisfiability, potential isomorphism) iff some designated player has a winning strategy. Thus, the typically game-theoretic notion of a strategy becomes a unifying idea across logic. This leads to surprising connections. E.g., in one and the same type of game, viz. that for model construction, we may encounter proofs as winning strategies for Critic, and models as winning strategies for Builder. Thus, logically very different notions turn out cousins after all. Sometimes also, strategies are new citizens asking for recognition in logic, such as semantic 'reasons' for truth or falsity in first-order evaluation games. These analogies suggest that underlying logic, there is a calculus of strategies, for combining them and proving their basic properties. We will return to this issue in Sections 6 and 11.

3.2 ∃-sickness, and its cure Despite their broad impact, strategies are not always taken seriously in the usual Adequacy Theorems. Note the existential quantifier in these results. E.g., truth amounts to the existence of a winning strategy for Verifier in an evaluation game, but we are not told precisely what strategies. This is an instance of a much more wide-spread disease in logic, of *∃-sickness*: the wilful hiding of more specific information under existential quantifiers. Sure symptoms of the disease are the use of indefinite articles “a”, or modal affixes “-ility”. We see this with phrases like “having a strategy" ("winnab–ility") in Adequacy Theorems. Another case of ∃-sickness is how a standard completeness theorem relates provability to validity, instead of a more informative match from *proofs* to semantic 'verifications'. Or, how many people study the logic of interpretability rather than that of the interpretations which do the real work. More domestically, e.g., temporal logic reads the past tense in “Lida fell down the stairs” as “at *some time* in the past”, whereas we usually have one particular episode in mind (cf. Barwise & Perry 1983).

Fortunately, the disease is often cured with a little exercise! We will give two illustrations. The first shows how to make an existential quantifier explicit by analyzing a standard proof – in this case, one concerning model comparison games:

Theorem Ehrenfeucht-Fraïssé adequacy explicitized

There exists an *explicit correspondence* between

(a) winning strategies for ***S*** in the *k*-round comparison game for ***M**, **N***

(b) first-order sentences ϕ of quantifier depth *k* with ***M***$|=\phi$ *, not **N*** $|=\phi$

Proof The proof of this result is a simple analysis of the usual Adequacy argument. *From (b) to (a).* Every such difference formula ϕ of quantifier depth *k* defines a uniform winning strategy for Spoiler in a *k*-round game between arbitrary models. Each round *k–m* starts with a match between linked objects chosen so far which differ on some subformula ψ of ϕ with quantifier depth *k–m*. By Boolean analysis, ***S*** then finds some existential subformula $\exists x\bullet\alpha$ of ψ with a matrix formula α of quantifier depth *k–m–1* on which the two models disagree. ***S***'s next choice is a witness in that model of the two where $\exists x\bullet\alpha$ holds. *From (a) to (b*). Each winning strategy σ for Spoiler induces a distinguishing formula of proper quantifier depth. To obtain this, let ***S*** make his first choice *d* in model ***M*** according to σ – and write down an existential quantifier for that object. Our formula under construction will be true in ***M***, and false in ***N***. We know that each choice of Duplicator for a corresponding object *e* in ***N*** gives a winning position for ***S*** in all remaining *k–1*-round games starting from an initial match *d–e*. By the inductive hypothesis, these induce distinguishing formulas of depth *k–1*. Now, the *Finiteness Lemma* for first-order logic over a fixed finite relational signature says that

For any fixed set of free variables and fixed quantifier depth,
only finitely many non-equivalent formulas exist.

In particular, only finitely many of the above distinguishing formulas can occur modulo logical equivalence. Some of these will start with 'their' first quantifier in ***M*** (say A_1, .., A_r) – others in $\boldsymbol{N}$ (say B_1, .., B_s). The total distinguishing formula for strategy σ is then the $\boldsymbol{M}$-true assertion $\exists x \bullet (A_1 \,\&\, .. \,\&\, A_r \,\&\, \neg B_1 \,\&\, ... \,\&\, \neg B_s)$ ■

Thus, Spoiler's winning strategies in a comparison game correspond to formulas, logical objects of independent interest. A similar match exists for Duplicator, but it is a bit harder to say what logical 'objects' correspond to her winning strategies. One might call them 'analogies', of some finite quality measured by the number k. Technically, they are cut-off versions of *potential isomorphisms*.

Of course, even this Theorem is still $\exists$-sick! But the remaining outer existential quantifier in its formulation may be harmless, in that its instantiation is the *proof*.

Our second illustration shows another way of high-lighting strategies, by analyzing their *multiplicity*. Consider a verification game for propositional formulas:

Fact Counting evaluation strategies

One can count the number of verifying (falsifying) strategies, say $\#(V, \phi)$ $(\#(F, \phi))$ for any propositional formula ϕ as follows:

$$\begin{array}{llllll}
\#(\mathbf{V}, p) & = & 1 & \#(\mathbf{F}, p) & = & 1 \\
\#(\mathbf{V}, \neg\phi) & = & \#(\mathbf{F}, \phi) & \#(\mathbf{F}, \neg\phi) & = & \#(\mathbf{V}, \phi) \\
\#(\mathbf{V}, \phi\vee\psi) & = & 2\#(\mathbf{V}, \phi)\bullet\#(\mathbf{V}, \psi) & \#(\mathbf{F}, \phi\vee\psi) & = & \#(\mathbf{F}, \phi)\bullet\#(\mathbf{F}, \psi) \\
\#(\mathbf{V}, \phi\wedge\psi) & = & \#(\mathbf{V}, \phi)\bullet\#(\mathbf{V}, \psi) & \#(\mathbf{F}, \phi\wedge\psi) & = & 2\#(\mathbf{F}, \phi)\bullet\#(\mathbf{F}, \psi)
\end{array}$$

Proof The rationale for the clauses is immediate from the standard definition of strategies. in game trees as functions assigning unique moves to players' turns. ■

Such counting becomes more complex with full first-order evaluation games.

3.3 Strategies: actions or powers? Strategies are stepwise instructions as to how players should act. This represents a very detailed level of game structure, which was suppressed by existential quantifiers of just 'having a strategy'. But upon reflection, one person's $\exists$-sickness may be another's sanity! In games, we are sometimes not interested in details of moves and actions, but just in the *control* that players have over possible *outcomes*. E.g., that Eve has a winning strategy really says that it is within her power – whatever Adam does – to make sure that the game ends in some specific set of runs or outcomes, designated as 'winning'. And players may also have other powers in games: via losing strategies, or strategies that guarantee long runs. Such powers are a natural level for describing influence in social settings. This level of control brings us to a first very general issue raised by logic games: at what level do we want to describe games – in terms of outcomes, or more detailed actions? This is one of many general game-theoretic issues lying behind logic games…

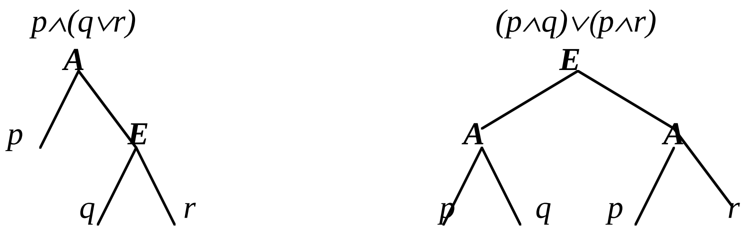
3.4 From logic games to game theory Logic games, though a very specialized class of rather high-brow activities, suggest issues which concern all games: cogent reasoning, playing at cards in smoky bars, or trying to move almost-round leather objects into cage-like structures. Some of these issues arise independently in game theory, others seem new. In what follows, we look at a few of the most pervasive ones. Our heuristics are inspired mainly by the evaluation games of Section 2.4, the most basic logic games. Here, the obvious bridge from logic to game theory are the various Adequacy Theorems of Section 2 – in particular, the one relating the central logical notion of truth with the central game-theoretic notion of a strategy, viz. the existence of a winning strategy for Verifier. Unimaginative logicians interpret such results as a Kiss of Death for the game-theoretic stance, as it provably does not add to what we know from standard logic already. As we shall see, the opposite is true – provided we listen with an unprejudiced ear to what logic games are trying to tell us. For the reader's information, as a counterpart to the logical literature, a compact and lucid background source on game theory is Osborne & Rubinstein 1994.

4 Game equivalences and game languages

4.1 When are two games the same? To raise our question in a simple manner, consider the propositional law of distribution for conjunction over disjunction:

$$p\wedge(q\vee r) \leftrightarrow (p\wedge q)\vee(p\wedge r)$$

The two finite trees in the following figure correspond to evaluation games for the two propositional formulas involved, letting ***A*** stand for Falsifier and ***E*** for Verifier.



This picture raises the following intuitive question

When are two games the same?

In particular, does the logical validity of Distribution mean that the pictured games are the same in some natural sense? Even without formal definitions, the reader may want to check that the answer depends on our viewpoint:

I *If we focus on turns and moves, then the two games are not equivalent:* they differ in ‘etiquette’ (who gets to play first) and in choice structure

This is the level of what game theorists call 'extensive games', with the familiar tree pictures involving details of choices and actions. But there are also 'strategic forms' of games, where we are just interested in listing outcomes that players can control. E.g., the fact that the order of players' turns is reversed by applying Distribution is immaterial then. At the latter level, the answer to our question becomes 'Yes':

II *Both players have the same powers of achieving outcomes in both games*
A can force the outcome to fall in the sets *{p}, {q, r}*
E can force the outcome to fall in the sets *{p, q}, {p, r}*.

Here, a player's *powers* are those sets *U* of outcomes for which (s)he has a *strategy* making sure that the game will end inside *U*, no matter what the other player does. On the left, ***A*** has strategies "left" and "right", yielding powers *{p}* and *{q, r}*. Player ***E*** also has two strategies, yielding powers *{p, q}, {p, r}*. On the right, ***E*** has two strategies "left" and "right", which give the same powers for ***E*** as on the left. By contrast, player ***A*** now has four strategies, which may be written ad-hoc as

"left: L, right: L", "left: L, right: R", "left: R, right: L", "left: R, right: R"

The first and fourth give the same powers for ***A*** as on the left, while the second and third strategy produce merely weaker powers subsumed by the former.

4.2 Game equivalences and game languages The more general issue behind the example is what are natural equivalences between games, setting coarser or finer levels of detail. This topic is investigated extensively in van Benthem 2002, using an analogy with process theories of computation, and *process equivalences* such as modal bisimulation. In particular, the preceding outcome–control views are like black-box input-output views of processes, whereas extensive games lie closer to views of computation endowing processes with richer internal states, including choices. Thus, structure theory of games is like general multi-agent process theory.

As is usual in mathematics, invariance relations between mathematical structures are just one side of a coin. The other side are the relevant properties of games that one wants to distinguish. E.g., for strategic games, it does not matter in which schedule ***E***, ***A*** took their turns: for extensive games, this is a relevant property. Such properties are expressed in *game languages* appropriate to the chosen 'equivalence level'. E.g., as shown in van Benthem 2002, a good language for describing extensive games is modal logic, as well as fixed-point extensions like the modal μ–calculus.

4.3 Logic and games: the plot thickens But then, we have reached a delicate point. Logic now enters our story in a different guise, because different description levels for games correspond to different formal languages with their associated logics. But in the latter mode, we are using logic to talk about general games. Thus, in one of those happy Hegelian inversions, in addition to *logic games*, there are also *game logics*. How the two live together will become clearer as we proceed.

5 Players' powers and modal forcing languages

In this section, we focus on the input–output level of outcomes and players' powers – broadening connections between logic and game theory.

5.1 Determinacy Section 5.4 stated that Verifier has a winning strategy if the relevant formula ϕ is true, and by the same light, Falsifier has a winning strategy if ϕ is false. This means that these games have an important game-theoretical property:

Fact Evaluation games are *determined*: one of the two players in *game* $(\phi, \boldsymbol{M}, s)$ must always have a winning strategy.

A general proof of this uses Zermelo's Theorem which says that all zero-sum two-player games of *finite depth* are determined. Stronger results include the Gale-Stewart Theorem: all infinite games with topologically open winning conditions for one player are determined. The latter result explains why logic games of model construction or model comparison are determined, even though their runs may be infinite. Critic (Spoiler) have open winning conditions, as Builder's (Duplicator's) failures always arise (if at all) at some finite stage. The strongest relevant result in this line is Martin's Theorem, which says that all infinite games are determined which have a winning condition lying in the Borel Hierarchy of sets. Non-open Borel winning conditions occur occasionally with logic games in computer science. Examples include *fairness* of runs for interactive game systems.

With non-Borel winning conventions, infinite games can become non-determined. We display one example, to make a general point about players' powers later on. Take any free ultrafilter $\boldsymbol{U^*}$ on the natural numbers, using the Axiom of Choice:

> Two players pick successive neighbouring closed intervals of natural numbers, of arbitrary finite sizes, producing a succession like this:
>
> $\boldsymbol{A}$: $[0, n_1]$, with $n_1>0$, $\boldsymbol{E}$: $[n_1+1, n_2]$, with $n_2> n_1+1$, etc.
>
> $\boldsymbol{E}$ wins if the union of all her intervals is in $\boldsymbol{U^*}$, otherwise, $\boldsymbol{A}$ wins.

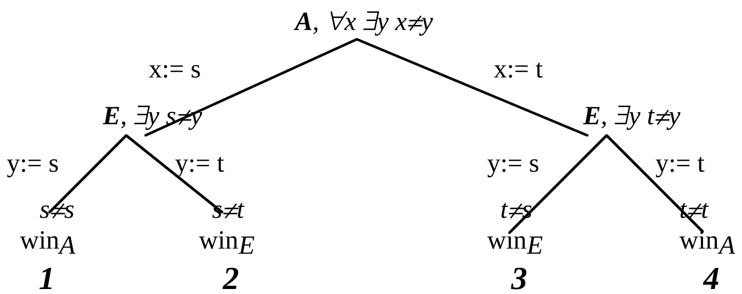
Winning sets in this game are not open for either player.

Fact The interval selection game is not determined.

Proof If player ***A*** had a winning strategy, ***E*** could use that with a one-step delay to *copy* ***A***'s responses to her own moves disguised as her moves. Both resulting sets of intervals (disjoint up to some finite initial segment!) would then have their unions in ***U****: which cannot be, as ***U**** was free. Likewise, ***E*** has no winning strategy. ■

There is a flourishing literature on determined games in descriptive set theory (cf. Löwe 2002B), but we will concentrate on more general game issues here.

5.2 Powers and representation There is more to players' powers, even in logic games, then just abilities to win. Consider the game in Section 2.4 for $\forall x\, \exists y\, x{\neq}y$ in a first-order model with two distinct objects *s*, *t* – where we now number outcomes:



That $\forall x\, \exists y\, Rxy$ is true is reflected in player ***E's*** having the obvious winning strategy "choose the object different from that chosen by ***A***" . But players have more strategies in this game, and calculating as in Section 4.1 we get their true powers:

A can force the sets {1, 2}, {3, 4}
E can force the sets {1, 3}, {1, 4}, {2, 3}, {2, 4}

Power families like these satisfy the following general properties, which are reminiscent of the definition of an ultrafilter, but now spread out over two players:

Monotonicity If *Y* is a power of ***i*** and $Y \subseteq Z$, then *Z* is also a power of ***i***
Consistency If *Y* is a power of ***A*** and *Z* a power of ***E***, then *Y*, *Z* overlap

Determinacy If *Y* is not a power of ***A*** *(**E**)*, then its set complement –*Y* in the total space of outcomes is a power of ***E*** (***A***).

These are also *all* relevant properties:

Fact An assignment of non-empty subsets of some set to two players represents their powers in some finite game iff these powers satisfy Monotonicity, Consistency, and Determinacy.

In non-determined games, the third condition drops out – after which we can do a similar representation result for monotone consistent families, in which their powers are realized in finite games of imperfect information (van Benthem 2001A; see also Section 10). A typical non-determined example would be the following specification:

powers of ***A*** *{1, 2}* powers of ***E*** *{1, 2)*

Alternatively, one can represent such families of non-determined powers using infinite games of perfect information – witness Section 8.

Digression The above conditions, and the representation result, implicitly assume the admissibility of identifying different outcome states of a game. This can happen in first-order evaluation games for states with the same variable assignment, or in chess, for the same board configurations with different histories. If we insist on *uniqueness* of outcome states, then additional conditions hold, reflecting a closer tie between strategies and powers. In particular, (a) the intersection of any two inclusion minimal power sets of two players is a singleton, and (b) each singleton outcome set can be obtained as such an intersection. A representation result for this case seems open (cf. van Benthem 2003). This issue will return for logic games in Section 9.

5.3 Modal forcing languages Games viewed at the level of players' powers have a natural associated modal *game logic*. Our purpose here is just to illustrate how this comes about – but cf. Parikh 1985. (For background in modal logic in general, see Blackburn, de Rijke and Venema 2001.) The language has proposition letters, Boolean operators, and the following *forcing modalities*:

$$\{G, \boldsymbol{i}\}\ \phi$$

saying that player ***i*** has a strategy for playing *G* which guarantees a set of outcomes all of which satisfy ϕ. Thus, ϕ. may express winning, but really, just any property of states. More formally, one can think of games as modal models ***M***, where states may also interpret game-external proposition letters – and set

$\boldsymbol{M}, s \models \{G, \boldsymbol{i}\}\ \phi$ iff there exists a power *X* for player ***i*** in *G* played from state *s* such that for all $x \in X : \boldsymbol{M}, x \models \phi$

To bring this in line with modal semantics in its better-known 'neighbourhood' version, one might use binary state-to-set *forcing relations* $\rho_G{}^{\boldsymbol{i}}\ s, Y$, and set

there exists a set *X* with $\rho_G{}^{\boldsymbol{i}}\ s, Y$ and $\forall x \in X\ \boldsymbol{M}, x \models \phi$

The main effect of this at the level of validities is the following.

Fact Modal logic with the forcing interpretation satisfies all principles of the minimal modal logic *K* except for distribution of *{}* over disjunctions.

In particular, $\{G, \boldsymbol{i}\}\ \phi$ is *upward monotone*:

if $\models\ \phi \rightarrow \psi$, then $\models \{G, \boldsymbol{i}\}\ \phi \rightarrow \{G, \boldsymbol{i}\}\ \psi$

But distribution over disjunctions is *not valid*:

$$\{G, \boldsymbol{i}\}\ \phi \vee \psi\ \rightarrow \{G, \boldsymbol{i}\}\phi\ \vee \{G, \boldsymbol{i}\}\psi$$

This is precisely the point of forcing. Other players may have powers that keep us from determining results precisely. I may have a winning strategy, but it may still be up to *you* exactly *where* my victory is going to take place. For instance, in the game of Section 5.2, ***E*** can force *{2, 3}*, but neither *{2}* nor *{3}*. Next, there are also two axioms relating powers of different players: these are the modal transcriptions the Consistency and Determinacy conditions of Section 5.2.

Finally, like for every well-behaved modal language, there is a matching notion of *bisimulation* between game models ***M***, which leaves truth of all modal forcing formulas invariant. We display it for later reference:

Definition A *power bisimulation* between two game models ***M***, ***N*** is a binary relation *E* between game states satisfying the following two conditions, for all ***i***:

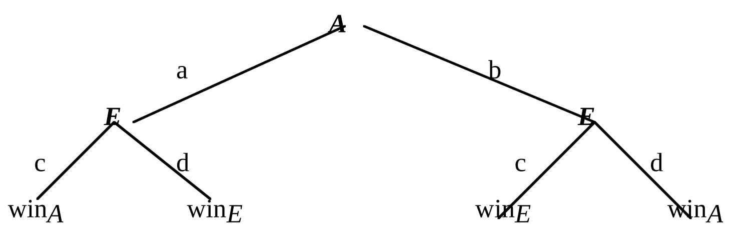
(1) if *xEy*, then *x*, *y* satisfy the same proposition letters

(2) if *xEy* and $\rho_{M}^{i}\, x,U$, then there exists a set *V* with $\rho_{N}^{i}\, y,V$ and $\forall v\in V\ \exists u\in U: uEv$; and vice versa.

The theory of this game logic is much like that of standard modal logic. Cf. van Benthem 2001D, Pauly 2001 for further details and results.

6 Extensive games and modal action logics

6.1 Extensive games as modal process models Moving beyond players' global powers, extensive games have a finer level of game states, turns, and individual moves. Such games are like ordinary models for a standard modal language with moves as atomic actions, and some special predicates, like ***end*** for being an endpoint, or $\boldsymbol{turn_i}$ for turns of player ***i***. Consider again a simple game as in Section 5.2:



The assertion that ***E*** has a winning strategy can be expressed in detail here by means of the following modal formula true in the root, which records intermediate moves:

$$[a]\langle d\rangle win_E \wedge [b]\langle c\rangle win_E$$

Alternatively, these models are the labelled transition systems of computer science, with a state space for computations by several interactive processors. A systematic study of extensive games as process models for modal languages is made in van Benthem 2002. Here we just remark that the usual process of 'solving' a game by means of a Zermelo-style colouring algorithm (cf. the discussion of determinacy in Section 5.1) really amounts to stage-wise computing a smallest (or greatest) fixed-point definition for ***E***'s winning positions. E.g., to define the above forcing modality $\{G, \mathbf{E}\}\phi$ in more detail, one can use the following modal recursion:

$$\{\mathbf{E}\}\phi \leftrightarrow ((\mathbf{end} \,\&\, \phi) \vee (\mathbf{turn}_E \,\&\, \bigvee_a \langle a\rangle\{\mathbf{E}\}\phi) \vee (\mathbf{turn}_A \,\&\, \&_a\, [a]\{\mathbf{E}\}\phi))$$

In terms of the *modal μ–calculus*, this says (using greatest fixed-points) that

$$\{\mathbf{E}\}\phi \leftrightarrow \nu p\bullet ((\mathbf{end} \,\&\, \phi) \vee (\mathbf{turn}_E \,\&\, \bigvee_a \langle a\rangle p) \vee (\mathbf{turn}_A \,\&\, \&_a\, [a]p))$$

This means that all games can be solved by model checking some assertion in an appropriate game logic over them. Thus, one particular logic game from Section 2, viz. model checking, may help us understand games in general!

6.2 Dynamic logic as strategy calculus Modal languages also have another use germane to games: they can define players' *strategies* explicitly. For, in extensive game trees, strategies for player ***i*** are nothing but *binary relations* which are functions on the turns for ***i***, while including all possible moves at other players' turns. Thus, they are of the same type as the above actions *a, b*. More generally, think of a dynamic logic with action expressions that can be formed from atomic ones using

> choice $\cup$, sequential composition ***;***, relational converse$^{\cup}$, finite iteration *, and the usual test program (ϕ)***?*** for propositions ϕ.

For instance, then, the winning strategy for ***E*** in the above game may be defined as follows (with *T* for an assertion which is always true):

$$(\langle a^{\cup}\rangle T)?\,;\, d) \cup (\langle b^{\cup}\rangle T)?\,;\, c)$$

This means that propositional dynamic logic (Harel, Kozen & Tiuryn 2000) can be used as a general *calculus of strategies*, of the sort mentioned in Section 3.1. Thus, one particular modal game logic can help us understand logic games in general.

7 Game constructions

Games do not occur by themselves, they live in families. A natural general theme are natural game-forming constructions. Logic games provide many instances of this.

7.1 Logical game operations For a start, the evaluation games of Section 2.4 provide the following game-theoretical take on the basic logical operations:

(a) *conjunction and disjunction are choices* for players :
$G \cap H$ is ***A***'s choice, $G \cup H$ is ***E***'s choice of G or H

(b) *negation is a role switch*, leading to the dual game G^d
with all the turns and win markings reversed in G.

But clearly, choice and dual are completely general operations forming new games out of old. Here is another such operation which operates inside evaluation games. Consider the rule for an existentially quantified formula $\exists x \psi(x)$:

E must pick an object d in ***M***, and play then continues with $\psi(d)$

Properly understood, the existential quantifier $\exists x$ does not serve as a game operation here: it clearly denotes an atomic game of 'object picking' by Verifier. The general game operation in this clause hides behind the phrase "continues", which signals

(c) *sequential composition of games G ; H*

These are just a few of the natural operations that form new games out of old. To arrive at further ones, consider the intuitive idea of 'conjunction' of games. So far we have two candidates, both unsatisfactory. Boolean conjunction $\cap$ forces a choice right at the start, with the game not chosen never accessed. Sequential composition *;* may lead to play of both games, if the first is ever completed. But now consider the plight of academics attracted by two games: “family” and “career”. Some make a Boolean choice for just creative celibacy or life-long bourgeoisie. Others fall for the lure of sequential composition, playing “career” first, hoping “family” will come later – which rarely happens. But most of us try to cope through a new operation:

(d) *parallel composition of games* $G \,||\, H$

This means playing a stretch in one game, then switching to the other, and so on – running around and trying to do the best in both. Logic games also provide examples of such interleaving (see Section 9.6). But even this is just one plausible parallel game construction: we might also proceed simultaneously in both games, and so on. Indeed, no complete operational repertoire is known either in logic or game theory.

7.2 Game algebra of sequential operations Game operations suggest game algebra: a calculus of equivalent game expressions. E.g., intuitively, the above choices for the two players are related by a De Morgan duality under role switch:

$$G \cap H \;=\; (G^d \cup H^d)^d$$

Or, typically, like composition of binary relations, game composition is associative. Another intuitive validity is left-distribution for composition over choice

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

By contrast, the right-distribution law

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

is not valid: ***E***'s choice on the left-hand, but not on the right-hand, side might crucially depend on the outcome of first playing game *G*. Another intuitively valid principle concerns role switch:

$$(G;H)^d \;=\; G^d \,;\, H^d$$

These intuitions may be made precise using the notions of Section 5 (for details, cf. van Benthem 2001D). The notion of game equivalence that fits best with first-order validity looks at players' powers for determining outcomes. Now, one can compute powers inductively for complex games using choice, dual, and composition. We define complex forcing relations

$\rho^{\boldsymbol{i}}{}_{G}\, x, Y$ player ***i*** has a strategy in game *G* which makes sure that *G* ends in a state in *Y* when started from state *x*

Fact The following equivalences hold:

$\rho^{\boldsymbol{E}}{}_{G\cup G'}\, x, Y$ iff $\rho^{\boldsymbol{E}}{}_{G}\, x, Y$ or $\rho^{\boldsymbol{E}}{}_{G'}\, x, Y$

$\rho^{\boldsymbol{A}}{}_{G\cup G'}\, x, Y$ iff $\exists Z, Z': \rho^{\boldsymbol{A}}{}_{G}\, x, Z$ and $\rho^{\boldsymbol{A}}{}_{G'}\, x, Z'$ and $Y = Z \cup Z'$

$\rho^{\boldsymbol{E}}{}_{G^{\boldsymbol{d}}}\, x, Y$ iff $\rho^{\boldsymbol{A}}{}_{G}\, x, Y$

$\rho^{\boldsymbol{A}}{}_{G^{\boldsymbol{d}}}\, x, Y$ iff $\rho^{\boldsymbol{E}}{}_{G}\, x, Y$

$\rho^{\boldsymbol{E}}{}_{G;G'}\, x, Y$ iff $\exists Z: \rho^{\boldsymbol{E}}{}_{G}\, x, Z$ and $\forall z \in Z\ \rho^{\boldsymbol{E}}{}_{G'}\, z, Y$

$\rho^{\boldsymbol{A}}{}_{G;G'}\, x, Y$ iff $\exists Z: \rho^{\boldsymbol{A}}{}_{G}\, x, Z$ and $\forall z \in Z\ \rho^{\boldsymbol{A}}{}_{G'}\, z, Y$

Using superset closure of powers, the second clause simplifies to

$\rho^{\boldsymbol{A}}{}_{G\cup G'}\, x, Y$ iff $\rho^{\boldsymbol{A}}{}_{G}\, x, Y$ and $\rho^{\boldsymbol{A}}{}_{G'}\, x, Y$

Remark If we also assume *determinacy*, in the earlier sense that

for each set *Y*, either ***E*** can force *Y*, or ***A*** can force *W–Y*

then we just need to define forcing relations for player ***E***, because

$\rho^{\boldsymbol{E}}{}_{G^{\boldsymbol{d}}}\, x, Y$ iff not $\rho^{\boldsymbol{E}}{}_{G}\, x, W-Y$

All powers for player ***A*** then follow by observing that

$\rho^{\boldsymbol{A}}{}_{G^{\boldsymbol{d}}}\, x, Y$ iff not $\rho^{\boldsymbol{E}}{}_{G}\, x, W-Y$

Now, take an algebraic language of game expressions starting with variables, and operations $\cup$, ^{d}, **;**. In addition take ι for the *idle game*, staying at the same state.

Definition Two game expressions *G*, *H* are *equivalent* (written as $G = H$) if they have the same power relations for their players in all game models. We also write $G \leq H$ in case of a universally valid *inclusion* between the respective powers.

7.3 Excursion: a complete system Game validity (first proposed in ' Taiwan old') validates the preceding observations about valid and non-valid equations, and many more. There is a simple complete system for this algebra of the game-theoretic analogues of the usual first-order operations. We display it here, merely to show a surprising fact vindicating our approach. Just underneath standard first-order logic, there lies a systematic game logic! For a completeness proof, cf. Goranko 2000. *Basic Game Algebra* consists of the following principles:

(1)	the laws of *De Morgan algebra* for choice and dual			
(2)	$G\,;(H\,;K)$	=	$(G\,;H)\,;K$	*associativity*
	$(G \cup H)\,;K$	=	$(G\,;K) \cup (H\,;K)$	*left-distribution*
	$(G\,;H)^d$	=	$G^d\,;H^d$	*dualization*
(3)	$G \leq H \ \rightarrow\ K\,;G \leq K\,;H$			*right-monotonicity*
(4)	$G\,;\iota \ = \ G \ = \ \iota\,;G$			

Moreover, De Morgan algebra consists of the following axioms:

$$x \cup x = x \qquad x \cap x = x \qquad x \cup y = y \cup x \qquad x \cap y = y \cap x$$

$$x \cup (y \cup z) = (x \cup y) \cup z \qquad x \cap (y \cap z) = (x \cap y) \cap z$$

$$x \cup (y \cap z) = (x \cup y) \cap (x \cup z) \qquad x \cap (y \cup z) = (x \cap y) \cup (x \cap z)$$

$$(x \cup y)^d = x^d \cap y^d \qquad (x \cap y)^d = x^d \cup y^d \qquad x^{dd} = x$$

For details about this system, cf. the above two references. For our later discussion in Section 9, it is worth noting that basic game algebra is *decidable*.

7.4 Dynamic game logic Basic game algebra can be embedded into a richer system, viz. the decidable 'dynamic game logic' of Parikh 1985 which enriches the earlier modal forcing language with modalities $\{G, \boldsymbol{i}\}\phi$ with complex game terms G. Typical for these systems is the interplay of two ingredients in one language:

dynamic component	expressions G for games
static component	propositions ϕ about states of play

This is like dynamic logics in computer science which manipulate program expressions and propositions about computational states together. For more on dynamic game logic, cf. Pauly 2001, van Benthem 1999–2002.

7.5 Logics of parallel game operations Parallel game constructions have been studied extensively in game semantics for *linear logic* whose tensor product involves interleaved games where Adam can switch between games. Under this interpretation, linear logic is a complete axiomatization of several central sequential and parallel game constructions. Cf. Blass 1992, Abramsky 1996, Girard 1997 for details.

8 Finite versus infinite games

8.1 The importance of infinite runs The intuitive emphasis so far has been on finite games, and their outcome states. But several logic games in Section 2 support *infinite runs*, witness model construction, model comparison, and even model checking for first-order languages with fixed-point operators. The same move was behind the shift from Zermelo's Theorem to the Gale-Stewart Theorem, where players produce infinite runs, marked as winning or losing. And game theory also has more than just finite matrices or tree pictures. Infinite games model situations with ongoing behaviour, such as iterated Prisoner's Dilemma in studying the possible emergence of social cooperation. Such ongoing behaviour is just as important as finite termination. A good analogy comes from computer science, using our process analogy for games. *Terminating programs* are meant to find some value, or finish some task within a finite time. But there are also crucial non-terminating programs like *operating systems* which ensure the proper functioning of some device: the longer the better. Or take linguistics. Language games for specific conversational tasks should terminate. But there is also the Great Game of Language: with discourse as the 'operating system of cognition'. This should keep functioning forever – and 'losing' would mean a break-down in that history, or failure to allow a proper say to all participants. Both kinds of game are needed then to model realistic phenomena.

8.2 Extending the game logic perspective Infinite games can still be studied by the logical techniques of Sections 3–7. Outcomes are now the runs themselves. But, both outcome and action levels still make sense. Here are two illustrations.

Example Computing powers

Recall the interval selection game of Section 5.1, used in the standard argument that non-determined games exist. Much more interesting information can be extracted from it by looking into players' powers. We can then prove the general fact that

Fact Identifying infinite runs that are equal up to finite initial segments, both players have *the same powers* in the interval selection game.

Thus, this perfect information game is about the simplest infinite realization of the following non-determined power specification from Section 5.2

powers of ***A*** *{1, 2}*
powers of ***E*** *{1, 2)*

For a finite game realization with imperfect information, see Section 10.

Example Temporal logic of games

Infinite games suggest the use of somewhat more expressive game logics. These arise, e.g., when thinking about minimal expressive power needed to formalize well-known game-theoretic arguments. For instance, the standard proof of the Gale-Stewart Theorem (Section 5.1) involves the following result true for all games:

Weak Determinacy

Either player ***E*** has a winning strategy, or player ***A*** has a strategy which forces infinite branches on which player ***E*** never has a winning strategy.

The usual proof then continues as follows. Given that ***E***'s winning set is open, this auxiliary principle implies that the branch forced by player ***A*** is a loss for ***E*** – so ***A*** has a winning strategy. The straightforward formalization of Weak Determinacy is in a *branching temporal logic*, evaluating formulas at ordered pairs <*h*, *t*> of a current branch *h* and current point *t* on it:

$$\boldsymbol{M}, h, t \ \models \ \{G, \boldsymbol{E}\}\, \phi \ \vee \{G, \boldsymbol{A}\}\, \mathcal{A}\, \neg\{G, \boldsymbol{E}\}\, \phi$$

Here, $\{G, \boldsymbol{i}\}\phi$ is a *modal-temporal forcing modality* extending that of Section 5.3:

There exists a strategy for player ***i*** ensuring that only runs result,
having the current history *h* up to point *t* as an initial segment,
which satisfy the temporal logic formula ϕ.

Moreover, the temporal logic comes in explicitly through the standard operator $\mathcal{A}$ ('always') in the right-hand disjunct. It says that a statement is *always true on the current branch*. The temporal logic format for describing infinite games seems very powerful. Most reasonable winning conventions can be given in some such format. Examples are the earlier-mentioned *safety* and *liveness* properties, as well as the winning conventions of infinite logic games. Also, like dynamic game logic (Section 7.4), temporal game logic involves a merge of internal and external game languages.

Infinite games are complex structures. In particular, they have huge unwieldy spaces of strategies – as high-lighted in the 'folk theorems' of game theory on a plethora of equilibria. Another general logical issue then is *finitization.* (cf. Section 9). To which extent can we know infinite strategies in a full infinite game through their finite approximations? The savvy reader will suspect that this is a lost hope in general, but that things look brighter if we get some royal help, e.g., from König's Lemma.

9 From game logics to logic games

Having discussed general game logic for a while, let us now return to logic games, and see what light our general perspective sheds up our familiar activities.

9.1 Questioning game equivalence For a start, to someone sensitized to the above general topics, a useful exercise is rereading familiar arguments about logic games, and ask "do I understand the assertions that are supposed to be 'obvious'?" As a concrete illustration, take the issue of game equivalence. The literature is full of unsupported statements to the effect that one logic game is 'equivalent' to another. E.g., Hodges 1999 tells us all games are equivalent to full infinite game trees, as we can disregard all undesired runs by calling them losses for ***E***. But this presupposes a coarse notion of game equivalence, and one biased in favour of just one player. Also, authors 'reformulate' logic games without stating precisely in what sense the results are equivalent. E.g., Barwise & van Benthem 1999 define infinite model comparison games inverting the schedule of Section 2.6. The game starts with one finite partial isomorphism between two models. Each round lets Duplicator ***D*** choose some family *F* of partial isomorphisms, followed by a selection by Spoiler ***S*** of one *f* in *F*. In the next round, ***D*** must select a set F^+ again, ***S*** then chooses a partial isomorphism in F^+ again, and so on. The back-and-forth property to be maintained by ***D*** is:

> For every object *a* in one model, there exists an object *b* in the other
> such that $f \cup \{(a, b)\} \in F^+$ – and likewise in the other direction.

This is equivalent to standard comparison games at the level of players' *powers* – and the trick is a bit like the distributive law of Section 4.1, inverting scopes of logical operators. In each round, ***D*** offers ***S*** a panorama of all choices he could make, plus her own responses to them. ***S*** then selects his own move plus ***D***'s pre-packaged response – thereby setting the new stage. In human terms, ***D*** behaves like a colleague of mine, who tries to speed up department meetings by saying: "Now you're going to say *A*, and I will say *B* – or, you're going to say *C*, and then I will say *D* – etcetera."

More generally, in terms of Section 5 versus Section 6, logic games seem biased toward outcomes and powers, rather than the fine-structure of actions and turns. Most intuitive equivalences between such games can be justified in this manner.

9.2 The outcome perspective in logic games The above points of general game logic can be made more precise in specific cases. Consider our running example of evaluation games for first-order logic. We give three concrete illustrations.

Richer denotations Any game ***game(M***, *s, ϕ)* assigns a much more structured denotation to a formula ϕ than just a truth value, viz. the complete power structure of the two players. We can think of these as their forcing relations $\rho^{\boldsymbol{i}}{}_{G}\, x, Y$ computed over the model ***M*** in the sense of Section 7.2. This suggest that there are several natural levels to assigning meaning, even for standard logical languages.

Power bisimulation At the level of powers, the appropriate notion of equivalence between models ***M***, ***N*** is the power bisimulation of Section 5.3. It is easy to translate this back into standard first-order terms – and then, we get a variant of standard *potential isomorphism* (cf. Section 2.6). This time, the states to be related are not tuples of objects from the two models, but variable assignments over ***M***, ***N***. The bisimulation, viewed as a family of links, then has to satisfy the obvious back-and-forth conditions having to do with shifting values for variables. Well-understood, this also seems the more natural notion for first-order logic in general.

Representation of general games Perhaps most strikingly, evaluation games seem adequate for games in general! The idea is that we can reinterpret any extensive game as a game where players evaluate an associated game-logical assertion. Consider the game in section 6.1. It is obviously outcome-equivalent in to an evaluation game for the associated modal formula $[a]<d>win_{\boldsymbol{E}} \wedge [b]<c>win_{\boldsymbol{E}}$ presented there. Modulo outcome equivalence, we can rearrange any finite extensive to one with a uniform alternating schedule, while making all runs of equal length. This allows us to write up an associated modal formula in iterated $[]<>$ form, whose evaluation game proceeds like the original game. Informal observations like this can be made more precise mathematically. Here is a technical representation result showing how logic games may be complete for game logics (van Benthem 2001D):

Theorem The basic algebra of sequential operations on arbitrary games coincides with the game algebra of first-order or modal evaluation games.

Thus, in particular, any non-valid principle of basic game algebra (cf. Section 7.3) must already have a counter-example in first-order evaluation games. More delicate issues arise when we demand *uniqueness* of outcomes (cf. Section 5.2) – as this can be done in first-order evaluation games, but not always in modal ones (cf. van Benthem 2003). Our general conclusion is that issues of game logic make sense for logic games, but also, that the general game-theoretic import of logic games – specialized though they seem – may be much larger than appears at first sight.

9.3 The action perspective in logic games Despite the noted power bias, it also makes sense to look at the fine-structure of logic games at their action level. Evidently, in this setting, far fewer games will be identified.

Finer levels of denotation Consider again first-order evaluation games. We now get much finer notions of denotation, leading to a subset of game equivalences, viz. those which leave the move and turn structure intact as far as ordinary modal logic cares about them. Some of the laws of basic game algebra in Section 7.3 survive this: examples are the De Morgan laws, or left-distribution of composition over choice. But other principles will fail, as they merely preserved powers – such as the propositional distribution of Section 4.1 reversing players' turns.

Open question Determine the complete basic game algebra at this modal level.

This analysis suggests that there are many natural levels of equivalence for first-order formulas. And this again ties up neatly with a persistent philosophical tradition of looking for various levels of identifying 'propositions' (cf. Lewis 1972).

Strategy calculus in dynamic logic Another point where individual actions come in are players' strategies in logic games (cf. Section 3). As we have noted, strategies across logic games unify such diverse logical notions as formulas, 'reasons', proofs, models, or semantic analogies. Underlying all of these is the dynamic logic of Section 6.2. This gives a fresh look at known notions. E.g., in first-order evaluation games, the item closest to strategies are *Skolem functions*. Dynamic logic suggests a calculus of definable Skolem functions, taken as relations: which seems a natural generalization. Its major operations would be choice, composition, and iteration of binary relations, allowing, amongst others, the standard sequential program constructions *IF P THEN* π *ELSE* π' and *WHILE P DO* π (cf. van Benthem 2002).

Incidentally, such an explicit format for analyzing strategies also makes sense after all in the dynamic game logic of Section 7.4. That calculus was $\exists$–sick in the sense of Section 4.2, because one just says that a strategy exists without naming it. But we can remedy this using enriched modal forcing assertions including a witness:

$$\{G, \boldsymbol{i}, \sigma\}\phi$$

stating that, in game G, strategy σ for player $\boldsymbol{i}$ achieves a set of outcomes satisfying proposition ϕ. Dynamic game logic in this guise still needs to be developed.

9.4 Excursion: strategy calculus in type-theoretic format Formats other than dynamic logic may be attractive, too. Type theories (cf. Barendregt 1992) manipulate statements of the form $\sigma : G$ interpreted as 'σ is a proof of assertion G', or 'σ is an object having property G'. Strategy calculi might manipulate statements

$\sigma : G$ $\qquad$ σ is a winning strategy for player $\boldsymbol{E}$ in game G

Such interpretations are given for linear logic games (cf. Section 7.5) in Abramsky & Jagadeesan 1994. Here is a simple example. Consider this sequent derivation for a propositional validity, whose steps are well-known valid inference rules:

$$A \Rightarrow A \qquad B \Rightarrow B$$
$$A, B \Rightarrow A \qquad A, B \Rightarrow B \qquad C \Rightarrow C$$
$$A, B \Rightarrow A \wedge B \qquad A, C \Rightarrow C$$
$$A, B \Rightarrow (A \wedge B) \vee C \qquad A, C \Rightarrow (A \wedge B) \vee C$$
$$A, B \vee C \Rightarrow (A \wedge B) \vee C$$
$$A \wedge (B \vee C) \Rightarrow (A \wedge B) \vee C$$

We want to make the strategy calculus behind this derivation explicit. In particular, this requires the identification of strategy combinations supporting the proof steps. Here is a concrete format of analysis, written with some ad-hoc notation:

$$x{:}A \Rightarrow x{:}A \qquad y{:}B \Rightarrow y{:}B$$
$$x{:}A,\ y{:}B \Rightarrow x{:}A \qquad x{:}A,\ y{:}B \Rightarrow y{:}B \qquad z{:}X \Rightarrow z{:}C$$
$$x{:}A,\ y{:}B \Rightarrow (x, y){:}A \cap B \qquad x{:}A, z{:}C \Rightarrow z{:}C$$
$$x{:}A,\ y{:}B \Rightarrow \langle L, (x, y)\rangle{:}(A \cap B) \cup C \qquad x{:}A, z{:}C \Rightarrow \langle R, z\rangle{:}(A \cap B) \cup C$$
$$x{:}A,\ u{:}(B \cup C) \Rightarrow \textbf{IF}\ head(u)=L\ \ \textbf{THEN}\ \langle x, tail(u)\rangle\ \textbf{ELSE}\ tail(u){:}\ (A \cap B) \cup C$$
$$v{:}\ A \cap (B \cup C) \Rightarrow \textbf{IF}\ head((v)_2)=L\ \ \textbf{THEN}\ \langle (v)_1, tail((v)_2)\rangle\ \textbf{ELSE}\ tail((v)_2){:}\ (A \cap B) \cup C$$

This derivation shows how to compose strategies in more complex games from strategies in subgames. The operations performing this task are completely general, not depending on the original idea of constructing a logical proof:

storing strategies for a player who is not to move	$< , >$
using a strategy from a list	$( )_i$
computing the first recommendation of a strategy	*head()*
as well as the remaining strategy	*tail()*
making a choice dependent on some information	*IF THEN ELSE*

As strategies encode very different logical objects: proofs, models, analogies, etc., the above derivation can stand for quite different things. It is recipe for constructing proofs, and the operations then encode what goes on as logical operations get added. But it also describes how any winning strategy for Verifier in an evaluation game for $A \wedge (B \vee C)$ can be turned into a winning strategy in $(A \wedge B) \vee C$. Not all logic games support operations of choice, though – and hence the above recipe may not make much sense (yet) for games of model construction, or model comparison.

9.5 Operations on logic games Logic games are not isolated activities, they can be systematically modified and combined (cf. Section 2.7). In particular, they do not just support sequential game operations, but also various parallel ones which have not been analyzed systematically to date. (A nice example is the interleaving of fixed-point games in the Stage Comparison Theorem of Moschovakis 1974.) There is probably a stable operational game structure behind current logic games, which should be an interesting testing ground for general game calculi. In this paper, we only offer an illustration showing the interest of such matters (van Benthem 1999).

Relating evaluation and comparison games In this excursion, our aim is to prove the informal equation '$E = H^2$' relating Hintikka games with Ehrenfeucht games. Logic games often involve the same idea – such as 'back and forth'. For instance, object-picking moves in evaluation games and in comparison games seem similar. This can be made precise in terms of an operation of interleaving games, and we can even correlate strategies in the two games directly. The Adequacy Theorem for finite-depth Ehrenfeucht-Fraïssé games of Section 2.6, $\exists$–cured in Section 4.2, suggests an explicit link between strategies across comparison and evaluation games for models $\boldsymbol{M}$, $\boldsymbol{N}$. First-order ϕ of quantifier depth k between $\boldsymbol{M}$, $\boldsymbol{N}$ drove winning strategies for Spoiler in the k-round comparison game between **M**, **N**. But we can do away with this intermediary! Let $\boldsymbol{M} \models \phi$, $\boldsymbol{N} \models \neg\phi$. This induces a winning strategy for Verifier in an evaluation game ***game(φ, M)*** *plus* one for Falsifier in ***game(φ, N)***.

Theorem There exists an effective correspondence between

(a) winning strategies for Spoiler in the k-round comparison game

(b) pairs of winning strategies for Verifier and Falsifier in some k-round evaluation game, played in opposite models.

Proof Without loss of generality, formulas can be assumed to be constructed from atoms with negations, disjunctions, and existential quantifiers only. *From (b) to (a).*

Let an *H-pair of depth k* consist of a formula ϕ of quantifier depth k plus a winning strategy σ for ***V*** in the ϕ-game in one of the models, and a winning strategy τ for ***F*** in the ϕ-game in the other model. We sketch how to *merge* σ, τ. Spoiler looks at the two evaluation games. Suppose ***V*** wins ϕ in ***M***, and ***F*** wins ϕ in ***N***. If ϕ is a negation $\neg\psi$, Spoiler switches to the obvious strategies for ***F*** and ***V*** w.r.t. ψ. (Note that this is internal computation: the opponent in the comparison game does not see any action yet.) If ϕ is a disjunction $\psi\vee\xi$, Spoiler uses his ***V***-strategy in the one model to choose a disjunct. His ***F***-strategy in the other model will also win against that disjunct. Proceeding in this way, the formula is broken down until an existential subformula $\exists x\psi$ is reached. Spoiler then uses his ***V***-strategy σ in the model where it lives, say ***M***, to pick a witness *d*. This model ***M*** and object *d* are his opening move in the first round of the Ehrenfeucht game. Next, what remains for Spoiler is still a winning strategy σ^- for ψ in ***M*** after this first move. Now, let Duplicator respond with any object *e* in the other model ***N***. This choice can also be seen as a move by Verifier in the evaluation game for $\exists x\psi$ in ***N***. Now we know that Falsifier still has a winning strategy τ^- for ψ in ***N*** after this first move. So, by induction, we still have an *H*-pair of depth *k-1*, which can be merged into a follow-up winning strategy for Spoiler in the *(k-1)*-round comparison game between ***M*** and ***N***. The total effect is a *k*-round ***S***-strategy. This argument yields an algorithm for Spoiler's computation.

From (a) to (b). This direction seems harder, as we have to 'decompose' one object: Spoiler's winning strategy, into two separate ones that must form a suitable *H*-pair. One proof of this follows our earlier construction of a difference formula of depth *k* from an ***S***-strategy (Section 4.2). This formula induces two evaluation strategies effectively. Let us describe 'splitting' of a comparison strategy directly. Consider any winning strategy for ***S*** in the *k*-round comparison game between two models ***M***, ***N***. In the first move, ***S*** chooses, say, model ***M*** and object *d*. Our desired formula will then start with an existential quantifier, and ***V*** has the winning strategy in ***M***. Let Duplicator now make any response *e* in ***N***. We know that Spoiler still has a *(k-1)*-round winning strategy in the two expanded models *(**M**, d)* , *(**N**, e)*. Inductively, we can find *H*-pairs of depth *k-1* for each choice *e* that Duplicator makes. Moreover, by the earlier Finiteness Lemma, only finitely many logically non-equivalent formulas can be involved in these pairs. Then, one over-all existential quantification over a suitable conjunction of formulas of depth *k-1* defines our desired *H*-pair of depth *k*. In particular, if it is ***V*** who has the winning strategy of a relevant *H*-pair ϕ in the model ***M***, put itself in the conjunction; otherwise, put its negation. ■

In modelling some logical activity as a game, there may be different natural levels. One can give finer or coarser modular structure, treating subgames as single actions, or bunches of separate moves as instances of one indeterministic action for a player. Operations are just the beginning of a logic of *game architecture*.

9.6 Finite versus infinite Finally, the issue of finite versus infinite also comes up in logic games. And as in Section 8, there is an issue of *finitization*. Consider model construction games, which may go on forever, so their game tree is infinite. Critic's winning strategies made Builder lose on each run, at some finite stage. These strategies were associated with *finite objects*, viz. proofs. How can this be? The reason is that the model construction game tree is *finitely branching*, so by *König's Lemma*, there must be some finite level at which Critic has already managed to block every construction attempt. This closed game tree is a finite object.

Such a reduction does not always hold. E.g., in infinite comparison games, Spoiler may be able to win against Duplicator, blocking each run at some finite stage, without their being an obvious finite object encoding this. In fact, there must be a formula of *infinitary* first-order logic witnessing the relevant difference, but there need not be a standard first-order one. Finitization, if available, can be very useful. E.g., Hirsch and Hodkinson 2002 show the following for their representation game (Section 2.7), using the finite branching for Builder's (though not Critic's!) moves:

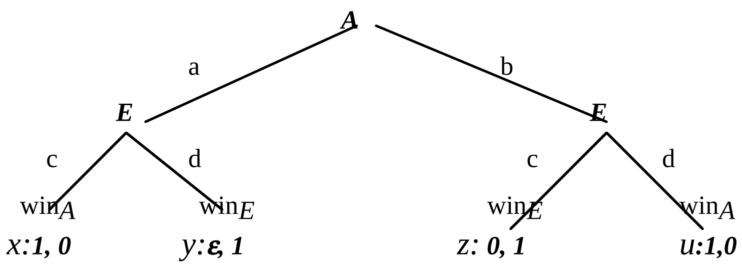
> Builder can win the infinite game if she can win all all finite cut-offs,

and her winning strategy is easy to piece together from these. Then representability becomes equivalent to a set of first-order assertions expressing Builder's being able to win all finite cut-offs, which leads to a perspicuous axiomatization. Incidentally, in this game, by the same reasoning, Critic's winning strategy, if available, must be a *finite object* again. It is a sort of proof that the given algebra is unrepresentable.

10 From game theory to logic games

Logic games are rather special activities, in that players' behaviour is much more constrained than that in ordinary game theory. In real games, players have refined *preferences* between outcomes beyond winning or losing, they also have to operate under *uncertainties* about what really happens during a move (think of card games), and moreover, there can be more players than two, leading to *coalitions*! Importing these concerns into logic games makes them much more realistic, even though – with a few exceptions – there is hardly any theory of the resulting activities. We will just skim a few of the resulting issues, as there are natural motivations for enriching logic games, even from the point of view of reasoning and other core business.

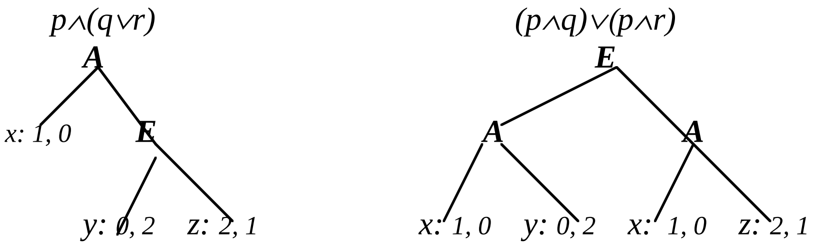
10.1 Preferences Preferences in logic games allows for finer descriptions of behaviour. Consider the game of Section 6.1, where ***A*** now has a slight preference for one site of defeat over the other (we write values for ***A***, ***E*** in that order):



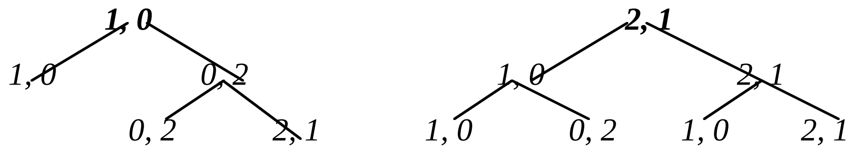
What will happen now? ***A*** can assume that, whatever he does, ***E*** will go for her most preferred outcome *y* or *z*. So, as he himself prefers y to z, he will choose "left", forcing ***E*** to end up in *y*. Thus the game ends in the run 'left-right' with outcome *y*. Indeed, this pair of strategies σ, τ is in *Nash Equilibrium*: neither player can gain by deviating from his strategy, assuming the other strategy in the pair not change.

This story assumes a certain notion of 'rationality', which drives the familiar game solution algorithm of Backward Induction, computing values via a Maximin rule.

Here is how this algorithm would work on the two evaluation games of Section 4.1, when we assume the preference structure indicated below:



We display all pairs (***A***–*value*, ***E***–*value)* computed bottom-up:



These trees correspond to different outcomes for the joint behaviour of the players. We predict outcome *x* on the left, but *z* on the right. There are many issues of general game logic for games with preferences (cf. van Benthem 2001C), but here we just consider one, which also affects logic games.

Game algebra with preferences Call two game expressions equivalent if their Nash equilibrium solutions are the same for every concrete realization including players' utilities. The preceding example shows the basic game algebra of Section 7.3 no longer qualifies, as propositional distribution fails. Vice versa, invalid equivalences may hold for special preference values. E.g., assuming rational behaviour, games *A* and $A \cup B$ are preference-equivalent for each evaluation where ***E*** prefers *A* to *B*. Of special interest are antagonistic zero-sum games, where ***A*** evaluates all outcomes in the opposite way from ***E***, as in logic games of winning and losing. Then some standard logic remains. Assume that different outcomes correspond to different preferences – otherwise, we might just as well identify them for game-theoretic purposes. Then it is easy to check the following

Fact With zero-sum preferences Boolean Absorption $A = A \cap (A \cup B)$ is valid, whereas with general preferences, it is not.

Open question

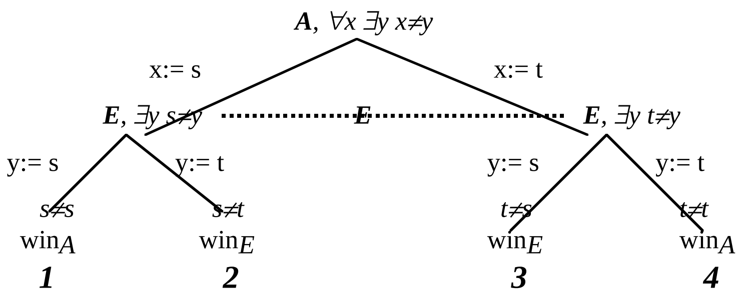
Find the complete basic game algebra under Nash equilibrium equivalence in case of (a) arbitrary preferences, (2) zero-sum preferences.

Logic games with preferences Values and preferences also make of sense in logic itself, as in many-valued logics, or preference models for logics of belief and conditionals. Another concrete source refers to game dynamics. Let us introduce resource structure into logic games, and say that, other things being equal, *players prefer outcome nodes lying on a shorter path from the root*. This assumes that players want to get to a winning node as soon as possible, or if no such node exists, to a losing node with the least effort. Then we can make more definite predictions about the course of games, and adapt definitions of validity and truth accordingly.

10.2 Solving games Making these connections does raise conceptual problems. In logic games, central notions are encoded in winning strategies, and in that case, the precise behaviour of the other player is unimportant: it will always be in Nash equilibrium with the winning strategy. But in game theory, a strategy models a type of behaviour vis-á-vis other players, and hence it is the *strategy profile for all players* that we are after. Nevertheless, multi-agent interactive behaviour seems crucial to logic, too. A proof is a long-term way of responding to objections, an isomorphism is a never-ending way of simulating one model in another, and sticking to the truth means emerging victorious no matter where your opponents in life try to push you. From a different perspective of resource-sensitivity, this also seems a major point of Girard 1997. For similar ideas in computer science, see Abramsky 1998. But then, we do seem to need a major change of perspective in logic. Based on preferences, game theory shows that Nash equilibria always exist for finite games if we are willing to admit *mixed strategies*, where pure actions are played with certain probabilities. What would be the logical point of such probabilistic solutions?

10.3 Imperfect information Standard logic games have perfect information. At each stage of play, players know exactly where they are in the game tree. But in many real games, players have imperfect information as to where they are in the game tree. E.g, in card games we do not know the complete distribution of the cards. Still players must move despite this partial ignorance.

Peculiarities of imperfect information games Games like this diverge from logic games in important respects. For instance, consider the evaluation game for the first-order formula $\forall x\ \exists y\ x{\neq}y$ in Section 4.2. Now assume that Verifier is ignorant of the object chosen by Falsifier in his opening move. In game-theoretic notation, the new tree looks as follows, with a *dotted line* indicating ***E***'s uncertainty:



This game is quite different from the earlier one. In particular, if we allow only *uniform strategies* that can be played without resolving the uncertainty – as seems reasonable – ***E*** has only 2 of her original 4 strategies left in this game: 'left' and 'right'. Then *determinacy is lost*: neither player has a winning strategy!

Games with imperfect information like this still support game logics at both power and action levels (cf. van Benthem 2001A). In particular, here is a formula of an *epistemic-dynamic* action logic describing player ***E***'s plight in the central nodes:

(a) $K_E\,(\langle y{:=}t\rangle win_E \vee \langle y{:=}s\rangle\ win_E)$
E knows that some move will make her win, picking either *s* or *t*

(b) $\neg K_E\ \langle y{:=}t\rangle win_E) \wedge \neg K_E\ \langle y{:=}s\rangle win_E)$
there is no particular move of which ***E*** knows that it will make her win.

This is the well–known 'de re – de dicto' distinction from philosophical logic. For instance, I may know that the ideal partner for me is walking out right there in the street, without ever finding out which one of these people was that person.

On the other hand, we can also describe games like this at the global level of powers. E.g., with uniform strategies, players' powers in the above game are as follows:

powers of ***A***	*{1, 2}, {3, 4}*
powers of ***E***	*{1, 3}, {2, 4}*

The analysis of Section 5.2 can be extended to this situation. Families of powers satisfy Monotonicity and Consistency, though not Determinacy. And conversely, the former two conditions suffice for representability of given powers for two players as those realized in some game of imperfect information.

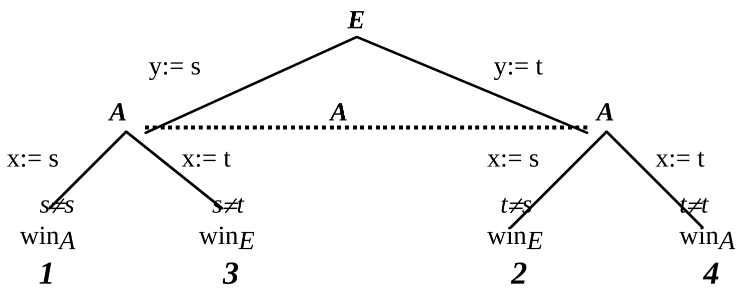
Logic games with imperfect information There is one exception to what was said above. Imperfect information has been added to logic games in the work of Hintikka & Sandu on *IF logic* (cf. Hintikka 2002 for a most recent version plus intended applications). In the slash notation of IF logic, the preceding game is written as

$$\forall x\ \exists y/\mathbf{x}\ x{\neq}y$$

expressing that the choice of the existential witness for *y* must not depend on the universal challenge for *x*. As in Section 4, a good test question for a notion of games like this is: when do two given IF formulas define the same games? For instance,

Is game $\forall x\ \exists y/\mathbf{x}\ Rxy$ 'equivalent to' $\exists y\ \forall x\ Rxy$?

The popular answer is YES, because ***E*** has the same winning powers in both. But in imperfect information games, specifying the powers of one player does not automatically tell us everything about the other player. Therefore, ***A*** may not agree that these two games are the same. And indeed, the so-called Thompson transformations of game theory (cf. Osborne & Rubinstein 1994) say that the correct equivalent is rather this game, with switched scheduling:



In terms of IF logic, $\forall x\ \exists y/\mathbf{x}\ x{\neq}y$ would really be 'equivalent to the slash formula representing the latter game, which is the much nicer quantifier exchange form

$$\exists y\ \forall x/\mathbf{y}\ x{\neq}y$$

IF games have generated a lot of literature concerning their interpretation (cf. van Benthem 2000 for an epistemic analysis). Even so, they show that introducing imperfect information into logic games is exciting and perhaps even useful. In particular, they might provide a normal form representation for validities in all imperfect information games, the way first-order evaluation games did for the general algebra of perfect information games (Section 9.2).

One might try similar moves with the other logic games mentioned in Section 2. Of course, the motivation for introducing imperfect information in such cases can have many sources. One might speculate about proofs where participants may have forgotten some things that has been said. Or, we play model comparison games with a fixed number of *pebbles*, representing some finite memory (Immerman & Kozen 1987), then players will not be able to distinguish the same assignments of objects to these pebbles, even when they occur at different stages of the game.

Uncertainty about the future Finally, even in games of perfect information, where we know our position in the game tree at any time, we have 'forward ignorance' of the future course of play. Nash equilibrium makes some predictions, but in general, we are in a partial deliberation about future actions and choices, based on beliefs about ourselves and others. There is a flourishing literature on this, involving belief revision and counterfactual reasoning (cf. Stalnaker 1999). The resulting game logics put philosophical logic on top of the mathematical logic of game structure. Such issues, too, make sense for studying logic games, but we forego them here.

10.4 More agents and coalitions Finally, many games involve more than two players, and hence the possibility of genuine coalitions: cf. Pauly 2001 for a study with standard logical tools. Many agents are the reality in conversation and debate, and they also make sense in logic games when thinking about teams of players – viewing Adam and Eve rather as some sort of Bourbaki. This suggestion is in the air today among logicians, but it does not yet seem to have had serious impact.

11 Conclusions

This paper has shown how logic games can be seen systematically as a very interesting subclass of the totality of all games, raising new issues of game logic beyond standard game theory precisely because they are somewhat better delineated. On the other hand, the study of logic games would also benefit from some systematic importation of ideas from general game theory. And finally, either way, we think all this supports the idea of viewing logic as a study of the dynamics, rather than just statics, of statement, reasoning, and communication.

12 References

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

S. Abramsky, 1998, 'From Computation to Interaction, towards a science of information', BCS/IEE Turing Lecture.

S. Abramsky & R. Jagadeesan, 1994, 'Games and Full Completeness for Multiplicative Linear Logic', *Journal of Symbolic Logic* 59:2, 543 – 574.

H. Barendregt, 1992, 'Lambda calculi with Types', in S. Abramsky, D. Gabbay & T. Maibaum, eds., *Handbook of Logic in Computer Science*, Clarendon Press, Oxford, 117-309.

J. Barwise & J. van Benthem, 1999, 'Interpolation, Preservation, and Pebble Games', *Journal of Symbolic Logic* 64:2, 881–903.

J. Barwise & J. Perry, 1983, *Situations and Attitudes*, The MIT Press, Cambridge (Mass.).

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

J. van Benthem, 1999 – 2002, *Logic in Games*, lecture notes, ILLC Amsterdam & Stanford University.

J. van Benthem, 1999, 'Games and Strategies inside Elementary Logic', invited lecture, 7th Asian Logic Conference, Hsi-Tou 1999. Manuscript, Institute for Logic, Language and Computation, University of Amsterdam.

J. van Benthem, 2000, 'Hintikka Self-Applied: on the epistemics of logic games with imperfect information'. To appear in Hintikka Volume, *Library of Living Philosophers*.

J. van Benthem, 2001A, 'Games in Dynamic Epistemic Logic', in G. Bonanno & W. van der Hoek, eds., *Bulletin of Economic Research* 53:4, 219–248.

J. van Benthem, 2001B, 'Logical Construction Games', to appear in H. Zeevat & N. Javashvili, eds., Proceedings Borjomi Autumn School 2000.

J. van Benthem, 2001C, 'Logic and game theory, close encounters of the third kind', to appear in G. Mints & R. Muskens, eds., *Proceedings 9th CSLI Workshop on Logic, Language and Computation*, CSLI Publications, Stanford.

J. van Benthem, 2001D, 'Logic Games are Complete for Game Logics', to appear in *Studia Logica.*

J. van Benthem, 2002, 'Extensive Games as Process Models', *Journal of Logic, Language and Information* 11, 289–313.

J. van Benthem, 2003, 'Powers and Representation in Games', manuscript, Institute for Logic, Language and Computation, University of Amsterdam.

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

A. Blass, 1992, 'A Game Semantics for Linear Logic', *Annals of Pure and Applied Logic* 56, 183 – 220.

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

C. Dutilh-Novaes, 2002, 'Medieval Obligationes as Argumentation Games', Philosophical Institute, University of Leiden.

J-Y Girard, 1997, 'The meaning of logical rules I, II', manuscripts, University of Sophia-Antipolis.

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

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

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

J. Hintikka, 2002, 'Hyperclassical Logic (a.a.a. IF Logic) and its Implications for Logical Theory', *Bulletin of Symbolic Logic* 8:3, 404–423.

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

R. Hirsch & I. Hodkinson, 2002, *Relational Algebras by Games*, Elsevier Science Publishers, Amsterdam.

W. Hodges, 1985, *Building Models by Games*, Cambridge University Press, Cambridge.

W. Hodges, 1999, *An Invitation to Logical Games*, lecture notes, department of mathematics, Queen Mary's College, London.

N. Immerman & D. Kozen, 1987, 'Definability With Bounded Number of Bound Variables', *Proceedings 2d IEEE Symposium on Logic in Computer Science*, 236-244.

D. Lewis, 1972, 'General Semantics', in D. Davidson & G. Harman, eds., *Semantics of Natural Language*, Reidel, Dordrecht, 169–218.

B. Löwe, 2002, 'Consequences of the Axiom of Blackwell Determinacy', *Bulletin of the Irish Mathematical Society* 49 (2002), p.43-69

Y. Moschovakis, 1974, *Elementary Induction on Abstract Structures*, North-Holland, Amsterdam.

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

Rohit Parikh, 1985, ‘The Logic of Games and its Applications’, *Annals of Discrete Mathematics* 24, 111–140.

M. Pauly, 2001, *Logic for Social Software*, dissertation DS-2001-10, Institute for Logic, Language and Computation, University of Amsterdam.

S. Rahman & H. Rueckert, eds., 2001, "New Perspectives in Dialogical Logic", guest issue of *Synthese* 127, Kluwer, Dordrecht.

R. Stalnaker, 1999, ‘Extensive and Strategic Form: Games and Models for Games’, *Research in Economics* 53:2, 93-291.

C. Stirling, 1999, 'Bisimulation, Modal Logic, and Model Checking Games, *Logic Journal of the IGPL 7:1*, 103–124. Special issue on Temporal Logic, edited by A. Montanari & Y. Venema.