# Logics of imperfect information: why sets of assignments?

Wilfrid Hodges
Queen Mary, University of London
w.hodges@qmul.ac.uk

## 1 The source of the question

In 1961 Leon Henkin [3] extended first-order logic by adding partially ordered arrays of quantifiers. He proposed a semantics for sentences $\phi$ that begin with quantifier arrays of this kind: $\phi$ is true in a structure $A$ if and only if there are a sentence $\phi^+$ and a structure $A^+$ such that:

- $\phi^+$ comes from $\phi$ by removing each existential quantifier $\exists y$ in the partially ordered prefix, and replacing each occurrence of the variable $y$ by a term $F(\bar{x})$ where $\bar{x}$ are the variables universally quantified 'before' $\exists y$ in the quantifier prefix (so that the new function symbols $F$ are *Skolem function symbols*),
- $A^+$ comes from $A$ by adding functions to interpret the Skolem function symbols in $\phi^+$, and
- $\phi^+$ is true in $A^+$.

For example the sentence

(1) $$\begin{matrix}(\forall x)(\exists y)\\(\forall z)(\exists w)\end{matrix}\ \psi(x,y,z,w)$$

is true in $A$ if and only if there are functions $F^A$, $G^A$ such that

(2) $$(A, F^A, G^A) \models (\forall x)(\forall z)\psi(x, F(x), z, G(z))$$

where $F$, $G$ stand for $F^A$, $G^A$ respectively.

Jon Barwise commented seventeen years later:

> ... the meaning of a branching quantifier expression of logic like:

(3)

$$\left.\begin{array}{ccc} \forall x & \text{—} & \exists y \\ \forall z & \text{—} & \exists w \end{array}\right> \psi(x, y, z, w)$$

> cannot be defined inductively in terms of simpler formulas, by explaining away one quantifier at a time. Rather, the whole block

(4)

$$\left.\begin{array}{ccc} \forall x & \text{—} & \exists y \\ \forall z & \text{—} & \exists w \end{array}\right>$$

> must be treated at once.

([1] with $\psi$ in place of Barwise's $A$, to avoid a clash of notation.) He offered a proof of what he called 'a precise version of this claim'. Unfortunately his proof proves much less than he said. It shows only that truth for Henkin's sentences is not an inductive verifiability relation in the sense of Barwise and Moschovakis [2]. The key point is that the inductive clauses can't be first-order; but this is hardly surprising.

## 2 Separating out the problems

With hindsight we can see that there are at least three problems that stand in the way of giving an inductive definition of truth for sentences with partially ordered quantifier arrays.

The first problem is that the definition needs to describe the effect of adding a single quantifier at the lefthand end of a formula. But for partially ordered quantifier arrays there may be several lefthand ends, and it makes a difference where we add the quantifier. The sentence (3) behaves quite differently from

(5)

$$\forall z \text{ — } \forall x \text{ — } \left.\begin{array}{c} \exists y \\ \exists w \end{array}\right> \psi(x, y, z, w)$$

I don't know that anybody has thought seriously about this problem; it needs a subtler notion of substitution than we are used to in logic. But in any case Jaakko Hintikka showed how to sidestep it by using a notation that shakes Henkin's formulas down into a linear form. The formula (1) above becomes

$$(\forall x)(\exists y)(\forall z/\exists y)(\exists w/\forall x)\psi(x,y,z,w) \tag{6}$$

in Hintikka's notation. (Hintikka also introduced a dual notion of falsehood in terms of Skolem functions for the universal quantifiers. Thus the slash $/\exists y$ in (6) expresses that the function for $z$ is independent of $y$.) This linear notation forms the syntax of the 'Independence-Friendly' IF logic of Hintikka and Sandu [4]. Hintikka also pointed out that the Skolem functions can be regarded as strategies in a game between $\forall$ and $\exists$; the resulting games form the game semantics for IF logic.

The second problem is that a Skolem function for an existential quantifier is a function of the preceding universally quantified variables, but not of the preceding existentially quantified ones. But for example the formula

$$(\exists w/\forall x)\psi(x,y,z,w) \tag{7}$$

gives no indication whether the variables $y$ and $z$ are going to be universally or existentially quantified; so from the formula alone we don't know what information can be fed into the Skolem functions for it.

In [5] I sidestepped this second problem by replacing the Skolem functions by general game strategies. Unlike a Skolem function, a strategy for a player can call on previous choices of either player. For games of perfect information this is a distinction without a difference; if player $\exists$ has a winning strategy using the previous choices of both players, then player $\exists$ has a winning strategy that depends only on the previous choices of player $\forall$. But for games of imperfect information, such as we have here, it makes a difference. I proposed marking the difference between Hintikka's games and mine by dropping the quantifiers after the slashes, and writing for example $(\forall z/y)$ where Hintikka writes $(\forall z/\exists y)$. In what follows I refer to the logic with my notation and the general game semantics as *slash logic*. During recent years many writers in this area (but never Hintikka himself) have transferred the name 'IF logic' to slash logic, often without realising the difference. Until the terminology settles down, we have to beware of examples and proofs that don't make clear which semantics they intend.

Given a solution to the third problem (below), solving the second problem for IF logic with the Skolem function semantics is tiresome but not difficult. The solution is to give several different interpretations for each formula, one for each guess about which free variables are going to be quantified existentially. As we add the quantifiers, we discard the wrong guesses—a bit like Cooper storage. Details are in [6]. I believe a similar trick should deal with the first problem when it has been correctly posed.

There remains the third problem, which is to find an inductive truth definition for slash logic. The paper [5] gave the trump semantics, which solves this problem.

It turned out that the main idea needed was to think of formulas as being satisfied not by assignments to their variables, but by *sets of assignments*. Why sets of assignments? Mathematically the idea is natural enough, but we can hardly see by intuition that it will work. For a while I almost convinced myself that an inductive truth definition for slash logic would need sets of sets of assignments, sets of sets of sets of assignments, and so on up the hierarchy of finite types.

An intuitive answer to the question 'Why sets of assignments?' could do marvels for making the inductive semantics of these sentences more appealing. But it's hardly clear where to look for an intuition. One obvious approach is to try generalising the semantics to other logics, in order to see what is needed where. But in the last ten years this hasn't happened. The semantics has been extended, but only to logics with broadly the same features as Henkin's original.

So we have to look elsewhere. One suggestion runs as follows. The trump semantics for formulas of slash logic was found by starting from a game semantics on sentences. The passage from sentences to games to trumps is too complex to support any strong intuition. So we should try to separate the games from the trump semantics. There are two natural ways to do this. The first way is to discard the games altogether and find a direct motivation for the trump semantics. Jouko Väänänen has made good progress in this direction [11].

The second way is to abandon the formulas and work directly with the games. This is the purpose of this paper.

## 3 The programme

We aim to extract the game-theoretic content of the games in [5], and extend it to a class of games that has no intrinsic connection with formulas or truth values. Finding the trump semantics was a matter of extending the truth values on sentences to semantic values on formulas, in such a way that the values on formulas could be built up by induction on complexity. We aim to do the same but in a purely game-theoretic setting: we define values on games, and we extend these values to values on subgames, again with the aim of defining these values by induction.

This description is too open-ended for comfort. Since we've thrown away the connection with truth, what counts as a correct extension to subgames?

Fortunately we know the formal core of the Tarski truth definition; section 4 below describes it in terms of fregean values. This formal description carries over straightforwardly from formulas to games, as soon as we have said what subgames are and what the value of a game is. Section 5 will describe the games and their subgames. Section 6 will propose suitable values for games. (There might be better choices here that I didn't think of.) Then sections 7 and 8 carry out the extension to fregean values, first for games of perfect information and then under imperfect information.

As hoped, the fregean value of a subgame is in terms of sets of assignments rather than single assignments. But also we can see a game-theoretic reason for this. The following summary will perhaps make more sense at the end of the paper, but I hope it conveys something already at this stage.

The value of a game is defined in terms of the existence of certain strategies for the players. The definition of values by induction on subgames builds up these strategies. Now one familiar move in building up a strategy $\sigma$ for a player $p$, at a place where another player $p'$ is about to move, is to find a strategy $\sigma_a$ corresponding to each possible choice $a$ by $p'$. We build $\sigma$ as follows: player $p$ waits to see what choice $a$ player $p'$ makes, and then proceeds with $\sigma_a$. But under conditions of imperfect information $p$ may not know what choice $p'$ is making. (Or to say the same thing in terms of information partitions, the composite strategy $\sigma$ may give different answers on data items in the same partition set.) So this kind of gluing is blocked. The consequence is that we can give player $p$ this strategy $\sigma$ only if we know that it works uniformly for all choices that player $p'$ can make. In other words, the information that we have to carry up from the subgames is not that certain data states allow strategies, but that certain *sets* of data states allow the same strategy. In short, we need to go up one type level in order to sidestep the fact that we can't in general glue strategies together.

## 4 Fregean values

This section summarises without proofs the main results in [8].

Suppose $\mathbb{E}$ is a set of objects called *expressions* (for example the formulas of a logic). We assume expressions can have parts that are also expressions. We write $F(\xi_1, \ldots, \xi_n)$ for an 'expression with holes', that gives an an expression $F(e_1, \ldots, e_n)$ when suitable expressions $e_1, \ldots, e_n$ are put in the holes, putting $e_i$ in hole $\xi_i$ for each $i$.

We assume given a family $\mathbb{F}$ of such 'frames' $F(\xi_1, \ldots, \xi_n)$, with four properties:

1. $\mathbb{F}$ is a set of nonempty partial functions on $\mathbb{E}$. ('Nonempty' means their domains are not empty.)

2. (Nonempty Composition) If $F(\xi_1, \ldots, \xi_n)$ and $G(\eta_1, \ldots, \eta_m)$ are frames, $1 \leqslant i \leqslant n$ and there is an expression

$$F(e_1, \ldots, e_{i-1}, G(f_1, \ldots, f_m), e_{i+1}, \ldots, e_n),$$

then $F(\xi_1, \ldots, \xi_{i-1}, G(\eta_1, \ldots, \eta_m), \xi_{i+1}, \ldots, \xi_n)$ is a frame.

3. (Nonempty Substitution) If $F(e_1, \ldots, e_n)$ is an expression, $n > 1$ and $1 \leqslant i \leqslant n$, then

$$F(\xi_1, \ldots, \xi_{i-1}, e_i, \xi_{i+1}, \ldots, \xi_n)$$

is a frame.

4. (Identity) There is a frame $1(\xi)$ such that for each expression $e$, $1(e) = e$.

Assume also that $S \subseteq \mathbb{E}$. We refer to the expressions in $S$ as *sentences*, since this is what they are in most applications to logics. Assume that a function $\mu$ with domain $S$ is given; $\mu$ gives a 'value' to each sentence.

Under these assumptions, we define a relation $\equiv$ on $\mathbb{E}$ as follows. (It expresses that two expressions make the same contribution to the $\mu$-values of sentences containing them.)

For all expressions $e$ and $f$, $e \equiv f$ if and only if

(a) for each frame $F(\xi)$ with one variable, $F(e)$ is in $S$ if and only if $F(f)$ is in $S$;

(b) whenever $F(e)$ and $F(f)$ are in $S$, $\mu(F(e)) = \mu(F(f))$.

Then $\equiv$ is an equivalence relation. If distinct values are assigned to the distinct equivalence classes, so that $|e|$ is the value assigned to the equivalence class of $e$, we call $|e|$ the *fregean value* of $e$.

**Lemma 1** *Suppose that for every expression $e$ there is a frame $F(\xi)$ such that $F(e)$ is a sentence. Then for each frame $G(\xi_1, \ldots, \xi_n)$ there is a function $h_G$ such that for all expressions $e_1, \ldots, e_n$ such that $G(e_1, \ldots, e_n)$ is an expression,*

$$|G(e_1, \ldots, e_n)| = h_G(|e_1|, \ldots, |e_n|).$$

In a common turn of phrase, Lemma 1 says that fregean values are 'compositional'. The function $h_G$ is called the *Hayyan function* of $G$ in [8].

**Lemma 2** *Let $e$ and $f$ be sentences. If $e \equiv f$ then $\mu(e) = \mu(f)$. Hence there is a function $r$ defined on the fregean values of sentences, such that for every sentence $e$, $\mu(e) = r(|e|)$.*

The message of these two lemmas together is that the function $\mu$, which is quite arbitrary, can always be defined through an inductive definition of fregean values, where the induction is on the complexity of expressions. The fregean values of atomic expressions have to be given directly as the base case. The Hayyan functions take care of the inductive steps, and finally the function $r$ reads off $\mu$ from the fregean values of the sentences. It turns out that in standard examples of truth definitions we can take $r$ to be the identity. One such case is where $E$ is the set of formulas of slash logic, $S$ is the set of sentences, $\mu$ is the assignment of truth values to sentences as given by the game semantics, and the fregean values are the values given to formulas by the trump semantics.

The intended applications of this machinery were languages. But nothing prevents us from carrying them over to games. The set $S$ will consist of the games and the set $\mathbb{E}$ will consist of the subgames of these games (in some suitable sense). We give each game $G$ a 'value' $\mu(G)$ in terms of the effects of strategies of the players, and then we compute fregean values for the subgames. This will yield an inductive definition for the values $\mu(G)$, working directly on the games without any intervention of formulas. In this setting we can test the effect of moving from games of perfect information to games of imperfect information. Does it move the values up a type?

## 5 The games

The first step in our programme is to define the games and the subgames so that the assumptions of section 4 hold. The requirements are fairly strong. The games have to be constructed inductively from their subgames. The notion of substituting one subgame for another has to make sense. So does the notion of imperfect information. Already the games start to look a little like formulas. But we can discard the notion that there are just two players, and we don't need the notions of winning and losing. The players need not be in competition with each other.

I think we need the following ingredients.

- First and trivially, there must be more than one player. (Otherwise the notion of imperfect information becomes degenerate.) We write $\mathcal{P}$ for the set of players.

- Second, as the game proceeds, the players build up a bank of data (corresponding to the assignment of elements to variables). It's enough to assume there is a set $\mathcal{Q}$ of *questions*, each of which has a nonempty set of answers. A *data state* is a function defined on a set of questions, taking each of these questions to one of its answers. We write $\mathcal{D}$ for the set of data states.

- Third, some of the choices of the players are structural; they have no effect on the data state, but they control who moves next, what the criteria are for deciding the payoffs, what information can be fed into strategies, and so forth. Thus at any stage of the play a data state $s$ and a sequence $\bar{c}$ of structural moves have been built up. The information fed into the play by the sequence $\bar{c}$ determines a 'subgame' which is to be played 'at' the data state $s$.

- Fourth, the fact that the same subgame can be played at different data states allows the possibility that the player who moves at this subgame may have incomplete information about the data state.

- Fifth, the players can move to subgames only finitely often in a play. Eventually they reach an atomic subgame; when they do, the payoff to each player is determined by the subgame and the data state. There is no need to assume that the payoffs are wins or losses; real number values will suffice, and they need not add up to zero.

- Sixth, for each subgame $H$ there is an associated set of questions $m(H)$, which we can call the *matter* of $H$, with the property that $H$ can be played as soon as we are given a data state $s$ which answers all questions in $m(H)$. (This corresponds to the free variables of a formula.) We need an assumption like this in order to make sense of substitution of subgames.

The following definitions are meant to give shape to these ingredients. They are strongly influenced by Parikh's paper [9].

We assume given the set $\mathcal{P}$ of players, the set $\mathcal{Q}$ of questions and the set $\mathcal{D}$ of data states, as above. Given a finite subset $W$ of $\mathcal{Q}$, we write $\mathcal{D} \restriction W$ for the set of all data states whose domain is $W$, and $\mathbb{R}^{\mathcal{D}\restriction W}$ for the set of all functions from $\mathcal{D} \restriction W$ to the set $\mathbb{R}$ of real numbers. If $s$ is a data state, $q$ is a question and $a$ is an answer to $q$, we write $s(q/a)$ for the data state whose domain is domain$(s) \cup \{q\}$, and which agrees with $s$ everywhere except that $s(q/a)(q) = a$.

We define inductively the set of *game parts*. The game parts will be formal sequences. Later we will explain how they are played. It might be cleaner to define the sequences first, and then define a game part $F(i)$ for each sequence $i$; but our approach saves on notation.

($\alpha$) For every finite $W \subseteq \mathcal{Q}$ and every function $f : \mathcal{P} \to \mathbb{R}^{\mathcal{D}\restriction W}$, $\langle W, f\rangle$ is a game part; its *matter* is $W$. Game parts of this form are *atomic*.

($\beta$) For every finite set $W \subseteq \mathcal{Q}$, every player $p \in \mathcal{Q}$, every partition $\pi$ of $\mathcal{D} \restriction W$, every question $q \in \mathcal{Q} \setminus W$ and every game part $J$ with matter $\subseteq W \cup \{q\}$, there is a game part $\langle W, p, \pi, q, J\rangle$ whose matter is $W$.

($\gamma$) For every finite set $W \subseteq \mathcal{Q}$, every player $p \in \mathcal{P}$, every partition $\pi$ of $\mathcal{D} \restriction W$ and every nonempty set $X$ of game parts with matter $\subseteq W$, there is a game part $\langle W, p, \pi, X\rangle$ whose matter is $W$.

For each game part $H$ with matter $W$ and each data state $s$ with domain $\supseteq W$, $H$ is played at $s$ as follows:

($\alpha$) If $H$ is $\langle W, f\rangle$ then there is an immediate payoff of $f(p)(s \restriction W)$ to each player $p$.

($\beta$) If $H$ is $\langle W, p, \pi, q, J\rangle$, then player $p$ moves by choosing an answer $a$ to the question $q$, and the play continues as a play of $J$ at the data state $s(q/a)$.

($\gamma$) If $H$ is $\langle W, p, \pi, X\rangle$, then player $p$ moves by choosing a game part $J \in X$, and the play continues as a play of $J$ at the data state $s$.

A *game* is a game part with empty matter. We say that a game part $H$ is a *subgame of* a game part $G$ if $H$ occurs as a subsequence of $G$. Note that a game part $G$ can have several occurrences of a game part $H$ in it.

Suppose a play of a game part $G$ at a state $s$ is in progress, and the players have just reached an occurrence of the subgame $H$ of $G$. Then the choices of the players consist of (1) a sequence $\bar{c}$ of subgames, starting at $G$ and finishing with $H$, namely the game parts chosen at subgames of the form ($\gamma$), and (2) a data state $t$ representing $s$ together with the choices made at subgames of the form ($\beta$). We call the pair $(\bar{c}, t)$ a *position* in the play. The pair $(\bar{c}, s)$ determines the domain of $t$ (namely, the union of the domain of $s$ and the set of questions answered at moves reported in $\bar{c}$); we call this domain the *current domain* at $(\bar{c}, s)$. The sequence $\bar{c}$ also determines the player who will move next; we call this player the *current player* at $\bar{c}$. The *current game part* at the position $(\bar{c}, t)$ is the final game part $H$ of $\bar{c}$. If this game part is of the form $\langle W, p, \pi, \ldots \rangle$, then $p$ and $\pi$ are respectively the *current player* and the *current partition* at $\bar{c}$. Here the current game part, player and partition depend only on $\bar{c}$, and the current domain depends only on $\bar{c}$ and $s$; but since the position $(\bar{c}, t)$ determines $s$, we can speak of any of these things as being *current* at the position $(\bar{c}, t)$.

A *strategy* for a player $p$ in a game part $G$ at a data state $s$ is a family $\sigma$ of functions $\sigma_{\bar{c}}$ indexed by the sequences $\bar{c}$ such that $p$ is the current player at $\bar{c}$. For each position $(\bar{c}, t)$ where $p$ is the current player, $\sigma_{\bar{c}}(t)$ is a possible move for $p$ in the current subgame. There is some redundancy here, because a strategy is defined at positions in the game which could never be reached if the strategy was followed; but this redundancy will never matter and it would be a nuisance to exclude it. Note that the strategy $\sigma$ depends on $s$ and not just on $G$; the dependence on $s$ will be important below.

So far the partitions have played no role. Their purpose is to restrict the allowed strategies, as follows. We say that a strategy $\sigma$ for player $p$ in game part $G$ at $s$ is *admissible at* $\bar{c}$ if for all data states $t, u$ with domain the current domain $W$ at $\bar{c}$, if

- $s \restriction W$ and $t \restriction W$ lie in the same partition set of the current partition at $\bar{c}$,

then

$$\sigma_{\bar{c}}(s) = \sigma_{\bar{c}}(t).$$

We say that a strategy $\sigma$ for $p$ is *admissible* if it is admissible at all $\bar{c}$ at which $p$ is the current player.

A game $G$ is *of perfect information* if all the partitions appearing in $G$ are trivial, i.e. all their partition sets are singletons. For a game of perfect information, every strategy is admissible. When we discuss games of perfect information, we can ignore their partitions; for example we can write $\langle W, p, \pi, X \rangle$ as $\langle W, p, X \rangle$.

In sections 7 and 8 below we will sometimes talk of strategies that are restricted to subsets of some set $\mathcal{D} \restriction W$. It makes sense to glue together several such strategies, provided that no two of them are defined on members of the same partition.

If $J$ is a game part occurring in the subgame $H$, and $J'$ is a subgame with the same matter as $J$, then we can form a new subgame $H(J'/J)$ by replacing the occurrence of $J$ by an occurrence of $J'$; the matter of $H(J'/J)$ is the same as that of $H$. (The notation is a shorthand; there might be other occurrences of $J$ in $H$, and these stay unchanged.)

It would be possible to substitute $J'$ for $J$ in $H$ even if the matter of $J'$ is not the same as that of $J$. But suppose the question $q$ is in the matter of $J'$ and not in that of $J$. Let $G$ be a game where some player chooses in turn answers to the questions in $m(J)$, and then the game continues as $J$. Then substituting $J'$ for $J$ in $G$ yields a subgame $G'$ whose matter contains $q$; so this substitution turns a game into a game part that is not a game. We will bar this kind of substitution. In other words, we will consider only substitutions where (a) of section 4 holds. So the significant question will be when (b) holds too.

Our notion of subgame is not the more familiar one due to Selten [10]. A Selten subgame in our context would be a pair $(G, s)$ where $G$ is a subgame that can be played at data state $s$. For us it is essential that the same subgame can be played at different data states. This is the game analogue of the fact that a subformula allows different assignments to its variables.

## 6 Game values

We define the *value* of a game $G$ to a player $p$, $\mu_p(G)$, to be the supremum of the reals $\lambda$ such that $p$ has an admissible strategy which ensures that the payoff to $p$ is at least $\lambda$. So $\mu_p(G)$ is an element of $\mathbb{R} \cup \{\pm\infty\}$. We can take the *value* $\mu(G)$ of $G$ to be the function taking each player $p$ to $\mu_p(G)$. But in fact all our calculations of values will consider one player at a time.

This is a generalisation of the assignment of truth values to sentences in logic. The logical case is where there are two players, the payoffs are all either 0 or 1, and for any atomic game part at any data state the payoffs to the two players add up to 1. One could generalise in other ways (for example taking values in a complete boolean algebra), but I chose something simple that seems to fit with the habits of game theory.

Now that we have the values of games, we can apply the framework of section 4. We take $\mathbb{E}$ to be the set of game parts, $S$ to be the set of games and $\mu$ to be the value function just defined. Every game part is a subgame of a game, so that the hypothesis of Lemma 1 holds. (It would have failed if we allowed the matter of a game part to be infinite, since only finitely many questions get answered during a play.)

Following section 4 we define a relation $\equiv$ on subgames: $H \equiv J$ if and only if

(a) $H$ and $J$ have the same matter, and

(b) for every game $G$ containing an occurrence of $H$, $\mu(G(J/H)) = \mu(G)$.

Then $\equiv$ is an equivalence relation on $\mathbb{E}$. If we can identify the equivalence classes, we can label them with fregean values, and then we have a definition of $\mu$ by induction on subgames.

## 7 The extension under perfect information

*In this section all games are of perfect information, so that all strategies are admissible.* Here the situation is familiar enough to suggest where to look for fregean values.

**Definition 3** *Let $G$ be a game part.*

*(a) Let $p$ be a player and $s$ a data state with domain $\supseteq m(G)$. Define the* value *of $G$ at $s$ to $p$, $v_p(G, s)$, by:*

$$v_p(G, s) = \sup\{\lambda : p \textit{ has a strategy which, when } G \textit{ is played at state } s, \textit{ guarantees that the payoff to } p \textit{ will be at least } \lambda\}.$$

*(b) We define $v_p(G)$ to be the function with domain $\mathcal{D} \restriction m(G)$, whose value for each $s$ in this set is $v_p(G, s)$.*

*(c) We define $v(G)$ to be the function with domain $\mathcal{P}$, whose value for each player $p$ is $v_p(G)$.*

In the case where $G$ is a game, $v_p(G) = \mu_p(G)$ for each player $p$, and so $v(G) = \mu(G)$. In the case where $G$ is atomic, there is an immediate payoff to each player for each $s \in \mathcal{D} \restriction m(G)$, and $v(G)$ records these payoffs.

The definition of $v(G)$ depends only on the values $v(G, s)$ where $s$ has domain $m(G)$. But $G$ can be played at data states $s$ with much larger domains. We need to show that for these $s$ the values $v(G, s)$ are determined by $v(G)$. (This is fundamental. If it failed, the values $v(G)$ wouldn't be fregean values obeying the conclusion of Lemma 1.)

**Lemma 4** *(Under perfect information.) Suppose a game part $G$ has matter $W$, and $s$ is a data state with domain $W' \supseteq W$. Let $p$ be a player. Then the value of $G$ at $s$ for $p$ is equal to the value of $G$ at $s \restriction W$ for $p$.*

**Proof** Suppose $\sigma$ is a strategy for $p$ in $G$ at $s \upharpoonright W$ that guarantees $p$ a payoff of at least $\lambda$. Let $\tau$ be the following strategy for $p$ in $G$ at $s$: ignore any answers to questions not in $W$, and use $\sigma$. Induction on the complexity of $G$ shows that this is a strategy for $p$ in $G$ at $s$; the ignored values are never needed, and in particular they make no difference to the payoff. Thus $\tau$ guarantees payoff at least $\lambda$ for $p$.

Conversely suppose $\tau$ is a strategy for $p$ in $G$ at $s$ that guarantees $p$ a payoff of at least $\lambda$. Then let $\sigma$ be the strategy for $p$ in $G$ at $s \upharpoonright W$ that uses $\tau$, filling in the extra values from $s$. Again $\sigma$ guarantees payoff at least $\lambda$ to $p$. □

**Theorem 5** *(Under perfect information.) Suppose $H$ and $H'$ are game parts with the same matter. Then $H \equiv H'$ if and only if $v(H) = v(H')$.*

**Proof** We fix a player $p$. Assuming that $v_p(H) = v_p(H')$, we prove that for every game part $G$ in which $H$ occurs as a subgame, $v_p(G) = v_p(G(H'/H))$. The proof is by induction on the complexity of $G$. By the lemma, we need only show that $v_p(G, s) = v_p(G(H'/H), s)$ when $s$ has domain $m(G)$.

($\alpha$) Suppose first that $G = H$. Then $G(H'/H) = H'$, so the result is immediate.

($\beta$) Suppose that $p'$ is a player, $G$ is $\langle W, p', q, J\rangle$, $s$ is a data state with domain $m(G)$, and $v_p(G, s) = \lambda$. Then for every $\lambda' < \lambda$, $p$ has a strategy for $G$ at $s$ which guarantees $p$ a payoff of at least $\lambda'$. We aim to show the same for $G(H'/H)$. There are two cases, according as $p'$ is $p$ or another player.

Suppose first that $p' \neq p$. Then

> For each $\lambda' < \lambda$ and each possible choice $a$ of $p'$ at $G$, there is a strategy $\sigma_a$ for $p$ for $J$ at $s(a/q)$ which guarantees $p$ a payoff of at least $\lambda'$.

Now by induction hypothesis $v_p(J) = v_p(J(H'/H))$, so the lemma tells us that $v_p(J, s(a/q)) = v_p(J(H'/H), s(a/q))$ for each $a$. It follows that for each $a$, player $p$ has a strategy $\tau_a$ for $J(H'/H)$ at $s(a/q)$ which guarantees $p$ a payoff of at least $\lambda'$. For each $\lambda' < \lambda$ we can glue these strategies together to produce a strategy $\tau$ for $p$ in $G$ at $s$, namely: Wait for the choice $a$ and then play $\tau_a$. This strategy guarantees $p$ a payoff of at least $\lambda'$. Hence again $v_p(G(H'/H), s) \geqslant v_p(G, s)$, and symmetry gives the converse.

The other case, where $p'$ is $p$, is similar but easier. The strategy $\sigma$ for $p$ at $G$ chooses an element $a$ to answer $q$, and then we need only consider $s(a/q)$ for this $a$, so that no gluing is needed.

($\gamma$) Suppose $p'$ is a player and $G$ is $\langle W, p', q, X\rangle$. Then the argument of case ($\beta$) applies with appropriate changes. Note that since the different subgame occurrences have their own strategy functions, there is no need for any gluing in this case.

This proves one direction. For the other, suppose $v_p(H) < v_p(H')$, and choose $\lambda$ with $v_p(H) < \lambda < v_p(H')$. Then $p$ has no strategy in $H$ that guarantees that for all $s$, $p$ will get payoff $\lambda$. Hence there is some $s$ such that $p$ can't guarantee to get $\lambda$, playing at $s$. Consider the game where some other player chooses assignments to the domain of $s$, then $p$ picks up and plays $H$. In this game $G$ player $p$ can't guarantee to get payoff $\lambda$, since the other player could play $s$. But by assumption player $p$ can guarantee to get payoff $\lambda$ in $G(H'/H)$. Hence $\mu(G(H'/H)) \neq \mu(G)$, so that $H \not\equiv H'$. $\square$

Suppose we restrict to any smaller class of games which is closed under substitution of subgames with the same matter, and under ($\beta$) of section 5. (An example of such a class is where in ($\gamma$) we require the set $X$ to be finite. This comes nearest to first-order logic.) Then the entire argument above goes through.

## 8 The extension under imperfect information

We turn to our major question. What is needed to repair the proof of Theorem 5 if we drop the assumption of perfect information?

Lemma 4 doesn't survive unaltered, but with a suitable definition of values we can still get the main point of the lemma, which is that the values at data states whose domain is the matter determine the values at all other data states. The proof of Theorem 5 also goes through except for one point: in the gluing at case ($\beta$), nothing guarantees that the resulting strategy $\tau$ is admissible. A little meditation shows that the problem is serious. No information about the existence of separate admissible strategies $\tau_a$ for $J(H'/H)$ at $s(a/q)$ is going to guarantee a single admissible strategy for $G(H'/H)$ at $s$.

So we have to carry up inductively the information that certain *sets* of data states lie within the domains of admissible strategies.

**Definition 6** *Let $G$ be a game part.*

*(a) Let $p$ be a player, and let $W$ be a finite set of questions $\supseteq m(G)$. Define the*

value *of $G$ at a subset $S$ of $\mathcal{D} \upharpoonright W$ to $p$, $v_p(G, S)$, by:*

$$v_p(G,S) = \sup\{\lambda : p \text{ has an admissible strategy which, when } G \text{ is played at any state } s \in S, \text{ guarantees that the payoff to } p \text{ will be at least } \lambda\}.$$

(b) *We define $v_p(G)$ to be the function with domain the power set of $\mathcal{D} \upharpoonright m(G)$, whose value for each subset $S$ of $\mathcal{D} \upharpoonright m(G)$ is $v_p(G, S)$.*

(c) *We define $v(G)$ to be the function with domain $\mathcal{P}$, whose value for each player $p$ is $v_p(G)$.*

Then as before, $v(G) = \mu(G)$ whenever $G$ is a game.

Suppose $W' \supseteq W$ and $S$ is a subset of $\mathcal{D} \upharpoonright W'$. We say that $s, t$ in $S$ are *in the same fibre* of $S$ along $W$ if $s \upharpoonright (W' \setminus W) = t \upharpoonright (W' \setminus W)$. This defines an equivalence relation, and its equivalence classes are called the *fibres* of $S$ along $W$.

**Lemma 7** *Suppose a game part $G$ has matter $W$, $W'$ is a finite set $\supseteq W$, and $S$ is a set of data states with domain $W'$. Let $p$ be a player. Then the value of $G$ at $S$ for $p$ is equal to the infimum of the values of $G$ at the fibres of $S$ along $W$ for $p$.*

**Proof**. (Cf. Lemma 7.4 of [5].) Let $\lambda$ be the infimum of the values of $G$ at the fibres of $S$ along $W$ for $p$. Then for each $\lambda' < \lambda$ and each fibre $\phi$ of $S$ along $W$, there is an admissible strategy $\sigma_\phi$ for $p$ on $\phi$, which guarantees $p$ a payoff of at least $\lambda'$. Fixing $\lambda'$, glue together these strategies on the separate fibres, to get a strategy $\sigma$ on $W$. In the definition of admissibility, elements of different fibres never agree off $W$; so the admissibility of the $\sigma_\phi$ guarantees that $\sigma$ is admissible. Thus the value of $G$ at $S$ for $p$ is at least $\lambda$.

An easier argument in the other direction shows that if the value of $G$ at $S$ for $p$ is at least $\lambda$, then the value at each fibre is at least $\lambda$ too. □

We repeat the Theorem, but under imperfect information and with the new definition of $v$.

**Theorem 8** *Suppose $H$ and $H'$ are game parts with the same matter. Then $H \equiv H'$ if and only if $v(H) = v(H')$.*

**Proof**. We fix a player $p$. Assuming that $v_p(H) = v_p(H')$, we prove that for every game part $G$ in which $H$ occurs as a subgame, $v_p(G) = v_p(G(H'/H))$. The proof is by induction on the complexity of $G$. By the lemma, we need only show that $v_p(G, S) = v_p(G(H'/H), S)$ when $S \subseteq \mathcal{D} \restriction m(G)$.

($\alpha$) The case where $G = H$ is as before.

($\beta$) Suppose that $p'$ is a player, $G$ is $\langle W, p', \pi, q, J\rangle$, $S \subseteq \mathcal{D} \restriction W$ and $v_p(G, S) = \lambda$. Then for every $\lambda' < \lambda$, $p$ has an admissible strategy for $G$ at $S$ which guarantees $p$ a payoff of at least $\lambda'$. We aim to show the same for $G(H'/H)$. There are two cases, according as $p'$ is $p$ or another player.

Suppose first that $p' = p$. Then:

> there is an admissible function $\sigma$ for $p$ such that $p$ has an admissible strategy for $J$ at $S^\sigma = \{s(\sigma(s)/q) : s \in S\}$ which guarantees $p$ a payoff of at least $\lambda'$.

Now by induction hypothesis $v_p(J) = v_p(J(H'/H))$. Hence by the lemma, $v_p(J, S^\sigma) = v_p(J(H'/H), S^\sigma)$. It follows that $p$ has an admissible strategy for $J(H'/H)$ at $S^\sigma$ which guarantees $p$ a payoff of at least $\lambda'$. Combining this strategy with $\sigma$, $p$ has an admissible strategy for $G(H'/H)$ at $S$ which guarantees a payoff of at least $\lambda'$. Thus $v_p(G) \leqslant v_p(G(H'/H))$, and symmetry gives the converse.

Next, suppose $p' \neq p$. The argument is the same, except that in place of $S^\sigma$ we use $S^q = \{s(a/q) : s \in S, a \text{ an answer to } q\}$ .

($\gamma$) Suppose that $p'$ is a player, $G$ is $\langle W, p', \pi, X\rangle$ and $S \subseteq \mathcal{D} \restriction W$. One can adjust the arguments of case ($\beta$) to this case without needing any new ideas.

Now conversely suppose that $m(H) = m(H') \neq \emptyset$, and for some $S$ with domain $m(H) = \{q_1, \ldots, q_k\}$, $v_p(H, S) < v_p(H', S)$. Then the same inequality must hold for some nonempty intersection of $S$ with a class of the current partition at $H$; so we can assume that $S$ lies in a single class of this partition. Choose $\lambda, \lambda'$ with $v_p(H, S) < \lambda' < \lambda < v_p(H', S)$. Let $f : \mathcal{P} \to \mathbb{R}^{\mathcal{D}\restriction m(H)}$ be the function that takes each player $p' \neq p$ to the constant function with value $0$, and that satisfies

$$f(p)(s) = \begin{cases} \lambda' & \text{if } s \in S, \\ \lambda & \text{otherwise.} \end{cases}$$

Let $p_0$ be some player other than $p$, and let $G$ be the game

$$\langle \emptyset, p_0, q_1, \langle \{q_1\}, p_0, q_2, \langle \ldots, q_k, \langle m(H), p, \{H, \langle m(H), f\rangle\}\rangle \ldots \rangle$$

where the missing partitions are all trivial, so that the information is perfect except perhaps within $H$.

In the game $G$, player $p_0$ can use the first $k$ moves to pick an element of $S$. Then by assumption $p$ has no admissible strategy in $H$ guaranteeing a payoff $\geqslant \lambda'$, and choosing $\langle m(H), f\rangle$ guarantees $p$ a payoff of only $\lambda'$. So $v_p(G) \leqslant \lambda'$. On the other hand $p$ has a strategy for $G(H'/H)$ which guarantees a payoff of at least $\lambda$. Namely, if $p_0$ chooses $s$ in $S$, then pick $H'$ and play a suitable admissible strategy at $S$ in $H'$; if $p_0$ chooses outside $S$, then choose $\langle m(H), f\rangle$ and collect $\lambda$. Since $G$ is a game, it follows that $H \not\equiv H'$. □

Just as in the previous section, the theorem still holds good if we restrict to a class of games with reasonable closure conditions. I omit details.

## 9 Conclusion

In the games above, we get fregean values for game parts by assigning values to sets of data states, not to single data states. This is the exact analogue of what happens in the semantics for slash logic as in [5]. The proofs make clear why this is the right level: in some sense the argument was *always* about sets of data states rather than data states one at a time—but in the case of perfect information we could disguise this fact by taking the data states one at a time and then gluing.

The games above do look rather like formulas. (I don't know whether they have any other application.) But our arguments show that some features of logical formulas are irrelevant to the fact that fregean values go with sets of data states. In particular the number of players is irrelevant as long as it is at least two. Competition between the players is irrelevant. Truth (as opposed to real number values) is also irrelevant. Last but not least, the information partitions that we allowed are much more general than those that arise from IF or slash logic.

**Acknowledgements** My warm thanks to the organisers of 'Interactive Logic: Games and Social Software' for a very stimulating meeting, and for giving me the opportunity to present this paper. The paper was completed during a visit to the Isaac Newton Institute in Cambridge.

## References

[1] Jon Barwise, 'On branching quantifiers in English', *Journal of Philosophical Logic* 8 (1978) 47–80.

[2] Jon Barwise and Yiannis N. Moschovakis, 'Global inductive definability', *Journal of Symbolic Logic* 43 (1978) 521–534.

[3] Leon Henkin, 'Some remarks on infinitely long formulas', in *Infinitistic methods: Prococeedings of symposium on foundations of mathematics*, Pergamon Press and Państwowe Wydawnictwo Naukowe, Warsaw 1961, pp. 167–183.

[4] Jaakko Hintikka and Gabriel Sandu, 'Game-theoretical semantics', in *Handbook of logic and language*, ed. Johan van Benthem and Alice ter Meulen, Elsevier 1996, pp. 361–410.

[5] Wilfrid Hodges, 'Compositional semantics for a language of imperfect information', *Logic Journal of the IGPL* **5** (1997) 539–563.

[6] Wilfrid Hodges, 'Some strange quantifiers', in *Structures in logic and computer science*, ed. Mycielski, J. et al., Lecture Notes in Computer Science 1261, Springer, Berlin 1997, pp. 51–65.

[7] Wilfrid Hodges, 'Formal features of compositionality', *Journal of Logic, Language and Information* 10 (2001) 7–28.

[8] Wilfrid Hodges, 'From sentence meanings to full semantics', Research Proceedings of Conference 'Logic 2005', Indian Institute of Technology in Bombay, ed. Amitabha Gupta et al. (to appear).

[9] Rohit Parikh, 'Propositional logics of programs: new directions', in *Foundations of Computation Theory*, Lecture Notes in Computer Science 158, Springer, Berlin 1983, pp. 347–359.

[10] Reinhard Selten, 'Spieltheoretische Behandlung eines Oligopolmodells mit Nachfrageträgheit', *Zeitschrift für die gesamte Staatwissenschaft* 121 (1965) 301–324, 667–689.

[11] Jouko Väänänen and Wilfrid Hodges, 'Dependence of variables construed as an atomic formula' (Preprint).