

# On Venema

Godhuli Mukherjee

September 29, 2025

## Abstract

This report takes a tour through Venema’s approach to the axiomatization of game equivalence. We start with outcome relations and their tame little properties, shift gears into a logical language, and then trade syntax for algebra via board and game algebras. The centerpiece is the representation theorem that every game algebra is secretly just a board algebra in disguise. Along the way we meet modules, prime filters, and separability tricks. Compared with Goranko’s syntactic route through normal forms, Venema’s path is more semantic and algebraic—less about juggling formulas and more about building structures.

## Contents

<b>1</b>	<b>The Set-up</b>	<b>2</b>
<b>2</b>	<b>Capturing all this in language of logic and raising the axiomatization problem</b>	<b>3</b>
<b>3</b>	<b>Reformulating in the language of algebras</b>	<b>3</b>
3.1	Board Algebras: A class of concrete algebras . . . . .	4
3.2	Rephrasing the axiomatization problem . . . . .	5
3.3	Game Algebras: A class of abstract algebras . . . . .	5
<b>4</b>	<b>Statement of the central Theorem and how it solves the problem of axiomatization</b>	<b>5</b>
<b>5</b>	<b>Diving into the proof</b>	<b>6</b>
5.1	A bird’s eye view of the task ahead . . . . .	6
5.2	Jigsaw pieces of the task, zoomed out . . . . .	6
5.3	Repository of tools . . . . .	7
5.4	The Pieces: up close . . . . .	10
<b>6</b>	<b>Concluding Remarks</b>	<b>14</b>

# 1 The Set-up

Let us begin with a bare-bones understanding of the framework on which this discussion will be mounted. The basic idea is as follows: Assume that we are dealing with 2 players, called 0 and 1, and with a game board which for the moment will be just a set  $B$  of objects that we call states or positions.

With any game  $g$  and each player  $i$  we will associate a outcome relation  $R_g^i$ ; i.e, a relation between positions and set of positions.

Intuitively, if a position  $p$  is in the relation  $R_g^i$  with the set  $T$  of positions, this means that in the position  $p$ , player  $i$  has a strategy of playing the game  $g$  in such a way that after play, the resulting state belongs to the set  $T$ .

In this paper we confine ourselves to the following restrictions on the outcome relations:

**Monotonicity:** If  $pR_g^i$  and  $T \subseteq U$  then  $pR_g^i$

**Consistency:** If  $pR_g^i$  then not  $pR_g^{1-i}(B - T)$

Here  $i=0,1$ .

Let us pause to unpack the meaning.

What does monotonicity capture?

If  $T$  is a set of positions that is contained in another set of positions  $U$ , then if at  $p$ ,  $i$  can force the outcome of  $g$  to lie in  $T$ , then  $i$  can force the outcome of  $g$  to be in  $U$ . Lying in  $T$  means lying in  $U$ . So the property of monotonicity captures this idea.

What does this consistency condition entail?

It says that the two players cannot simultaneously guarantee complementary outcomes from the same position. If player  $i$  can force the game into some set of positions  $T$  starting from  $p$ , then the opponent (player  $1 - i$ ) cannot, from that same position, force the game into the complement of  $T$ . In other words, strategies of the two players are mutually exclusive: one player's ability to secure an outcome rules out the other's ability to guarantee its negation. This reflects the intuitive idea that the game should not allow contradictory guarantees.

From here we have a natural notion of equivalence between games, making  $g$  and  $h$  equivalent on a certain game board if both players have the same power in  $g$  as in  $h$ ; that is, if  $R_g^i = R_h^i$  for each player  $i$ .

Below we list some of the most natural game operations that one could consider in this context:

**choice:**  $g \vee_i h$  is the game in which the first move is that player  $i$  chooses whether to play  $g$  or  $h$ .

**dualization:**  $-g$  is the game  $g$  but with the roles of the two players reversed.

**composition:**  $g \diamond h$  is the game in which a play of  $g$  is followed by a play of  $h$  repeatedly.

We have the following natural definitions (1) of outcome relations of composed games:

- $pR_{g \vee_i T}^i$  iff  $pR_g^i T$  or  $pR_h^i T$
- $pR_{g \vee_{1-i} h T}^i$  iff  $pR_g^i T$  and  $pR_h^i T$
- $pR_{-g}^i T$  iff  $pR_g^{1-i} T$
- $pR_{-g}^i T$  iff  $pR_g^{1-i} T$
- $pR_{g \diamond h}^i$  iff  $pR_g^i U$  for some set  $U$  such that  $uR_h^i T$  for all  $u \in U$

It is easily verifiable that the conditions of monotonicity and consistency are propagated. Notion of game equivalence also extends.

## 2 Capturing all this in language of logic and raising the axiomatization problem

Define game language comprising of atomic games and function symbols  $\{\vee_0, \vee_1, -, \diamond\}$ . A game expression is a term in this game language built out of atomic games and function symbols.

Eg:  $g \vee_0 h, g \vee_1 h, -g, g \diamond h, g \diamond -h$  etc

Models for this language are called game boards. A game board is a pair  $\mathcal{B} = (B, R)$  such that  $B$  is a set of positions and  $R$  is a map assigning to each atomic game  $x$  a pair  $(R_x^0, R_x^1)$  of outcome relations on  $B$  that are monotone and consistent.

Inductively, we can use (1) to define outcome relations for composed games.

We say that two game expressions are equivalent on a game board  $\mathcal{B} = (B, R)$ , notation:  $\mathcal{B} \models g \approx h$  if  $R_g^i = R_h^i$  for each player  $i$ .

We call  $g$  and  $h$  are equivalent if they are equivalent on every game board.

**An obvious problem is to find a complete axiomatization of this semantic notion of equivalence**

## 3 Reformulating in the language of algebras

We have seen that in a game board  $B$ , for any atomic  $g$ , we have  $(R_0^g, R_1^g)$ : a pair of monotone and consistent outcome relations on  $B$ . They give rise to monotone outcome relations for compound games too, following the reduction rules in (1).

Thus,

$$R_i^g \subseteq \mathcal{P}(B \times \mathcal{P}(B)), \quad i = 0, 1.$$

This motivates the definition of a class of concrete algebras called *board algebras*.

### 3.1 Board Algebras: A class of concrete algebras

Fix a set  $B$ .

- $\mathcal{O}(B) = \mathcal{P}(B \times \mathcal{P}(B))$ : all outcome relations on  $B$ .
- $\mathcal{O}_m(B) \subseteq \mathcal{O}(B)$ : the set of monotone outcome relations.
- $\mathcal{G}(B)$ : the set of pairs of outcome relations,

$$\mathcal{G}(B) = \mathcal{O}(B) \times \mathcal{O}(B).$$

- $\mathcal{G}_m(B)$ : the set of pairs of monotone outcome relations,

$$\mathcal{G}_m(B) = \mathcal{O}_m(B) \times \mathcal{O}_m(B).$$

- $\mathcal{G}_{mc}(B)$ : the set of consistent pairs of monotone outcome relations.

Consider the following operations on  $M = \mathcal{G}(B)$ :

$$(R_1, R_2) \sqcup (S_1, S_2) = (R_1 \cup S_1, R_2 \cup S_2) \quad \text{representing } g \vee h,$$

$$(R_1, R_2) \sqcap (S_1, S_2) = (R_1 \cap S_1, R_2 \cap S_2) \quad \text{representing } g \wedge h,$$

$$(R_1, R_2)^- = (R_2, R_1) \quad \text{representing } \neg g,$$

$$(R_1, R_2) \square (S_1, S_2) = (R_1 \circ S_1, R_2 \circ S_2) \quad \text{representing } g \diamond h,$$

where

$$R \circ S := \{(p, T) \mid (p, U) \in R \text{ for some } U \subseteq T \text{ and } \forall u \in U, (u, T) \in S\}.$$

Now define the **full outcome algebra** on  $B$  to be the structure

$$\mathcal{O}(B) = (\mathcal{G}(B), \sqcup, \sqcap, ^-, \square).$$

Subalgebras of  $\mathcal{O}(B)$  are called **outcome algebras on  $B$** .

An outcome algebra of the form  $(A, \sqcup, \sqcap, ^-, \square)$  is called **monotone** if  $A \subseteq \mathcal{G}_m(B)$  and a **board algebra** if  $A \subseteq \mathcal{G}_{mc}(B)$ .

The **full monotone outcome algebra** on  $B$  and the **full board algebra** on  $B$  are defined as the structures

$$\mathcal{G}_m(B) = (\mathcal{G}_m(B), \sqcup, \sqcap, ^-, \square), \quad \mathcal{G}_{mc}(B) = (\mathcal{G}_{mc}(B), \sqcup, \sqcap, ^-, \square),$$

respectively.

The class of board algebras is denoted by  $\mathbb{B}$ .

Equivalence of game expressions can now simply be stated as the validity of the corresponding game equation in the class of board algebras.

**Definition:** Two game expressions  $g$  and  $h$  are called equivalent, notation:  $\mathbb{B} \models g \approx h$ , if the equation  $g \approx h$  is valid in every board algebra.

### 3.2 Rephrasing the axiomatization problem

The problem of axiomatizing the notion of game equivalence thus reduces to finding an axiomatization for the set of game equations that are valid in the class of board algebras, i.e. finding a set of equations that make  $g \approx h$  provable in the syntax.

### 3.3 Game Algebras: A class of abstract algebras

The following are Van Benthem's proposed axioms.

$$x \vee x \approx x \quad (\text{G1})$$

$$x \vee y \approx y \vee x \quad (\text{G2})$$

$$x \vee (y \vee z) \approx (x \vee y) \vee z \quad (\text{G3})$$

$$x \vee (x \wedge y) \approx x \quad (\text{G4})$$

$$x \vee (y \wedge z) \approx (x \vee y) \wedge (x \vee z) \quad (\text{G5})$$

$$-( -x) \approx x \quad (\text{G6})$$

$$-(x \vee y) \approx -x \wedge -y \quad (\text{G7})$$

$$(x \circ y) \circ z \approx x \circ (y \circ z) \quad (\text{G8})$$

$$(x \vee y) \circ z \approx (x \circ z) \vee (y \circ z) \quad (\text{G9})$$

$$-x \circ -y \approx -(x \circ y) \quad (\text{G10})$$

$$y \preceq z \rightarrow x \circ y \preceq x \circ z \quad (\text{G11})$$

I would like to draw your attention to G11. What G11 tells us is that monotonicity is built into  $\diamond$ . This observation will be useful later.

Here  $s \preceq t$  is an abbreviation of the equation  $s \vee t \approx t$ .

A **distributive lattice** is any algebra  $\mathcal{D} = (D, \vee, \wedge)$  satisfying the equations G1–G5; a **de Morgan lattice** is an algebra  $\mathcal{M} = (M, \vee, \wedge, -)$  satisfying the equations G1–G7. Finally, a **game algebra** is a structure  $\mathcal{G} = (G, \vee, \wedge, -, \circ)$  satisfying the axioms G1–G11.

We let  $\mathbf{G}$  denote the class of game algebras. Existence of game algebra is guaranteed by existence of board algebras.

## 4 Statement of the central Theorem and how it solves the problem of axiomatization

**Theorem:** Every game algebra is isomorphic to a board algebra.

$$\mathcal{G} = \mathbb{B}$$

So let us try to see why this solves the axiomatization problem.

We get  $G \models g \approx h \forall G$  iff  $\mathbb{B} \models g \approx h \forall \mathbb{B}$

By **Birkhoff Completeness Theorem** for equational logic, we get  $g \approx h$  is derivable from G1-11 iff  $G \models g \approx h$ . So  $G \models g \approx h \forall G$  iff  $\mathbb{B} \models g \approx h \forall \mathbb{B}$  iff  $g \approx h$  is derivable from G1-11.

So,  $g$  and  $h$  are equivalent game expressions iff  $g \approx h$  is derivable from G1-11.

## 5 Diving into the proof

### 5.1 A bird's eye view of the task ahead

Let us briefly sketch the intuitions underlying the proof.

Obviously, the first problem that we encounter when trying to represent a game algebra as a board algebra is to find a suitable board. We shall find this. Details to be divulged later.

Now consider the maps

$$\begin{aligned} \diamond_g : G &\rightarrow G \\ \diamond_g a &= g \diamond a \end{aligned}$$

These maps are monotone by G11. So this gives us idea that  $\diamond_g$  can be 'thought' of as some monotone outcome relation on the 'board' lurking in the background (to be brought to light soon).

So with every element  $g$  we can associate a monotone outcome relation on the 'board'. Right now represented by  $\diamond_g$ .

However, there are still two problems we run into:

1. This association  $g \mapsto \diamond_g$  need not be injective. To resolve this we will involve a separation argument, by adding elements to  $\mathbb{G}$ , one of which will separate  $g$  and  $h$ , two distinct elements.
2. So far with all this (vague) construction of the isomorphism that we ought to make, we are at the level of monotone only. We must have consistency too to be in the realm of board algebra. We will show that any monotone outcome algebra can be embedded in a board algebra.

### 5.2 Jigsaw pieces of the task, zoomed out

**Definition:** Let  $\mathcal{G} = (G, \vee, \wedge, \neg, \circ)$  be a game algebra. A *module over  $\mathcal{G}$*  is an algebra  $\mathcal{M} = (M, \vee, \wedge, \neg, (\diamond_g)_{g \in G})$  such that  $(M, \vee, \wedge, \neg)$  is a de Morgan algebra, and  $(\diamond_g)_{g \in G}$  is a family of unary monotone operations on  $M$  satisfying the following equations:

$$\begin{aligned} (M1) \quad & \diamond_{g \vee h} x \approx \diamond_g x \vee \diamond_h x \\ (M2) \quad & \diamond_{g \wedge h} x \approx \diamond_g x \wedge \diamond_h x \\ (M3) \quad & \diamond_{g \circ h} x \approx \diamond_g \diamond_h x \\ (M4) \quad & \diamond_{-g} x \approx -\diamond_g -x \end{aligned}$$

A game module is *separable* if for all distinct elements  $g$  and  $h$  of  $G$  there is an  $x \in M$  such that  $\Diamond_g x \neq \Diamond_h x$ .

Note that with this definition, we may indeed see a given game algebra  $\mathcal{G} = (G, \vee, \wedge, \neg, \circ)$  as a module over its de Morgan reduct if we put  $\Diamond_g a = g \circ a$ . We will not introduce any notation to distinguish these two perspectives on game algebras.

Our proof of the representation theorem for game algebras involves the following three steps:

1. We will prove that every game algebra, seen as a module over itself, can be embedded in a *separable* module over itself. From this it follows that over every game algebra there is a separable module.
2. We will prove that if  $\mathcal{M}$  is a separable module over  $\mathcal{G}$ , then  $\mathcal{G}$  is isomorphic to some monotone outcome algebra over  $\mathcal{M}$ .
3. Finally, we will prove that any monotone outcome algebra can be embedded in a board algebra, cf. Proposition 6.1.

The proof of Theorem is immediate from these results.

### 5.3 Repository of tools

**Definition:** Let  $\mathcal{D} = (D, \vee, \wedge)$  be a distributive lattice. A *filter* is a subset  $F$  of  $D$  which is upward closed (if  $a \in F$  and  $a \leq b$  then  $b \in F$ ) and closed under meets (if  $a, b \in F$  then  $a \wedge b \in F$ ). A filter  $F$  is *prime* if  $a \vee b \in F$  implies that at least one of  $a$  and  $b$  belongs to  $F$ .

Let  $B_{\mathcal{D}}$  denote the set of prime filters of  $\mathcal{D}$ . Given an element  $a \in D$ , define

$$\hat{a} = \{p \in B_{\mathcal{D}} \mid a \in p\},$$

that is:  $\hat{a}$  denotes the set of prime filters to which  $a$  belongs.

**Definition:** Let  $\mathcal{D} = (D, \vee, \wedge)$  be a distributive lattice. Given a set  $T$  of prime filters, let  $F_T$  denote the set of elements  $a$  of  $D$  such that  $a \in p$  for every  $p \in T$ , or, equivalently,

$$F_T = \{a \in D \mid T \subseteq \hat{a}\}.$$

A set  $C$  of prime filters is *closed* if it is the intersection of sets of the form  $\hat{a}$ ; or, equivalently, if

$$C = \bigcap \{\hat{a} \mid a \in F_C\}.$$

Given a set  $T$  of prime filters, let  $\overline{T}$  be the smallest closed superset of  $T$ ; it is not hard to see that

$$\overline{T} = \bigcap_{a \in F_T} \hat{a}.$$

**Definition:** Let  $\mathcal{D} = (D, \vee, \wedge)$  be a distributive lattice. A map  $\Diamond : D \rightarrow D$  is *monotone* if  $\Diamond a \leq \Diamond b$  whenever  $a \leq b$ .

A **monotone lattice expansion** is an algebra  $(D, \vee, \wedge, \diamond)$  such that  $\diamond$  is a monotone operation on the distributive lattice  $(D, \vee, \wedge)$ . All definitions concerning distributive lattices apply to monotone lattice expansions as well.

The prime example of monotone lattice operations stems from monotone outcome relations. Let  $R$  be an outcome relation on  $B$ , and define the operation  $m_R : \mathcal{P}(B) \rightarrow \mathcal{P}(B)$  by

$$m_R(T) = \{p \in B \mid pRT\}.$$

It is easy to verify that  $R$  is monotone if and only if  $m_R$  is a monotone relation on the power set lattice of  $B$ .

**Definition:** Given a monotone lattice expansion  $\mathcal{D} = (D, \vee, \wedge, \diamond)$ , let  $Q_\diamond$  be the outcome relation on the board of prime filters of  $\mathcal{D}$  given by

$$pQ_\diamond T \quad \text{iff} \quad \diamond a \in p \text{ for all } a \in F_T.$$

It can be checked this outcome relation is monotone.

Below we state and sketch the proof of The Prime Filter Theorem which will be useful now and what is to come.

**Theorem:**  $(D, \vee, \wedge)$  be a distributive lattice and let  $F$  be a filter with  $x_0$ . Then there is a prime filter  $P$  such that  $x_0 \notin P$  and  $F \subseteq P$ .

**Proof:** The proof idea is as follows: We can consider the set of all such filters  $K$  with the property that  $x_0 \notin K$ . This set is non-empty. It can be shown every chain has an upper bound. So by Zorn's lemma, this set will have a maximal element. This maximal element can be shown to qualify as the prime filter we are looking for.

**Proposition:** For any distributive lattice  $\mathcal{D} = (D, \vee, \wedge, \diamond)$ , the map  $(\cdot)^\wedge$  is an embedding of  $D$  in  $(\mathcal{P}(B_D), \cup, \cap, m_{Q_\diamond})$ .

**Proof:** Checking that the binary operations are respected follows from definitions of  $\wedge$ , filter and a prime filter.

That this map is injective is proved as follows: Suppose  $\hat{a} = \hat{b}$  but  $a \neq b$ . WLOG  $a \not\leq b$ . Consider the filter

$$F_a = \{d \in D \mid a \leq d\}.$$

Then  $a \in F_a$  but  $b \notin F_a$ . By the prime filter theorem, there is some prime filter  $p \in B_D$  such that  $a \in p$  but  $b \notin p$ . But this contradicts  $\hat{a} = \hat{b}$ .

Furthermore,

$$\diamond \hat{a} = m_{Q_\diamond} \hat{a}$$

follows immediately from the observation that for any prime filter  $p \in B_D$  and any  $a \in D$ :

$$pQ_\diamond \hat{a} \quad \text{iff} \quad \diamond a \in p.$$



In order to prove, first assume that  $pQ_\diamond \hat{a}$ . Since  $a \in F_{\hat{a}}$  it follows that  $\diamond a \in p$  by definition of  $Q_\diamond$ . For the other direction, suppose that  $\diamond a \in p$  and let  $b$  be an arbitrary element of  $F_{\hat{a}}$ . By definition of  $F_{\hat{a}}$  this means that  $\hat{a} \subseteq \hat{b}$ , so since the map is an embedding of  $\mathcal{D}$  into  $(\mathcal{P}(B_D), \cup, \cap)$ , we obtain that  $a \leq b$ . Monotonicity of  $\diamond$  gives that  $\diamond a \leq \diamond b$ , so we find  $\diamond b \in p$  since  $p$  is a prime filter. Because  $b$  was arbitrary this means that  $pQ_\diamond \hat{a}$ , as required.

**Lemma:** Let  $A \subseteq D$  be a downward directed set in the monotone lattice expansion  $\mathcal{D} = (D, \vee, \wedge, \diamond)$ , and let  $p$  be some prime filter of  $D$ . Then

$$pQ_\diamond \bigcap_{a \in A} \hat{a} \quad \text{iff} \quad \diamond a \in p \text{ for all } a \in A.$$

**Proof:** Since the direction from left to right follows immediately from the monotonicity of  $Q_\diamond$ , we concentrate on the other direction.

Assume that  $\diamond a \in p$  for all  $a \in A$ , and suppose for contradiction that

$$pQ_\diamond \bigcap_{a \in A} \hat{a}$$

does not hold. Then by definition of  $Q_\diamond$  and the fact that  $\bigcap_{a \in A} \hat{a}$  is closed, there is a  $b \in D$  such that  $\diamond b \notin p$  and

$$\bigcap_{a \in A} \hat{a} \subseteq \hat{b}.$$

We claim that there are finitely many elements  $a_0, \dots, a_n \in A$  satisfying

$$a_0 \wedge \dots \wedge a_n \leq b.$$

To see why this must be the case, consider the filter  $F$  generated by  $A$ ; that is,  $F$  is the set of elements  $d \in D$  for which there are  $a_0, \dots, a_n \in A$  such that

$$a_0 \wedge \dots \wedge a_n \leq d.$$

If  $b$  would not belong to  $F$  then by the Prime Filter Theorem there would be a prime filter  $q$  with  $F \subseteq q$  and  $b \notin q$ . This  $q$  would then be such that

$$q \in \bigcap_{a \in A} \hat{a} \quad \text{while} \quad q \notin \hat{b},$$

which clearly cannot be the case. Hence,  $b$  does belong to  $F$  which proves our claim. Since  $A$  is downward directed, there is an element  $a \in A$  such that

$$a \leq a_0 \wedge \dots \wedge a_n.$$

But then we also have that  $a \leq b$ , so by monotonicity of  $\diamond$  and the fact that  $\diamond b \notin p$  we find that  $\diamond a \notin p$ , which contradicts our assumption that  $\diamond a \in p$  for all  $a \in A$ .

## 5.4 The Pieces: up close

In this section we will show how we can represent a game algebra  $\mathcal{G}$  as a monotone outcome algebra once we know that there is some separable module over  $\mathcal{G}$ .

The basic idea is as follows. Assume that  $\mathcal{M} = (M, \vee, \wedge, -, \diamond_g)_{g \in G}$  is a module over the game algebra  $\mathcal{G}$ . By Definition 3.5, with every operation  $\diamond_g$  of  $\mathcal{M}$  we may associate a monotone outcome relation  $Q_g$  on  $B_{\mathcal{M}}$  (for brevity, we will write  $Q_g$  rather than  $Q_{\diamond_g}$ ).

The representation map embedding the game algebra  $\mathcal{G}$  into the full monotone outcome algebra over  $B_{\mathcal{M}}$  will map an element  $g$  of  $G$  to the pair  $(Q_g, Q_{-g})$  of outcome relations on  $B_{\mathcal{M}}$ . The injectivity of this map will follow from the separability of the module; in order to prove that it is a homomorphism we need the following proposition which is one of the main technical results of the paper.

**Proposition 1:** Let  $\mathcal{M} = (M, \vee, \wedge, -, \diamond_g)_{g \in G}$  be a module over the game algebra  $\mathcal{G} = (G, \vee, \wedge, -, \diamond)$  and let  $g$  and  $h$  be arbitrary elements of  $G$ . Then we have

1.  $Q_{g \vee h} = Q_g \cup Q_h$ ,
2.  $Q_{g \wedge h} = Q_g \cap Q_h$ ,
3.  $Q_{g \diamond h} = Q_g \circ Q_h$ ,
4. if  $\diamond_g a \neq \diamond_h a$  for some  $a \in M$  then  $Q_g \neq Q_h$ .

Before sketching the proof, let us see how this solves the representation problem.

**Proposition 2:** Let  $\mathcal{M} = (M, \vee, \wedge, -, \diamond_g)_{g \in G}$  be a separable module over the game algebra  $\mathcal{G} = (G, \vee, \wedge, -, \diamond)$ . Then the map

$$\text{rep} : G \rightarrow G(B_{\mathcal{M}})$$

given by

$$\text{rep}(g) = (Q_g, Q_{-g})$$

is an embedding of  $\mathcal{G}$  into  $G_m(B_{\mathcal{M}})$ .

**Proof:** That it is a homomorphism follows from 1,2,3 of the last proposition. It is an embedding follows from separability and 4 of the last proposition.

Now let us sketch the idea of the proof of 1. Let  $p$  be an arbitrary prime filter and  $T$  an arbitrary set.

To prove 1, Assume  $T$  is of the form  $\hat{a}$ . Then  $PQ_g \hat{a} \in T$  follows from  $\diamond$  and prime filter properties, together with the game module axioms.

The general case: The proof for the general case uses the above, filter properties, and game module axioms.

Proof of 2 is done along similar lines.

In 3 proving  $pQ_gT$  or  $pQ_hTp \implies pQ_{g\Diamond h}T$  is routine when using the definition of outcome relations and game module axioms.

For the other direction, suppose that  $pQ_g\Diamond hT$ . Define  $U = \bigcap \{\hat{b} \mid b \in F_T\}$ . From this definition it is immediate that for every element  $u \in U$  and each  $b \in F_T$ , it holds that  $\Diamond_h b \in u$ . Thus, from this definition it follows that  $uQ_hT$  for every  $u \in U$ . Now observe that  $A = \{\Diamond_h b \mid b \in F_T\}$  is a downward directed subset of  $M$ . Hence, by lemma, if we prove  $\Diamond_g a \in p$  for every  $a \in A$ , we get  $pQ_gU$ . Finally, this gives us  $p(Q_g \circ Q_h)T$ .

Proof of 4 follows from Prime Filter Theorem immediately.

**Proposition 3:** Let  $\mathcal{G} = (G, \vee, \wedge, -, \Diamond)$  be a game algebra. Then  $\mathcal{G}$ , seen as a game module over itself, can be embedded in a separable game module  $\mathcal{G}'$  over  $\mathcal{G}$ .

**Proof:** In this proof we will fix a game algebra  $\mathcal{G} = (G, \vee, \wedge, -, \Diamond)$ . Recall that the module perspective on  $\mathcal{G}$  means that we identify  $\mathcal{G}$  with the structure  $(G, \vee, \wedge, -, \Diamond_g)_{g \in G}$ . We will show that  $\mathcal{G}$  can be embedded in a  $\mathcal{G}$ -module  $(G', \vee', \wedge', -, \Diamond'_g)_{g \in G}$ . To do so, we will first concentrate on extending the de Morgan reduct  $(G, \vee, \wedge, -)$  of  $\mathcal{G}$ , and then show how to extend the operations  $\Diamond'_g$  to the extended de Morgan lattice. The basic intuition underlying our approach is that we aim at adding a single separating element to  $M$ ; that is, an object  $s$  that will satisfy  $\Diamond_g s = g$  for every  $g \in G$ .

The first part of the construction can be applied to arbitrary de Morgan lattices; fix such an algebra  $\mathcal{M} = (M, \vee, \wedge, -)$ . In a number of steps we will define an extension  $\mathcal{M}'$  of  $\mathcal{M}$  which satisfies certain nice properties.

The most hassle-free way to add new elements is by introducing top/bottom elements, as we know exactly how to define the operations involving them. Infact, these elements will serve more purpose as we will see soon.

Let  $D$  be any distributive lattice. Define  $D^b = D \cup \{\top, \perp\}$ , where  $\top, \perp$  are new elements. The top and bottom added are treated to be strictly new elements here.

Given any de Morgan lattice  $\mathcal{M} = (M, \vee, \wedge, -)$ , let  $M''$  be the set  $(M \cup \{\top, \perp\})^2$ , and let  $\vee''$  and  $\wedge''$  be the coordinatewise join and meet operations on  $M''$ . In other words, the algebra  $(M'', \vee'', \wedge'')$  is the distributive lattice product  $(M, \vee, \wedge)^b \times (M, \vee, \wedge)^b$ .

The operation

$$-': M'' \rightarrow M''$$

is defined by  $-'(x, y) = (-y, -x)$ . and we define  $\mathcal{M}''$  as the structure  $(M'', \vee'', \wedge'', -')$ . In general,  $\mathcal{M}''$  is not isomorphic to  $\mathcal{M}^b \times \mathcal{M}^b$ .

The diagonal map  $\Delta : M \longrightarrow M''$  is an embedding.

Lest we tend to lose sight, let us remind ourselves that we seek to embed a game alge-

bra  $\mathcal{G}$  seen as a game module over itself, inside a separable game module  $\mathcal{G}'$  over  $\mathcal{G}$ . The above construction will take care of it at the De-Morgan level. The pair  $(\top, \perp)$  will be the 'esteemed' separating element. For this purpose,  $\mathcal{M}''$  seems too big for the purpose of embedding  $\mathcal{M}$  inside it and also being separable. The subalgebra generated by  $\Delta(M)$  and  $(\top, \perp)$  does the job in the most minimal sense.

So we take  $\mathcal{M}'$ , our target algebra. More precisely,

$$\mathcal{M}' = (M \cup \{\top\} \times (M \cup \{\perp\}) \cup \{(\top, \top), (\perp, \perp)\}).$$

By the above discussion,  $\Delta : M \longrightarrow M'$  is an embedding.

Finally, we focus on the game module set-up. Let  $(G', \vee', \wedge', -)$  be defined as above. It is left to define an operation  $\diamond'_g : G' \longrightarrow G'$  for every element  $g \in G'$ , to turn it into a game module over  $G$ . We will motivate the construction before revealing it in full glory:

It is natural to want the diagonal map  $\Delta : G \rightarrow G'$  to be an embedding in the game module level as well, after already having it in the de Morgan algebra level.

Meaning we want the following to happen:  $\Delta(\diamond'_g a) = \diamond'_g(\Delta(a))$

$$\Delta(\diamond'_g a) = \diamond'_g(a, a).$$

$$\diamond'_g(a, a) = (\diamond'_g a, \diamond'_g a).$$

This gives us the hint about what to do with the pairs  $(a, b)$  where  $a, b \in G$ . Send  $\diamond'_g(a, b) \mapsto (\diamond_g a, \diamond_g b)$ .

Next we want the element  $(\top, \perp)$  to be the separating element at the  $g'$ -level. Put  $\Delta'_g(\top, \perp) = (g, g)$  for every  $g \in G$ . We are still left with elements of the form

$$(\top, a) \quad \text{with } a \neq \perp \quad \text{and} \quad (a, \perp) \quad \text{with } a \neq \top.$$

The elements bigger than  $(\top, a)$  are of the form  $(\top, -)$ . So sending  $(\top, a)$  to  $(\top, \top)$  will preserve monotonicity from above and below. Similarly, elements smaller than  $(a, \perp)$  are of the form  $(-, \perp)$ . So sending  $(a, \perp)$  to  $(\perp, \perp)$  preserves monotonicity again.

The above highlights another utility of adding  $\top$  and  $\perp$ . Adding anything else would not tell us easily (if at all possibly) where exactly to send such elements. Top/bottom makes life easy due to its canonical relationship with other elements. We know exactly what to do with them.

Formally, we define, for an arbitrary element  $g \in G$ , the map

$$\diamond'_g : G' \rightarrow G'$$

by putting

$$\diamond'_g(a, b) = \begin{cases} (\diamond_g a, \diamond_g b) & \text{if } a, b \in G, \\ (g, g) & \text{if } a = \top \text{ and } b = \perp, \\ (\top, \top) & \text{if } a = \top \text{ and } b \neq \perp, \\ (\perp, \perp) & \text{if } a \neq \top \text{ and } b = \perp. \end{cases}$$

Given a game algebra  $\mathcal{G}$ , let  $\mathcal{G}'$  be the module

$$(G', \vee', \wedge', -', \diamond'_g)_{g \in G}.$$

**Claim:** If  $\mathcal{G}$  is a game algebra, then  $\mathcal{G}'$  is a separable game module.

**Proof of Claim.** We have already seen that the structure  $(G', \vee', \wedge', -')$  is a de Morgan algebra, and it is easy to see that the operations  $\diamond'_g$  are all monotone. The conditions M1–4 can be checked via straightforward case distinctions.

Note that separability of  $\mathcal{G}'$  is immediate by the definition: if  $g$  and  $h$  are distinct elements of  $G$  then

$$\diamond'_g(\top, \perp) = (g, g) \neq (h, h) = \diamond'_h(\top, \perp).$$

Since the diagonal map

$$\Delta : a \rightarrow (a, a)$$

is obviously an embedding of the  $G$ -module  $G$  in the  $G$ -module  $\mathcal{G}'$ , this proves the proposition.

Next we prove that every monotone outcome algebra is isomorphic to a board algebra. This means that the consistency requirement does not give any extra valid equations.

**Proposition 4:** Let  $A = (A, \sqcup, \sqcap, -, \square)$  be a monotone outcome algebra over the set  $B$ . Then  $A$  is isomorphic to a board algebra over the set  $B' = B \cup \{\infty\}$  (where  $\infty \notin B$ ).

**Proof:** Suppose that  $A = (A, \sqcup, \sqcap, -, \square)$  is a monotone outcome algebra over the set  $B$ , and let  $\infty$  be an object not in  $B$ . Given a monotone outcome relation  $R$  over  $B$ , define  $R'$  as the following outcome relation over the set  $B' = B \cup \{\infty\}$ :

$$R' = \{(\infty, T) \mid \infty \in T\} \cup \{(p, T) \mid p \in B, \infty \in T \text{ and } (p, T^-) \in R\},$$

where  $T^-$  denotes the set  $T \setminus \{\infty\}$ .

It is obvious that any pair of relations  $(R', S')$  is consistent since for any  $p \in B'$  we have  $(p, T) \in R'$  only if  $\infty \in T$  and likewise for  $S'$ . Thus  $(p, T) \in R'$  implies that  $(p, B' \setminus T) \notin S'$ . We claim that the function mapping a pair of outcome relations  $(R, S)$  to the pair  $(R', S')$  is an embedding of  $A$  in the board algebra over  $B$ . This follows immediately from the observation that the operation  $(\cdot)'$  distributes over unions, intersections and compositions of relations. Proof is easy.

## 6 Concluding Remarks

Venema's representation theorem shows that every game algebra is isomorphic to a board algebra, neatly resolving the axiomatization problem. Goranko's syntactic path and Venema's algebraic one may look different on the surface, but they ultimately converge on the same destination: a robust account of game equivalence. All roads lead to Rome!