LOGIC OF GAMES

Andreas Blass

University of Michigan
Ann Arbor, MI 48109

ablass@umich.edu

Games in Logic

Games in Logic

Two main sorts of logical games.

Games in Logic

Two main sorts of logical games.

(1) Truth of formulas in a structure is expressible by games.

$\exists$ and $\vee$ become choices for proponent (P).

$\forall$ and $\wedge$ become choices for opponent (O).

Winner depends on atomic and negated atomic formulas.

Games in Logic

Two main sorts of logical games.

(1) Truth of formulas in a structure is expressible by games.

$\exists$ and $\vee$ become choices for proponent (P).

$\forall$ and $\wedge$ become choices for opponent (O).

Winner depends on atomic and negated atomic formulas.

(2) Provability is expressible by games.

P exhibits a rule with the formula under consideration as its conclusion.

O chooses a premise of that rule, which becomes the new formula under consideration.

Whoever can't move loses.

O wins infinite plays.

Games in Logic

Two main sorts of logical games.

(1) Truth of formulas in a structure is expressible by games.

$\exists$ and $\vee$ become choices for proponent (P).

$\forall$ and $\wedge$ become choices for opponent (O).

Winner depends on atomic and negated atomic formulas.

(2) Provability is expressible by games.

P exhibits a rule with the formula under consideration as its conclusion.

O chooses a premise of that rule, which becomes the new formula under consideration.

Whoever can't move loses.

O wins infinite plays.

This talk will be almost entirely about (1).

Games in Logic

Two main sorts of logical games.

(1) Truth of formulas in a structure is expressible by games.

$\exists$ and $\vee$ become choices for proponent (P).

$\forall$ and $\wedge$ become choices for opponent (O).

Winner depends on atomic and negated atomic formulas.

(2) Provability is expressible by games.

P exhibits a rule with the formula under consideration as its conclusion.

O chooses a premise of that rule, which becomes the new formula under consideration.

Whoever can't move loses.

O wins infinite plays.

This talk will be almost entirely about (1).
Semantics rather than deduction.

Games in Logic

Two main sorts of logical games.

(1) Truth of formulas in a structure is expressible by games.

$\exists$ and $\vee$ become choices for proponent (P).

$\forall$ and $\wedge$ become choices for opponent (O).

Winner depends on atomic and negated atomic formulas.

(2) Provability is expressible by games.

P exhibits a rule with the formula under consideration as its conclusion.

O chooses a premise of that rule, which becomes the new formula under consideration.

Whoever can't move loses.

O wins infinite plays.

This talk will be almost entirely about (1).
Semantics rather than deduction.
Games will be 2-player, win-lose games of perfect information.

The Curse of Determinacy

The Curse of Determinacy

If all plays of a game are finite, then the game is determined.

The Curse of Determinacy

If all plays of a game are finite, then the game is determined.

The logic of such games is just classical logic. For example, $A \vee \neg A$ is valid.

The Curse of Determinacy

If all plays of a game are finite, then the game is determined.

The logic of such games is just classical logic.

For example, $A \vee \neg A$ is valid.

How can one get non-classical logics of games?

The Curse of Determinacy

If all plays of a game are finite, then the game is determined.

The logic of such games is just classical logic.

For example, $A \vee \neg A$ is valid.

How can one get non-classical logics of games?

- Allow plays of infinite length. (Gale-Stewart, Martin)

The Curse of Determinacy

If all plays of a game are finite, then the game is determined.

The logic of such games is just classical logic.

For example, $A \vee \neg A$ is valid.

How can one get non-classical logics of games?

- Allow plays of infinite length. (Gale-Stewart, Martin)
- Require winning strategies to be computable. (Rabin, Japaridze)

The Curse of Determinacy

If all plays of a game are finite, then the game is determined.

The logic of such games is just classical logic.

For example, $A \vee \neg A$ is valid.

How can one get non-classical logics of games?

- Allow plays of infinite length. (Gale-Stewart, Martin)
- Require winning strategies to be computable. (Rabin, Japaridze)
- Require winning strategies to be history-free. (Abramsky, Jagadeesan)

The Curse of Determinacy

If all plays of a game are finite, then the game is determined.

The logic of such games is just classical logic.

For example, $A \vee \neg A$ is valid.

How can one get non-classical logics of games?

- Allow plays of infinite length. (Gale-Stewart, Martin)
- Require winning strategies to be computable. (Rabin, Japaridze)
- Require winning strategies to be history-free. (Abramsky, Jagadeesan)
- Require winning strategies to be uniform under addition of new options to games. (Abramsky, Jagadeesan)

The Curse of Determinacy

If all plays of a game are finite, then the game is determined.

The logic of such games is just classical logic.

For example, $A \vee \neg A$ is valid.

How can one get non-classical logics of games?

- Allow plays of infinite length. (Gale-Stewart, Martin)
- Require winning strategies to be computable. (Rabin, Japaridze)
- Require winning strategies to be history-free. (Abramsky, Jagadeesan)
- Require winning strategies to be uniform under addition of new options to games. (Abramsky, Jagadeesan)
- Allow different rules depending on who moves first. (Abramsky, Jagadeesan)

Complexity of Strategies

Complexity of Strategies

A **really playable** game is one where

Complexity of Strategies

A **really playable** game is one where

- each move is a finite object (e.g., natural number),

Complexity of Strategies

A **really playable** game is one where

- each move is a finite object (e.g., natural number),
- there is an algorithm deciding, for every position

Complexity of Strategies

A **really playable** game is one where

- each move is a finite object (e.g., natural number),
- there is an algorithm deciding, for every position
  - whether the play is ended,

Complexity of Strategies

A **really playable** game is one where

- each move is a finite object (e.g., natural number),
- there is an algorithm deciding, for every position
  - whether the play is ended,
  - if so, who won,

Complexity of Strategies

A **really playable** game is one where

- each move is a finite object (e.g., natural number),
- there is an algorithm deciding, for every position
  - whether the play is ended,
  - if so, who won,
  - if not, who is to move next,

Complexity of Strategies

A **really playable** game is one where

- each move is a finite object (e.g., natural number),
- there is an algorithm deciding, for every position
  - whether the play is ended,
  - if so, who won,
  - if not, who is to move next,
  - and whether any proposed move is legal,

Complexity of Strategies

A **really playable** game is one where

- each move is a finite object (e.g., natural number),
- there is an algorithm deciding, for every position
  - whether the play is ended,
  - if so, who won,
  - if not, who is to move next,
  - and whether any proposed move is legal,
- and each play ends after finitely many moves.

Complexity of Strategies

A **really playable** game is one where

- each move is a finite object (e.g., natural number),
- there is an algorithm deciding, for every position
  - whether the play is ended,
  - if so, who won,
  - if not, who is to move next,
  - and whether any proposed move is legal,
- and each play ends after finitely many moves.

Rabin showed that, although every such game has a winning strategy for one of the players, there need not be a computable winning strategy.

Complexity of Strategies

A **really playable** game is one where

- each move is a finite object (e.g., natural number),
- there is an algorithm deciding, for every position
  - whether the play is ended,
  - if so, who won,
  - if not, who is to move next,
  - and whether any proposed move is legal,
- and each play ends after finitely many moves.

Rabin showed that, although every such game has a winning strategy for one of the players, there need not be a computable winning strategy.

In fact, for each hyperarithmetical set A, there is a really playable game such that A is computable from each winning strategy.

Analogies

Analogies

- Classical Logic
- Intuitionistic Logic
- Game Semantics

Analogies

- Classical Logic
- Intuitionistic Logic
- Game Semantics

- Truth
- Provability
- Winning Strategy

Analogies

- Classical Logic
- Intuitionistic Logic
- Game Semantics

- Truth
- Provability
- Winning Strategy

- Deterministic algorithm
- Non-deterministic algorithm
- Alternating algorithm

Analogies

- Classical Logic
- Intuitionistic Logic
- Game Semantics

- Truth
- Provability
- Winning Strategy

- Deterministic algorithm
- Non-deterministic algorithm
- Alternating algorithm

- Excluded Middle
- Kripke Schema
- "Lorenzen Schema"

Analogies

- Classical Logic
- Intuitionistic Logic
- Game Semantics

- Truth
- Provability
- Winning Strategy

- Deterministic algorithm
- Non-deterministic algorithm
- Alternating algorithm

- Excluded Middle
- Kripke Schema
- "Lorenzen Schema"

Lorenzen Schema: For each formula A, there is a really playable game (as in Rabin's theorem) such that A holds iff P has a winning strategy in that game.

Other Operations on Games

Other Operations on Games

"Multiplicative" operations first arose from trying to understand reducibility.

Other Operations on Games

"Multiplicative" operations first arose from trying to understand reducibility.

"If you show me how to win G (as P), then I can win H (also as P)."

Other Operations on Games

"Multiplicative" operations first arose from trying to understand reducibility.

"If you show me how to win G (as P), then I can win H (also as P)."

This led to $\otimes$ and its dual.

Other Operations on Games

"Multiplicative" operations first arose from trying to understand reducibility.

"If you show me how to win G (as P), then I can win H (also as P)."

This led to $\otimes$ and its dual.

Also a version of exponential modality

Other Operations on Games

"Multiplicative" operations first arose from trying to understand reducibility.

"If you show me how to win G (as P), then I can win H (also as P)."

This led to $\otimes$ and its dual.

Also a version of exponential modality

But linear **logic** came later.

Semantics vs. Syntax

Semantics vs. Syntax

Original game semantics agreed with affine logic on additive sequents.

Semantics vs. Syntax

Original game semantics agreed with affine logic on additive sequents, but validated more multiplicative sequents.

Semantics vs. Syntax

Original game semantics agreed with affine logic on additive sequents, but validated more multiplicative sequents.

Multiplicative fragment gave all instances of binary tautologies.

Semantics vs. Syntax

Original game semantics agreed with affine logic on additive sequents, but validated more multiplicative sequents.

Multiplicative fragment gave all instances of binary tautologies.

Abramsky and Jagadeesan modified the semantics to get exactly multiplicative linear logic plus the Mix rule.

$$\frac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma, \Delta} \qquad\qquad \frac{}{\vdash}$$

Semantics vs. Syntax

Original game semantics agreed with affine logic on additive sequents, but validated more multiplicative sequents.

Multiplicative fragment gave all instances of binary tautologies.

Abramsky and Jagadeesan modified the semantics to get exactly multiplicative linear logic plus the Mix rule.

$$\frac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma, \Delta} \qquad\qquad \frac{}{\vdash}$$

Hyland and Ong modified it further to get exactly multiplicative linear logic.

Semantics vs. Syntax

Original game semantics agreed with affine logic on additive sequents, but validated more multiplicative sequents.

Multiplicative fragment gave all instances of binary tautologies.

Abramsky and Jagadeesan modified the semantics to get exactly multiplicative linear logic plus the Mix rule.

$$\frac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma, \Delta} \qquad\qquad \frac{}{\vdash}$$

Hyland and Ong modified it further to get exactly multiplicative linear logic.

But the additive fragment no longer works well.

Semantics vs. Syntax

Original game semantics agreed with affine logic on additive sequents, but validated more multiplicative sequents.

Multiplicative fragment gave all instances of binary tautologies.

Abramsky and Jagadeesan modified the semantics to get exactly multiplicative linear logic plus the Mix rule.

$$\frac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma, \Delta} \qquad\qquad \frac{}{\vdash}$$

Hyland and Ong modified it further to get exactly multiplicative linear logic.

But the additive fragment no longer works well.

Taking semantics as primary, we don't have good axiomatic systems for game-validity.

Semantics vs. Syntax

Original game semantics agreed with affine logic on additive sequents, but validated more multiplicative sequents.

Multiplicative fragment gave all instances of binary tautologies.

Abramsky and Jagadeesan modified the semantics to get exactly multiplicative linear logic plus the Mix rule.

$$\frac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma, \Delta} \qquad\qquad \frac{}{\vdash}$$

Hyland and Ong modified it further to get exactly multiplicative linear logic.

But the additive fragment no longer works well.

Taking semantics as primary, we don't have good axiomatic systems for game-validity.

Japaridze has deductive systems for various fragments of computability logic.

Semantics vs. Syntax

Original game semantics agreed with affine logic on additive sequents, but validated more multiplicative sequents.

Multiplicative fragment gave all instances of binary tautologies.

Abramsky and Jagadeesan modified the semantics to get exactly multiplicative linear logic plus the Mix rule.

$$\frac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma, \Delta} \qquad\qquad \frac{}{\vdash}$$

Hyland and Ong modified it further to get exactly multiplicative linear logic.

But the additive fragment no longer works well.

Taking semantics as primary, we don't have good axiomatic systems for game-validity.

Japaridze has deductive systems for various fragments of computability logic.

But the flavor is still game-like more than logical.

Categories

Categories

The initial (i.e., free) category with products and coproducts has objects that look like finite-length games.

Categories

The initial (i.e., free) category with products and coproducts has objects that look like finite-length games.

Morphisms $1 \to G$ are winning strategies for P in G.

Categories

The initial (i.e., free) category with products and coproducts has objects that look like finite-length games.

Morphisms $1 \to G$ are winning strategies for P in G.

Morphisms $G \to 0$ are winning strategies for O in G.

Categories

The initial (i.e., free) category with products and coproducts has objects that look like finite-length games.

Morphisms $1 \to G$ are winning strategies for P in G.

Morphisms $G \to 0$ are winning strategies for O in G.

Morphisms $G \to H$ are like reductions, but with identifications to obtain associativity.

Categories

The initial (i.e., free) category with products and coproducts has objects that look like finite-length games.

Morphisms $1 \to G$ are winning strategies for P in G.

Morphisms $G \to 0$ are winning strategies for O in G.

Morphisms $G \to H$ are like reductions, but with identifications to obtain associativity.

This resembles an important idea of Japaridze: Don't require players to move in a particular order.

Categories

The initial (i.e., free) category with products and coproducts has objects that look like finite-length games.

Morphisms $1 \to G$ are winning strategies for P in G.

Morphisms $G \to 0$ are winning strategies for O in G.

Morphisms $G \to H$ are like reductions, but with identifications to obtain associativity.

This resembles an important idea of Japaridze:
Don't require players to move in a particular order.
But speed doesn't count.

Categories

The initial (i.e., free) category with products and coproducts has objects that look like finite-length games.

Morphisms $1 \to G$ are winning strategies for P in G.

Morphisms $G \to 0$ are winning strategies for O in G.

Morphisms $G \to H$ are like reductions, but with identifications to obtain associativity.

This resembles an important idea of Japaridze:
Don't require players to move in a particular order.
But speed doesn't count.
"Static games"

Categories

The initial (i.e., free) category with products and coproducts has objects that look like finite-length games.

Morphisms $1 \to G$ are winning strategies for P in G.

Morphisms $G \to 0$ are winning strategies for O in G.

Morphisms $G \to H$ are like reductions, but with identifications to obtain associativity.

This resembles an important idea of Japaridze: Don't require players to move in a particular order.

But speed doesn't count.

"Static games"

The precise connection has not yet been worked out.

Different Exponentials

Different Exponentials

Girard noted that linear logic's proof rules for ! don't determine it.

Different Exponentials

Girard noted that linear logic's proof rules for ! don't determine it.

Game semantics has (at least) two natural versions of $!G$.

Different Exponentials

Girard noted that linear logic's proof rules for ! don't determine it.

Game semantics has (at least) two natural versions of $!G$.

Both have many copies of G, and P wins if he wins all copies that are completed.

Different Exponentials

Girard noted that linear logic's proof rules for ! don't determine it.

Game semantics has (at least) two natural versions of $!G$.

Both have many copies of G, and P wins if he wins all copies that are completed.

In one version, P must play the same moves in any two copies as long as O does.

Different Exponentials

Girard noted that linear logic's proof rules for ! don't determine it.

Game semantics has (at least) two natural versions of $!G$.

Both have many copies of G, and P wins if he wins all copies that are completed.

In one version, P must play the same moves in any two copies as long as O does.

In the other, the copies are independent.

Different Exponentials

Girard noted that linear logic's proof rules for ! don't determine it.

Game semantics has (at least) two natural versions of $!G$.

Both have many copies of G, and P wins if he wins all copies that are completed.

In one version, P must play the same moves in any two copies as long as O does.

In the other, the copies are independent.

First version represents a single, re-usable resource.

Second represents a stream of resources of the same type.

Different Exponentials

Girard noted that linear logic's proof rules for ! don't determine it.

Game semantics has (at least) two natural versions of $!G$.

Both have many copies of G, and P wins if he wins all copies that are completed.

In one version, P must play the same moves in any two copies as long as O does.

In the other, the copies are independent.

First version represents a single, re-usable resource.

Second represents a stream of resources of the same type.

Japaridze's examples:

Different Exponentials

Girard noted that linear logic's proof rules for ! don't determine it.

Game semantics has (at least) two natural versions of $!G$.

Both have many copies of G, and P wins if he wins all copies that are completed.

In one version, P must play the same moves in any two copies as long as O does.

In the other, the copies are independent.

First version represents a single, re-usable resource.

Second represents a stream of resources of the same type.

Japaridze's examples:

$$!(A \oplus B) \vdash (!A) \oplus (!B)$$

is valid only in the first version.

Different Exponentials

Girard noted that linear logic's proof rules for ! don't determine it.

Game semantics has (at least) two natural versions of $!G$.

Both have many copies of G, and P wins if he wins all copies that are completed.

In one version, P must play the same moves in any two copies as long as O does.

In the other, the copies are independent.

First version represents a single, re-usable resource.

Second represents a stream of resources of the same type.

Japaridze's examples:

$$!(A \oplus B) \vdash (!A) \oplus (!B)$$

is valid only in the first version, and

$$A \otimes !(A \to (A \otimes B)) \vdash !B$$

only in the second.