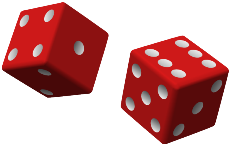
Sequential Games and Optimal Strategies

Paulo Oliva

(based on joint work with M. Escardó)

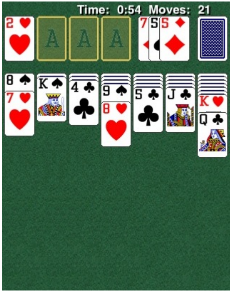
Queen Mary, University of London, UK

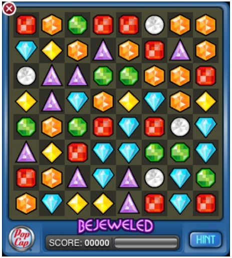


Logic Colloquium

Paris, 25 July 2010

Single-player Games





Two-player Games

Two **players**: Black and White

Two-player Games

Two **players**: Black and White

Possible **outcomes**:

- Black wins
- White wins
- Draw

Two-player Games

Two **players**: Black and White

Possible **outcomes**:

- Black wins
- White wins
- Draw

Strategy: Choice of move at round k given previous moves

Another Game

Two **players**: John and Julia

Another Game

Two **players**: John and Julia

John splits a cake. Julia chooses one of the two pieces

Another Game

Two **players**: John and Julia

John splits a cake. Julia chooses one of the two pieces

Possible **outcomes**:

- John gets $N\%$ of the cake (John's payoff)
- Julia gets $(100 - N)\%$ of the cake (Julia's payoff)

THE ROYAL SOCIETY

Another Game

Two **players**: John and Julia

John splits a cake. Julia chooses one of the two pieces

Possible **outcomes**:

- John gets $N\%$ of the cake (John's payoff)
- Julia gets $(100 - N)\%$ of the cake (Julia's payoff)

Best strategy for John is to split cake into half

It is not a "winning strategy" but it is an **optimal strategy**

It maximises his payoff

Number of Player vs Number of Rounds

Number of players is not essential

It is important what the “goal” at each round is

Rounds with “**same goal**” mean played by “**same player**”

Number of Player vs Number of Rounds

Number of players is not essential

It is important what the "goal" at each round is

Rounds with "**same goal**" mean played by "**same player**"

How to describe the goal at a particular round?

Number of Player vs Number of Rounds

Number of players is not essential

It is important what the "goal" at each round is

Rounds with "**same goal**" mean played by "**same player**"

How to describe the goal at a particular round?

You could say: The goal is to win!

But maybe this is not possible (or might not even make sense)

Instead, the goal should be described as

a choice of outcome from each set of possible outcomes

As in...

Q: How much would you like to play for your flight?

As in...

Q: How much would you like to play for your flight?

A: As little as possible!

Target function

If $R =$ set of outcomes and $X =$ set of possible moves then

$$\phi \in (X \to R) \to R$$

describes the desired outcome $\phi p \in R$ given that the outcome of the game $px \in R$ for each move $x \in X$ is given.

Target function

If R = set of outcomes and X = set of possible moves then

$$\phi \in (X \to R) \to R$$

describes the desired outcome $\phi p \in R$ given that the outcome of the game $px \in R$ for each move $x \in X$ is given.

In the example:

$$\begin{array}{lcl} X & = & \textit{possible flights} \\ R & = & \textit{real number} \\ X \to R & = & \textit{price of each flight} \\ \phi & = & \textit{minimal value functional} \end{array}$$

Outline

Outline

Generalised quantifiers

$$\phi : (X \to R) \to R$$

Generalised quantifiers

$$\phi : (X \to R) \to R$$

For instance

Operation	ϕ	:	$(X \to R) \to R$
Quantifiers	$\forall_X, \exists_X$	:	$(X \to \mathbb{B}) \to \mathbb{B}$
Double negation	$\neg\neg X$	:	$(X \to \bot) \to \bot$
Integration	$\int_0^1$	:	$([0,1] \to \mathbb{R}) \to \mathbb{R}$
Supremum	$\sup_{[0,1]}$	:	$([0,1] \to \mathbb{R}) \to \mathbb{R}$
Limit	$\lim$	:	$(\mathbb{N} \to R) \to R$
Fixed point operator	fix_X	:	$(X \to X) \to X$

Generalised quantifiers

$$\phi : (X \to R) \to R \qquad (\equiv K_R X)$$

For instance

Operation	ϕ	:	$(X \to R) \to R$
Quantifiers	$\forall_X, \exists_X$	:	$(X \to \mathbb{B}) \to \mathbb{B}$
Double negation	$\neg\neg X$	:	$(X \to \bot) \to \bot$
Integration	$\int_0^1$	:	$([0,1] \to \mathbb{R}) \to \mathbb{R}$
Supremum	$\sup_{[0,1]}$	:	$([0,1] \to \mathbb{R}) \to \mathbb{R}$
Limit	$\lim$	:	$(\mathbb{N} \to R) \to R$
Fixed point operator	fix_X	:	$(X \to X) \to X$

Nested quantifiers $\equiv$ single quantifier on **product space**

Nested quantifiers $\equiv$ single quantifier on **product space**

$$\exists x^X \forall y^Y p(x, y)$$

Nested quantifiers $\equiv$ single quantifier on **product space**

$$\exists x^X \forall y^Y p(x,y) \quad \stackrel{\mathbb{B}}{=} \quad (\exists_X \otimes \forall_Y)(p^{X \times Y \to \mathbb{B}})$$

Nested quantifiers $\equiv$ single quantifier on **product space**

$$\exists x^X \forall y^Y p(x,y) \quad \stackrel{\mathbb{B}}{\equiv} \quad (\exists_X \otimes \forall_Y)(p^{X \times Y \to \mathbb{B}})$$

$$\sup_x \int_0^1 p(x,y)dy \quad \stackrel{\mathbb{R}}{\equiv} \quad (\sup \otimes \textstyle\int)(p^{[0,1]^2 \to \mathbb{R}})$$

Nested quantifiers $\equiv$ single quantifier on **product space**

$$\exists x^X \forall y^Y p(x,y) \quad \stackrel{\mathbb{B}}{\equiv} \quad (\exists_X \otimes \forall_Y)(p^{X \times Y \to \mathbb{B}})$$

$$\sup_x \int_0^1 p(x,y)dy \quad \stackrel{\mathbb{R}}{\equiv} \quad (\sup \otimes \textstyle\int)(p^{[0,1]^2 \to \mathbb{R}})$$

Definition (Product of Generalised Quantifiers)

Given $\phi \colon KX$ and $\psi \colon KY$ define $\phi \otimes \psi \ : \ K(X \times Y)$

$$(\phi \otimes \psi)(p) :\stackrel{R}{\equiv} \phi(\lambda x^X . \psi(\lambda y^Y . p(x,y)))$$

where $p \colon X \times Y \to R$.

Theorem (Mean Value Theorem)

For any $p \in C[0,1]$ *there is a point* $a \in [0,1]$ *such that*

$$\int_0^1 p = p(a)$$

Theorem (Mean Value Theorem)

For any $p \in C[0,1]$ *there is a point* $a \in [0,1]$ *such that*

$$\int_0^1 p = p(a)$$

Theorem (Maximum Value Theorem)

For any $p \in C[0,1]$ *there is a point* $a \in [0,1]$ *such that*

$$\sup p = p(a)$$

Theorem (Witness Theorem)

For any $p\colon X \to \mathbb{B}$ there is a point $a \in X$ such that

$$\exists x^X p(x) \quad \Leftrightarrow \quad p(a)$$

(similar to Hilbert's ε-term).

Theorem (Witness Theorem)

For any $p\colon X \to \mathbb{B}$ there is a point $a \in X$ such that

$$\exists x^X p(x) \quad \Leftrightarrow \quad p(a)$$

(similar to Hilbert's ε-term).

Theorem (Counter-example Theorem)

For any $p\colon X \to \mathbb{B}$ there is a point $a \in X$ such that

$$\forall x^X p(x) \quad \Leftrightarrow \quad p(a)$$

(a is counter-example to p if one exists).

Let $JX \equiv (X \to R) \to X$.

Let $JX \equiv (X \to R) \to X$.

Definition (Selection Functions)

$\varepsilon \colon JX$ is called a **selection function** for $\phi \colon KX$ if

$$\phi(p) = p(\varepsilon p)$$

holds for all $p \colon X \to R$.

Let $JX \equiv (X \to R) \to X$.

Definition (Selection Functions)

$\varepsilon\colon JX$ is called a **selection function** for $\phi\colon KX$ if

$$\phi(p) = p(\varepsilon p)$$

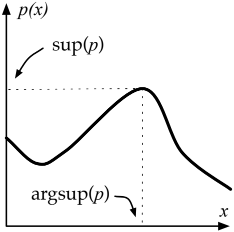
holds for all $p\colon X \to R$.

Definition (Attainable Quantifiers)

A generalised quantifier $\phi\colon KX$ is called **attainable** if it has a selection function $\varepsilon\colon JX$.

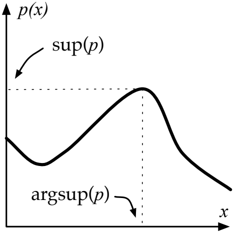
For Instance

- $\sup\colon K_{\mathbb{R}}[0,1]$ is an attainable quantifier as
$$\sup(p) = p(\operatorname{argsup}(p))$$
where $\operatorname{argsup}\colon J_{\mathbb{R}}[0,1]$.



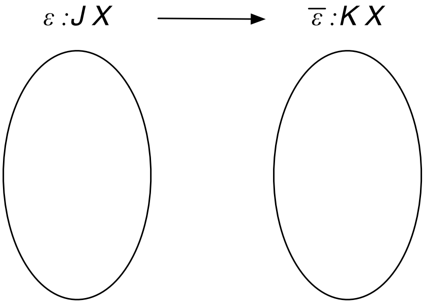
For Instance

- $\sup\colon K_{\mathbb{R}}[0,1]$ is an attainable quantifier as
$$\sup(p) = p(\operatorname{argsup}(p))$$
where $\operatorname{argsup}\colon J_{\mathbb{R}}[0,1]$.



- $\mathsf{fix}\colon K_X X$ is an attainable quantifier as
$$\mathsf{fix}(p) = p(\mathsf{fix}(p))$$
where $\mathsf{fix}\colon J_X X \; (= K_X X)$.

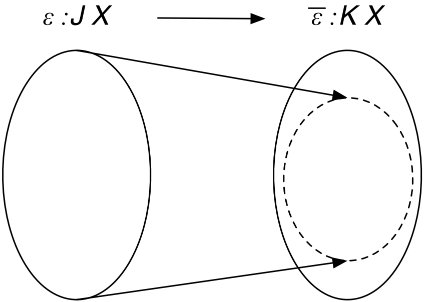
Selection Functions and Generalised Quantifiers



Every selection function $\varepsilon \colon JX$ defines a quantifier $\overline{\varepsilon} \colon KX$

$$\overline{\varepsilon}(p) \quad = \quad p(\varepsilon(p))$$

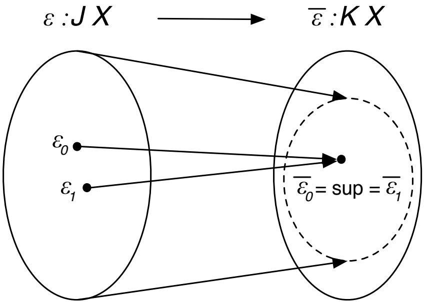
Selection Functions and Generalised Quantifiers



Not all quantifiers are attainable, e.g. $R = \{0, 1\}$

$$\phi(p) \quad = \quad 0$$

Selection Functions and Generalised Quantifiers



Different ε might define same ϕ, e.g. $X = [0,1]$ and $R = \mathbb{R}$

$$\begin{aligned}\varepsilon_0(p) &= \mu x . \sup p = p(x) \\ \varepsilon_1(p) &= \nu x . \sup p = p(x)\end{aligned}$$

Quantifier Elimination

Suppose

$$
\begin{aligned}
\exists x\, q(x) &= q(\varepsilon q) \\
\forall y\, q(y) &= q(\delta q).
\end{aligned}
$$

Quantifier Elimination

Suppose

$$\begin{aligned} \exists x\, q(x) &= q(\varepsilon q) \\ \forall y\, q(y) &= q(\delta q). \end{aligned}$$

Then

$$\exists x \forall y\, p(x, y) = \exists x\, p(x, b(x))$$

where

$$b(x) = \delta(\lambda y. p(x, y))$$

Quantifier Elimination

Suppose

$$\begin{aligned} \exists x\, q(x) &= q(\varepsilon q) \\ \forall y\, q(y) &= q(\delta q). \end{aligned}$$

Then

$$\begin{aligned} \exists x \forall y\, p(x,y) &= \exists x\, p(x, b(x)) \\ &= p(a, b(a)) \end{aligned}$$

where

$$\begin{aligned} b(x) &= \delta(\lambda y. p(x,y)) \\ a &= \varepsilon(\lambda x. p(x, b(x))). \end{aligned}$$

Quantifier Elimination

Suppose

$$\begin{aligned} \exists x\, q(x) &= q(\varepsilon q) \\ \forall y\, q(y) &= q(\delta q). \end{aligned}$$

Then

$$\begin{aligned} (\exists_X \otimes \forall_Y)(p) &= \exists x\, p(x, b(x)) \\ &= p(a, b(a)) \end{aligned}$$

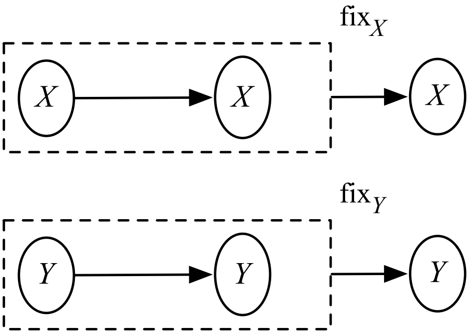
where

$$\begin{aligned} b(x) &= \delta(\lambda y.p(x,y)) \\ a &= \varepsilon(\lambda x.p(x, b(x))). \end{aligned}$$

Bekič's Lemma

Lemma

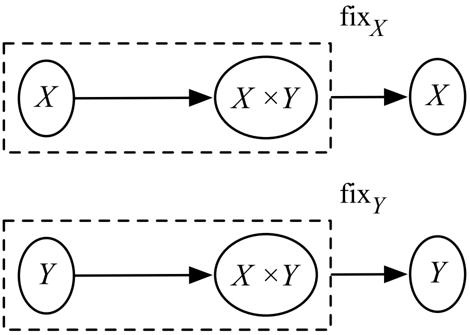
If X and Y have fixed point operators then so does $X \times Y$.



Bekič's Lemma

Lemma

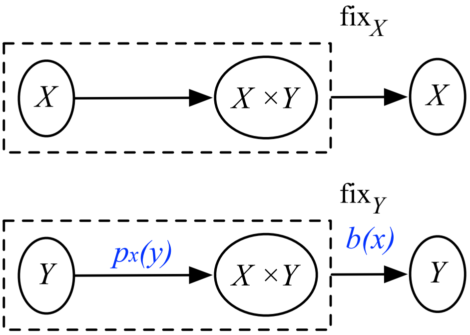
If X and Y have fixed point operators then so does $X \times Y$.



Bekič's Lemma

Lemma

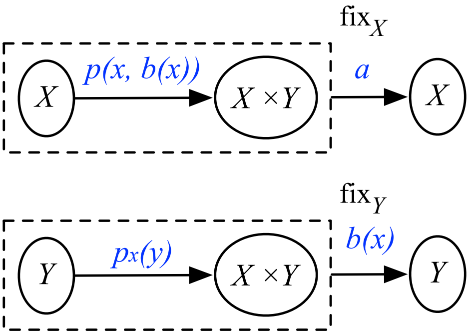
If X and Y have fixed point operators then so does $X \times Y$.



Bekič's Lemma

Lemma

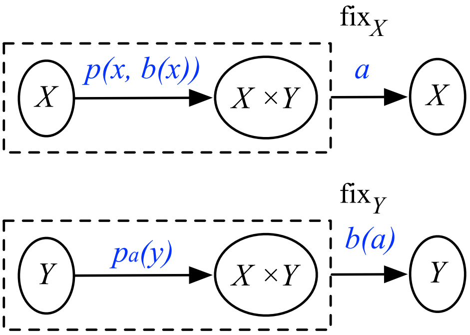
If X and Y have fixed point operators then so does $X \times Y$.



Bekič's Lemma

Lemma

If X and Y have fixed point operators then so does $X \times Y$.



Definition (Product of Selection Functions)

Given $\varepsilon \colon JX$ and $\delta \colon JY$ define $\varepsilon \otimes \delta \colon J(X \times Y)$ as

$$(\varepsilon \otimes \delta)(p^{X \times Y \to R}) \stackrel{X \times Y}{:=} (\quad , \qquad)$$

where

Definition (Product of Selection Functions)

Given $\varepsilon \colon JX$ and $\delta \colon JY$ define $\varepsilon \otimes \delta \colon J(X \times Y)$ as

$$(\varepsilon \otimes \delta)(p^{X \times Y \to R}) \stackrel{X \times Y}{:=} (\quad, b(\quad))$$

where

$$b(x) \quad := \quad \delta(\lambda y.p(x,y)).$$

Definition (Product of Selection Functions)

Given $\varepsilon \colon JX$ and $\delta \colon JY$ define $\varepsilon \otimes \delta \colon J(X \times Y)$ as

$$(\varepsilon \otimes \delta)(p^{X \times Y \to R}) \overset{X \times Y}{:=} (a, b(a))$$

where

$$\begin{array}{lcl} a & := & \varepsilon(\lambda x.p(x, b(x))) \\ b(x) & := & \delta(\lambda y.p(x, y)). \end{array}$$

Definition (Product of Selection Functions)

Given $\varepsilon \colon JX$ and $\delta \colon JY$ define $\varepsilon \otimes \delta \colon J(X \times Y)$ as

$$(\varepsilon \otimes \delta)(p^{X \times Y \to R}) \overset{X \times Y}{:=} (a, b(a))$$

where

$$\begin{aligned} a &:= \varepsilon(\lambda x.p(x, b(x))) \\ b(x) &:= \delta(\lambda y.p(x, y)). \end{aligned}$$

Theorem

$\overline{\varepsilon} \otimes \overline{\delta} = \overline{\varepsilon \otimes \delta}$

Iterated Product of Selection Functions

Finite iteration

$$\bigotimes_{i=k}^{n} \mathcal{E}_i \stackrel{J\Pi X_i}{=} \mathcal{E}_k \otimes \left(\bigotimes_{i=k+1}^{n} \mathcal{E}_i \right)$$

Iterated Product of Selection Functions

Infinite iteration $(R$ discrete, $R^{\Pi X_i}$ continuous$)$

$$\bigotimes_{i=k}^{\infty} \varepsilon_i \stackrel{J\Pi X_i}{=} \varepsilon_k \otimes \left(\bigotimes_{i=k+1}^{\infty} \varepsilon_i \right)$$

Iterated Product of Selection Functions

Infinite iteration I (R discrete, $R^{\Pi X_i}$ continuous)

$$\bigotimes_{i=k}^{\infty} \varepsilon_i \overset{J\Pi X_i}{=} \varepsilon_k \otimes \left(\bigotimes_{i=k+1}^{\infty} \varepsilon_i \right)$$

Infinite iteration II ($l \colon R \to \mathbb{N}$, $\mathbb{N}^{\Pi X_i}$ continuous)

$$\left(\bigotimes_{i=k}^{\infty} \varepsilon_i \right)(q) \overset{\Pi X_i}{=} \begin{cases} \mathbf{c} & \text{if } k < l(q(\mathbf{c})) \\ \left(\varepsilon_k \otimes \left(\bigotimes_{i=k+1}^{\infty} \varepsilon_i \right) \right)(q) & \text{otherwise} \end{cases}$$

Iterated Product of Selection Functions

Infinite iteration I (R discrete, $R^{\Pi X_i}$ continuous) = MBR

$$\bigotimes_{i=k}^{\infty} \varepsilon_i \stackrel{J\Pi X_i}{=} \varepsilon_k \otimes \left(\bigotimes_{i=k+1}^{\infty} \varepsilon_i \right)$$

Infinite iteration II ($l \colon R \to \mathbb{N}$, $\mathbb{N}^{\Pi X_i}$ continuous) = SBR

$$\left(\bigotimes_{i=k}^{\infty} \varepsilon_i \right)(q) \stackrel{\Pi X_i}{=} \begin{cases} \mathbf{c} & \text{if } k < l(q(\mathbf{c})) \\ \left(\varepsilon_k \otimes \left(\bigotimes_{i=k+1}^{\infty} \varepsilon_i\right)\right)(q) & \text{otherwise} \end{cases}$$

Outline

Finite Games (n rounds)

Definition (A tuple $(R, (X_i)_{i<n}, (\phi_i)_{i<n}, q)$ where)

- R is the set of **possible outcomes**
- X_i is the set of **available moves** at round i
- $\phi_i \colon K_R X_i$ is the **goal quantifier** for round i
- $q \colon \Pi_{i=0}^{n-1} X_i \to R$ is the **outcome function**

Finite Games (n rounds)

Definition (A tuple $(R, (X_i)_{i<n}, (\phi_i)_{i<n}, q)$ where)

- R is the set of **possible outcomes**
- X_i is the set of **available moves** at round i
- $\phi_i \colon K_R X_i$ is the **goal quantifier** for round i
- $q \colon \Pi_{i=0}^{n-1} X_i \to R$ is the **outcome function**

Definition (Strategy)

Family of mappings

$$\mathsf{next}_k \colon \Pi_{i=0}^{k-1} X_i \to X_k$$

Optimal Strategies

Definition (Strategic Play)

Given strategy next_k and partial play $\vec{a} = a_0, \ldots, a_{k-1}$, the **strategic extension** of $\vec{a}$ is $\mathbf{b}^{\vec{a}} = b_k^{\vec{a}}, \ldots, b_{n-1}^{\vec{a}}$ where

$$b_i^{\vec{a}} = \mathsf{next}_i(\vec{a}, b_k^{\vec{a}}, \ldots, b_{i-1}^{\vec{a}}).$$

Optimal Strategies

Definition (Strategic Play)

Given strategy next_k and partial play $\vec{a} = a_0, \ldots, a_{k-1}$, the **strategic extension** of $\vec{a}$ is $\mathbf{b}^{\vec{a}} = b_k^{\vec{a}}, \ldots, b_{n-1}^{\vec{a}}$ where

$$b_i^{\vec{a}} = \text{next}_i(\vec{a}, b_k^{\vec{a}}, \ldots, b_{i-1}^{\vec{a}}).$$

Definition (Optimal Strategy)

Strategy next_k is **optimal** if for any partial play $\vec{a}$

$$q(\vec{a}, \mathbf{b}^{\vec{a}}) = \phi_k(\lambda x_k.q(\vec{a}, x_k, \mathbf{b}^{\vec{a},x_k})).$$

Examples

Example (Nash Equilibrium with common payoff)

Moves X_i	Sets of moves
Outcomes R	Payoff $\mathbb{R}$
Goal quantifier ϕ_i	Maximal value function
Outcome function q	Payoff function $q\colon \Pi_{i=0}^{n-1} X_i \to \mathbb{R}$

Examples

Example (Nash Equilibrium with common payoff)

Moves X_i	Sets of moves
Outcomes R	Payoff $\mathbb{R}$
Goal quantifier ϕ_i	Maximal value function
Outcome function q	Payoff function $q \colon \Pi_{i=0}^{n-1} X_i \to \mathbb{R}$

Optimal strategy

$$\mathsf{next}_k(x_0, \ldots, x_{k-1}) = \operatorname{argsup}_{x_k} \sup_{x_{k+1}} \ldots \sup_{x_{n-1}} q(\vec{x})$$

Examples

Example (Satisfiability)

Moves X_i	Booleans $\mathbb{B}$
Outcomes R	Boolean $\mathbb{B}$
Goal quantifier ϕ_i	Existential quantifier $\exists\colon K_{\mathbb{B}}\mathbb{B}$
Outcome function q	Formula $q(x_0, \ldots, x_{n-1})$

Examples

Example (Satisfiability)

Moves X_i	Booleans $\mathbb{B}$
Outcomes R	Boolean $\mathbb{B}$
Goal quantifier ϕ_i	Existential quantifier $\exists$: $K_{\mathbb{B}}\mathbb{B}$
Outcome function q	Formula $q(x_0, \ldots, x_{n-1})$

Optimal strategy

$$\text{next}_k(x_0, \ldots, x_{k-1}) \quad = \quad x_k \text{ such that } \exists x_{k+1} \ldots \exists x_{n-1} q(\vec{x})$$

(if possible)

Theorem (Main Theorem for Finite Games)

If ϕ_k are attainable with selection functions ε_k then

$$\mathsf{next}_k(x_0, \ldots, x_{k-1}) \stackrel{X_k}{=} \left(\left(\bigotimes_{i=k}^{n-1} \varepsilon_i \right) (q_{x_0,\ldots,x_{k-1}}) \right)_0$$

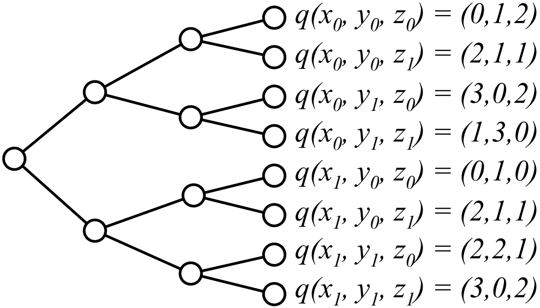
is an **optimal strategy** *for the game* $(R, (X_i)_{i<n}, (\phi_i)_{i<n}, q)$. *Moreover,*

$$\vec{a} = \left(\bigotimes_{i=0}^{n-1} \varepsilon_i \right) (q)$$

is the **strategic play**.

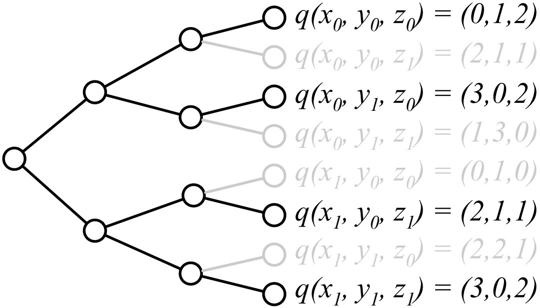
Nash equilibrium (sequential games)

$$q \colon X \times Y \times Z \to \mathbb{R}^3$$



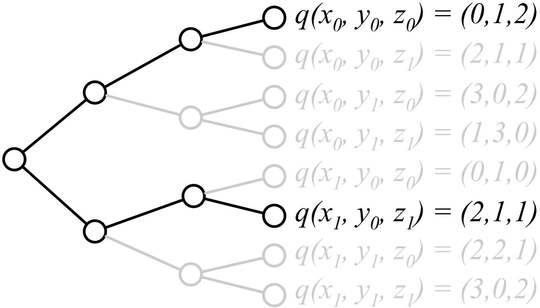
Nash equilibrium (sequential games)

$$q\colon X \times Y \times Z \to \mathbb{R}^3$$



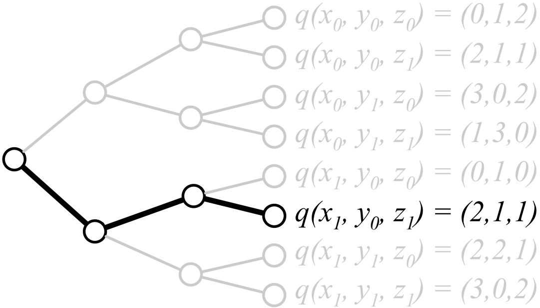
Nash equilibrium (sequential games)

$$q\colon X \times Y \times Z \to \mathbb{R}^3$$



Nash equilibrium (sequential games)

$$q\colon X \times Y \times Z \to \mathbb{R}^3$$



Backward Induction

Let $q \colon \Pi_{i=1}^{n} X_i \to \mathbb{R}^n$ be a payoff function

$\mathrm{argmax}_i(p)$ { $\qquad$ $[\mathrm{argmax}_i \colon (X_i \to \mathbb{R}^n) \to X_i]$

  return $x \in X_i$ such that $p(x)$ has maximal i-coordinate

}

Backward Induction

Let $q \colon \Pi_{i=1}^{n} X_i \to \mathbb{R}^n$ be a payoff function

$\mathrm{argmax}_i(p)$ { $\quad$ $[\mathrm{argmax}_i \colon (X_i \to \mathbb{R}^n) \to X_i]$
  return $x \in X_i$ such that $p(x)$ has maximal i-coordinate
}

$\mathsf{sol}_i(x_1, \ldots, x_{i-1})$ { $\quad$ $[\mathsf{sol}_i \colon \Pi_{k=1}^{i-1} X_k \to \Pi_{k=i}^{n} X_k]$
  if $i = n + 1$ return $\langle\,\rangle$
  else
    $y := \mathrm{argmax}_i(\lambda x.q(\mathsf{sol}_{i+1}(x_1, \ldots, x_{i-1}, x)))$
    return $y * \mathsf{sol}_{i+1}(x_1, \ldots, x_{i-1}, y)$
}

Backward Induction

Let $q \colon \Pi_{i=1}^{n} X_i \to \mathbb{R}^n$ be a payoff function

$\mathrm{argmax}_i(p)$ { $\qquad$ $[\mathrm{argmax}_i \colon (X_i \to \mathbb{R}^n) \to X_i]$
  return $x \in X_i$ such that $p(x)$ has maximal i-coordinate
}

$\mathsf{sol}_i(x_1, \ldots, x_{i-1})$ { $\qquad$ $[\mathsf{sol}_i \colon \Pi_{k=1}^{i-1} X_k \to \Pi_{k=i}^{n} X_k]$
  if $i = n + 1$ return $\langle\,\rangle$
  else
    $y := \mathrm{argmax}_i(\lambda x . q(\mathsf{sol}_{i+1}(x_1, \ldots, x_{i-1}, x)))$
    return $y * \mathsf{sol}_{i+1}(x_1, \ldots, x_{i-1}, y)$
}

$\langle x_1, \ldots, x_n \rangle := \mathsf{sol}_1(\,)$

Backward Induction

Payoff function $q \colon \Pi_{i=1}^{n} X_i \to \mathbb{R}^n$

Each selection function

$$\mathrm{argmax}_i \colon (X_i \to \mathbb{R}^n) \to X_i$$

finds a point where the argument is i-maximal

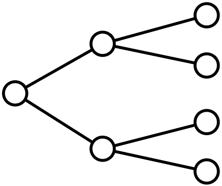
Product

$$\mathsf{sol}_1(\,) = \left(\bigotimes_{i=1}^{n} \mathrm{argmax}_i \right)(q)$$

calculates a **strategy profile in Nash equilibrium**.

Backtracking

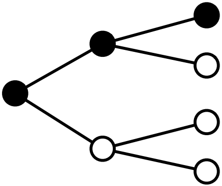
good: $X \times Y \to \mathbb{B}$



Generic algorithm has type $(X \times Y \to \mathbb{B}) \to X \times Y$.

Backtracking

good: $X \times Y \to \mathbb{B}$

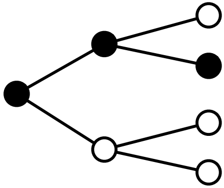


$\text{good}(x_1, y_1) = \text{F}$

Generic algorithm has type $(X \times Y \to \mathbb{B}) \to X \times Y$.

Backtracking

good: $X \times Y \to \mathbb{B}$



$\text{good}(x_1, y_1) = \mathrm{F}$

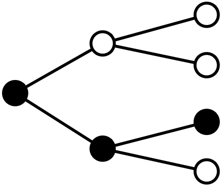
$\text{good}(x_1, y_2) = \mathrm{F}$

$\text{good}(x_2, y_1) = \mathrm{T}$

Generic algorithm has type $(X \times Y \to \mathbb{B}) \to X \times Y$.

Backtracking

$\text{good}\colon X \times Y \to \mathbb{B}$



$\text{good}(x_1,y_1) = \text{F}$

$\text{good}(x_1,y_2) = \text{F}$

$\text{good}(x_2,y_1) = \text{T}$

Generic algorithm has type $(X \times Y \to \mathbb{B}) \to X \times Y$.

For Instance – Eight Queens Problem

$\varepsilon(p)$ {
  for $(i := 1; i \leq 8; i{+}{+})$ do
    if $p(i)$ return i
  return 1
}

$[\, \varepsilon \colon (8 \to \mathbb{B}) \to 8 \,]$

For Instance – Eight Queens Problem

$$
\begin{array}{l}
\varepsilon(p)\ \{ \\
\quad \text{for } (i := 1; i \leq 8; i{+}{+}) \text{ do} \\
\qquad \text{if } p(i) \text{ return } i \\
\quad \text{return } 1 \\
\}
\end{array}
\qquad [\, \varepsilon\colon (8 \to \mathbb{B}) \to 8 \,]
$$

$$
\begin{array}{l}
\mathsf{sol}_i(x_1, \ldots, x_{i-1})\ \{ \\
\quad \text{if } i > 8 \text{ return } \langle\ \rangle \\
\quad \text{else} \\
\qquad y := \varepsilon(\lambda x_i.\mathsf{good}(\mathsf{sol}_{i+1}(x_1, \ldots, x_i))) \\
\qquad \text{return } y * \mathsf{sol}_{i+1}(x_1, \ldots, x_{i-1}, y) \\
\}
\end{array}
\qquad [\, \mathsf{sol}_i\colon 8^{i-1} \to 8^{9-i} \,]
$$

For Instance – Eight Queens Problem

$\varepsilon(p)$ {
  for $(i := 1; i \leq 8; i{+}{+})$ do
    if $p(i)$ return i
  return 1
}

$[\, \varepsilon\colon (8 \to \mathbb{B}) \to 8 \,]$

$\mathsf{sol}_i(x_1, ..., x_{i-1})$ {
  if $i > 8$ return $\langle\ \rangle$
  else
    $y := \varepsilon(\lambda x_i.\mathsf{good}(\mathsf{sol}_{i+1}(x_1, \ldots, x_i)))$
    return $y * \mathsf{sol}_{i+1}(x_1, \ldots, x_{i-1}, y)$
}

$[\, \mathsf{sol}_i : 8^{i-1} \to 8^{9-i} \,]$

$\langle x_1, \ldots, x_8 \rangle := \mathsf{sol}_1(\ )$

For Instance – Eight Queens Problem

good: $8^8 \to \mathbb{B}$ checks if argument is solution to 8QP.

For Instance – Eight Queens Problem

good: $8^8 \to \mathbb{B}$ checks if argument is solution to 8QP.

Selection function

$$\varepsilon\colon (8 \to \mathbb{B}) \to 8$$

finds argument $\varepsilon_i p \in 8$ such that $p(\varepsilon_i p)$ holds

For Instance – Eight Queens Problem

good: $8^8 \longrightarrow \mathbb{B}$ checks if argument is solution to 8QP.

Selection function

$$\varepsilon\colon (8 \longrightarrow \mathbb{B}) \longrightarrow 8$$

finds argument $\varepsilon_i p \in 8$ such that $p(\varepsilon_i p)$ holds

$$\mathsf{sol}_1(\,) = \left(\bigotimes_{i=1}^{8} \varepsilon_i\right)(\mathsf{good})$$

calculates a solution to 8 queen problem.

Classical Arithmetic

Finite product interprets **bounded collection**

Classical Arithmetic

Finite product interprets **bounded collection**

E.g. consider the infinite PHP

$$\forall c^{\mathbb{N} \to n} \exists b^n \forall i \exists j (j \geq i \wedge c(j) = b)$$

Classical Arithmetic

Finite product interprets **bounded collection**

E.g. consider the infinite PHP

$$\forall c^{\mathbb{N}\to n} \exists b^n \forall i \exists j (j \geq i \wedge c(j) = b)$$

Equivalent (dialectica) to

$$\forall c^{\mathbb{N}\to n}, \forall \varepsilon^{J_{\mathbb{N}}\mathbb{N}} \exists b^n, p^{\mathbb{N}\to\mathbb{N}} (\overline{\varepsilon}_b p \geq \varepsilon_b p \wedge c(\overline{\varepsilon}_b p) = b)$$

Classical Arithmetic

Finite product interprets **bounded collection**

E.g. consider the infinite PHP

$$\forall c^{\mathbb{N}\to n} \exists b^n \forall i \exists j (j \geq i \wedge c(j) = b)$$

Equivalent (dialectica) to

$$\forall c^{\mathbb{N}\to n}, \forall \varepsilon^{J_{\mathbb{N}}\mathbb{N}} \exists b^n, p^{\mathbb{N}\to\mathbb{N}} (\overline{\varepsilon}_b p \geq \varepsilon_b p \wedge c(\overline{\varepsilon}_b p) = b)$$

Witnessed by

$$b = c(\overline{\left(\bigotimes_{i=0}^{n-1} \varepsilon_i\right)(\max)})$$

Outline

Finite but Unbounded Games

Finite but Unbounded Games

Example (Chess)

Moves X_i	Valid chess moves
Outcomes R	White, black, draw, e.g. $\{-1, 0, 1\}$
Goal quantifier ϕ_{2i}	Maximisation function
Goal quantifier ϕ_{2i+1}	Minimisation function
Outcome function q	Adjudication on a given play α

Finite but Unbounded Games

Example (Chess)

Moves X_i	Valid chess moves
Outcomes R	White, black, draw, e.g. $\{-1, 0, 1\}$
Goal quantifier ϕ_{2i}	Maximisation function
Goal quantifier ϕ_{2i+1}	Minimisation function
Outcome function q	Adjudication on a given play α

The game is drawn, upon a correct claim by the player having the move, if

a. *he writes on his scoresheet, and declares to the arbiter his intention to make a move which shall result in the last* 50 *moves having been made by each player without the movement of any pawn and without the capture of any piece*, or
b. *the last* 50 *consecutive moves have been made by each player without the movement of any pawn and without the capture of any piece.*

With this rule, it can be shown that the game is finite, assuming that given the option to call for a draw, at least one player will do so.

Finite but Unbounded Games

Definition (Tuple $(R, (X_i)_{i \in \mathbb{N}}, (\phi_i)_{i \in \mathbb{N}}, q)$ where...)

- R is the set of possible **discrete** outcomes
- X_i is the set of available moves X_i at round $i \in \mathbb{N}$
- $\phi_i \colon K_R X_i$ are goal quantifiers for round $i \in \mathbb{N}$
- $q \colon \Pi_{i=0}^{\infty} X_i \to R$ is a **continous** outcome function

Optimal Strategies

Definition (Strategic Play)

Given strategy next_k and partial play $\vec{a} = a_0, \ldots, a_{k-1}$, the **strategic extension** of $\vec{a}$ is $\beta^{\vec{a}} = \beta^{\vec{a}}(k), \beta^{\vec{a}}(k+1), \ldots$ where

$$\beta^{\vec{a}}(i) = \text{next}_i(\vec{a}, \beta^{\vec{a}}(k), \ldots, \beta^{\vec{a}}(i-1)).$$

Optimal Strategies

Definition (Strategic Play)

Given strategy next_k and partial play $\vec{a} = a_0, \ldots, a_{k-1}$, the **strategic extension** of $\vec{a}$ is $\beta^{\vec{a}} = \beta^{\vec{a}}(k), \beta^{\vec{a}}(k+1), \ldots$ where

$$\beta^{\vec{a}}(i) = \mathsf{next}_i(\vec{a}, \beta^{\vec{a}}(k), \ldots, \beta^{\vec{a}}(i-1)).$$

Definition (Optimal Strategy)

Strategy next_k is **optimal** if for any partial play $\vec{a}$

$$q(\vec{a} * \beta^{\vec{a}}) = \phi_k(\lambda x_k.q(\vec{a} * x_k * \beta^{\vec{a},x_k})).$$

Finite but Unbounded Games

Theorem (Main Theorem for Finite but Unbounded Games)

If ϕ_k are attainable with selection functions ε_k then

$$\mathsf{next}_k(x_0, \ldots, x_{k-1}) \stackrel{X_k}{=} \left(\left(\bigotimes_{i=k}^{\infty} \varepsilon_i \right) (q_{x_0, \ldots, x_{k-1}}) \right)_0$$

is an **optimal strategy** *for the game $(R, (X_i)_{i \in \mathbb{N}}, (\phi_i)_{i \in \mathbb{N}}, q)$. Moreover,*

$$\alpha = \left(\bigotimes_{i=0}^{\infty} \varepsilon_i \right) (q)$$

is the **strategic play**.

Classical Analysis

Mathematical analysis is based on **comprehension**

$$\exists f^{\mathbb{N}\to\mathbb{B}} \forall n^{\mathbb{N}} (fn \leftrightarrow A_n).$$

Classical Analysis

Mathematical analysis is based on **comprehension**

$$\exists f^{\mathbb{N} \to \mathbb{B}} \forall n^{\mathbb{N}} (fn \leftrightarrow A_n).$$

Comprehension follows classically from **countable choice**

$$\forall n^{\mathbb{N}} \exists b^{\mathbb{B}} A_n(b) \longrightarrow \exists f^{\mathbb{N} \to \mathbb{B}} \forall n^{\mathbb{N}} A_n(fn).$$

Classical Analysis

Mathematical analysis is based on **comprehension**

$$\exists f^{\mathbb{N}\to\mathbb{B}} \forall n^{\mathbb{N}} (fn \leftrightarrow A_n).$$

Comprehension follows classically from **countable choice**

$$\forall n^{\mathbb{N}} \exists b^{\mathbb{B}} A_n(b) \to \exists f^{\mathbb{N}\to\mathbb{B}} \forall n^{\mathbb{N}} A_n(fn).$$

Countable choice is classically computational up to $\mathbf{DNS}$

$$\forall n^{\mathbb{N}} \neg\neg A_n \to \neg\neg \forall n^{\mathbb{N}} A_n.$$

Double negation shift

The double negation shift $\mathbf{DNS}$

$$\forall n \neg\neg A_n \to \neg\neg \forall n A_n$$

corresponds to the type

$$\Pi_n K_\perp A_n \to K_\perp \Pi_n A_n.$$

Double negation shift

The double negation shift $\mathbf{DNS}$

$$\forall n \neg\neg A_n \to \neg\neg \forall n A_n$$

corresponds to the type

$$\Pi_n K_\perp A_n \to K_\perp \Pi_n A_n.$$

If $\perp \to A_n$, this is equivalent to

$$\Pi_n J_\perp A_n \to J_\perp \Pi_n A_n.$$

Double negation shift

The double negation shift **DNS**

$$\forall n \neg\neg A_n \to \neg\neg \forall n A_n$$

corresponds to the type

$$\Pi_n K_\perp A_n \to K_\perp \Pi_n A_n.$$

If $\perp \to A_n$, this is equivalent to

$$\Pi_n J_\perp A_n \to J_\perp \Pi_n A_n.$$

The type of the **countable product** of selection functions!

Bar recursion

Not a coincidence!

Modified bar recursion is equivalent to

$$\bigotimes_{i=k}^{\infty} \varepsilon_i \overset{J\Pi X_i}{=} \varepsilon_k \otimes \left(\bigotimes_{i=k+1}^{\infty} \varepsilon_i \right)$$

Spector's bar recursion is equivalent to

$$\left(\bigotimes_{i=k}^{\infty} \varepsilon_i \right)(q) \overset{\Pi X_i}{=} \begin{cases} \mathbf{c} & \text{if } k < l(q(\mathbf{c})) \\ \left(\varepsilon_k \otimes \left(\bigotimes_{i=k+1}^{\infty} \varepsilon_i \right) \right)(q) & \text{otherwise} \end{cases}$$

Summary and Further Applications

- New notion of **sequential game** based on gen. quantifiers
- E.g. Nash equilibrium, backtracking, Bekič's lemma

Summary and Further Applications

- New notion of **sequential game** based on gen. quantifiers
- E.g. Nash equilibrium, backtracking, Bekič's lemma
- Product of sel. fct. calculates **optimal strategies**

Summary and Further Applications

- New notion of **sequential game** based on gen. quantifiers
- E.g. Nash equilibrium, backtracking, Bekič's lemma
- Product of sel. fct. calculates **optimal strategies**
- Product of sel. fct. = **bar recursion**

Summary and Further Applications

- New notion of **sequential game** based on gen. quantifiers
- E.g. Nash equilibrium, backtracking, Bekič's lemma
- Product of sel. fct. calculates **optimal strategies**
- Product of sel. fct. = **bar recursion**
- New "**negative translation**" based on J strong monad

Summary and Further Applications

- New notion of **sequential game** based on gen. quantifiers
- E.g. Nash equilibrium, backtracking, Bekič's lemma
- Product of sel. fct. calculates **optimal strategies**
- Product of sel. fct. $=$ **bar recursion**

- New "**negative translation**" based on J strong monad
- Functional interpretations (**proof mining**)

Theorems	$\mapsto$	games
Proofs	$\mapsto$	winning strategies

A Few Open Questions

1. Equivalent notion for **simultaneous games**

   Logic: Branching quantifiers

   GT: Standard Nash equilibrium

A Few Open Questions

1. Equivalent notion for **simultaneous games**
   Logic: Branching quantifiers
   GT: Standard Nash equilibrium
2. Relation to **BBC functional**
   Berardi, Bezem, Coquand 1998

A Few Open Questions

1. Equivalent notion for **simultaneous games**

   Logic: Branching quantifiers

   GT: Standard Nash equilibrium

2. Relation to **BBC functional**

   Berardi, Bezem, Coquand 1998

3. **Approximately** attainable quantifiers

   Family ε_n approximates a selection function

A Few Open Questions

1. Equivalent notion for **simultaneous games**
   Logic: Branching quantifiers
   GT: Standard Nash equilibrium
2. Relation to **BBC functional**
   Berardi, Bezem, Coquand 1998
3. **Approximately** attainable quantifiers
   Family ε_n approximates a selection function
4. Is product of selection functions equivalent to product of quantifiers?

A Few Open Questions

1. Equivalent notion for **simultaneous games**

   Logic: Branching quantifiers

   GT: Standard Nash equilibrium

2. Relation to **BBC functional**

   Berardi, Bezem, Coquand 1998

3. **Approximately** attainable quantifiers

   Family ε_n approximates a selection function

4. Is product of selection functions equivalent to product of quantifiers?

5. Other places where $\bigotimes$ appear?

References

M. Escardó and P. Oliva
Selection functions, bar recursion and backward induction
MSCS, 20(2):127-168, 2010

M. Escardó and P. Oliva
The Peirce translation and the double negation shift
LNCS, CiE'2010

M. Escardó and P. Oliva
Computational interpretations of analysis via products of selection functions
LNCS, CiE'2010