Sequential Games and Optimal Strategies

Paulo Oliva

Queen Mary University of London

LIAMF, USP

São Paulo, 20 October 2011
Outline

1. Player-based Games
2. Quantifiers and Selection Functions
3. Playerless Games
4. Computing Optimal Strategies
Outline

1. Player-based Games
2. Quantifiers and Selection Functions
3. Playerless Games
4. Computing Optimal Strategies
## Single-player Games

**Sudoku**

```
 8  4  2  9  4  6
2  5  7  4  1  8  9  7
 9  1  5  8  3  4
 5  2  6  7  2  1  3
 4  6  9  7  8
1 1  3  2  4  3  7  5
 3  2  3  4  5  7  6
 3  6  5  1  3  2  1
 7  1  4  7  9  4  3  2
```

*Difficulty: Hard*

*Time: 19:09*

**Solitaire**

```
2 7 5 4 8 7 6 5 4
8 7 5 4 7 6 5 4 3
7 6 5 4 3 2 1 2 3
```

*Time: 0:54*  
*Moves: 21*

**Bejeweled**

```
```

*Score: 00000*
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)
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
Traditional Game Theory

Game defined via:

- Set of **players** $P$
- Sets of **moves** $X_i$ for each player $i \in P$
- Set of **outcomes** $R$
- **Preference relations** on $R$ for each player $i \in P$
- **Outcome function** mapping plays to outcomes
Set of Players 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”
Set of Players 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?
Set of Players 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 pay for your flight?
As in...

Q: How much would you like to pay for your flight?
A: As little as possible!
Target function

If $R = \text{set of outcomes}$ and $X = \text{set of possible moves}$ then

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

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:**

- $X$ = *possible flights*
- $R$ = *real number*
- $X \to R$ = *price of each flight*
- $\phi$ = *minimal value functional*
Outline

1. Player-based Games
2. Quantifiers and Selection Functions
3. Playerless Games
4. Computing Optimal Strategies
Generalised quantifiers

\[
\phi : (X \rightarrow R) \rightarrow R
\]
Generalised quantifiers

\[ \phi : (X \to R) \to R \]

For instance

<table>
<thead>
<tr>
<th>Operation</th>
<th>( \phi : (X \to R) \to R )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantifiers</td>
<td>( \forall_X, \exists_X : (X \to \mathbb{B}) \to \mathbb{B} )</td>
</tr>
<tr>
<td>Double negation</td>
<td>( \neg \neg X : (X \to \bot) \to \bot )</td>
</tr>
<tr>
<td>Integration</td>
<td>( \int_0^1 : ([0, 1] \to \mathbb{R}) \to \mathbb{R} )</td>
</tr>
<tr>
<td>Supremum</td>
<td>( \sup_{[0,1]} : ([0, 1] \to \mathbb{R}) \to \mathbb{R} )</td>
</tr>
<tr>
<td>Limit</td>
<td>( \lim : (\mathbb{N} \to \mathbb{R}) \to \mathbb{R} )</td>
</tr>
<tr>
<td>Fixed point operator</td>
<td>( \text{fix}_X : (X \to X) \to X )</td>
</tr>
</tbody>
</table>
### Generalised quantifiers

\[ \phi : (X \to R) \to R \quad (\equiv K_R X) \]

#### For instance

<table>
<thead>
<tr>
<th>Operation</th>
<th>( \phi : (X \to R) \to R )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantifiers</td>
<td>( \forall X, \exists X : (X \to \mathbb{B}) \to \mathbb{B} )</td>
</tr>
<tr>
<td>Double negation</td>
<td>( \neg\neg X : (X \to \bot) \to \bot )</td>
</tr>
<tr>
<td>Integration</td>
<td>( \int_0^1 : ([0, 1] \to \mathbb{R}) \to \mathbb{R} )</td>
</tr>
<tr>
<td>Supremum</td>
<td>( \sup_{[0,1]} : ([0, 1] \to \mathbb{R}) \to \mathbb{R} )</td>
</tr>
<tr>
<td>Limit</td>
<td>( \lim : (\mathbb{N} \to \mathbb{R}) \to \mathbb{R} )</td>
</tr>
<tr>
<td>Fixed point operator</td>
<td>( \text{fix}_X : (X \to X) \to X )</td>
</tr>
</tbody>
</table>
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 $p: X \to \mathbb{B}$ there is a point $a \in X$ such that

$$\exists x^X p(x) \iff p(a)$$

(similar to Hilbert’s $\varepsilon$-term).
Theorem (Witness Theorem)

For any \( p: X \to \mathbb{B} \) there is a point \( a \in X \) such that

\[
\exists x^X p(x) \iff p(a)
\]

(similar to Hilbert’s \( \varepsilon \)-term).

Theorem (Counter-example Theorem)

For any \( p: X \to \mathbb{B} \) there is a point \( a \in X \) such that

\[
\forall x^X p(x) \iff p(a)
\]

(\( a \) 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)**

$\varepsilon : JX$ is called a **selection function** for $\phi : (X \to R) \to R$ if

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

holds for all $p : X \to R$.
Let \( JX \equiv (X \rightarrow R) \rightarrow X \)

**Definition (Selection Functions)**

\( \varepsilon : JX \) is called a **selection function** for \( \phi : (X \rightarrow R) \rightarrow 2^R \) if

\[
p(\varepsilon p) \in \phi(p)
\]

holds for all \( p : X \rightarrow R \)
Let $JX \equiv (X \to R) \to X$

**Definition (Selection Functions)**

$\varepsilon : JX$ is called a **selection function** for $\phi : (X \to R) \to 2^R$ if

$$p(\varepsilon p) \in \phi(p)$$

holds for all $p : X \to R$

**Definition (Attainable Quantifiers)**

A generalised quantifier $\phi : KX$ is called **attainable**

if it has a selection function $\varepsilon : JX$
For Instance

- \( \text{sup} : K_{\mathbb{R}}[0, 1] \) is an attainable quantifier as
  \[
  \text{sup}(p) = p(\text{argsup}(p))
  \]
  where \( \text{argsup} : J_{\mathbb{R}}[0, 1] \)
For Instance

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

- \( \text{fix}: K_X X \) is an attainable quantifier as
  \[
  \text{fix}(p) = p(\text{fix}(p))
  \]
  where \( \text{fix}: J_X X \) (\( K_X X \))
Every selection function $\varepsilon : JX$ defines a quantifier $\overline{\varepsilon} : KX$

$$\overline{\varepsilon}(p) = p(\varepsilon(p))$$
Selection Functions and Generalised Quantifiers

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

$$\phi(p) = 0$$
Selection Functions and Generalised Quantifiers

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

$$\varepsilon_0(p) = \mu x. \sup p = p(x)$$
$$\varepsilon_1(p) = \nu x. \sup p = p(x)$$
Outline

1. Player-based Games
2. Quantifiers and Selection Functions
3. Playerless Games
4. Computing Optimal Strategies
Finite Sequential Games

**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 \(i\)
- \(\phi_i: (X_i \rightarrow R) \rightarrow 2^R\) is the **goal quantifier** for round \(i\)
- \(q: \prod_{i=0}^{n-1} X_i \rightarrow R\) is the **outcome function**
Finite Sequential Games

**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 $i$
- $\phi_i : (X_i \rightarrow R) \rightarrow 2^R$ is the **goal quantifier** for round $i$
- $q : \prod_{i=0}^{n-1} X_i \rightarrow R$ is the **outcome function**

**Definition (Strategy)**

Family of mappings

$$\text{next}_k : \prod_{i=0}^{k-1} X_i \rightarrow X_k$$
Optimal Strategies

Definition (Strategic Play)

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

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

Definition (Strategic Play)

Given strategy $\text{next}_k$ and partial play $\vec{a} = a_0, \ldots, a_{k-1}$, the **strategic extension** of $\vec{a}$ is $b^{\vec{a}} = b^{\vec{a}}_k, \ldots, b^{\vec{a}}_{n-1}$ where

$$b^{\vec{a}}_i = \text{next}_i(\vec{a}, b^{\vec{a}}_k, \ldots, b^{\vec{a}}_{i-1})$$

Definition (Optimal Strategy)

Strategy $\text{next}_k$ is **optimal** if for any partial play $\vec{a}$

$$q(\vec{a}, b^{\vec{a}}) \in \phi_k(\lambda x_k.q(\vec{a}, x_k, b^{\vec{a}}, x_k))$$
## Examples

**Example (Nash Equilibrium with common payoff)**

<table>
<thead>
<tr>
<th>Moves</th>
<th>Sets of moves</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_i$</td>
<td></td>
</tr>
<tr>
<td>Outcomes $R$</td>
<td>Payoff $\mathbb{R}$</td>
</tr>
<tr>
<td>Goal quantifier $\phi_i$</td>
<td>Maximal value function</td>
</tr>
<tr>
<td>Outcome function $q$</td>
<td>Payoff function $q : \Pi_{i=0}^{n-1} X_i \rightarrow \mathbb{R}$</td>
</tr>
</tbody>
</table>

Optimal strategy $\text{next}^k(x_0,...,x_{k-1}) = \arg\sup_x x_k \sup_{x_{k+1}} ... \sup_{x_{n-1}} q(\vec{x})$
Examples

Example (Nash Equilibrium with common payoff)

- **Moves** $X_i$
- **Outcomes** $R$
- **Goal quantifier** $\phi_i$
- **Outcome function** $q$

| Sets of moves | Payoff $\mathbb{R}$ | Maximal value function | Payoff function $q : \Pi_{i=0}^{n-1} X_i \rightarrow \mathbb{R}$ |

**Optimal strategy**

\[
\text{next}_k(x_0, \ldots, x_{k-1}) = \text{argsup}_{x_k} \sup_{x_{k+1}} \ldots \sup_{x_{n-1}} q(\vec{x})
\]
Examples

**Example (Satisfiability)**

<table>
<thead>
<tr>
<th>Moves</th>
<th>$X_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outcomes</td>
<td>$R$</td>
</tr>
<tr>
<td>Goal quantifier</td>
<td>$\phi_i$</td>
</tr>
<tr>
<td>Outcome function</td>
<td>$q$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Booleans</th>
<th>$\mathbb{B}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>$\mathbb{B}$</td>
</tr>
<tr>
<td>Existential quantifier</td>
<td>$\exists: K_{\mathbb{B}}\mathbb{B}$</td>
</tr>
<tr>
<td>Formula</td>
<td>$q(x_0, \ldots, x_{n-1})$</td>
</tr>
</tbody>
</table>
Examples

Example (Satisfiability)

<table>
<thead>
<tr>
<th>Moves</th>
<th>Outcomes</th>
<th>Goal quantifier</th>
<th>Outcome function</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_i$</td>
<td>$R$</td>
<td>$\phi_i$</td>
<td>$q(x_0, \ldots, x_{n-1})$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optimal strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{next}<em>k(x_0, \ldots, x</em>{k-1}) = x_k \text{ such that } \exists x_{k+1} \ldots \exists x_{n-1} q(\vec{x})$ (if possible)</td>
</tr>
</tbody>
</table>
Outline

1. Player-based Games
2. Quantifiers and Selection Functions
3. Playerless Games
4. Computing Optimal Strategies
Backward Induction (Classical Game Theory)

Three players, payoff function $q : X \times Y \times Z \rightarrow \mathbb{R}^3$

Each player is trying to maximise their own payoff

$q(x_0, y_0, z_0) = (0,1,2)$
$q(x_0, y_0, z_1) = (2,1,1)$
$q(x_0, y_1, z_0) = (3,0,2)$
$q(x_0, y_1, z_1) = (1,3,0)$
$q(x_1, y_0, z_0) = (0,1,0)$
$q(x_1, y_0, z_1) = (2,1,1)$
$q(x_1, y_1, z_0) = (2,2,1)$
$q(x_1, y_1, z_1) = (3,0,2)$
Backward Induction (Classical Game Theory)

Three players, payoff function $q: X \times Y \times Z \rightarrow \mathbb{R}^3$

Each player is trying to maximise their own payoff

$q(x_0, y_0, z_0) = (0,1,2)$
$q(x_0, y_0, z_1) = (2,1,1)$
$q(x_0, y_1, z_0) = (3,0,2)$
$q(x_0, y_1, z_1) = (1,3,0)$
$q(x_1, y_0, z_0) = (0,1,0)$
$q(x_1, y_0, z_1) = (2,1,1)$
$q(x_1, y_1, z_0) = (2,2,1)$
$q(x_1, y_1, z_1) = (3,0,2)$
Backward Induction (Classical Game Theory)

Three players, payoff function $q : X \times Y \times Z \rightarrow \mathbb{R}^3$

Each player is trying to maximise their own payoff

\[
\begin{align*}
q(x_0, y_0, z_0) &= (0, 1, 2) \\
q(x_0, y_0, z_1) &= (2, 1, 1) \\
q(x_0, y_1, z_0) &= (3, 0, 2) \\
q(x_0, y_1, z_1) &= (1, 3, 0) \\
q(x_1, y_0, z_0) &= (0, 1, 0) \\
q(x_1, y_0, z_1) &= (2, 1, 1) \\
q(x_1, y_1, z_0) &= (2, 2, 1) \\
q(x_1, y_1, z_1) &= (3, 0, 2)
\end{align*}
\]
Backward Induction (Classical Game Theory)

Three players, payoff function \( q : X \times Y \times Z \rightarrow \mathbb{R}^3 \)

Each player is trying to maximise their own payoff

\[
q(x_0, y_0, z_0) = (0,1,2) \\
q(x_0, y_0, z_1) = (2,1,1) \\
q(x_0, y_1, z_0) = (3,0,2) \\
q(x_0, y_1, z_1) = (1,3,0) \\
q(x_1, y_0, z_0) = (0,1,0) \\
q(x_1, y_0, z_1) = (2,1,1) \\
q(x_1, y_1, z_0) = (2,2,1) \\
q(x_1, y_1, z_1) = (3,0,2)
\]
Backward Induction (Classical Game Theory)

Let $\arg\max_i : (X_i \to \mathbb{R}^n) \to X_i$ find a point $x \in X_i$

at which the function $p : X_i \to \mathbb{R}^n$ has maximal $i$-value
Backward Induction (Classical Game Theory)

Let $\text{argmax}_i : (X_i \rightarrow \mathbb{R}^n) \rightarrow X_i$ find a point $x \in X_i$ at which the function $p : X_i \rightarrow \mathbb{R}^n$ has maximal $i$-value.

Consider $n$ player. Given $q : \prod_{i=0}^{n-1} X_i \rightarrow \mathbb{R}^n$, define

$$\text{Bl} (s) \overset{\Pi_{j=|s|}^{n-1} X_j}{\equiv} \begin{cases} \lfloor \rfloor & \text{if } n = |s| \\ c_s \ast \text{Bl}(s \ast c_i) & \text{otherwise} \end{cases}$$

where $c_s = \text{argmax}_{|s|} (\lambda x. q(s \ast x \ast \text{Bl}(s \ast x)))$.
Backward Induction (Classical Game Theory)

Let \( \text{argmax}_i : (X_i \rightarrow \mathbb{R}^n) \rightarrow X_i \) find a point \( x \in X_i \)
at which the function \( p : X_i \rightarrow \mathbb{R}^n \) has maximal \( i \)-value

Consider \( n \) player. Given \( q : \Pi_{i=0}^{n-1} X_i \rightarrow \mathbb{R}^n \), define

\[
\text{BI}(s) \quad \Pi_{j=|s|}^{n-1} X_j \begin{cases} 
[] & \text{if } n = |s| \\
\text{c}_s \ast \text{BI}(s \ast c_i) & \text{otherwise}
\end{cases}
\]

where \( c_s = \text{argmax}_{|s|} (\lambda x. q(s \ast x \ast \text{BI}(s \ast x))) \)

Each player’s **optimal strategy** can be described as

\[
\text{next}_i(s) = \text{argmax}_{|s|} (\lambda x. q(s \ast x \ast \text{BI}(s \ast x)))
\]

\( p : X_{|s|} \rightarrow \mathbb{R}^n \)
Let
\[
  s: X^* \quad q: X^* \rightarrow R \quad \varepsilon_s: J_R X
\]
Spector’s Bar Recursion (1962)

Let 

\[ s : X^* \quad q : X^* \rightarrow R \quad \varepsilon_s : J_R X \]

Given \( s, \omega \) and \( \varepsilon_s \) define

\[
\text{BR}(s) \overset{X^*}{=} \begin{cases} 
[\ ] & \text{if } n = |s| \\
 c \ast \text{BR}(s \ast c) & \text{otherwise}
\end{cases}
\]

where \( c = \varepsilon_s(\lambda x.q(s \ast x \ast \text{BR}(s \ast x))) \)
Spector’s Bar Recursion (1962)

Let

\[ s : X^* \quad q : X^* \rightarrow R \quad \varepsilon_s : J_{RX} \]

Given \( s, \omega \) and \( \varepsilon_s \) define

\[ \text{BR}(s) \overset{X^*}{=} \begin{cases} \text{[]} & \text{if } n = |s| \\ c \ast \text{BR}(s \ast c) & \text{otherwise} \end{cases} \]

where \( c = \varepsilon_s(\lambda x. q(s \ast x \ast \text{BR}(s \ast x))) \)
Spector’s Bar Recursion (1962)

Let

\[ s : X^* \quad q : X^* \to R \quad \varepsilon_s : J_{R_X} \]

Given \( s, \omega \) and \( \varepsilon_s \) define

\[ \text{BR}(s) \overset{X^*}{=} \begin{cases} [] & \text{if } n = |s| \\ c * \text{BR}(s * c) & \text{otherwise} \end{cases} \]

where \( c = \varepsilon_s(\lambda x. q(s * x * \text{BR}(s * x))) \)

Spector actually defined a much more general recursion scheme where stopping condition depends on the play \( s \)
Main Theorem

Theorem (Escardó/O.’2011)

Given game \((R, X, \phi, q)\), if \(\phi\) are attainable with selection functions \(\varepsilon_i\) then

\[ \text{next}(s) \overset{x}{=} (\text{BR}(s))_0 \]

is an optimal strategy, i.e.

\[ q(s \ast b^s) \in \phi_{|s|}(\lambda x. q(s \ast x \ast b^{s*x})) \]

where \(b^s\) is the strategic extension of partial play \(s\)
Summary and Further Connections

- New notion of **sequential game** based on quantifiers
Summary and Further Connections

- New notion of **sequential game** based on quantifiers
- Generalisation of backward induction, based on selection functions, calculates **optimal strategies**
Summary and Further Connections

- New notion of **sequential game** based on quantifiers
- Generalisation of backward induction, based on selection functions, calculates **optimal strategies**
- Relates Nash equilibrium, backtracking, Bekič’s lemma
Summary and Further Connections

- New notion of **sequential game** based on quantifiers
- Generalisation of backward induction, based on selection functions, calculates **optimal strategies**
- Relates Nash equilibrium, backtracking, Bekič’s lemma
- Connection to proof theory
  \[ KA \rightarrow A \] corresponds to **double negation elimination**
  \[ JA \rightarrow A \] corresponds to **Peirce’s law**
Summary and Further Connections

- New notion of **sequential game** based on quantifiers
- Generalisation of backward induction, based on selection functions, calculates **optimal strategies**
- Relates Nash equilibrium, backtracking, Bekić’s lemma
- Connection to proof theory
  
  \[ KA \rightarrow A \] corresponds to **double negation elimination**
  
  \[ JA \rightarrow A \] corresponds to **Peirce’s law**
- Calculation of strategies in general corresponds to Spector’s bar recursion, used in the proof of **consistency of classical analysis**
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
Sequential games and optimal strategies
*Proceedings of the Royal Society A, 2011*