Algorithmic game theory: Game models

Nicola Gatti

Politecnico di Milano, Italy
1 Introduction

2 Models

- Mechanisms in strategic form
- Mechanisms in extensive form
  - Normal form
  - Multi-agent form
  - Sequence form
- Bayesian games
- Repeated games
- Infinitely repeated games
- Stochastic games
A game is formally defined by a pair:

- *Mechanism* $M$, defining the rules of the game
- *Strategies* $\sigma$, defining the behavior of each agent in the game
Game model

A game is formally defined by a pair:
- *Mechanism* $M$, defining the rules of the game
- *Strategies* $\sigma$, defining the behavior of each agent in the game

There are three main classes of mechanisms:
- *Strategic–form* mechanisms: agents play without observing the actions undertaken by the opponents (simultaneous games)
- *Extensive–form* mechanisms: there is a sequential tree–based structure according which an agent can observe some opponents’ actions
- *Stochastic–form* mechanisms: there is a sequential graph–based structure according which an agent can observe some opponents’ actions
Definition

A strategic–form mechanism is a tuple

\[ \mathcal{M} = (N, \{A\}_{i \in N}, X, f, \{U\}_{i \in N}) \]

- \( N \): set of agents
- \( A_i \): set of actions available to agent \( i \)
- \( X \): set of outcomes
- \( f : \times_{i \in N} A_i \rightarrow X \): outcome function
- \( U_i : X \rightarrow \mathbb{R} \): utility function of agent \( i \)
Example: Rock–Paper–Scissors

- $N = \{\text{agent 1, agent 2}\}$
- $A_1 = A_2 = \{R, P, S\}$
- $X = \{\text{win1, win2, tie}\}$
- $f(R, S) = f(P, R) = f(S, P) = \text{win 1}, f(S, R) = f(R, P) = f(P, S) = \text{win2, tie otherwise}$
- $U_i(\text{win}) = 1, U_i(\text{win} \neq i) = -1, U_i(\text{tie}) = 0$
Example: Rock–Paper–Scissors

- \( N = \{\text{agent 1, agent 2}\} \)
- \( A_1 = A_2 = \{R, P, S\} \)
- \( X = \{\text{win1, win2, tie}\} \)
- \( f(R, S) = f(P, R) = f(S, P) = \text{win 1}, f(S, R) = f(R, P) = f(P, S) = \text{win2, tie otherwise} \)
- \( U_i(\text{win}_i) = 1, U_i(\text{win} \neq i) = -1, U_i(\text{tie}) = 0 \)

Matrix–based representation

\[
\begin{array}{c|ccc}
& R & P & S \\
\hline 
R & 0, 0 & -1, 1 & 1, -1 \\
P & 1, -1 & 0, 0 & -1, 1 \\
S & -1, 1 & 1, -1 & 0, 0 \\
\end{array}
\]
Example: three–player game

- $A_1 = \{a, b\}$
- $A_2 = \{L, R\}$
- $A_3 = \{A, B, C\}$

<table>
<thead>
<tr>
<th></th>
<th>$L$</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>2,2,1</td>
<td>0,3,0</td>
</tr>
<tr>
<td>$b$</td>
<td>3,0,2</td>
<td>1,1,4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$L$</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>2,3,0</td>
<td>0,4,1</td>
</tr>
<tr>
<td>$b$</td>
<td>3,1,2</td>
<td>1,2,0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$L$</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>2,1,0</td>
<td>1,0,2</td>
</tr>
<tr>
<td>$b$</td>
<td>0,3,1</td>
<td>2,3,1</td>
</tr>
</tbody>
</table>
Matrix–based games

Classification

- **Matrix game**: the agents’ utilities can be represented by a unique matrix (this happens with two–agent constant–sum games: $U_1 + U_2 = constant$ for every entry)
- **Bimatrix game**: two–agent general–sum games
- **Polymatrix game**: the utility $U_i$ of each agent $i$ can be expressed as a set of matrices $U_{i,j}$ depending only on the actions of agent $i$ and agent $j$
  - with non–polymatrix games, $U_i$ has $\prod_{j \in N} |A_j|$ entries
  - with polymatrix games, $U_i$ has $|A_i| \sum_{j \in N, j \neq i} |A_j|$ entries
A strategy $\sigma_i$ of agent $i$ is a probability distribution over the actions $A_i$.

Call $x_{i,j}$ the probability with which agent $i$ plays action $j$ and $x_i$ the vector of $x_{i,j}$, we need that:
- $x_i \geq 0$
- $1^T x_i = 1$

A strategy profile $\sigma$ is the collection of one strategy per agent, $\sigma = (\sigma_1, \ldots, \sigma_{|N|})$. 
A strategy $\sigma_i$ of agent $i$ is a probability distribution over the actions $A_i$.

Call $x_{i,j}$ the probability with which agent $i$ plays action $j$ and $x_i$ the vector of $x_{i,j}$, we need that:
1. $x_i \geq 0$
2. $1^T x_i = 1$

A strategy profile $\sigma$ is the collection of one strategy per agent, $\sigma = (\sigma_1, \ldots, \sigma_{|N|})$.

Example

With Rock–Paper–Scissors games can be:

$$x_1 = \begin{bmatrix} x_{1,R} = 0.2 \\ x_{1,P} = 0.8 \\ x_{1,S} = 0.0 \end{bmatrix} \quad x_2 = \begin{bmatrix} x_{2,R} = 0.6 \\ x_{2,P} = 0.0 \\ x_{2,S} = 0.4 \end{bmatrix}$$
The expected utility of an agent $i$ related to an action $j$ is:

$$
\left( U_i \prod_{k \in N, k \neq i} x_k \right)_j
$$

where $(A)_j$ is the $j$–th row of matrix $A$

$U_i \prod_{k \in N, k \neq i} x_k$ is the vector of expected utilities of agent $i$

The expected utility of an agent $i$ related to a strategy $x_i$ is:

$$
x_i^T U_i \prod_{k \in N, k \neq i} x_k
$$
Expected utility (2)

Example

\[ U_1 = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \quad x_1 = \begin{bmatrix} 0.2 \\ 0.8 \\ 0.0 \end{bmatrix} \quad x_2 = \begin{bmatrix} 0.6 \\ 0.0 \\ 0.4 \end{bmatrix} \]

- The expected utilities related to each action of agent 1 are:
  \[ \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0.6 \\ 0.0 \\ 0.4 \end{bmatrix} = \begin{bmatrix} 0.4 \\ 0.2 \\ -0.6 \end{bmatrix} \]

- The expected utility related to the strategy of agent 1 is:
  \[ \begin{bmatrix} 0.2 & 0.8 & 0.0 \end{bmatrix} \cdot \begin{bmatrix} 0.4 \\ 0.2 \\ -0.6 \end{bmatrix} = 2.4 \]
Definition

Given two games with utility functions $U_1, \ldots, U_{|N|}$ and $U'_1, \ldots, U'_{|N|}$ respectively, if, for every $i \in N$, there is an affine transformation between $U_i$ and $U'_i$ such that $U'_i = \alpha_i U_i + \beta_i A_1$ where $A_1$ is a matrix of ones, then the two games are equivalent.
Game equivalence

**Definition**
Given two games with utility functions $U_1, \ldots, U_{|N|}$ and $U'_1, \ldots, U'_{|N|}$ respectively, if, for every $i \in N$, there is an affine transformation between $U_i$ and $U'_i$ such that $U'_i = \alpha_i U_i + \beta_i A_1$ where $A_1$ is a matrix of ones, then the two games are equivalent.

**Example**

$U_1 = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}$

$U_1' = 3 \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 4 \\ 4 & 1 & -2 \\ -2 & 4 & 1 \end{bmatrix}$
Games in extensive form with perfect information

Definition

An extensive–form mechanism with perfect information is a tuple

\[ \mathcal{M} = (N, \{A\}_i \in N, \{V_i\}_i \in N, T, \iota, \rho, \chi, \{U_i\}_i \in N) \]

- \( N \): set of agents
- \( A_i \): set of actions available to agent \( i \), \( A = \bigcup_{i \in N} A_i \)
- \( V_i \): set of decision nodes of agent \( i \), \( V = \bigcup_{i \in N} V_i \)
- \( T \): set of terminal nodes
- \( \iota : \times_{i \in N} V \to N \): player function, returning the agent playing at a given decision node
- \( \rho : V \to \wp(A) \): action function, returning the available actions at a given decision node
- \( \chi : V \times A \to V \cup T \): next node function, returning the next node given a node and an action
- \( U_i : X \to \mathbb{R} \): utility function of agent \( i \)
Example (1)

- $N = \{1, 2\}$
- $V_1 = \{1.1\}$ and $V_2 = \{2.1\}$
- $A_1 = \{L, R\}$ and $A_2 = \{l, r\}$
- $T = \{t_1, t_2, t_3\}$
- $\nu(1.1) = 1$ and $\nu(2.1) = 2$
- $\rho(1.1) = \{L, R\}$ and $\rho(2.1) = \{l, r\}$
- $\chi(1.1, L) = t_1, \chi(1.1, R) = 2.1, \chi(2.1, l) = t_2$, and $\chi(2.1, r) = t_3$
- $U_1(t_1) = 2, U_1(t_2) = 3, U_1(t_3) = 0$, and $U_2(t_1) = 2, U_2(t_2) = 1, U_2(t_3) = 0$
Example (2)

```
1.1
  L
  2.1
    l_1
    1, 1
  r_1
    2, 2
  R
    2.2
      l_2
      3, 3
      r_2
      4, 4
```
Games in extensive form with imperfect information

Definition

An extensive–form mechanism with imperfect information is a tuple

\[
\mathcal{M} = (N, \{A\}_{i \in N}, \{V_i\}_{i \in N}, T, \iota, \rho, \chi, \{U\}_{i \in N}, \{H_i\}_{i \in N})
\]

- \((N, \{A\}_{i \in N}, \{V_i\}_{i \in N}, T, \iota, \rho, \chi, \{U\}_{i \in N})\) is a perfect information game
- \(H_i\) induces a partition \(V_i = \bigcup_{h \in H_i} V_{i,h}\) such that for all \(w, w' \in V_{i,h}\) we have \(\rho(w) = \rho(w')\)
- \(V_{i,h}\): information set

The concept of information set expresses: a set of decision nodes such that when an agent plays in one of them it cannot recognize the node in which it is playing.
agent 2 cannot distinguish action $M$ from action $R$

the two decision nodes of agent 2 reached from actions $M$ and $R$ constitute an information set

the actions available at different nodes of the same information set are the same (otherwise an agent could recognize the node in which it is)
a strategic–form game can be represented in extensive form such there is exactly one information set per agent
Strategic form of an extensive–form game: normal form

**Definition**

The normal–form of an extensive–form game is defined as follows:

- the normal–form actions (called *plans*) are tuple specifying one action for each information set
- the payoffs associated to a profile of plans are the payoffs reachable by the execution of the plans
Example (1)

Algorithmic game theory

Nicola Gatti (Politecnico di Milano, Italy)
Example (2)

\[
\begin{array}{c}
\langle L \rangle \\
\langle M \rangle \\
\langle R \rangle \\
\end{array}
\begin{array}{c|cc}
\langle l \rangle & \langle r \rangle \\
1,1 & 1,1 \\
2,2 & 3,3 \\
4,4 & 5,5 \\
\end{array}
\]

Nicola Gatti (Politecnico di Milano, Italy)
Example (3)

```
1
\[ D \]
\[ L \quad R \]
\[ 3,3,2 \quad 0,0,0 \]

\[ C \]
\[ 3 \]
\[ L \quad R \]
\[ 4,4,0 \quad 0,0,1 \]

\[ 2 \quad c \]
\[ d \]
\[ 1,1,1 \]
```

\[
\langle C \rangle \\
\langle C \rangle \\
\langle D \rangle \\
\langle D \rangle \\
\langle L \rangle \\
\langle R \rangle \\
\]

\[
| \langle c \rangle | \quad | \langle d \rangle | \\
|-----------------|-----------------|-----------------|-----------------| \\
| \langle C \rangle | 1,1,1 | 4,4,0 | 1,1,1 | 0,4,1 | \\
| \langle D \rangle | 3,3,2 | 3,3,2 | 0,0,0 | 0,0,1 |
Size of the normal form

Observation
The normal form rises exponentially in the number of information set, even keeping constant the number of outcomes.

Example
Consider a two-stage game where:
- agent 1 plays in the first stage with $m$ actions
- agent 2 plays in the second stage with $n$ actions
- the number of outcomes is $m \cdot n$

We have:
- with completely imperfect information the size of the normal form is $m \cdot n$
- with perfect information the size of the normal form is $n^m$
Example (1)
Example (2)

![Game Tree Diagram]

Nicola Gatti (Politecnico di Milano, Italy)
Reduced normal form

Observation
Many plans in the normal form are equivalent and therefore they can be removed.

Definition
The reduced normal form is obtained from the normal form by removing plans that differ for actions at unreachable information sets.

Observation
Although the reduced normal form is more compact than the normal form, it keeps to be exponential in the number of information sets.
Example (1)
Example (2)

<table>
<thead>
<tr>
<th>Configuration</th>
<th>( l )</th>
<th>( r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \langle L, L_1, L_2, L_3 \rangle )</td>
<td>1, 1</td>
<td>1, 1</td>
</tr>
<tr>
<td>( \langle L, L_1, L_2, R_3 \rangle )</td>
<td>1, 1</td>
<td>1, 1</td>
</tr>
<tr>
<td>( \langle L, L_1, R_2, L_3 \rangle )</td>
<td>1, 1</td>
<td>1, 1</td>
</tr>
<tr>
<td>( \langle L, L_1, R_2, R_3 \rangle )</td>
<td>1, 1</td>
<td>1, 1</td>
</tr>
<tr>
<td>( \langle L, R_1, L_2, L_3 \rangle )</td>
<td>1, 1</td>
<td>1, 1</td>
</tr>
<tr>
<td>( \langle L, R_1, L_2, R_3 \rangle )</td>
<td>1, 1</td>
<td>1, 1</td>
</tr>
<tr>
<td>( \langle L, R_1, R_2, L_3 \rangle )</td>
<td>1, 1</td>
<td>1, 1</td>
</tr>
<tr>
<td>( \langle L, R_1, R_2, R_3 \rangle )</td>
<td>1, 1</td>
<td>1, 1</td>
</tr>
<tr>
<td>( \langle M, L_1, L_2, L_3 \rangle )</td>
<td>2, 3</td>
<td>1, 1</td>
</tr>
<tr>
<td>( \langle M, L_1, L_2, R_3 \rangle )</td>
<td>2, 3</td>
<td>0, 0</td>
</tr>
<tr>
<td>( \langle M, L_1, R_2, L_3 \rangle )</td>
<td>2, 3</td>
<td>0, 0</td>
</tr>
<tr>
<td>( \langle M, L_1, R_2, R_3 \rangle )</td>
<td>2, 3</td>
<td>0, 0</td>
</tr>
<tr>
<td>( \langle M, R_1, L_2, L_3 \rangle )</td>
<td>5, 2</td>
<td>1, 1</td>
</tr>
<tr>
<td>( \langle M, R_1, L_2, R_3 \rangle )</td>
<td>5, 2</td>
<td>1, 1</td>
</tr>
<tr>
<td>( \langle M, R_1, R_2, L_3 \rangle )</td>
<td>5, 2</td>
<td>0, 0</td>
</tr>
<tr>
<td>( \langle M, R_1, R_2, R_3 \rangle )</td>
<td>5, 2</td>
<td>0, 0</td>
</tr>
<tr>
<td>( \langle R, L_1, L_2, L_3 \rangle )</td>
<td>1, 3</td>
<td>2, 0</td>
</tr>
<tr>
<td>( \langle R, L_1, L_2, R_3 \rangle )</td>
<td>0, 2</td>
<td>5, 1</td>
</tr>
<tr>
<td>( \langle R, L_1, R_2, L_3 \rangle )</td>
<td>1, 3</td>
<td>2, 0</td>
</tr>
<tr>
<td>( \langle R, L_1, R_2, R_3 \rangle )</td>
<td>0, 2</td>
<td>5, 1</td>
</tr>
<tr>
<td>( \langle R, R_1, L_2, L_3 \rangle )</td>
<td>1, 3</td>
<td>2, 0</td>
</tr>
<tr>
<td>( \langle R, R_1, L_2, R_3 \rangle )</td>
<td>0, 2</td>
<td>5, 1</td>
</tr>
<tr>
<td>( \langle R, R_1, R_2, L_3 \rangle )</td>
<td>1, 3</td>
<td>2, 0</td>
</tr>
<tr>
<td>( \langle R, R_1, R_2, R_3 \rangle )</td>
<td>0, 2</td>
<td>5, 1</td>
</tr>
</tbody>
</table>
Other representations: multi–agent form

Multi–agent form

In the multi–agent form, there is a different fictitious player for each information and fictitious players of the same player have the same utility

Observations

- Multi–agent form casts an $n$–player game into an $m$–player game with $m = \sum_{i \in N} |H_i|$, therefore in the multi–agent form the number of player is much larger
- The size of the multi–agent form is exponential in the number of outcomes, but it can be reduced to be polynomial size by graphical models (called graphical games)
- Multi–agent form presents computational problems, given that, in practice, the complexity of computing solutions rises as the number of players rises
Example

\begin{center}
\begin{tikzpicture}[level distance=1.5cm, level 1/.style={sibling distance=3cm}, level 2/.style={sibling distance=1.5cm}, every node/.style={draw, circle, minimum size=5mm}]

\node {1.1}
child {node {2.1}
child {node {l_1}
child {node {1,1}}}
child {node {r_1}
child {node {2,2}}}}
child {node {R}
child {node {l_2}
child {node {3,3}}}
child {node {r_2}
child {node {4,4}}}}}
\end{tikzpicture}
\end{center}

\begin{center}
\begin{tabular}{|c|c|c|}
\hline
 & $l_1$ & $r_1$ \\
\hline
$L$ & 1,1,1 & 2,2,2 \\
$R$ & 3,3,3 & 3,3,3 \\
\hline
\end{tabular}
\end{center}

\begin{center}
\begin{tabular}{|c|c|c|}
\hline
 & $l_1$ & $r_1$ \\
\hline
$L$ & 1,1,1 & 2,2,2 \\
$R$ & 4,4,4 & 4,4,4 \\
\hline
\end{tabular}
\end{center}
Strategies (normal form)

A strategy $\pi_i$ is a probability distribution over the plans of player $i$:
$$\sum_{j \in P_i} \pi_i(j) = 1$$

Behavioral strategies (multi-agent form)

A behavioral strategy $\sigma_i$ is a probability distribution over the actions of each single information set:
$$\sum_{j \in \rho(h)} \sigma_i(j) = 1 \text{ for all } h \in H_i$$

Definition

The path is the subset of the game tree traversed with strictly positive probability given a (behavioral) strategy profile.
Example (1)
Example (2)

\[ \sigma_1(1.1) = \begin{cases} 
L & 0.3 \\
M & 0.5 \\
R & 0.2 
\end{cases} \]

\[ \sigma_1(1.2) = \begin{cases} 
L_1 & 0.1 \\
R_1 & 0.9 
\end{cases} \]

\[ \sigma_1(1.3) = \begin{cases} 
L_2 & 0.0 \\
R_2 & 1.0 
\end{cases} \]

\[ \sigma_1(1.4) = \begin{cases} 
L_3 & 0.6 \\
R_3 & 0.4 
\end{cases} \]

\[ \sigma_2(2.1) = \begin{cases} 
l & 1.0 \\
r & 0.0 
\end{cases} \]
Expected utility with behavioral strategies

**Definition**
- A profile of behavioral strategy profile \( \sigma = (\sigma_i, \sigma_{-i}) \) induces a probability \( p_{\sigma}(t) \) distribution over \( t \in T \)
- The utility expected by agent \( i \) from \( (\sigma_i, \sigma_{-i}) \) is \( \sum_{t \in T} p_{\sigma}(t)U_i(t) \)

**Observations**
- The expected utility of an agent depends non-linearly even on its own behavioral strategy (i.e., on the multiplication of the behavioral strategies over different information sets)
- The expected utility of an agent depends linearly on its own strategies
**Definition**

A game is with perfect recall if every player recalls all the observed (own and opponents’) actions.

**Observations**

- Perfect recall poses constraints over the structure of the information set.
- Some possible information sets are not allowed.
- Two decision nodes of agent $i$ that are reached by playing two different actions of $i$ cannot belong to the same information set (otherwise agent $i$ forgets its own actions).
- A similar condition can be posed on the basis of the action undertaken by the opponents.
- Every game with perfect information is with perfect recall.
- Every game with a unique information set per player is with perfect recall.
Example with imperfect recall

\[ \begin{array}{c|c|c}
\langle l \rangle & \langle r \rangle \\
1,0 & 1,0 \\
5,1 & 2,2 \\
\end{array} \]
Theorem

In games with perfect recall, there exists for each mixed strategy profile $\pi$ a behavioral strategy profile $\sigma$ such that both strategies induce the same probability distribution over the terminal nodes.
player 1 will play $\sigma_1(1.1, L) = \frac{98}{198}$

(R, l) will be played by the players
**Casting a behavioral strategy as a strategy**

**Procedure**
- Call $j$ a plan
- Call $a_h(j)$ the action played at information set $h$ in plan $j$
- $\pi_i(j) = \prod_{h \in H_i} \sigma_i(h, a_h(j))$

**Observations**
- The above procedure is not unique
- Different behavioral strategies differing for off-the-path strategies can be expressed as the same strategy
Example

\[ \sigma_1(1.1, L_1) = 1 \quad \sigma_1(1.2, L_2) = 1 \quad \pi_1(\langle L_1, L_2 \rangle) = 1 \]
\[ \sigma_1(1.1, L_1) = 1 \quad \sigma_1(1.2, L_2) = 0 \quad \pi_1(\langle L_1, R_2 \rangle) = 1 \]

- But \( \pi_1(\langle L_1, L_2 \rangle) = 1 \) and \( \pi_1(\langle L_1, R_2 \rangle) = 1 \) are equivalent strategies
- With the reduced normal form, we would have a unique strategy \( \langle L_1, * \rangle \) and therefore
  - each behavioral strategy can be written as a unique strategy, but
  - different behavioral strategies can be written as the same strategy
Casting a strategy as a behavioral strategy

**Procedure**

- Call $j$ a plan
- Call $c_i^h$ the set of strategies that are compatible w.r.t. $h$, i.e., strategy profiles $(\pi_i, \pi_{-i})$ such that $h$ is traversed with strictly positive probability by the combination of $\pi_i$ with opponents’ strategy $\pi_{-i}$
- Call $c_i^h(a)$ the set of strategies that are compatible w.r.t. $h$ and that prescribe agent $i$ to play action $a$ at $h$
- $\sigma_i(h, a) \cdot \sum_{j \in c_i^h} \pi_i(j) = \sum_{j \in c_i^h(a)} \pi_i(j)$
- The term multiplying $\sigma_i$ is a normalization term
- If there is no compatible strategy, $\sigma_i$ can be any

**Observation**

A single strategy can be represented as a number of different behavioral strategies, differing for the off-the-path strategies
Sequence form

- Every sequence–form action is a sequence of multi–agent form actions.
- There is a fictitious sequence, called empty sequence and denoted by $\emptyset_i$.
- Sequences can be terminal if they lead to terminal nodes for some opponents’ sequences or non–terminal.
- Given a sequence $q_i$, the sequence $q'_i$ extends $q_i$ if there is an action $a$ such that $q'_i = q_i|a$.
- Utilities are represented by sparse matrices in which payoffs are present only for a combination of terminal sequences.
- Sequence form is composed by matrices corresponding to utilities and a set of constraints over the agents’ strategies.
- Constraints assure consistency of the strategies: $x_{\emptyset_i} = 1$, $x_{q_i} = \sum_{q'_i = q_i|a} x_{q'_i}$. 

Nicola Gatti (Politecnico di Milano, Italy)
Example (1)
Example (2)

<table>
<thead>
<tr>
<th></th>
<th>$\emptyset_1$</th>
<th>$\langle L \rangle$</th>
<th>$\langle M \rangle$</th>
<th>$\langle R \rangle$</th>
<th>$\langle M, L_1 \rangle$</th>
<th>$\langle M, R_1 \rangle$</th>
<th>$\langle M, L_2 \rangle$</th>
<th>$\langle M, R_2 \rangle$</th>
<th>$\langle R, L_3 \rangle$</th>
<th>$\langle R, R_3 \rangle$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>1, 1</td>
<td></td>
<td></td>
<td>2, 3</td>
<td>5, 2</td>
<td>1, 1</td>
<td>0, 0</td>
<td>1, 3</td>
<td>0, 2</td>
</tr>
<tr>
<td>2</td>
<td>$\emptyset_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
x_{\emptyset_1} = 1 \\
x_{\langle L \rangle} + x_{\langle M \rangle} + x_{\langle R \rangle} - x_{\emptyset_1} = 0 \\
x_{\langle M, L_1 \rangle} + x_{\langle M, R_1 \rangle} - x_{\langle M \rangle} = 0 \\
x_{\langle M, L_2 \rangle} + x_{\langle M, R_2 \rangle} - x_{\langle M \rangle} = 0 \\
x_{\langle R, L_3 \rangle} + x_{\langle R, R_3 \rangle} - x_{\langle R \rangle} = 0 \\
x_{\emptyset_2} = 1 \\
x_{\langle L \rangle} + x_{\langle R \rangle} - x_{\emptyset_2} = 0
\]
Example (3)

\[ x_1 = \begin{bmatrix} x_{\emptyset} & x_{\langle L \rangle} & x_{\langle M \rangle} & x_{\langle R \rangle} & x_{\langle M, L_1 \rangle} & x_{\langle M, R_1 \rangle} & x_{\langle M, L_2 \rangle} & x_{\langle M, R_2 \rangle} & x_{\langle R, L_3 \rangle} & x_{\langle R, R_3 \rangle} \end{bmatrix} \]

\[ x_2 = \begin{bmatrix} x_{\emptyset} & x_{\langle I \rangle} & x_{\langle R \rangle} \end{bmatrix} \]

\[ F_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \]

\[ f_1 = \begin{bmatrix} 1 \\
0 \\
0 \end{bmatrix} \]

\[ f_2 = \begin{bmatrix} 1 \\
0 \end{bmatrix} \]

\[ F_1 x_1 = f_1 \]

\[ F_2 x_2 = f_2 \]

\[ F_1 \begin{bmatrix} x_{\emptyset} & x_{\langle L \rangle} & x_{\langle M \rangle} & x_{\langle R \rangle} \end{bmatrix} = \begin{bmatrix} 1 \\
0 \\
0 \end{bmatrix} \]

\[ F_2 \begin{bmatrix} x_{\emptyset} & x_{\langle I \rangle} & x_{\langle R \rangle} \end{bmatrix} = \begin{bmatrix} 1 \\
0 \end{bmatrix} \]
The size of the sequence form is linear in the size of the game tree
There are exactly $|T|$ entries in the utility matrices, these being sparse

A sequence–form strategy $x_i$ is a randomization over the sequences satisfying $F_i x_i = f_i$
Every pure sequence–form strategy $x_i$ corresponds to a single plan (in the reduced normal form, to many plans in the non-reduced normal form)
Every sequence–form strategy $x_i$ can be casted as a number of behavioral strategies
Expected utility of a strategy profile

It is defined as in the normal–form case:

$$E_x[U_i(x)] = x_i U_i x_{-i}$$
Games with incomplete information

Definition
A game with incomplete information is characterized by at least a player that does not know the payoffs of the opponents

Example
Suppose that agent 1 does not know the payoffs of agent 2, while agent 2 knows agent 1’s payoffs

\[
\begin{array}{c|cc}
B & b & s \\
\hline
B & 2,\cdot & 0,\cdot \\
S & 0,\cdot & 1,\cdot \\
\end{array}
\]

what player 1 knows
Games with uncertain information

Definition
A game with incomplete information is characterized by at least a player that has uncertain information over the payoffs of the opponents.

Example
Suppose that agent 1 is uncertain over the payoffs of agent 2, while agent 2 knows agent 1’s payoffs.

\[
\begin{array}{c|cc}
 & b & s \\
\hline
B & 2,1 & 0,0 \\
S & 0,0 & 1,2 \\
\end{array}
\]
probability 0.2

\[
\begin{array}{c|cc}
 & b & s \\
\hline
B & 2,0 & 0,2 \\
S & 0,1 & 1,0 \\
\end{array}
\]
probability 0.8
Harsanyi’s transformation

**Definition**

- Each player $i$ is characterized by a number of types $\theta_{i,j}$ and each type $\theta_{i,j}$ is characterized by a utility $U_{i,j}$ and by a probability $\omega_{i,j}$.
- Each player is uncertain over the type of its opponents, i.e., each player is uncertain over the game it is playing.
- At the beginning of the game, nature draws the type of each player.
- Each player observes its own type before playing, while it does not observe the opponents’ types.
- The resulting game is in extensive form with imperfect information.
Example (1)
Example (2): normal form

<table>
<thead>
<tr>
<th></th>
<th>$\langle b_1, b_2 \rangle$</th>
<th>$\langle b_1, s_2 \rangle$</th>
<th>$\langle s_1, b_2 \rangle$</th>
<th>$\langle s_1, s_2 \rangle$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 $\langle B \rangle$</td>
<td>2, $\omega_{2.1}$</td>
<td>$2\omega_{2.1}, \omega_{2.1}$</td>
<td>$2\omega_{2.2}, 0$</td>
<td>0, $\omega_{2.2}$</td>
</tr>
<tr>
<td>2 $\langle S \rangle$</td>
<td>0, $\omega_{2.2}$</td>
<td>$\omega_{2.2}, 2\omega_{2.1}$</td>
<td>$\omega_{2.1}, \omega_{2.2}$</td>
<td>1, $2\omega_{2.1}$</td>
</tr>
</tbody>
</table>
Example (3): sequence form

<table>
<thead>
<tr>
<th></th>
<th>$\emptyset_1$</th>
<th>$\langle B \rangle$</th>
<th>$\langle S \rangle$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\emptyset_2$</td>
<td>$\langle b_1 \rangle$</td>
<td>$\langle s_1 \rangle$</td>
</tr>
<tr>
<td></td>
<td>$2\omega_{2.1}, \omega_{2.1}$</td>
<td>$0, 0$</td>
<td>$2\omega_{2.2}, 0$</td>
</tr>
<tr>
<td></td>
<td>$0, 0$</td>
<td>$\omega_{2.1}, 2\omega_{2.1}$</td>
<td>$0, \omega_{2.2}$</td>
</tr>
<tr>
<td></td>
<td>$0, 0$</td>
<td>$\omega_{2.2}$</td>
<td>$\omega_{2.2}, 0$</td>
</tr>
</tbody>
</table>

$x_{\emptyset_1} = 1$

$x_{\langle B \rangle} + x_{\langle S \rangle} - x_{\emptyset_1} = 0$

$x_{\emptyset_2} = 1$

$x_{\langle b_1 \rangle} + x_{\langle s_1 \rangle} - x_{\emptyset_2} = 0$

$x_{\langle b_2 \rangle} + x_{\langle s_2 \rangle} - x_{\emptyset_2} = 0$
Bayesian games

Definition

A Bayesian game is a tuple

\[ \mathcal{M} = (N, \{\Theta\}_i \in N, \{\Omega_i\}_i \in N, \{A_i\}_i \in N, X, f, \{U_i\}_i \in N) \]

- \( N \): set of agents
- \( \Theta_i \): set of types of agent \( i \)
- \( \Omega_i \): probability distribution over the types of agent \( i \)
- \( A_i \): set of actions available to agent \( i \)
- \( X \): set of outcomes
- \( f : \times_{i \in N} A_i \rightarrow X \): outcome function
- \( U_i : X \times \Theta_i \rightarrow \mathbb{R} \): utility function of agent \( i \)
A repeated game is a strategic–form (extensive–form) game repeated a given number of times.

Observations:
- Any repeated game can be modeled as an extensive–form game.
- In a repeated game, the payoffs are assigned at the end of each game, while in the extensive–form game the payoffs are assigned at the end (however, the two models are the same).
- The payoffs are given by the sum of each repetition of the game.
Repeated games

**Infinitely repeated game**

In the case the game is infinitely repeated, the payoffs would go to infinity, therefore a temporal discount factor $\delta \in (0, 1)$ is used as $\delta^t$ where $t$ is the repetition.

**Observation**

The use of the discount factor is equivalent to say that the game can conclude at the next iteration with a probability $1 - \delta$. 
TO DO