Logic, Games and Puzzles

Eric Pacuit

November 10, 2005

ILLC, University of Amsterdam

staff.science.uva.nl/~epacuit

epacuit@science.uva.nl

Introduction

- Game theoretic reasoning is an important part of logic: semantic evaluation games, model comparison games, etc.

Introduction

- Game theoretic reasoning is an important part of logic: semantic evaluation games, model comparison games, etc.

- What about logical reasoning in game theoretic analyses?

Introduction

- Game theoretic reasoning is an important part of logic: semantic evaluation games, model comparison games, etc.

- What about logical reasoning in game theoretic analyses?

  - Develop ("well-behaved") logical languages that can express game theoretic concepts, such as the *Nash equilibrium*.

  - Use logic to analyze social interactive strategic situations.

Introduction

- Game theoretic reasoning is an important part of logic: semantic evaluation games, model comparison games, etc.

- What about logical reasoning in game theoretic analyses?

  - Develop ("well-behaved") logical languages that can express game theoretic concepts, such as the *Nash equilibrium*.

  - Use logic to analyze social interactive strategic situations.

    * For example, develop logics for reasoning about social procedures *with an eye to creating new ones.*

Introduction

- Game theoretic reasoning is an important part of logic: semantic evaluation games, model comparison games, etc.

- What about logical reasoning in game theoretic analyses?
  - Develop ("well-behaved") logical languages that can express game theoretic concepts, such as the *Nash equilibrium*.
  - Use logic to analyze social interactive strategic situations.
    * For example, develop logics for reasoning about social procedures *with an eye to creating new ones.*

Introduction: What is social software?

Social software is an interdisciplinary research program that combines mathematical tools and techniques from game theory and computer science in order to analyze and design social procedures.

Introduction: What is social software?

Social software is an interdisciplinary research program that combines mathematical tools and techniques from game theory and computer science in order to analyze and design social procedures.

Research in Social Software can be divided into three different but related categories:

Introduction: What is social software?

Social software is an interdisciplinary research program that combines mathematical tools and techniques from game theory and computer science in order to analyze and design social procedures.

Research in Social Software can be divided into three different but related categories:

- Mathematical Models of Social Situations

Introduction: What is social software?

Social software is an interdisciplinary research program that combines mathematical tools and techniques from game theory and computer science in order to analyze and design social procedures.

Research in Social Software can be divided into three different but related categories:

- Mathematical Models of Social Situations
- A Theory of Correctness of Social Procedures

Introduction: What is social software?

Social software is an interdisciplinary research program that combines mathematical tools and techniques from game theory and computer science in order to analyze and design social procedures.

Research in Social Software can be divided into three different but related categories:

- Mathematical Models of Social Situations
- A Theory of Correctness of Social Procedures
- Designing Social Procedures

Introduction: What is social software

Social software is an interdisciplinary research program that combines mathematical tools and techniques from game theory and computer science in order to analyze and design social procedures.

Introduction: What is social software

Social software is an interdisciplinary research program that combines mathematical tools and techniques from game theory and computer science in order to analyze and design social procedures.

Related Fields:

- Computer Science (algorithms, logic of programs, ...),
- Philosophical Logic (epistemic logic, deontic logic, ...),
- Game Theory (mechanism design, ...),
- Artificial Intelligence (multi-agent systems, ...)

Introduction: What is social software

Social software is an interdisciplinary research program that combines mathematical tools and techniques from game theory and computer science in order to analyze and design social procedures.

Introduction: What is social software

Social software is an interdisciplinary research program that combines mathematical tools and techniques from game theory and computer science in order to analyze and design social procedures.

Just as computer programs have logical properties which are analyzed by means of appropriate logics of programs, social procedures also have logical properties which can be analyzed with the appropriate logical tools.

Introduction: What is social software

Social software is an interdisciplinary research program that combines mathematical tools and techniques from game theory and computer science in order to analyze and design social procedures.

Just as computer programs have logical properties which are analyzed by means of appropriate logics of programs, social procedures also have logical properties which can be analyzed with the appropriate logical tools.

- Game Logic (Parikh, Pauly)
- Coalitional Logic (Pauly, Goranko)
- Game Algebra (van Benthem, Goranko, Venema, Pauly)

Introduction: What is social software

Social software is an interdisciplinary research program that combines mathematical tools and techniques from game theory and computer science in order to analyze and design social procedures.

Introduction: What is social software

Social software is an interdisciplinary research program that combines mathematical tools and techniques from game theory and computer science in order to analyze and design social procedures.

Social procedures occur at two levels

1. Individual algorithms set up by a **group** of agents
   (taking a train, dropping a class, ...)
2. Algorithms that require a **group** even in the execution
   (voting procedures, fair division algorithms, piano duets, ...)

Introduction: What is social software

Social software is an interdisciplinary research program that combines mathematical tools and techniques from game theory and computer science in order to analyze and design social procedures.

For more information see [P] *and* [PP2]

- *Social Software*, Rohit Parikh, Synthese 2002
- *Frameworks for Social Software*, EP and Rohit Parikh, forthcoming

Motivating Example I

Ann would like Bob to attend her talk; however, she only wants Bob to attend if he is interested in the subject of her talk, not because he is just being polite.

Motivating Example I

Ann would like Bob to attend her talk; however, she only wants Bob to attend if he is interested in the subject of her talk, not because he is just being polite.

There is a very simple procedure to solve Ann's problem: *have a (trusted) friend tell Bob the time and subject of her talk.*

Motivating Example I

Ann would like Bob to attend her talk; however, she only wants Bob to attend if he is interested in the subject of her talk, not because he is just being polite.

There is a very simple procedure to solve Ann's problem: *have a (trusted) friend tell Bob the time and subject of her talk.*

Taking a cue from computer science, we can ask is this procedure correct?

Motivating Example I

Yes, if

1. Ann knows about the talk.

Motivating Example I

Yes, if

1. Ann knows about the talk.

2. Bob knows about the talk.

Motivating Example I

Yes, if

1. Ann knows about the talk.
2. Bob knows about the talk.
3. Ann knows that Bob knows about the talk.

Motivating Example I

Yes, if

1. Ann knows about the talk.
2. Bob knows about the talk.
3. Ann knows that Bob knows about the talk.
4. Bob *does not* know that Ann knows that he knows about the talk.

Motivating Example I

Yes, if

1. Ann knows about the talk.
2. Bob knows about the talk.
3. Ann knows that Bob knows about the talk.
4. Bob *does not* know that Ann knows that he knows about the talk.
5. *And nothing else.*

Motivating Example II

Ann and Bob each own a horse and each would like to sell their horse.

Charles is willing to pay $100 to the owner whose horse can run the **fastest**.

Motivating Example II

Ann and Bob each own a horse and each would like to sell their horse.

Charles is willing to pay $100 to the owner whose horse can run the **fastest.**

In fact, Ann's horse is faster running at 30 MPH, while Bob's horse can only run at 25 MPH.

The obvious procedure that Charles can use to determine who to give the money to is to ask Ann and Bob to race, and give $100 to the winner.

Motivating Example II

	0	10	20	25
0	50,50	0,100	0,100	0,100
10	100,0	50,50	0,100	0,100
20	100,0	100,0	50,50	0,100
25	100,0	100,0	100,0	50,50
30	100,0	100,0	100,0	100,0

Motivating Example II

	0	10	20	25
0	50,50	0,100	0,100	0,100
10	100,0	50,50	0,100	0,100
20	100,0	100,0	50,50	0,100
25	100,0	100,0	100,0	50,50
30	100,0	100,0	100,0	100,0

On the other hand, if Charles wants to buy the *slower* horse, the above procedure will no longer work.

Adjusted Winner

Adjusted winner (AW) is an algorithm for dividing n divisible goods among two people (invented by Steven Brams and Alan Taylor).

For more information see

- *Fair Division: From cake-cutting to dispute resolution* by Brams and Taylor, 1998
- *The Win-Win Solution* by Brams and Taylor, 2000
- www.nyu.edu/projects/adjustedwinner

Adjusted Winner: Example

Adjusted Winner: Example

Suppose Ann and Bob are dividing three goods: A, B, and C.

Adjusted Winner: Example

Suppose Ann and Bob are dividing three goods: A, B, and C.

Step 1. Both Ann and Bob divide 100 points among the three goods.

Adjusted Winner: Example

Suppose Ann and Bob are dividing three goods: A, B, and C.

Step 1. Both Ann and Bob divide 100 points among the three goods.

Item	Ann	Bob
A	5	4
B	65	46
C	30	50
Total	100	100

Adjusted Winner: Example

Suppose Ann and Bob are dividing three goods: A, B, and C.

Step 2. The agent who assigns the most points receives the item.

Adjusted Winner: Example

Suppose Ann and Bob are dividing three goods: A, B, and C.

Step 2. The agent who assigns the most points receives the item.

Item	Ann	Bob
A	5	4
B	65	46
C	30	50
Total	100	100

Adjusted Winner: Example

Suppose Ann and Bob are dividing three goods: A, B, and C.

Step 2. The agent who assigns the most points receives the item.

Item	Ann	Bob
A	5	0
B	65	0
C	0	50
Total	70	50

Adjusted Winner: Example

Suppose Ann and Bob are dividing three goods: A, B, and C.

Step 3. Equitability adjustment:

Adjusted Winner: Example

Suppose Ann and Bob are dividing three goods: A, B, and C.

Step 3. Equitability adjustment:

Notice that $65/46 \geq 5/4 \geq 1 \geq 30/50$

Item	Ann	Bob
A	5	4
B	65	46
C	30	50
Total	100	100

Adjusted Winner: Example

Suppose Ann and Bob are dividing three goods: A, B, and C.

Step 3. Equitability adjustment:

Adjusted Winner: Example

Suppose Ann and Bob are dividing three goods: A, B, and C.

Step 3. Equitability adjustment:

Give A to Bob (the item whose ratio is closest to 1)

Adjusted Winner: Example

Suppose Ann and Bob are dividing three goods: A, B, and C.

Step 3. Equitability adjustment:

Give A to Bob (the item whose ratio is closest to 1)

Item	Ann	Bob
A	5	0
B	65	0
C	0	50
Total	70	50

Adjusted Winner: Example

Suppose Ann and Bob are dividing three goods: A, B, and C.

Step 3. Equitability adjustment:

Give A to Bob (the item whose ratio is closest to 1)

Item	Ann	Bob
A	0	4
B	65	0
C	0	50
Total	65	54

Adjusted Winner: Example

Suppose Ann and Bob are dividing three goods: A, B, and C.

Step 3. Equitability adjustment:

Still not equal, so give (some of) B to Bob: $65p = 100 - 46p$.

Item	Ann	Bob
A	0	4
B	65	0
C	0	50
Total	65	54

Adjusted Winner: Example

Suppose Ann and Bob are dividing three goods: A, B, and C.

Step 3. Equitability adjustment:

yielding $p = 100/111 = 0.9009$

Item	Ann	Bob
A	0	4
B	65	0
C	0	50
Total	65	54

Adjusted Winner: Example

Suppose Ann and Bob are dividing three goods: A, B, and C.

Step 3. Equitability adjustment:

yielding $p = 100/111 = 0.9009$

Item	Ann	Bob
A	0	4
B	58.559	4.559
C	0	50
Total	58.559	58.559

Adjusted Winner: Formal Definition

Suppose that $G_1, \ldots, G_n$ is a fixed set of goods.

Adjusted Winner: Formal Definition

Suppose that $G_1, \ldots, G_n$ is a fixed set of goods.

A **valuation** of these goods is a vector of natural numbers $\langle a_1, \ldots, a_n \rangle$ whose sum is 100.

Let $\alpha, \alpha', \alpha'', \ldots$ denote possible valuations for Ann and $\beta, \beta', \beta'', \ldots$ denote possible valuations for Bob.

Adjusted Winner: Formal Definition

Suppose that $G_1, \ldots, G_n$ is a fixed set of goods.

Adjusted Winner: Formal Definition

Suppose that $G_1, \ldots, G_n$ is a fixed set of goods.

An **allocation** is a vector of n real numbers where each component is between 0 and 1 (inclusive). An allocation $\sigma = \langle s_1, \ldots, s_n \rangle$ is interpreted as follows.

For each $i = 1, \ldots, n$, s_i is the proportion of G_i given to Ann.

Thus if there are three goods, then $\langle 1, 0.5, 0 \rangle$ means, “Give all of item 1 and half of item 2 to Ann and all of item 3 and half of item 2 to Bob."

Adjusted Winner: Formal Definition

Suppose that $G_1, \ldots, G_n$ is a fixed set of goods.

Adjusted Winner: Formal Definition

Suppose that $G_1, \ldots, G_n$ is a fixed set of goods.

$V_A(\alpha, \sigma) = \sum_{i=1}^{n} a_i s_i$ is the total number of points that Ann receives.

$V_B(\beta, \sigma) = \sum_{i=1}^{n} b_i(1 - s_i)$ is the total number of points that Bob receives.

Thus AW can be viewed as a function from pairs of valuations to allocations: $\mathsf{AW}(\alpha, \beta) = \sigma$ if σ is the allocation produced by the AW algorithm.

Adjusted Winner: Formal Definition

$\mathsf{AW}(\alpha, \beta)$ is defined as follows:

Adjusted Winner: Formal Definition

$\mathsf{AW}(\alpha, \beta)$ is defined as follows:

Rename the goods so that

$$a_1/b_1 \geq a_2/b_2 \geq \cdots a_r/b_r \geq 1 > a_{r+1}/b_{r+1} \geq \cdots a_n/b_n$$

Adjusted Winner: Formal Definition

$\mathsf{AW}(\alpha, \beta)$ is defined as follows:

Rename the goods so that

$$a_1/b_1 \geq a_2/b_2 \geq \cdots a_r/b_r \geq 1 > a_{r+1}/b_{r+1} \geq \cdots a_n/b_n$$

1. Give all the goods $G_1, \ldots, G_r$ to Ann and $G_{r+1}, \ldots, G_n$ to Bob. Let X, Y be the number of points received by Ann and Bob respectively. Assume for simplicity that $X \geq Y$.

Adjusted Winner: Formal Definition

$\mathsf{AW}(\alpha, \beta)$ is defined as follows:

Rename the goods so that

$$a_1/b_1 \geq a_2/b_2 \geq \cdots a_r/b_r \geq 1 > a_{r+1}/b_{r+1} \geq \cdots a_n/b_n$$

1. Give all the goods $G_1, \ldots, G_r$ to Ann and $G_{r+1}, \ldots, G_n$ to Bob. Let X, Y be the number of points received by Ann and Bob respectively. Assume for simplicity that $X \geq Y$.

2. If $X = Y$, then stop. Otherwise, transfer a portion of G_r from Ann to Bob which makes $X = Y$. If equitability is not achieved even with all of G_r going to Bob, transfer $G_{r-1}, G_{r-2}, \ldots, G_1$ in that order to Bob until equitability is achieved.

Fairness Conditions

- **Proportional** if both Ann and Bob receive at least 50% of their valuation. That is, $\sum_{i=1}^{n} s_i a_i \geq 50$ and $\sum_{i=1}^{n}(1 - s_i)b_i \geq 50$
- **Envy-Free** if no party is willing to give up its allocation in exchange for the other player's allocation. That is, $\sum_{i=1}^{n} s_1 a_i \geq \sum_{i=1}^{n}(1 - s_i)a_i$ and $\sum_{i=1}^{n}(1 - s_i)b_i \geq \sum_{i=1}^{n} s_i b_i$.
- **Equitable** if both players receive the same total number of points. That is $\sum_{i=1}^{n} s_i a_i = \sum_{i=1}^{n}(1 - s_i)b_i$
- **Efficient** if there is no other allocation that is strictly better for one party without being worse for another party. That is for each allocation $\sigma' = \langle s'_1, \ldots, s'_n \rangle$ if $\sum_{i=1}^{n} a_i s'_i > \sum_{i=1}^{n} a_i s_i$, then $\sum_{i=1}^{n}(1 - s'_i)b_i < \sum_{i=1}^{n}(1 - s_i)b_i$. (Similarly for Bob).

Fairness Conditions

- **Proportional** if both Ann and Bob receive at least 50% of their valuation. That is, $\sum_{i=1}^{n} s_i a_i \geq 50$ and $\sum_{i=1}^{n}(1 - s_i)b_i \geq 50$
- **Envy-Free** if no party is willing to give up its allocation in exchange for the other player's allocation. That is, $\sum_{i=1}^{n} s_1 a_i \geq \sum_{i=1}^{n}(1 - s_i)a_i$ and $\sum_{i=1}^{n}(1 - s_i)b_i \geq \sum_{i=1}^{n} s_i b_i$.
- **Equitable** if both players receive the same total number of points. That is $\sum_{i=1}^{n} s_i a_i = \sum_{i=1}^{n}(1 - s_i)b_i$
- **Efficient** if there is no other allocation that is strictly better for one party without being worse for another party. That is for each allocation $\sigma' = \langle s'_1, \ldots, s'_n \rangle$ if $\sum_{i=1}^{n} a_i s'_i > \sum_{i=1}^{n} a_i s_i$, then $\sum_{i=1}^{n}(1 - s'_i)b_i < \sum_{i=1}^{n}(1 - s_i)b_i$. (Similarly for Bob).

Fairness Conditions

- **Proportional** if both Ann and Bob receive at least 50% of their valuation. That is, $\sum_{i=1}^{n} s_i a_i \geq 50$ and $\sum_{i=1}^{n}(1 - s_i)b_i \geq 50$
- **Envy-Free** if no party is willing to give up its allocation in exchange for the other player's allocation. That is, $\sum_{i=1}^{n} s_1 a_i \geq \sum_{i=1}^{n}(1 - s_i)a_i$ and $\sum_{i=1}^{n}(1 - s_i)b_i \geq \sum_{i=1}^{n} s_i b_i$.
- **Equitable** if both players receive the same total number of points. That is $\sum_{i=1}^{n} s_i a_i = \sum_{i=1}^{n}(1 - s_i)b_i$
- **Efficient** if there is no other allocation that is strictly better for one party without being worse for another party. That is for each allocation $\sigma' = \langle s'_1, \ldots, s'_n \rangle$ if $\sum_{i=1}^{n} a_i s'_i > \sum_{i=1}^{n} a_i s_i$, then $\sum_{i=1}^{n}(1 - s'_i)b_i < \sum_{i=1}^{n}(1 - s_i)b_i$. (Similarly for Bob).

Fairness Conditions

- **Proportional** if both Ann and Bob receive at least 50% of their valuation. That is, $\sum_{i=1}^{n} s_i a_i \geq 50$ and $\sum_{i=1}^{n}(1 - s_i)b_i \geq 50$

- **Envy-Free** if no party is willing to give up its allocation in exchange for the other player's allocation. That is, $\sum_{i=1}^{n} s_1 a_i \geq \sum_{i=1}^{n}(1 - s_i)a_i$ and $\sum_{i=1}^{n}(1 - s_i)b_i \geq \sum_{i=1}^{n} s_i b_i$.

- **Equitable** if both players receive the same total number of points. That is $\sum_{i=1}^{n} s_i a_i = \sum_{i=1}^{n}(1 - s_i)b_i$

- **Efficient** if there is no other allocation that is strictly better for one party without being worse for another party. That is for each allocation $\sigma' = \langle s'_1, \ldots, s'_n \rangle$ if $\sum_{i=1}^{n} a_i s'_i > \sum_{i=1}^{n} a_i s_i$, then $\sum_{i=1}^{n}(1 - s'_i)b_i < \sum_{i=1}^{n}(1 - s_i)b_i$. (Similarly for Bob).

Easy Observations

- For two-party disputes, proportionality and envy-freeness are equivalent.

- AW only produces equitable allocations (equitability is essentially built in to the procedure).

- AW produces allocations σ that have the following property: there is at most one i such that $0 \leq \sigma_i \leq 1$ and for all $j \neq i$, $\sigma_j \in \{0, 1\}$.

Adjusted Winner **is Fair**

Theorem (Brams and Taylor, 1995) *AW produces allocations that are efficient, equitable and envy-free (with respect to the announced valuations)*

Some Questions

- Can we extend to more than two agents?

Some Questions

- Can we extend to more than two agents?
- Can we make use of geometric intuitions?
  - Is AW a "continuous" function?

Some Questions

- Can we extend to more than two agents?

- Can we make use of geometric intuitions?
  - Is AW a "continuous" function?

- Can an agent benefit by making use of information about the other agent's valuation?

Some Questions

- Can we extend to more than two agents?

- Can we make use of geometric intuitions?
  - Is AW a "continuous" function?

- Can an agent benefit by making use of information about the other agent's valuation?

- The agents' utility functions are assumed to be linear, what about non-linear utility functions?

Some Questions

- Can we extend to more than two agents?

- Can we make use of geometric intuitions?
  - Is AW a "continuous" function?

- Can an agent benefit by making use of information about the other agent's valuation?

- The agents' utility functions are assumed to be linear, what about non-linear utility functions?

See EP, Parikh and Salame (2005), for more information.

Strategizing

Can the agents improve their allocation by misrepresenting their preferences?

Strategizing

Can the agents improve their allocation by misrepresenting their preferences?

Yes

Strategizing

Can the agents improve their allocation by misrepresenting their preferences?

Yes

However, while honesty may not always be the best policy it is the only **safe** *one, i.e., the only one which will guarantee 50%.*

See *Safe votes, sincere votes, and strategizing* [PP] for more on the notion of a "safe" strategy.

Strategizing

Item	Ann	Bob
Matisse	75	25
Picasso	25	75

Ann will get the Matisse and Bob will get the Picasso and each gets 75 of his or her points.

Strategizing: Example

Suppose Ann knows Bob's preferences, but Bob does not know Ann's.

Item	Ann	Bob
M	75	25
P	25	75

Item	Ann	Bob
M	26	25
P	74	75

So Ann will get M plus a portion of P.

According to Ann's announced allocation, she receives 50 points

According to Ann's actual allocation, she receives $75 + 0.33 * 25 = 83.33$ points.

Strategizing: Example

Suppose *both* players know each other's preferences but neither knows that the other knows their own preference.

Item	Ann	Bob
M	75	25
P	25	75

Item	Ann	Bob
M	26	74
P	74	26

Each will get 74 of his or her announced points, but each one is really getting only 25 of his or her *true* points.

Strategizing: Example

Suppose *both* players know each other's preferences. Moreover, Ann knows that Bob knows her preference and Bob doesn't know that Ann knows.

Item	Ann	Bob
M	26	74
P	74	26

Item	Ann	Bob
M	73	74
P	27	26

What happens as the level of knowledge increases?

Some Puzzles

Thus both **epistemic** and **game-theoretic** analyses are important when reasoning about social software.

Some Puzzles

Thus both **epistemic** and **game-theoretic** analyses are important when reasoning about social software.

- Epistemic puzzles: muddy children, sum and product, dining cryptographers, etc.
- Social aggregation puzzles
- Epistemic foundations of game theory

Aggregating Preferences: Some Notation

- Suppose that there are n individuals and two alternatives x and y

- Let xP_iy denote that i prefers x to y and xI_iy denote that i is indifferent between x and y

Aggregating Preferences: Some Notation

- For each i there is a variable $D_i \in \{-1, 0, 1\}$ where

$$D = \begin{cases} -1 & \text{if } \; yP_ix \\ 0 & \text{if } \; xI_iy \\ 1 & \text{if } \; xP_iy \end{cases}$$

- $f : \{-1, 0, 1\}^n \rightarrow \{-1, 0, 1\}$ is the *group decision function*

Simple Majority Procedure

For $k \in \{-1, 0, 1\}$, let

$$N_k(D_1, \ldots, D_n) = |\{i \mid D_i = k\}|$$

Let $\vec{D} = \langle D_1, \ldots, D_n \rangle$

f is a **simple majority decision method** iff

$$f(\vec{D}) = \begin{cases} -1 & \text{if } N_1(\vec{D}) - N_{-1}(\vec{D}) < 0 \\ 0 & \text{if } N_1(\vec{D}) - N_{-1}(\vec{D}) = 0 \\ 1 & \text{if } N_1(\vec{D}) - N_{-1}(\vec{D}) > 0 \end{cases}$$

Properties of group decision funcitons

A group decision function f is

- **Decisive** if it is a total function

- **Symmetric** if $f(D_1, \ldots, D_n) = f(D_{j(1)}, \ldots, D_{j(n)})$ for all permutations j. I.e., f is symmetric in all of its arguments.

- **Neutral** if $f(-D_1, \ldots, -D_n) = -f(D_1, \ldots, D_n)$

- **Positively Responsive** if $D = f(D_1, \ldots, D_n) = 0$ or 1, and $D'_i = D_i$ for all $i \neq i_0$, and $D'_{i_0} > D_{i_0}$, then $D' = f(D'_1, \ldots, D'_n) = 1$

May’s Theorem

Theorem (May, 1952) A group decision function is the method of simple majority decision if and only if it is always decisive, symmetric, neutral and positively responsive

Generalizing May's Theorem

In May's Theorem, the agents are making a single binary choice between two alternatives. What about more general situations?

Generalizing May's Theorem

In May's Theorem, the agents are making a single binary choice between two alternatives. What about more general situations?

- Agents choose between between more than two alternatives.
- There are multiple interconnected propositions on which simultaneous decisions are to be made.

Condorcet Paradox

Suppose that there are three agents choosing between three alternatives.

$$\begin{array}{ll} P_1 & a > b > c \\ P_2 & b > c > a \\ P_3 & c > a > b \end{array}$$

Pairwise majority voting produces a non-transitive group preference.

$$\begin{array}{ll} P_1 & a > b > c \\ P_2 & b > c > a \\ P_3 & c > a > b \end{array}$$

- $a > b$?
- $b > c$?
- $a > c$?

$$\begin{array}{ll} P_1 & a > b > c \\ P_2 & b > c > a \\ P_3 & c > a > b \end{array}$$

- $a > b$?
- $b > c$?
- $a > c$?

P_1	$a > b > c$
P_2	$b > c > a$
P_3	$c > a > b$

- $a > b$? Yes
- $b > c$?
- $a > c$?

P_1	$a > b > c$
P_2	$b > c > a$
P_3	$c > a > b$

- $a > b$? Yes
- $b > c$? Yes
- $a > c$?

P_1	$a > b > c$
P_2	$b > c > a$
P_3	$c > a > b$

- $a > b$? Yes
- $b > c$? Yes
- $a > c$? No

Arrow's Theorem: Some Notation

Let $\mathcal{R}$ be the set of all reflexive, transitive and connected relations on a set of candidates X.

A **social welfare function** F is a function $F : \mathcal{R}^n \to \mathcal{R}$

Suppose that $R = F(R_1, \ldots, R_n)$

Arrow's Theorem: Conditions

- **Universal Domain**: F is a total function
- **Weak Pareto Principle**: For any two candidates x, y if xR_iy for each agent i then $xF(\vec{R})y$
- **Independence of Irrelevant Alternatives**: Suppose that $\vec{R}$ and $\vec{R}^*$ are two preference profiles and x and y are two candidates such that for all individuals i, if xR_iy iff xR_i^*y then $xF(\vec{R})y$ iff $xF(\vec{R}^*)y$.
- **Non-Dictatorship**: There does not exist an individual i such that for all profiles $\vec{R} \in \mathcal{R}^n$, if xR_iy then $xF(\vec{R})y$.

Arrow's Theorem

Theorem (Arrow 1951/1963) There exists no social welfare function which satisfies Universal Domain, Weak Pareto Principle, Independence of Irrelevant Alternatives and Non-Dictatorship.

The Doctrinal Paradox

P: “UvA teachers get a 10% raise"

Q: “The quality of education for all students will increase"

$P \rightarrow Q$: “If UvA teachers get a 10% raise, then the quality of education for all students will increase"

	P	$P \rightarrow Q$	Q
Individual 1	True	True	True
Individual 2	True	False	False
Individual 3	False	True	False
Majority	True	True	False

A Second Paradox (Kornhauser and Sager 1993)

P: a valid contract was in place

Q: the defendant's behaviour was such as to breach a contract of that kind

R: the court is required to find the defendant liable.

	P	Q	$(P \wedge Q) \leftrightarrow R$	R
1	yes	yes	yes	yes
2	yes	no	yes	no
3	no	yes	yes	no

Should we accept R?

	P	Q	$(P \wedge Q) \leftrightarrow R$	R
1	yes	yes	yes	yes
2	yes	no	yes	no
3	no	yes	yes	no

Should we accept R? No, a simple majority votes no.

	P	Q	$(P \wedge Q) \leftrightarrow R$	R
1	yes	yes	yes	yes
2	yes	no	yes	no
3	no	yes	yes	no

Should we accept R? Yes, a majority votes yes for P and Q and $(P \wedge Q) \leftrightarrow R$ is a legal doctrine.

	P	Q	$(P \wedge Q) \leftrightarrow R$	R
1	yes	yes	yes	yes
2	yes	no	yes	no
3	no	yes	yes	no

List and Pettit Impossibility Result

Suppose there are n agetns and let $\mathcal{L}$ be a propositional language.

Personal judgement sets: a *consistent, complete* and *deductively closed* set of formulas — a maximally consistent set.

A **collective judgement aggregation function:** Let $\mathbb{M} = \{\Gamma \mid \Gamma \text{ is a maximally consistent set}\}$ then a collective aggregation function is defined as follows:

$$F : \mathbb{M}^n \to \mathbb{M}$$

Some Conditions

Universal Domain F is a total function

Anonymity For all $\vec{\Gamma} \in \mathbb{M}^n$, $F(\Gamma_1, \ldots, \Gamma_n) = F(\Gamma_{\pi(1)}, \ldots, \Gamma_{\pi(n)})$ for all permutations π

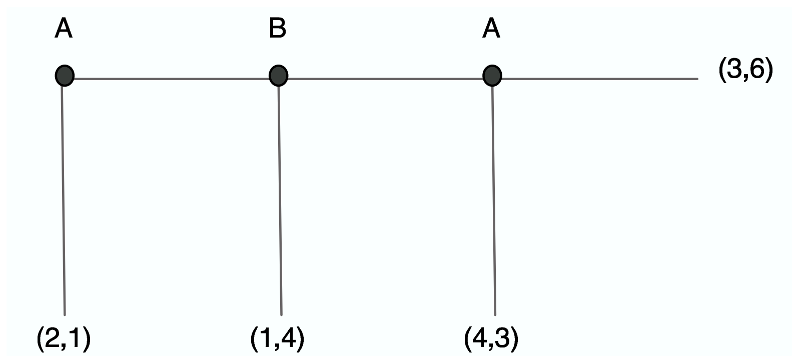
Systematicity There exists a function $f : \{0,1\}^n \rightarrow \{0,1\}$ such that for any $\vec{\Gamma} \in \mathbb{M}^n$,
$F(\Gamma_1, \ldots, \Gamma_n) = \{\phi \in X \mid f(\delta_1(\phi), \ldots, \delta_n(\phi)) = 1\}$, where, for each agent i and each $\phi \in X$, $\delta_i(\phi) = 1$ if $\phi \in \Gamma_i$ and $\delta_i(\phi) = 0$ if $\phi \notin \Gamma_i$

Theorem (List and Pettit, 2001) There exists no judgement aggregation function generating complete, consistent and deductively closed collective sets of judgements which satisfies Universal Domain, Anonymity and Systematicity.

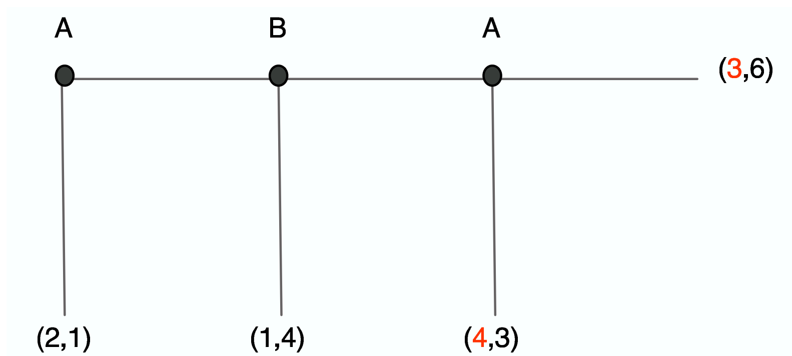
Brief Survey of the Literature

- See `personal.lse.ac.uk/LIST/doctrinalparadox.htm` for a detailed overview of the current state of affairs. Some highlights:
- Other impossibility results: Pauly and van Hees (2003), van Hees (2004), GŁrdenfors (2004), among others
- List and Pettit (2005) compare their impossibility result with Arrow's Theorem
- Pauly (2005), develops a modal logic for reasoning about judgement aggregation

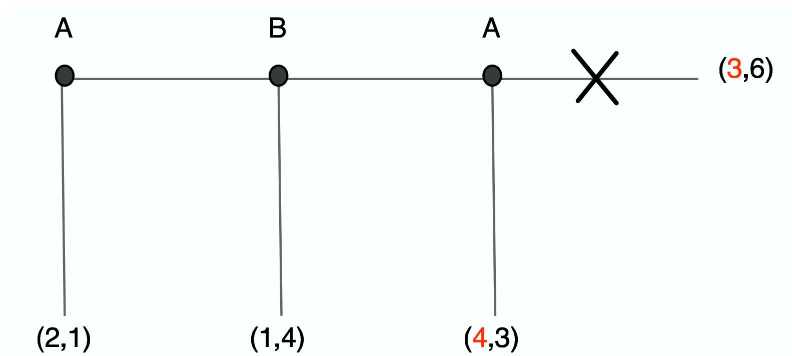
Backward Induction



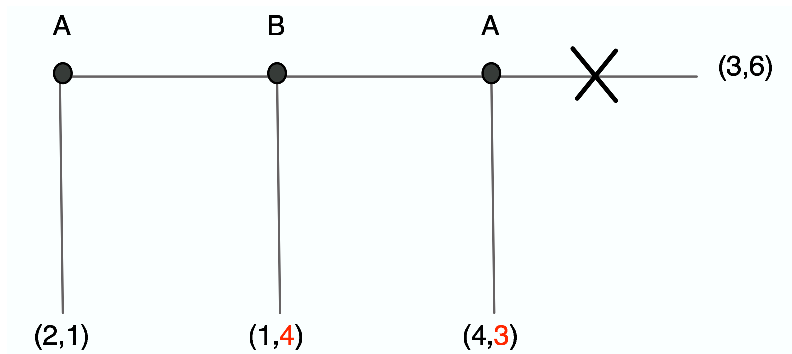
Backward Induction



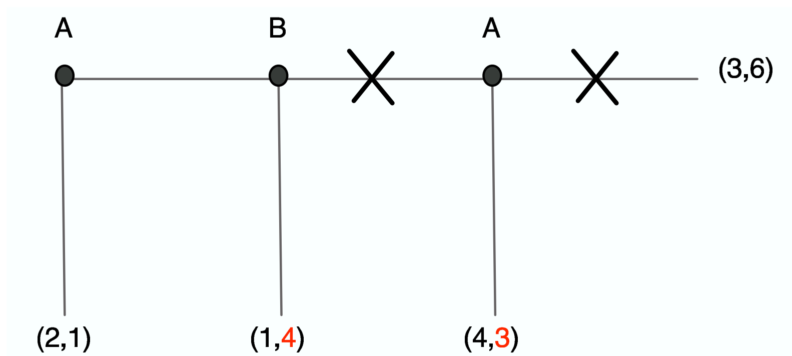
Backward Induction



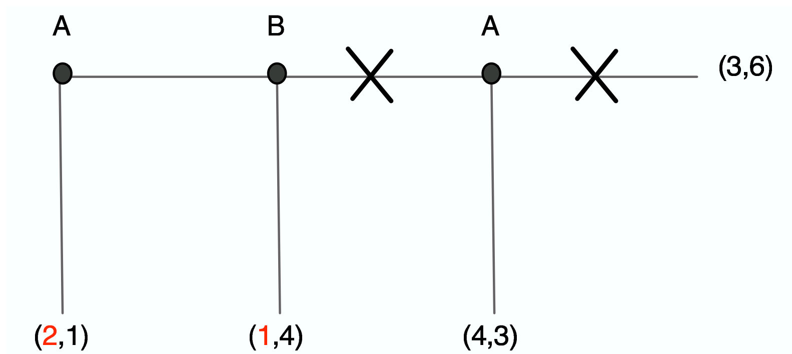
Backward Induction



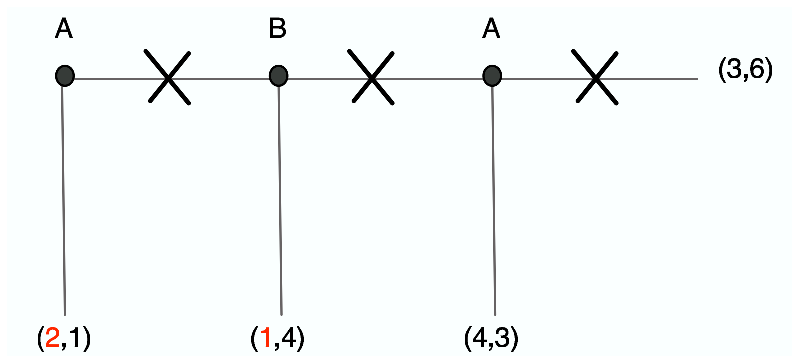
Backward Induction



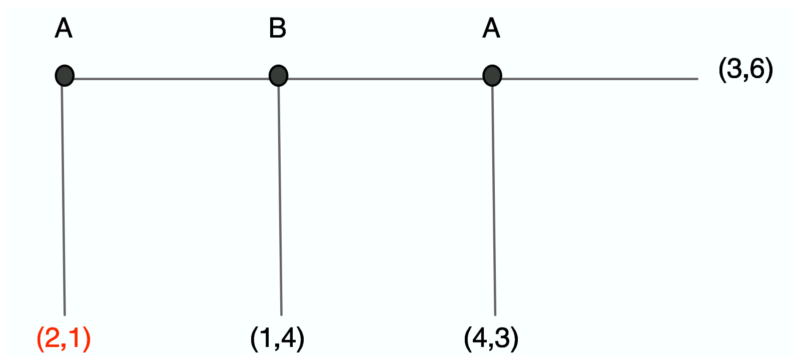
Backward Induction



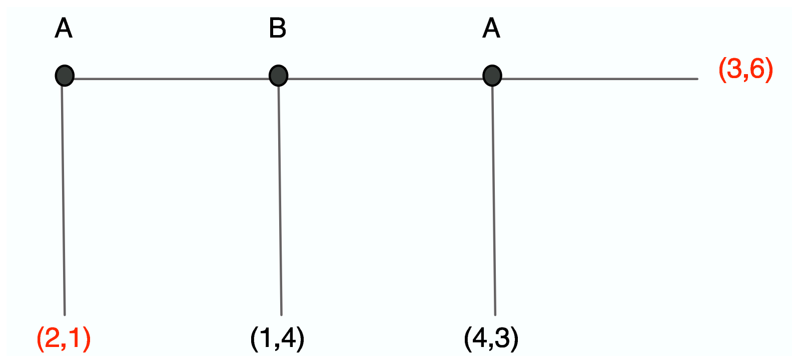
Backward Induction



Backward Induction



Backward Induction



Making a "rational" decision

A Puzzle

Ann believes that Bob assumes* that
Ann believes that Bob's assumption is wrong.

Does Ann believe that Bob's assumption is wrong?

* An **assumption** (or strongest belief) is a belief that implies all other beliefs.

Main Impossibility Result

No belief model can be complete for a language that contains first-order logic

Belief Model: a set of states for each player, and a relation for each player that specifies when a state of one player considers a state of the other player to be possible.

Language: the language used by the players to formulate their beliefs

Complete: A belief model is complete for a language if every statement in a player's langauge which is possible (i.e. true for some states) can be assumed by the player.

Epistemic Program in Game Theory

Belief models and languages are artifacts created by the analyst to describe a strategic situation. *Shouldn't these artifacts be thought of as, in some sense, available to the players themselves?*

- Basic limitation to the analysis of games: *If the analyst's tools are available to the players, there are statements that the players can think about but cannot assume. The model must be incomplete.*

Completeness is very relevant for the epistemic foundations of various solution concepts. (see [Battigalli and Siniscalchi, 2002] and [Brandenburger and Keisler, 2000])

Open Question

Can we find a logic $\mathcal{L}$ such that

1. Complete belief models for $\mathcal{L}$ exist for each game;

2. notions such as rationality, belief in rationality, etc. are expressible in $\mathcal{L}$; and

3. the ingredients in 1 and 2 can be combined to yield various well-known game-theoretic solution concepts.

Belief Models

A **belief model** is a structure

$$\mathcal{M} = (U^a, U^b, P^a, P^b, \ldots)$$

where

- U^a, U^b are nonempty sets

- $P^a \subsetneq U^a \times U^b$ and $P^b \subsetneq U^b \times U^a$

- the following sentences are true (the relations are serial)

$$\forall x \exists y P^a(x, y) \qquad \forall y \exists x P^b(y, x)$$

**x and y range over U^a and U^b respectively.

Beliefs and Assumptions

x **believes** a set $Y \subseteq U^b$ if $\{y \mid P^a(x, y)\} \subseteq Y$

x **assumes** a set $Y \subseteq U^b$ if $\{y \mid P^a(x, y)\} = Y$

Languages

A language for a player should be a set of statements that the player can think about.

Given a belief model $\mathcal{M}$. A **language for** $\mathcal{M}$ is a set $\mathcal{L} = \mathcal{L}^a \cup \mathcal{L}^b$ where $\mathcal{L}^a \subseteq 2^{U^a}$ and $\mathcal{L}^b \subseteq 2^{U^b}$

$\mathcal{M}$ is **complete for** a language $\mathcal{L}$ if each nonempty $Y \in \mathcal{L}^b$ is assumed by some $x \in U^a$, and each nonempty $X \in \mathcal{L}^a$ is assumed by some $y \in U^b$.

Impossibility Results

- (Brandenburger and Keisler, 2005) No belief model is complete for its power-set language

- (Brandenburger and Keisler, 2005), Let $\mathcal{M}$ be a belief model and let $\mathcal{L}$ contain the first-order language for $\mathcal{M}$. Then $\mathcal{M}$ cannot be complete for $\mathcal{L}$.

  – Suppose that in $\mathcal{M}$, $\forall y P^a(x_1, y)$ and

  $$x_2 \;\; believes \;\; [y \;\; believes \;\; [x \;\; believes \, \forall x P^b(y,x)]]$$

  Then x_2 $believes$ $\neg P^b(y, x_2)$

  – Suppose that $\mathcal{M}$ is a belief model. Then no element $x_0 \in U^a$ believes the formula

  $$y \;\; assumes \;\; [x \;\; believes \, \neg P^b(y,x)]$$

For more information on the epistemic program in game theory, see Brandenburger (2005), *The Power of Paradox*.

Conclusions

- A number of examples have been discussed, but no "theory" was presented.

- We have not discussed using logical methods for proving correctness of social procedures (see the work of Parikh, Pauly, van Otterloo, Wooldridge, van der Hoek, etc.)

- For more information on fair division algorithms, see *Fair Division* by Brams and Taylor. Also, the **MARA** group organized in part by U. Endriss.

- See me for a bibliography (epacuit@science.uva.nl)