Pillage Games and Formal Proofs
Past Work, Future Plans

Manfred Kerber[1] Colin Rowat[2]
University of Birmingham
Computer Science Economics

[1]www.cs.bham.ac.uk/~mmk
[2]www.socscistaff.bham.ac.uk/rowat

6 October 2011

Overview

Motivation:

- Proofs in economics use typically undergraduate level proofs
- Proofs in economics are error prone (just as in other theoretical fields)
- Formalization should be achievable
- Automation (or minimization of user interactions) as goal

Overview

Motivation:

- Proofs in economics use typically undergraduate level proofs
- Proofs in economics are error prone (just as in other theoretical fields)
- Formalization should be achievable
- Automation (or minimization of user interactions) as goal

Outline

- Basic Theory
- Pseudo Algorithm
- Examples
- A Lemmas and Theorema
- Plans
- Summary

Power Function

$\mathcal{X} \equiv \{\{x_i\}_{i \in I} | x_i \geq 0, \sum_{i \in I} x_i = 1\}$., the following axioms can be defined. A power function π satisfies

WC if $C \subset C' \subseteq I$ then $\pi(C, \boldsymbol{x}) \leq \pi(C', \boldsymbol{x}) \, \forall \boldsymbol{x} \in \mathcal{X}$;

Power Function

$\mathcal{X} \equiv \{\{x_i\}_{i \in I} | x_i \geq 0, \sum_{i \in I} x_i = 1\}$., the following axioms can be defined. A power function π satisfies

WC if $C \subset C' \subseteq I$ then $\pi(C, \boldsymbol{x}) \leq \pi(C', \boldsymbol{x})\, \forall \boldsymbol{x} \in \mathcal{X}$;

WR if $y_i \geq x_i \forall i \in C \subseteq I$ then $\pi(C, \boldsymbol{y}) \geq \pi(C, \boldsymbol{x})$;

Power Function

$\mathcal{X} \equiv \{\{x_i\}_{i \in I} | x_i \geq 0, \sum_{i \in I} x_i = 1\}$., the following axioms can be defined. A power function π satisfies

WC	if $C \subset C' \subseteq I$ then $\pi(C, \boldsymbol{x}) \leq \pi(C', \boldsymbol{x}) \, \forall \boldsymbol{x} \in \mathcal{X}$;
WR	if $y_i \geq x_i \forall i \in C \subseteq I$ then $\pi(C, \boldsymbol{y}) \geq \pi(C, \boldsymbol{x})$; and
SR	if $\emptyset \neq C \subseteq I$ and $y_i > x_i \forall i \in C$ then $\pi(C, \boldsymbol{y}) > \pi(C, \boldsymbol{x})$.

The Same in Theorema (WC)

WC if $C \subset C' \subseteq I$ then $\pi(C, \boldsymbol{x}) \leq \pi(C', \boldsymbol{x})\ \forall \boldsymbol{x} \in \mathcal{X}$

Definition["WC", any[π, n], bound[allocation$_n$[x]],

$$\mathrm{WC}[\pi, n] :\Leftrightarrow n \in \mathbb{N} \wedge \underset{\substack{C1,C2 \\ C1 \subset C2 \wedge C2 \subseteq I[n]}}{\forall} \ \underset{x}{\forall}\ \pi[C2, x] \geq \pi[C1, x]]$$

]

The Same in Theorema (WR)

WR if $y_i \geq x_i \forall i \in C \subseteq I$ then $\pi(C, \boldsymbol{y}) \geq \pi(C, \boldsymbol{x})$

Definition["WR", any[π, n], bound[allocation$_n$[x], allocation$_n$[y]],

$$\mathrm{WR}[\pi, n] :\Leftrightarrow n \in \mathbb{N} \wedge \Big(\underset{\substack{C \\ C \subseteq I[n]}}{\forall} \underset{x,y}{\forall} \Big(\big(\underset{i \in C}{\forall} y_i \geq x_i \big) \implies \pi[C, y] \geq \pi[C, x] \Big) \Big)$$

]

The Same in Theorema (SR)

SR if $\emptyset \neq C \subseteq I$ and $y_i > x_i \forall i \in C$ then $\pi(C, \boldsymbol{y}) > \pi(C, \boldsymbol{x})$.

Definition["SR", any[π, n], bound[allocation$_n[x]$, allocation$_n[y]$],

$$\mathrm{SR}[\pi, n] :\Leftrightarrow n \in \mathbb{N} \wedge \Big(\underset{\substack{C \\ C \subseteq I[n] \wedge C \neq \emptyset}}{\forall} \; \underset{x,y}{\forall} \big(\big(\underset{i \in C}{\forall} y_i > x_i \big) \Longrightarrow \pi[C, y] > \pi[C, x] \big) \Big) \Big]$$

Properties

Other important properties that power functions may have:

- **AN** if $\sigma : I \to I$ is a 1:1 onto function permuting the agent set, $i \in C \Leftrightarrow \sigma(i) \in C'$, and $x_i = x'_{\sigma(i)}$ then $\pi(C, \boldsymbol{x}) = \pi(C', \boldsymbol{x}')$.
- **CX** $\pi(C, \boldsymbol{x})$ is continuous in $\boldsymbol{x}$.
- **RE** if $i \notin C$ and $\pi(\{i\}, \boldsymbol{x}) > 0$ then $\pi(C \cup \{i\}, \boldsymbol{x}) > \pi(C, \boldsymbol{x})$.

Domination

Def$_{\succ\!\!-}$ An allocation $\boldsymbol{y}$ dominates an allocation $\boldsymbol{x}$, written $\boldsymbol{y} \succ\!\!- \boldsymbol{x}$, iff $\pi(W, \boldsymbol{x}) > \pi(L, \boldsymbol{x})$; where $W \equiv \{i \,|\, y_i > x_i\}$ and $L \equiv \{i \,|\, x_i > y_i\}$. W = win set & L lose set.

Def$_D$ For $\mathcal{Y} \subset \mathcal{X}$, let $D(\mathcal{Y}) \equiv \{\boldsymbol{x} \in \mathcal{X} \,|\, \exists \boldsymbol{y} \in \mathcal{Y} \text{ s.t. } \boldsymbol{y} \succ\!\!- \boldsymbol{x}\}$ be the dominion of $\mathcal{Y}$. $U(\mathcal{Y}) = \mathcal{X} \backslash D(\mathcal{Y})$, the set of allocations undominated by any allocation in $\mathcal{Y}$.

Core and stable set

Def$_{\mathcal{K}}$ The core, $\mathcal{K}$, is the set of undominated allocations, $U(\mathcal{X})$.

Def$_{\mathcal{S}}$ A set of allocations, $\mathcal{S} \subseteq \mathcal{X}$, is a stable set iff it satisfies

internal stability,	$\mathcal{S} \cap D(\mathcal{S}) = \emptyset$	(IS)
external stability,	$\mathcal{S} \cup D(\mathcal{S}) = \mathcal{X}$	(ES)

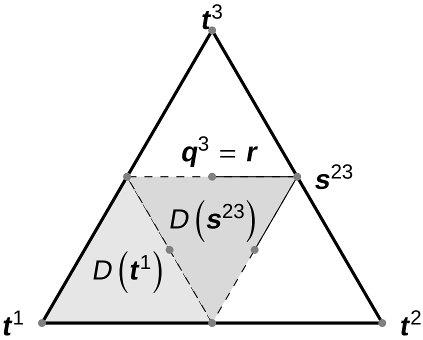
The conditions combine to yield $\mathcal{S} = \mathcal{X} \backslash D(\mathcal{S})$. The core necessarily belongs to any existing stable set.

Wealth Is Power

$$\mathrm{WIP}\pi[C, x] := \sum_{i \in C} x_i$$

Wealth Is Power

$$\mathrm{WIP}\pi[C, x] := \sum_{i \in C} x_i$$

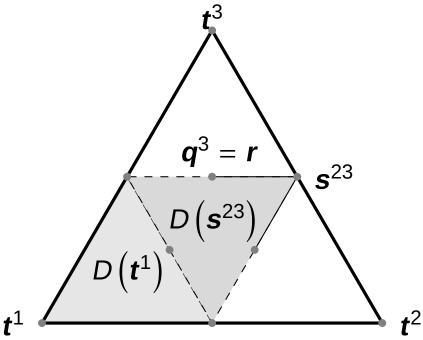


Wealth Is Power

$$\mathrm{WIP}\pi[C,x] := \sum_{i\in C} x_i$$

Stable Set: $S =$

$$\left\{ \begin{array}{l} (0,0,1),(0,1,0),(1,0,0), \\ (0,\frac{1}{2},\frac{1}{2}),(\frac{1}{2},0,\frac{1}{2}),(\frac{1}{2},\frac{1}{2},0), \\ (\frac{1}{4},\frac{1}{4},\frac{1}{2}),(\frac{1}{4},\frac{1}{2},\frac{1}{4}),(\frac{1}{2},\frac{1}{4},\frac{1}{4}), \end{array} \right\}$$



The stable set in $n = 3$ with AN, CX, and RE

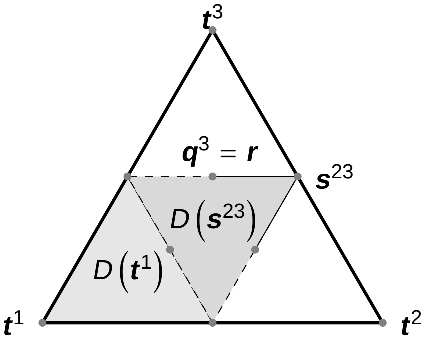
if $\pi\left(\{i\}, \mathbf{t}^i\right) < \pi\left(\{j,k\}, \mathbf{t}^i\right)$
    then $\mathcal{S} = \mathcal{D}_1 \setminus \mathcal{D}_0$
    else
        if $R^i = \emptyset$
            then return "no stable set exists"
            else
                if $\pi\left(\{j\}, \mathbf{s}^{jk}\right) \geq \pi\left(\{i,k\}, \mathbf{s}^{jk}\right)$
                    then $\mathcal{S} = \mathcal{D}_1 \cup \left\{\mathcal{S}^i\right\}_{i=1}^3$
                    else
                        $\mathcal{S} = \mathcal{D}_0 \cup \left\{\mathcal{S}^i\right\}_{i=1}^3 \cup \mathcal{P}$
                end if
        end if
end if
return $\mathcal{S}$

Wealth Is Power

$$\mathrm{WIP}\pi[C, x] := \sum_{i \in C} x_i$$

Stable Set: $S =$

$$\left\{ \begin{array}{l} (0,0,1),(0,1,0),(1,0,0), \\ (0,\frac{1}{2},\frac{1}{2}),(\frac{1}{2},0,\frac{1}{2}),(\frac{1}{2},\frac{1}{2},0), \\ (\frac{1}{4},\frac{1}{4},\frac{1}{2}),(\frac{1}{4},\frac{1}{2},\frac{1}{4}),(\frac{1}{2},\frac{1}{4},\frac{1}{4}), \end{array} \right\}$$



Strength In Numbers with $\nu > 1$

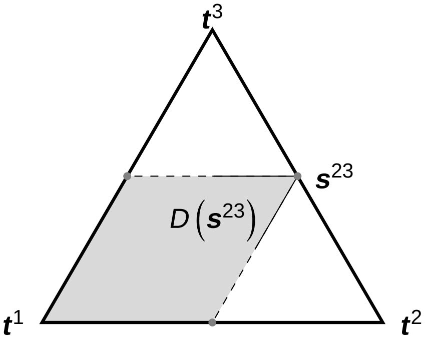
$$\mathrm{SIN}\pi_\nu[C, x] := \sum_{i \in C} (x_i + \nu)$$

with $\nu > 1$

Strength In Numbers with $\nu > 1$

$$\mathrm{SIN}\pi_\nu[C, x] := \sum_{i \in C} (x_i + \nu)$$

with $\nu > 1$

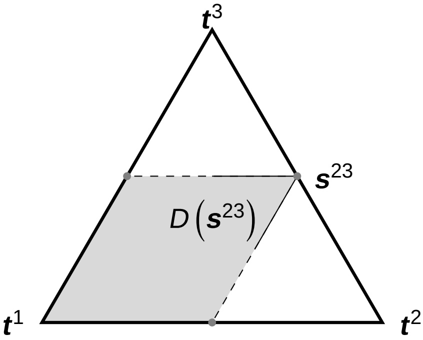


Strength In Numbers with $\nu > 1$

$$\mathrm{SIN}\pi_\nu[C, x] := \sum_{i \in C} (x_i + \nu)$$

with $\nu > 1$

Stable Set: $S = \left\{(0, \frac{1}{2}, \frac{1}{2}), (\frac{1}{2}, 0, \frac{1}{2}), (\frac{1}{2}, \frac{1}{2}, 0)\right\}$

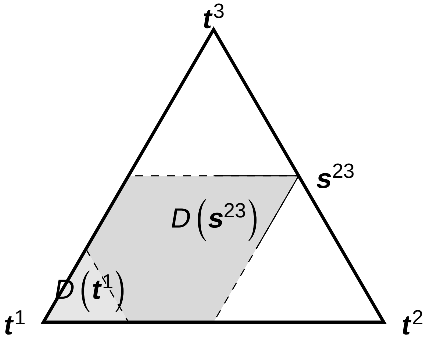


Strength In Numbers with $0 < \nu < 1$

$$\mathrm{SIN}\pi_\nu[C, x] := \sum_{i \in C} (x_i + \nu)$$

with $0 < \nu < 1$

no stable set exists



Proof of a Lemma

(One Lemma of 14 lemmas, 12 theorems, and 4 corollaries)

Lemma

When $n = 3$: 1. $\mathcal{K} = \emptyset$ *implies* $\mathbf{t}^i \in D\left(\mathbf{s}^{jk}\right)$ *for distinct* $i, j, k \in I$.

Proof.

1. As $\mathcal{K} = \emptyset$, no agent can defend its holdings against both others, so that $\pi\left(\{i\}, \mathbf{t}^i\right) < \pi\left(\{j,k\}, \mathbf{t}^i\right)$ for distinct i, j and k. As $\{j,k\}$ prefers $\boldsymbol{s}^{jk}$ to $\boldsymbol{t}^i$, this ensures that $\boldsymbol{s}^{jk} \mathrel{\varepsilon\!\!-} \boldsymbol{t}^i$.

$\square$

Representation and Proof

- sTEX a semantic version of TEX. (Do not use macros of the type `\mathbb{N}` but of type `\naturalNumbers`).
- Prove all theorems in Theorema and Isabelle.
- Extract challenge problems for higher order theorem proving (to be represented in TPTP).

Representation and Proof

- sTEX a semantic version of TEX. (Do not use macros of the type `\mathbb{N}` but of type `\naturalNumbers`).
- Prove all theorems in Theorema and Isabelle.
- Extract challenge problems for higher order theorem proving (to be represented in TPTP).
- Extend study to games not satsifying additional restrictions.

Representation and Proof

- sTEX a semantic version of TEX. (Do not use macros of the type `\mathbb{N}` but of type `\naturalNumbers`).
- Prove all theorems in Theorema and Isabelle.
- Extract challenge problems for higher order theorem proving (to be represented in TPTP).
- Extend study to games not satsifying additional restrictions.
- Test higher order theorem provers (Leo).
- Extract proof tactics.
- Reuse proofs tactics.
- Extract computational content.
- Guide proofs by computational models.

Representation and Proof

- sTEX a semantic version of TEX. (Do not use macros of the type `\mathbb{N}` but of type `\naturalNumbers`).
- Prove all theorems in Theorema and Isabelle.
- Extract challenge problems for higher order theorem proving (to be represented in TPTP).
- Extend study to games not satsifying additional restrictions.
- Test higher order theorem provers (Leo).
- Extract proof tactics.
- Reuse proofs tactics.
- Extract computational content.
- Guide proofs by computational models.
- Exploit results.

Summary (Part I)

The pseudo algorithm:

- Non-computational in several aspects
- Evaluation by a mixture of reasoning and computing. Can compute the stable set of WIP, SIN, assumed the corresponding lemmas are available.
- Plan: Extend the computational part, e.g., represent infinite set in a finite way. Use underlying Mathematica to compute solutions of equations.

Summary (Part II)

- Axiomatic approach in theoretical economics valuable (eliminate errors, even without full proof)
- Good field with non-trivial but not very deep mathematics.
- Formalisation in Theorema is easy and fast even for beginners.
- Automation at least partially possible. Reasoning requires more expert knowledge and work.
- Theorema offers mixture of reasoning and computation. Very useful for determining stable sets.