Verifying Auctions

Manfred Kerber Marco Caminati
Christoph Lange Colin Rowat

University of Birmingham
Computer Science Economics
www.cs.bham.ac.uk/research/projects/formare/

Munich, 8 Jan 2016

supported by
EPSRC grant EP/J007498/1 and
LMS Computer Science Small Grant

Overview

Motivation:

- Proofs in economics use often undergraduate level maths
- Proofs in economics are error prone (just as in any other theoretical fields)
- Formalization should be achievable – not just for computer scientists, but also for economists (?)
- Understand problems with the usage of theorem proving systems (!)

Overview

Motivation:

- Proofs in economics use often undergraduate level maths
- Proofs in economics are error prone (just as in any other theoretical fields)
- Formalization should be achievable – not just for computer scientists, but also for economists (?)
- Understand problems with the usage of theorem proving systems (!)

Outline

- Related Work
- Pillage games
- Auction theory
- Open questions
- Summary

Some Related Work

Arrow's impossibility theorem

A constitution respects $\mathtt{UN}$ *if society puts alternative* a *strictly above* b *whenever every individual puts* a *strictly above* b*. The constitution respects* $\mathtt{IIA}$ *if the social relative ranking (higher, lower, or indifferent) of two alternatives* a *and* b *depends only on their relative ranking by every individual. The constitution is a* $\mathtt{D}$ *by individual* n *if for every pair* a *and* b*, society strictly prefers* a *to* b *whenever* n *strictly prefers* a *to* b*. [Geanakoplos 05]*

Arrow's impossibility theorem

A constitution respects $\mathtt{UN}$ if society puts alternative a strictly above b whenever every individual puts a strictly above b. The constitution respects $\mathtt{IIA}$ if the social relative ranking (higher, lower, or indifferent) of two alternatives a and b depends only on their relative ranking by every individual. The constitution is a $\mathtt{D}$ by individual n if for every pair a and b, society strictly prefers a to b whenever n strictly prefers a to b. [Geanakoplos 05]

Theorem (Arrow – 3 Proofs by Geanakoplos 2005)

(For two or more agents, and three or more alternatives,) any constitution that respects transitivity, $\mathtt{IIA}$, and $\mathtt{UN}$ is a $\mathtt{D}$.

Arrow's impossibility theorem (Cont'd)

- *"Social choice theory turns out to be perfectly suitable for mechanical theorem proving. . . . However, it is unclear if this will lead to new insights into either social choice theory or theorem proving."* [Nipkow09]

Arrow's impossibility theorem (Cont'd)

- *"Social choice theory turns out to be perfectly suitable for mechanical theorem proving. . . . However, it is unclear if this will lead to new insights into either social choice theory or theorem proving."* [Nipkow09]
- *"we form an interesting conjecture and then prove it using the same [mechanized] techniques as in the previous proofs. . . . the newly proved theorem . . . subsumes both Arrow's and Wilson's theorems."* [Tang-Lin09]

Arrow's impossibility theorem (Cont'd)

- *"Social choice theory turns out to be perfectly suitable for mechanical theorem proving. . . . However, it is unclear if this will lead to new insights into either social choice theory or theorem proving."* [Nipkow09]
- *"we form an interesting conjecture and then prove it using the same [mechanized] techniques as in the previous proofs. . . . the newly proved theorem . . . subsumes both Arrow's and Wilson's theorems."* [Tang-Lin09]
- *"When applied to a space of 20 principles for preference extension familiar from the literature, this method yields a total of 84 impossibility theorems, including both known and nontrivial new results."* [Geist-Endress-11]

References

Gea01 John D. Geanakoplos. Three brief proofs of Arrow's impossibility theorem. Discussion Paper 1123RRR. New Haven: Cowles Foundation, 2001.

Gea05 John D. Geanakoplos. "Three brief proofs of Arrow's impossibility theorem". In: Economic Theory 26.1 (2005), pp. 211–215.

Nip09 Tobias Nipkow. "Social choice theory in HOL: Arrow and Gibbard-Satterthwaite". In: Journal of Automated Reasoning 43.3 (2009), pp. 289–304.

Wie07 Freek Wiedijk. "Arrow's impossibility theorem". In: Journal of Formalized Mathematics 15.4 (2007), pp. 171–174.

Wie09 Freek Wiedijk. "Formalizing Arrow's theorem". In: Sādhanā 34.1 (2009), pp. 193–220.

References (Cont'd)

TaLi09 Pingzhong Tang and Fangzhen Lin. "Computer-aided proofs of Arrow's and other impossibility theorems". In: Artificial Intelligence 173.11 (2009), pp. 1041–1053.

GrEn09 Umberto Grandi and Ulle Endriss. "First-Order Logic Formalisation of Arrow's Theorem". In: Proceedings of the 2nd International Workshop on Logic, Rationality and Interaction (LORI-2009). Ed. by X. He, J. Horty, and E. Pacuit. Lecture Notes in Artificial Intelligence 5834. Springer, 2009, pp. 133–146.

GeEn11 Christian Geist and Ulle Endriss. "Automated search for impossibility theorems in social choice theory: ranking sets of objects". In: Journal of Artificial Intelligence Research 40 (2011), pp. 143–174.

References (Cont'd)

AgHoWo09 Thomas Ågotnes, Wiebe van der Hoek, and Michael Wooldridge. "Reasoning about coalitional games". In: Artificial Intelligence 173.1 (2009), pp. 45–79.

VeLeOn06 René Vestergaard, Pierre Lescanne, and Hiroakira Ono. The inductive and modal proof of Aumann's theorem on rationality. Technical Report IS-RR-2006-009. Japan Advanced Institute of Science and Technology, 2006.

WeDeFi09 M. P. Webster, L. Dennis, and M. Fisher. Model-checking auctions, coalitions and trust. Tech. Rep. ULCS-09-004, Computer Science, Univ. of Liverpool, 2009.

Pillage Games

Pillage Games

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

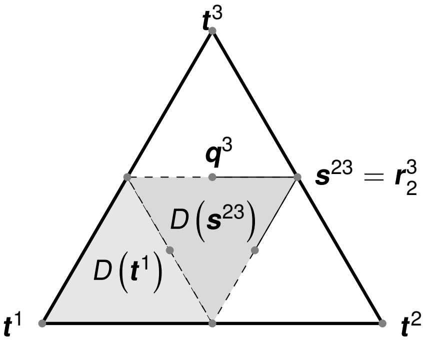
WC (weak coalition monotonicity)
if $C \subset C' \subseteq I$ then $\pi(C, \boldsymbol{x}) \leq \pi(C', \boldsymbol{x}) \, \forall \boldsymbol{x} \in \mathcal{X}$;

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

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

Wealth Is Power

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

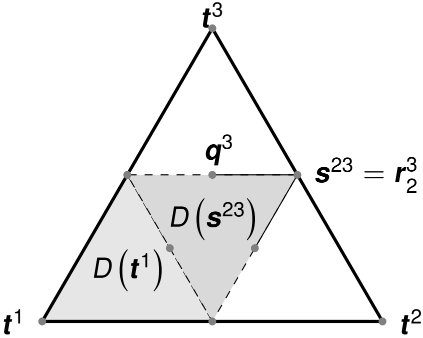


Wealth Is Power

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

Stable Set: $S =$

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



Some Results

Formalization: Theorema I. Represent the main definitions and results

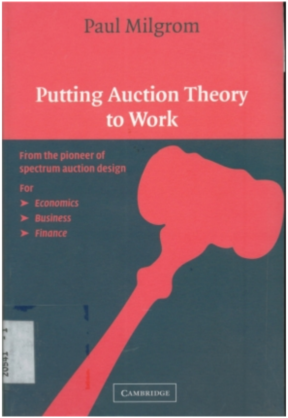
Proofs: Prove some theorems in Theorema

Pseudo Algorithm: Summarize the results in a Theorema algorithm with oracle, where the oracle is given by lemmas which can be proved in Theorema.

Presentation at ICE 2012 (Initiative for Computational Economics, ice.uchicago.edu/) $\rightsquigarrow$ look into other areas.

We organized a symposium at this year's AISB convention on Do-Form: Enabling Domain Experts to use Formalised Reasoning
www.cs.bham.ac.uk/research/projects/formare/events/aisb2013

Auctions


Paul Milgrom
Putting Auction Theory to Work
From the pioneer of spectrum auction design
For
➤ Economics
➤ Business
➤ Finance
CAMBRIDGE

Auctions

Auctions

Auctions allocate trillions of dollars in goods and services every year.

Auctions

Auctions

Auctions allocate trillions of dollars in goods and services every year.

Auctions are a mechanism to distribute resources (e.g., eBay, ICANN, possibly High-Frequency Trading [Peter Cramton])

Auctions

Auctions

Auctions allocate trillions of dollars in goods and services every year.

Auctions are a mechanism to distribute resources (e.g., eBay, ICANN, possibly High-Frequency Trading [Peter Cramton])

Given: a set of individual bids for a good (not necessarily the same as the *value* an individual ascribes to the good!)

Goals:
- give the good to the bidder who values it most
- determine prices
- maximize revenue

Auctions

Auctions

Auctions allocate trillions of dollars in goods and services every year.

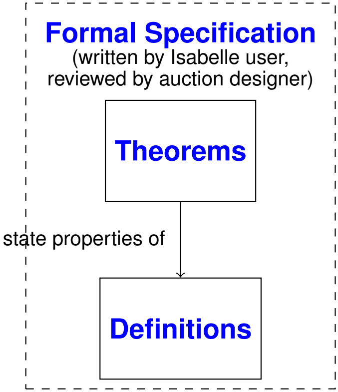
Auctions are a mechanism to distribute resources (e.g., eBay, ICANN, possibly High-Frequency Trading [Peter Cramton])

Given: a set of individual bids for a good (not necessarily the same as the *value* an individual ascribes to the good!)

Goals:
- give the good to the bidder who values it most
- determine prices
- maximize revenue

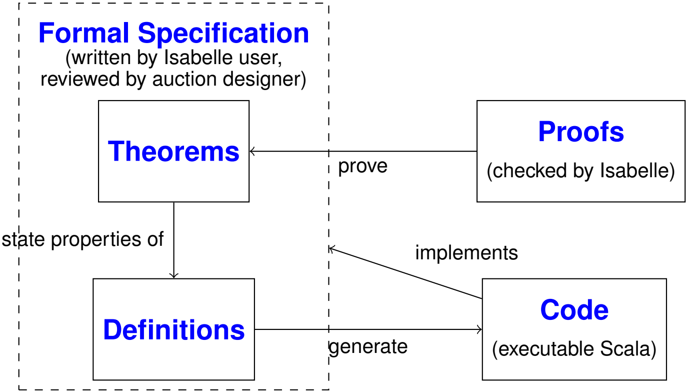
- New auctions are designed and some properties are proved.
- Strict rules must be followed.
- New auctions may have problems.

From Specifications to Software



High-level outline of our approach

From Specifications to Software



High-level outline of our approach

Vickrey's Theorem

Second-price auction: a highest bidder wins, pays highest *remaining* bid.

Theorem (Vickrey 1961)

In a second-price auction, "truth-telling" (i.e. submitting a bid equal to one's actual valuation of the good) is a weakly dominant strategy. The auction is efficient.

- earliest result in modern auction theory
- simple environment in which to gain intuitions

Weakly Dominant Strategy

A definition

Given some auction, a strategy profile $\boldsymbol{b}$ supports an equilibrium in weakly dominant strategies if, for each $i \in N$ and any $\hat{\boldsymbol{b}} \in \mathbb{R}^n$ with $\hat{b}_i \neq b_i$, $u_i\left(\hat{b}_1, \ldots, \hat{b}_{i-1}, b_i, \hat{b}_{i+1}, \ldots, \hat{b}_n\right) \geq u_i\left(\hat{\boldsymbol{b}}\right)$. I.e., whatever others do, i will not be better off by deviating from the original bid b_i.

VCG Mechanism to Auction a Set of Goods

- Extension of a single good auction to a simultaneous auction of a set of goods (by Vickrey, Clarke, Groves), a so-called ‘combinatorial’ auction.
- Users can bid on any (non-empty) subset.
- Second price mechanism.
- Computationally expensive

An Impression of the Type of Maths Involved

$$X^* \in \operatorname*{arg\,max}_{X_1,\dots,X_N} \sum_{n=1}^{N} b_n(X_n) \text{ s.t. } \bigcup_{n=1}^{N} X_n \subseteq \Omega \text{ and } X_n \cap X_{n'} = \emptyset \text{ for } n \neq n' \quad (1)$$

at prices

$$p_n \equiv \alpha_n - \sum_{m \neq n} b_m(X_m^*) \quad (2)$$

where

$$\alpha_n \equiv \max_{\substack{X_m \\ m=1,\dots,N, m\neq n}} \left\{ \sum_{m \neq n} b_m(X_m) \;\middle|\; \bigcup_{m \neq n} X_m \subseteq \Omega \text{ and } X_m \cap X_{m'} = \emptyset \text{ for } m \neq m' \right\} \quad (3)$$

Experiments with 4 Systems for Single Good Auctions

1. **Isabelle/HOL** (with Markarius Wenzel): higher-order logic (typed), interactive theorem proving environment, document-oriented IDE
2. **Theorema 2.0** (with Wolfgang Windsteiger): FOL + set theory, textbook-style documents (Mathematica notebooks), proof management GUI
3. **Mizar** (Marco Caminati): FOL + set theory, text editor, proof checker
4. **Hets/CASL/TPTP** (with Till Mossakowski): sorted FOL, text editor, proof management GUI, front-end to local or remote automated provers

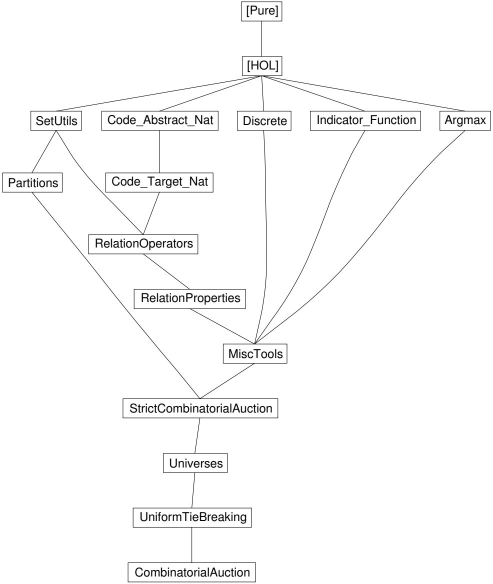
Choice for Isabelle

- Strong user community
- Extended library
- In-built automatic reasoners

Choice for Isabelle

- Strong user community
- Extended library
- In-built automatic reasoners
- Code extraction mechanism: Code extraction is a major selling point, since auction designers do not doubt their theorems.

Theory graph generated by Isabelle



An Isabelle Code snippet

The set of maximal allocations:

```
abbreviation "vcgas N G b r  == Outside' {seller} `
  ((argmax∘setsum) (randomBids' N G b r)
   ((argmax∘setsum) b (allAllocations (N∪{seller}) (set G)))
```

The unique winning allocation after tie breaking.

```
abbreviation "vcga N G b r == the_elem (vcgas N G b r)"
```

Classical vs Constructive Mathematics

Two possibilities to concepts such as injections:

- Classical: The set of all injections from a set A to a set B can be defined as the set of all relations R with domain A and a range that is a subset of B such that R and R^{-1} are right-unique.

Classical vs Constructive Mathematics

Two possibilities to concepts such as injections:

- Classical: The set of all injections from a set A to a set B can be defined as the set of all relations R with domain A and a range that is a subset of B such that R and R^{-1} are right-unique.
- Constructive: Give a recursive definition (for finite sets):

  Base case: The injections from the empty set to an arbitrary set consists of the empty relation.

  Step case: Assume all injections from A to B given. Take an additional element a not yet in A, extend mappings so that they map a to an element that is not in the range of A.

Isabelle

A flavour of Isabelle:

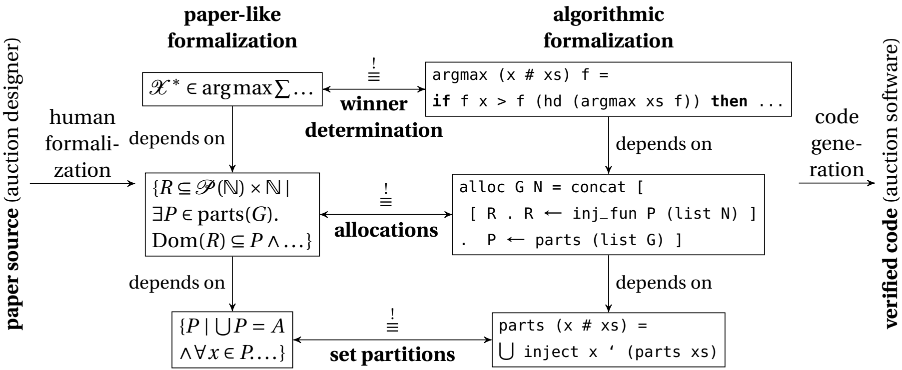
```
definition injections ::
   "'a set ⇒ 'b set ⇒ ('a × 'b) set set"
where "injections X Y =
    {R . Domain R = X ∧
         Range R ⊆ Y ∧
         runiq R ∧
         runiq (R¯)}"
```

Isabelle (Cont'd)

```
fun injections_alg ::
    "'a list ⇒ 'b::linorder set ⇒ ('a × 'b) set list"
where "injections_alg [] Y = [{}]" |
      "injections_alg (x # xs) Y =
    concat [ [ R +* {(x,y)} .
                y ← sorted_list_of_set (Y - Range R) ]
              . R ← injections_alg xs Y ]"
```

Why two versions and how to reconcile

- The classical definitions are closer to standard mathematical descriptions (more like in a textbook) and in consequence easier for auction designers to check.
- We need the constructive definition since only they allow to extract (Scala) code.



Bridging Theorem

Solution: Have classical plus constructive definitions and show bridging theorems.

```
theorem injections_equiv:
  assumes "finite Y" and "distinct X"
  shows  "set (injections_alg X Y) = injections (set X) Y"
proof -
  let ?P="\<lambda> l. distinct l \<longrightarrow>
          (set (injections_alg l Y)=injections (set l) Y)"
  have "?P []" using injectionsFromEmptyAreEmpty list.set(1)
       lm099 by metis
  moreover have "\<forall>x xs. ?P xs \<longrightarrow> ?P (x#xs)"
    using assms(1) lm101 by
    (metis distinct.simps(2) insert_is_Un list.simps(15))
  ultimately have "?P X" by (rule structInduct)
  then show ?thesis using assms(2) by blast
qed
```

file:///home/mmk/research/formare/code/auction/isabelle/Auction/Vcg/afp/Vickrey_Clarke_Groves/Universes.thy (line 1221 ff)

Theorems Proved

1. VCG auctions are functions.
2. VCG allocations are pairwise disjoint.
3. In VCG allocations only goods in the auction are allocated.
4. Prices in VCG auctions are non-negative.

Adaptation from Second Price to First Price

1. The Winner Determination Problem (WDP) is the same.
2. The prices are much easier to establish (everybody just pays their bid) so although the original proofs do not go through, Isabelle can find the proofs itself.

```
abbreviation "firstPriceP N \<Omega> b r n ==
 b (n, winningAllocationAlg N \<Omega> r b,, n)"
```

Code Generation into Scala

The code can be extracted by an Isabelle command

```
export_code ...  in Scala module_name VCG file "file.scala"
```

Distinguish:

- an interface (wrapper) for I/O (hand written) vs
- trusted code extracted via Isabelle for the computational part.

Example:

file:///home/mmk/research/formare/code/auction/scala/addedWrapper.scala

Tentative Lessons

- Representation is non-trivial, since it is partly not easy to understand the theorems, partly it is easy to make mistakes.
- Find mistakes by use and proof.
- Notice hidden assumptions
- Often proofs that look simple, are still non-trivial for theorem provers.
- First rationalize proofs.
- HOL vs FOL, automated vs interactive ATPs differences are not that relevant after all (but the complexity of the argument).

Further Work – Open Questions

- Extend to dynamic auctions
- Implement efficient algorithms for combinatorial auctions
- Adapt to modern auctions
- Get back to auction designers

Summary

- Computer Science in general, Theorem Proving in special can support economists.
- We have to work towards adjusting our methods to economics problems.

Summary

- Computer Science in general, Theorem Proving in special can support economists.
- We have to work towards adjusting our methods to economics problems.
- Specialist knowledge is required, the systems in its current form are still difficult to use by non-experts.

Summary

- Computer Science in general, Theorem Proving in special can support economists.
- We have to work towards adjusting our methods to economics problems.
- Specialist knowledge is required, the systems in its current form are still difficult to use by non-experts. Surprisingly this is no real problem. Auction designers cooperate with other experts such as lawyers. The real problem is to convince them of the usefulness of the general approach!
- There are many challenging problems. Further info: www.cs.bham.ac.uk/research/projects/formare/