

On Vickrey’s Theorem and the Use of ACL2 for Formal Reasoning in Economics (Extended Abstract)

Ruben Gamboa

University of Wyoming
Laramie, WY, USA
ruben@uwyo.edu

John Cowles

University of Wyoming
Laramie, WY, USA
cowles@uwyo.edu

The ForMaRE project (**f**ormal **m**athematical **r**easoning in **e**conomics) is an interdisciplinary effort with the goal of increasing confidence in theoretical results in economics [3]. In a nutshell, the idea is to find suitable automated reasoning platforms for reasoning about economic theory. This requires collaboration among economists and formal methodists, but in an ideal world, the automated reasoning platforms will become sufficiently advanced that economists can use them directly. As such, researchers in the ForMaRE project recently analyzed the suitability of four different automated reasoning platforms for this project: Isabelle, Theorema, Mizar, and Hets/CASL/TPTP. A major portion of the analysis was the formalization in each of these systems of Vickrey’s theorem on the equilibrium and efficiency of second-price auctions. Conspicuously absent from the list of theorem provers considered is ACL2. This paper describes our attempt to rectify this oversight.

The theorem that was chosen to evaluate the different platforms is Vickrey’s auction theorem. In an auction, there is some good that has only subjective value—i.e., different people will assign different values to the good. Each of a number of bidders will place a (silent) bid for the good, and one of the bidders (perhaps chosen at random) who placed the highest bid will win the good. Vickrey’s theorem states that for each bidder, bidding the value they place on the good is an optimal strategy.

This can be formalized in ACL2 as follows. First of all, there are N bidders, so all the information about each bidder can be stored in a list. Each bidder’s subjective value for the good is stored in a list henceforth called `value`. An actual bid consists of a list that has each bidder’s bid for the good.

The utility of a bid is the sum of each bidder’s utility. A bidder who loses the bid has a utility of zero—they did not win the good, but on the other hand, they did not pay any money. A bidder who wins the bid has a utility equal to $v_i - \max_{j \neq i} b_j$ where v_i is the value they assign to the good and b_j is the bid made by bidder j . Note that the utility is not just $v_i - b_i$. The bidder does not pay his or her own bid. Instead, the bidder pays the highest bid made by somebody else, which may be lower than his or her bid. This “second-price” style auction results in higher values for the owner of the good, since it disincentivizes bidders from low-balling the bid in the hope that everyone else is low-balling, too.

In ACL2, the winner of a bid can be selected using an `encapsulate`. The only constraint is that the winner must be one of the bidders, and the winner’s bid must be the maximum. Our local witness chooses the first such bidder, but the winner could be picked at random among the highest bids.

The theorem claims that the optimal strategy for a bidder i is to bid the value they place on the good. Suppose this is the case for bidder i in bid b . Now consider some other bid b' that is equal to b except that bidder i places a different bid. We wish to show that the utility of b' is no better than the utility of b .

Our proof is based on the outline given in [3]. It considers four (similar) cases, depending on whether bidder i wins or loses the auction with bids b or b' . If bidder i wins both or loses both, then the utility is the same. The reason is that bidder i assigns the same value to the good, and bidder i will pay the same

amount for the good, since the amount is determined by the non-winning bids, which are the same in b and b' . If bidder i loses bid b but wins b' , then the utility of b is zero (since bidder i did not win that bid), the utility of b' is non-negative. This is because the price paid is the second-highest bid, which must be less than bidder i 's bid, i.e., his or her value of the good. Finally, if bidder i wins bid b but loses b' , then the utility of b' is zero, but the utility of b may be greater, since it's the difference between bidder i 's value of the good and the second-highest bid (which can be no higher than i 's bid, equal to this value).

ACL2 can prove all four cases, after suitable lemmas concerning the largest possible value of the second-highest bid are introduced. But that raises the question, is ACL2 a suitable platform for reasoning about economics? The leaders of the ForMaRE consider various criteria to compare different systems.

The first criteria is the level of detail and explicitness required. In our case, we had to provide explicit definitions for concepts such as “second-highest bid”, then prove the usual lemmas about them. We suspect this is comparable to other provers.

As far as expressiveness versus efficiency, ACL2 strikes a nice balance. The restriction to first-order logic was not a factor in this formalization (though it may be for others). A more serious objection is the lack of randomness. We sidestepped this issue here with an `encapsulate`, but this approach is unlikely to scale to more complex economic scenarios.

The ForMaRE team also assessed proof development and management of the various tools. The biggest assistance they identified was the ability to defer proofs of lemmas until later, which ACL2 does with `skip-proofs`. They also found that using external tools made the proofs easier. It may be that ACL2's connections to SAT solvers, for example, will prove useful in a larger effort to verify theorems from economics.

One aspect where ACL2 does not fare well is in term input syntax. The economists found Theorema's syntax (basically Mathematica's) to be the most helpful. The economists may not like Common LISP.

Another interesting aspect is comprehensibility and trustability of the output. To illustrate this issue, we can share an anecdote. The proof of the first case outlined above was easy enough that we were only mildly surprised that ACL2 was able to do it automatically. But when ACL2 proved all four cases automatically, we became suspicious. The “problem” turned out to a bug in the definition of `valid-bid-p`. Efforts that blend theorem proving with test cases, as in `DoubleCheck` or `ACL2(s)`, would help to solve this problem [2, 1].

In conclusion, ACL2 is more than capable of proving theorems in the domain of economics, and we think this is an exciting new application domain for formal reasoning. Certainly, ACL2 is at least as good for this application as the systems considered by the ForMaRE leaders. Our only concern is that the lack of randomness in ACL2 will pose an unsurmountable challenge to the formalization of deep results in economics, which is something we plan to pursue in the future.

References

- [1] P. Dillinger, P. Manolios, J. S. Moore & D. Vroon (2007): *ACL2s: The ACL2 Sedan*. In: *Proceedings of the 29th International Conference on Software Engineering, Research Demonstration Track*, doi:10.1016/j.entcs.2006.09.018
- [2] C. Eastlund (2009): *DoubleCheck your theorems*. In: *Proceedings of the Eighth International Workshop on the ACL2 Theorem Prover and its Applications (ACL2-09)*.
- [3] Christoph Lange, Colin Rowat & Manfred Kerber (2013): *The ForMaRE Project Formal Mathematical Reasoning in Economics*. In Jacques Carette, David Aspinall, Christoph Lange, Petr Sojka & Wolfgang Windsteiger, editors: *Intelligent Computer Mathematics, Lecture Notes in Computer Science 7961*, Springer Berlin Heidelberg, pp. 330–334, doi:10.1007/978-3-642-39320-4_23.