

TheoryGuru: A Mathematica Package to Apply Quantifier Elimination Technology to Economics

Mulligan, C, Davenport, JH & England, M

Author post-print (accepted) deposited by Coventry University's Repository

Original citation & hyperlink:

Mulligan, C, Davenport, JH & England, M 2018, TheoryGuru: A Mathematica Package to Apply Quantifier Elimination Technology to Economics. in Mathematical Software: Proceedings of the International Congress on Mathematical Software (ICMS 2018). Lecture Notes in Computer Science, vol. 10931, Springer, pp. 369-378.

https://dx.doi.org/10.1007/978-3-319-96418-8_44

DOI 10.1007/978-3-319-96418-8_44

ISSN 0302-9743

Publisher: Springer

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-96418-8_44

Copyright © and Moral Rights are retained by the author(s) and/ or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This item cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder(s). The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

This document is the author's post-print version, incorporating any revisions agreed during the peer-review process. Some differences between the published version and this version may remain and you are advised to consult the published version if you wish to cite from it.

TheoryGuru: A Mathematica Package to apply Quantifier Elimination Technology to Economics

Casey B. Mulligan¹, James H. Davenport², and Matthew England³

¹ University of Chicago, USA

`c-mulligan@uchicago.edu`

² University of Bath, UK

`J.H.Davenport@bath.ac.uk`

³ Coventry University, UK

`Matthew.England@coventry.ac.uk`

Abstract. We consider the use of Quantifier Elimination (QE) technology for automated reasoning in economics. There is a great body of work considering QE applications in science and engineering but we demonstrate here that it also has use in the social sciences. We explain how many suggested theorems in economics could either be proven, or even have their hypotheses shown to be inconsistent, automatically via QE. However, economists who this technology could benefit are usually unfamiliar with QE, and the use of mathematical software generally. This motivated the development of a MATHEMATICA Package **TheoryGuru**, whose purpose is to lower the costs of applying QE to economics. We describe the package's functionality and give examples of its use.

Keywords: Quantifier Elimination, Economics Reasoning

1 Introduction

A general task in economic reasoning is to determine whether, with variables $\mathbf{v} = (v_1, \dots, v_n)$, a hypothesis $H(\mathbf{v})$ follows from assumptions $A(\mathbf{v})$, i.e. is it the case that $\forall \mathbf{v}. A \Rightarrow H$? Ideally the answer would be **True** or **False**, but in practice life is more complicated: the answer could differ depending on the value of \mathbf{v} ; or the assumptions could even be contradictory, i.e. $A(\mathbf{v})$ alone is **False**.

We can categorise these possibilities via the outcome of a pair of quantified statements (Table 1). Should technology provide any one automatically then an economist gains important information: either a proof or a disproof of their theory; or an identification of where their theory may be true (a description of $\{\mathbf{v} : A(\mathbf{v}) \Rightarrow H(\mathbf{v})\}$); or the knowledge that their assumptions contradict.

Such technology could also allow for exploration. An economist could vary the question: the assumptions generating a **True** result can be weakened, or those generating a **Mixed** result strengthened, by quantifying more or less of \mathbf{v} .

For example, we can partition \mathbf{v} into $\mathbf{v}_1, \mathbf{v}_2$ and ask for $\{\mathbf{v}_1 : \forall \mathbf{v}_2. A(\mathbf{v}_1, \mathbf{v}_2) \Rightarrow H(\mathbf{v}_1, \mathbf{v}_2)\}$. The result is a formula in the free variables \mathbf{v}_1 that weakens or strengthens the assumptions. If generated automatically the economist gains information about how to reformulate assumptions that justify her hypothesis.

Table 1. Table of possible outcomes from a potential theorem $\forall \mathbf{v}. A \Rightarrow H$

	$\neg \exists \mathbf{v}[A \wedge \neg H]$	$\exists \mathbf{v}[A \wedge \neg H]$
$\exists \mathbf{v}[A \wedge H]$	True	Mixed
$\neg \exists \mathbf{v}[A \wedge H]$	Contradictory Assumptions	False

1.1 Quantifier Elimination

Such problems fall within the framework of *Quantifier Elimination* (QE): the generation of an equivalent quantifier-free formula from one that contains quantifiers. QE is known to be possible over real closed fields thanks to the seminal work of Tarski [17]. Practical implementations followed with Collins' Cylindrical Algebraic Decomposition [4] and Weispfenning's Virtual Substitution [19]. There are modern implementations of QE in many computer algebra systems.

QE has found many applications within engineering and the life sciences. Recent examples include: the derivation of optimal numerical schemes [5], weight minimisation for truss design [3], and biological network analysis [1]. However, applications in the social sciences are lacking (the nearest we can find is [8]). On the few occasions when QE has been mentioned in economics it has been dismissed as infeasible, e.g. "something that is do-able in principle, but not by any computer that you and I are ever likely to see" [12]. But that dismissal is based on theoretical complexity results rather than experience with actual software applied to actual economic reasoning. Many meaningful economics problems can be studied with modern QE implementations⁴, with the barrier to further use acceptance by the community, and experience with the software.

1.2 A New Mathematica Package TheoryGuru

This motivated the development of a new tool to aid the application of QE to economics: a package called **TheoryGuru** to run in the MATHEMATICA computer algebra system [20]. This is able to parse input from economists, run some error and sanity checks, and then utilise MATHEMATICA's QE tools and offer interpretations of the results. These QE tools are accessed by the MATHEMATICA **Resolve** command with some of the underlying algorithms described in [13], [14], [15], [16]. The paper proceeds in Section 2 by introducing the functionality of **TheoryGuru**. Then in the remaining sections we describe examples of its use.

2 Functionality

2.1 Main Functionality

The purpose of **TheoryGuru** is to lower the costs of applying QE to economics. Hence it assumes the expression of reasoning in the format traditional to the field: as a conclusion to be possibly deduced from a set of assumptions.

⁴ A dataset of 45 economic reasoning examples that may be tackled with QE technology is available here: <https://doi.org/10.5281/zenodo.1226892>

The core functionality of `TheoryGuru` is then as follows:

- Check for errors:** Provide warnings on likely typographical errors in variables (e.g., when a variable appears only once in the entire formula) or formula structure (e.g. the user may have confused = with ==).
- Parse input:** This includes identifying from context whether a variable is a vector, scalar, or boolean; processing input given in a *pretty* mathematical notation (e.g. derivatives) into a format accepted by `Resolve`; standardizing dot products and integrals (e.g., distribute plus and alphabetically sort arguments of commutative operators).
- Adding standard assumptions:** If dot products are present, then add to user assumptions the necessary and sufficient conditions for the Gramian matrix (representing dot products for all pairs of vectors) to be positive semi-definite. This rules out vectors with imaginary elements.
- Check assumptions:** The package will next check that the assumptions provided are not mutually contradictory: the situation of the bottom left entry in Table 1. This is done via a call to `Resolve` to check there is at least one solution to $A(v)$ – a fully existentially quantified QE sub-problem.
- Form main QE input:** Automatically assemble the two Tarski formulae for the main calls (as given in Table 1).
- Make algorithm choices:** Currently this refers to (a) whether to process a universal or existential sentence and (b) the variable ordering determining the sequence for eliminating quantifiers. It is well known that the choice (b), while not affecting the correctness of the output, can have a large effect on computational resources required [6].
- Output interpretation:** Then after making the two calls to MATHEMATICA's `Resolve`, the package interprets the results by identifying the relevant cell from Table 1. The package also suggests what to do next: e.g., when applicable, show a counterexample, solve simultaneous equations appearing in the assumptions, or redo the QE with some free variables.

2.2 Access and Documentation

To access `TheoryGuru` the reader will need a modern version of MATHEMATICA⁵ and then installation follows from simply running the command: `Get["http://economicreasoning.com"]` which produces an interface as in Figure 1. Not only does this install the underlying code, it also provides links to tutorials, tips, help and a large bank of examples (as shown in Figure 1). The examples are also available online at `http://examples.economicreasoning.com/` as both interactive MATHEMATICA notebooks and static pdfs.

The main functionality is accessed via the function `TheoryGuru` which requires two arguments: a collection of assumptions and a hypothesis.

⁵ The `Resolve` function has evolved and improved over the years and so the performance of `TheoryGuru` will alter correspondingly.



Fig. 1. Initial load screen of TheoryGuru

3 Examples of TheoryGuru Use

3.1 Tax Incidence

Our first, admittedly simple, *Tax Incidence* example is about the effect of a tax on buyers and sellers in a market. Each transaction involves the buyer paying `price` to the seller in addition to paying `tax` to the government. The symbolic functions `demand(.)` and `supply(.)` represent the quantities that buyers purchase and sellers sell, respectively, as a function of the price they pay or receive (so for the buyer that includes the tax). A market equilibrium `price` has the quantity demanded equal to the quantity supplied. The equilibrium condition can be input to MATHEMATICA as shown in the top cell of Figure 2, which assigns the condition the natural language name `Equilibrium`.

```

In[2]:= Equilibrium = demand[price + tax] == supply[price];
In[3]:= TheoryGuru[{{ $\frac{dEquilibrium}{d tax}$ , demand'[price + tax] < 0, supply'[price] > 0},  $\frac{dprice}{d tax} \leq 0$ }]
Out[3]= True

```

Fig. 2. Tax Incidence example in TheoryGuru

The first argument of the call to `TheoryGuru` in the second cell of Figure 2 is a set of assumptions. The first of these is that changing the `tax` changes the market from one equilibrium to another⁶. The remaining two constrain the slope of the demand and supply curves in the neighborhood of the market equilibrium. The second argument is the hypothesis the user wishes to test, in this case whether the price impact is negative or zero.

The call causes `TheoryGuru` to automatically assemble Tarski formulae, in which it recognizes `demand'(price+tax)` and `supply'(price)` as partially interpreted functions [7, page 73]. Following the generic format presented above,

⁶ The notation is consistent with an economist saying that she “totally differentiates the equilibrium condition”. This differentiation is automatically performed by MATHEMATICA when the `TheoryGuru` function is evaluated.

there are two QE problems for TheoryGuru to consider: the existence of an example and the existence of a counterexample. Here the output is simply True because there is no counterexample: i.e., no way to have a positive price impact while satisfying the assumptions.

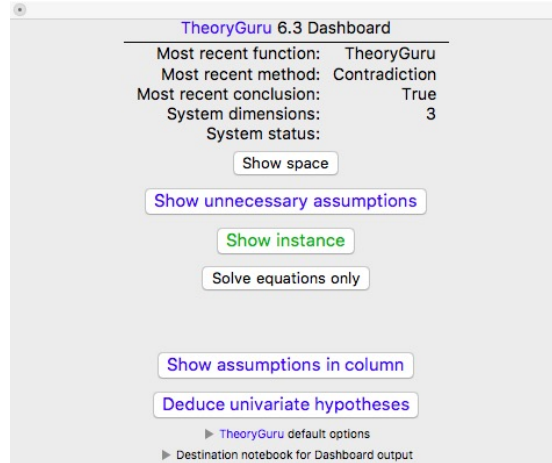


Fig. 3. The TheoryGuru dashboard

When TheoryGuru evaluates, a dashboard (Figure 3) appears summarizing the calculation and offering the user possible next steps. In the tax incidence example, the user may be wondering what else can be concluded about the price impact. The button labelled "Deduce univariate hypotheses" on the dashboard serves this purpose. Pressing it automatically generates a call to the function TheoryPossibilities as shown in Figure 4. Here, one or more free variables are provided by the user, or else variables are chosen by the software (giving priority to total derivative variables and alphabetical order). The assumptions are then projected on each of the free variables separately (eliminating existential quantifiers from all variables except that one), with the resulting formulae simplified. In this example we discover the price impact must be strictly between -1 and 0 .

```

In[4]:= TheoryPossibilities[(* optional: replace this comment with a list of variables *)
]//OtherTools`TFPrintL;
Using MostRecentAssumption and
Forming hypotheses from  $\left\{\frac{dprice}{dtax}\right\}$ .
-1 <  $\frac{dprice}{dtax}$  < 0

```

Automatically typed
by clicking
dashboard

Fig. 4. TheoryPossibilities call from dashboard button to propose new hypothesis

Users can be forgetful or have an imperfect understanding of an economic model. In the top cell of Figure 5 no definitive conclusion about price impact is reached because the user has forgotten to constrain the supply curve's slope.

```

In[5]= TheoryGuru[{ $\frac{d\text{Equilibrium}}{d\text{tax}}$ , demand'[price + tax] < 0},  $\frac{d\text{price}}{d\text{tax}} \leq 0$ ]
Out[5]= True for some, False for others

In[6]= TheorySufficient[]
Out[6]= supply'[price] ≥ 0

```

Fig. 5. Example use of `TheorySufficient` to recommend additional assumptions

The forgotten assumption can be discovered with `TheorySufficient`. It assembles the formula $A \wedge \neg H$ defining counterexamples. It then projects that set on each of the axes (three in this example). It then shows the disjunction of each formula, after each is simplified based on the assumptions and then negated. Here two formulae are discarded because they are **False** or identical to H . The third is the missing supply-slope restriction output in the second cell of Figure 5. Note that, by construction, any of `TheorySufficient`'s disjunction branches, together with the user's (insufficient) assumptions, imply the user's hypothesis.

3.2 Gender Wage Gap

We now look at a more involved *Gender Wage Gap* example that studies the effect of wage inequality on women's supply of human capital to the market. Women are assumed to have (possibly correlated) skills h and r in market work and non-market activities, respectively. These skills have a population distribution modelled with the joint density function $f(h, r)$, which is normalized to have unconditional means of zero. Women work if and only if their non-market log wage $r + \mu_r$ is less than $\sigma h + \mu_w$, their market log wage. It follows that mean non-market and market log wages are μ_r and μ_w , respectively. The employment rate of women is $p(\sigma, \mu_w - \mu_r)$ as defined in the top cell of Figure 6, and the average skill in the market is $S(\sigma, \mu_w - \mu_r)$ as defined in the next cell.

$$\begin{aligned}
 \text{In[7]= } p[\sigma, \text{meangap}_] &= \int_{-\infty}^{\infty} \int_{-\infty}^{\sigma h - \text{meangap}_} f[h, r] \, dr \, dh; \\
 \text{In[8]= } S[\sigma, \text{meangap}_] &= \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\sigma h - \text{meangap}_} h f[h, r] \, dr \, dh}{p[\sigma, \text{meangap}_]}; \\
 \text{In[9]= } \text{DefineExperiment} &= \left\{ \frac{dp[\sigma, \mu_w - \mu_r]}{dz} = \frac{d\mu_w}{dz} = 0, \frac{d\sigma}{dz} > 0 \right\};
 \end{aligned}$$

Fig. 6. Defining variables for the Gender Wage Gap example

```

In[10]= PositiveEmployment = p[σ, μR - μW] > 0;
In[11]= InequalityOnTheMargin =  $\frac{\int_{-\infty}^{\infty} h^2 f[h, \mu W - \mu R + h \sigma] dh}{\int_{-\infty}^{\infty} f[h, \mu W - \mu R + h \sigma] dh} > \left( \frac{\int_{-\infty}^{\infty} h f[h, \mu W - \mu R + h \sigma] dh}{\int_{-\infty}^{\infty} f[h, \mu W - \mu R + h \sigma] dh} \right)^2 \wedge \int_{-\infty}^{\infty} f[h, \mu W - \mu R + h \sigma] dh > 0;$ 
In[12]= SkillImpact =  $\frac{dS[\sigma, \mu R - \mu W]}{dz};$ 

```

Fig. 7. Further definitions and restrictions to the Gender Wage Gap example.

```

In[13]= TheoryGuru[{DefineExperiment, PositiveEmployment, InequalityOnTheMargin}, SkillImpact > 0]
Out[13]= True

In[14]= TheorySpace[]//OtherTools`TFPrintL;
Using MostRecentTheory.
 $\left\{ \frac{d\mu R}{dz}, \frac{d\mu W}{dz}, \frac{d\sigma}{dz}, \int_{-\infty}^{\infty} f(h, h\sigma - \mu R + \mu W) dh, \int_{-\infty}^{\infty} h f(h, h\sigma - \mu R + \mu W) dh, \right.$ 
 $\left. \int_{-\infty}^{\infty} h^2 f(h, h\sigma - \mu R + \mu W) dh, \int_{-\infty}^{\infty} \int_{-\infty}^{h\sigma - \mu R + \mu W} f(h, r) dr dh, \int_{-\infty}^{\infty} \int_{-\infty}^{h\sigma - \mu R + \mu W} h f(h, r) dr dh \right\}$ 

```

Fig. 8. Evaluating the Gender Wage Gap example

In [11] the Gaussian model is used to show how wage inequality, as modelled by σ , affects the average skill in the market. However, the *selection rule* result – that is, the effect of σ on S holding constant p by varying μ_r – does not require the Gaussian assumption. To show this, we define z to be any change in σ and μ_r that increases σ and holds p constant, as defined in the third cell of Figure 6 (DefineExperiment). Figure 7 then assigns natural language to two definitions (top and bottom cells) as well as restrictions on partially interpreted functions (middle cell). The top cell of Figure 8 shows that a positive skill impact can be deduced from the assumed properties of the partially interpreted functions. In economics terms: inequality increases the average skill that women supply to the market, thereby narrowing the measured wage gap with men.

At first glance, the gender wage gap example appears to involve integrable probability density functions rather than the scalar variables required by the QE algorithms employed by MATHEMATICA's Resolve function. But the reasoning in this and many other examples depends on the probability density functions only as they are summarized by various scalars. TheoryGuru automatically discovers these scalar variables, which can be viewed by the user who clicks "Show space" on TheoryGuru's dashboard. The result of that click is the last cell of Figure 8.

4 Run Times

Figure 9 shows the run times for several of the function calls shown above. As explained, each evaluation involves preparation of a QE problem for MATHEMATICA's Resolve function, followed by that QE call. The cell numbers refer to those used by MATHEMATICA in the screen shots above. The figure's first run time column is for the entire evaluation of the TheoryGuru command. The next

Function call	Dimensions represented	Run times in seconds		
		Entire evaluation	QE only Universal	QE only Existential
TheoryGuru (cell 3)	3	0.10	< 0.01	< 0.01
TheoryPossibilities (cell 4)	3	0.19	N/A	< 0.01
TheorySufficient (cell 6)	3	0.15	N/A	< 0.01
TheoryGuru (cell 13)	8	3.31	< 0.01	< 0.01
Addenda: TheoryGuru queries about production function concavity				
3 inputs, delineated individually	12	1.56	0.41	< 0.01
4 inputs, delineated individually	18	469.68	8.89	0.03
Any number of inputs (vector mode)	10	0.41	0.13	< 0.01

Notes: Run time was calculated with Mathematica 11.2 on a Macbook Pro Mid 2014 2.8GHz Intel i7 Quadcore processor, running TheoryGuru in Parallel Mode with popup dashboard already open and using the default tuning for variable search. Universal (existential) QE refers to deciding the existence of a counterexample (example), respectively, with the Tarski formula already assembled and variable-elimination sequence already determined.

Fig. 9. Run times for several examples

column shows, when applicable, the amount of time it took for just the “universal” QE regarding the existence of a counterexample. The final column is the amount of time it took for just the “existential” QE regarding the existence of an example (the faster QE for calls that have no counterexamples).

In order to explore the limits of the software, we consider queries regarding the concavity of quasiconcave production functions, whose economics and algebra we discuss in [10]. The three-input version evaluates in less than two seconds. The four-input version is considerably more complicated, and evaluates in about eight minutes primarily because of a long search to find a relatively efficient order for eliminating quantifiers⁷. The problem can be solved more elegantly with vectors, with a quicker run time as shown in the final row (see also [9]).

5 Final Thoughts

We have demonstrated how economic reasoning may be automated using QE procedures and how the TheoryGuru tools greatly reduce the costs to an economist of accessing that technology. We note that a set of benchmark examples that originate from economics and may be tackled with QE is now available at <https://doi.org/10.5281/zenodo.1226892> and described in [10]. Future work will involve considering how the underlying QE technology could be optimised for such examples, whose structure is often not well represented in the broader QE literature.

⁷ It is once that order is obtained that the corresponding QE needs only 8.89 seconds.

References

1. Bradford, R., Davenport, J., England, M., Errami, H., Gerdt, V., Grigoriev, D., Hoyt, C., Košta, M., Radulescu, O., Sturm, T., Weber, A.: A case study on the parametric occurrence of multiple steady states. In: Proc. ISSAC '17. pp. 45–52. ACM (2017)
2. Caviness, B., Johnson, J.: Quantifier Elimination and Cylindrical Algebraic Decomposition. Texts & Monographs in Symbolic Computation, Springer-Verlag (1998)
3. Charalampakis, A., Chatzigiannelis, I.: Analytical solutions for the minimum weight design of trusses by cylindrical algebraic decomposition. *Archive of Applied Mechanics* 88(1), 39–49 (2018)
4. Collins, G.: Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In: Proc. 2nd GI Conference on Automata Theory and Formal Languages. pp. 134–183. Springer-Verlag (reprinted in [2]) (1975)
5. Erascu, M., Hong, H.: Real quantifier elimination for the synthesis of optimal numerical algorithms (Case study: Square root computation). *J. Symb. Comp.* 75, 110–126 (2016)
6. Huang, Z., England, M., Wilson, D., Davenport, J., Paulson, L., Bridge, J.: Applying machine learning to the problem of choosing a heuristic to select the variable ordering for cylindrical algebraic decomposition. In: Proc. CICM '14 (LNAI vol. 8543), pp. 92–107. Springer International (2014)
7. Kroening, D., Strichman, O.: Decision Procedures: An Algorithmic Point of View. Springer, New York (2013)
8. Li, X., Wang, D.: Computing equilibria of semi-algebraic economies using triangular decomposition and real solution classification. *J. Math. Econ.* 54, 48–58 (2014)
9. Mulligan, C.: Automated Economic Reasoning with Quantifier Elimination. NBER working paper no. 22922 (2016)
10. Mulligan, C., Bradford, R., Davenport, J.H., England, M., Tonks, Z.: Non-linear Real Arithmetic Benchmarks derived from Automated Reasoning in Economics. To Appear In: Proc. 3rd International Workshop on Satisfiability Checking and Symbolic Computation (SC²) (2018)
11. Mulligan, C., Rubinstein, Y.: Selection, investment, and women’s relative wages over time. *Q. J. Econ.* 123(3), 1061–1110 (2008)
12. Steinhorn, C.: Tame Topology and O-Minimal Structures, pp. 165–191. Springer Berlin Heidelberg (2008)
13. Strzeboński, A.: Cylindrical algebraic decomposition using validated numerics. *J. Symb. Comp.* 41(9), 1021–1038 (2006)
14. Strzeboński, A.: Computation with semialgebraic sets represented by cylindrical algebraic formulas. In: Proc. ISSAC '10, pp. 61–68. ACM (2010)
15. Strzeboński, A.: Solving polynomial systems over semialgebraic sets represented by cylindrical algebraic formulas. In: Proc. ISSAC '12, pp. 335–342. ACM (2012)
16. Strzeboński, A.: Cylindrical algebraic decomposition using local projections. *J. Symb. Comp.* 76, 36–64 (2016)
17. Tarski, A.: A Decision Method For Elementary Algebra And Geometry. RAND Corporation, Santa Monica, CA (reprinted in [2]) (1948)
18. Wada, Y., Matsuzaki, T., Terui, A., Arai, N.: An automated deduction and its implementation for solving problem of sequence at university entrance examination. In: Proc. ICMS 2016 (LNCS vol. 9725), pp. 82–89. Springer (2016)
19. Weispfenning, V.: The complexity of linear problems in fields. *J. Symb. Comp.* 5(1/2), 3–27 (1988)
20. Wolfram, S.: The Mathematica Book. Wolfram Research Inc. (2000)