Considerations on Approaches and Metrics in Automated Theorem Generation/Finding in Geometry

Pedro Quaresma*
CISUC / Department of Mathematics, University of Coimbra, Portugal
pedro@mat.uc.pt

Pierluigi Graziani†
Department of Pure and Applied Sciences, University of Urbino, Italy
pierluigi.graziani@uniurb.it

Stefano M. Nicoletti‡
Formal Methods and Tools, University of Twente, Enschede, the Netherlands
s.m.nicoletti@utwente.nl

The pursue of what are properties that can be identified to permit an automated reasoning program to generate and find new and interesting theorems is an interesting research goal (pun intended). The automatic discovery of new theorems is a goal in itself, and it has been addressed in specific areas, with different methods. The separation of the "weeds", uninteresting, trivial facts, from the "wheat", new and interesting facts, is much harder, but is also being addressed by different authors using different approaches. In this paper we will focus on geometry. We present and discuss different approaches for the automatic discovery of geometric theorems (and properties), and different metrics to find the interesting theorems among all those that were generated. After this description we will introduce the first result of this article: an undecidability result proving that having an algorithmic procedure that decides for every possible Turing Machine that produces theorems, whether it is able to produce also interesting theorems, is an undecidable problem. Consequently, we will argue that judging whether a theorem prover is able to produce interesting theorems remains a non deterministic task, at best a task to be addressed by program based in an algorithm guided by heuristics criteria. Therefore, as a human, to satisfy this task two things are necessary: an expert survey that sheds light on what a theorem prover/finder of interesting geometric theorems is, and—to enable this analysis—other surveys that clarify metrics and approaches related to the interestingness of geometric theorems. In the conclusion of this article we will introduce the structure of two of these surveys —the second result of this article— and we will discuss some future work.

1 Introduction

In *Automated Reasoning: 33 Basic Research Problems*, Larry Wos, wrote about the problems that computer programs that reason face. Problem 31 is still open and object of active research [56, 57]:

> *Wos' Problem 31*—What properties can be identified to permit an automated reasoning program to find new and interesting theorems, as opposed to proving conjectured theorems?

Two problems in a single sentence: *new* and *interesting* theorems. The automatic discovery of new theorems is a goal in itself, it has been addressed in specific areas, with different methods. The separation

*Partially supported by FCT – Foundation for Science and Technology, I.P., within the scope of the project CISUC – UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020.

†Partially supported by Italian Ministry of Education, University and Research through the PRIN 2017 project "The Manifest Image and the Scientific Image" prot. 2017ZNWW7F_004.

‡Funded by ERC Consolidator Grant 864075 (*CAESAR*).

P. Quaresma and Z. Kovács (Ed.): Automated Deduction in Geometry 2023 (ADG 2023).
EPTCS 398, 2024, pp. 85–100, doi:10.4204/EPTCS.398.12

of the "weeds", uninteresting, trivial facts, from the "wheat", new and interesting facts, is much harder, but is being addressed also, by different authors using different approaches.

Paraphrasing, again, Wos, "since a reasoning program can be instructed to draw some (possible large) set of conclusions" what should be the "criteria that permit the program to select from those the ones (if any) that correspond to interesting results."

Different fields have come across the finding of new and interesting theorems' questions.

Regarding the novelty side: there are different views of approaching new mathematical results. One of those approaches is the systematic exploration of a given broad area of mathematical knowledge, generating, by different means, new theorems and expecting to find interesting ones among those generated (that will be analysed in section 4) [14, 15, 19, 25, 30, 35, 36, 46]. Another approach is given by the pursue of mathematical discovery in specific areas, e.g. *Computing Locus Equations* [1, 9], *Automated Discovery of Angle Theorems* [54], *Automated Discovery of Geometric Theorems Based on Vector Equations* [42], *Automated Generation of Geometric Theorems from Images of Diagrams* [13], *Automatic Discovery of Theorems in Elementary Geometry* [48]. These approaches do not aim to address the problem of automated theorem finding in itself but, for example, to find complementary hypotheses for a given geometric statements to become true [48] i.e. automatic discovery for specific areas.[1]

Regarding the interestingness side we are aware that relevant literature can be found in different areas. For example in automated theorem proving [19, 25, 30, 46] and in sociological studies on the concept of proving [20, 39, 40], in cognitive and educational science studies on the concept of proving [2, 11, 21, 33, 45, 52] and in semiotics and epistemology of mathematics [3, 4, 5, 8, 12].

Despite the cited studies, the Wos' problem is still on the table. On the contrary, a new result of undecidability can be added to the problem, i.e. having an algorithmic procedure that decides for every possible Turing Machine that produces theorems, whether it is able to produce also interesting theorems, is an undecidable problem. Consequently, we can argue that judging whether a theorem prover is able to produce interesting theorems remains a non deterministic task, at best a task to be addressed by program based in an algorithm guided by heuristics criteria. Therefore, as a human, to satisfy this task we need expert survey that sheds light on what a theorem prover/finder of interesting geometric theorems is, and—to enable this analysis—other surveys that clarify metrics and approaches related to the interestingness of geometric theorems.

Structure of the paper. In section 2 the issue of Automated Theorem Generation (ATG) is discussed. In section 3 we discuss the deductive approach in ATG. In section 4 the issue of Automated Theorem Finding (ATF) is analysed. In section 5 we present an undecidability result concerning the problem of finding interesting theorems and its conceptual consequences. In section 6 we will introduce the structure of two surveys to empirically explore the interestingness of theorems in geometry and its potential application in theorem proving/finding (a third survey). Finally, we will discuss some future work.

2 Automated Theorem Generation

Automated theorem generation, independently of being interesting, or not, can be addressed in several ways [46].

[1]We left aside the notion of discovery in education, given that, in that area, the goal is the student's discovery of "new" (for them) theorems, giving the student the possibility of freely making conjectures and having an interactive/automatic deduction support in the exploration of those "new" theorems [10, 32, 37, 38].

The Inductive approach, is a natural approach. Conclusions are drawn by going from the specific to the general. Exploring a given domain, seeking for properties that emerge from a set of particular cases and making a conjecture about the general case.

Dynamic Geometry Software (DGS) can be seen as software environments to inductively explore new knowledge. Making a geometric construction, constrained by a given set of geometric properties, and then moving the free point around will show all the fix-points, conjecturing if those new fixed relations between objects are true in all cases, or not. For example the Pappus' Theorem, in this case, a well-known theorem: are the intersection points (see Figure 1) G, H and I, collinear? By moving, in the DGS, the free-points it seems that they are, it remains to prove it.

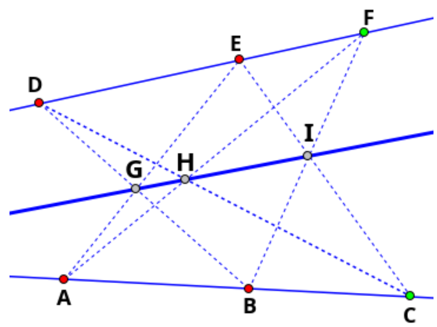

Figure 1: Pappus' Theorem

The inductive approach has the advantage of being stimulated by observations in the domain. but has the disadvantage that induction is unsound. A famous example of such unsound inductive approach can be seen in the Euclid Parallel lines Postulate, that nevertheless was very fruitful, giving raise to different geometries.

The Generative approach, i.e. the generation of conjectures, testing them for theorem-hood. The simplest form of generation is syntactic, in which conjectures are created by mechanical manipulation of symbols, e.g. [44]. The *MCS* program generates conjectures syntactically and filters them against models of the domain [59]. A stronger semantically based approach is taken by the *HR* program, which generates conjectures based on examples of concepts in the domain [18]. A theory exploration system called *QuickSpec*, works by interleaving term generation with random testing to form candidate conjectures [34]. In [34] the conjecture generation approaches are classified into three categories: heuristic rule-based systems, term generation-and-testing and neural network-based systems. The *RoughSpec* system adds to *QuickSpec* the notion of *shapes* of theorems, specifying the shapes of conjectures the user is interested in, and thus limiting the search [22].

Like induction, generation is unsound. However, if the rules by which the generation is performed are sufficiently conservative then this approach may generate a higher fraction of theorems than the inductive approach.

The Manipulative Approach, conjectures are generated from existing theorems. An existing theorem is modified by operations such as generalisation, specialisation, combination, etc. This approach is used in abstraction mapping, which converts a theorem to a simpler theorem, and uses a solution to the simpler theorem to help find a solution to the original theorem [43]. Manipulation of ATP theorems has also been used to produce new theorems for testing the robustness of ATP systems' performances [55].

An advantage of the manipulative approach is that, if the manipulations are satisfiability preserving, then theorems, rather than conjectures, are produced from existing theorems. However, the conjectures produced by the manipulative approach are typically artificial in nature, and thus uninteresting.

The Deductive Approach, consequences are generated by application of sound inference rules to the axioms and previously generated logical consequences. This can be done by an appropriately configured saturation-based ATP system.

The advantage of this approach is that only logical consequences are ever generated. The challenge of this approach is to avoid the many uninteresting logical consequences that can be generated.

3 The Deductive Approach

Some systems addresses, explicitly, the generation of new geometric results using different approaches. In the following some of these approaches are described.

3.1 Strong Relevant Logic-based Forward Deduction Approach

In [27] the authors argue for the fundamental difference between the Automated Theorem Proving (ATP) and the Automated Theorem Finding (ATF). ATP is the process of finding a justification for an explicitly specified statement from given premises which are already known facts or previously assumed hypotheses. ATF is the process to find out or bring to light that which was previously unknown. Where ATP is all about known (old) facts, ATF is about previously unknown conclusions from given premises. Jingde Cheng [15] claims that classical mathematical logic, its various classical conservative extensions, and traditional (weak) relevant logics cannot satisfactorily underlie epistemic processes in scientific discovery, presenting an approach based on strong relevant logic. Hongbiao Gao et al. have followed this approach applying it for several domains such as NBG set theory, Tarski's Geometry and Peano's Arithmetic [26, 27, 29, 30]

3.2 Rule Based Systems

The rule-based automated deduction system are often used when the proof itself is an object of interest (and not only the end result), given that the proofs are developed from the hypothesis and sets of axioms, to the conclusion by application of the inference rules, the proofs are "readable".

Example of such approaches can be seen in systems like *QED-Tutrix* [23, 24] and *JGEx* [58], both for geometry. In the tutorial system *QED-Tutrix*, the rule based automated theorem prover goal is to find the many possible branches of the proof tree, in order to be able to help the student approaching the proof of a geometric conjecture. In the *JGEx* system we can have the proof in a "readable" and "visual" renderings and also the set of all properties that can be deduced from the construction.

One of the ATP built-in in *JGEx* is an implementation of the geometry deductive database method [16, 58]. Using a breadth-first forward chaining a fix-point for the conjecture at hand is reached. For that geometric construction and the rules of the method, the fix-point gives us all the properties that can be deduced, some already known facts, but also new facts (not necessary interesting ones).

The geometry deductive database method proceeds by using a simple algorithm where, starting from the geometric construction D_0, the rules, R, are applied over and over till a fix-point, D_k is reached:

$$\boxed{D_0} \overset{R}{\subset} \boxed{D_1} \overset{R}{\subset} \quad \cdots \quad \overset{R}{\subset} \boxed{D_k} \quad \text{(fix-point)} \tag{1}$$

In figure 2 an example, using *JGEx*, is shown. On the right, the geometric construction, on the left, the fix-point, with all the facts that were found for that construction.

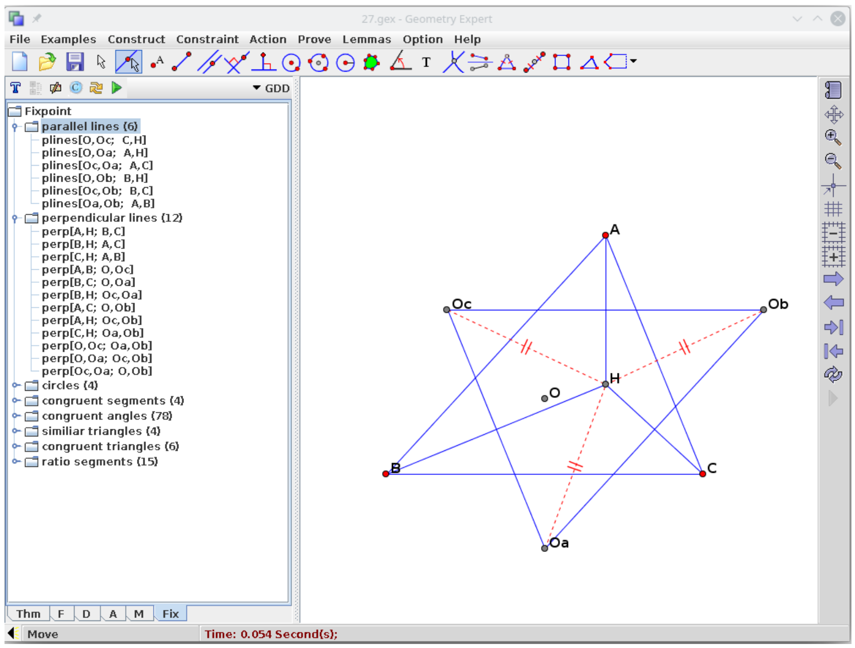

Figure 2: Fix-point in *JGEx*

A new open source implementation of this method, *OGP-GDDM*,[2] is described in [7]. It will be integrated in the *Open Geometry Prover Community Project* (OGPCP) [6]. One of the medium-term goals of the *OGP-GDDM* project, is to develop a meta-prover, a program capable to receive different sets of rules and synthesise a specific ATP for those rules.

3.3 Algebraic Approaches

A similar approach is taken in the well-known dynamic geometry system *GeoGebra*.[3] The *GeoGebra Discovery* version[4] has the capability to find, from a user defined geometric construction, properties about that construction. *GeoGebra Discovery* reports some facts that were systematically checked from a list of possible features including identical points, parallel or perpendicular lines, equal long segments, collinearity or concyclicity. This is not a deductive method so the generation process must be externally verified, *GeoGebra Discovery* do that by recurring to a built-in algebraic automated theorem prover based in the Gröbner bases method [35, 36].

[2] `https://github.com/opengeometryprover/OpenGeometryProver`
[3] `https://www.geogebra.org/`
[4] `https://github.com/kovzol/geogebra-discovery`

4 Automated Theorem Finding

Apart from our research goal of finding the interesting geometric theorems among all those that were automatically generated, the pursue of measures of interestingness has applicability in the interactive and automated theorem proving area. In that area a common use of interestingness is to improve the efficiency of the programs, tailoring the search space, making the search depth limited and guaranteeing that only comprehensible concepts are produced [19].

A goal, pursued with different approaches by many researchers, is the creation of strong AI methods capable of complex research-level proofs, mathematical discovery, and automated formalisation of today's vast body of mathematics [47]. The *MATHsAiD* (Mechanically Ascertaining Theorems from Hypotheses, Axioms and Definitions) project aimed to build a tool for automated theorem-discovery, from a set of user-supplied axioms and definitions. In the words of its authors, *MATHsAiD 2.0* can conjecture and prove interesting Theorems in high-level theories, including Theorems of current mathematical significance, without generating an unacceptable number of uninteresting theorems [41]. The *TacticToe* system, combines reinforcement-learning with Monte-Carlo proof search on the level of *HOL4* tactics [31]. The *ENIGMA-NG* system uses efficient neural and gradient-boosted inference guidance for the ATP *E*, improving its efficiency [17]. This two systems, one for interactive provers and the other to automatic provers, are examples of systems that uses discovery and filtering for improving the efficiency of automated deduction systems.

4.1 The Deductive Approach Algorithm

The different approaches found in the literature [18, 27, 46] share, in their general lines, the same algorithm: for a given logical fragment, select a initial set of facts and then a cycle of generation/filtering is applied until some stopping condition is matched (see Fig. 4.1).

4.2 Filtering Interesting Theorems

A first level of filtering (run-time filter) should discard the obvious tautologies and also conjectures proved false by empirical evidence.

The filtering for interesting theorems or for uninteresting conjectures, two sides of the same coin, is done by application of a series of filters. These filters are still to be validated, being of speculative nature [19, 29, 30, 46].

Obviousness: the number of inferences in its derivation. Obviousness estimates the difficulty of proving a formula, it can be given by the number of inferences in its derivation.

Weight: the effort required to read a formula. The weight score of a formula is the number of symbols it contains.

Complexity: the effort required to understand a formula, the number of distinct function and predicate symbols it contains.

Surprisingness: measures new relationships between concepts and properties.

Intensity: measures how much a formula summarises information from the leaf ancestors in its derivation tree.

Adaptivity: measures how tightly the universally quantified variables of a formula are constrained (for formulae in clause normal form).

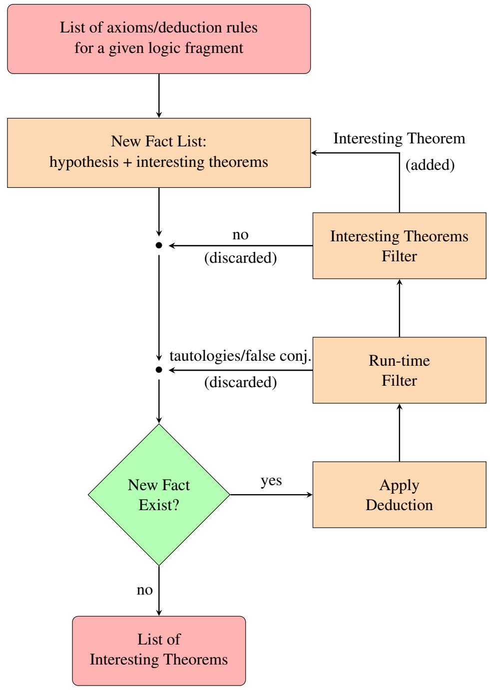

Figure 3: New and Interesting Theorems Algorithm

Focus: measures the extent to which a formula is making a positive or negative statement about the domain of application.

Usefulness: measures how much an interesting theorem has contributed to proofs of further interesting theorems.

In spite of the relevance of these metrics, it would be appropriate to construct an expert survey with which we could validate them by referring to a significant public of experts. We believe this kind of survey would be relevant not only to face Wos' problem, but also to better understand how to construct and evaluate software that generates/finds interesting theorems. Despite having only relevant metrics and approaches regarding Wos' problem, while not yet having formal results, we can prove a relevant result that concerns the second issue, i.e., the question regarding *Interesting Turing Machines*, i.e., programs capable of generating interesting new geometric results.

5 Undecidability Result

In section 4.2 the application of filters was discussed, these filters are based on some measures of interestingness that are still to be validated and that are applied in an heuristic way. Is it possible to have a deterministic approach, i.e., is it possible to write a computer program that in a deterministic way, find interesting theorems? We show, as an application of the Rice's theorem [49, 50, 51] (see Lemma 1), that it is undecidable to determine, for a given Turing Machine, whether the language recognised by it has the (non-trivial) property of finding interesting theorems.

Definition 1 (Non-Trivial Property)**.** *A property p of a formal language is* non-trivial *if:*

- *there exist a recursively enumerable language having the property p;*
- *there exist a recursively enumerable language not having the property p.*

Lemma 1 (Rice's Theorem). *Let p be any non-trivial property of the language of a Turing machine. The problem of determining whether a given Turing machine's language has property p is undecidable.*

Theorem 1 (Undecidability Result). *For any given Turing Machine, it is undecidable to determine, whether the language recognised by it has the property of finding interesting theorems.*

Proof. All programs (Turing machines) capable of automated theorem proving and by extension generating/finding geometric theorems rely on a formal language to describe the geometric constructions, conjectures and proofs. For example we can consider the (full) *First-Order Form* (FOF)[5] of *TPTP* [53] and the formal axiomatic theories for geometry based on that language.[6]

Let p be the property of that language that says that theorem t is interesting, for any conceivable definition of interestingness, then there exist a recursively enumerable language having the property p. It will be enough to restrict the language in such a way that the theorem t, and only this, would be recognised. But, it also exist a recursively enumerable language not having the property p. It would be enough to restrict that language in such a way that only tautologies would be recognised. Tautologies are, for any conceivable definition of interestingness, uninteresting. We have proved that p, the property that can establish if a given theorem is interesting, is a non-trivial property.

Having establish that the property p is non-trivial, then, by application of Rice's theorem, it is undecidable to determine for any given Turing machine M, whether the language recognised by M has the property p. □

In other words, it is undecidable to have a deterministic program that can find interesting problems. At best this is a task to be addressed by programs based on algorithms guided by heuristics criteria.

6 Designing Interesting Surveys

In light of our undecidability result, to understand what experts mean by, "a program that is able to also prove interesting theorems", must be done referring to empirical data, via the formulation of an expert survey. However, for it to be fulfilled, one has to first reach a minimal degree of agreement on the definition of interestingness of theorems. How could one speak about programs that produce such theorems? In order to achieve this agreement, an empirical exploration of the notion of interestingness and of what it concretely entails is paramount. This exploration requires to situate the notion of interestingness historically and socio-culturally, considering logical, epistemological, sociological, cognitive, semiotic and pedagogical aspects of the issue. Probably—and as Wos already implies—interestingness entails different tangible properties, which differ in given centuries, geographical locations and societies. Moreover, in some cases we say that a theorem is interesting for what we can call *global reasons* e.g., Euclid's theorem on the infinitude of the set of prime numbers, Zorn's lemma and Gödel's Theorems are interesting due to their role in mathematics, logic and computer science. Other times for *local reasons* e.g., in relation to what we are teaching our students at that moment. In order to assess which tangible properties—both global and local—interestingness entails today, we are proposing to conduct two *expert surveys* with two statistically significant pools of participants.

[5]`http://tptp.cs.miami.edu/TPTP/QuickGuide/`

[6]TPTP Axioms Files for geometry, `https://www.tptp.org/cgi-bin/SeeTPTP?Category=Axioms`, e.g. *Tarski geometry axiom*, `GEO001` and `GEO002`, *Deductive Databases Method in Geometry*, `GEO012`.

Influencing factors. Gao et al. performed an extensive analysis of areas like Set Theory, Peano's arithmetic and Tarski's Geometry, looking for the relevance of structural factors, such as the degree of logical connectives in the theorem, the propositional schema of the formula formalising the theorem, the abstract level of predicates and functions in the theorem and the deduction distance of a theorem [19, 28, 30, 46]. Some of these structural aspects might be related to our cognitive dynamics. But also the epistemological role of a theorem with respect to other theorems might be a relevant feature; or the educational role that some theorems have with respect to some notions might influence their interestingness. Finally, the history of a theorem—e.g. Fermat's last theorem—could add points to its interestingness, which, in the case of Fermat's last theorem, might be already caused by the technicalities of the proof itself.

Designing the surveys. Taking all these factors into consideration, we would propose to design three surveys that question experts from different fields.

Before describing the surveys below some clarifications are necessary. We will use the term "expert" to mean mathematics teachers at primary, middle, and high schools, and professors or researchers in pure and applied mathematics at universities or at research centres. Furthermore, we will focus on the case study of geometry, hence interesting theorems in geometry. The reasons for this restriction to geometry are as follows: on the one hand, considering all fields of research in mathematics might require a too large number of experts and could produce too many divergent ideas. On the other hand, having in mind an application of the results in automatic theorem proving as a target, it seems appropriate to move into an area were there are many different methods and many automated provers implementing those methods. Finally, geometry is a kind of language common to many areas of mathematics and has been a domain for reflection since the early years of mathematics teaching.

Finally, these surveys are intended to involve mathematics teachers, but their outcome does not target mathematics education. Of course, this is a possible target, but it is not the primary goal of these starting surveys.

6.1 Three Surveys

In the first survey, we will ask the experts both to indicate some situations in which they remember to have used the adjective interesting concerning a theorem, and to explain the use of this expression. In addition, we will ask experts to list several geometric theorems they find interesting, and to list several geometric theorems they find not interesting, both from elementary and higher geometry, explaining the reasons for their answers (see Appendix A). This first survey is already under way, the steering committee is already approaching it and the authors of accepted papers in the conference, *14th International Conference on Automated Deduction in Geometry* (ADG 2023)[7], were invited to participate. We are planning to enlarge it to our network of contacts and we invite the interested reader to also participate, answering it.[8] We are planning to begin collect and analyse the answers in February, 2024.

We will use the information from this survey to define a list of characteristics (A,B,C, ...) of a theorem that offer sufficient reasons to attribute interestingness to it. We will assign weights to the various characteristics by considering the answers to this first survey.

After the first survey, we will implement a second one. This second survey will consider a list of theorems that, in different percentages, have the characteristics inferred from the first survey. We will

[7] ADG 2023, 14th International Conference on Automated Deduction in Geometry, Belgrade, Serbia, September 20-22, 2023.

[8] `https://docs.google.com/forms/d/e/1FAIpQLScIXZbLPBHTLvmQ28P30Cm_-lkOrM7e6rab7ho0WrAFwf_mbQ/viewform?usp=sf_link`

submit the second survey to a set of experts different from those used in the first survey. We will ask these experts whether they find the theorems listed interesting or not. We will ask them to rate, using a Likert scale,[9] the degree of impact that having certain characteristics plays in their attribution of interestingness (see Appendix B).

This second group will allow us to understand whether the characteristics isolated through the first survey are sufficient conditions to affirm that a theorem is interesting.

With an agreement on what an interesting theorem is, based on empirical research, we could query experts in theorem generators/finders design, with another survey (the third survey) asking how to design software able to produce these interesting theorems.

After that, we will focus our empirical inquiry on programs, driven by heuristics based on our findings, able to find interesting theorems.

We have established a *steering committee* to design the surveys and who will oversee the submission of the surveys to experts around the world.

The steering committee consists of the following scholars:

- Thierry Dana-Picard, Jerusalem College of Technology, Jerusalem, Israel;
- James Davenport, University of Bath, United Kingdom;
- Pierluigi Graziani, University of Urbino, Urbino, Italy;
- Pedro Quaresma, University of Coimbra, Coimbra, Portugal;
- Tomás Recio, University Antonio de Nebrija, Madrid, Spain.

7 Conclusions

The pursuit of new and interesting theorems in geometry, by automatic means is an interesting open problem. From the point of view of generating new information the deductive approach seems the most appropriated, given that: only logical consequences are ever generated and also the paths to those new theorems can be analysed from the point of view of the geometric theory used, i.e. in the process of generating new facts, geometric proofs of their validity are produced. Already existing implementations, e.g. the deductive databases method (DDM) implemented in *JGEx*, and new implementations, e.g. the *GeoGebra Discovery* and the new implementation of the DDM, the *OGPCP-GDDM* prover, can be used. The separation of the uninteresting, trivial facts, from the new and interesting facts is much harder. The current approaches are based in ad-hoc measures, proposed by experts from the field, but nevertheless, not substantiated by any study approaching that problem. Our goal is to fulfil that gap, to produce a comprehensive survey, supported in a large set of mathematicians, in order to be able to return to that question and to develop filters supported by the findings of that survey.

Acknowledgements The authors wish to thank Francisco Botana, Thierry Dana-Picard, James Davenport and Tomás Recio for their support in the pursue of this long term project.

[9] A Likert scale is a question which is a five-point or seven-point scale. The choices range from Strongly Agree to Strongly Disagree so the survey maker can get a holistic view of people's opinions. It was developed in 1932 by the social psychologist Rensis Likert.

References

[1] Miguel Á. Abánades, Francisco Botana, Antonio Montes & Tomás Recio (2014): *An algebraic taxonomy for locus computation in dynamic geometry*. Computer-Aided Design 56, pp. 22–33, doi:10.1016/j.cad.2014.06.008.

[2] Gilles Aldon, Pierre-Yves Cahuet, Viviane Durand-Guerrier, Mathias Front, Didier Krieger, Michel Mizony & Claire Tardy C (2010): *Expérimenter des problèmes innovants en mathématiques à l'école*. INRP. Available at `https://hal.archives-ouvertes.fr/hal-00989132`.

[3] Ferdinando Arzarello & Cristina Sabena (2010): *Semiotic and theoretic control in argumentation and proof activities*. Educational Studies in Mathematics 77(2-3), pp. 189–206, doi:10.1007/s10649-010-9280-3.

[4] Michael Aschbacher (2005): *Highly complex proofs and implications of such proofs*. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 363(1835), pp. 2401–2406, doi:10.1098/rsta.2005.1655.

[5] Jeremy Avigad (2006): *Mathematical Method and Proof*. Synthese 153(1), pp. 105–159, doi:10.1007/s11229-005-4064-5. Available at `http://www.jstor.org/stable/27653412`.

[6] Nuno Baeta & Pedro Quaresma (2021): *Open Geometry Prover Community Project*. Electronic Proceedings in Theoretical Computer Science 352, pp. 129–138, doi:10.4204/EPTCS.352.14.

[7] Nuno Baeta & Pedro Quaresma (2023): *Towards a Geometry Deductive Database Prover*. Annals of Mathematics and Artificial Intelligence 91(6), pp. 851–863, doi:10.1007/s10472-023-09839-0.

[8] Henk Barendregt & Freek Wiedijk (2005): *The challenge of computer mathematics*. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 363(1835), pp. 2351–2375, doi:10.1098/rsta.2005.1650.

[9] Francisco Botana, Miguel A. Abánades & Jesús Escribano (2007): *Computing Locus Equations for Standard Dynamic Geometry Environments*. In Yong Shi, G. Dick van Albada, Jack Dongarra & Peter M. A. Sloot, editors: International Conference on Computational Science, Lecture Notes in Computer Science 4488, Springer Berlin Heidelberg, pp. 227–234, doi:10.1007/978-3-540-72586-2_32.

[10] Francisco Botana & José L. Valcarce (2002): *A dynamic-symbolic interface for geometric theorem discovery*. Computers and Education 38(1-3), pp. 21–35, doi:10.1016/S0360-1315(01)00089-6.

[11] Alan Bundy, Mateja Jamnik & Andrew Fugard (2005): *What is a proof?* Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 363(1835), pp. 2377–2391, doi:10.1098/rsta.2005.1651.

[12] Tyler Burge (1998): *Computer proof, apriori knowledge, and other minds: The sixth philosophical perspectives lecture*. Philosophical perspectives 12, pp. 1–37, doi:10.1093/acprof:oso/9780199672028.003.0013.

[13] Xiaoyu Chen, Dan Song & Dongming Wang (2014): *Automated generation of geometric theorems from images of diagrams*. Annals of Mathematics and Artificial Intelligence 74(3-4), pp. 1–26, doi:10.1007/s10472-014-9433-7.

[14] Jingde Cheng (1995): *Entailment calculus as the logical basis of automated theorem finding in scientific discovery*. In: Systematic Methods of Scientific Discovery: Papers from the 1995 Spring Symposium, AAAI Press, pp. 105–110.

[15] Jingde Cheng (2000): *A Strong Relevant Logic Model of Epistemic Processes in Scientific Discovery*. In E. Kawaguchi, H. Kangassalo, H. Jaakkola & I.A. Hamid, editors: Information Modelling and Knowledge Bases XI, Frontiers in Artificial Intelligence and Applications 61, IOS Press, pp. 136–159, doi:10.1007/3-540-49292-5_42.

[16] Shang-Ching Chou, Xiao-Shan Gao & Jing-Zhong Zhang (2000): *A Deductive Database Approach to Automated Geometry Theorem Proving and Discovering*. Journal of Automated Reasoning 25(3), pp. 219–246, doi:10.1023/A:1006171315513.

[17] Karel Chvalovský, Jan Jakubův, Martin Suda & Josef Urban (2019): *ENIGMA-NG: Efficient Neural and Gradient-Boosted Inference Guidance for E*. In: *Lecture Notes in Computer Science*, Springer International Publishing, pp. 197–215, doi:10.1007/978-3-030-29436-6_12.

[18] Simon Colton (2002): *The HR Program for Theorem Generation*. In Andrei Voronkov, editor: *Automated Deduction—CADE-18*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 285–289, doi:10.1007/3-540-45620-1_24.

[19] Simon Colton, Alan Bundy & Toby Walsh (2000): *On the notion of interestingness in automated mathematical discovery*. *International Journal of Human-Computer Studies* 53(3), pp. 351–375, doi:10.1006/ijhc.2000.0394.

[20] Richard A De Millo, Richard J Lipton & Alan J Perlis (1979): *Social processes and proofs of theorems and programs*. *Communications of the ACM* 22(5), pp. 271–280, doi:10.1145/359104.359106.

[21] Bruno D'Amore & MI Fandiño Pinilla (2016): *Una formula per la misurazione oggettiva della difficoltà di comprensione di un testo di matematica da parte degli studenti. Uso valutativo e uso didattico*. *La matematica e la sua didattica, 1* 2, pp. 59–78.

[22] Sólrún Halla Einarsdóttir, Nicholas Smallbone & Moa Johansson (2020): *Template-based Theory Exploration: Discovering Properties of Functional Programs by Testing*. In: *IFL 2020: Proceedings of the 32nd Symposium on Implementation and Application of Functional Languages*, ACM, doi:10.1145/3462172.3462192.

[23] Ludovic Font (2021): *Génération automatique de preuves pour un logiciel tuteur en géométrie*. phdthesis, Polytechnique Montréal. Available at `https://publications.polymtl.ca/9090/`.

[24] Ludovic Font, Philippe R. Richard & Michel Gagnon (2018): *Improving QED-Tutrix by Automating the Generation of Proofs*. In Pedro Quaresma & Walther Neuper, editors: *Proceedings 6th International Workshop on Theorem proving components for Educational software, Gothenburg, Sweden, 6 Aug 2017*, *Electronic Proceedings in Theoretical Computer Science* 267, Open Publishing Association, pp. 38–58, doi:10.4204/EPTCS.267.3.

[25] Hongbiao Gao & Jingde Cheng (2015): *An epistemic programming approach for automated theorem finding*. In: *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCIxCC)*, IEEE, doi:10.1109/ICCI-CC.2015.7259365.

[26] Hongbiao Gao & Jingde Cheng (2017): *Measuring Interestingness of Theorems in Automated Theorem Finding by Forward Reasoning: A Case Study in Peano's Arithmetic*. In Ngoc Thanh Nguyen, Satoshi Tojo, Le Minh Nguyen & Bogdan Trawinski, editors: *Intelligent Information and Database Systems*, *Lecture Notes in Computer Science* 10192, Springer International Publishing, pp. 115–124, doi:10.1007/978-3-319-54430-4_12.

[27] Hongbiao Gao, Yuichi Goto & Jingde Cheng (2014): *A systematic methodology for automated theorem finding*. *Theoretical Computer Science* 554, pp. 2–21, doi:10.1016/j.tcs.2014.06.028.

[28] Hongbiao Gao, Yuichi Goto & Jingde Cheng (2015): *A Set of Metrics for Measuring Interestingness of Theorems in Automated Theorem Finding by Forward Reasoning: A Case Study in NBG Set Theory*. In: *Intelligence Science and Big Data Engineering. Big Data and Machine Learning Techniques*, Springer International Publishing, pp. 508–517, doi:10.1007/978-3-319-23862-3_50.

[29] Hongbiao Gao, Jianbin Li & Jingde Cheng (2018): *Measuring Interestingness of Theorems in Automated Theorem Finding by Forward Reasoning: A Case Study in Tarski's Geometry*. In: *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, IEEE, doi:10.1109/SmartWorld.2018.00064.

[30] Hongbiao Gao, Jianbin Li & Jingde Cheng (2019): *Measuring Interestingness of Theorems in Automated Theorem Finding by Forward Reasoning Based on Strong Relevant Logic*. In: *2019 IEEE International Conference on Energy Internet (ICEI)*, IEEE, pp. 356–361, doi:10.1109/ICEI.2019.00069.

[31] Thibault Gauthier, Cezary Kaliszyk, Josef Urban, Ramana Kumar & Michael Norrish (2021): *TacticToe: Learning to Prove with Tactics*. Journal of Automated Reasoning 65(2), pp. 257–286, doi:10.1007/s10817-020-09580-x.

[32] Gila Hanna, David Reid & Michael de Villiers, editors (2019): *Proof Technology in Mathematics Research and Teaching*. Springer, doi:10.1007/978-3-030-28483-1.

[33] Kirsti Hemmi, Erika Julin & Ray Pörn (2017): *Misconceptions and developmental proof*. In: CERME 10.

[34] Moa Johansson & Nicholas Smallbone (2021): *Automated Conjecturing in QuickSpec*. In: 1st Mathematical Reasoning in General Artificial Intelligence Workshop, ICLR 2021.

[35] Zoltán Kovács, Tomás Recio & M. Pilar Vélez (2021): *Automated reasoning tools in GeoGebra discovery*. ACM Communications in Computer Algebra 55(2), pp. 39–43, doi:10.1145/3493492.3493495.

[36] Zoltán Kovács & Jonathan H. Yu (2022): *Automated Discovery of Geometrical Theorems in GeoGebra*. In João Marcos, Walther Neuper & Pedro Quaresma, editors: Proceedings 10th International Workshop on Theorem Proving Components for Educational Software, Electronic Proceedings in Theoretical Computer Science 354, pp. 1–12, doi:10.4204/EPTCS.354.1.

[37] Fou-Lai Lin, Feng-Jui Hsieh, Gila Hanna & Michael de Villiers, editors (2009): *Proceedings of the ICMI Study 19 conference: Proof and Proving in Mathematics Education*. 1, The Department of Mathematics, National Taiwan Normal University.

[38] Fou-Lai Lin, Feng-Jui Hsieh, Gila Hanna & Michael de Villiers, editors (2009): *Proceedings of the ICMI Study 19 conference: Proof and Proving in Mathematics Education*. 2, The Department of Mathematics, National Taiwan Normal University.

[39] Donald MacKenzie (1995): *The automation of proof: A historical and sociological exploration*. IEEE Annals of the History of Computing 17(3), pp. 7–29, doi:10.1109/85.397057.

[40] Donald MacKenzie (1999): *Slaying the Kraken: The sociohistory of a mathematical proof*. Social studies of science 29(1), pp. 7–60, doi:10.1177/030631299029001002.

[41] R. L. McCasland, A. Bundy & P. F. Smith (2017): *MATHsAiD: Automated mathematical theory exploration*. Applied Intelligence 47(3), pp. 585–606, doi:10.1007/s10489-017-0954-8.

[42] Xicheng Peng, Qihang Chen, Jingzhong Zhang & Mao Chen (2021): *Automated Discovery of Geometric Theorems Based on Vector Equations*. Journal of Automated Reasoning 65(6), pp. 711–726, doi:10.1007/S10817-021-09591-2.

[43] D. A. Plaisted (1980): *Abstraction mappings in mechanical theorem proving*. In: 5th Conference on Automated Deduction Les Arcs, France, July 8–11, 1980, Springer Berlin Heidelberg, pp. 264–280, doi:10.1007/3-540-10009-1_21.

[44] David A. Plaisted (1994): *The search efficiency of theorem proving strategies*. In: Automated Deduction — CADE-12, Springer Berlin Heidelberg, pp. 57–71, doi:10.1007/3-540-58156-1_5.

[45] George Polya (2004): *How to solve it: A new aspect of mathematical method*. 246, Princeton university press.

[46] Yury Puzis, Yi Gao & G. Sutcliffe (2006): *Automated Generation of Interesting Theorems*. In: FLAIRS Conference.

[47] Markus N. Rabe & Christian Szegedy (2021): *Towards the Automatic Mathematician*. In André Platzer & Geoff Sutcliffe, editors: Automated Deduction – CADE 28, Springer International Publishing, pp. 25–37, doi:10.1007/978-3-030-79876-5_2.

[48] T. Recio & M. P. Vélez (1999): *Automatic Discovery of Theorems in Elementary Geometry*. J. Autom. Reason. 23, pp. 63–82, doi:10.1023/A:1006135322108. Available at `http://dl.acm.org/citation.cfm?id=594128.594243`.

[49] H. G. Rice (1953): *Classes of recursively enumerable sets and their decision problems*. Transactions of the American Mathematical Society 74(2), pp. 358–366, doi:10.2307/1990888.

[50] Hartley Rogers Jr (1987): *Theory of recursive functions and effective computability*. MIT press.

[51] Michael Sipser (1997): *Introduction to the Theory of Computation*. PWS Publishing Company.

[52] Andreas J Stylianides & Guershon Harel (2018): *Advances in mathematics education research on proof and proving: An international perspective*. Springer, doi:10.1007/978-3-319-70996-3.

[53] Geoff Sutcliffe (2017): *The TPTP Problem Library and Associated Infrastructure*. Journal of Automated Reasoning 59(4), pp. 483–502, doi:10.1007/s10817-017-9407-7.

[54] Philip Todd (2021): *A Method for the Automated Discovery of Angle Theorems*. In: Proceedings of ADG 2021, 352, Open Publishing Association, pp. 148–155, doi:10.4204/EPTCS.352.17.

[55] Andrei Voronkov (2000): *CASC-16-1/2*. Preprint Series CSSPP-4, The University of Manchester. Available at `http://www.s.man.a.uk/preprints/index.html`.

[56] Larry Wos (1988): *Automated Reasoning: 33 Basic Research Problems*. Prentice-Hall.

[57] Larry Wos (1993): *The problem of automated theorem finding*. Journal of Automated Reasoning 10(1), pp. 137–138, doi:10.1007/BF00881868.

[58] Zheng Ye, Shang-Ching Chou & Xiao-Shan Gao (2011): *An Introduction to Java Geometry Expert*. In Thomas Sturm & Christoph Zengler, editors: Automated Deduction in Geometry, Lecture Notes in Computer Science 6301, Springer Berlin Heidelberg, pp. 189–195, doi:10.1007/978-3-642-21046-4_10.

[59] Jian Zhang (1999): *System Description: MCS: Model-Based Conjecture Searching*. In: Automated Deduction — CADE-16, Springer Berlin Heidelberg, pp. 393–397, doi:10.1007/3-540-48660-7_37.

A First Survey—Interesting Theorems

With this survey the goal will be to find the characteristics that make a theorem interesting, or not. A list of questions about geometric theorems found to be interesting, or not interesting.

For an initial pool of expert on the area it is our intention to use the network created for the submission of the COST proposal, *iGEOMXXI*.[10] This survey will be available online, based on an online survey tool.[11]

A.1 Interesting and Why?

A list of situations/explanations about interesting theorems.

[10] OC-2020-1-24509, Building a Networked Environment for Geometric Reasoning (iGEOMXXI), The submitted Action (not funded) focused on the exploration of new paradigms and methodologies for supporting formal reasoning in the field of Geometry. A network of 49 experts from 19 countries.

[11] e.g. *LimeSurvey*, `https://www.limesurvey.org/`

Can you describe in detail a situation (during classes or lectures) in which you have used the adjective interesting applied to a theorem in geometry?

***n*th Situation** ______________________________

____

____

____

Can you explain in detail the reasons why you used the adjective interesting in the first situation?

***n*th Explanation** ______________________________

____

____

A.2 Five Interesting Theorems in Geometry

A list of 5 questions, each about an interesting theorem.

Can you list at least five theorems in geometry that you consider interesting?

Theorem *n* ______________________________

____

____

Can you explain in detail the reason for your choice by listing at least five adjectives that describe characteristics of the previous theorem making it interesting?

____

____

____

A.3 Five Not Interesting Theorems in Geometry

A list of 5 questions, each about a not interesting theorem.

Can you list at least five theorems in geometry that you consider NOT interesting?

Theorem n ______________________________

____

____

Can you explain in detail the reason for your choice by listing at least five adjectives that describe characteristics of the previous theorem making it NOT interesting?

____

____

____

B Second Survey—Characteristics of Interesting Theorems

This survey will only be designed after studying the results of the first survey. The second survey will propose theorems (taken from the first survey) and will provide characteristics (taken from the first survey) for each of them. The survey will ask the participants to express their opinion on characteristics that (presumably) make the theorems interesting or not interesting.

This survey will be available online, based on an online survey tool.[9]

Please express whether you consider the following theorems interesting or not, and why?

Is Theorem n interesting?
☐ YES ☐ NO

Why? Because it has the characteristic A.
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly Agree

Why? Because it has the characteristic B.
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly Agree

Why? Because it has the characteristic C.
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly Agree

Why? Because it has the characteristic D.
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly Agree