A Generator of Geometry Deductive Database Method Provers

Pedro Quaresma[1] Nuno Baeta[1]

[1]University of Coimbra, Portugal (pedro@mat.uc.pt,nmsbaeta@gmail.com)

ADG 2025, 1-2 August, Stuttgart, Germany

A Generator of Geometry Deductive Database Method Provers

The Geometry Automated-Theorem-Provers (GATP) based on the deductive database method use a data-based search strategy to improve the efficiency of forward chaining. An implementation of such a method is expected to be able to efficiently prove a large set of geometric conjectures, producing readable proofs [1, 2].

The *Provers-Generator for the Geometric Deductive Databases Method* (*PG* for short), it is a program that, given a set of rules, generates a prover, an *OGP-GDDM*-prover,[1] for that specific set of rules.

The applications in areas such as education are very important given the possibility, opened by the *PG*, of having a prover, capable of producing readable proofs, adapted to a specific audience [3, 4, 5].

[1] Open Geometry Prover Community Project (OGPcp) https://github.com/opengeometryprover

Why? — Possible Applications of the PG

A rule based theorem prover: an introduction to proofs in secondary schools., Teles, J., Santos, V., and Quaresma, P., EPTCS 375:24-37.

The introduction of automated deduction systems in secondary schools face several bottlenecks. (. . .) the dissonance between the outcomes of GATP and the normal practice of conjecturing and proving in schools is a major barrier to a wider use of such tools in an educational environment.

Choosing an appropriate set of rules and an automated method that can use those rules is a major challenge.

An OGP-GDDM prover

The geometry deductive database method (GDDM) is a synthetic method that uses forward chaining to prove non-trivial geometry theorems efficiently [2].

The goals of the OGP-GDDM prover are: to produce a GATP that is efficient, flexible, with natural language and visual renderings, and implemented as an open source library [1].

Flexibility means:

- implementing the prover as an open source library, not hard-coded inside a given program.
- implement the inference rules as SQL data manipulation language queries.

An OGP-GDDM prover

Our first implementation uses the set of rules described by Chou et al. (2000) [2].[2] The rules were hard-coded as SQL queries

But. . . we realised [1] that *JGEx* did not implement the rules stated in [2].

So:

- a new implementation, for the new set of rules (*JGEx* rules), is needed.
- following the research done in [3, 4, 5] we need many new implementations, one for each set of rules that best adapt to many different learning situations . . .

[2] Chout et al., A deductive database approach to automated geometry theorem proving and discovering

The PG

So. . . instead of "running behind all the set of rules needed", we decided to build the *Provers-Generator* (PG).

The PG is a two-step solution/implementation of the geometric deductive database method:

- from a given set of rules, generate the code of the corresponding GDDM prover;
- compile the generated code, to obtain the GDDM prover.

The PG

To create the prover, a set of inference rules is supplied to PG, that, in turn, will generate the $C{+}{+}$ code necessary to implement the rules, as well as their usage.

After compiling the generated code, the prover is ready to be used.

Given a conjecture in FOF,[3] it will generate the fix-point (the set of all known and derived facts) and, if the hypothesis belongs to the fix-point, the conjecture is proved, otherwise, it is unknown.

[3] To be changed to ADG-Lib

The PG

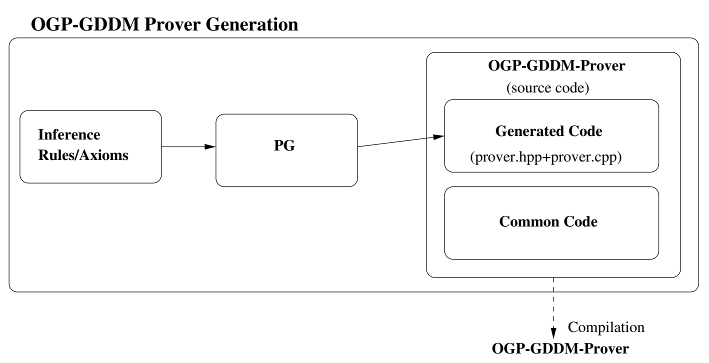

Figure: PG — Generation

Common code: all the source code that is common to all GDDM-type provers, i.e., the reading and parsing of the files, the database management, and the global inference mechanism.

The Generated GDDM Prover

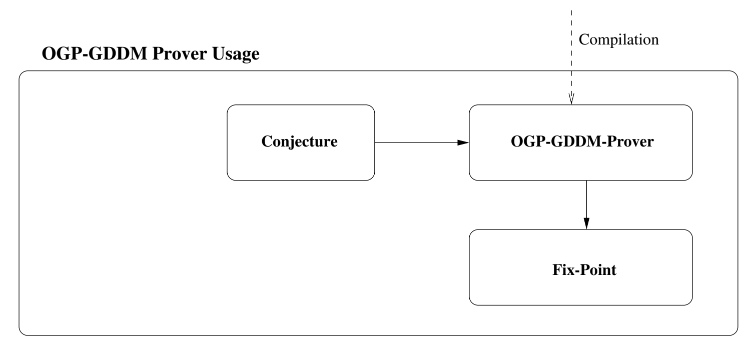

Figure: PG — Usage

The generated GATP is, after compilation, a normal GDDM prover for the specific set of axioms chosen at generation time.

PG — How to Build a GATP

- Download the PG code:

  https://github.com/opengeometryprover/GDDM/tree/master/provers/pg

- Compile the PG code (in provers/pg/src)[4]: $ make

  It will produce the "pg" executable.

- Copy the executable and the axiom set[5] to a new directory and run it: $ > ./pg chou.ax

  It will will produce the source code for the new GATP ("proverchou").

- Compile the "proverchou" code: $ make

  It will produce the "proverchou" executable.

---

[4] Linux: the tools needed are, *make*, *C++* compiler, *Flex* and *Bison*.

[5] `chou.ax` is a file containing a set of axioms [2].

PG — An Example

- An example: In a triangle the base line and the midpoints line are parallel (geo0007.p):

```
fof (geo0007, conjecture, ! [ A, B, C, D, E ] :
    ((midp(D,C,A) & midp(E,A,B)) => para (B,C,E,D))) .
```

- $ ./proverchou geo0007.p

```
proverchou - PG Generated
Copyright (C) 2025 Nuno Baeta, Pedro Quaresma
Distributed under GNU GPL 3.0 or later
Conjecture is PROVED, in: 0.010058s
Fix-point found, in: 125.494s
Fix-point saved to file 'geo0007.fp'.
```

- Fix-point (geo0007.fp):

```
Fix-point
    midp(D, C, A)
    midp(E, A, B)
    midp(D, A, C)
    cong(D, C, D, A)
    coll(D, C, A) ...
```

Future Work

- To be able to generate a prover as a library $\hookrightarrow$ to include the prover within DGSs and any other program.
- To improve efficiency.
- To output an annotated proof-tree.
- To adopt ADG-Lib format.
- To build appropriate sets of axioms.

Future Work

- To be able to generate a prover as a library $\hookrightarrow$ to include the prover within DGSs and any other program.
- To improve efficiency.
- To output an annotated proof-tree.
- To adopt ADG-Lib format.
- To build appropriate sets of axioms.
- To explore an implementation using *datalog*.

Future Work

- To be able to generate a prover as a library $\hookrightarrow$ to include the prover within DGSs and any other program.
- To improve efficiency.
- To output an annotated proof-tree.
- To adopt ADG-Lib format.
- To build appropriate sets of axioms.
- To explore an implementation using *datalog*.

Thank You

Bibliography I

Nuno Baeta and Pedro Quaresma.
Towards a geometry deductive database prover.
Annals of Mathematics and Artificial Intelligence, 91(6):851–863, may 2023.

Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang.
A deductive database approach to automated geometry theorem proving and discovering.
Journal of Automated Reasoning, 25(3):219–246, 2000.

Pedro Quaresma, Vanda Santos, and Joana Teles.
Proof exploration using dynamic geometry systems with integrated automated deduction capabilities.
International Journal of Mathematical Education in Science and Technology, pages 1–25, July 2024.

Bibliography II

Vanda Santos, Joana Teles, and Pedro Quaresma.
Exploring quadrilaterals: An interactive task for 7th grade students using geogebra classroom.
International Journal for Technology in Mathematics Education, 31(3):107–116, September 2024.

Joana Teles, Vanda Santos, and Pedro Quaresma.
A rule based theorem prover: an introduction to proofs in secondary schools.
Electronic Proceedings in Theoretical Computer Science, 375:24–37, mar 2023.