APRiL: a close-to-natural language for geometric proofs

Mikhail Koroteev, Ilya Krukov and Artem Smirnov

Abstract. There are a lot of interactive or automatic theorem provers used by professional computer scientists or mathematicians. However such tools have a steep learning curve and they are not intended for schools. On the other hand, software that is used widely in education lacks proving capabilities or replaces synthetic proofs with algebraic methods which are powerful but useless in education. We are developing a new computer language intended for writing proofs in a close to natural way, powerful enough to express proofs in Euclidean geometry and designed especially to be used in education.

Introduction

Euclidean geometry occupies a special place in education. Usually this is the first (and often the only) mathematical discipline that makes students familiar with proofs and reasoning rules. Being so important, geometry also imposes a special responsibility to teachers: they have to not only introduce proving rules for students, but also check their proofs for validity. This is a tedious and error-prone work that cannot be automated due to absence of formal proof language suitable for schools. Even professionals suffer from lack of a language that is formal enough to express mathematical proofs and at the same time close enough to the natural language to allow easy reading. Leslie Lamport in [1] and in [2] proposes a method of writing proofs that makes it much harder to prove things that are not true. Vladimir Voevodsky in his unfinished paper [3] started the description of a format suitable for both proof verification software and human beings.

There are a lot of proof-related tools for professional mathematicians, computer scientists, and logicians. Unfortunately, none of them can be used in higher school due to a steep learning curve.

A lot of attempts were undertaken to improve the situation: Naproche ([4] and [5]), SAD/ForTheL [7], AGPT ([9] and [10]), and the Incredible Proof Machine [8]. As far as we know, only AGPT was actively used in education with very promising results according to [10] but there is no any sources of this system and we cannot find any publicly available project for evaluation.

There are software tools that are used intensively in geometry teaching. GeoGebra [16] is the most widely known example. They lack the ability to write and check formal proofs. Most often asserted statements checked numerically or verified using algebraic methods. Being mathematically correct, algebraic methods cannot be understood by humans and hence, they are useless for learning how to prove. A notable exception is the CanfigureIt project [17]. They provide a web interface for entering synthetic proofs of geometric facts. The project has nice web interface, looks very promising but public available set of tasks is very limited.

1. Our goals

We faced the necessity in intelligible language for writing geometric proofs during our work on math games such as Euclidea or Pythagorea [18]. We started to develop a proof language that fits our needs. The required language should be:

- Robust and flexible enough to express statements and proofs of Euclidean geometry.
- Formal enough to express facts and proofs with any desired level of rigor.
- ... but allowing users to skip boring details if necessary.
- Natural and comprehensible for non-professionals. People without PhD in computer science should be able to read, understand, and write statements and proofs in the language, without or with minimal training. Moreover, it is highly desirable to keep the language accessible for school students.
- Embeddable and not resource-demanding to work on mobile platforms.

We call our dream language "APRiL" (which is an acronym for "A PRoof Language"). Each requirement from the list above forces specific design decisions.

2. The current state and plans

The development is on an early stage now. We just have a preliminary language design, parser, interpreter prototype and a beta-version of Base library with axioms for 2D Euclidean geometry and most basic definitions. The nearest plans include:

- Complete interpreter and embedded library development.
- Complete base library development.
- Integrate APRiL in our math apps [18].
- Support external first-order logic theorem provers and/or SMT (Satisfiability Modulo Theories) solvers to introduce automatic proving capabilities.

3. Sample proof in APRiL

```
theorem SAS:
[AB] = [A'B'],
[BC] = [B'C'],
Angle(ABC) = Angle(A'B'C')
==>
Triangle(ABC) = Triangle(A'B'C')
end

def IsoscelesTriangle(ABC):
[AB] = [BC]
end

theorem Euclid_p5:
is IsoscelesTriangle(ABC)
==>
Angle(BAC) = Angle(BCA)
end

proof Euclid_p5:
bis: let [BD] is Bisector for (Triangle(ABC)).

Triangle(ABD) = Triangle(CBD) where {
st1: [AB] = [BC] due to is IsoscelesTriangle(ABC).
st2: [BD] = [BD].
st3: Angle(ABD) = Angle(CBD) by bis.
Triangle(ABD) = Triangle(CBD) by theorem SAS(st1, st2, st3).
}

Angle(BAD) = Angle(BCD) by property Triangle(ABD) = Triangle(CBD).
Angle(BAC) = Angle(BCA).
end
```

References

[1] Leslie Lamport, *How to Write a Proof.* 1993.

[2] Leslie Lamport, *How to Write a 21st Century Proof.* 2011.

[3] Vladimir Voevodsky, *On an approach to conveniently formalize mathematics.* 2006.

[4] Kühlwein, Daniel and Cramer, Marcos and Koepke, P and Schroder, Bernhard, *The naproche system.* 2009.

[5] Cramer M.,Fisseni B., Koepke P. and others, *The Naproche Project Controlled Natural Language Proof Checking of Mathematical Texts.* 2010.

[6] Fuchs, Norbert and Höfler, Stefan and Kaljurand, Kaarel and Rinaldi, Fabio and Schneider, Gerold, *Attempto Controlled English: A Knowledge Representation Language Readable by Humans and Machines.* 2005.

[7] Andrei Paskevich, *The syntax and semantics of the ForTheL language.* 2007.

[8] Breitner J., emphVisual Theorem Proving with the Incredible Proof Machine. 2016.

[9] Wang, Ke and Su, Zhendong, *Automated Geometry Theorem Proving for Human-readable Proofs.* 2015.

[10] Wang, Ke and Su, Zhendong, *Interactive, Intelligent Tutoring for Auxiliary Constructions in Geometry Proofs.* 2017.

[11] Michael Beeson and Julien Narboux and Freek Wiedijk, *Proof-checking Euclid.* 2017.

[12] Michael Beeson, *Constructive Geometry and the Parallel Postulate.* 2014.

[13] Michael Beeson, *A Constructive Version of Tarski's Geometry.* 2014.

[14] Julien Narboux, *A graphical user interface for formal proofs in geometry.* 2007.

[15] Julien Narboux, *Mechanical theorem proving in Tarski's geometry.* 2007.

[16] Hohenwarter M. and Preiner J, *Dynamic Mathematics with GeoGebra.* 2007.

[17] *CanFigureIt® Geometry.* `https://www.canfigureit.com/`

[18] *Euclidea.* `https://www.euclidea.xyz`

[19] Beeson, Michael and Boutry, Pierre and Braun, Gabriel and Gries, Charly and Narboux, Julien, *GeoCoq.* 2018.

Mikhail Koroteev
Horis International Ltd
Saint-Petersburg, Russia
e-mail: `mk@hil-hk.com`

Ilya Krukov
Horis International Ltd
Saint-Petersburg, Russia
e-mail: `ik@hil-hk.com`

Artem Smirnov
Horis International Ltd
Saint-Petersburg, Russia
e-mail: `as@hil-hk.com`