The GeoCoq library and its porting to Isabelle

Roland Coghetto   Julien Narboux

September 2022

EuroProofNet Workshop on the development, maintenance, refactoring and search of large libraries of proofs

Overview of the talk

- What is GeoCoq ?
- Discuss questions important for porting GeoCoq to other proof assistants
  - Classical of constructive logic ?
  - First-order or Higher-order logic ?
  - How automation is used ?
- Report on experiments about porting GeoCoq to other proof assistants.

1. Overview of GeoCoq
   - Foundations
   - Two formalizations of the Elements
   - Arithmetization of Geometry
   - 34 parallel postulates
   - Technical aspects
2. What features GeoCoq uses ?
   - First-order vs higher-order logic
   - Constructive or classical Logic ?
   - Automation ?
3. Porting GeoCoq to other proof assistants
   - Automatically: The Elements in Dedukti
   - Manually: IsaGeoCoq

Why use GeoCoq as a test case for proof translation ?

- Euclid's Elements is an influential work in the history of maths.
- An interesting fragment of GeoCoq: a formalization of Euclid's book 1 is using very few features: no inductive type, no fixpoint, no reflexivity, no computations, morally first-order.

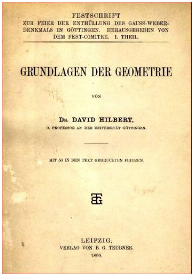
Outline

1. Overview of GeoCoq
   - Foundations
   - Two formalizations of the Elements
   - Arithmetization of Geometry
   - 34 parallel postulates
   - Technical aspects
2. What features GeoCoq uses ?
3. Porting GeoCoq to other proof assistants

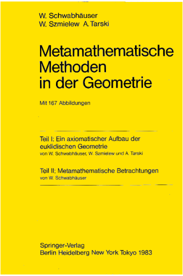
GeoCoq

- An Open Source library about foundations of geometry
- Contributors: Michael Beeson, Gabriel Braun, Pierre Boutry, Charly Gries, Julien Narboux, Pascal Schreck
- Size: $>$ 3900 Lemmas, $>$ 130000 lines
- License: LGPL3

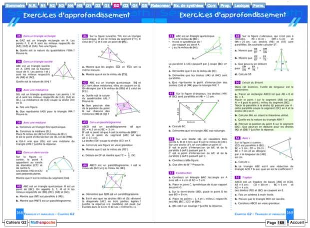
Euclide



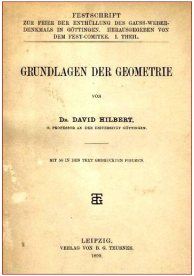
Hilbert



Tarski



Exercises


FESTSCHRIFT
ZUR FEIER DER ENTHÜLLUNG DES GAUSS-WEBER-DENKMALS IN GÖTTINGEN. HERAUSGEGEBEN VON DEM FEST-COMITEE. I. THEIL.

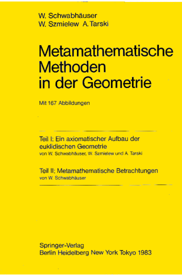
GRUNDLAGEN DER GEOMETRIE

VON

DR. DAVID HILBERT,
O. PROFESSOR AN DER UNIVERSITÄT GÖTTINGEN.

MIT 50 IN DEN TEXT GEDRUCKTEN FIGUREN.

LEIPZIG,
VERLAG VON B. G. TEUBNER.
1899.


W. Schwabhäuser
W. Szmielew A. Tarski

Metamathematische Methoden in der Geometrie

Mit 167 Abbildungen

Teil I: Ein axiomatischer Aufbau der euklidischen Geometrie
von W. Schwabhäuser, W. Szmielew und A. Tarski

Teil II: Metamathematische Betrachtungen
von W. Schwabhäuser

Springer-Verlag
Berlin Heidelberg New York Tokyo 1983

Euclide Hilbert Tarski

What we have:

Axiom systems Tarski's, Hilbert's, Euclid's and variants.

Foundations In arbitrary dimension, in neutral geometry. Betweenness, Two-sides, One-side, Collinearity, Midpoint, Symmetric point, Perpendicularity, Parallelism, Angles, Co-planarity, . . .

Classic theorems Pappus, Pythagoras, Thales' intercept theorem, Thales' circle theorem, nine point circle, Euler line, orthocenter, circumcenter, incenter, centroid, quadrilaterals, Sum of angles, Varignon's theorem, . . .

Arithmetization Coordinates and possibility to use Gröbner basis.

An Euclidean model of Tarski's and Hilbert's axioms using Pythagorean ordered field

High-school Some exercises

What is missing:

- Consequence of continuity: trigonometry, areas
- Model of equal-area axioms (but available in HOL-Light !)
- Model of hyperbolic geometry (but available in Isabelle !)
- Complex geometry (but available in Isabelle !)

Foundations of geometry

1. Synthetic geometry
2. Analytic geometry
3. Metric geometry
4. Transformations based approaches

Synthetic approach

Assume some undefined geometric objects + geometric predicates + axioms . . .
The name of the assumed types are not important.

- Hilbert's axioms:

  types: points, lines and planes
  predicates: incidence, between, congruence of segments, congruence of angles

- Tarski's axioms:

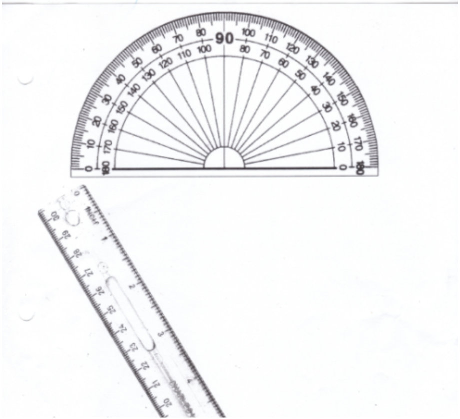
  types: points
  prédicats: between, congruence

- . . . many variants

Analytic approach

We assume we have numbers (a field $\mathbb{F}$).
We define geometric objects by their coordinates.
Points := $\mathbb{F}^n$

Metric approach



Compromise between synthetic and metric approach.
We assume both:

- numbers (a field)
- geometric objects
- axioms

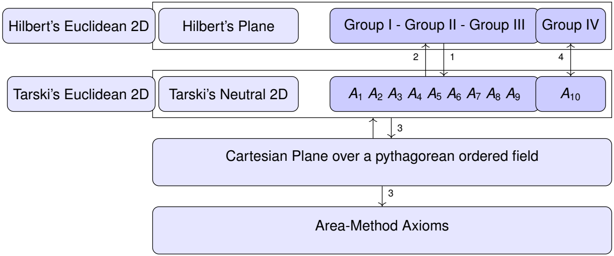
- Birkhoff's axioms: points, lines, reals, ruler and protractor
- Chou-Gao-Zhang's axioms: points, numbers, three geometric quantities

Transformation groups

Erlangen program. Foundations of geometry based on group actions and invariants.

Felix Klein

Overview of the axiom systems



[1] Gabriel Braun, Pierre Boutry, and Julien Narboux (June 2016). "From Hilbert to Tarski". In: Eleventh International Workshop on Automated Deduction in Geometry. Proceedings of ADG 2016

[2] Gabriel Braun and Julien Narboux (Sept. 2012). "From Tarski to Hilbert". English. In: Post-proceedings of Automated Deduction in Geometry 2012. Vol. 7993. LNCS

[3] Pierre Boutry, Gabriel Braun, and Julien Narboux (2019). "Formalization of the Arithmetization of Euclidean Plane Geometry and Applications". In: Journal of Symbolic Computation 98

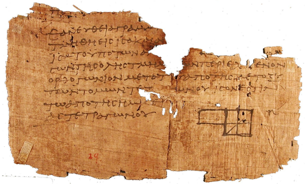
[4] Pierre Boutry et al. (2017). "Parallel postulates and continuity axioms: a mechanized study in intuitionistic logic using Coq". In: Journal of Automated Reasoning

Outline

1. Overview of GeoCoq
   - Foundations
   - Two formalizations of the Elements
   - Arithmetization of Geometry
   - 34 parallel postulates
   - Technical aspects
2. What features GeoCoq uses ?
3. Porting GeoCoq to other proof assistants

The Elements

- A very influential mathematical book (more than 1000 editions).
- First known example of an axiomatic approach.



Book 2, Prop V, Papyrus d'Oxyrhynchus (year 100)

Euclid

First project

- Joint work with Charly Gries and Gabriel Braun
- Mechanizing proofs of Euclid's **statements**
- Not Euclid's proofs!
- Trying to minimize the assumptions:
  - Parallel postulate
  - Elementary continuity
  - Archimedes' axiom

Second project

- Joint work with Michael Beeson and Freek Wiedijk [5]
- Formalizing Euclid's **proofs**
- A not minimal axiom system
- Filling the gaps in Euclid

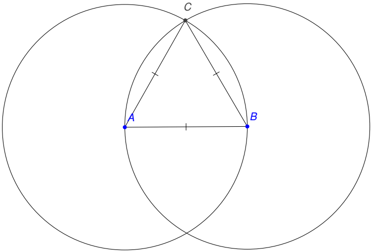
[5] Michael Beeson, Julien Narboux, and Freek Wiedijk (2019). "Proof-checking Euclid". In: Annals of Mathematics and Artificial Intelligence 85.2-4

Example

Proposition (Book I, Prop 1)

Let A and B be two points, build an equilateral triangle on the base AB.

Proof: Let $\mathcal{C}_1$ and $\mathcal{C}_2$ the circles of center A and B and radius AB. Take C at the intersection of $\mathcal{C}_1$ and $\mathcal{C}_2$. The distance AB is congruent to AC, and AB is congruent to BC. Hence, ABC is an equilateral triangle.

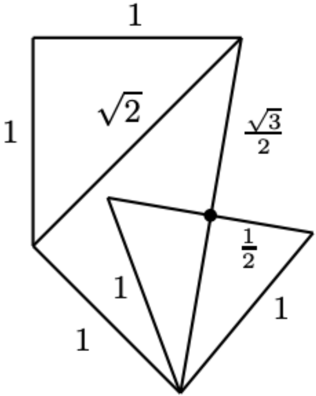


Book I, Prop 1

In the spirit of *reverse mathematics*, we proved two statements:

1. Assuming no continuity, but the parallel postulate (solving a challenge proposed by Beeson)[6].
2. Assuming circle/circle continuity, but not the parallel postulate (trivial).

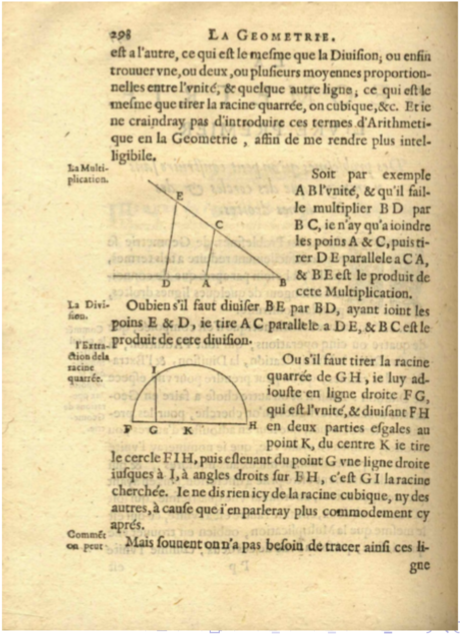
Pambuccian has shown that these assumptions are minimal.



[6] Michael Beeson (2013). "Proof and Computation in Geometry". In: Automated Deduction in Geometry (ADG 2012). Vol. 7993. Springer Lecture Notes in Artificial Intelligence

Arithmetization of Geometry

René Descartes (1925).
La géométrie.



298 LA GEOMETRIE.

eſt a l'autre, ce qui eſt le meſme que la Diuiſion; ou enfin trouuer vne, ou deux, ou pluſieurs moyennes proportionnelles entre l'vnité, & quelque autre ligne; ce qui eſt le meſme que tirer la racine quarrée, on cubique, &c. Et ie ne craindray pas d'introduire ces termes d'Arithmetique en la Geometrie, affin de me rendre plus intelligibile.

La Multiplication.

Soit par exemple A B l'vnité, & qu'il faille multiplier B D par B C, ie n'ay qu'a ioindre les poins A & C, puis tirer D E parallele a C A, & B E eſt le produit de cete Multiplication.

La Diuiſion.

Oubien s'il faut diuiſer B E par B D, ayant ioint les poins E & D, ie tire A C parallele a D E, & B C eſt le produit de cete diuiſion.

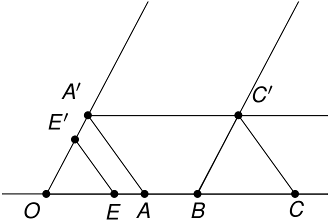
l'Extraction dela racine quarrée.

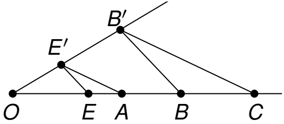
Ou s'il faut tirer la racine quarrée de G H, ie luy adiouſte en ligne droite F G, qui eſt l'vnité, & diuiſant F H en deux parties eſgales au point K, du centre K ie tire le cercle F I H, puis eſleuant du point G vne ligne droite iuſques à I, à angles droits ſur F H, c'eſt G I la racine cherchée. Ie ne dis rien icy de la racine cubique, ny des autres, à cauſe que i'en parleray plus commodement cy aprés.

Commét on peut

Mais ſouuent on n'a pas beſoin de tracer ainſi ces ligne

Addition and multiplication





Algebra/Geometry

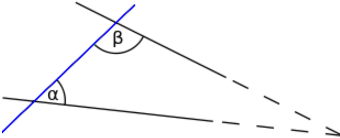
Continuity	Axiom
	ordered Pythagorean field[7]
circle/line continuity	ordered Euclidean field [8]
FO Dedekind cuts	real closed field [9]
Dedekind	reals

[7] the sum of squares is a square
[8] positive are square
[9] F is euclidean and every polynomial of odd degree has at least one root in F.

Outline

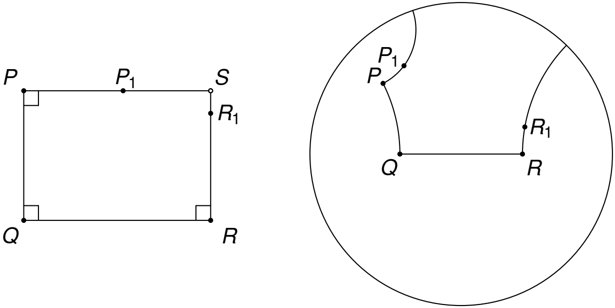
Euclid 5th postulate

"If two lines are drawn which intersect a third in such a way that the sum of the inner angles on one side is less than two right angles, then the two lines inevitably must intersect each other on that side if extended far enough."

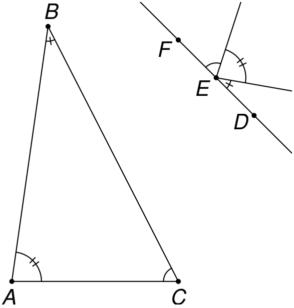


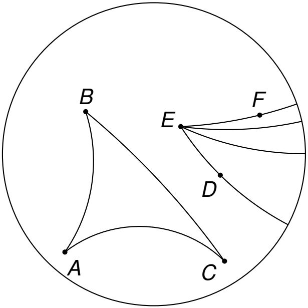
Bachmann's Lotschnittaxiom

If $p \perp q$, $q \perp r$ and $r \perp s$ then p and s meet.

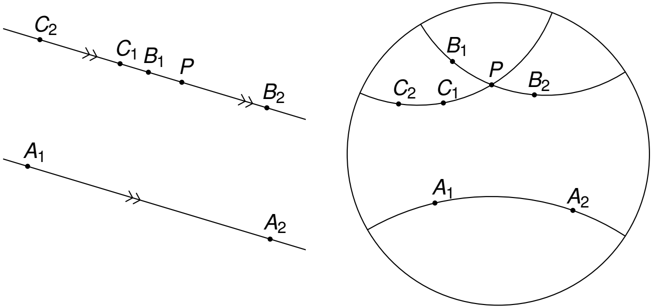


Triangle postulate

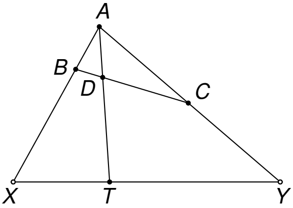


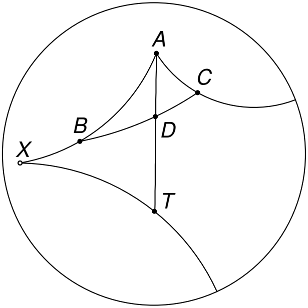


Playfair's postulate

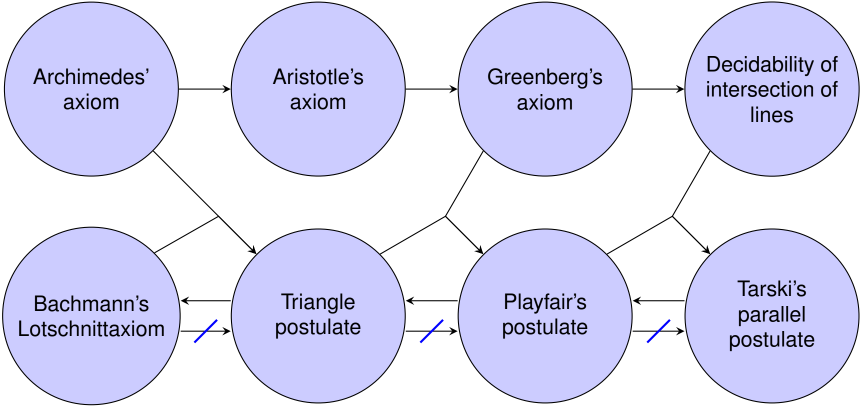


Tarski's postulate

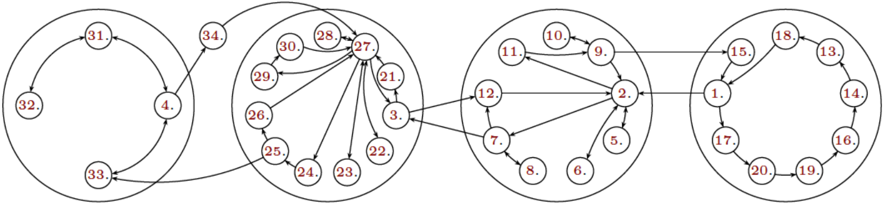




Four groups


Archimedes' axiom
Aristotle's axiom
Greenberg's axiom
Decidability of intersection of lines
Bachmann's Lotschnittaxiom
Triangle postulate
Playfair's postulate
Tarski's parallel postulate

Sorting 34 postulates



10

---

[10] Pierre Boutry et al. (2017). "Parallel postulates and continuity axioms: a mechanized study in intuitionistic logic using Coq". In: Journal of Automated Reasoning

An ”axiom free” development

Axiom = global variable

```
Class Tarski_neutral_dimensionless :=
{
 Tpoint : Type;
 Bet : Tpoint -> Tpoint -> Tpoint -> Prop;
 Cong : Tpoint -> Tpoint -> Tpoint -> Tpoint -> Prop;
 cong_pseudo_reflexivity : forall A B, Cong A B B A;
 cong_inner_transitivity : forall A B C D E F,
   Cong A B C D -> Cong A B E F -> Cong C D E F;
 cong_identity : forall A B C, Cong A B C C -> A = B;
 segment_construction : forall A B C D,
   exists E, Bet A B E /\ Cong B E C D;
   ...
```

Then, we can also formalize some meta-theoretical results:

”Equivalence” between axiom systems:

```
Instance Hilbert_euclidean_follows_from_Tarski_euclidean :
  Hilbert_euclidean
  Hilbert_neutral_follows_from_Tarski_neutral.
```

Outline

First-order vs Higher-order logic in GeoCoq

- Formally all proofs are higher-order: forall predicates ”cong” and ”bet” verifying the axioms, if . . . then . . .
- But many proofs a locally first-order (if we assume the axioms to be in the context).
- Tarski’s axiom system is meant to be expressed in FOL.

Use of higher-order logic

- Meta-theoretical results
- In the proof of Pappus' theorem [11]: the concept of class of equivalence of congruent segments is used. Gödel tells us there is a first-order proof, but can we obtain it automatically using normalization ?
- Continuity axioms

[11] Gabriel Braun and Julien Narboux (Feb. 2017). "A synthetic proof of Pappus' theorem in Tarski's geometry". In: Journal of Automated Reasoning 58.2

Hilbert's line completeness

Axiom V.2: "An extension (An extended line from a line that already exists, usually used in geometry) of a set of points on a line with its order and congruence relations that would preserve the relations existing among the original elements as well as the fundamental properties of line order and congruence that follows from Axioms I-III and from V-1 is impossible."

> *Hilbert's own completeness axiom, added in other editions as V-2, takes the somewhat awkward form of requiring that it be impossible to properly extend the sets and relations satisfying the other axioms so that all the other axioms still hold.*
>
> – Martin 1998, p. 175

Formalization in Coq

We need to quantify over models of other axioms[12] :

```
Definition completeness_for_planes := forall
  (Tm: Tarski_neutral_dimensionless)
  (Tm2 : Tarski_neutral_dimensionless_with_decidable_p
  (M : Tarski_2D Tm2)
  (f : @Tpoint Tn -> @Tpoint Tm),
  @archimedes_axiom Tm ->
  extension f ->
  forall A, exists B, f B = A.
```

[12] Charly Gries, Julien Narboux, and Pierre Boutry (Jan. 2019). “Axiomes de continuité en géométrie neutre : une étude mécanisée en Coq”. In: Journées Francophones des Langages Applicatifs 2019. Acte des Journées Francophones des Langages Applicatifs (JFLA 2019)

Constructive of classical logic ?

Intuitionist logic [13]

- Assuming : $\forall A, B$: Points, $A = B \vee A \neq B$
- We prove : excluded middle for all other predicates,

[13] Pierre Boutry et al. (July 2014). "A short note about case distinctions in Tarski's geometry". In: Proceedings of the 10th Int. Workshop on Automated Deduction in Geometry. Vol. TR 2014/01. Proceedings of ADG 2014

Constructive of classical logic ?

Intuitionist logic [13]

- Assuming : $\forall A, B$: Points, $A = B \vee A \neq B$
- We prove : excluded middle for all other predicates, except line intersection

[13] Pierre Boutry et al. (July 2014). "A short note about case distinctions in Tarski's geometry". In: Proceedings of the 10th Int. Workshop on Automated Deduction in Geometry. Vol. TR 2014/01. Proceedings of ADG 2014

Use of automation in GeoCoq

1. Standard automation
2. Reflexive tactics
3. Gröbner bases

Automatic proof of Col and Coplanar properties

We use a reflexive tactic to prove some transitivity properties of collinearity and coplanarity[14].

[14] Pierre Boutry, Julien Narboux, and Pascal Schreck (Oct. 2015). "A reflexive tactic for automated generation of proofs of incidence to an affine variety".

Characterization of geometric predicates

Geometric predicate	Characterization			
$AB \equiv CD$	$(x_A - x_B)^2 + (y_A - y_B)^2 - (x_C - x_D)^2 + (y_C - y_D)^2$	$=$	0	
Bet $A\ B\ C$	$\exists t, 0 \leq t \leq 1 \wedge$ $t(x_C - x_A) = x_B - x_A$			$\wedge$
	$t(y_C - y_A) = y_B - y_A$			
$Col\ A\ B\ C$	$(x_A - x_B)(y_B - y_C) - (y_A - y_B)(x_B - x_C)$	$=$	0	
I midpoint of AB	$2x_I - (x_A + x_B)$	$=$	0	$\wedge$
	$2y_I - (y_A + y_B)$	$=$	0	
$PerABC$	$(x_A - x_B)(x_B - x_C) + (y_A - y_B)(y_B - y_C)$	$=$	0	
$AB \parallel CD$	$(x_A - x_B)(x_C - x_D) + (y_A - y_B)(y_C - y_C)$	$=$	0	$\wedge$
	$(x_A - x_B)(x_A - x_B) + (y_A - y_B)(y_A - y_B)$	$\neq$	0	$\wedge$
	$(x_C - x_D)(x_C - x_D) + (y_C - y_D)(y_C - y_D)$	$\neq$	0	
$AB \perp CD$	$(x_A - x_B)(y_C - y_D) - (y_A - y_B)(x_C - x_D)$	$=$	0	$\wedge$
	$(x_A - x_B)(x_A - x_B) + (y_A - y_B)(y_A - y_B)$	$\neq$	0	$\wedge$
	$(x_C - x_D)(x_C - x_D) + (y_C - y_D)(y_C - y_D)$	$\neq$	0	

Formalization technique: bootstrapping

Manually bet, cong, equality, col

Automatically midpoint, right triangles, parallelism and perpendicularity

Automation

Using Gröbner's bases, but this is not a theorem about polynomials:

```
Lemma centroid_theorem : forall A B C A1 B1 C1 G,
  Midpoint A1 B C ->
  Midpoint B1 A C ->
  Midpoint C1 A B ->
  Col A A1 G ->
  Col B B1 G ->
  Col C C1 G \/ Col A B C.
Proof.
intros A B C A1 B1 C1 G; convert_to_algebra; decompose_coordinates.
intros; spliter. express_disj_as_a_single_poly; nsatz.
Qed.
```

Outline

Euclid in Dedukti

Formalization of Euclid Book 1: 238 lemmas, 20klocs (15% of GeoCoq).
Features: no inductive, no fixpoint, no reflexivity, first-order proofs, simple tactics.
Yoan Geran has exported our formalization of Euclid/Book 1 to: Coq, HOL-Light, Lean, Matita, PVS and Open Theory
`https://github.com/Karnaj/sttfa_geocoq_euclid`
The (compressed) size of the translated proofs are multiplied by 10 (Lean, Matita, Coq), 25 (Hol-Light) and 50 (PVS).

IsaGeoCoq

Roland Coghetto started to port GeoCoq to Isabelle. First version is available in AFP since 2021 (22klocs):
`https://www.isa-afp.org/entries/IsaGeoCoq.html`
The second version is in preparation:
`https://github.com/CoghettoR/IsaGeoCoq2_R1` contains 2850 lemmas, 18 locales and 92klocs making it one of the largest Isabelle contributions (roughly 75% of GeoCoq).

The approach

- Port all the statements.
- Try proving them automatically using Sledgehammer.
- If it fails, introduce intermediate statements taken from the Coq formalization (mainly existential statements) and repeat.

This approach is also advocated by Yacine El Haddad for verifying TSTP proof traces [15].

[15] Yacine El Haddad (Sept. 2021). "Integrating Automated Theorem Provers in Proof Assistants". en. PhD thesis. Université Paris-Saclay

Results

- 56% of the first 1500 propositions can be solved
- but sledgehammer fails more on the rest of GeoCoq: proofs involving inductive predicates, coplanarity, longer proofs
- 2476 goals solved by Metis, 1524 by Meson, 20 proofs reconstructed in Isar.
- Roland chose to formalize Archimedes property differently: using integers rather than an inductive predicate.
- Roland added about 200 lemmas not present in GeoCoq, mostly about Archimedes and proves that his version is equivalent to ours.
- Parts not ported: arithmetization (hard), continuity (hard), Elements (easy)

Perspectives

- Align axiom systems with other AFP entries:
  - Poincaré Disk model by Danijela Simić, Filip Marić and Pierre Boutry
  - The Independence of Tarski's Euclidean Axiom by T. J. M. Makarios

Conclusion

Difficulties:

- Ad-hoc tactics can not be replaced by general purpose automation, and tactics are hard to port, proof traces maybe be too long.

Opportunities:

- Having GeoCoq in Isabelle and other proof assistants can be interesting for applications in Robotics and Education.
- For training AI we have a large contribution reasonably aligned between Isabelle and Coq (+ some pieces in Mizar/Metamath/TPTP).

Risk:

- There is a high risk that the contributions will diverge, some developments produced for a proof assistant, won't be ported to the other one.

Thank you

A proof using several proof assistants

Several persons have tried to mechanize Tarski's geometry, either automatically or interactively. As an example we show different versions of the proof of Lemma 2.11.

- In GeoCoq
- In IsaGeoCoq version 1
- In IsaGeoCoq version 2
- Mizar
- Metamath
- Automatic proof using otter

GeoCoq

```
    Lemma l2_11 : forall A B C A' B' C',
 Bet A B C -> Bet A' B' C' -> Cong A B A' B' ->
 Cong B C B' C' ->
 Cong A C A' C'.
Proof.
    intros.
    induction (eq_dec_points A B).
      subst B.
      assert (A' = B') by
     (apply (cong_identity A' B' A); Cong).
      subst; Cong.
    apply cong_commutativity; apply (five_segment A A' B B' C C' A 
Qed.
```

IsaGeoCoq V1

```
lemma l2_11:
  assumes "Bet A B C" and
    "Bet A′ B′ C′" and
    "Cong A B A′ B′" and
    "Cong B C B′ C′"
  shows "Cong A C A′ C′"
  by (smt assms(1) assms(2) assms(3) assms(4) cong_right_commutativ
```

IsaGeoCoq V2

```
lemma l2_11:
  assumes "Bet A B C" and
    "Bet A' B' C'" and
    "Cong A B A' B'" and
    "Cong B C B' C'"
  shows "Cong A C A' C'"
proof cases
  assume "A = B"
  thus ?thesis
    using assms(3) assms(4) cong_reverse_identity by blast
next
  assume "A <> B"
  thus ?thesis
    using five_segment Tarski_neutral_dimensionless_axioms assms(1)
      cong_commutativity cong_trivial_identity by blast
qed
```

Isabelle - Makarios

```
theorem th2_11:
    assumes hypotheses:
      "B a b c"
      "B a' b' c'"
      "a b \<congruent> a' b'"
      "b c \<congruent> b' c'"
    shows "a c \<congruent> a' c'"
  proof cases
    assume "a = b"
    with <a b \<congruent> a' b'> have "a' = b'" by (simp add: A3_reversed)
    with <b c \<congruent> b' c'> and <a = b> show ?thesis by simp
  next
    assume "a <> b"
    moreover
      note A5' [of a b c a a' b' c' a'] and
        unordered_pair_equality [of a c] and
        unordered_pair_equality [of a' c']
    moreover
      from OFS_def [of a b c a a' b' c' a'] and
          hypotheses and
          th2_8 [of a a'] and
          unordered_pair_equality [of a b] and
          unordered_pair_equality [of a' b']
        have "OFS a b c a a' b' c' a'" by (simp add: C_SC_equiv)
    ultimately show ?thesis by (simp add: C_SC_equiv)
  qed
```

Mizar

```
theorem Satz2p11: ::GTARSKI1:24
  between a,b,c & between a9,b9,c9 & a,b equiv a9,b9 & b,c equiv b9
  implies a,c equiv a9,c9
  proof
    assume
A1: between a,b,c & between a9,b9,c9 & a,b equiv a9,b9 & b,c equiv b
A2: S is satisfying_SST_A5;
    b,a equiv a9,b9 by A1,Satz2p4; then
A3: a,b,c,a AFS a9,b9,c9,a9 by A1,Satz2p5,Satz2p8;
    per cases;
    suppose a = b;
      hence thesis by A1,Satz2p2,GTARSKI1:def 7;
    end;
    suppose a <> b;
      then c,a equiv c9,a9 by A3,A2;
      then a,c equiv c9,a9 by Satz2p4;
      hence thesis by Satz2p5;
    end;
  end;
```

Metamath

```
Theoremtgcgrextend
• |- P = ( Base ` G )
• |- .- = ( dist ` G )
• |- I = ( Itv ` G )
• |- ( ph -> G e. TarskiG )
• |- ( ph -> A e. P )
• |- ( ph -> B e. P )
• |- ( ph -> C e. P )
• |- ( ph -> D e. P )
• |- ( ph -> E e. P )
• |- ( ph -> F e. P )
• |- ( ph -> B e. ( A I C ) )
• |- ( ph -> E e. ( D I F ) )
• |- ( ph -> ( A .- B ) = ( D .- E ) )
• |- ( ph -> ( B .- C ) = ( E .- F ) )
|- ( ph -> ( A .- C ) = ( D .- F ) )
SHOW TYPES
StepHypRefExpression1E tgcgrextend.4
+|- ( ph -> ( B .- C ) = ( E .- F ) )
21adantr
+|- ( ( ph /\ A = B ) -> ( B .- C ) = ( E .- F ) )
3simpr
+|- ( ( ph /\ A = B ) -> A = B )
43oveq1d
+|- ( ( ph /\ A = B ) -> ( A .- C ) = ( B .- C ) )
5E tkgeom.p
+|- P = ( Base ` G )
6E tkgeom.d
+|- .- = ( dist ` G )
7E tkgeom.i
+|- I = ( Itv ` G )
8E tkgeom.g
+|- ( ph -> G e. TarskiG )
98adantr
+|- ( ( ph /\ A = B ) -> G e. TarskiG )
10E tgcgrextend.a
+|- ( ph -> A e. P )
1110adantr
+|- ( ( ph /\ A = B ) -> A e. P )
12E tgcgrextend.b
+|- ( ph -> B e. P )
1312adantr
+|- ( ( ph /\ A = B ) -> B e. P )
14E tgcgrextend.d
+|- ( ph -> D e. P )
1514adantr
+|- ( ( ph /\ A = B ) -> D e. P )
16E tgcgrextend.e
+|- ( ph -> E e. P )
1716adantr
+|- ( ( ph /\ A = B ) -> E e. P )
18E tgcgrextend.3
+|- ( ph -> ( A .- B ) = ( D .- E ) )
1918adantr
+|- ( ( ph /\ A = B ) -> ( A .- B ) = ( D .- E ) )
205, 6, 7, 9, 11, 13, 15, 17, 19, 3tgcgreq
+|- ( ( ph /\ A = B ) -> D = E )
2120oveq1d
+|- ( ( ph /\ A = B ) -> ( D .- F ) = ( E .- F ) )
222, 4, 213eqtr4d
+|- ( ( ph /\ A = B ) -> ( A .- C ) = ( D .- F ) )
238adantr
+|- ( ( ph /\ A =/= B ) -> G e. TarskiG )
24E tgcgrextend.c
+|- ( ph -> C e. P )
2524adantr
+|- ( ( ph /\ A =/= B ) -> C e. P )
2610adantr
+|- ( ( ph /\ A =/= B ) -> A e. P )
27E tgcgrextend.f
+|- ( ph -> F e. P )
2827adantr
+|- ( ( ph /\ A =/= B ) -> F e. P )
2914adantr
+|- ( ( ph /\ A =/= B ) -> D e. P )
3012adantr
+|- ( ( ph /\ A =/= B ) -> B e. P )
3116adantr
+|- ( ( ph /\ A =/= B ) -> E e. P )
32simpr
+|- ( ( ph /\ A =/= B ) -> A =/= B )
33E tgcgrextend.1
+|- ( ph -> B e. ( A I C ) )
3433adantr
+|- ( ( ph /\ A =/= B ) -> B e. ( A I C ) )
35E tgcgrextend.2
+|- ( ph -> E e. ( D I F ) )
3635adantr
+|- ( ( ph /\ A =/= B ) -> E e. ( D I F ) )
3718adantr
+|- ( ( ph /\ A =/= B ) -> ( A .- B ) = ( D .- E ) )
381adantr
+|- ( ( ph /\ A =/= B ) -> ( B .- C ) = ( E .- F ) )
395, 6, 7, 23, 26, 29tgcgrtriv
+|- ( ( ph /\ A =/= B ) -> ( A .- A ) = ( D .- D ) )
405, 6, 7, 23, 26, 30, 29, 31, 37tgcgrcomlr
+|- ( ( ph /\ A =/= B ) -> ( B .- A ) = ( E .- D ) )
415, 6, 7, 23, 26, 30, 25, 29, 31, 28, 26, 29, 32, 34, 36, 37, 38, 39, 40axtg5seg
+|- ( ( ph /\ A =/= B ) -> ( C .- A ) = ( F .- D ) )
425, 6, 7, 23, 25, 26, 28, 29, 41tgcgrcomlr
+|- ( ( ph /\ A =/= B ) -> ( A .- C ) = ( D .- F ) )
4322, 42pm2.61dane
+|- ( ph -> ( A .- C ) = ( D .- F ) )
```

Otter

Tarski Formalization Project (Otter)

http://www.michaelbeeson.com/research/FormalTarski/index.php
http:
//www.michaelbeeson.com/research/FormalTarski/Proofs/Satz2.11.prf

```
Length of proof is 14.  Level of proof is 4.

---------------- PROOF ----------------

1 [] E(x,y,y,x).
2 [] -E(x,y,z,v)| -E(x,y,z2,v2)|E(z,v,z2,v2).
6 [] -E(x,y,x1,y1)| -E(y,z,y1,z1)| -E(x,v,x1,v1)| -E(y,v,y1,v1)| -T(x,y,z)| -T(x1,y1,z1)|x=y|E(z,v,z1,v1).
8 [] -E(xa,xb,xc,xd)|E(xc,xd,xa,xb).
10 [] -E(xa,xb,xc,xd)|E(xb,xa,xc,xd).
11 [] -E(xa,xb,xc,xd)|E(xa,xb,xd,xc).
12 [] E(xa,xa,xb,xb).
13 [] T(a,b,c).
14 [] T(a1,b1,c1).
15 [] E(a,b,a1,b1).
16 [] E(b,c,b1,c1).
17 [] -E(a,c,a1,c1).
28 [binary,15.1,11.1] E(a,b,b1,a1).
32 [binary,15.1,8.1] E(a1,b1,a,b).
44 [binary,16.1,8.1] E(b1,c1,b,c).
52 [binary,17.1,11.2] -E(a,c,c1,a1).
53 [binary,17.1,10.2] -E(c,a,a1,c1).
55 [binary,17.1,8.2] -E(a1,c1,a,c).
77 [binary,28.1,10.1] E(b,a,b1,a1).
86 [hyper,2,28,1] E(b1,a1,b,a).
192 [binary,52.1,10.2] -E(c,a,c1,a1).
206 [ur,2,1,53] -E(c1,a1,c,a).
256 [hyper,6,15,16,12,77,13,14,unit_del,192] a=b.
276 [para_from,256.1.2,44.1.3] E(b1,c1,a,c).
343 [hyper,6,32,44,12,86,14,13,unit_del,206] a1=b1.
356 [para_from,343.1.1,55.1.1] -E(b1,c1,a,c).
357 [binary,356.1,276.1] $F.
```