# Proof by Cases in Euclidean Geometry

Sana Stojanović Đurđević and Danijela Simić
sana.stojanovic.djurdjevic@matf.bg.ac.rs, danijela.simic@matf.bg.ac.rs
Faculty of Mathematics, University of Belgrade

International Conference on Mathematical and Computational Linguistics for Proofs
EuroProofNet Symposium
September 17, 2025

# Basic idea

- Use textbook proofs to generate verifiable readable proofs
- Textbook proofs - want to keep them and use them!
  - Readable
  - Informal
  - Error prone
  - Crucial in education
- Machine verifiable proofs - want to generate them!
  - Rarely readable
  - Formal
  - Error free
  - Crucial for science
- Formulate a language for description of textbook-like proofs precise enough to keep the essence of the original proof (and generate verifiable proof trace)
- Target audience: high-school and university students

# Example - Simple theorem and its proof

Theorem *If a point $C$ does not belong to a line $p$, then there exists a plane such that the point $C$ and the line $p$ lie on that plane.*

D1 The line $p$ contains at least two points $A$ and $B$

D2 Points $A$, $B$ and $C$ are non-collinear

D3 There exists a plane $\alpha$ that contains non-collinear points $A$, $B$ i $C$

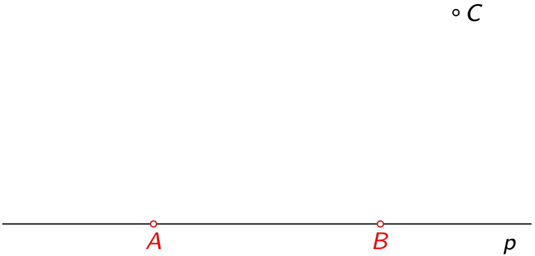
D4 Line $p$ lies on the plane $\alpha$

# Theorem


Theorem *If a point C does not belong to a line p, then there exists a plane such that the point C and the line p lie on that plane.*

- D1 The line $p$ contains at least two points $A$ and $B$
- D2 Points $A$, $B$ and $C$ are non-collinear
- D3 There exists a plane $\alpha$ that contains non-collinear points $A$, $B$ i $C$
- D4 Line $p$ lies on the plane $\alpha$

# First step of the proof




Theorem *If a point $C$ does not belong to a line $p$, then there exists a plane such that the point $C$ and the line $p$ lie on that plane.*

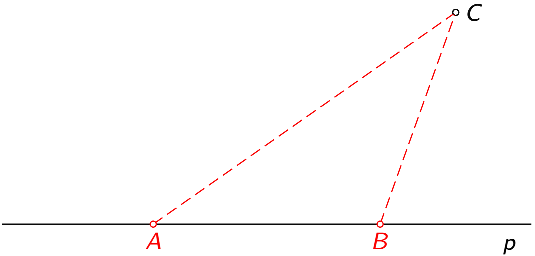
D1 The line $p$ contains at least two points $A$ and $B$

D2 Points $A$, $B$ and $C$ are non-collinear

D3 There exists a plane $\alpha$ that contains non-collinear points $A$, $B$ i $C$

D4 Line $p$ lies on the plane $\alpha$

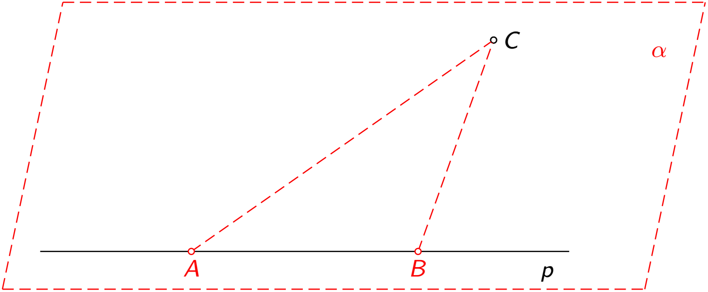
# Second step of the proof




**Theorem** *If a point $C$ does not belong to a line $p$, then there exists a plane such that the point $C$ and the line $p$ lie on that plane.*

- D1 The line $p$ contains at least two points $A$ and $B$
- D2 Points $A$, $B$ and $C$ are non-collinear
- D3 There exists a plane $\alpha$ that contains non-collinear points $A$, $B$ i $C$
- D4 Line $p$ lies on the plane $\alpha$

# Third step of the proof



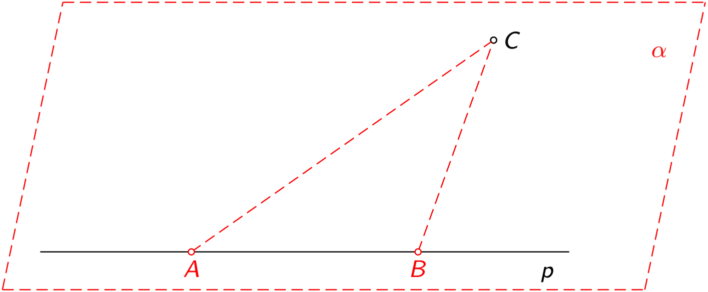

Theorem *If a point $C$ does not belong to a line $p$, then there exists a plane such that the point $C$ and the line $p$ lie on that plane.*

D1 The line $p$ contains at least two points $A$ and $B$

D2 Points $A$, $B$ and $C$ are non-collinear

D3 There exists a plane $\alpha$ that contains non-collinear points $A$, $B$ i $C$

D4 Line $p$ lies on the plane $\alpha$




Theorem *If a point $C$ does not belong to a line $p$, then there exists a plane such that the point $C$ and the line $p$ lie on that plane.*

- D1 The line $p$ contains at least two points $A$ and $B$
- D2 Points $A$, $B$ and $C$ are non-collinear
- D3 There exists a plane $\alpha$ that contains non-collinear points $A$, $B$ i $C$
- D4 Line $p$ lies on the plane $\alpha$

# What is a proof?

- A proof is a logical argument
- Under the following assumptions
  - the axiomatic system
  - the inference rules of a deductive system
  - the statement is expressed in the language of that system
- *[Marc Bezem and Dimitri Hendriks]: A proof explains why the theorem is true, and a formal proof does so in great detail.*
- Type of the proof defines two main contexts of theorem proving:
  - Textbook proofs
    the emphasis is on the idea of the proof, often short proofs
  - Machine verifiable proofs
    the emphasis is on correctness of the proof, often very detailed proofs

# Influence of education to theorem proving

- Different type of theorem proving
  - ATP - Answer to the question: Is the given conjecture a theorem?
  - ITP - A detailed answer to the question: Is the given proof of a conjecture valid?
  - A combination of the previous two approaches
- Not so long ago, formal proofs were present just in journal and conference papers
- Today interactive theorem provers and formalizations of relevant mathematical knowledge is present in undergraduate programs
- But still informal proofs are necessary as the first step in understanding and teaching of mathematics

# Motivation

- Interactive theorem proving - challenges
    - Formalization is very challenging process, far from trivial
    - Learning curve is very steep
    - Structure of the formal proof often deviates significantly from the original proof
- We want to keep the structure of the informal proof!
    - Textbook proofs have been relied on for several hundred years, they are essential for everyday mathematical practice, and in most cases should not be changed
    - We want to create the system that can verify the informal proof

# Our approach

- Statement
  [Start]
- Informal textbook proof
  [Create]
- Semi-formal proof (CL inspired language)
  [Formulate]
- Coherent logic proof objects (CLV - coherent logic vernacular)
  [Automatically generate]
- Formal proofs of a theorem (Isabelle, Coq/Rocq, natural language)
  [Automatically generate]

# Coherent logic

- CL formula ($A_i$ literals, $B_j$ conjunctions of literals, $n > 0$, $m > 0$)

$$A_1 \wedge \ldots \wedge A_n \Rightarrow \exists \vec{x_1} B_1 \vee \ldots \vee \ \exists \vec{x_m} B_m$$

- Implicitly universally quantified formula. Existential quantification is allowed just in the conclusion of the formula
- First used by Skolem, and later popularized by Bezem et al.
- Classical provability is the same as intuitionistic provability
- No function symbols (of arity greater than 0) and no negation
- Additional predicate symbols negation, sorts, and functions:
  - $\forall \vec{x}(R(\vec{x}) \wedge \overline{R}(\vec{x}) \Rightarrow \bot), \forall \vec{x}(R(\vec{x}) \vee \overline{R}(\vec{x}))$
  - $point(A) \wedge point(B) \wedge point(C) \wedge col(A, B, C)$
- Scolem, Bezem, Narboux (Tarski), Avigad (Elements)
- Features: Easily describe and generate proofs in
  - languages of different interactive theorem provers (Isabelle, Coq/Rocq)
  - readable, natural language proofs

## Coherent logic vernacular - CLV

- *Vernacular* - a language or a dialect native to a region or a country (Marriam-Webster definition)
- *Mathematical vernacular*
  - *A formalism proposed for trying to put a substantial part of mathematical vernacular into the formal system (de Bruijn, 1980's)*
  - *There is a canonical style of presenting mathematics that people discover independently: something like a natural mathematical vernacular (Freek Wiedijk 2000)*
- *Coherent logic vernacular*
  - Not a mathematical vernacular
  - Light-weight proof representation
  - Suitable for automatization and translation to various languages

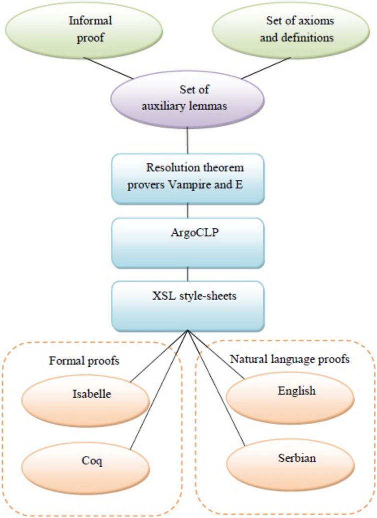
# Coherent logic prover and CLV

- ArgoCLP - CL theorem prover (Stojanović, Marinković, Janičić)
  - Input - axiomatic system and a conjecture in TPTP format
  - Output - proof trace in CLV format
- CLV - A dialect for CL (Stojanović, Narboux, Bezem, Janičić)
  - The proof steps use only the following rules:
    *modus ponens, case splits, assumptions, ex falso quodlibet*
  - Simple and expressive
  - XML-based format for proof representation in CL (with additional verification of proof structure)

# Coherent logic framework for automated formalization - ArgoGeoChecker

- Input
  - An informal (semi-formal) proof of a theorem
  - Set of axioms and definitions expressed in CL
- Output CLV proof trace for validation and translation to:
  - Isabelle, Coq/Rocq (formal proofs)
  - English, Serbian (readable proofs - LaTeX)
- Automated theorem provers
  - Resolution theorem provers (*Vampire*, *E*) (for minimizing set of axioms)
  - Coherent logic theorem prover (*ArgoCLP*) (used for CLV generation)
- The steps of the semi-formal proof will be verified individually
- A formal document is generated always, even when some steps were not proven (labeled with *sorry* in Isabelle, and with *Admitted* in Coq/Rocq)

# The architecture of the system



# Formal rendering of an informal proof (1)

- Preserve the explanatory value of the proof
  - Do not use a general translation procedure to CL (use the translations specific to Euclidean geometry)
  - Extract only the relevant information from the proof (change of context or a new relevant step is introduced)
    - the existence of the intersection point of two lines
    - the conclusion that certain points are collinear
    - case split assumptions
- Types of proofs
  - Direct linear proofs - a finite sequence of formal statements (each proof step is either a left hand side or a right hand side of a CL formula)
  - Proofs by cases
  - Proofs by contradiction

# Formal rendering of an informal proof (2)

The first step is assuming the premiss of the conjecture to be proved, i.e. a conjunction of first-order atoms (left hand side of a CL formula)

Subsequent step(s) is a possibly existentially quantified conjunction of first-order atoms

- right hand side of a CL formula if $m = 1$
- or an assumption of the current case (if $m > 1$)

The last step is the goal of the informal proof, i.e. a disjunction of a possibly existentially quantified conjunctions (right hand side of a CL formula for $m \geq 1$)

The conclusion of the conjecture will not always be the last step of the semi-formal proof (existentially quantified objects can be constructed earlier in the proof, and the rest of the proof will show that those objects satisfy the required properties)

# Formal rendering of an informal proof (3)

- *assume* - the first step always starts with the word assume (gives the context of the theorem being proved)
- *let* - introduces new objects
- *have* - introduces new relations over the existing objects
- *suppose* - introducing a local assumption (case split on atomic formule)
  - Can be used for case distinctions as well as for proof by contradiction
  - Proof by contradiction always establishes an atomic formulae or its negation
- *contradiction* - releases a local assumption

# Formal rendering of an informal proof (4)

- Syntax is simple and self explanatory (easy integration into the TPTP conforming formulas)
- Unique objects

$$\exists! x P(x) \equiv \exists x P(x) \wedge (\forall x \forall y P(x) \wedge P(y) \rightarrow x = y)$$

- Explicit nondegeneracy assumptions and giving name to an existing objects

Original *There exists a point D that does not belong to the line AB*

Transformed *For different points A and B, let L be the line determined by the points A and B, led D be a point that does not belong to the line L*

Semi-formal
```
have (A!=B)
let [L]:(line(L) & inc_po_l(A,L) & inc_po_l(B,L))
let [D]:(point(D) & ninc_po_l(D,L))
```

# Textbook proof and semi-formal proof

**Teorema (textbook)**

*If a point $C$ does not belong to a line $p$, then there exists a plane such that the point $C$ and the line $p$ lie on that plane.*

Theorem *If a point $C$ does not belong to a line $p$, then there exists a plane such that the point $C$ and the line $p$ lie on that plane.*

- D1 The line $p$ contains at least two points $A$ and $B$
- D2 Points $A$, $B$ and $C$ are non-collinear
- D3 There exists a plane $\alpha$ that contains non-collinear points $A$, $B$ i $C$
- D4 Line $p$ lies on the plane $\alpha$

**Teorema (semi-formal)**

*If a point $C$ does not belong to a line $p$, then there exists a plane such that the point $C$ and the line $p$ lie on that plane.*

```
0. assume [P,C] : (line(P) & point(C) & ninc_po_l(C,P))
1. let [A,B] : (point(A) & point(B) & A!=B & inc_po_l(A,P) & inc_po_l(B,P))
2. have (ncol(A,B,C))
3. let [R] : (plane(R) & inc_po_pl(A,R) & inc_po_pl(B,R) & inc_po_pl(C,R))
4. have (inc_l_pl(P,R) & inc_po_pl(C,R))
```

# Semi-formal proof, uniqueness

**Teorema (uniqueness)**

*If a point $C$ does not belong to a line $p$, and there exist two planes $\alpha$ and $\beta$ such that the point $C$ and line $p$ lie on both of them, then those two planes are equal.*

```
0. assume [P,C,R1,R2] : (line(P) & point(C) & plane(R1) & plane(R2) &
                        ninc_po_l(C,P) & inc_po_pl(C,R1) & inc_l_pl(P,R1) &
                        inc_po_pl(C,R2) & inc_l_pl(P,R2))
1. let [A,B] : (point(A) & point(B) & A!=B & inc_po_l(A,P) & inc_po_l(B,P))
2. have (ncol(A,B,C))
3. have (inc_po_pl(A,R1) & inc_po_pl(B,R1) & inc_po_pl(C,R1) & inc_po_pl(A,R2) &
        inc_po_pl(B,R2) & inc_po_pl(C,R2))
4. have (R1 = R2)
```

# Semi-formal proof with case distinctions

**Teorema**

*If the following relations hold:* $bet(A, B, C)$, $bet(A, D, C)$ *then* $\neg bet(B, A, D)$.

```
0. assume [A,B,C,D] : (point(A) & point(B) & point(C) & bet(A,B,C) & bet(A,D,C))
1. suppose (bet(B,A,D))
  2. have (bet(D,A,B))
  3. have (bet(D,A,C))
  4. have (nbet(A,D,C))
  5. contradiction
6. suppose (nbet(B,A,D))
  7. have (nbet(B,A,D))
```

## The axiomatic system $E$ (Avigad, Dean, Mumma)

- Euclid's *Elements* were criticized for using informations from the diagram — *intuition* over the *rules*
- But, Euclidean practice is stable and in some way still used!
- Jeremy Avigad et al. noted: *Over the centuries, the style of diagram-based argumentation of Euclid's Elements was held to be the paradigm of rigor, and presentation much like Euclid's are still used today to introduce students to the notions of proof*
- Formal axiomatic system $E$ faitful to Euclid
  - Faithful representation of the proofs
  - Diagrammatic reasoning in Euclid's proofs is controlled and guided by a distinct logic
  - Diagram is the representation of the relevant data (such as incidence, intersection,...; but not for congruence - explicitly stated)

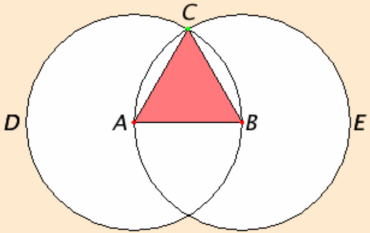
# Proofs in $E$

- Form of the theorems:
  *Given points, lines, circles, satisfying...,*
  *there are points, lines, circles satisfying...*
- Implicit assumption: objects are assumed to be distinct, triangles are assumed to be nondegenerate - this has to be stated explicitly
- *Demonstration* rules and *construction* rules
- Reasoning is linear, except: proof by contradiction, using a case distinction

# First Euclid's postulate in Elements

*To construct an equilateral triangle on a given finite straight line.*

Let $AB$ be the given finite straight line.




It is required to construct an equilateral triangle on the straight line $AB$.

Describe the circle $BCD$ with center $A$ and radius $AB$. Again describe the circle $ACE$ with center $B$ and radius $BA$. Join the straight lines $CA$ and $CB$ from the point $C$ at which the circles cut one another to the points $A$ and $B$. I.Post.3 I.Post.1

Now, since the point $A$ is the center of the circle $CDB$, therefore $AC$ equals $AB$. Again, since the point $B$ is the center of the circle $CAE$, therefore $BC$ equals $BA$. I.Def.15

But $AC$ was proved equal to $AB$, therefore each of the straight lines $AC$ and $BC$ equals $AB$.

And things which equal the same thing also equal one another, therefore $AC$ also equals $BC$. C.N.1

Therefore the three straight lines $AC$, $AB$, and $BC$ equal one another.

Therefore the triangle $ABC$ is equilateral, and it has been constructed on the given finite straight line $AB$. I.Def.20

Q.E.F.

## First Euclid's postulate in axiomatic system $E$

- *Proposition I.1.*
  *On a given straight line, to construct an equilateral triangle.*
- Proposition I.1. *Assume $a$ and $b$ are two distinct points. Construct point $c$ such that $\overline{ab} = \overline{bc}$ and $\overline{bc} = \overline{ca}$.*
  Proof.
  Let $\alpha$ be the circle with center $a$ passing through $b$.
  Let $\beta$ be the circle with center $b$ passing through $a$.
  Let $c$ be a point on the intersection of $\alpha$ and $\beta$.
  Have $\overline{ab} = \overline{ac}$ [since they are radii of $\alpha$].
  Have $\overline{ba} = \overline{bc}$ [since they are radii of $\beta$].
  Hence $\overline{ab} = \overline{bc}$ and $\overline{bc} = \overline{ca}$.
  Q.E.F.
- *For the sake of intelligibility, we sometimes add comments, in brackets. Once again, these play no role in the formal proof (Avigad et. al)*

# First Euclid's postulate - semi-formal proof

- **Semi-formal proof:**

```
0. assume [A,B] : (point(A) & point(B) & A != B)
1. let [K1] : (circle(K1) & center(A,K1) & onc(B,K1))
2. let [K2] : (circle(K2) & center(B,K2) & onc(A,K2))
3. let [C]   : (point(C) & onc(C,K1) & onc(C,K2) & intersectscc(K1,K2)
4. have (cong(A,B,A,C))
5. have (cong(B,A,B,C))
6. have (cong(A,B,B,C) & cong(B,C,C,A))
```

- We do not use comments!

# Autoformalizing Euclidean Geometry

- *LeanEuclid* — Framework for autoformalizing Euclidean geometry (Murphy, Yang, Sun, Li, Anandkumar, Si)
- Automated reasoning engine based on SMT solvers, large language models (LLMs)
- Fills the gaps in the proofs - gaps need to be small enough
- Implemented E in Lean
- Manually formalized proofs into Lean

# Axiomatic system E in Lean

- Axioms of *E* are built into the system
- They provide several customized tactics that cover most steps used by Euclid (but a proof can be made of arbitrary tactics)
  - `euclid_intros` — creates context for the current theorem
  - `euclid_apply` — applies a rule in a forward direction
  - `euclid_assert` — adds a new fact
  - `euclid_finish` — verifies that the conclusion is derived
- Check if the specific rule can be applied in the given context (of known facts) or (if not) tries to prove left hand side of the rule

```
intersection_circles BCD ACE : intersectsCircle BCD ACE
→ ∃ c : Point, (onCircle c BCD) ∧ (onCircle c ACE)
```

# First Euclid's postulate in LeanEuclid

- theorem proposition$_1$ : $\forall$ (a b : Point) (AB : Line), distinctPointsOnLine a b AB $\rightarrow$
  $\exists c$ : *Point*, |(c - a)| = |(a - b)| $\wedge$ |(c - b)| = |(a - b)| :=
  ```
  by
    euclid_intros
    euclid_apply circle_from_points a b as BCD
    euclid_apply circle_from_points b a as ACE
    euclid_apply intersection_circles BCD ACE as c
    euclid_apply point_on_circle_onlyif a b c BCD
    euclid_apply point_on_circle_onlyif b a c ACE
    use c
    euclid_finish
  ```
- They use the comments from the E proofs!
- The implicit steps of the SMT engine can use only non-construction rules, whereas explicit steps performed by humans (or machine learning) can use any rules

# Conclusion and future work

- Simple approach, suited for the beginners in automated and interactive theorem proving
- Semi-formal proofs resemble those found in mathematical textbooks
  - Keep the conceptual framework for formulating and proving theorems
- Automatically generated proof trace is translated to
  - Readable formal proofs that can be easily be used as a part of a larger formalization project (in Isabelle and in Coq/Rocq)
  - Natural language proofs with usual predicate symbols and mathematical dialect
- Students can explore, can practice more and can see the process of proving geometric theorems as a challenge for themselves
- There are room for improvement: eliminating the trivial steps, remembering previously proven theorems, using natural language as the input, creating (online) software for interactive theorem proving

# Bloom's taxonomy

The proposed approach can integrate steps of the Bloom's taxonomy in the teaching process

- *Remember* - by using a few simple trivial examples in the beginning students can learn proof steps effectively
- *Understand* - by exploring certain steps on their own, students will comprehend their meaning
- *Apply* - by applying (successfully or not) certain steps they discover the necessary assumptions and learn to apply them in new situations
- *Analyze* - by finding errors and fixing them, students learn to analyze the proof and question future ideas
- *Evaluate* - by assesing the current solution they need to choose next steps
- *Create* - by completing the previous phases, they can eventually are able to create the solution on their own