When a GUI for dynamic geometry becomes an interactive proof language.

Julien Narboux
under the supervision of
Hugo Herbelin

LIX, École Polytechnique

Thursday 14 April 2005, Chambery, France

Outline

Introduction

A diagram can be seen as a high level description of a proof.

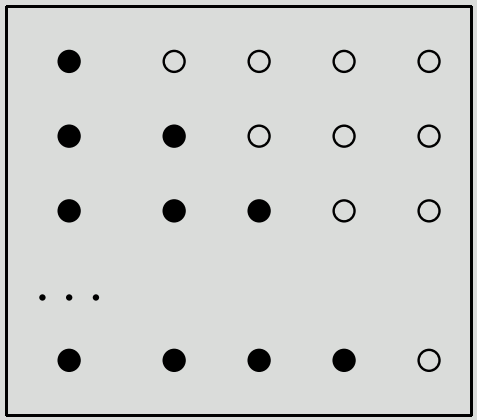
Gauss (7-year-old) :

$$1+2+3+\ldots+n=\frac{n(n+1)}{2}$$

Introduction

A diagram can be seen as a high level description of a proof.

Gauss (7-year-old) :

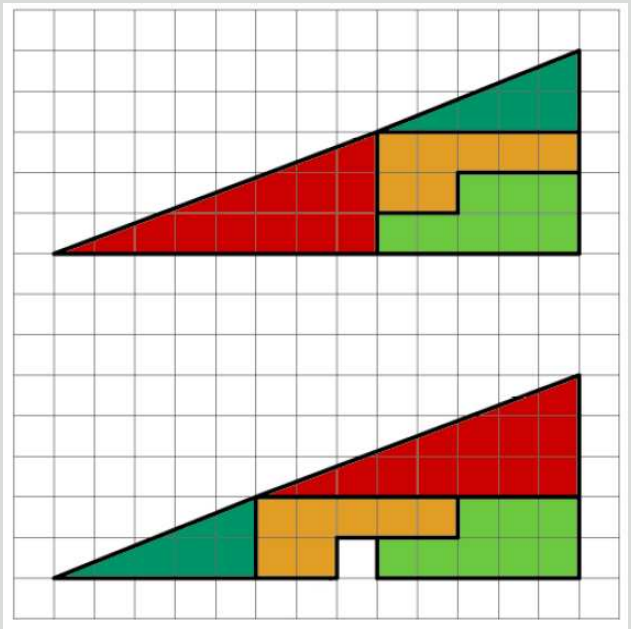


$$1 + 2 + 3 + \ldots + n = \frac{n(n+1)}{2}$$

But sometimes a diagram can be misleading.

Misleading ?

a



[a]Original author unknown, this diagram is from Daniel Winterstein's Phd.

Related work.

Diagrammatic Reasoning.

- The Diamond system (Mateja Jamnik's phd)
- Dr Doodle (Daniel Winterstein's phd)
- CDEG (Nathaniel Miller's phd)

Related work.

Diagrammatic Reasoning.

- The Diamond system (Mateja Jamnik's phd)
- Dr Doodle (Daniel Winterstein's phd)
- CDEG (Nathaniel Miller's phd)

Related work.

Diagrammatic Reasoning.

- The Diamond system (Mateja Jamnik's phd)
- Dr Doodle (Daniel Winterstein's phd)
- CDEG (Nathaniel Miller's phd)

Related work.

Interactive Geometry.

There are quite many interactive geometry software :

Related work.

Interactive Geometry.

There are quite many interactive geometry software :
Cabri Geometer, CaR, Cinderella, Déclic, Dr. Geo, Euclid, Euklid DynaGeo, Eukleides, Gava, GeoExp, GeoFlash, GeoLabo, Geometria, Geometrix, Geometry Explorer, GeoPlanW, GeoSpaceW, GEUP, GeoView, GEX, GRACE, KGeo, KIG, Non-Euclid, Sketchpad, XCas...

Related work.

Interactive Geometry.

But few can deal with proofs :

Cabri Geometer, CaR, Cinderella, Déclic, Dr. Geo, Euclid, Euklid DynaGeo, Eukleides, Gava, GeoExp, GeoFlash, GeoLabo, Geometria, Geometrix, Geometry Explorer, GeoPlanW, GeoSpaceW, GEUP, GeoView, GEX, GRACE, KGeo, KIG, Non-Euclid, Sketchpad, XCas. . .

Interactive geometry software are mainly used for educational purposes but few of them can deal with proofs.

Cinderella	Probalistic method, no proof shown.
Geometrix	Interactive proof system using a base of lemmas.
Geometry Explorer	Implementation of the full angle method using prolog, and visualization of the proofs in a diagrammatic way.
GeoView	Uses GeoPlan and Pcoq to visualize statements.
GEX/Geometer	Implementation of the area method, of Wu's method and of deductive database methods, visualization of statments only.

Cinderella	Probalistic method, no proof shown.
Geometrix	Interactive proof system using a base of lemmas.
Geometry Explorer	Implementation of the full angle method using prolog, and visualization of the proofs in a diagrammatic way.
GeoView	Uses GeoPlan and Pcoq to visualize statements.
GEX/Geometer	Implementation of the area method, of Wu's method and of deductive database methods, visualization of statments only.

Cinderella	Probalistic method, no proof shown.
Geometrix	Interactive proof system using a base of lemmas.
Geometry Explorer	Implementation of the full angle method using prolog, and visualization of the proofs in a diagrammatic way.
GeoView	Uses GeoPlan and Pcoq to visualize statements.
GEX/Geometer	Implementation of the area method, of Wu's method and of deductive database methods, visualization of statments only.

Cinderella	Probalistic method, no proof shown.
Geometrix	Interactive proof system using a base of lemmas.
Geometry Explorer	Implementation of the full angle method using prolog, and visualization of the proofs in a diagrammatic way.
GeoView	Uses GeoPlan and Pcoq to visualize statements.
GEX/Geometer	Implementation of the area method, of Wu's method and of deductive database methods, visualization of statments only.

Cinderella	Probalistic method, no proof shown.
Geometrix	Interactive proof system using a base of lemmas.
Geometry Explorer	Implementation of the full angle method using prolog, and visualization of the proofs in a diagrammatic way.
GeoView	Uses GeoPlan and Pcoq to visualize statements.
GEX/Geometer	Implementation of the area method, of Wu's method and of deductive database methods, visualization of statments only.

My goal is to combine these features:

- dynamic geometry
- automation
- proof assistant
- interactive manual/diagrammatic proof

Motivations

- The use of a proof assistant provides a way to combine geometrical proofs with larger proofs (involving induction for instance).
- There are facts than can not be visualized graphically and there are facts that are difficult to understand without a graphical approach.
- We should have both the ability to make arbitrarily complex proofs and use a base of known lemmas.
- The verification of the proofs by the proof assistant provides a high level of confidence.

Motivations

- The use of a proof assistant provides a way to combine geometrical proofs with larger proofs (involving induction for instance).
- There are facts than can not be visualized graphically and there are facts that are difficult to understand without a graphical approach.
- We should have both the ability to make arbitrarily complex proofs and use a base of known lemmas.
- The verification of the proofs by the proof assistant provides a high level of confidence.

Motivations

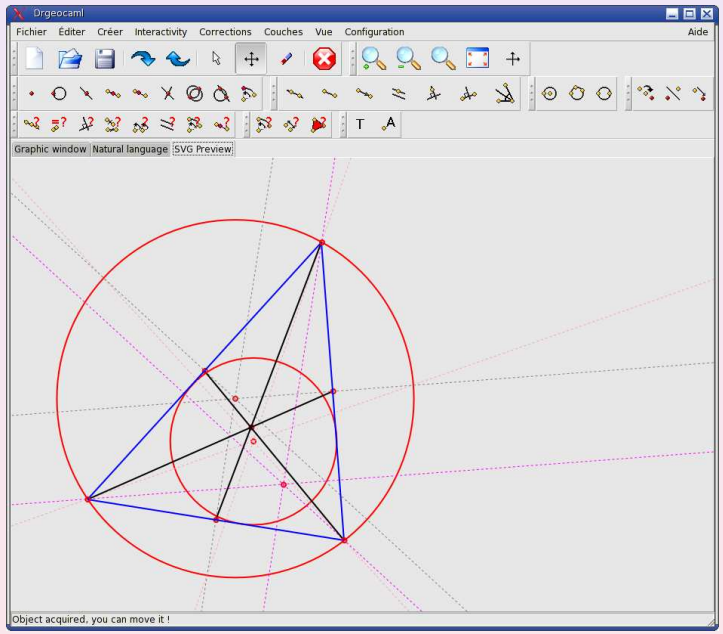
- The use of a proof assistant provides a way to combine geometrical proofs with larger proofs (involving induction for instance).
- There are facts than can not be visualized graphically and there are facts that are difficult to understand without a graphical approach.
- We should have both the ability to make arbitrarily complex proofs and use a base of known lemmas.
- The verification of the proofs by the proof assistant provides a high level of confidence.

Motivations

- The use of a proof assistant provides a way to combine geometrical proofs with larger proofs (involving induction for instance).
- There are facts than can not be visualized graphically and there are facts that are difficult to understand without a graphical approach.
- We should have both the ability to make arbitrarily complex proofs and use a base of known lemmas.
- The verification of the proofs by the proof assistant provides a high level of confidence.

A quick overview of DrGeoCaml

- Based on *Nicolas François*' work.
- Writen using `ocaml` and `lablgtk2`.
- Distributed under the GPL2 licence.



`https://gna.org/projects/geocaml`

Drgeocaml's features:

- points, lines, circles, vectors, segments, intersections, perpendicular lines, perpendicular bissectors,angle bissectors. . .
- central symetry, translation, and axial symetry
- text labels
- measures of angle, distances and areas

- I/O: natural language, .kir, .zir .csr / .svg, .png
- properties tests (collinearity,orthogonality,. . .)

Missing features:

- Common features:
  - loci and conics
  - macros
  - animations
- Interactive proof: to be discussed today !

What about colors and style ? should that be included in the proof language ?

What is a diagram ?

What is the difference between a diagram, a formula and a picture ?

- There are elements of the diagram which do not have any meaning.
- A diagrams is an example of something.
- A diagram is well defined.

What is a diagram ?

What is the difference between a diagram, a formula and a picture ?

- There are elements of the diagram which do not have any meaning.
- A diagrams is an example of something.
- A diagram is well defined.

What is a diagram ?

What is the difference between a diagram, a formula and a picture ?

- There are elements of the diagram which do not have any meaning.
- A diagrams is an example of something.
- A diagram is well defined.

Generic Sketches

My goal is :

- to define when a diagram is a proper example of something.
- to keep the direct intuitive representation of the geometrical facts.
- but I don't want to deal with the ambiguities created by the size of the pencil

Generic Sketches

My goal is :

- to define when a diagram is a proper example of something.
- to keep the direct intuitive representation of the geometrical facts.
- but I don't want to deal with the ambiguities created by the size of the pencil

Generic Sketches

My goal is :

- to define when a diagram is a proper example of something.
- to keep the direct intuitive representation of the geometrical facts.
- but I don't want to deal with the ambiguities created by the size of the pencil

Figure

A figure is a the couple formed of a set of points and a set of geometric constraints over these points.

Sketch

A sketch is a maping from the set of points of a figure to R^2 which is compatible with the constraints of that figure.

Generic Sketches (Informal Definition)

Given a figure F, we say that a set of sketches G is generic for some property P of arity n iff

$$\forall (A_1, \ldots, A_n) \in F^n_{points}$$

$$(\forall s \in G, P_f(s(A_1), \ldots, s(A_n)) = \top) \Rightarrow (\forall Pts, F_C \rightarrow P(A_1, \ldots, A_n))$$

Figure

A figure is a the couple formed of a set of points and a set of geometric constraints over these points.

Sketch

A sketch is a maping from the set of points of a figure to R^2 which is compatible with the constraints of that figure.

Generic Sketches (Informal Definition)

Given a figure F, we say that a set of sketches G is generic for some property P of arity n iff

$$\forall (A_1, \ldots, A_n) \in F^n_{points}$$

$$(\forall s \in G, P_f(s(A_1), \ldots, s(A_n)) = \top) \Rightarrow (\forall Pts, F_C \rightarrow P(A_1, \ldots, A_n))$$

Figure

A figure is a the couple formed of a set of points and a set of geometric constraints over these points.

Sketch

A sketch is a maping from the set of points of a figure to R^2 which is compatible with the constraints of that figure.

Generic Sketches (Informal Definition)

Given a figure F, we say that a set of sketches G is generic for some property P of arity n iff

$$\forall (A_1, \ldots, A_n) \in F^n_{points}$$

$$(\forall s \in G, P_f(s(A_1), \ldots, s(A_n)) = \top) \Rightarrow (\forall Pts, F_C \rightarrow P(A_1, \ldots, A_n))$$

Examples

Three free points.

collinear any non degenerated triangle is a generic sketch

collinear and equal_length non isosceles, non degenerated triangles are generic sketches

Three collinear points

Between $\longrightarrow$ Several Cases

Negative predicates

- Negative predicates are difficult to visualize.

Three points

$\neq \rightarrow$ you need to put the three points at the same place.

Gui/Language

Unification	Diagram matching
Proof of an existential theorem	Macro
Assert a new fact	Mark a fact on the diagram
Delete an assumption	Delete some object
Prove that the theorem is false	Collect counter examples

Animation or Graph ?

- The process of doing the diagram is important.
- We need a way to reflect that.

Animation or Graph ?

- The process of doing the diagram is important.
- We need a way to reflect that.

- We do need to mix visualization, automation, and interactive proof.
- Visualizing negative statments is difficult.
- Diagrams have to come with a notion which tells what we can deduce from the diagram.
- We need animation.

Thank you !

- We do need to mix visualization, automation, and interactive proof.
- Visualizing negative statments is difficult.
- Diagrams have to come with a notion which tells what we can deduce from the diagram.
- We need animation.

Thank you !