Geometry Explorer: A Tool for Generating Diagrammatic Full-Angle Method Proofs

Sean Wilson and Jacques D. Fleuriot
{sean.wilson, jacques.fleuriot}@ed.ac.uk

School of informatics

Overview

- The Full-Angle Method.
- Geometry Explorer.
- Related Work and Conclusions.

The Full-Angle Method

Overview of the Full-Angle Method

- Automatic theorem proving method for Euclidean geometry, developed by Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang.

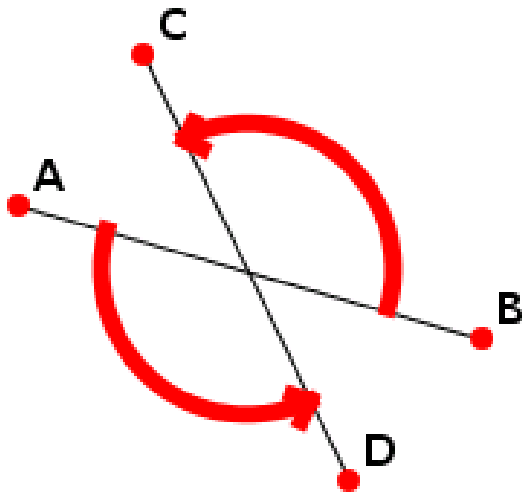
- Can generate *multiple*, *short*, *human-readable* and *diagram independent* proofs to theorems.

- Mainly relies on a high-level geometric invariant called the *full-angle* to prove theorems.

Full-Angles

- Intuitively, the full-angle between the *ordered* pair of lines u and v (written as $\angle[u, v]$) is the anti-clockwise rotation required to make u parallel to v.

- For example, measuring the full-angle $\angle[AB, CD]$:



Properties of Full-Angles

- $\angle[u, v] = -\angle[v, u]$
- $\angle[AB, CD] = \angle[BA, CD] = \angle[AB, DC] = \angle[BA, DC]$
- $\angle[AB, CD] = \angle[AB, EF] + \angle[EF, CD]$
- If $u \parallel v$ then $\angle[u, v] = \angle[0]$ is a constant (a flat full-angle).
- If $u \perp v$ then $\angle[u, v] = \angle[1]$ is a constant (a right full-angle).
- $\angle[1] + \angle[1] = \angle[0]$

Theorem Proving with Full-Angles

1. Limited but exhaustive forward-chaining is used on the hypotheses facts to discover additional useful facts.

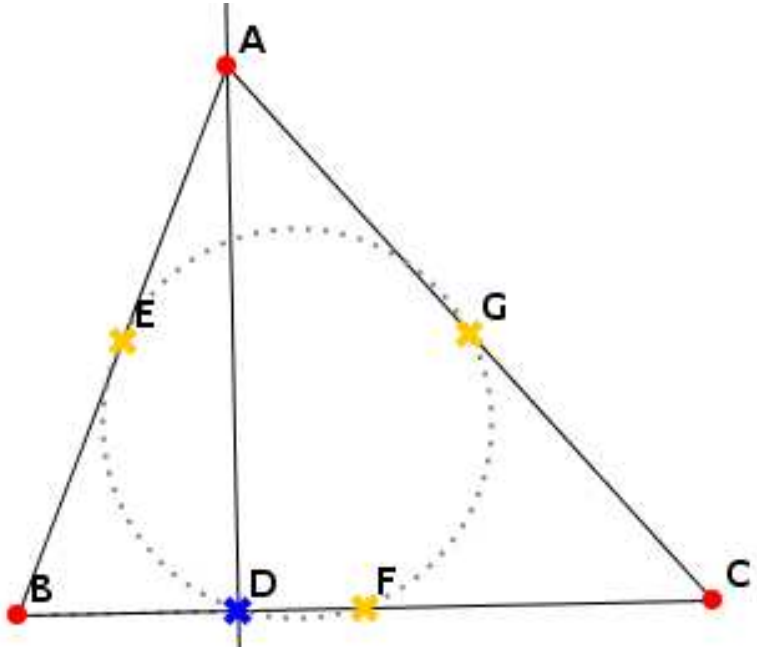
2. The conjecture is converted to the form $\angle[0] = \sum_i (f_i \cdot n_i)$, where f_i is a full-angle and n_i is its integer coefficient.

3. Conditional rewrite rules are used with the known facts to substitute full-angles in the conjecture equation with equal expressions.

4. A proof is found when the conjecture equation is transformed to $\angle[0] = \angle[0]$. This can be found using traditional tree-based search techniques.

Motivational Example

(Nine Point Circle Theorem) Let AD be the altitude on BC and let the midpoints of the sides AB, BC and CA of $\triangle ABC$ be E, F and G respectively. Show that D, E, F and G are on the same circle.



Backward-Chaining Proof

$$-\angle[GE, GD] + \angle[FE, FD]$$

$$(\angle[GE, GD] = -\angle[GD, DC] \text{ using R1 } (GE \parallel DC \text{ (see 1)}))$$

$$= \angle[GD, DC] + \angle[FE, FD]$$

$$(\angle[GD, DC] = \angle[DA, CA] + \angle[1] \text{ using R5 } (G \text{ is the circumcenter of } DCA \text{ (see 2)}))$$

$$= \angle[FE, FD] + \angle[DA, CA] + \angle[1]$$

$$(\angle[FE, FD] = -\angle[FD, CA] \text{ using R1 } (FE \parallel CA \text{ (see 3)}))$$

$$= -\angle[FD, CA] + \angle[DA, CA] + \angle[1]$$

$$(\angle[FD, CA] = \angle[DA, CA] + \angle[1] \text{ using R2 } (FD \perp DA \text{ (see 4)}))$$

$$= \angle[0]$$

Forward-Chaining Proof

New facts found:

1. $GE \parallel DC$ because of 5 and 6 (F6).
2. G is the circumcenter of DCA because G is the midpoint of CA and 7 (F5).
3. $FE \parallel CA$ because E is the midpoint of BA and F is the midpoint of CB (F3).
4. $FD \perp DA$ because $DA \perp CB$ and 8 (F2).
5. $GE \parallel CB$ because E is the midpoint of BA and G is the midpoint of CA (F3).
6. $DC \parallel CB$ because D, C and B are collinear (F1).
7. $DC \perp DA$ because $DA \perp CB$ and 6 (F2).
8. $FD \parallel CB$ because of 6 and 9 (F6).

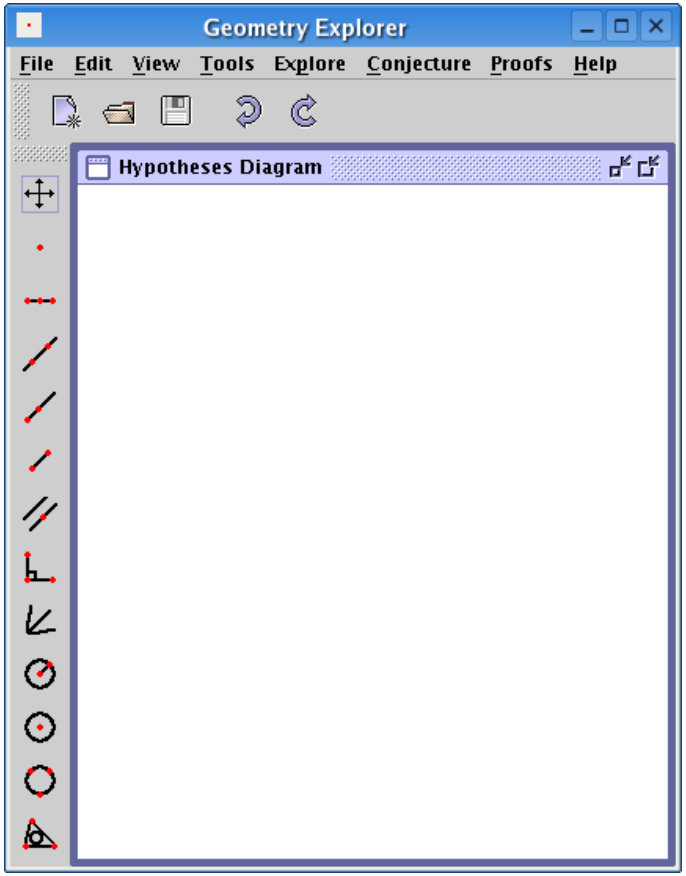
...

Geometry Explorer

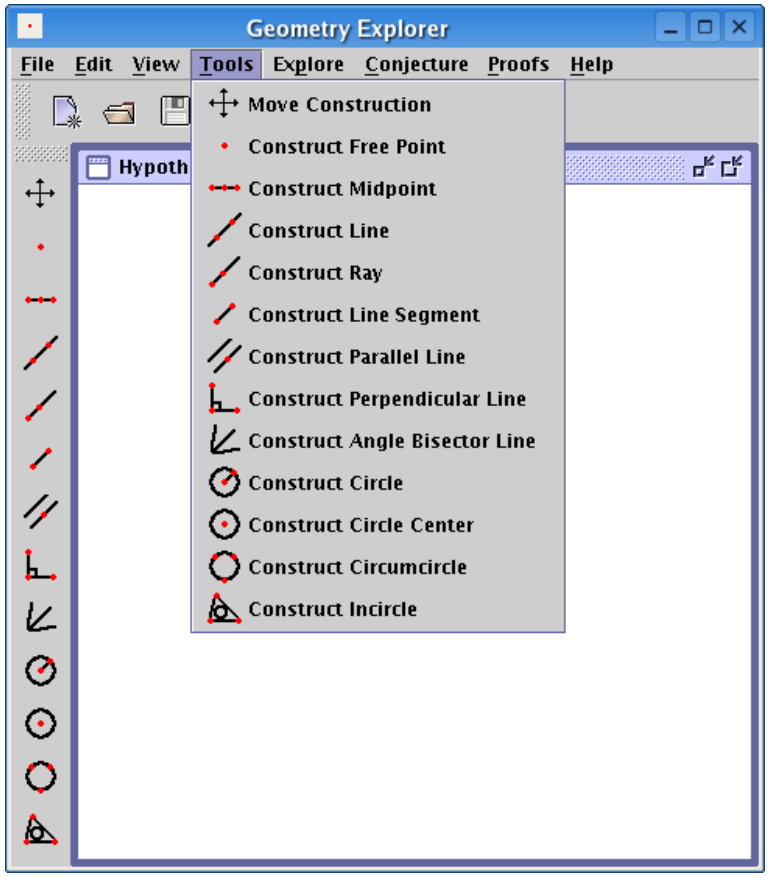
Geometry Explorer Features

- Allows geometry theorems to be specified via a *dynamic geometry* interface.
- Integrates with a full-angle method prover to generate proofs.
- Uses the theorem diagram with the full-angle method proofs to produce novel diagrammatic proof visualisations that aim to be more intuitive to interpret than the traditional textual proofs.
- Different instances of the diagrammatic proofs can be explored by manipulating the original diagram.

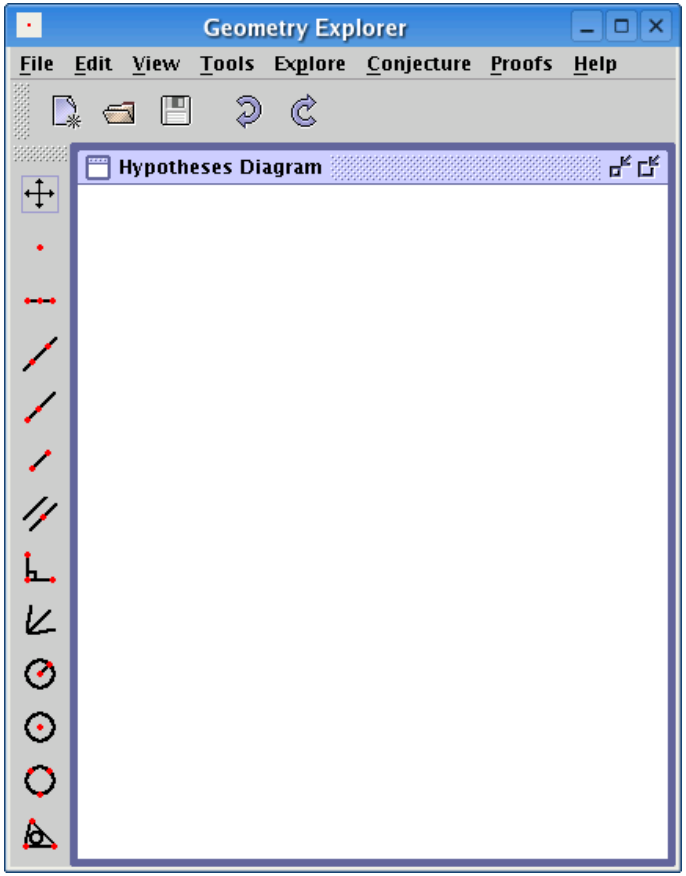
The Geometry Explorer GUI



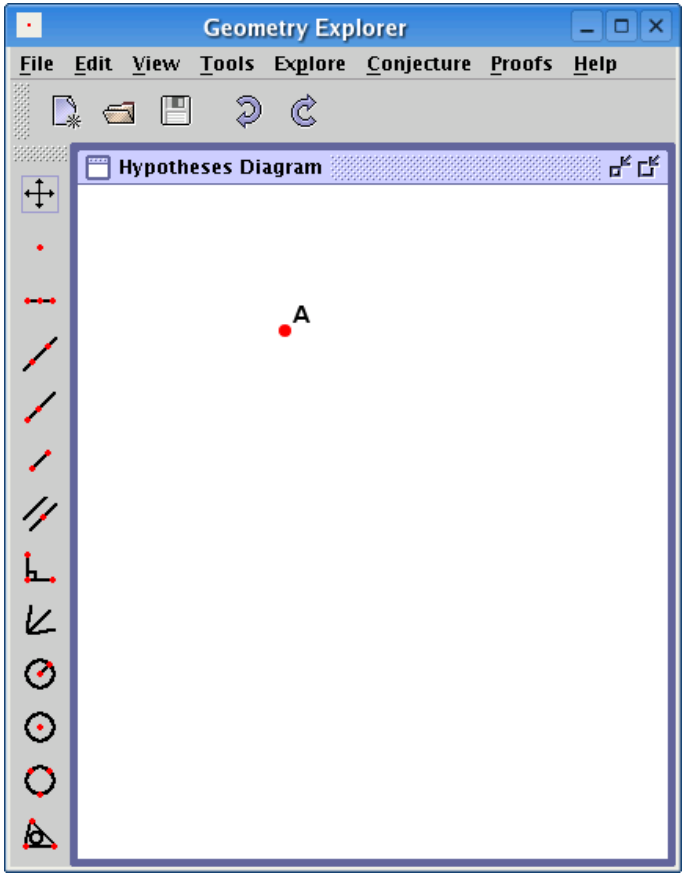
Construction Tools



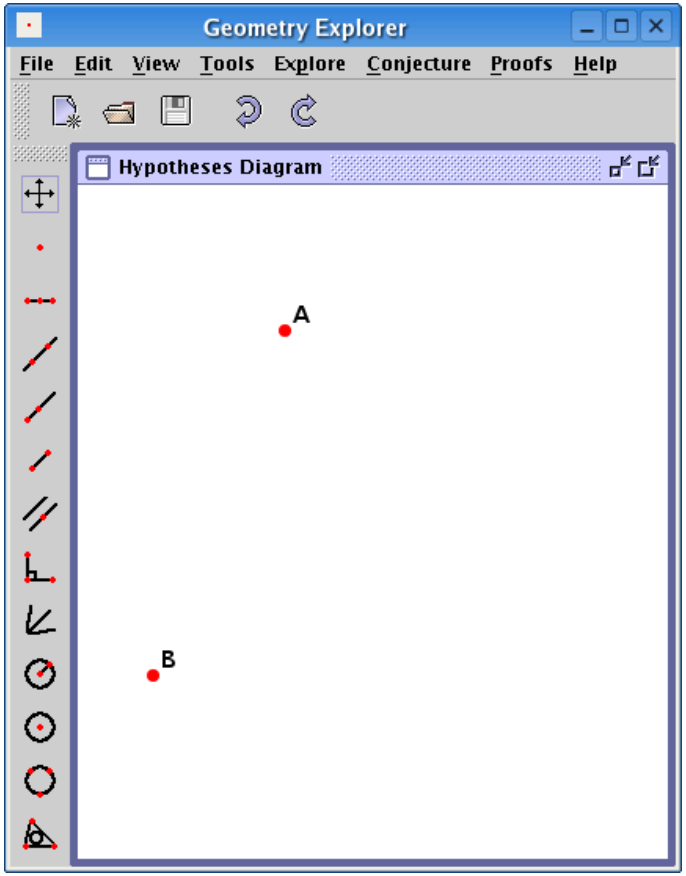
Constructing the Hypotheses Diagram



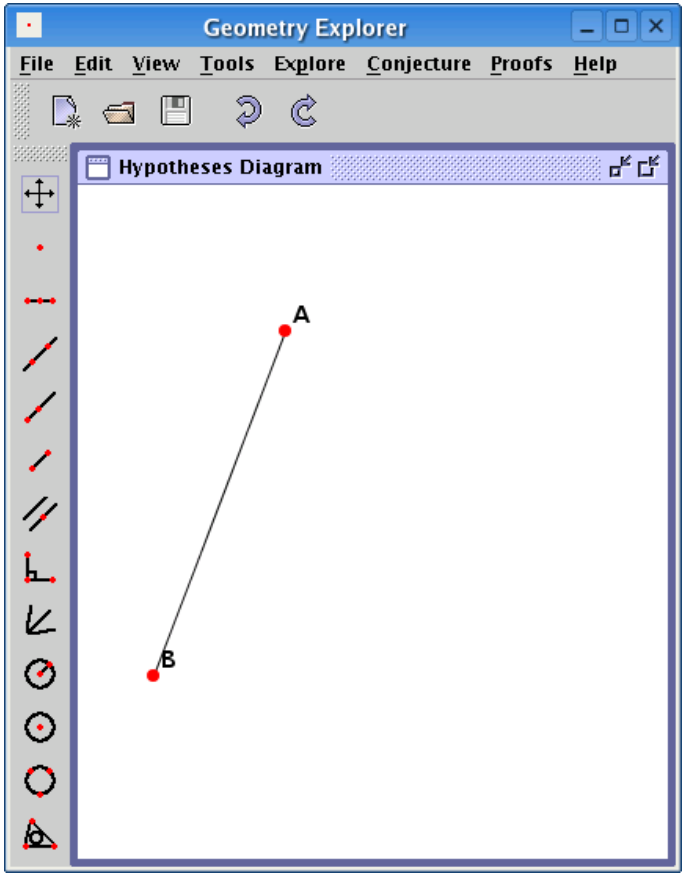
Constructing the Hypotheses Diagram



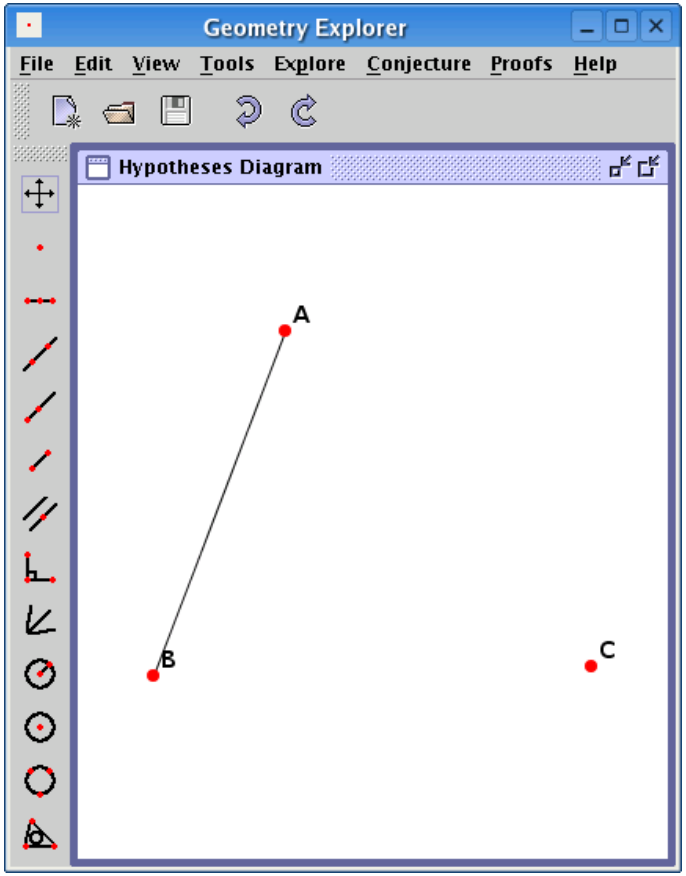
Constructing the Hypotheses Diagram



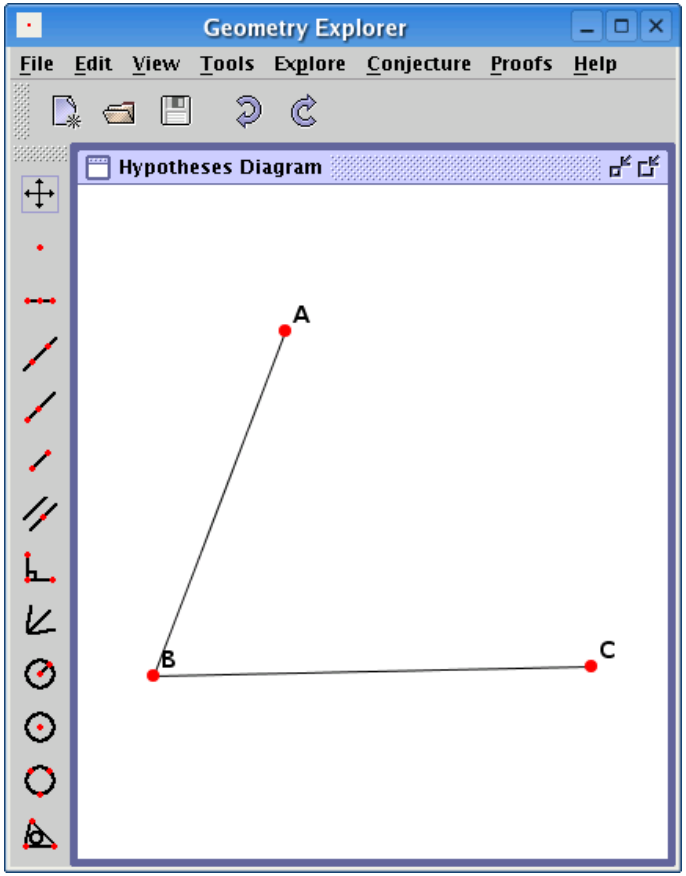
Constructing the Hypotheses Diagram



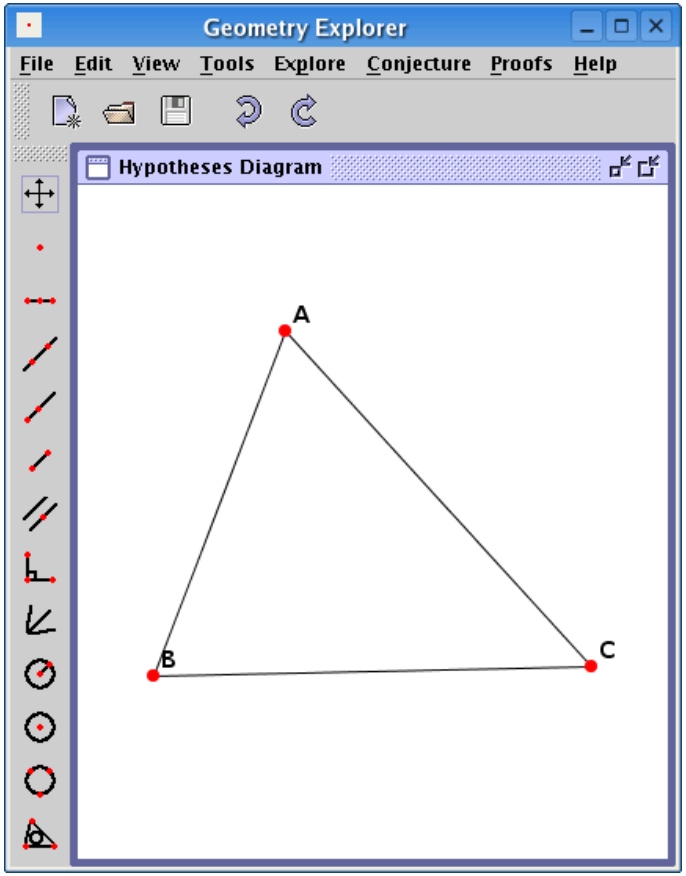
Constructing the Hypotheses Diagram



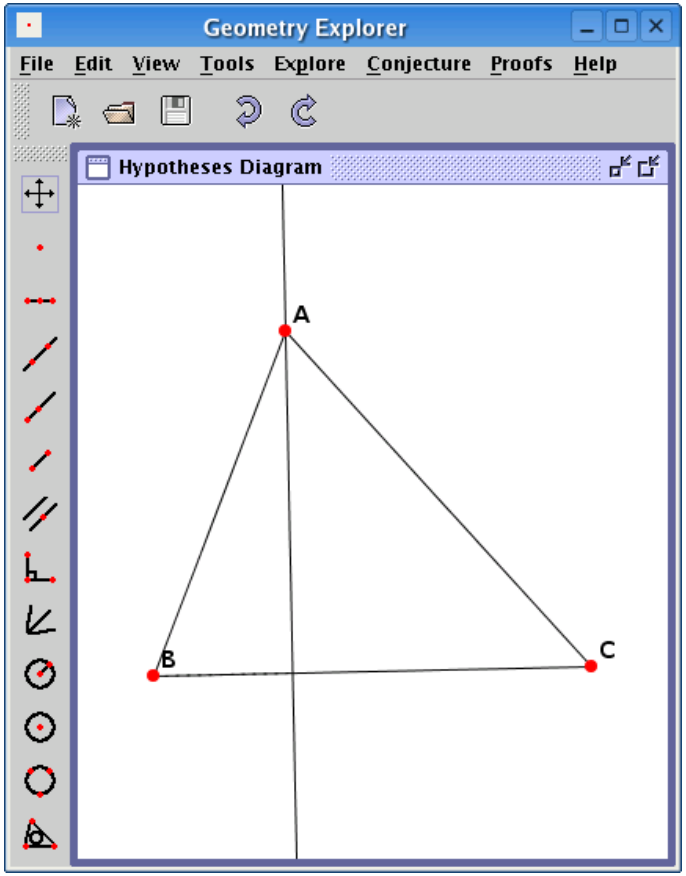
Constructing the Hypotheses Diagram



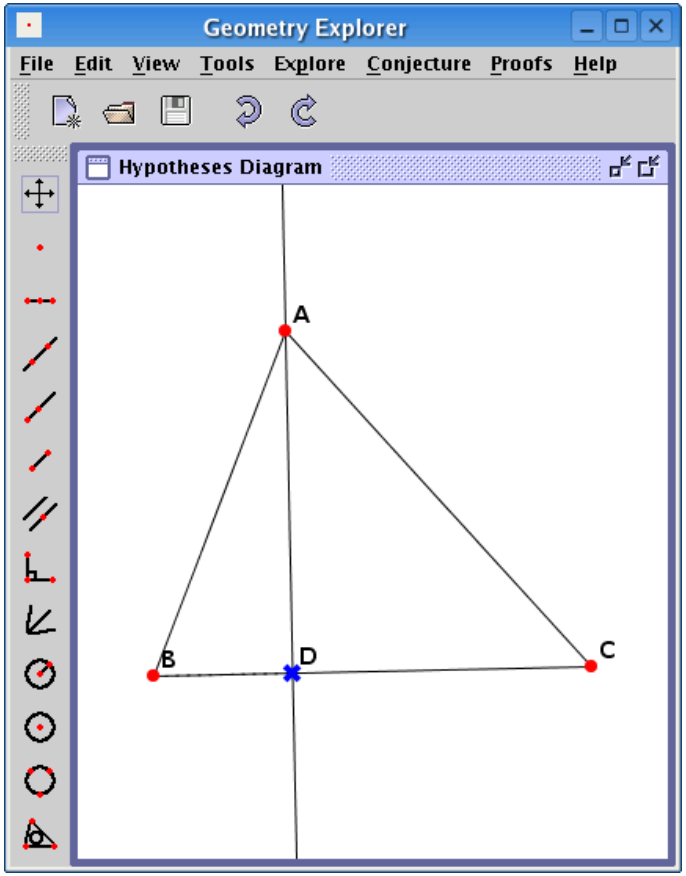
Constructing the Hypotheses Diagram



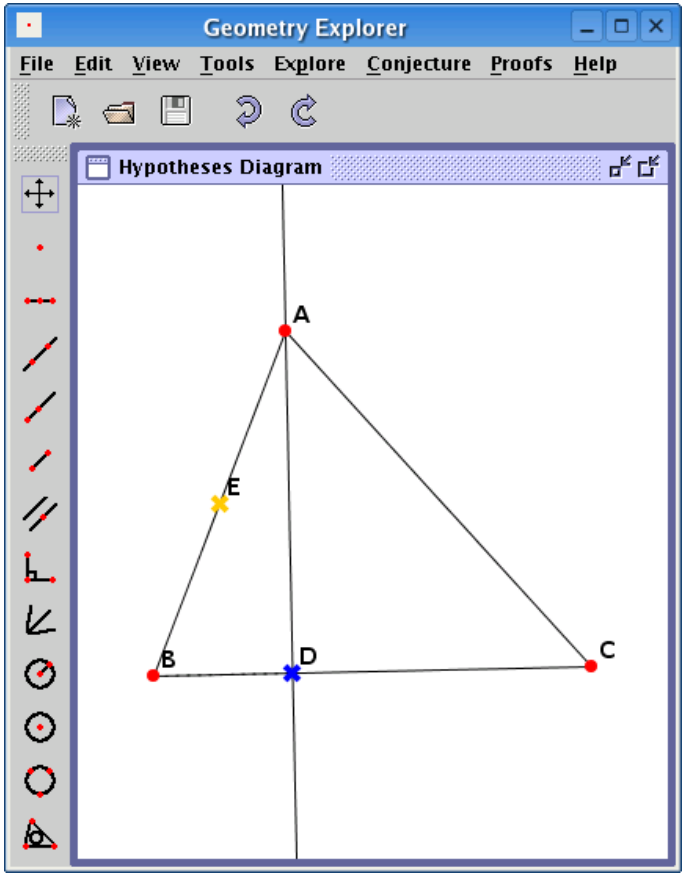
Constructing the Hypotheses Diagram



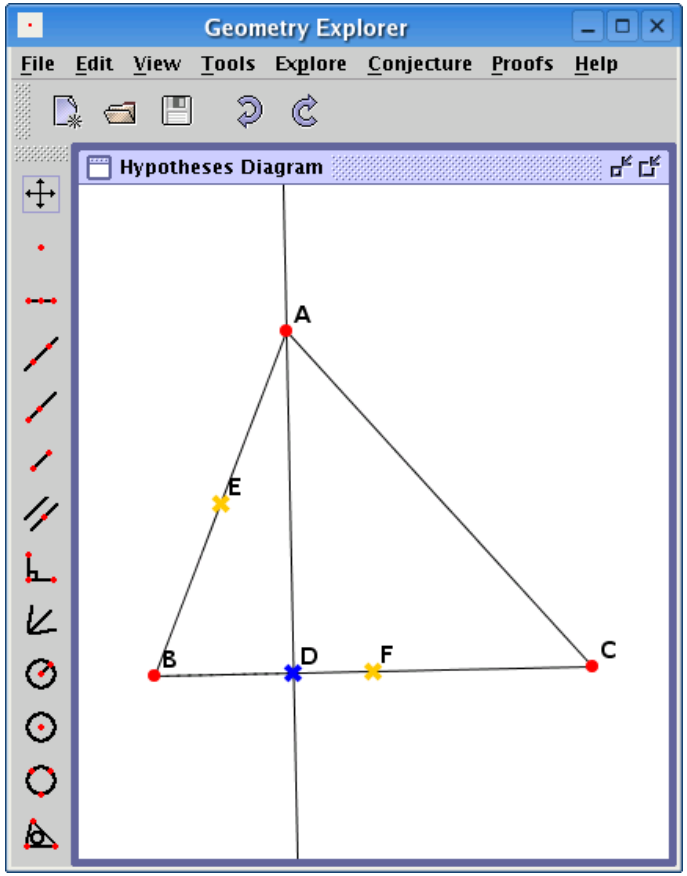
Constructing the Hypotheses Diagram



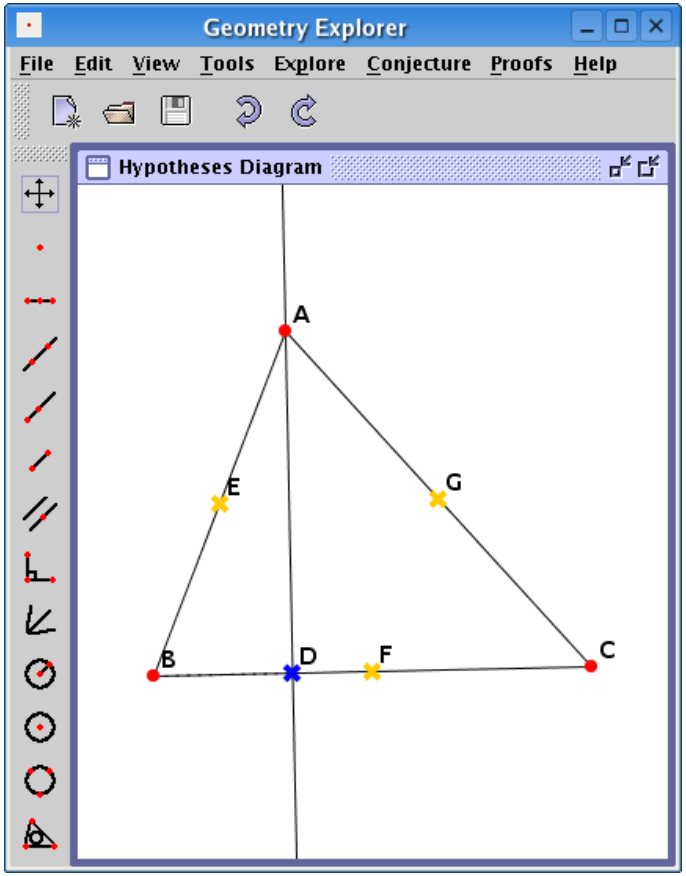
Constructing the Hypotheses Diagram



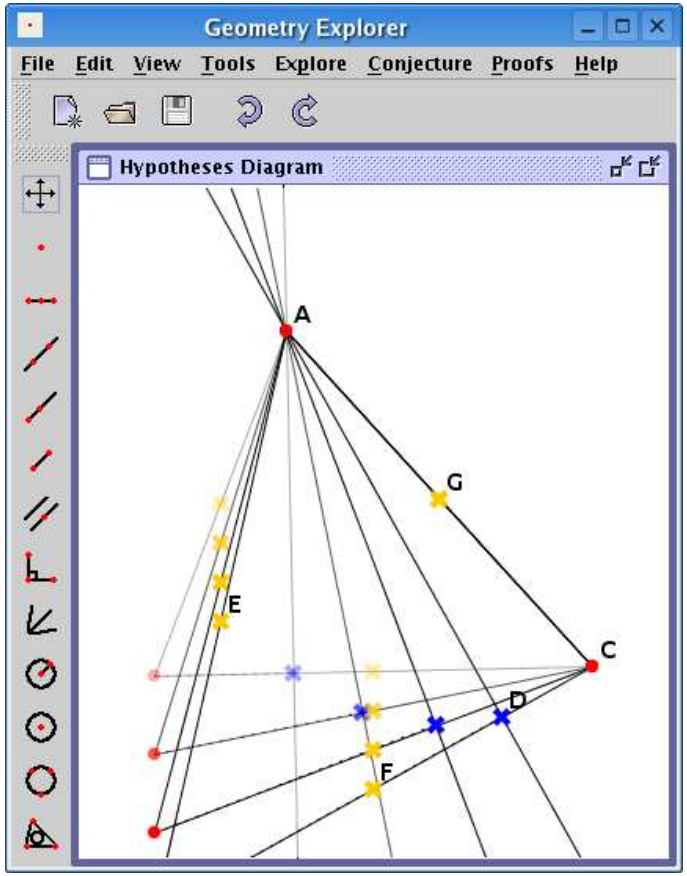
Constructing the Hypotheses Diagram



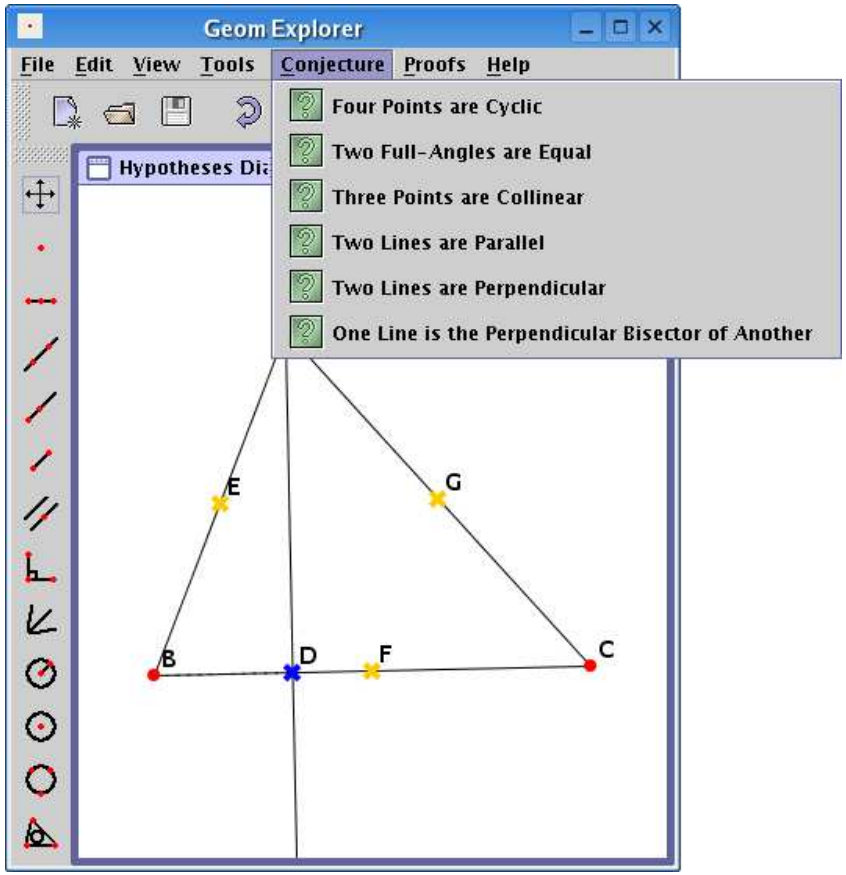
Constructing the Hypotheses Diagram



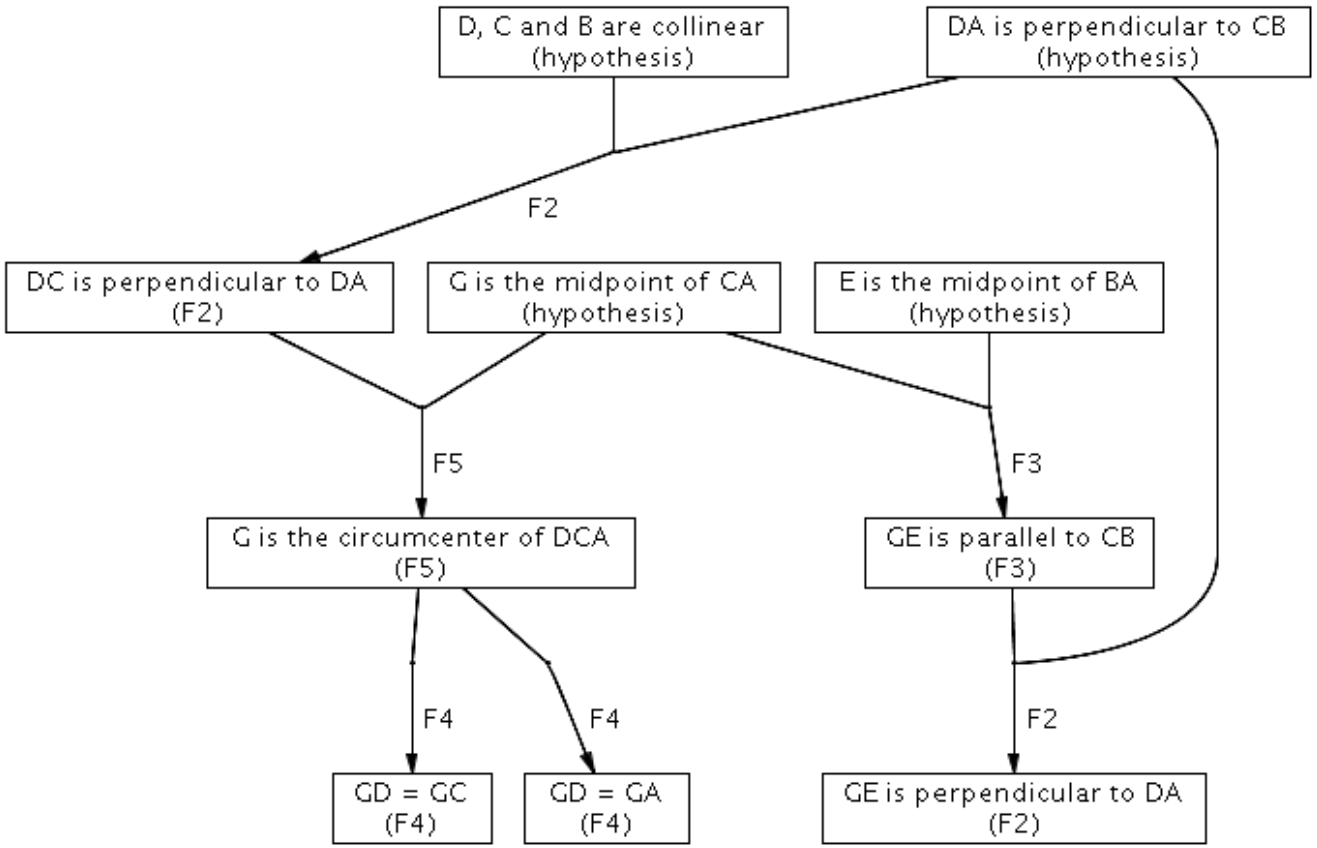
Manipulating the Hypotheses Diagram



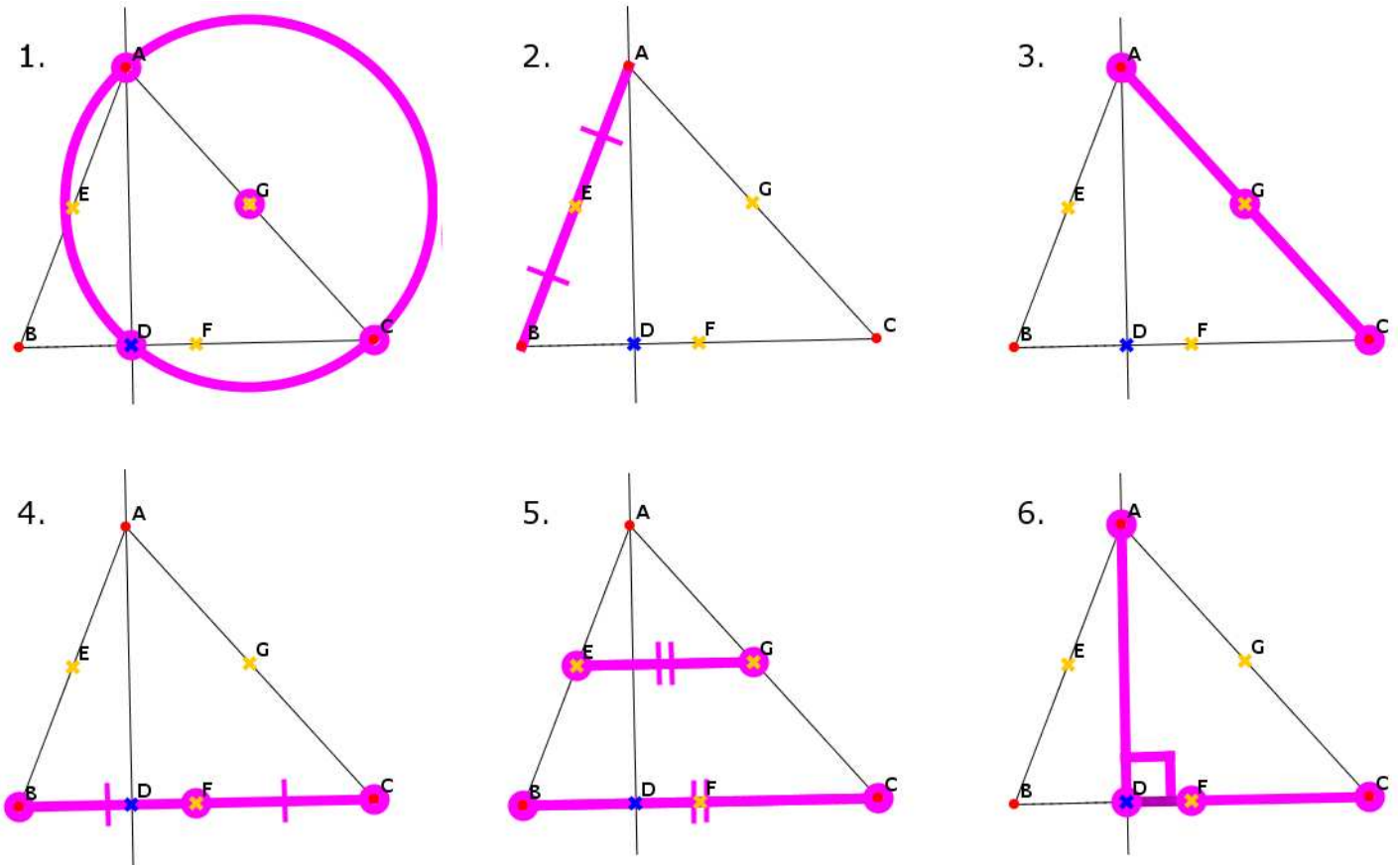
Specifying the Conjecture



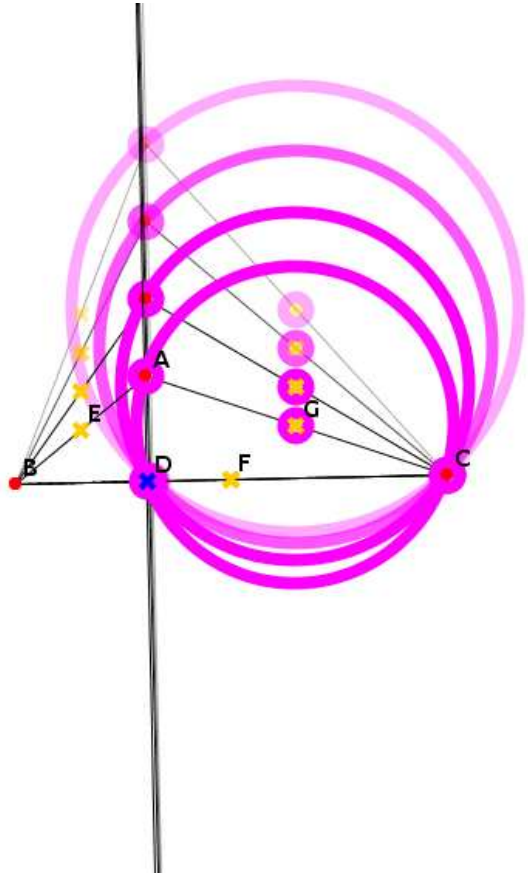
Visualising Forward-Chaining



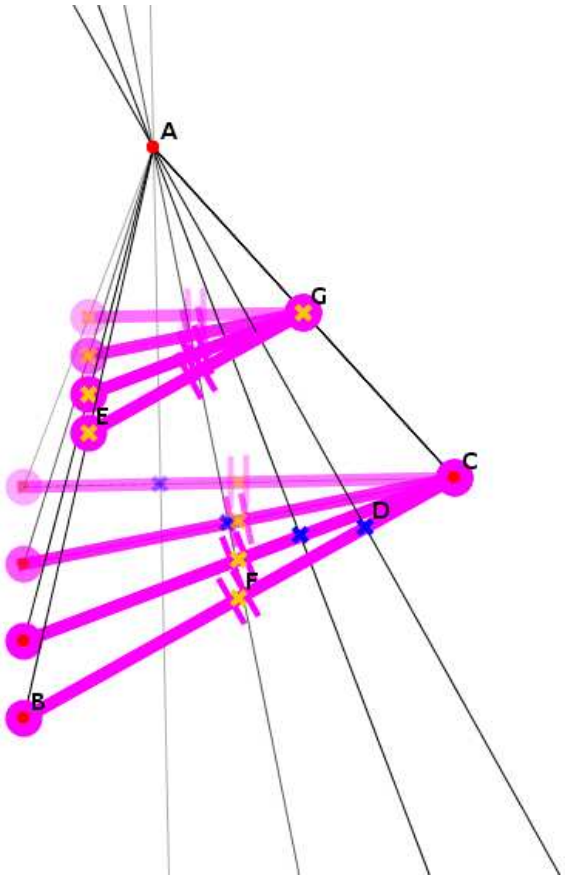
Visualising Geometric Properties



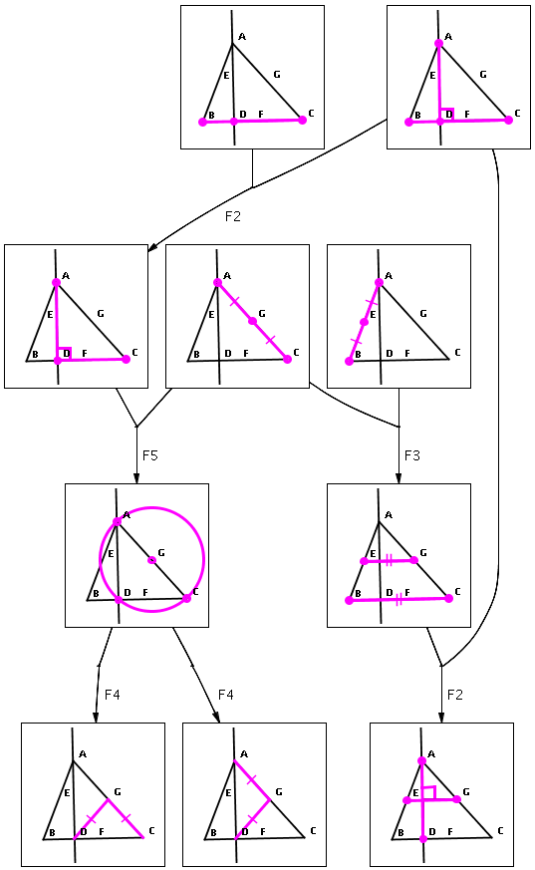
Exploring a Discovered Circumcircle



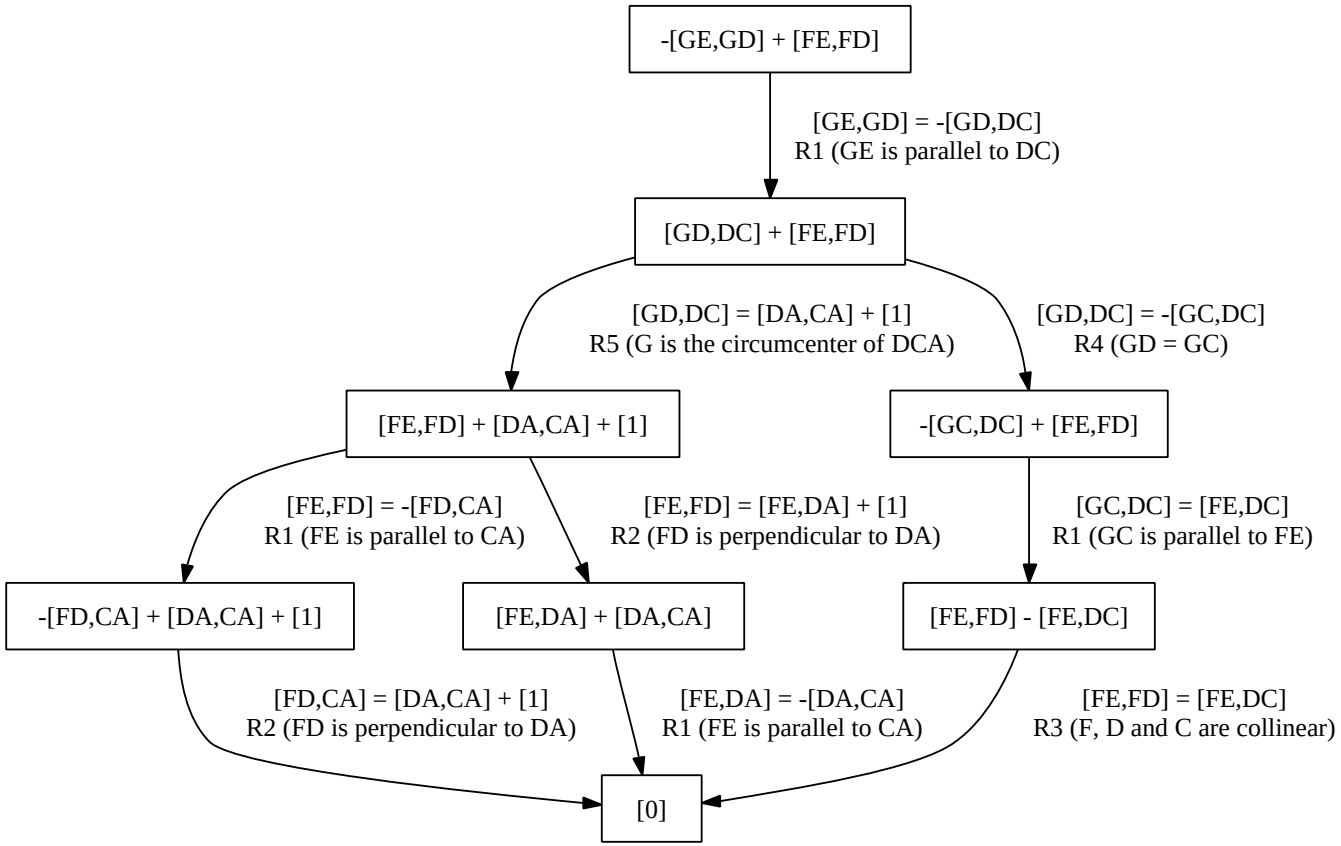
Exploring Discovered Parallel Lines



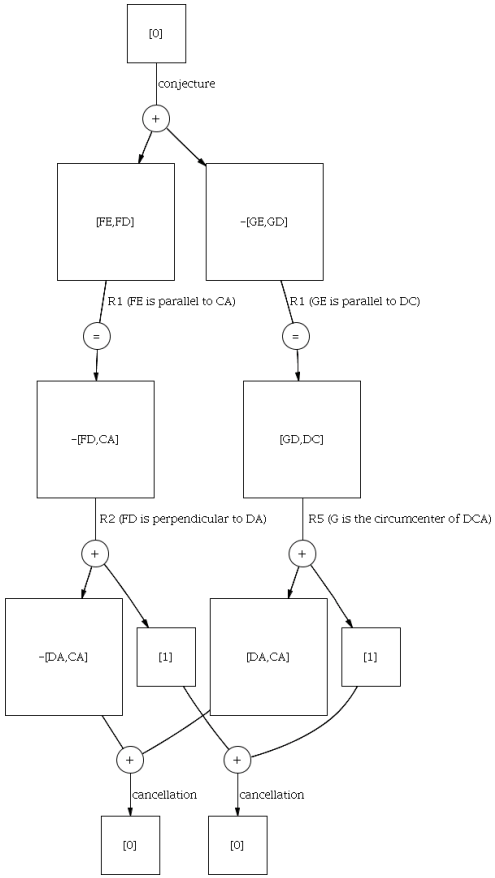
Visualising Forward-Chaining



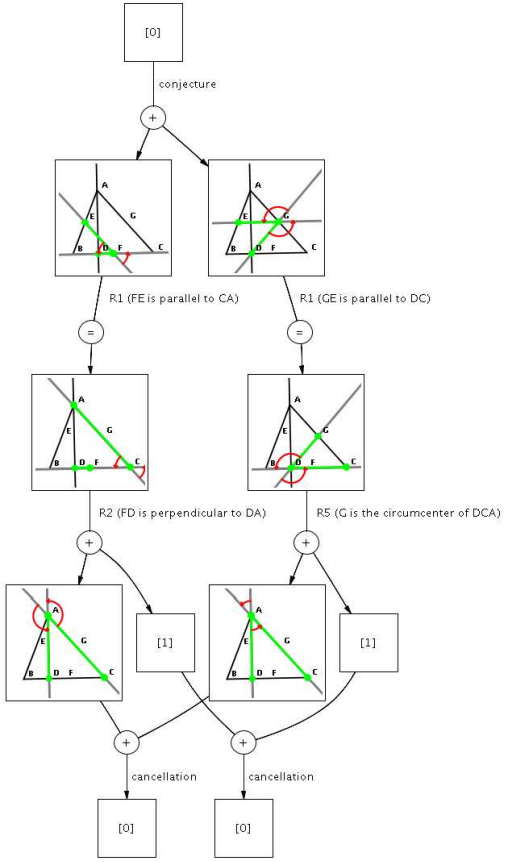
Visualising Multiple Proofs



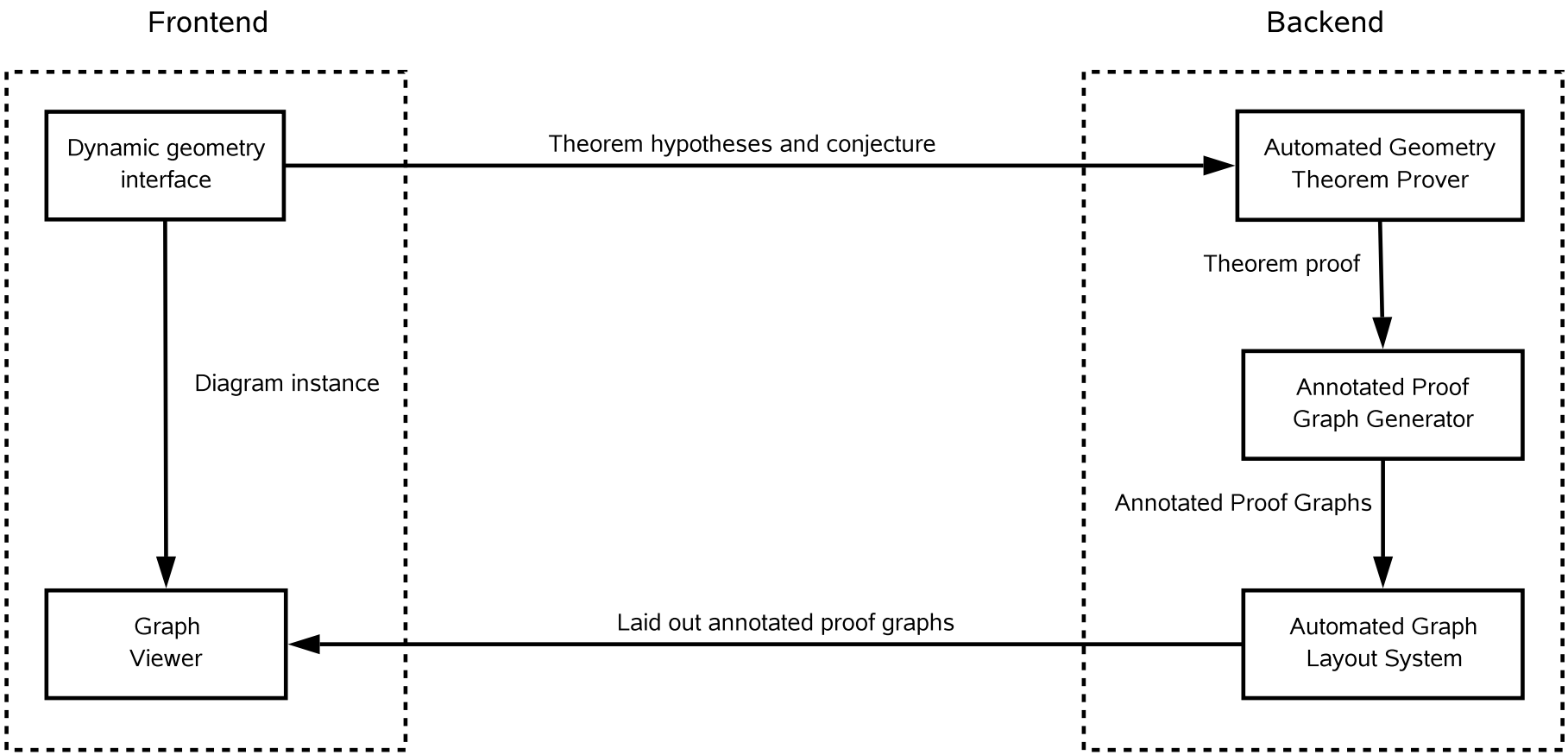
Visualising Backward-Chaining



Visualising Backward-Chaining



System Overview



Related Work and Conclusions

Related Work

Geoview: Generates dynamic geometry diagrams from Pcoq theorem statements.

MMP/Geometer: Generates dynamic geometry diagrams from theorem statements, allows diagrams for theorem input and implements several automatic methods (including area, full-angle, deductive database, Wu's).

GEOTHER: Produces dynamic diagrams from theorem statements and implements several Maple-based algebraic methods (including Wu's and Gröbner basis).

GeoProof: Uses a dynamic diagram for theorem input, interfaces with Coq to provide automatic proving methods (area, Wu's, Gröbner basis).

Cinderella: Dynamic geometry software featuring a randomised theorem checker. An extension allows theorem proving with the Gröbner basis method.

Conclusions

- We found the forward-chaining diagrammatic proofs intuitive, but they cannot prove hard theorems alone.
- We found the backward-chaining diagrammatic proofs less intuitive but they can be used to prove hard theorems.
- We hope to look at ways to generate intuitive and powerful diagrammatic proofs. This can include exploring heuristics to select diagrammatically intuitive proofs, the use of animation, interactive theorem proving with diagrams and using other ATP methods that lend themselves to diagrammatic representation.

Questions?