GCLC, Construction Problems, Coherent Logic and All That

Predrag Janičić
www.matf.bg.ac.rs/~janicic
Automated Reasoning GrOup (ARGO)
Faculty of Mathematics
University of Belgrade, Serbia

University of Strasbourg, France, July 19, 2012.

Faculty of Mathematics, University of Belgrade

- University of Belgrade (http://www.bg.ac.rs)
  - Established in early 1800's
  - One of the oldest and largest in the region
  - Around 90000 students and 4000 members of teaching staff
- Faculty of Mathematics (http://www.matf.bg.ac.rs)
  - Around 1500 students and 80 members of teaching staff
  - Departments for pure mathematics, computer science, astronomy...

Automated Reasoning GrOup (ARGO)

- Area:
  - automated theorem proving
  - decision procedures/SAT/SMT
  - interactive theorem proving (Isabelle)
  - geometry reasoning
- 9 members
- More at: http://argo.matf.bg.ac.rs/

Automated Reasoning GrOup (ARGO) — People


Predrag Janičić


Filip Marić


Sana Stojanović


Danijela Petrović


Vesna Marinković


Ivan Petrović


Mladen Nikolić


Milan Banković

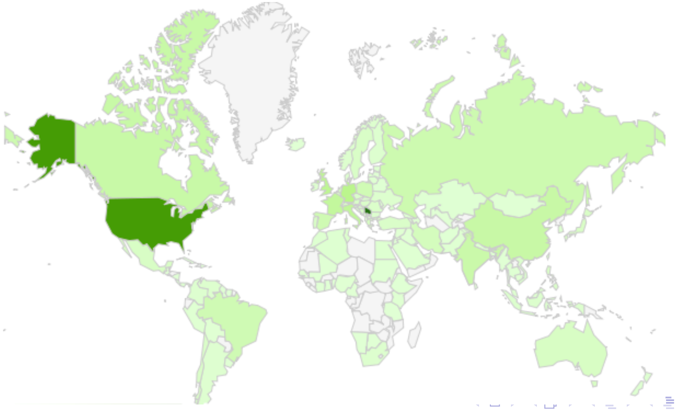

Mirko Stojadinović

GCLC Tool — Main Applications

- GCLC: a geometry tool for
  - mathematical education
  - producing high-quality mathematical illustrations (export to different formats)
  - storing mathematical contents
  - studies of automated geometrical reasoning
- First version released in 1996, still maintained
- Versions for Windows and Linux, freely available from http://www.matf.bg.ac.rs/~janicic/gclc

GCLC — Users

- Thousands of users, used in high-schools and university courses, and for publishing worldwide



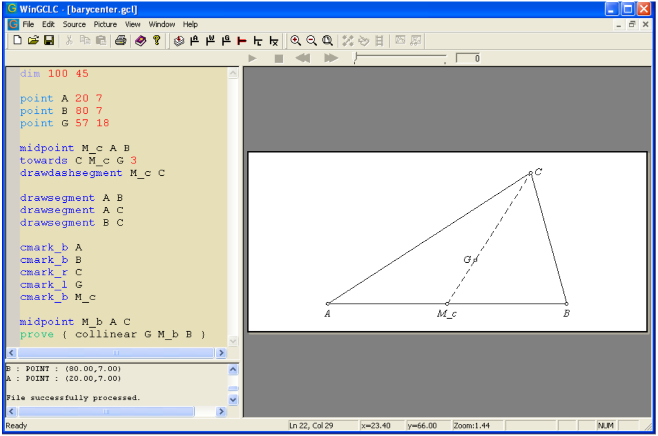
GCLC: Basic Principles

- A construction is a formal procedure, not an image
- GCLC uses a custom geometry language and procedural specifications of geometry figures
- Images can be produced from descriptions, but not vice-versa!
- All instructions are given explicitly, in GCLC language
- Instructions for describing **contents**
- Instructions for describing **presentation**

GCLC Language

- Support for geometrical primitive constructions, compound constructions, transformations, etc.
- Symbolic expressions, while-loops, user-defined procedures
- Conics, 2D and 3D curves, 3D surfaces

Example


WinGCLC - [barycenter.gcl]
File Edit Source Picture View Window Help
dim 100 45
point A 20 7
point B 80 7
point G 57 18
midpoint M_c A B
towards C M_c G 3
drawdashsegment M_c C
drawsegment A B
drawsegment A C
drawsegment B C
cmark_b A
cmark_b B
cmark_r C
cmark_l G
cmark_b M_c
midpoint M_b A C
prove { collinear G M_b B }
B : POINT : (80.00,7.00)
A : POINT : (20.00,7.00)
File successfully processed.
Ready
Ln 22, Col 29
x=23.40
y=66.00
Zoom:1.44
NUM
C
G
A
M_c
B

Theorem Provers Built-into GCLC

- There are three theorem provers built-into GCLC:
  - a theorem prover based on the area method (Chou et.al 1992)
  - a theorem prover based on the Wu's method (Wu 1977)
  - a theorem prover based on the Gröbner bases method (Buchberger 1965)
- Deal with conjectures that corresponds to properties of constructions
- All provers are very efficient and can prove many non-trivial theorems in only milliseconds.
- The theorem provers are tightly built-in: the user has just to state the conjecture, for example:

```
prove { identical O1 O2 }
```

Processing Specifications of Constructions

- Syntactical check
- Semantical check (e.g., whether two concrete points determine a line)
- Deductive check — verifies if a construction is regular (e.g., whether two constructed points never determine a line)

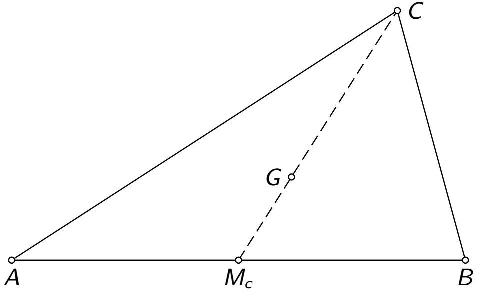
Synthesizing Constructions

- Checking correctness of constructions is all fine...
- ...but can be automate synthesizing of constructions
- Our approach next to be presented (joint work with Vesna Marinković)

Example Problem

Problem: *Construct a triangle ABC given vertices A and B and the barycenter G*

Example Solution



Construction: *Construct the midpoint M_c of the segment AB; then construct the vertex C such that $M_cG : M_cC = 1/3$*

Existing Approaches and Corpora

- Several existing approaches, including:
  - Schreck (1995)
  - Gao and Chou (1998)
  - Gulwani et al. (2011)

Wernick's Corpus

- One of systematically built corpora, created in 1982, some variants in the meanwhile
- Task: construct a triangle given three located points selected from the following list:
  - A, B, C – vertices
  - I, O – incenter and circumcenter
  - H, G – orthocenter and barycenter
  - M_a, M_b, M_c – the side midpoints
  - H_a, H_b, H_c – feet of altitudes
  - T_a, T_b, T_c – intersections of the internal angles bisectors with the opposite sides

Wernick's Problems (2)

139 non-trivial, significantly different, problems; 25 redundant (R) or locus-restricted (L); 72 solvable (S), 16 unsolvable (U); 25 still with unknown status

1.	A, B, O	L
2.	A, B, M_a	S
3.	A, B, M_c	R
4.	A, B, G	S
5.	A, B, H_a	L
6.	A, B, H_c	L
7.	A, B, H	S
8.	A, B, T_a	S
9.	A, B, T_c	[illegible]

[illegible]	[illegible]	57. A, H, I S [9]	85. M_a, M_b, H_a S	113. M_a, T_b, T_c
[illegible]	[illegible]	[illegible] A, T_a, T_b S [9]	86. M_a, M_b, H_c S	114. M_a, T_b, I U [9]
[illegible]	[illegible]	[illegible] T_a, I L	87. M_a, M_b, H S [9]	115. G, H_a, H_b U [9]
[illegible]	[illegible]	[illegible] T_b, T_c S	88. M_a, M_b, T_a U [9]	116. G, H_a, H S
[illegible]	[illegible]	[illegible] I S	89. M_a, M_b, T_c U [9]	117. G, H_a, T_a S
[illegible]	[illegible]	[illegible] M_b S	90. M_a, M_b, I U [10]	118. G, H_a, T_b
[illegible]	[illegible]	[illegible] G S	91. M_a, G, H_a L	119. G, H_a, I
[illegible]	[illegible]	[illegible] L	92. M_a, G, H_b S	120. G, H, T_a U [9]
[illegible]	[illegible]	[illegible] S	93. M_a, G, H S	121. G, H, I U [9]
[illegible]	[illegible]	[illegible] S	94. M_a, G, T_a S	122. G, T_a, T_b
[illegible]	[illegible]	[illegible] L	95. M_a, G, T_b U [9]	123. G, T_a, I
[illegible]	[illegible]	[illegible] U [9]	96. M_a, G, I S [9]	124. H_a, H_b, H_c S
[illegible]	[illegible]	[illegible] S	97. M_a, H_a, H_b S	125. H_a, H_b, H S
[illegible]	[illegible]	[illegible] S	98. M_a, H_a, H L	126. H_a, H_b, T_a S
[illegible]	[illegible]	[illegible] R	99. M_a, H_a, T_a L	127. H_a, H_b, T_c
[illegible]	[illegible]	[illegible] U [9]	100. M_a, H_a, T_b U [9]	128. H_a, H_b, I
[illegible]	[illegible]	[illegible] U [9]	101. M_a, H_a, I S	129. H_a, H, T_a L
[illegible]	[illegible]	[illegible] H_b U [9]	102. M_a, H_b, H_c L	130. H_a, H, T_b U [9]
[illegible]	[illegible]	[illegible] H S	103. M_a, H_b, H S	131. H_a, H, I S [9]
[illegible]	[illegible]	[illegible] T_a S	104. M_a, H_b, T_a S	132. H_a, T_a, T_b
[illegible]	[illegible]	[illegible] H_a, T_b	105. M_a, H_b, T_b S	133. H_a, T_a, I S
[illegible]	[illegible]	[illegible] H_a, I	106. M_a, H_b, T_c U [9]	134. H_a, T_b, T_c
[illegible]	[illegible]	[illegible] O, H, T_a U [9]	107. M_a, H_b, I U [9]	135. H_a, T_b, I
[illegible]	[illegible]	80. O, H, I U [9]	108. M_a, H, T_a U [9]	136. H, T_a, T_b
25. $A,$ [illegible]	[illegible]	81. O, T_a, T_b	109. M_a, H, T_b U [10]	137. H, T_a, I
26. $A, M_a,$ [illegible]	[illegible] S	82. O, T_a, I S [9]	110. M_a, H, I U [10]	138. T_a, T_b, T_c U [11]
27. A, M_a, I S [9]	55. A, H, T_a S	83. M_a, M_b, M_c S	111. M_a, T_a, T_b U [10]	139. T_a, T_b, I S
28. A, M_b, M_c S	56. A, H, T_b U [9]	84. M_a, M_b, G S	112. M_a, T_a, I S	

Basic Approach (1)

- A careful analysis of all available solutions performed
- Solutions use high-level rules, e.g:
  - *if barycenter G and circumcenter O are known, then the orthocenter H can be constructed*
  - *if two triangle vertices are given, then the side bisector can be constructed*
- In total: $\approx$ 70 rules used

Basic Approach (2)

- Implemented in Prolog
- Simple forward chaining mechanism for search procedure
- Solves most of solvable examples from Wernick's list in less than 1s and with the maximal search depth 9
- But... there are too many rules! (it is not problem to search over them, but to invent and systematize them)

Separation of Concepts – Definitions, Lemmas, Construction Steps (1)

Motivating example: *Construct the midpoint M_c of AB and then construct C such that $M_cG : M_cC = 1 : 3$* uses the following:

- M_c is the side midpoint of AB
- G is the barycenter of ABC
- it holds that $M_cG = 1/3M_cC$
- given points X and Y, it is possible to construct the midpoint of the segment XY
- given points X and Y, it is possible to construct a point Z, such that: $XY : XZ = 1 : k$

Separation of Concepts – Definitions, Lemmas, Construction Steps (2)

Motivating example: *Construct the midpoint M_c of AB and then construct C such that* $M_cG : M_cC = 1 : 3$ uses the following:

- M_c is the side midpoint of AB (definition of M_c)
- G is the barycenter of ABC (definition of G)
- it holds that $M_cG = 1/3M_cC$ (lemma)
- given points X and Y, it is possible to construct the midpoint of the segment XY (construction primitive)
- given points X and Y, it is possible to construct a point Z, such that: $XY : XZ = 1 : k$ (construction primitive)

Advanced Approach

- Task: Determine the sets of definitions, lemmas and construction primitives such that all needed high-level (instantiated) construction rules can be built from them
- From:
  - it holds that $M_c G = 1/3 M_c C$ (lemma)
  - given points X and Y, it is possible to construct a point Z, such that: $XY : XZ = 1 : r$ (construction primitive)

  we can derive:
  - given M_c and G, it is possible to construct C

Advanced Approach: Rule Derivation

- Controlled instantiations of lemmas
- All construction rules derived from:
  - 11 definitions (including Wernick's notation)
  - 29 simple lemmas
  - 18 construction primitives (including elementary construction steps)
- Deriving rules is performed once, in preprocessing phase (takes approx. 20s)

Advanced Approach: Re-evaluation

- Another corpus: construct a triangle given three lengths from the following set:
  - $|AB|$, $|BC|$, $|AC|$: lengths of the sides;
  - $|AM_a|$, $|BM_b|$, $|CM_c|$: lengths of the medians;
  - $|AH_a|$, $|BH_b|$, $|CH_c|$: lengths of the altitudes.
- For 17 (out of total of 20) problems, additional: 2 defs, 2 lemmas, and 9 construction steps were needed
- For additional corpora, we expect less and less additions

Output: Constructions in GCLC Form (Example)

```
% free points
point A 30 5
point B 70 5
point G 57 14
% synthesized construction
midpoint M_c A B
towards C M_c G 3
drawdashsegment M_c C
% drawing the triangle ABC
drawsegment A B
drawsegment A C
drawsegment B C
```

Verification

- But... it is not only about synthesis/constructing!
- Verification (correctness proof) is also needed (not "correct by construction")
- "If the objects ... are constructed in the given way, then they meet the specification"
- GCLC theorem provers are used (e.g. the area method, the Gröbner bases method, Wu's method)
- The provers also provide NDG conditions

Existence?

1. But... it is not only about synthesis and verification!
2. Do the constructed objects exist at all? (recall: "If the objects ... are constructed in the given way, then they meet the specification")
3. Using the NDG conditions provided by the provers, we should prove that the constructed objects do exist
4. For this task we are planning to use our prover for coherent logic and generate formal proofs

What is Coherent Logic

- CL formulae are of the form:

$$A_1(\vec{x}) \wedge \ldots \wedge A_n(\vec{x}) \Rightarrow \exists \vec{y}_1\ B_1(\vec{x}, \vec{y}_1) \vee \ldots \vee \exists \vec{y}_m\ B_m(\vec{x}, \vec{y}_m)$$

  A_i are literals, B_i are conjunctions of literals
- No function symbols of arity greater than 0
- No negation
- Intuitionistic logic
- First used by Skolem, recently popularized by Bezem et al.
- Our system — joint work with Mladen Nikolić

Features of CL

- Coherent logic (also: *geometric logic*) is a fragment of FOL
- The problem of deciding $\Gamma \vdash \Phi$ is semi-decidable
- Good features:
  - certain quantification allowed
  - direct, intuitive, readable proofs
  - simple generation of formal (machine verifiable) proofs...

Realm of CL

- A number of theories and theorems can be formulated directly and simply in CL
- Example: large fraction of Euclidean geometry belongs to CL
- Example: *for any two points there is a point between them*
- Conjectures in abstract algebra, confluence theory, lattice theory, and many more (Bezem et al)

CL Proof System

- CL allows a simple, natural proof system (natural deduction style), based on forward ground reasoning
- Existential quantifiers are eliminated by introducing witnesses
- A conjecture is kept unchanged and proved directly (refutation, Skolemization and clausal form are not used)

CL provers

- Euclid by Stevan Kordić and Predrag Janičić (1992)
- CL prover by Marc Bezem and Coquand (2005)
- ML prover by Berghofer and Bezem (2006)
- Geo by Hans de Nivelle (2008)
- ArgoCLP by Sana Stojanović, Vesna Pavlović and Predrag Janičić (2009)
- However, they are still not generally efficient

Example: Proof Generated by ArgoCLP

Let us prove that $p = r$ by reductio ad absurdum.

1. Assume that $p \neq r$.
    2. It holds that the point A is incident to the line q or the point A is not incident to the line q (by axiom of excluded middle).
        3. Assume that the point A is incident to the line q.
            4. From the facts that $p \neq q$, and the point A is incident to the line p, and the point A is incident to the line q, it holds that the lines p and q intersect (by axiom ax_D5).
            5. From the facts that the lines p and q intersect, and the lines p and q do not intersect we get a contradiction.

            Contradiction.
        6. Assume that the point A is not incident to the line q.
            7. From the facts that the lines p and q do not intersect, it holds that the lines q and p do not intersect (by axiom ax_nint_l_l_21).
            8. From the facts that the point A is not incident to the line q, and the point A is incident to the plane α, and the line q is incident to the plane α, and the point A is incident to the line p, and the line p is incident to the plane α, and the lines q and p do not intersect, and the point A is incident to the line r, and the line r is incident to the plane α, and the lines q and r do not intersect, it holds that $p = r$ (by axiom ax_E2).
            9. From the facts that $p = r$, and $p \neq r$ we get a contradiction.

            Contradiction.

Therefore, it holds that $p = r$.

This proves the conjecture.

On the Other Hand: CDCL Solvers

- SAT and SMT solvers are at rather mature stage
- The most efficient ones are CDCL solvers
- However, only universal quantification is allowed
- Producing readable and/or formal proofs is often challenging
- Goal: combine good features of CL and CDCL
- Goal: build an efficient CDCL prover for CL

Three Pillars of Our Approach

The presented approach is motivated by:

Suitability of CL: a number of good features; potentials for obtaining readable and formal proofs

Practical advances in CDCL SAT solving: a huge progress in both high-level and low-level algorithmic techniques

Theoretical advances in CDCL SAT solving: SAT solvers described in terms of state transition systems, which enabled a deeper understanding and a rigorous analysis

Abstract State Transition Systems for SAT

- Inspiration and starting point: transition systems for SAT
- First system: Nieuwenhuis, Oliveras, and Tinelli (2006)
- We build upon: the system by Krstić and Goel (2007)

Krstić and Goel's System

Decide:

$$\frac{l \in L \qquad l, \bar{l} \notin M}{M := M|l}$$

UnitPropag:

$$\frac{l \vee l_1 \vee \ldots \vee l_k \in F \qquad \bar{l}_1, \ldots, \bar{l}_k \in M \qquad l, \bar{l} \notin M}{M := M\ l^i}$$

Conflict:

$$\frac{C = no\_cflct \qquad \bar{l}_1 \vee \ldots \vee \bar{l}_k \in F \qquad l_1, \ldots, l_k \in M}{C := \{l_1, \ldots, l_k\}}$$

Explain:

$$\frac{l \in C \qquad l \vee \bar{l}_1 \vee \ldots \vee \bar{l}_k \in F \qquad l_1, \ldots, l_k \prec l}{C := C \cup \{l_1, \ldots, l_k\} \setminus \{l\}}$$

Learn:

$$\frac{C = \{l_1, \ldots, l_k\} \qquad \bar{l}_1 \vee \ldots \vee \bar{l}_k \notin F}{F := F \cup \{\bar{l}_1 \vee \ldots \vee \bar{l}_k\}}$$

Backjump:

$$\frac{C = \{l, l_1, \ldots, l_k\} \qquad \bar{l} \vee \bar{l}_1 \vee \ldots \vee \bar{l}_k \in F \qquad \text{level } l > m \geq \text{level } l_i}{C := no\_cflct \qquad M := M^m\ \bar{l}^i}$$

Forget:

$$\frac{C = no\_cflct \qquad c \in F \qquad F \setminus c \models c}{F := F \setminus c}$$

Restart:

$$\frac{C = no\_cflct}{M := M^{[0]}}$$

CL state transition system (forward rules)

Decide:

$$\frac{l \in \mathcal{A}(\Sigma) \qquad l \not\uparrow \qquad l \not\downarrow}{M := M|l \qquad \Sigma := \Sigma|}$$

Intro:

$$\frac{\exists \vec{y}\ l \in M \qquad (\exists \vec{y}\ l)\lambda \in \mathcal{A}(\Sigma) \qquad l\lambda\lambda' \not\uparrow \ \text{ for any } \lambda'}{M := M^\frown l[y_1 \mapsto c^{\ell+1}, \ldots, y_k \mapsto c^{\ell+k}]\lambda \qquad \Sigma := \Sigma^\frown c^{\ell+1}, \ldots, c^{\ell+k} \qquad \ell := \ell + k}$$

Unit propagate left:

$$\frac{\mathcal{P} \cup \{l\} \Rightarrow \mathcal{Q} \in^{n_1} \Gamma \qquad \mathcal{P} \Rightarrow \mathcal{Q} \downarrow^m_\lambda \qquad m(\mathcal{P} \cup \mathcal{Q}) \subseteq^{n_2} M \qquad \bar{l}\lambda \not\uparrow \qquad \bar{l}\lambda \not\downarrow}{M := M^{\frown \max(n_1, n_2)} \bar{l}\lambda}$$

Unit propagate right:

$$\frac{\mathcal{P} \Rightarrow \mathcal{Q} \cup \{l\} \in^{n_1} \Gamma \qquad \mathcal{P} \Rightarrow \mathcal{Q} \downarrow^m_\lambda \qquad m(\mathcal{P} \cup \mathcal{Q})^{n_2} \subseteq M \qquad l\lambda \not\uparrow \qquad l\lambda \not\downarrow}{M := M^{\frown \max(n_1, n_2)} l\lambda}$$

Branch end:

$$\frac{\mathcal{C}_2 = \{no\_cflct\} \qquad \mathcal{P} \Rightarrow \mathcal{Q} \in \Gamma \qquad \mathcal{P} \Rightarrow \mathcal{Q} \downarrow}{\mathcal{C}_1 := \mathcal{P} \qquad \mathcal{C}_2 := \mathcal{Q}}$$

CL state transition system (backward rules)

Explain left ∀:

$$\frac{\mathcal{C}_1 \Rightarrow \mathcal{C}_2 \downarrow^m \quad I \in m(\mathcal{C}_1) \quad \mathcal{S} = m^{-1}(I) \quad \mathcal{S} \rightrightarrows \forall \vec{x} p(\vec{v}, \vec{x}) \qquad \mathcal{P} \Rightarrow \mathcal{Q} \cup \{p(\vec{v}', \vec{x}')\} \in \Gamma \quad \mathcal{P} \Rightarrow \mathcal{Q} \downarrow^{m'} \quad m'(\mathcal{P} \cup \mathcal{Q}) \prec I \quad \overline{\forall \vec{x} p(\vec{v}, \vec{x})} \times_\lambda p(\vec{v}', \vec{x}')}{\mathcal{C}_1 := (\forall \vec{x}' \mathcal{P} \cup (\mathcal{C}_1 \setminus \mathcal{S}))\lambda \quad \mathcal{C}_2 := (\exists \vec{x}' \mathcal{Q} \cup \mathcal{C}_2)\lambda}$$

Explain left ∃:

$$\frac{\mathcal{C}_1 \Rightarrow \mathcal{C}_2 \downarrow^m \quad I \in m(\mathcal{C}_1) \quad \mathcal{S} = m^{-1}(I) \quad \mathcal{S} \rightrightarrows_\sigma p(\vec{v}, \vec{x}) \qquad \mathcal{P} \Rightarrow \mathcal{Q} \cup \{\exists \vec{x}' p(\vec{v}', \vec{x}')\} \in \Gamma \quad \mathcal{P} \Rightarrow \mathcal{Q} \downarrow^{m'} \quad m'(\mathcal{P} \cup \mathcal{Q}) \prec I \quad \overline{p(\vec{v}, \vec{x})} \times_\lambda \exists \vec{x}' p(\vec{v}', \vec{x}')}{\mathcal{C}_1 := (\mathcal{P} \cup \forall \vec{x}(\mathcal{C}_1\sigma \setminus \mathcal{S}\sigma))\lambda \quad \mathcal{C}_2 := (\mathcal{Q} \cup \exists \vec{x}(\mathcal{C}_2\sigma))\lambda}$$

Explain right ∀:

$$\frac{\mathcal{C}_1 \Rightarrow \mathcal{C}_2 \downarrow^m \quad I \in m(\mathcal{C}_2) \quad \mathcal{S} = m^{-1}(I) \quad \mathcal{S} \rightrightarrows_\sigma p(\vec{v}, \vec{x}) \qquad \{\forall \vec{x}' p(\vec{v}', \vec{x}')\} \cup \mathcal{P} \Rightarrow \mathcal{Q} \in \Gamma \quad \mathcal{P} \Rightarrow \mathcal{Q} \downarrow^{m'} \quad m'(\mathcal{P} \cup \mathcal{Q}) \prec I \quad p(\vec{v}, \vec{x}) \times_\lambda \overline{\forall \vec{x}' p(\vec{v}', \vec{x}')}}{\mathcal{C}_1 := (\mathcal{P} \cup \forall \vec{x}(\mathcal{C}_1\sigma))\lambda \quad \mathcal{C}_2 := (\mathcal{Q} \cup \exists \vec{x}(\mathcal{C}_2\sigma \setminus \mathcal{S}\sigma))\lambda}$$

Explain right ∃:

$$\frac{\mathcal{C}_1 \Rightarrow \mathcal{C}_2 \downarrow^m \quad I \in m(\mathcal{C}_2) \quad \mathcal{S} = m^{-1}(I) \quad \mathcal{S} \rightrightarrows \exists \vec{x} p(\vec{v}, \vec{x}) \qquad \{p(\vec{v}', \vec{x}')\} \cup \mathcal{P} \Rightarrow \mathcal{Q} \in \Gamma \quad \mathcal{P} \Rightarrow \mathcal{Q} \downarrow^{m'} \quad m'(\mathcal{P} \cup \mathcal{Q}) \prec I \quad \exists \vec{x} p(\vec{v}, \vec{x}) \times_\lambda \overline{p(\vec{v}', \vec{x}')}}{\mathcal{C}_1 := (\forall \vec{x}' \mathcal{P} \cup \mathcal{C}_1)\lambda \quad \mathcal{C}_2 := (\exists \vec{x}' \mathcal{Q} \cup (\mathcal{C}_2 \setminus \mathcal{S}))\lambda}$$

Learn:

$$\frac{\mathcal{C}_2 \neq \{no\_cflct\} \quad \mathcal{C}_1 \Rightarrow \mathcal{C}_2 \notin \Gamma}{\Gamma := \Gamma \frown \mathcal{C}_1 \Rightarrow \mathcal{C}_2}$$

Backjump:

$$\frac{\mathcal{C}_1 \Rightarrow \mathcal{C}_2 \in \Gamma \quad \mathcal{C}_1 \Rightarrow \mathcal{C}_2 \downarrow^m \quad I \in m(\mathcal{C}_1) \quad \mathcal{S} = m^{-1}(I) \quad \mathcal{C}_1 \setminus \mathcal{S} \Rightarrow \mathcal{C}_2 \downarrow_\lambda^{m'} \qquad m' \subseteq m \quad m'(\mathcal{C}_1 \setminus \mathcal{S} \cup \mathcal{C}_2) \subseteq^n M \quad I \in^{n'} M \quad n \leq t < n' \quad \mathcal{S}\lambda \rightrightarrows I'}{M := M^t \frown^n \overline{I}' \quad \Sigma := \Sigma^t \quad \mathcal{C}_1 := \emptyset \quad \mathcal{C}_2 := \{no\_cflct\}}$$

Basic properties

- Sound
- Complete with additional rule for iterative deepening

Example of system operation

(Ax1) $p(x,y) \wedge q(x,y) \Rightarrow \bot$
(Ax2) $s(x) \Rightarrow \exists y\ q(x,y)$
(Ax3) $s(x) \vee q(y,y)$

(Conj) $(\forall x \forall y\ p(x,y)) \Rightarrow \bot$

Rule applied	Σ	$\Gamma \setminus \mathcal{AX}$ (lemmas)	M	$\mathcal{C}_1 \Rightarrow \mathcal{C}_2$
	a	$\emptyset$	$p(x,y)$	$\emptyset \Rightarrow \{no\_cflct\}$
Decide	$a\|$	$\emptyset$	$p(x,y)\|s(x)$	$\emptyset \Rightarrow \{no\_cflct\}$
U.p.r. (Ax2)	$a\|$	$\emptyset$	$p(x,y)\|s(x), \exists y\ q(x,y)$	$\emptyset \Rightarrow \{no\_cflct\}$
Intro	$a\|b$	$\emptyset$	$p(x,y)\|s(x), \exists y\ q(x,y), q(a,b)$	$\emptyset \Rightarrow \{no\_cflct\}$
B.e. (Ax1)	$a\|b$	$\emptyset$	$p(x,y)\|s(x), \exists y\ q(x,y), q(a,b)$	$p(x,y) \wedge q(x,y) \Rightarrow \bot$
E.l. $\exists$ (Ax2)	$a\|b$	$\emptyset$	$p(x,y)\|s(x), \exists y\ q(x,y), q(a,b)$	$\forall y\ p(x,y) \wedge s(x) \Rightarrow \bot$
Learn	$a\|b$	$\forall y\ p(x,y) \wedge s(x) \Rightarrow \bot$	$p(x,y)\|\underline{s(x)}, \exists y\ q(x,y), q(a,b)$	$\forall y\ p(x,y) \wedge s(x) \Rightarrow \bot$
B.j.	a	$\forall y\ p(x,y) \wedge s(x) \Rightarrow \bot$	$p(x,y), \overline{s(x)}$	$\emptyset \Rightarrow \{no\_cflct\}$
U.p.r. (Ax3)	a	$\forall y\ p(x,y) \wedge s(x) \Rightarrow \bot$	$p(x,y), \overline{s(x)}, q(y,y)$	$\emptyset \Rightarrow \{no\_cflct\}$
B.e. (Ax1)	a	$\forall y\ p(x,y) \wedge s(x) \Rightarrow \bot$	$p(x,y), \overline{s(x)}, q(y,y)$	$p(x,y) \wedge q(x,y) \Rightarrow \bot$
E.r. (Ax3)	a	$\forall y\ p(x,y) \wedge s(x) \Rightarrow \bot$	$p(x,y), \overline{s(x)}, q(y,y)$	$p(x,x) \Rightarrow s(z)$
E.r. (lemma)	a	$\forall y\ p(x,y) \wedge s(x) \Rightarrow \bot$	$p(x,y), \overline{s(x)}, q(y,y)$	$p(x,x) \wedge \forall y\ p(z,y) \Rightarrow \bot$

Forward chaining proofs

$$\dfrac{\dfrac{s(x)\lor q(y,y) \quad p(x,y)\land q(x,y)\Rightarrow\bot}{p(x,x)\Rightarrow s(z)} \quad \dfrac{s(x)\Rightarrow \exists y\; q(x,y) \quad p(x,y)\land q(x,y)\Rightarrow\bot}{\forall y\; p(x,y)\land s(x)\Rightarrow\bot}}{p(x,x)\land\forall y\; p(z,y)\Rightarrow\bot}$$

$$\dfrac{s(x)\Rightarrow \exists y\; q(x,y) \quad p(x,y)\land q(x,y)\Rightarrow\bot}{\forall y\; p(x,y)\land s(x)\Rightarrow\bot} \qquad \dfrac{\dfrac{\dfrac{\bot\vdash\bot}{q(a,b)\vdash\bot}\Rightarrow(Ax1)}{\exists y\; q(a,y)\vdash\bot}\exists}{\mathcal{AX}, p(a,y), s(a)\vdash\bot}\Rightarrow(Ax2)$$

$$\dfrac{s(x)\lor q(y,y) \quad p(x,y)\land q(x,y)\Rightarrow\bot}{p(x,x)\Rightarrow s(z)} \qquad \dfrac{\dfrac{s(b)\vdash s(b)}{s(x)\vdash s(b)}\mathit{Inst} \quad \dfrac{\dfrac{\bot\vdash s(b)}{q(a,a)\vdash s(b)}\Rightarrow(Ax1)}{q(y,y)\vdash s(b)}\mathit{Inst}}{\mathcal{AX}, p(a,a)\vdash s(b)}\lor(Ax3)$$

Forward chaining proofs

$$\dfrac{\dfrac{\dfrac{\bot \vdash \bot}{q(a,b) \vdash \bot}\ \Rightarrow (Ax1)}{\exists y\ q(a,y) \vdash \bot}\ \exists}{\mathcal{AX}, p(a,y), s(a) \vdash \bot}\ \Rightarrow (Ax2) \qquad + \qquad \dfrac{\dfrac{s(b) \vdash s(b)}{s(x) \vdash s(b)}\ \mathit{Inst} \qquad \dfrac{\dfrac{\bot \vdash s(b)}{q(a,a) \vdash s(b)}\ \Rightarrow (Ax1)}{q(y,y) \vdash s(b)}\ \mathit{Inst}}{\mathcal{AX}, p(a,a) \vdash s(b)}\ \vee (Ax3)$$

$$\downarrow$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\bot \vdash \bot}{p(a,b) \vdash \bot}\ \Rightarrow (Ax1)}{q(a,b) \vdash \bot}\ \mathit{Inst}}{\exists y\ q(a,y) \vdash \bot}\ \exists}{s(a) \vdash \bot}\ \Rightarrow (Ax2)}{s(x) \vdash \bot}\ \mathit{Inst} \qquad \dfrac{\dfrac{\dfrac{\bot \vdash \bot}{p(a,a) \vdash \bot}\ \Rightarrow (Ax1)}{q(a,a) \vdash \bot}\ \mathit{Inst}}{q(y,y) \vdash \bot}\ \mathit{Inst}}{\mathcal{AX}, p(x,y) \vdash \bot}\ \vee (Ax3)$$

Readable proof

- *Assume* $\forall x \forall y \ p(x, y)$.
- *By (Ax3), it holds* $\forall x \ s(x)$ *or* $\forall y \ q(y, y)$.
- *Assume* $\forall x \ s(x)$.
  - *From* $\forall x \ s(x)$, *it holds* $s(a)$.
  - *By (Ax2), it holds* $\exists y \ q(a, y)$.
  - *From* $\exists y \ q(a, y)$, *there is* b *such that* $q(a, b)$.
  - *From* $\forall x \forall y \ p(x, y)$, *it holds* $p(a, b)$.
  - *By (Ax1), this leads to contradiction.*
- *Assume* $\forall y \ q(y, y)$.
  - *From* $\forall y \ q(y, y)$, *it holds* $q(a, a)$.
  - *From* $\forall x \forall y \ p(x, y)$, *it holds* $p(a, a)$.
  - *By (Ax1), this leads to contradiction.*

Related work

- Euclid (Janičić, Kordić) — CL-geometry, simple backtracking, ground reasoning, iterative deepening
- Bezem's CL prover (Bezem) — CL, simple backtracking, ground reasoning, breadth first search
- Geometric resolution and Geo (de Nivelle) — CL-like, backtracking with lemma learning, ground reasoning
- ArgoCLP (Stojanović, Pavlović, Janičić) — CL, simple backtracking, ground reasoning, iterative deepening
- Model evolution calculus and Darwin (Baumgartner, Tinelli, Fuchs,Pelzer) — clausal fragment, CDCL-style procedure
- EPR (Piskač, de Moura, Bjorner) — clausal fragment without function symbols, CDCL-style procedure

Conclusions and future work

- Goal — integrated framework for:
  - Solving construction problems
  - Visualizing constructions
  - Proving that the construction objects exist
  - Proving that the constructed objects meet the specification