# ArgoCaLyPso — SAT-Inspired Coherent Logic Prover

Mladen Nikolić and Predrag Janičić
Automated Reasoning GrOup (ARGO)
Faculty of Mathematics
University of Belgrade

Belgrade, February, 2011.

# Motivation

- Coherent logic (CL) (also called geometric logic) is a fragment of FOL
- Good features: certain quantification allowed, direct, readable proofs, simple generation of formal proofs...
- However, existing provers for CL are still not very efficient
- SAT and SMT solvers are at rather mature stage
- However, only universal quantification is allowed; producing readable and/or formal proofs is often challenging;
- Goal: build an efficient prover for CL based on SAT/SMT

# 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
- The problem of deciding $\Gamma \vdash \Phi$ is semi-decidable
- First used by Skolem, recently popularized by Bezem et al.

# CL Realm

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

# CL Proof System

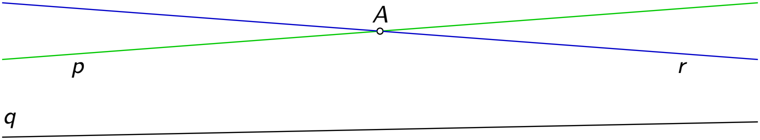
- CL has a natural proof system (natural deduction style), based on forward ground reasoning
- Existential quantifiers are eliminated by introducing witnesses
- A conjecture is kept unchanged and proved directly (refutation, Skolemization and clausal form are not used)
- CL is a suitable framework for producing readable and for producing formal proofs

# ArgoCLP Prover

- Developed by Sana Stojanović, Vesna Pavlović, Predrag Janičić (2009), based on the prover Euclid (developed by Stevan Kordić and Predrag Janičić, 1995.)
- Sound and complete
- A number of techniques that increase efficiency (some of them sacrificing completeness)
- Both formal (Isabelle) and natural language proofs can be exported
- Applied primarily in geometry, proved tens of theorems

# Geometry Example

Assuming that $p \neq q$, and $q \neq r$, and the line $p$ is incident to the plane $\alpha$, and the line $q$ is incident to the plane $\alpha$, and the line $r$ is incident to the plane $\alpha$, and the lines $p$ and $q$ do not intersect, and the lines $q$ and $r$ do not intersect, and the point $A$ is incident to the plane $\alpha$, and the point $A$ is incident to the line $p$, and the point $A$ is incident to the line $r$, show that $p = r$.



# Generated Proof

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.

# Generated Proof (2)

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 $\alpha$, and the line $q$ is incident to the plane $\alpha$, and the point $A$ is incident to the line $p$, and the line $p$ is incident to the plane $\alpha$, 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 $\alpha$, 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.

# CDCL-based CL Prover — ArgoCaLyPso

- Motivation: use forward-chaining with CDCL-like techniques
- In several ways similar to ArgoCLP but with a new search engine
- As the previous version, the prover is forward-chaining based, but guided by DPLL-style search procedure, uses or will use decide, backjump, learn, etc.
- Uses to some extent the architecture of ArgoSAT (by Filip Marić)
- C++, currently $\approx$ 10000 lines of code, but not yet finished

# ArgoCaLyPso and Abstract Transition System

- Described in terms of abstract transition system

Instantiate:

$$\frac{A(x_1, x_2, \ldots, x_i, \ldots, x_n) \in F \qquad a \in \Sigma}{F := F \cup \{A(x_1, x_2, \ldots, a, \ldots, x_n)\}}$$

Intro:

$$\frac{\exists y. A \in F \qquad a \notin \Sigma}{F := F \cup \{A[y \mapsto a]\} \qquad \Sigma := \Sigma \cup \{a\}}$$ where $A$ does not contain free universally quantified variables

Resolve:

$$\frac{l_1 \vee \ldots \vee l_i \vee \ldots \vee l_n \in F \qquad M \models \bar{l} \qquad (l_1 \vee \ldots \vee l_{i_1} \vee l_{i+1} \vee \ldots \vee l_n)\sigma \notin F}{F := F \cup \{(l_1 \vee \ldots \vee l_{i_1} \vee l_{i+1} \vee \ldots \vee l_n)\sigma\}}$$

where $\sigma$ is a most general unifier for $l_i$ and $l$.

# ArgoCaLyPso and Abstract Transition System

- Related to the SAT transition system by Krstić and Goel
- Properties of this system have been formally proved (by Filip Marić)
- Hopefully, ArgoCaLyPso could benefit from that proof

# ArgoCaLyPso and FOL

- The trail contains FOL literals
- The axioms make the initial set of clauses
- The set of clauses can be extended by instances of existing clauses or resolvents between existing clauses and literals from the trail
- Example: if the set of clauses contains $p(x) \Rightarrow q(x) \vee r(x)$ and the trail contains $p(a)$, then the clause $q(a) \vee r(a)$ can be added
- Existential quantifiers are eliminated by introducing witnesses

# ArgoCaLyPso and Search

- One can perform DPLL-like search until all the clauses are satisfied, and then produce new clauses by instantiation, resolution or elimination of existential quantifiers
- The search on one branch is finished if $\bot$ (as in CDCL solvers) or *the conclusion of the goal formula has been reached*
- When one branch is closed, all irrelevant preceding branching points are skipped in further search (backjump)

# ArgoCaLyPso and Search (2)

- The rule decide can be performed on ground clauses $A_1 \vee \ldots \vee A_n$ (in DPLL, decide is applied on implicit clauses $p \vee \neg p$)
- Example: *for three different collinear points A, B, and C one of them is between the other two*
- In ArgoCaLyPso, the *axiom of excluded middle* is explicit, and it is not necessarily used

# Some issues in prover development

- Iterative deepening and object explosion
- Rapid production of new clauses
- Constraining decide rule
- Rule ordering
- Handling equality
- Predicate symmetry
- CL formula is DNF, rather then clause

# Features not implemented yet

- Lemma learning
- Export of formal proofs
- Predicate symmetry for arity greater than 2
- Guiding heuristics and implementational tricks

# Preliminary experiments

- Examples from geometry and rewriting
- Limited comparison to Vampire

# Related work

- Euclid and ArgoCLP
- Marc Bezem's CL prover
- Instance based provers (Darwin)
- EPR solvers

# Conclusions and Future Work

- Hopefully, efficient CDCL-based CL prover
- Hopefully, acceptably efficient SAT solver
- Applications in geometry (and education)
- Applications in program synthesis