CDCL-based Abstract State Transition System for Coherent Logic

Mladen Nikolić   Predrag Janičić

Faculty of Mathematics
University of Belgrade

Fifth Workshop on Formal and Automated Theorem Proving and Applications, Belgrade, 2012.

- FATPA 2011: ArgoCaLyPso - SAT Inspired Coherent Logic Prover
- FATPA 2012: CDCL-based Abstract State Transition System for Coherent Logic

Overview

- CL generalities
- The CDCL based system
- Related work
- Conclusions and further work

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
- The most efficient ones are CDCL solvers
- 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
- No negation
- Intuitionistic logic
- 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)
- It allows for producing readable and formal proofs
- Can a CDCL based system do this?

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

Setup

- Signature: $\Sigma^\infty = \{c^1, c^2, \ldots\}$, Π
- Axioms: $\mathcal{AX}$
- Conjecture: $\forall \overrightarrow{x}(\mathcal{H}^0(\overrightarrow{x}) \Rightarrow \mathcal{G}^0(\overrightarrow{x}))$
- $\mathcal{H} = \mathcal{H}^0(\overrightarrow{x})\lambda$, $\mathcal{G} = \mathcal{G}^0(\overrightarrow{x})\lambda$

Quantified literals

- Quantified atoms
  - $P(a, b)\checkmark$
  - $\forall x P(x, b)\checkmark$
  - $\exists y P(a, y)\checkmark$
  - $\cancel{\forall x \exists y P(x, y)}$
- Negative quantified literals w.r.t. $\mathcal{G} = \exists y Q(a, y) \vee R(b, c)$
  - $P(a, b) \Rightarrow \bot$
  - $\forall \overrightarrow{x}(P(\overrightarrow{x}, b) \Rightarrow R(b, c))$
  - $P(a, b) \Rightarrow Q(a, b) \vee R(b, c)$

Relation $\models$ (entailment of atoms)

- $P(x, y) \models P(a, y)$
- $P(x, y) \models P(a, b)$
- $P(a, b) \models \exists y \; P(a, y)$
- $\{P(x, y), Q(x), R(b)\} \models P(a, b)$

Relation $\perp_\sigma^S$ (conflict)

- $\mathcal{G} = \exists y Q(a, y) \vee R(b, c)$
- $S = \{P(a), \forall x(T(x, b) \Rightarrow \bot)\}$
- $\sigma = [x \mapsto a, z \mapsto b]$
- If $S \subseteq M$, it holds

$$P(x) \Rightarrow \exists y T(y, z) \;\; \perp_\sigma^S \;\; M$$

$$P(x) \Rightarrow \exists y T(y, z) \vee Q(x, b) \;\; \perp_\sigma^S \;\; M$$

States

- State: $S(\Sigma, \Gamma, M, \mathcal{C}_1 \Rightarrow \mathcal{C}_2, \mathcal{C}, ind)$
- $\Sigma_0 = consts(\mathcal{AX} \cup \mathcal{H} \cup \mathcal{G})$
- Initial state: $S_0(\Sigma_0, \mathcal{AX}, \mathcal{H}, \emptyset \Rightarrow \emptyset, \emptyset, |\Sigma_0|)$
- Accepting state: S such that literals from $\mathcal{C}$ are implied by $\mathcal{AX}$ and $\mathcal{H}$ (at level 0).
- Rejecting state: S such that it is not an accepting state and no rules are applicable.
- State can be changed by application of the rules of the system

Decide

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

$$\frac{l \in \mathcal{QA}(\Sigma) \qquad M \not\models l \qquad l \not\perp M}{M := M|l \qquad \Gamma := \Gamma| \qquad \Sigma := \Sigma|}$$

$$\frac{\exists y P(a, y) \in \mathcal{QA}(\Sigma) \qquad M = Q(a)}{M = Q(a)|\ \exists y P(a, y)}$$

Intro

$$\frac{\exists x.l \in M \qquad M \cup \Gamma \setminus \{\exists x.l\} \not\models \exists x.l}{ind := ind + 1 \qquad \Gamma := \Gamma \frown l[x \mapsto c^{ind}] \qquad \Sigma := \Sigma \frown c^{ind}}$$

$$\frac{M = \exists y P(c^1, y) \qquad \Gamma = \exists y P(c^1, y) \qquad \Sigma = c^1 \qquad ind = 1}{ind = 2 \qquad \Gamma = \exists y P(c^1, y)\ P(c^1, c^2) \qquad \Sigma = c^1\ c^2}$$

Unit propagate left

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

$$\frac{\begin{array}{c}\mathcal{A} \cup \{l\} \Rightarrow \mathcal{B} \in^{m_1} \Gamma \qquad \mathcal{A} \Rightarrow \mathcal{B} \perp_\sigma^S M \\ S \subseteq^{m_2} M \qquad M \not\models \{l\sigma\} \Rightarrow \mathcal{G} \qquad \{l\sigma\} \Rightarrow \mathcal{G} \not\perp M\end{array}}{M := M^{\frown \max(m_1,m_2)}\{l\sigma\} \Rightarrow \mathcal{G}}$$

$$\frac{\Gamma = R(x) \Rightarrow \exists y P(x,y) | \qquad M = \forall y (P(a,y) \Rightarrow \mathcal{G}) | \ldots}{M = \forall y (P(a,y) \Rightarrow \mathcal{G})\ R(a) \Rightarrow \mathcal{G} | \ldots}$$

Branch end

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

$$\frac{\mathcal{C} = \emptyset \qquad \mathcal{A} \Rightarrow \mathcal{B} \in^0 \Gamma \qquad \mathcal{A} \Rightarrow \mathcal{B} \perp^S M}{\mathcal{C}_1 \Rightarrow \mathcal{C}_2 := \mathcal{A} \Rightarrow \mathcal{B} \qquad \mathcal{C} := S}$$

$$\frac{P(x) \Rightarrow Q(x) \in \Gamma \qquad M = P(a)\ \ Q(a) \Rightarrow \perp}{\mathcal{C}_1 \Rightarrow \mathcal{C}_2 = P(x) \Rightarrow Q(x) \qquad \mathcal{C} = \{P(a), Q(a) \Rightarrow \perp\}}$$

Explain left

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

$$\frac{\mathcal{A} \Rightarrow \mathcal{B} \cup \{b\} \in^0 \Gamma_{sko} \qquad \mathcal{A} \Rightarrow \mathcal{B} \perp^S M \qquad a \in \mathcal{C}_1 \qquad a\lambda = b\lambda \qquad l \in \mathcal{C} \qquad a\sigma = l\sigma \qquad S \prec_M l}{\mathcal{C}_1 \Rightarrow \mathcal{C}_2 := ((\mathcal{C}_1 \cup \mathcal{A})\lambda \setminus \{a\lambda\} \Rightarrow (\mathcal{C}_2 \cup \mathcal{B})\lambda)_{lift} \qquad \mathcal{C} := \mathcal{C} \setminus \{l\} \cup S}$$

$$\frac{P(x) \Rightarrow \exists y Q(x,y) \in^0 \Gamma \qquad \mathcal{C}_1 \Rightarrow \mathcal{C}_2 = Q(x,y) \wedge L(x,y) \Rightarrow R(x,y) \qquad M = L(x,y)\ \forall y(R(a,y) \Rightarrow \perp)\ P(a)\ \exists y Q(a,y)}{\mathcal{C}_1 \Rightarrow \mathcal{C}_2 = P(x) \wedge \forall z L(x,z) \Rightarrow \exists y R(x,y)}$$

Learn

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

$$\frac{\mathcal{C} \neq \emptyset \qquad \mathcal{C}_1 \Rightarrow \mathcal{C}_2 \notin \Gamma}{\Gamma := \Gamma \frown^0 \mathcal{C}_1 \Rightarrow \mathcal{C}_2}$$

$$\frac{\Gamma = \exists x R(x) |\ P(x) \Rightarrow \exists y Q(y) \qquad \mathcal{C}_1 \Rightarrow \mathcal{C}_2 = Q(x) \Rightarrow R(x)}{\Gamma = \exists x R(x)\ Q(x) \Rightarrow R(x) |\ P(x) \Rightarrow \exists y Q(y)}$$

Backjump

$$\frac{C = \{l, l_1, \ldots, l_k\} \qquad \bar{l} \vee \bar{l}_1 \vee \ldots \vee \bar{l}_k \in F \qquad \text{level } l_i \leq m < \text{level } l}{C := \emptyset \qquad M := M^m\ \bar{l}}$$

$$\frac{\begin{array}{c}\mathcal{C}_1 \Rightarrow \mathcal{C}_2 \in^0 \Gamma \qquad \mathcal{C}_1 \setminus \{l\} \Rightarrow \mathcal{C}_2 \perp_\sigma^{\mathcal{C}\setminus\{l'\}} M \\ \mathcal{C} \subseteq^{n'} M \qquad \mathcal{C} \setminus \{l'\} \subseteq^n M \qquad n \leq m < n'\end{array}}{\mathcal{C} := \emptyset \qquad M := M^{m \frown n}\{l\sigma\} \Rightarrow \mathcal{G} \qquad \Gamma := \Gamma^m \qquad \Sigma := \Sigma^m}$$

$$\frac{P(x) \Rightarrow Q(x) \in^0 \Gamma \qquad M = \forall x(Q(x) \Rightarrow \perp) |\ R(a) |\ L(x) |\ P(b)}{M = \forall x(Q(x) \Rightarrow \perp)\ \forall x(P(x) \Rightarrow \mathcal{G})}$$

Basic properties

- Sound
- Complete

Forward chaining proofs

- Extraction enabled by
  - Coherent form
  - Avoiding refutation and Skolemization
- Restricted decide
- Exploiting conflict analysis

Forward chaining proofs

$$P(x) \Rightarrow \exists y Q(x,y) \qquad Q(x,y) \wedge L(x,y) \Rightarrow R(x,y)$$
$$P(x) \wedge \forall z L(x,z) \Rightarrow \exists y R(x,y)$$

$$M = P(a)\ L(a,z)$$
$$M = P(a)\ L(a,z)\ \exists y Q(a,y)$$
$$M = P(a)\ L(a,z)\ \exists y Q(a,y)\ Q(a,b)$$
$$M = P(a)\ L(a,z)\ \exists y Q(a,y)\ Q(a,b)\ R(a,b)$$

Related work

- Euclid and ArgoCLP
- Marc Bezem's CL prover
- Model evolution calculus and Darwin
- EPR solvers

Conclusions and future work

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