CDCL-based Abstract State Transition System for Coherent Logic

Mladen Nikolić   Predrag Janičić
Faculty of Mathematics
University of Belgrade

SVARM Workshop
Manchester, June 30-July 01, 2012.

Overview

- *Note:* this work will be presented also at CICM/Calculemus 2012 conference
- Overview of the talk:
  - Coherent logic (CL) and our motivation
  - The CDCL-based abstract transition system for CL
  - Related work
  - Conclusions and further work

What is Coherent Logic

- CL formulae are of the form:

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

  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.

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:

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

Setup

- Signature: Σ; axioms: $\mathcal{AX}$; conjecture: $\forall \vec{x}(\mathcal{H}^0(\vec{x}) \Rightarrow \mathcal{G}^0(\vec{x}))$
- $\mathcal{H} = \mathcal{H}^0(\vec{x})\lambda, \;\; \mathcal{G} = \mathcal{G}^0(\vec{x})\lambda$
- State: $S(\Sigma, \Gamma, M, \mathcal{C}_1, \mathcal{C}_2, \ell)$
- Initial state: $S_0(\Sigma_0, \mathcal{AX}, \mathcal{H}, \emptyset, \emptyset, |\Sigma_0|)$
- Final states: those in which no rules are applicable
- Slightly extended CL language:

$$\forall \vec{x}\; p_1(\vec{v}, \vec{x}) \wedge \ldots \wedge \forall \vec{x}\; p_n(\vec{v}, \vec{x}) \Rightarrow \exists \vec{y}\; q_1(\vec{v}, \vec{y}) \vee \ldots \vee \exists \vec{y}\; q_m(\vec{v}, \vec{y})$$

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 $\forall$:

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

Explain left $\exists$:

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

Explain right $\forall$:

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

Explain right $\exists$:

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

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:

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

Decide

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

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

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

Generalized resolution for conflict analysis

$$\frac{\mathcal{P} \Rightarrow \mathcal{Q} \cup \{\exists \vec{y} p(\vec{x}, \vec{y})\} \quad \{p(\vec{x}', \vec{y}')\} \cup \mathcal{P}' \Rightarrow \mathcal{Q}'}{(\mathcal{P} \cup \forall \vec{y}' \mathcal{P}' \Rightarrow \mathcal{Q} \cup \exists \vec{y}' \mathcal{Q}')\lambda}$$

$$\frac{\mathcal{P} \Rightarrow \mathcal{Q} \cup \{p(\vec{x}, \vec{y})\} \quad \{\forall \vec{x}' p(\vec{x}', \vec{y}')\} \cup \mathcal{P}' \Rightarrow \mathcal{Q}'}{(\forall \vec{x} \mathcal{P} \cup \mathcal{P}' \Rightarrow \exists \vec{x} \mathcal{Q} \cup \mathcal{Q}')\sigma}$$

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	$a\vert$	$\emptyset$	$p(x,y)\vert s(x)$	$\emptyset \Rightarrow \{no\_cflct\}$
U.p.r. (Ax2)	$a\vert$	$\emptyset$	$p(x,y)\vert s(x), \exists y\ q(x,y)$	$\emptyset \Rightarrow \{no\_cflct\}$
Intro	$a\vert b$	$\emptyset$	$p(x,y)\vert s(x), \exists y\ q(x,y), q(a,b)$	$\emptyset \Rightarrow \{no\_cflct\}$
B.e. (Ax1)	$a\vert b$	$\emptyset$	$p(x,y)\vert s(x), \exists y\ q(x,y), q(a,b)$	$p(x,y) \wedge q(x,y) \Rightarrow \bot$
E.l. $\exists$ (Ax2)	$a\vert b$	$\emptyset$	$p(x,y)\vert s(x), \exists y\ q(x,y), q(a,b)$	$\forall y\ p(x,y) \wedge s(x) \Rightarrow \bot$
Learn	$a\vert b$	$\forall y\ p(x,y) \wedge s(x) \Rightarrow \bot$	$p(x,y)\vert s(x), \exists y\ q(x,y), q(a,b)$	$\forall y\ p(x,y) \wedge s(x) \Rightarrow \bot$
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

$$\dfrac{\dfrac{\dfrac{\bot \vdash \bot}{q(a,b) \vdash \bot}\Rightarrow (Ax1)}{\exists y\ q(a,y) \vdash \bot}\exists}{\mathcal{AX},\, p(a,y),\, s(a) \vdash \bot}\Rightarrow (Ax2)$$

$$+$$

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

$$\downarrow$$

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

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

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

Setup

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

Quantified literals

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

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)

- $\mathcal{G} = \exists y Q(a, y) \vee R(b, c)$
- $S = \{P(a), \forall x(T(x, b) \Rightarrow \perp)\}$
- $\sigma = [x \mapsto a, z \mapsto b]$
- 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: $S(\Sigma, \Gamma, M, \mathcal{C}_1, \mathcal{C}_2, \ell)$
- $\Sigma_0 = consts(\mathcal{AX} \cup \mathcal{H} \cup \mathcal{G})$
- Initial state: $S_0(\Sigma_0, \mathcal{AX}, \mathcal{H}, \emptyset, \emptyset, |\Sigma_0|)$
- Accepting state: S such that literals that make $\mathcal{C}_1 \Rightarrow \mathcal{C}_2$ conflicting 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

Forward chaining proofs

- Extraction enabled by
  - Preservation of coherent form
  - Avoiding refutation and Skolemization
- Proof extraction from conflict analysis

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