Theorem Provers, SMT, and Interpolation

Philipp Rümmer
Uppsala University
Sweden

CP meets CAV
June 27th 2012

Outline

- Some SMT challenges from verification
- Quantifiers in SMT
  $\rightarrow$ First-order version of SMT
- Computation of Craig interpolants

Disclaimer

- Highly biased challenges (my point of view)
- Some results shown are not by myself
- Some results shown are joint work

Reasoning + first-order logic (FOL)

	SAT/SMT solvers
	DPLL(T), CDCL, Nelson-Oppen
	E-matching, heuristics
	Complete on ground fragment
	Many built-in theories

Reasoning + first-order logic (FOL)

First-order provers	**SAT/SMT solvers**
Resolution, superposition, tableaux, etc.	DPLL(T), CDCL, Nelson-Oppen
(Free) variables, unification	E-matching, heuristics
Complete for FOL	Complete on ground fragment
	Many built-in theories

Reasoning + first-order logic (FOL)

First-order provers	**SAT/SMT solvers**
Resolution, superposition, tableaux, etc.	DPLL(T), CDCL, Nelson-Oppen
(Free) variables, unification	E-matching, heuristics
Complete for FOL	Complete on ground fragment
	Many built-in theories
Tailored to algebra, logic, etc.	Tailored to verification; (usually) incomplete on quantified problems

Classical paradigms in logical reasoning

Analytic $\rightarrow$ Case-based	**Synthetic** $\rightarrow$ Consequence-based
Gentzen-style sequents	Syllogisms
Tableaux	Hilbert-style calculi
Hypertableaux	Resolution
Model evolution	Superposition
Model generation	Knuth-Bendix
DPLL	Gröbner bases

Classical paradigms in logical reasoning

Analytic → Case-based	**Synthetic** → Consequence-based
Gentzen-style sequents	Syllogisms
Tableaux	Hilbert-style calculi
Hypertableaux	Resolution
Model evolution	Superposition
Model generation	Knuth-Bendix
DPLL	Gröbner bases

SAT combines both paradigms

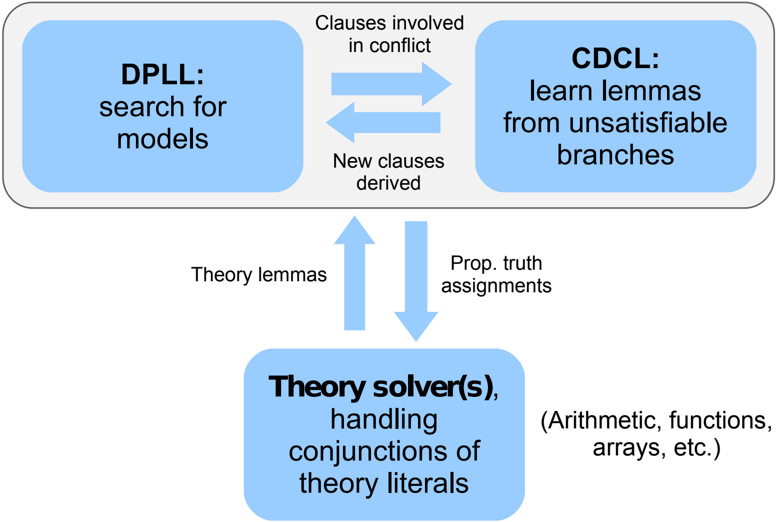
DPLL:
search for models

Clauses involved in conflict →

← New clauses derived

CDCL:
learn lemmas from unsatisfiable branches

From SAT to SMT

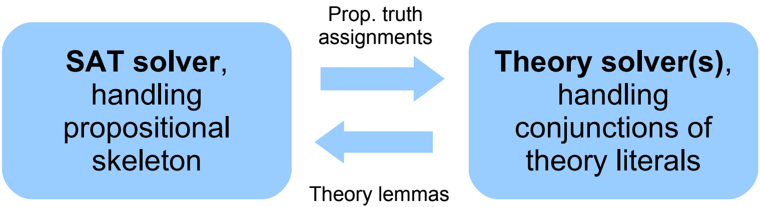


Some challenging theories

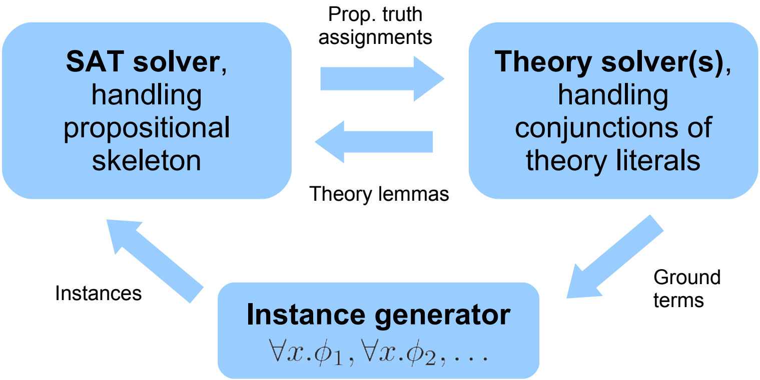
- Integers
- Non-linear arithmetic
- Floating-point arithmetic
- Words/strings

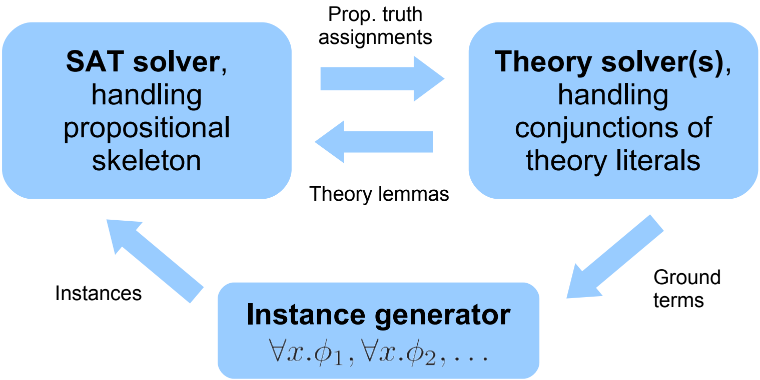
Quantifiers in SMT

(one of the main challenges)


SAT solver,
handling
propositional
skeleton
Prop. truth
assignments
Theory lemmas
Theory solver(s),
handling
conjunctions of
theory literals

Quantifiers





- E-matching (Simplify, Stanford Pascal Verifier)
- Complete instantiation; counterexample-based [Ge, de Moura, 09]
- Superposition [de Moura, BjÃ¸rner, 09]

E-Matching

Matching of **triggers** (modulo equations)

$$\frac{}{\Gamma, \forall \bar{x}.\phi \vdash \qquad , \Delta}$$

E-Matching

Matching of triggers (modulo equations)

$$\frac{}{\Gamma, \forall \bar{x}.\phi[t[\bar{x}]] \ \vdash \qquad\qquad , \Delta}$$

- Identify triggers (sub-expressions) in quantified formulae

E-Matching

Matching of **triggers** (modulo equations)

$$\frac{}{\Gamma, \forall \bar{x}.\phi[t[\bar{x}]] \;\vdash\; \psi[t[\bar{s}]], \Delta}$$

- Identify triggers (sub-expressions) in quantified formulae
- Check for **matching ground terms**

E-Matching

Matching of triggers (modulo equations)

$$\frac{\Gamma, \forall \bar{x}.\phi[t[\bar{x}]], [\bar{x}/\bar{s}]\phi[t[\bar{x}]] \vdash \psi[t[\bar{s}]], \Delta}{\Gamma, \forall \bar{x}.\phi[t[\bar{x}]] \vdash \psi[t[\bar{s}]], \Delta}$$

- Identify triggers (sub-expressions) in quantified formulae
- Check for **matching ground terms**
- Create **ground instances** resulting from match

E-Matching

Matching of triggers (modulo equations)

$$\frac{\Gamma, \forall \bar{x}.\phi[t[\bar{x}]], [\bar{x}/\bar{s}]\phi[t[\bar{x}]] \;\vdash\; \psi[t[\bar{s}]], \Delta}{\Gamma, \forall \bar{x}.\phi[t[\bar{x}]] \;\vdash\; \psi[t[\bar{s}]], \Delta}$$

- Identify triggers (sub-expressions) in quantified formulae
- Check for **matching ground terms**
- Create **ground instances** resulting from match

```
\forall int a, i, v;
   sel(sto(a, i, v), i) = v

\forall int a, i1, i2, v;
   (i1 != i2 ->
   sel(sto(a, i1, v), i2) = sel(a, i2))
```

E-Matching

Matching of triggers (modulo equations)

$$\frac{\Gamma, \forall \bar{x}.\phi[t[\bar{x}]], [\bar{x}/\bar{s}]\phi[t[\bar{x}]] \ \vdash \ \psi[t[\bar{s}]], \Delta}{\Gamma, \forall \bar{x}.\phi[t[\bar{x}]] \ \vdash \ \psi[t[\bar{s}]], \Delta}$$

- Identify triggers (sub-expressions) in quantified formulae
- Check for **matching ground terms**
- Create **ground instances** resulting from match

```
\forall int a, i, v;
   sel(sto(a, i, v), i) = v

\forall int a, i1, i2, v;
   (i1 != i2 ->
   sel(sto(a, i1, v), i2) = sel(a, i2))
```

Examples

$$b \doteq \mathtt{sto}(a, 1, 2) \rightarrow \mathtt{sel}(b, 2) \doteq \mathtt{sel}(a, 2)$$

```
\forall int a, i, v;
   sel(sto(a, i, v), i) = v

\forall int a, i1, i2, v;
   (i1 != i2 ->
   sel(sto(a, i1, v), i2) = sel(a, i2))
```

Examples

$$b \doteq \mathtt{sto}(a, 1, 2) \;\rightarrow\; \mathtt{sel}(b, 2) \doteq \mathtt{sel}(a, 2)$$

$$b \doteq \mathtt{sto}(a, 1, 2) \;\rightarrow\; \exists x.\, \mathtt{sel}(b, x) \doteq \mathtt{sel}(a, 2)$$

```
\forall int a, i, v;
   sel(sto(a, i, v), i) = v

\forall int a, i1, i2, v;
   (i1 != i2 ->
   sel(sto(a, i1, v), i2) = sel(a, i2))
```

Examples

$$b \doteq \mathtt{sto}(a, 1, 2) \rightarrow \mathtt{sel}(b, 2) \doteq \mathtt{sel}(a, 2)$$
$$b \doteq \mathtt{sto}(a, 1, 2) \rightarrow \exists x.\, \mathtt{sel}(b, x) \doteq \mathtt{sel}(a, 2)$$
$$b \doteq \mathtt{sto}(a, 1, 2) \rightarrow \exists x.\, \mathtt{sel}(b, x + 1) \doteq \mathtt{sel}(a, 2)$$
$$b \doteq \mathtt{sto}(a, 1, 2) \rightarrow \exists x.\, \mathtt{sel}(b, x) \doteq \mathtt{sel}(a, x)$$

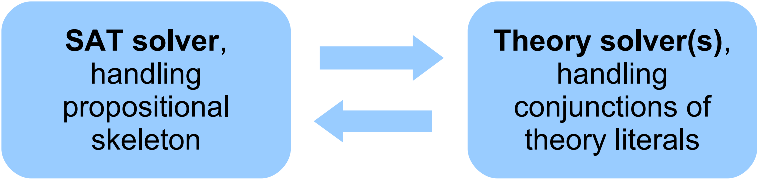
```
\forall int a, i, v;
   sel(sto(a, i, v), i) = v

\forall int a, i1, i2, v;
   (i1 != i2 ->
   sel(sto(a, i1, v), i2) = sel(a, i2))
```

E-Matching

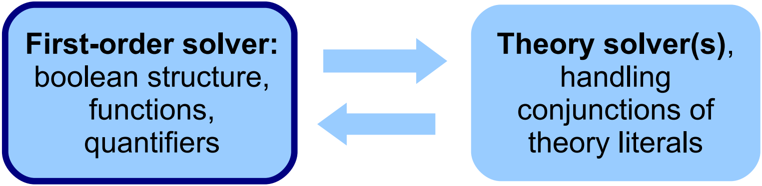
- Heuristic → **incomplete**
- Good for “simple” instances
- **User guidance** possible → triggers
- But also brittle, easy to choose wrong triggers
- **Fast** → only **ground** reasoning

- Restrictions particularly problematic for **“deductive verification”**
  ⇒ Complicated specifications without good triggers

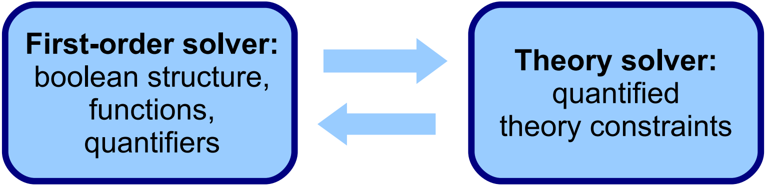
Small engines



Small engines



Small engines



First-order SMT

Putting things together

Current choices:

KE-tableau/DPLL	FOL
Theory procedures	Arithmetic
Free variables + constraints	Quantifiers
E-matching	Axiomatisation of theories

- Interesting completeness results
- Experimental implementation: PRINCESS
- More details in [LPAR'08], [LPAR'12]
- Long-term goal: framework as general as SMT

In the example

$$\overline{\mathcal{AX} \vdash b \doteq \mathtt{sto}(a,1,2) \rightarrow \exists x.\, \mathtt{sel}(b,x) \doteq \mathtt{sel}(a,2)}$$

$\mathcal{AX}$ =

```
\forall int a, i, v;
   sel(sto(a, i, v), i) = v
\forall int a, i1, i2, v;
   (i1 != i2 ->
   sel(sto(a, i1, v), i2) = sel(a, i2))
```

In the example

$$\frac{\overline{\mathcal{AX}, b \doteq \mathtt{sto}(a, 1, 2) \vdash \exists x.\, \mathtt{sel}(b, x) \doteq \mathtt{sel}(a, 2)}}{\mathcal{AX} \vdash b \doteq \mathtt{sto}(a, 1, 2) \rightarrow \exists x.\, \mathtt{sel}(b, x) \doteq \mathtt{sel}(a, 2)}$$

$\mathcal{AX}$ =

```
\forall int a, i, v;
   sel(sto(a, i, v), i) = v
\forall int a, i1, i2, v;
   (i1 != i2 ->
   sel(sto(a, i1, v), i2) = sel(a, i2))
```

In the example

$$\dfrac{\dfrac{\overline{\mathcal{AX}, b \doteq \mathtt{sto}(a,1,2) \vdash \mathtt{sel}(b,X) \doteq \mathtt{sel}(a,2)}}{\mathcal{AX}, b \doteq \mathtt{sto}(a,1,2) \vdash \exists x.\, \mathtt{sel}(b,x) \doteq \mathtt{sel}(a,2)}}{\mathcal{AX} \vdash b \doteq \mathtt{sto}(a,1,2) \rightarrow \exists x.\, \mathtt{sel}(b,x) \doteq \mathtt{sel}(a,2)}$$

$\mathcal{AX}$ =

```
\forall int a, i, v;
   sel(sto(a, i, v), i) = v
\forall int a, i1, i2, v;
   (i1 != i2 ->
   sel(sto(a, i1, v), i2) = sel(a, i2))
```

In the example

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\ldots, 1 \dot{\neq} X \rightarrow \mathtt{sel}(b, X) \doteq \mathtt{sel}(a, 2) \vdash \mathtt{sel}(b, X) \doteq \mathtt{sel}(a, 2)}}{\mathcal{AX}, b \doteq \mathtt{sto}(a, 1, 2) \vdash \mathtt{sel}(b, X) \doteq \mathtt{sel}(a, 2)}}{\mathcal{AX}, b \doteq \mathtt{sto}(a, 1, 2) \vdash \exists x.\, \mathtt{sel}(b, x) \doteq \mathtt{sel}(a, 2)}}{\mathcal{AX} \vdash b \doteq \mathtt{sto}(a, 1, 2) \rightarrow \exists x.\, \mathtt{sel}(b, x) \doteq \mathtt{sel}(a, 2)}$$

$\mathcal{AX}$ =

```
\forall int a, i, v;
   sel(sto(a, i, v), i) = v
\forall int a, i1, i2, v;
   (i1 != i2 ->
   sel(sto(a, i1, v), i2) = sel(a, i2))
```

In the example

$$
\dfrac{\dfrac{\dfrac{\dfrac{\overset{*}{[1 \doteq\!\!\!\!/\; X]}\\ \vdots}{\ldots, 1 \not\doteq X \rightarrow \texttt{sel}(b, X) \doteq \texttt{sel}(a, 2) \vdash \texttt{sel}(b, X) \doteq \texttt{sel}(a, 2)}}{\mathcal{AX}, b \doteq \texttt{sto}(a, 1, 2) \vdash \texttt{sel}(b, X) \doteq \texttt{sel}(a, 2)}}{\mathcal{AX}, b \doteq \texttt{sto}(a, 1, 2) \vdash \exists x.\, \texttt{sel}(b, x) \doteq \texttt{sel}(a, 2)}}{\mathcal{AX} \vdash b \doteq \texttt{sto}(a, 1, 2) \rightarrow \exists x.\, \texttt{sel}(b, x) \doteq \texttt{sel}(a, 2)}
$$

$\mathcal{AX} =$

```
\forall int a, i, v;
   sel(sto(a, i, v), i) = v
\forall int a, i1, i2, v;
   (i1 != i2 ->
   sel(sto(a, i1, v), i2) = sel(a, i2))
```

Linear integer arithmetic + uninterpreted predicates:

$$t \;::=\; \alpha \;\Big|\; x \;\Big|\; c \;\Big|\; \alpha t + \cdots + \alpha t$$

$$\phi \;::=\; \phi \land \phi \;\Big|\; \phi \lor \phi \;\Big|\; \neg\phi \;\Big|\; \forall x.\phi \;\Big|\; \exists x.\phi$$
$$\Big|\; t \doteq 0 \;\Big|\; t \mathrel{\dot{\geq}} 0 \;\Big|\; t \mathrel{\dot{\leq}} 0 \;\Big|\; \alpha \mid t \;\Big|\; p(t, \ldots, t)$$

- t ... terms
- ϕ ... formulae
- x ... variables
- c ... constants
- p ... uninterpreted predicates (fixed arity)
- α ... integer literals ($\mathbb{Z}$)

The base logic [LPAR'08]

Linear integer arithmetic + uninterpreted predicates:

$$t ::= \alpha \mid x \mid c \mid \alpha t + \cdots + \alpha t$$

$$\phi ::= \phi \wedge \phi \mid \phi \vee \phi \mid \neg\phi \mid \forall x.\phi \mid \exists x.\phi$$
$$\mid t \doteq 0 \mid t \,\dot{\geq}\, 0 \mid t \,\dot{\leq}\, 0 \mid \alpha \mid t \mid p(t, \ldots, t)$$

- Functions encoded as relations (later)
- Subsumes FOL and Presburger arithmetic (PA)
- Valid formulae are not enumerable [Halpern, 1991]

Constrained sequents

Notation used here:

$$\underbrace{\Gamma \vdash \Delta}_{\substack{\text{Antecedent, Succedent} \\ \text{(sets of formulae)}}} \; \underbrace{\Downarrow C}_{\substack{\text{Constraint/approximation} \\ \text{(formula)}}}$$

Definition

$\Gamma \vdash \Delta \Downarrow C$ is *valid* if the formula $C \to \bigwedge \Gamma \to \bigvee \Delta$ is valid.

Iterative proof construction

$$\Gamma \;\vdash\; \Delta \;\Downarrow ?$$

Iterative proof construction

analytic reasoning
about input formula
(SMT-like) $\uparrow$

$$\Gamma \vdash \Delta \Downarrow ?$$

Iterative proof construction

analytic reasoning
about input formula
(SMT-like) $\uparrow$

$$\begin{array}{c} \Gamma_1 \vdash \Delta_1 \Downarrow ? \\ \vdots \\ \Gamma \vdash \Delta \Downarrow ? \end{array}$$

Iterative proof construction

analytic reasoning
about input formula
(SMT-like) $\uparrow$

$$\frac{\Gamma_2 \;\vdash\; \Delta_2 \;\Downarrow ?}{\Gamma_1 \;\vdash\; \Delta_1 \;\Downarrow ?}$$

$$\vdots$$

$$\Gamma \;\vdash\; \Delta \;\Downarrow ?$$

Iterative proof construction

analytic reasoning about input formula (SMT-like) $\uparrow$

$$\cfrac{\cfrac{\Gamma_3 \vdash \Delta_3 \Downarrow ?}{\Gamma_2 \vdash \Delta_2 \Downarrow ?}}{\Gamma_1 \vdash \Delta_1 \Downarrow ?}$$

$$\vdots$$

$$\Gamma \vdash \Delta \Downarrow ?$$

Iterative proof construction

analytic reasoning
about input formula
(SMT-like)

$\uparrow$

$$\begin{array}{c} * \\ \vdots \\ \dfrac{\dfrac{\Gamma_3 \;\vdash\; \Delta_3 \;\Downarrow ?}{\Gamma_2 \;\vdash\; \Delta_2 \;\Downarrow ?}}{\Gamma_1 \;\vdash\; \Delta_1 \;\Downarrow ?} \\ \vdots \\ \Gamma \;\vdash\; \Delta \;\Downarrow ? \end{array}$$

Iterative proof construction

analytic reasoning
about input formula
(SMT-like) $\uparrow$

$$\begin{array}{c} * \\ \vdots \\ \Gamma_3 \vdash \Delta_3 \Downarrow ? \\ \hline \Gamma_2 \vdash \Delta_2 \Downarrow ? \\ \hline \Gamma_1 \vdash \Delta_1 \Downarrow ? \\ \vdots \\ \Gamma \vdash \Delta \Downarrow ? \end{array}$$

$\downarrow$ propagation
of constraints

Iterative proof construction

analytic reasoning about input formula (SMT-like) ↑

$$
\cfrac{\cfrac{\cfrac{*}{\vdots}}{\Gamma_3 \vdash \Delta_3 \Downarrow C_1}}{\cfrac{\Gamma_2 \vdash \Delta_2 \Downarrow ?}{\cfrac{\Gamma_1 \vdash \Delta_1 \Downarrow ?}{\cfrac{\vdots}{\Gamma \vdash \Delta \Downarrow ?}}}}
$$

↓ propagation of constraints

Iterative proof construction

analytic reasoning about input formula (SMT-like) ↑

$$\begin{array}{c} * \\ \vdots \\ \Gamma_3 \vdash \Delta_3 \Downarrow C_1 \\ \hline \Gamma_2 \vdash \Delta_2 \Downarrow C_2 \\ \hline \Gamma_1 \vdash \Delta_1 \Downarrow ? \\ \vdots \\ \Gamma \vdash \Delta \Downarrow ? \end{array}$$

↓ propagation of constraints

Iterative proof construction

analytic reasoning about input formula (SMT-like) $\uparrow$

$$
\dfrac{\dfrac{\dfrac{\vdots^{*}}{\Gamma_3 \vdash \Delta_3 \Downarrow C_1}}{\Gamma_2 \vdash \Delta_2 \Downarrow C_2}}{\dfrac{\Gamma_1 \vdash \Delta_1 \Downarrow C_3}{\vdots \atop \Gamma \vdash \Delta \Downarrow ?}}
$$

$\downarrow$ propagation of constraints

Iterative proof construction

analytic reasoning about input formula (SMT-like) $\uparrow$

$$
\begin{array}{c}
* \\
\vdots \\
\dfrac{\dfrac{\Gamma_3 \vdash \Delta_3 \Downarrow C_1}{\Gamma_2 \vdash \Delta_2 \Downarrow C_2}}{\Gamma_1 \vdash \Delta_1 \Downarrow C_3} \\
\vdots \\
\Gamma \vdash \Delta \Downarrow C
\end{array}
$$

$\downarrow$ propagation of constraints

Iterative proof construction

analytic reasoning about input formula (SMT-like) $\uparrow$

$$
\dfrac{\dfrac{\dfrac{\vdots\,*}{\Gamma_3 \vdash \Delta_3 \Downarrow C_1}}{\Gamma_2 \vdash \Delta_2 \Downarrow C_2}}{\Gamma_1 \vdash \Delta_1 \Downarrow C_3}
\qquad
\begin{array}{c}\vdots\\ \Gamma \vdash \Delta \Downarrow C\end{array}
$$

$\downarrow$ propagation of constraints

- Constraints are **simplified** during propagation
- If C is **valid**, then so is $\Gamma \vdash \Delta$
- If C is **satisfiable**, it describes a solution for $\Gamma \vdash \Delta$
- If C is unsatisfiable, expand the proof tree further . . .

Iterative proof construction

$$
\begin{array}{c}
* \\
\vdots \\
\Gamma_3 \vdash \Delta_3 \Downarrow C_1 \\
\hline
\Gamma_2 \vdash \Delta_2 \Downarrow C_2 \\
\hline
\Gamma_1 \vdash \Delta_1 \Downarrow C_3 \\
\vdots \\
\Gamma \vdash \Delta \Downarrow C
\end{array}
$$

analytic reasoning about input formula (SMT-like) $\uparrow$

$\downarrow$ propagation of constraints

- Constraints are **simplified** during propagation
- If C is **valid**, then so is $\Gamma \vdash \Delta$
- If C is **satisfiable**, it describes a solution for $\Gamma \vdash \Delta$
- If C is unsatisfiable, expand the proof tree further . . .

- **Theories have two roles:** analytic + propagation

A few proof rules

$$\frac{\Gamma \vdash \phi, \Delta \Downarrow C \qquad \Gamma \vdash \psi, \Delta \Downarrow D}{\Gamma \vdash \phi \wedge \psi, \Delta \Downarrow C \wedge D} \text{ AND-RIGHT}$$

$$\frac{\Gamma, [x/c]\phi, \forall x.\phi \vdash \Delta \Downarrow [x/c]C}{\Gamma, \forall x.\phi \vdash \Delta \Downarrow \exists x.C} \text{ ALL-LEFT}$$

(c is fresh)

$$\frac{\Gamma, p(\bar{s}) \vdash p(\bar{t}), \bar{s} \doteq \bar{t}, \Delta \Downarrow C}{\Gamma, p(\bar{s}) \vdash p(\bar{t}), \Delta \Downarrow C} \text{ PRED-UNIFY}$$

$$\frac{*}{\Gamma, \phi_1, \ldots \vdash \psi_1, \ldots, \Delta \Downarrow \neg\phi_1 \vee \cdots \vee \psi_1 \vee \cdots} \text{ CLOSE}$$

(selected formulae are predicate-free)

A few proof rules

$$\frac{\Gamma \vdash \phi, \Delta \Downarrow C \qquad \Gamma \vdash \psi, \Delta \Downarrow D}{\Gamma \vdash \phi \wedge \psi, \Delta \Downarrow C \wedge D} \text{ AND-RIGHT}$$

$$\frac{\Gamma, [x/c]\phi, \forall x.\phi \vdash \Delta \Downarrow [x/c]C}{\Gamma, \forall x.\phi \vdash \Delta \Downarrow \exists x.C} \text{ ALL-LEFT}$$

(c is fresh)

$$\frac{\Gamma, p(\bar{s}) \vdash p(\bar{t}), \bar{s} \doteq \bar{t}, \Delta \Downarrow C}{\Gamma, p(\bar{s}) \vdash p(\bar{t}), \Delta \Downarrow C} \text{ PRED-UNIFY}$$

$$\frac{*}{\Gamma, \phi_1, \ldots \vdash \psi_1, \ldots, \Delta \Downarrow \neg\phi_1 \vee \cdots \vee \psi_1 \vee \cdots} \text{ CLOSE}$$

(selected formulae are predicate-free)

+ Theory rules!

In the example

$$
\dfrac{\dfrac{\dfrac{\dfrac{\overset{*}{[\,1 \not\doteq X\,]}}{\ldots,\ 1 \not\doteq X \rightarrow \texttt{sel}(b,X) \doteq \texttt{sel}(a,2) \ \vdash\ \texttt{sel}(b,X) \doteq \texttt{sel}(a,2)}}{\mathcal{AX},\ b \doteq \texttt{sto}(a,1,2) \ \vdash\ \texttt{sel}(b,X) \doteq \texttt{sel}(a,2)}}{\mathcal{AX},\ b \doteq \texttt{sto}(a,1,2) \ \vdash\ \exists x.\ \texttt{sel}(b,x) \doteq \texttt{sel}(a,2)}}{\mathcal{AX} \ \vdash\ b \doteq \texttt{sto}(a,1,2) \ \rightarrow\ \exists x.\ \texttt{sel}(b,x) \doteq \texttt{sel}(a,2)}
$$

In the example

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{\vdash\ 1 \not\doteq X \ \Downarrow ?} \qquad \begin{array}{c}\vdots\\ \mathtt{sel}(b,X) \doteq \mathtt{sel}(a,2)\ \vdash\ \mathtt{sel}(b,X) \doteq \mathtt{sel}(a,2)\ \Downarrow ?\end{array}}{\ldots, 1 \not\doteq X \to \mathtt{sel}(b,X) \doteq \mathtt{sel}(a,2)\ \vdash\ \mathtt{sel}(b,X) \doteq \mathtt{sel}(a,2)\ \Downarrow ?}}{\mathcal{AX}, b \doteq \mathtt{sto}(a,1,2)\ \vdash\ \mathtt{sel}(b,X) \doteq \mathtt{sel}(a,2)\ \Downarrow ?}}{\mathcal{AX}, b \doteq \mathtt{sto}(a,1,2)\ \vdash\ \exists x.\ \mathtt{sel}(b,x) \doteq \mathtt{sel}(a,2)\ \Downarrow ?}}{\mathcal{AX}\ \vdash\ b \doteq \mathtt{sto}(a,1,2) \ \to\ \exists x.\ \mathtt{sel}(b,x) \doteq \mathtt{sel}(a,2)\ \Downarrow ?}$$

In the example

$$\dfrac{\dfrac{\dfrac{\dfrac{*}{\vdash\ 1 \not\doteq X\ \Downarrow 1 \not\doteq X} \qquad \begin{array}{c} * \\ \vdots \\ \mathtt{sel}(b,X) \doteq \mathtt{sel}(a,2)\ \vdash\ \mathtt{sel}(b,X) \doteq \mathtt{sel}(a,2)\ \Downarrow \mathit{true} \end{array}}{\ldots, 1 \not\doteq X \to \mathtt{sel}(b,X) \doteq \mathtt{sel}(a,2)\ \vdash\ \mathtt{sel}(b,X) \doteq \mathtt{sel}(a,2)\ \Downarrow 1 \not\doteq X}}{\mathcal{AX}, b \doteq \mathtt{sto}(a,1,2)\ \vdash\ \mathtt{sel}(b,X) \doteq \mathtt{sel}(a,2)\ \Downarrow 1 \not\doteq X}}{\dfrac{\mathcal{AX}, b \doteq \mathtt{sto}(a,1,2)\ \vdash\ \exists x.\ \mathtt{sel}(b,x) \doteq \mathtt{sel}(a,2)\ \Downarrow \exists X. 1 \not\doteq X}{\mathcal{AX}\ \vdash\ b \doteq \mathtt{sto}(a,1,2)\ \to\ \exists x.\ \mathtt{sel}(b,x) \doteq \mathtt{sel}(a,2)\ \Downarrow \mathit{true}}}$$

Correctness

Lemma (Soundness)

It's sound!

Lemma (Completeness)

Complete for fragments:

- *FOL*
- *PA*
- *Purely existential formulae*
- *Purely universal formulae*
- *Universal formulae with finite parametrisation (same as $\mathcal{ME}$(LIA))*

Functions

Functions

Functions almost like in SMT:

- Terms are always **flattened**
- n-ary **function** f becomes $(n+1)$-ary **predicate** f_p
  E.g.

$$
\begin{array}{rcl}
g(f(x), a) & \rightsquigarrow & f(x) = c \wedge g(c, a) = d \\
 & \rightsquigarrow & f_p(x, c) \wedge g_p(c, a, d)
\end{array}
$$

Functions

Functions almost like in SMT:

- Terms are always **flattened**
- n-ary **function** f becomes $(n+1)$-ary **predicate** f_p
  E.g.

$$
\begin{aligned}
g(f(x), a) \quad &\rightsquigarrow \quad f(x) = c \wedge g(c, a) = d \\
&\rightsquigarrow \quad f_p(x, c) \wedge g_p(c, a, d)
\end{aligned}
$$

- Axioms necessary: **Totality + Functionality**

$$
\begin{aligned}
&\forall \bar{x}. \exists y.\ f_p(\bar{x}, y) \\
&\forall \bar{x}, y_1, y_2.\ (f_p(\bar{x}, y_1) \to f_p(\bar{x}, y_2) \to y_1 \doteq y_2)
\end{aligned}
$$

Functions

Functions almost like in SMT:

- Terms are always **flattened**
- n-ary **function** f becomes $(n+1)$-ary **predicate** f_p
  E.g.

$$
\begin{aligned}
g(f(x), a) \quad & \leadsto \quad f(x) = c \wedge g(c, a) = d \\
& \leadsto \quad f_p(x, c) \wedge g_p(c, a, d)
\end{aligned}
$$

- Axioms necessary: **Totality + Functionality**

$$
\begin{aligned}
& \forall \bar{x}. \exists y.\ f_p(\bar{x}, y) \\
& \forall \bar{x}, y_1, y_2.\ (f_p(\bar{x}, y_1) \rightarrow f_p(\bar{x}, y_2) \rightarrow y_1 \doteq y_2)
\end{aligned}
$$

- Very closely resembles **congruence closure**

Relative completeness

In **SMT solvers**:

- Choice of triggers determines **provability**
- Bad triggers $\rightarrow$ bad luck

In the **first-order SMT calculus**:

- Choice of triggers determines **performance**
- Regardless of triggers, **the same formulae are provable**
- E-matching is complemented by **free variables + unification**

Practicality

	AUFLIA+p (193)	**AUFLIA-p** (193)
Z3	191	191
PRINCESS	**145**	**137**
CVC3	132	128

- Implementation of our calculus in PRINCESS
- Unsatisfiable AUFLIA benchmarks from SMT-comp 2011
- Intel Core i5 2-core, 3.2GHz, timeout 1200s, 4Gb
- `http://www.philipp.ruemmer.org/princess.shtml`

- Currently running: **CASC 2012**

Related work

- $\mathcal{ME}$(LIA):
  [Baumgartner, Tinelli, Fuchs, 08], [Baumgartner, Tinelli, 11]
- SPASS+T
  [Prevosto, Waldmann, 06]
- DPLL($\mathcal{SP}$)
  [de Moura, Bjørner, 08]
- Complete instantiation
  [Ge, de Moura, 09]
- Saturation + theories, e.g.
  [Stickel, 85], [Bürchert, 90],
  [Bachmair, Ganzinger, Waldmann, 94],
  [Korovin, Voronkov, 07],
  [Althaus, Kruglov, Weidenbach, 09]
- . . .

Conclusion

- **Overall challenge:**
  Combine the **theories** and **performance** of SMT solvers with the **completeness** of FOL provers
- Presented work is one step in this direction

Ongoing work:

- Better **unification** on term level
- Better heuristics for introducing **free variables**
- Lemma learning
- Generalisation to other theories

Computation of Craig Interpolants

Motivation: inference of invariants

Generic verification problem ("safety")

$$\{\ \texttt{pre}\ \}\ \texttt{while (}\star\texttt{) Body}\ \{\ \texttt{post}\ \}$$

Standard approach: loop rule using invariant

$$\frac{\texttt{pre} \Rightarrow \phi \qquad \{\ \phi\ \}\ \texttt{Body}\ \{\ \phi\ \} \qquad \phi \Rightarrow \texttt{post}}{\{\ \texttt{pre}\ \}\ \texttt{while (}\star\texttt{) Body}\ \{\ \texttt{post}\ \}}$$

How to compute ϕ automatically?

From intermediate assertions to invariants

{pre} Body; Body {post} ?

↓

Bounded model checking problem ✓

↓

Compute intermediate assertion ψ_1

↓

{pre} Body {ψ_1}   {ψ_1} Body {post}

[McMillan, 2003]

From intermediate assertions to invariants

{pre} Body; Body {post} ?

↓

Bounded model checking problem ✓

↓

Compute intermediate assertion ψ_1

↓

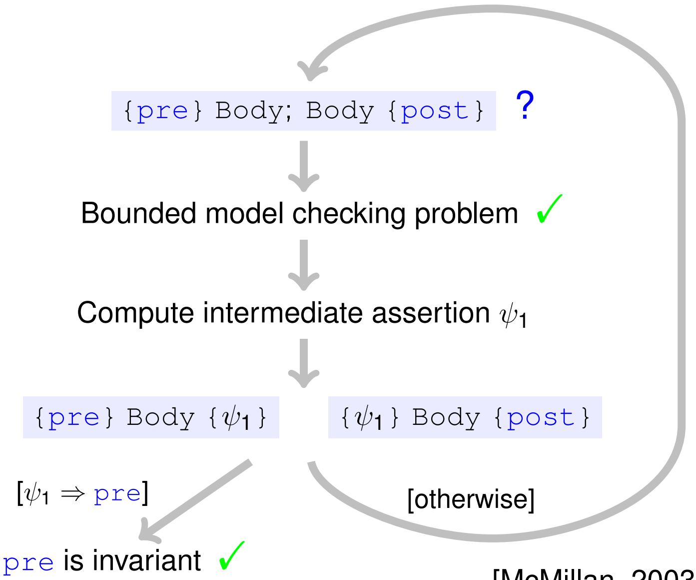
{pre} Body {ψ_1} {ψ_1} Body {post}

[$\psi_1 \Rightarrow$ pre]

pre is invariant ✓

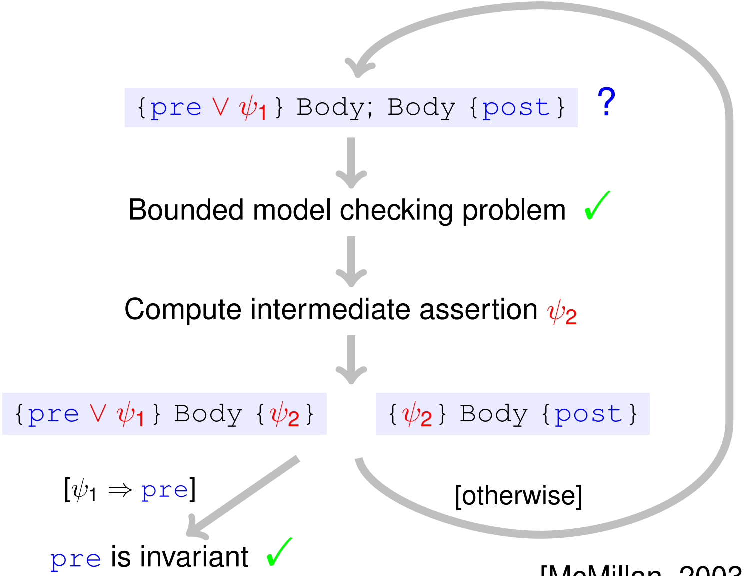
[McMillan, 2003]

From intermediate assertions to invariants



[McMillan, 2003]

From intermediate assertions to invariants



[McMillan, 2003]

From intermediate assertions to invariants

$\{\texttt{pre} \lor \psi_1\}$ Body; Body $\{\texttt{post}\}$?

Bounded model checking problem ✓

Compute intermediate assertion ψ_2

$\{\texttt{pre} \lor \psi_1\}$ Body $\{\psi_2\}$ $\{\psi_2\}$ Body $\{\texttt{post}\}$

$[\psi_2 \Rightarrow \texttt{pre} \lor \psi_1]$

[otherwise]

$\texttt{pre} \lor \psi_1$ is invariant ✓

[McMillan, 2003]

From intermediate assertions to invariants

{ pre $\vee$ ψ_1 } Body; Body { post } ?

Bounded model checking problem ✓

Compute intermediate assertion ψ_2

{ pre $\vee$ ψ_1 } Body { ψ_2 }

{ ψ_2 } Body { post }

[$\psi_2 \Rightarrow$ pre $\vee$ ψ_1]

...

pre $\vee$ ψ_1 is invariant ✓

[McMillan, 2003]

How to compute intermediate assertions?

VC generation

`{ pre }`	`pre` (s_0)
`Body;`	$\rightarrow$ `Body` (s_0, s_1)
`Body`	$\rightarrow$ `Body` (s_1, s_2)
`{ post }`	$\rightarrow$ `post` (s_2)

How to compute intermediate assertions?

VC generation

```
{ pre }
 Body;
 Body
{ post }
```

$$\begin{array}{l} \texttt{pre}\,(s_0) \\ \rightarrow \texttt{Body}\,(s_0, s_1) \\ \rightarrow \texttt{Body}\,(s_1, s_2) \\ \rightarrow \texttt{post}\,(s_2) \end{array}$$

Theorem (Craig, 1957)

Suppose $A \rightarrow C$ is a valid FOL implication.
Then there is a formula I (an interpolant) such that

- $A \rightarrow I$ *and* $I \rightarrow C$ *are valid,*
- *every non-logical symbol of I occurs in both A and C.*

How to compute intermediate assertions?

VC generation →

`{ pre }`	$\texttt{pre}\,(s_0)$	$A(s_0, s_1)$
`Body;`	$\rightarrow \texttt{Body}\,(s_0, s_1)$	↓
- - - - - - - - - - - -	- - - - - - - - - - - -	$I(s_1)$
`Body`	$\rightarrow \texttt{Body}\,(s_1, s_2)$	↓
`{ post }`	$\rightarrow \texttt{post}\,(s_2)$	$C(s_1, s_2)$

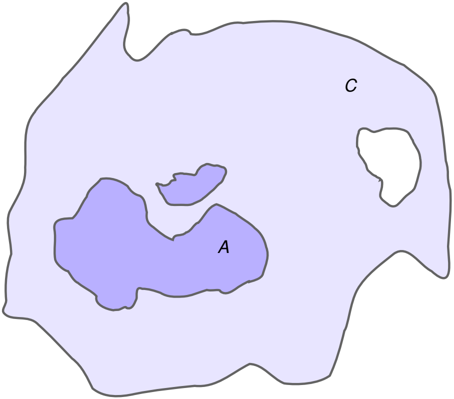
Theorem (Craig, 1957)

Suppose $A \rightarrow C$ is a valid FOL implication.
Then there is a formula I (an interpolant) such that

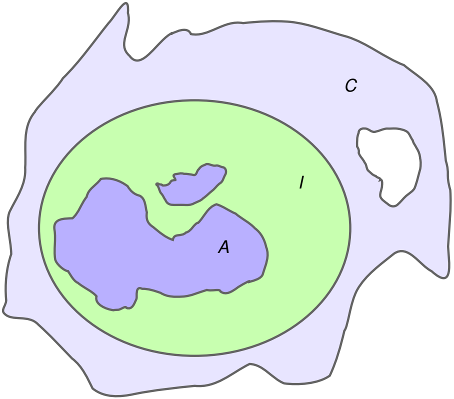
- *$A \rightarrow I$ and $I \rightarrow C$ are valid,*
- *every non-logical symbol of I occurs in both A and C.*

Illustration

Interpolation problem: $A \rightarrow I \rightarrow C$



Interpolation problem: $A \to I \to C$



Reverse interpolants

Definition

Suppose $A \wedge B$ is unsatisfiable.
A reverse interpolant is a formula I such that

- $A \rightarrow I$ and $B \rightarrow \neg I$ are valid,
- every non-logical symbol of I occurs in both A and B.

Lemma

I is reverse interpolant for $A \wedge B$

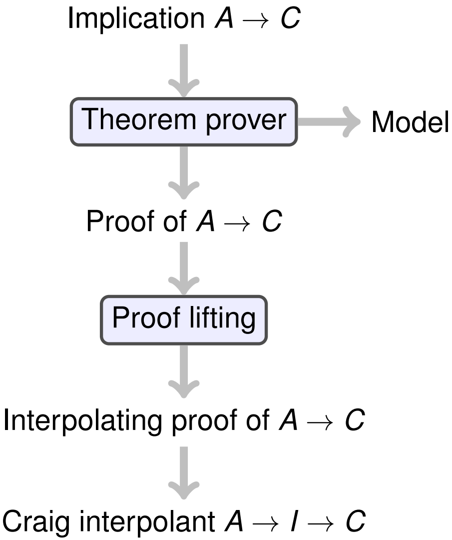
$$\Longleftrightarrow$$

I is interpolant for $A \rightarrow \neg B$

Available interpolation engines (incomplete . . .)

- Foci
- CSIsat
- MathSAT
- SMTInterpol
- OpenSMT
- iZ3
- Princess

Proof-based interpolation techniques



Interpolating propositional logic

- Interpolation procedures available for many calculi
- Overview paper for resolution proofs: [D'Silva et al, 2010]

- Shown here: interpolants from a Gentzen-style proof (similar to calculus from before, but without constraints)

Basic idea of proof lifting in a sequent calculus

Interpolation problem: $A \to I \to C$

$$\cfrac{\cfrac{\cfrac{*}{\vdots}}{\Gamma_3 \vdash \Delta_3}}{\cfrac{\Gamma_2 \vdash \Delta_2}{\cfrac{\Gamma_1 \vdash \Delta_1}{\cfrac{\vdots}{A \vdash C}}}}$$

Basic idea of proof lifting in a sequent calculus

Interpolation problem: $A \to I \to C$

annotation of formulae with labels $\uparrow$

$$
\cfrac{\cfrac{\begin{array}{c} * \\ \vdots \\ \Gamma_3 \vdash \Delta_3 \end{array}}{\Gamma_2 \vdash \Delta_2}}{\begin{array}{c} \Gamma_1 \vdash \Delta_1 \\ \vdots \\ A \vdash C \end{array}}
$$

Basic idea of proof lifting in a sequent calculus

Interpolation problem: $A \to I \to C$

annotation of formulae with labels $\uparrow$

$$\cfrac{\cfrac{\begin{matrix} * \\ \vdots \\ \Gamma_3 \vdash \Delta_3 \end{matrix}}{\Gamma_2 \vdash \Delta_2}}{\begin{matrix} \Gamma_1 \vdash \Delta_1 \\ \vdots \\ \lfloor A \rfloor_L \vdash \lfloor C \rfloor_R \end{matrix}}$$

Basic idea of proof lifting in a sequent calculus

Interpolation problem: $A \to I \to C$

annotation of formulae with labels $\uparrow$

$$
\begin{array}{ccc}
 & * & \\
 & \vdots & \\
\Gamma_3 & \vdash & \Delta_3 \\
\hline
\Gamma_2 & \vdash & \Delta_2 \\
\hline
\Gamma_1^* & \vdash & \Delta_1^* \\
 & \vdots & \\
\lfloor A \rfloor_L & \vdash & \lfloor C \rfloor_R
\end{array}
$$

Basic idea of proof lifting in a sequent calculus

Interpolation problem: $A \rightarrow I \rightarrow C$

annotation of formulae with labels $\uparrow$

$$
\begin{array}{c}
* \\
\vdots \\
\Gamma_3 \ \vdash \ \Delta_3 \\
\hline
\Gamma_2^* \ \vdash \ \Delta_2^* \\
\hline
\Gamma_1^* \ \vdash \ \Delta_1^* \\
\vdots \\
\lfloor A \rfloor_L \ \vdash \ \lfloor C \rfloor_R
\end{array}
$$

Basic idea of proof lifting in a sequent calculus

Interpolation problem: $A \to I \to C$

annotation of formulae with labels $\uparrow$

$$
\begin{array}{c}
* \\
\vdots \\
\Gamma_3^* \;\vdash\; \Delta_3^* \\
\hline
\Gamma_2^* \;\vdash\; \Delta_2^* \\
\hline
\Gamma_1^* \;\vdash\; \Delta_1^* \\
\vdots \\
\lfloor A \rfloor_L \;\vdash\; \lfloor C \rfloor_R
\end{array}
$$

Basic idea of proof lifting in a sequent calculus

Interpolation problem: $A \to I \to C$

annotation of formulae with labels $\uparrow$

$$
\dfrac{\dfrac{\dfrac{\vdots^{*}}{\Gamma_3^* \vdash \Delta_3^*}}{\Gamma_2^* \vdash \Delta_2^*}}{\Gamma_1^* \vdash \Delta_1^*}
$$

$$\vdots$$

$$\lfloor A \rfloor_L \vdash \lfloor C \rfloor_R$$

$\downarrow$ propagation of interpolants

Basic idea of proof lifting in a sequent calculus

Interpolation problem: $A \to I \to C$

annotation of formulae with labels $\uparrow$

$$
\begin{array}{c}
*\\
\vdots\\
\dfrac{\Gamma_3^* \vdash \Delta_3^* \blacktriangleright I_3}{\dfrac{\Gamma_2^* \vdash \Delta_2^*}{\Gamma_1^* \vdash \Delta_1^*}}\\
\vdots\\
\lfloor A \rfloor_L \vdash \lfloor C \rfloor_R
\end{array}
$$

$\downarrow$ propagation of interpolants

Basic idea of proof lifting in a sequent calculus

Interpolation problem: $A \to I \to C$

annotation of formulae with labels $\uparrow$

$$\dfrac{\dfrac{\begin{array}{c}*\\ \vdots \\ \Gamma_3^* \vdash \Delta_3^* \blacktriangleright I_3\end{array}}{\Gamma_2^* \vdash \Delta_2^* \blacktriangleright I_2}}{\begin{array}{c}\Gamma_1^* \vdash \Delta_1^* \\ \vdots \\ \lfloor A \rfloor_L \vdash \lfloor C \rfloor_R\end{array}}$$

$\downarrow$ propagation of interpolants

Basic idea of proof lifting in a sequent calculus

Interpolation problem: $A \rightarrow I \rightarrow C$

annotation of formulae with labels $\uparrow$

$$
\cfrac{\cfrac{\begin{array}{c} * \\ \vdots \\ \Gamma_3^* \vdash \Delta_3^* \blacktriangleright I_3 \end{array}}{\Gamma_2^* \vdash \Delta_2^* \blacktriangleright I_2}}{\begin{array}{c} \Gamma_1^* \vdash \Delta_1^* \blacktriangleright I_1 \\ \vdots \\ \lfloor A \rfloor_L \vdash \lfloor C \rfloor_R \end{array}}
$$

$\downarrow$ propagation of interpolants

Basic idea of proof lifting in a sequent calculus

Interpolation problem: $A \to I \to C$

$$
\dfrac{\dfrac{\dfrac{*}{\vdots}}{\Gamma_3^* \vdash \Delta_3^* \blacktriangleright I_3}}{\dfrac{\Gamma_2^* \vdash \Delta_2^* \blacktriangleright I_2}{\dfrac{\Gamma_1^* \vdash \Delta_1^* \blacktriangleright I_1}{\vdots \atop \lfloor A \rfloor_L \vdash \lfloor C \rfloor_R \blacktriangleright I}}}
$$

annotation of formulae with labels $\uparrow$

$\downarrow$ propagation of interpolants

Labelled formulae

Interpolation problem: $A \rightarrow I \rightarrow C$

Labelled formula	**Intuition**
$\lfloor \phi \rfloor_L$	"ϕ is subformula of A"
$\lfloor \phi \rfloor_R$	"ϕ is subformula of C"

Example

Non-interpolating proof:

$$\dfrac{\dfrac{\dfrac{*}{p \vdash p, q, r}}{\neg p, p \vdash q, r} \qquad \dfrac{*}{q, p \vdash q, r}}{\dfrac{\neg p \lor q, p \vdash q, r}{\neg p \lor q, p \vdash q \lor r}}$$

Example

Non-interpolating proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{*}{p \vdash p, q, r}}{\neg p, p \vdash q, r} \qquad \dfrac{*}{q, p \vdash q, r}}{\neg p \lor q, p \vdash q, r}}{\neg p \lor q, p \vdash q \lor r}$$

Lifted interpolating proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{*}{\lfloor p \rfloor_L \vdash \lfloor p \rfloor_L}}{\lfloor \neg p \rfloor_L, \lfloor p \rfloor_L \vdash \ldots} \qquad \dfrac{*}{\lfloor q \rfloor_L, \lfloor p \rfloor_L \vdash \lfloor q \rfloor_R, \lfloor r \rfloor_R}}{\lfloor \neg p \lor q \rfloor_L, \lfloor p \rfloor_L \vdash \lfloor q \rfloor_R, \lfloor r \rfloor_R}}{\lfloor \neg p \lor q \rfloor_L, \lfloor p \rfloor_L \vdash \lfloor q \lor r \rfloor_R}$$

Example

Non-interpolating proof:

$$\dfrac{\dfrac{\dfrac{*}{p \vdash p, q, r}}{\neg p, p \vdash q, r} \qquad \dfrac{*}{q, p \vdash q, r}}{\dfrac{\neg p \vee q, p \vdash q, r}{\neg p \vee q, p \vdash q \vee r}}$$

Lifted interpolating proof:

$$\dfrac{\dfrac{\dfrac{*}{\lfloor p\rfloor_L \vdash \lfloor p\rfloor_L \blacktriangleright \mathit{false}}}{\lfloor\neg p\rfloor_L, \lfloor p\rfloor_L \vdash \ldots \blacktriangleright \mathit{false}} \qquad \dfrac{*}{\lfloor q\rfloor_L,\lfloor p\rfloor_L \vdash \lfloor q\rfloor_R, \lfloor r\rfloor_R \blacktriangleright q}}{\dfrac{\lfloor \neg p\vee q\rfloor_L, \lfloor p\rfloor_L \vdash \lfloor q\rfloor_R,\lfloor r\rfloor_R \blacktriangleright \mathit{false}\vee q}{\lfloor \neg p\vee q\rfloor_L,\lfloor p\rfloor_L \vdash \lfloor q\vee r\rfloor_R \blacktriangleright q}}$$

Interpolating propositional rules

$$\frac{\Gamma, \lfloor\phi\rfloor_L \vdash \Delta \blacktriangleright I \qquad \Gamma, \lfloor\psi\rfloor_L \vdash \Delta \blacktriangleright J}{\Gamma, \lfloor\phi \vee \psi\rfloor_L \vdash \Delta \blacktriangleright I \vee J} \text{ OR-LEFT-L}$$

$$\frac{\Gamma, \lfloor\phi\rfloor_R \vdash \Delta \blacktriangleright I \qquad \Gamma, \lfloor\psi\rfloor_R \vdash \Delta \blacktriangleright J}{\Gamma, \lfloor\phi \vee \psi\rfloor_R \vdash \Delta \blacktriangleright I \wedge J} \text{ OR-LEFT-R}$$

$$\frac{\Gamma, \lfloor\phi\rfloor_D, \lfloor\psi\rfloor_D \vdash \Delta \blacktriangleright I}{\Gamma, \lfloor\phi \wedge \psi\rfloor_D \vdash \Delta \blacktriangleright I} \text{ AND-LEFT}$$

$$\frac{\Gamma \vdash \lfloor\phi\rfloor_D, \Delta \blacktriangleright I}{\Gamma, \lfloor\neg\phi\rfloor_D \vdash \Delta \blacktriangleright I} \text{ NOT-LEFT}$$

$$\frac{*}{\Gamma, \lfloor\phi\rfloor_L \vdash \lfloor\phi\rfloor_L, \Delta \blacktriangleright \mathit{false}} \text{ CLOSE-LL}$$

$$\frac{*}{\Gamma, \lfloor\phi\rfloor_R \vdash \lfloor\phi\rfloor_R, \Delta \blacktriangleright \mathit{true}} \text{ CLOSE-RR}$$

$$\frac{*}{\Gamma, \lfloor\phi\rfloor_L \vdash \lfloor\phi\rfloor_R, \Delta \blacktriangleright \phi} \text{ CLOSE-LR}$$

$$\frac{*}{\Gamma, \lfloor\phi\rfloor_R \vdash \lfloor\phi\rfloor_L, \Delta \blacktriangleright \neg\phi} \text{ CLOSE-RL}$$

$$\frac{\Gamma, \lfloor[x/t]\phi\rfloor_L, \lfloor\forall x.\phi\rfloor_L \vdash \Delta \blacktriangleright I}{\Gamma, \lfloor\forall x.\phi\rfloor_L \vdash \Delta \blacktriangleright \forall_{Rt}\, I} \text{ ALL-LEFT-L}$$

$$\frac{\Gamma, \lfloor[x/t]\phi\rfloor_R, \lfloor\forall x.\phi\rfloor_R \vdash \Delta \blacktriangleright I}{\Gamma, \lfloor\forall x.\phi\rfloor_R \vdash \Delta \blacktriangleright \exists_{Lt}\, I} \text{ ALL-LEFT-R}$$

$$\frac{\Gamma, \lfloor[x/c]\phi\rfloor_D \vdash \Delta \blacktriangleright I}{\Gamma, \lfloor\exists x.\phi\rfloor_D \vdash \Delta \blacktriangleright I} \text{ EX-LEFT}$$

$$\frac{\Gamma \vdash \lfloor[x/c]\phi\rfloor_D, \Delta \blacktriangleright I}{\Gamma \vdash \lfloor\forall x.\phi\rfloor_D, \Delta \blacktriangleright I} \text{ ALL-RIGHT}$$

Interpolating integer arithmetic . . .

Some theory rules for integers

Linear combination of inequalities ($\alpha > 0, \beta > 0$)

$$\frac{\Gamma,\ \ldots, \alpha s + \beta t \dot{\leq} 0 \ \vdash\ \Delta}{\Gamma,\ s \dot{\leq} 0,\ t \dot{\leq} 0 \ \vdash\ \Delta}\ \text{FM-ELIM}'$$

Strengthening inequalities (subsumes rounding, Gomory cuts)

$$\frac{\Gamma, t \doteq 0 \ \vdash\ \Delta \qquad \Gamma, t + 1 \dot{\leq} 0 \ \vdash\ \Delta}{\Gamma, t \dot{\leq} 0 \ \vdash\ \Delta}\ \text{STRENGTHEN}'$$

Some theory rules for integers

Linear combination of inequalities ($\alpha > 0, \beta > 0$)

$$\frac{\Gamma,\ \ldots, \alpha s + \beta t \dot{\leq} 0 \ \vdash\ \Delta}{\Gamma,\ s \dot{\leq} 0,\ t \dot{\leq} 0 \ \vdash\ \Delta}\ \text{FM-ELIM}'$$

Strengthening inequalities (subsumes rounding, Gomory cuts)

$$\frac{\Gamma, t \doteq 0 \ \vdash\ \Delta \qquad \Gamma, t + 1 \dot{\leq} 0 \ \vdash\ \Delta}{\Gamma, t \dot{\leq} 0 \ \vdash\ \Delta}\ \text{STRENGTHEN}'$$

- Calculus contains both analytic and synthetic rules
  $\Rightarrow$ More general form of labels needed

Extended labelled formulae

Interpolation problem: $A \to I \to C$

Labelled formula	Intuition
$\lfloor \phi \rfloor_L$	"ϕ is subformula of A"
$\lfloor \phi \rfloor_R$	"ϕ is subformula of C"
$\phi\,[\psi]$	"ψ is A-contribution to ϕ" ψ is the *partial interpolant* of ϕ

Selection of interpolating integer rules

Linear combination of inequalities ($\alpha > 0, \beta > 0$)

$$\frac{\Gamma, \ldots, \alpha s + \beta t \mathrel{\dot{\leq}} 0\,[\alpha s^A + \beta t^A \mathrel{\dot{\leq}} 0] \;\vdash\; \Delta \;\blacktriangleright I}{\Gamma, s \mathrel{\dot{\leq}} 0\,[s^A \mathrel{\dot{\leq}} 0], t \mathrel{\dot{\leq}} 0\,[t^A \mathrel{\dot{\leq}} 0] \;\vdash\; \Delta \;\blacktriangleright I}\ \text{FM-ELIM}$$

Closure rules

$$\frac{*}{\Gamma, \alpha \mathrel{\dot{\leq}} 0\,[t^A \mathrel{\dot{\leq}} 0] \;\vdash\; \Delta \;\blacktriangleright t^A \mathrel{\dot{\leq}} 0}\ \text{CLOSE-INEQ}$$

Interpolating proof example

$$
\dfrac{\dfrac{\dfrac{\dfrac{*}{\ldots,\ 3 \mathrel{\dot{\le}} 0\,[6x \mathrel{\dot{\le}} 0] \ \vdash\ \blacktriangleright\, x \mathrel{\dot{\le}} 0}}{\ldots,\ 3x \mathrel{\dot{\le}} 0\,[3x \mathrel{\dot{\le}} 0],\ -2x+1 \mathrel{\dot{\le}} 0\,[0 \mathrel{\dot{\le}} 0] \ \vdash\ \blacktriangleright\, x \mathrel{\dot{\le}} 0}}{\ldots,\ 3x-2 \mathrel{\dot{\le}} 0\,[3x-2 \mathrel{\dot{\le}} 0],\ -2x+1 \mathrel{\dot{\le}} 0\,[0 \mathrel{\dot{\le}} 0] \ \vdash\ \blacktriangleright\, x \mathrel{\dot{\le}} 0}}{\begin{array}{c} a+x \mathrel{\dot{\le}} 0\,[a+x \mathrel{\dot{\le}} 0], \\ -a+2x-2 \mathrel{\dot{\le}} 0\,[-a+2x-2 \mathrel{\dot{\le}} 0], \quad \vdash \quad \blacktriangleright\, x \mathrel{\dot{\le}} 0 \\ -2x+1 \mathrel{\dot{\le}} 0\,[0 \mathrel{\dot{\le}} 0] \end{array}}
$$

Original proof

$$
\dfrac{\dfrac{\dfrac{\dfrac{*}{\ldots,\ 3 \mathrel{\dot{\le}} 0 \ \vdash}\ \text{INEQ-CLOSE}'}{\ldots,\ 3x \mathrel{\dot{\le}} 0,\ -2x+1 \mathrel{\dot{\le}} 0 \ \vdash}\ \text{FM-ELIM}' \qquad \ldots}{\ldots,\ 3x-2 \mathrel{\dot{\le}} 0,\ -2x+1 \mathrel{\dot{\le}} 0 \ \vdash}\ \text{STRENGTHEN}' \times 2}{a+x \mathrel{\dot{\le}} 0,\ -a+2x-2 \mathrel{\dot{\le}} 0,\ -2x+1 \mathrel{\dot{\le}} 0 \ \vdash}\ \text{FM-ELIM}'
$$

Literature

- Difference logic
  [McMillan, 2006]
- Integer equalities + divisibility constraints
  [Jain, Clarke, Grumberg, 2008]
- Unit-two-variable-per-inequality
  [Cimatti, Griggio, Sebastiani, 2009]

- Simplex-based, full PA
  [Lynch, Tang, 2008]
  $\Rightarrow$ Leaves local symbols/quantifiers in interpolants

Literature (2)

Proof-based methods for full PA:

- Sequent calculus-based
  [Brillout, Kroening, Rümmer, Wahl, 2010]
- Simplex-based, special branch-and-cut rule
  [Kroening, Leroux, Rümmer, 2010]
- Simplex-based, targeting SMT
  [Griggio, Le, Sebastiani, 2011]
- From Z3 proofs
  [McMillan, 2011]

Conclusion

- Interpolation engines are today available for many logics/theories
- Not quite as mature yet as SMT in general

Remaining challenges:

- mixed-integer, bit-vectors, full first-order logic, quantifier-free arrays, ...
- exploration of the interpolant space

Thanks for your attention!