Craig Interpolation for Integer Arithmetic, Uninterpreted Functions, and the Theory of Arrays

Angelo Brillout[1] Daniel Kroening[2] Jérôme Leroux[3]
Philipp Rümmer[4] Thomas Wahl[2]

[1]ETH Zurich

[2]University of Oxford

[3]Laboratoire Bordelais de Recherche en Informatique

[4]Uppsala University

SVARM, April 2nd, 2011

Motivation: inference of invariants

Generic verification problem (“safety”)

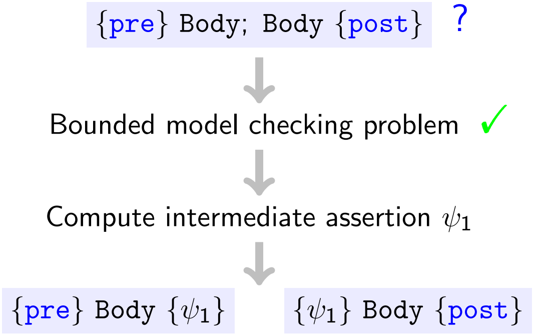
$$\{\ \texttt{pre}\ \}\ \texttt{while (*) Body}\ \{\ \texttt{post}\ \}$$

Standard approach: loop rule using invariant

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

How to compute ϕ automatically?

From intermediate assertions to invariants



{pre} Body; Body {post} ?

Bounded model checking problem ✓

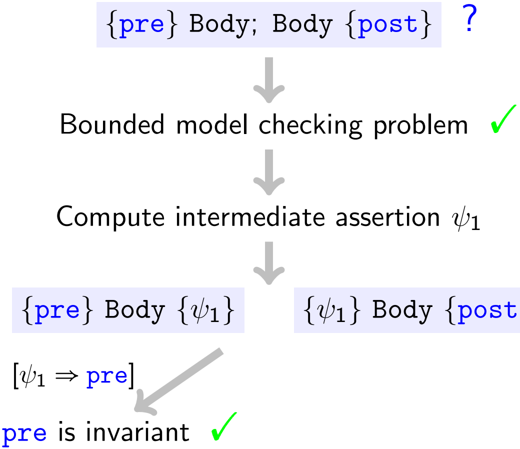
Compute intermediate assertion ψ_1

{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}

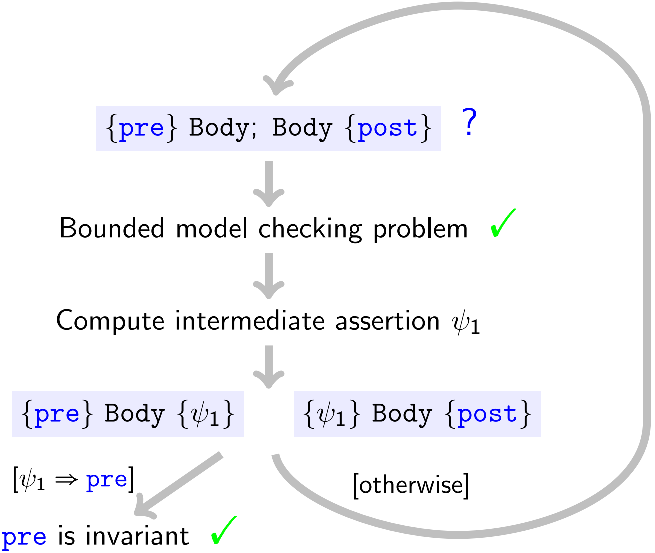
{ψ_1} `Body` {`post`}

[$\psi_1 \Rightarrow$ `pre`]

`pre` is invariant ✓

[McMillan, 2003]

From intermediate assertions to invariants



$\{\mathtt{pre}\}\ \mathtt{Body;\ Body}\ \{\mathtt{post}\}$?

Bounded model checking problem ✓

Compute intermediate assertion ψ_1

$\{\mathtt{pre}\}\ \mathtt{Body}\ \{\psi_1\}$

$\{\psi_1\}\ \mathtt{Body}\ \{\mathtt{post}\}$

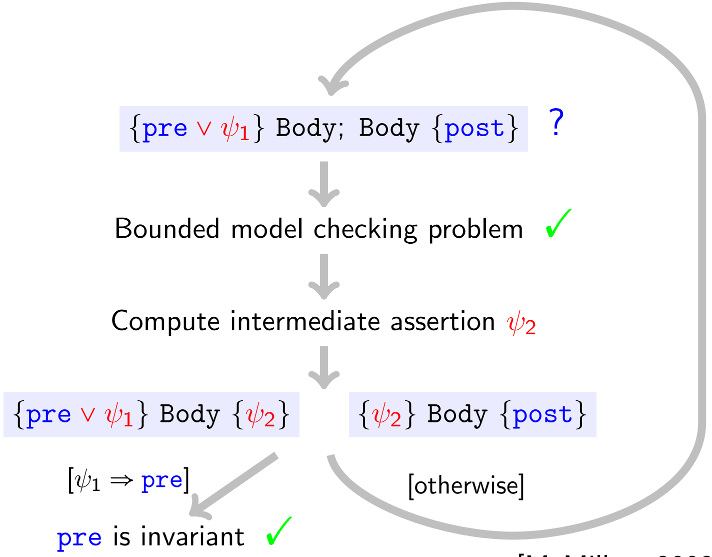
$[\psi_1 \Rightarrow \mathtt{pre}]$

pre is invariant ✓

[otherwise]

[McMillan, 2003]

From intermediate assertions to invariants



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

Bounded model checking problem ✓

Compute intermediate assertion ψ_2

$\{\texttt{pre} \vee \psi_1\}$ Body $\{\psi_2\}$

$\{\psi_2\}$ Body $\{\texttt{post}\}$

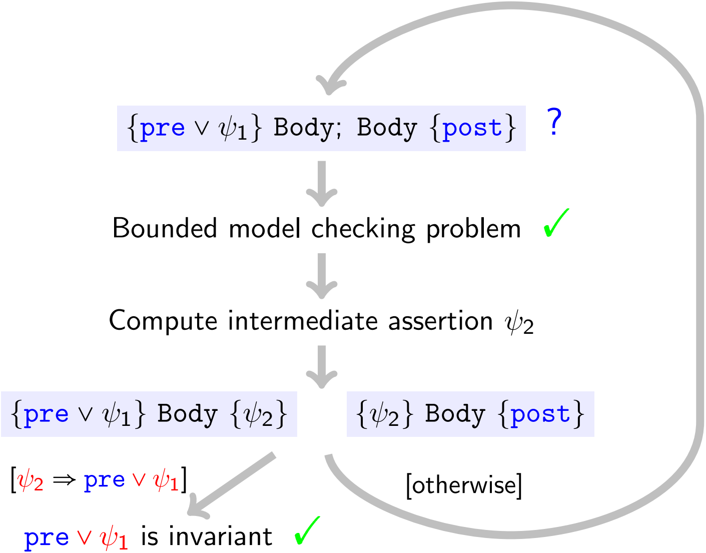
$[\psi_1 \Rightarrow \texttt{pre}]$

[otherwise]

$\texttt{pre}$ is invariant ✓

[McMillan, 2003]

From intermediate assertions to invariants



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

Bounded model checking problem ✓

Compute intermediate assertion ψ_2

$\{\texttt{pre} \vee \psi_1\}\ \texttt{Body}\ \{\psi_2\}$

$\{\psi_2\}\ \texttt{Body}\ \{\texttt{post}\}$

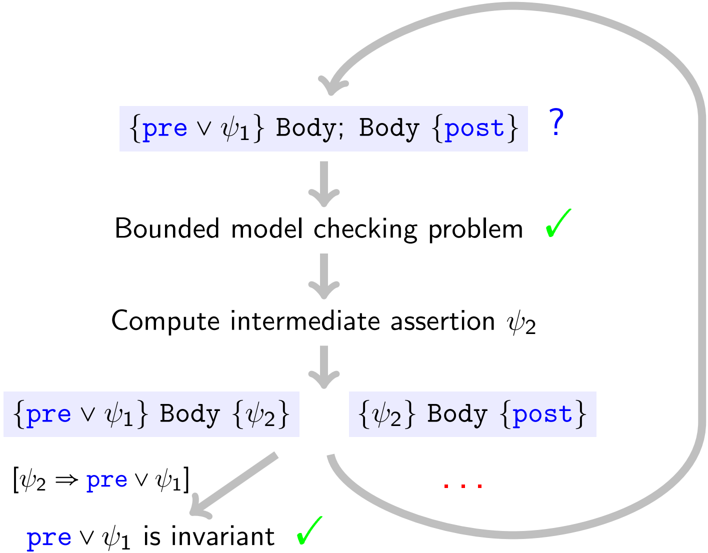
$[\psi_2 \Rightarrow \texttt{pre} \vee \psi_1]$

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

[otherwise]

[McMillan, 2003]

From intermediate assertions to invariants



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

Bounded model checking problem ✓

Compute intermediate assertion ψ_2

$\{\texttt{pre} \vee \psi_1\}\ \texttt{Body}\ \{\psi_2\}$

$\{\psi_2\}\ \texttt{Body}\ \{\texttt{post}\}$

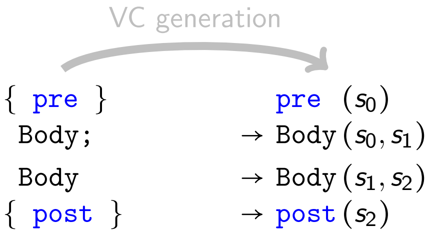
$[\psi_2 \Rightarrow \texttt{pre} \vee \psi_1]$

...

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

[McMillan, 2003]

How to compute intermediate assertions?

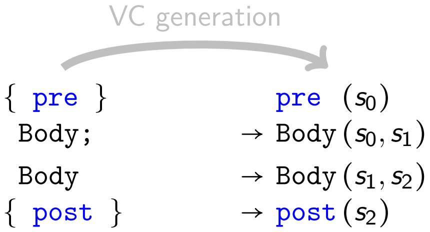


VC generation

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

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

How to compute intermediate assertions?



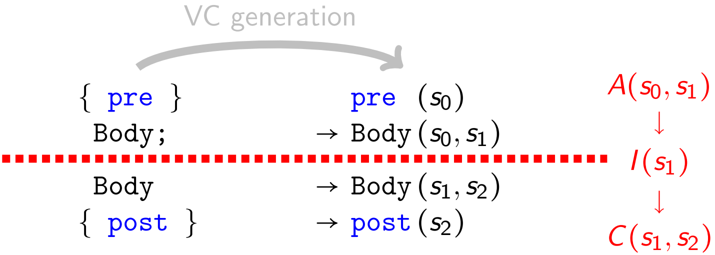
{ pre }	pre (s_0)
Body;	$\rightarrow$ Body (s_0, s_1)
Body	$\rightarrow$ Body (s_1, s_2)
{ post }	$\rightarrow$ post (s_2)

Theorem (Craig, 1957)

Suppose $A \Rightarrow C$ is a valid 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 } $\quad \text{pre}\,(s_0)$ $\quad A(s_0, s_1)$

Body; $\quad \to \text{Body}\,(s_0, s_1)$ $\quad \downarrow$

$I(s_1)$

Body $\quad \to \text{Body}\,(s_1, s_2)$ $\quad \downarrow$

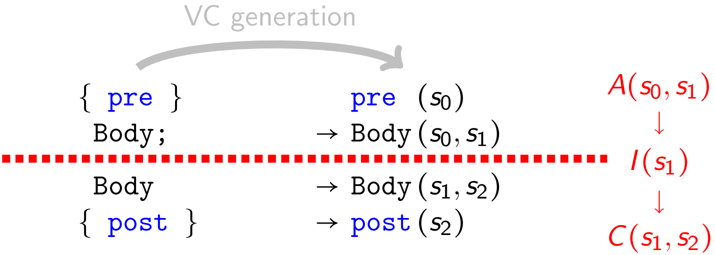
{ post } $\quad \to \text{post}\,(s_2)$ $\quad C(s_1, s_2)$

Theorem (Craig, 1957)

Suppose $A \Rightarrow C$ is a valid 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 } $\quad \mathtt{pre}\,(s_0) \quad A(s_0, s_1)$

Body; $\quad \rightarrow \mathtt{Body}\,(s_0, s_1)$ $\quad \downarrow$

$I(s_1)$

Body $\quad \rightarrow \mathtt{Body}\,(s_1, s_2)$ $\quad \downarrow$

{ post } $\quad \rightarrow \mathtt{post}\,(s_2) \quad C(s_1, s_2)$

Theorem (Craig, 1957)

Suppose $A \Rightarrow C$ is a valid 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.*

Interpolant I can be computed from proofs of $A \Rightarrow C$

Interpolation + theories

Interpolation procedures need to support the program logic:

```
int a[],i;
max = a[0];
for (i = 1; i < n; ++i)
  if (a[i] > max)
    max = a[i];
assert (max >= a[i/2]);
```

E.g., combined use of linear integer arithmetic and arrays

Theories investigated by us

- Quantifier-free Presburger Arithmetic (PA) [IJCAR, 2010]
  (linear integer arithmetic) [LPAR, 2010]

$+$

- Quantifiers (Q) [VERIFY, 2010]
- Uninterpreted predicates (UP) [VMCAI, 2011]
- Uninterpreted functions (UF)
- Arrays (AR)

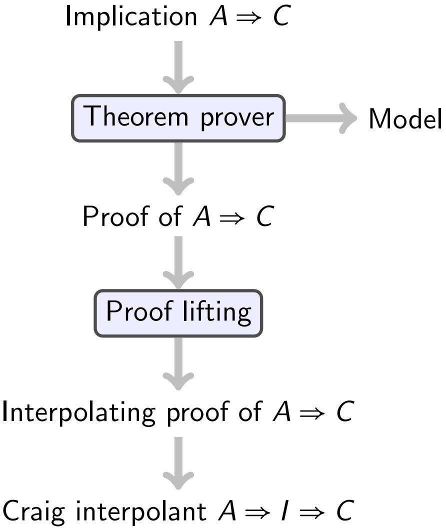
Theories investigated by us

- Quantifier-free Presburger Arithmetic (PA) [IJCAR, 2010]
  (linear integer arithmetic) [LPAR, 2010]

$+$

- Quantifiers (Q) [VERIFY, 2010]
- Uninterpreted predicates (UP) [VMCAI, 2011]
- Uninterpreted functions (UF)
- Arrays (AR)

Interpolation outline



Implication $A \Rightarrow C$

Theorem prover

Model

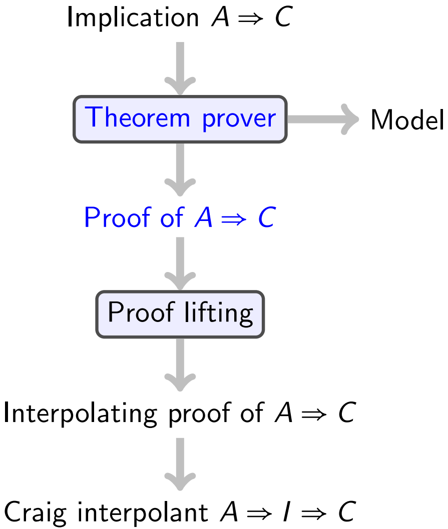
Proof of $A \Rightarrow C$

Proof lifting

Interpolating proof of $A \Rightarrow C$

Craig interpolant $A \Rightarrow I \Rightarrow C$

Interpolation outline



Implication $A \Rightarrow C$

Theorem prover

Model

Proof of $A \Rightarrow C$

Proof lifting

Interpolating proof of $A \Rightarrow C$

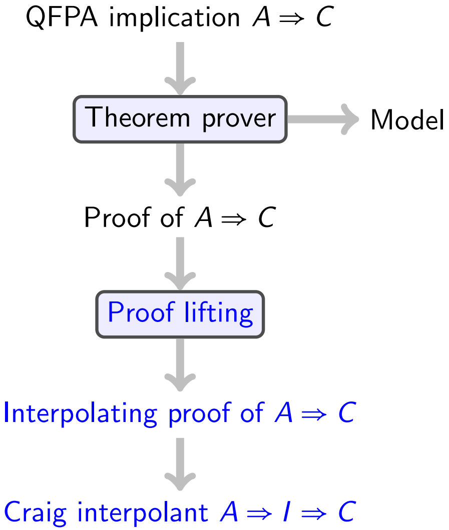
Craig interpolant $A \Rightarrow I \Rightarrow C$

Underlying calculus for Presburger Arithmetic

- Gentzen-style sequent calculus for PA [LPAR, 2008]

	Calculus rules	**Possible procedures**
Equalities	Linear combination, fresh constants	Omega eq. elimination, Smith decomposition
Inequalities	Linear combination, rounding, ineq. splitting	Omega test, Simplex + Gomory cuts + branch-and-bound
Prop. logic	Standard Gentzen propositional rules	

Interpolation outline



QFPA implication $A \Rightarrow C$

Theorem prover

Model

Proof of $A \Rightarrow C$

Proof lifting

Interpolating proof of $A \Rightarrow C$

Craig interpolant $A \Rightarrow I \Rightarrow C$

Basic idea of proof lifting

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

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

Basic idea of proof lifting

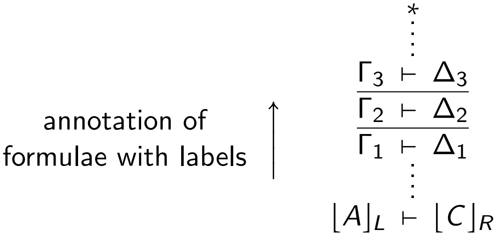
Interpolation problem: $A \Rightarrow I \Rightarrow C$

annotation of formulae with labels ↑

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

Basic idea of proof lifting

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

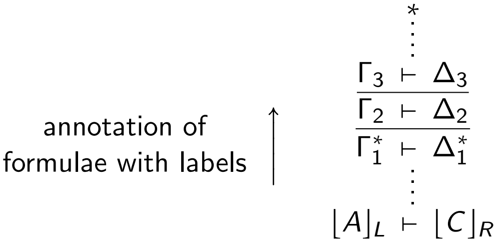


annotation of formulae with labels ↑

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

Basic idea of proof lifting

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

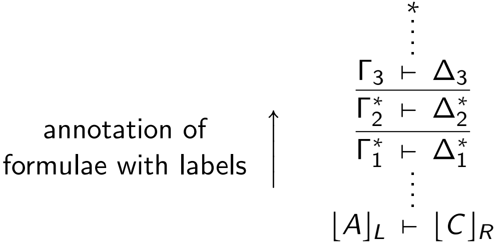


annotation of formulae with labels

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

Basic idea of proof lifting

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

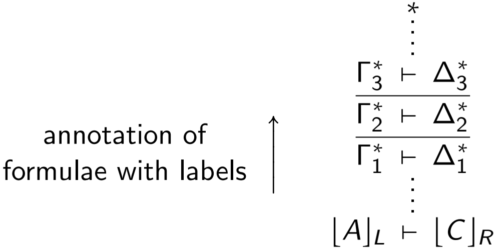


annotation of formulae with labels

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

Basic idea of proof lifting

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



annotation of formulae with labels

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

$$\vdots$$

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

Basic idea of proof lifting

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

$$
\begin{array}{ccccc}
 & & * & & \\
 & & \vdots & & \\
 & & \Gamma_3^* \vdash \Delta_3^* & & \\
 & & \hline & & \\
\text{annotation of} & \Big\uparrow & \Gamma_2^* \vdash \Delta_2^* & \Big\downarrow & \text{propagation of} \\
\text{formulae with labels} & & \hline & & \text{interpolants} \\
 & & \Gamma_1^* \vdash \Delta_1^* & & \\
 & & \vdots & & \\
 & & \lfloor A \rfloor_L \vdash \lfloor C \rfloor_R & &
\end{array}
$$

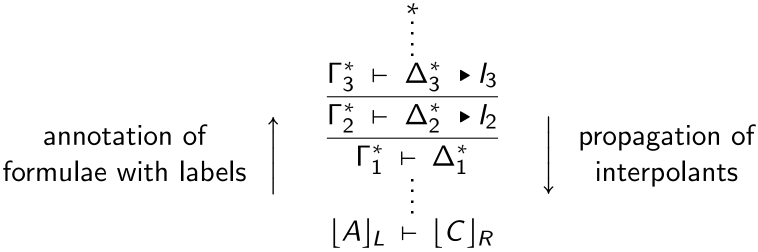
Basic idea of proof lifting

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

$$\begin{array}{ccccc} & & * & & \\ & & \vdots & & \\ \text{annotation of} & & \dfrac{\Gamma_3^* \vdash \Delta_3^* \;\blacktriangleright I_3}{\dfrac{\Gamma_2^* \vdash \Delta_2^*}{\Gamma_1^* \vdash \Delta_1^*}} & & \text{propagation of} \\ \text{formulae with labels} & \Big\uparrow & & \Big\downarrow & \text{interpolants} \\ & & \vdots & & \\ & & \lfloor A \rfloor_L \vdash \lfloor C \rfloor_R & & \end{array}$$

Basic idea of proof lifting

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



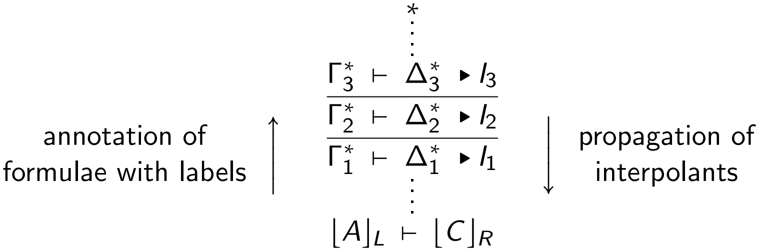
annotation of formulae with labels ↑

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

↓ propagation of interpolants

Basic idea of proof lifting

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



annotation of formulae with labels ↑

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

↓ propagation of interpolants

Basic idea of proof lifting

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

$$\begin{array}{ccccc} & & * & & \\ & & \vdots & & \\ \begin{array}{c}\text{annotation of}\\ \text{formulae with labels}\end{array} & \Big\uparrow & \dfrac{\dfrac{\Gamma_3^* \vdash \Delta_3^* \blacktriangleright I_3}{\Gamma_2^* \vdash \Delta_2^* \blacktriangleright I_2}}{\Gamma_1^* \vdash \Delta_1^* \blacktriangleright I_1} & \Big\downarrow & \begin{array}{c}\text{propagation of}\\ \text{interpolants}\end{array} \\ & & \vdots & & \\ & & \lfloor A \rfloor_L \vdash \lfloor C \rfloor_R \blacktriangleright I & & \end{array}$$

Properties of the interpolating calculus

Lemma (Soundness)

The annotation at the root of a closed proof is a valid interpolant.

Lemma (Completeness)

Every proof can be lifted to an interpolating proof.
This implies: completeness for PA.

Generality

Applicable to various procedures:

- Simplex + cuts (cf. [Griggio, Le, Sebastiani, 2011])
- Omega test

Properties of the interpolating calculus

Lemma (Soundness)

The annotation at the root of a closed proof is a valid interpolant.

Lemma (Completeness)

Every proof can be lifted to an interpolating proof.
This implies: completeness for PA.

Generality

Applicable to various procedures:

- Simplex + cuts (cf. [Griggio, Le, Sebastiani, 2011])
- Omega test

Can be generalised to further theories . . .

Beyond Presburger Arithmetic

- Quantifier-free Presburger Arithmetic (PA) [IJCAR, 2010]
  (linear integer arithmetic) [LPAR, 2010]

$+$

- Quantifiers (Q) [VERIFY, 2010]
- Uninterpreted predicates (UP) [VMCAI, 2011]
- Uninterpreted functions (UF)
- Arrays (AR)

Fragments of extensions of Presburger Arithmetic

Considered logics:

- PA+UP, PA+UF: PA + unint. predicates/functions
- QPA+UP, QPA+UF: PA + quantifiers + $\cdots$
- PA+AR: PA + *select*, *store* functions

$$\begin{aligned}
\phi &::= t = t \;\Big|\; t \leq t \;\Big|\; \alpha \mid t \;\Big|\; p(\bar{t}) \;\Big|\; \phi \wedge \phi \;\Big|\; \phi \vee \phi \;\Big|\; \neg\phi \;\Big|\; \forall x.\phi \;\Big|\; \exists x.\phi \\
t &::= \alpha \;\Big|\; c \;\Big|\; x \;\Big|\; \alpha t + \cdots + \alpha t \;\Big|\; f(\bar{t})
\end{aligned}$$

Interesting questions

- Closure under interpolation
- Practical interpolation procedures

Definition

Logic L is **closed under interpolation** if
for all $A, B \in F$ such that $A \Rightarrow B$, there is an interpolant expressible in L.

[Kapur et al, 2006: "L is **interpolating**"]

Known results

(Q)PA ⇒ closed under interpolation
(as it allows quantifier elimination)

PA+AR ⇒ not closed
(not even without PA, [Kapur et al, 2006])

QPA+AR ⇒ closed
(add quantifiers for local variables)

QPA+UP ⇒ not closed
QPA+UF (since interpolation could simulate
second-order quantifier elimination)

Known results

(Q)PA	⇒	closed under interpolation (as it allows quantifier elimination)
PA+AR	⇒	not closed (not even without PA, [Kapur et al, 2006])
QPA+AR	⇒	closed (add quantifiers for local variables)
QPA+UP QPA+UF	⇒	not closed (since interpolation could simulate second-order quantifier elimination)
PA+UP	⇒	?
PA+UF	⇒	?

New negative result

Theorem

PA+UP is **not** *closed under interpolation.*

(Similarly for PA+UF)

New negative result

Theorem

PA+UP is **not** *closed under interpolation.*

(Similarly for PA+UF)

Example

$$\phi \quad :: \quad (2c = y \wedge p(c)) \quad \Rightarrow \quad (2d = y \Rightarrow p(d))$$

Interpolants:
$$\boxed{\begin{array}{rll} \text{strongest:} & I_1 : & \exists c.\,(2c = y \wedge p(c)) \\ \text{weakest:} & I_2 : & \forall d.\,(2d = y \Rightarrow p(d)) \end{array}}$$

No quantifier-free interpolants exist!

Closure results

(Q)PA	⇒	closed under interpolation (as it allows quantifier elimination)
PA+AR	⇒	not closed (not even without PA, [Kapur et al, 2006])
QPA+AR	⇒	closed (add quantifiers for local variables)
QPA+UP QPA+UF	⇒	not closed (since interpolation could simulate second-order quantifier elimination)
PA+UP	⇒	not closed
PA+UF	⇒	not closed

Positive results

Lemma (interpolants with quantifiers)

If $A \Rightarrow B$ is a valid PA+UP formula, then there is a QPA+UP interpolant $A \Rightarrow I \Rightarrow B$.

(Similarly for PA+UF, PA+AR.)

Theorem (extension of PA+UP)

There is a (natural) extension of PA+UP that is

- *decidable, and*
- *closed under interpolation.*

(Similarly for PA+UF.)

How to close PA+UP under interpolation

Consider example:

$$\phi \quad :: \quad (2c = y \wedge p(c)) \quad \Rightarrow \quad (2d = y \Rightarrow p(d))$$

“Feels-like interpolant”: $p(\frac{y}{2})$

How to close PA+UP under interpolation

Consider example:

$$\phi \quad :: \quad (2c = y \wedge p(c)) \quad \Rightarrow \quad (2d = y \Rightarrow p(d))$$

"Feels-like interpolant": $p(\frac{y}{2})$

Definition

PAID+UP = PA+UP plus guarded quantification:

$$\exists x.(\alpha x = t \wedge \phi) \qquad \forall x.(\alpha x = t \Rightarrow \phi) \qquad (\alpha \neq 0,\ x \text{ not in } t)$$

How to close PA+UP under interpolation

Consider example:

$$\phi \quad :: \quad (2c = y \wedge p(c)) \quad \Rightarrow \quad (2d = y \Rightarrow p(d))$$

"Feels-like interpolant": $p(\frac{y}{2})$

Definition

PAID+UP = PA+UP plus guarded quantification:

$$\exists x.(\alpha x = t \wedge \phi) \qquad \forall x.(\alpha x = t \Rightarrow \phi) \qquad (\alpha \neq 0,\ x \text{ not in } t)$$

Is this just to accommodate ϕ's interpolant??

Interpolating in PAID+UP

Theorem

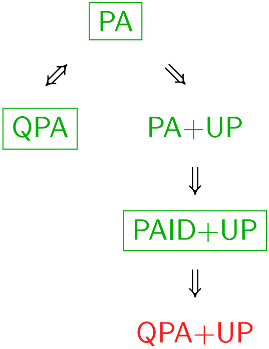
PAID+UP is closed under interpolation.

(Similarly for PAID+UF)

Proof:

1. Define a restricted version of our calculus that only generates PAID+UP interpolants
   - Only unify atoms $p(\bar{s}), p(\bar{t})$ or terms $f(\bar{s}), f(\bar{t})$ if $\bar{s} = \bar{t}$ has been derived
2. Show that the restricted calculus is still complete for PAID+UP

Summary of logics



PA

QPA

PA+UP

PAID+UP

QPA+UP

Legend:

decidable

undecidable

□ = closed under interpolation

⇓ = subset

What do we have?

- Sound + complete interpolating calculus for PAID+UP, PAID+UF, PAID+AR
- Generated interpolants stay within PAID+UP, PAID+UF, QPA+AR
- Calculus is close to procedures used in SMT solvers
- Combinations UP+UF+AR are straightforward

Future directions:

- Extensions of PAID+AR closed under interpolation? (+ decidable)
- Implementations
- Integration in Yorsh + Musuvathi's combination framework?

Related work: integer arithmetic interpolation

- Reduction to FOL
  [Kapur, Majumdar, Zarba, 2006]
- Simplex-based
  [Lynch, Tang, 2008]
- Sequent calculus-based
  [Brillout, Kroening, Rümmer, Wahl, 2010]
- Again Simplex-based
  [Kroening, Leroux, Rümmer, 2010]
- Simplex-based, targetting SMT
  [Griggio, Le, Sebastiani, 2011]

Related work: interpolation beyond integer arithmetic

- Uninterpreted functions
  [McMillan, 2005], [Fuchs, Goel, Grundy, Krstić, Tinelli, 2009]
- Theory of arrays
  [Kapur, Majumdar, Zarba, 2006], [McMillan, 2008]
- First-order logic
  [Hoder, Kovács, Voronkov, 2010]
- Quantifiers
  [Christ, Hoenicke, 2010]

- Combination of interpolation procedures
  [Yorsh, Musuvathi, 2005]

End of Talk.