Beyond Quantifier-Free Interpolation in Extensions of Presburger Arithmetic

Angelo Brillout,[1] Daniel Kroening,[2]
Philipp Rümmer,[3] Thomas Wahl[2]

[1]ETH Zurich

[2]Oxford University

[3]Uppsala University

Deduction at Scale, March 10th, 2011

Motivation: invariant inference

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

Goal: infer post conditions of loop executions

Instrument: inductive loop invariant ϕ

$$\frac{pre \Rightarrow \phi \qquad \{\phi\}\ body\ \{\phi\} \qquad \phi \Rightarrow post}{\{pre\}\ body^*\ \{post\}}$$

ϕ can be found using Craig interpolation + fixed-point iteration
[McMillan, 2003]

Interpolation: definition

Definition (Craig interpolant)

Given FO formulae A, B such that $A \Rightarrow B$ is valid, an **interpolant** is a formula I such that

1. $A \Rightarrow I$, $I \Rightarrow B$
2. $symb(I) \subseteq symb(A) \cap symb(B)$

Example: p is an interpolant for $p \wedge q \Rightarrow p \vee r$.

If A, B are FO formulas with $A \Rightarrow B$, there is an interpolant.
[W. Craig, 1957]

In this talk . . .

Starting point:

- Interpolation in quantifier-free Presburger Arithmetic (PA) [IJCAR, 2010]

We investigate combination with:

- Quantifiers
- Uninterpreted predicates (UP)
- Uninterpreted functions (UF)
- Theory of arrays (AR)

In this talk . . .

Starting point:

- Interpolation in quantifier-free Presburger Arithmetic (PA) [IJCAR, 2010]

We investigate combination with:

- Quantifiers
- Uninterpreted predicates (UP)
- Uninterpreted functions (UF)
- Theory of arrays (AR)

- Paper: [VMCAI, 2011]
- Earlier version: [VERIFY, 2010]

Interesting questions

- Decidability of validity
  - PA is decidable (also with quantifiers $\Rightarrow$ QPA)
  - PA+UP, PA+UF, PA+AR are decidable
  - QPA+* is undecidable
- Closure under interpolation
- Practical interpolation procedures

Fragments of extensions of Presburger Arithmetic

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

where

- ϕ is a formula
- t is a term
- α is an integer literal
- $c/p/f$ is a constant/UP/UF (uninterpreted)
- x is an integer variable

Fragments of extensions of Presburger Arithmetic

$$\phi ::= t = t \mid t \leq t \mid \alpha \mid t \mid p(\bar{t}) \mid \phi \wedge \phi \mid \phi \vee \phi \mid \neg\phi \mid \forall x.\phi \mid \exists x.\phi$$

$$t ::= \alpha \mid c \mid x \mid \alpha t + \cdots + \alpha t \mid f(\bar{t})$$

gives rise to various fragments/logics:

- PA: no UP, UF, quantifiers
- QPA: PA + quantifiers
- PA+UP, PA+UF
- QPA+UP, QPA+UF
- PA+AR: PA + distinguished functions *select*, *store*

Which fragments are closed under interpolation?

Definition

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

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

Known results

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

Is PA+UP closed under interpolation?

Is PA+UP closed under interpolation?

Consider example:

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

Is PA+UP closed under interpolation?

Consider example:

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

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

Is PA+UP closed under interpolation?

Consider example:

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

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

Really:

strongest:	I_1 :	$\exists c.\,(2c = y \wedge p(c))$
weakest:	I_2 :	$\forall d.\,(2d = y \Rightarrow p(d))$

Can we eliminate $\exists, \forall$ in the ϕ example?

Is PA+UP closed under interpolation?

Consider example:

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

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

Really:

strongest:	I_1 :	$\exists c.\,(2c = y \wedge p(c))$
weakest:	I_2 :	$\forall d.\,(2d = y \Rightarrow p(d))$

Can we eliminate $\exists, \forall$ in the ϕ example? No!

- I_1, I_2 cannot be expressed in PA+UP (i.e., without quantifiers)

New negative result

Theorem

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

(Similarly for PA+UF)

Closure results

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

Closure results

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

Anything positive?

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.)

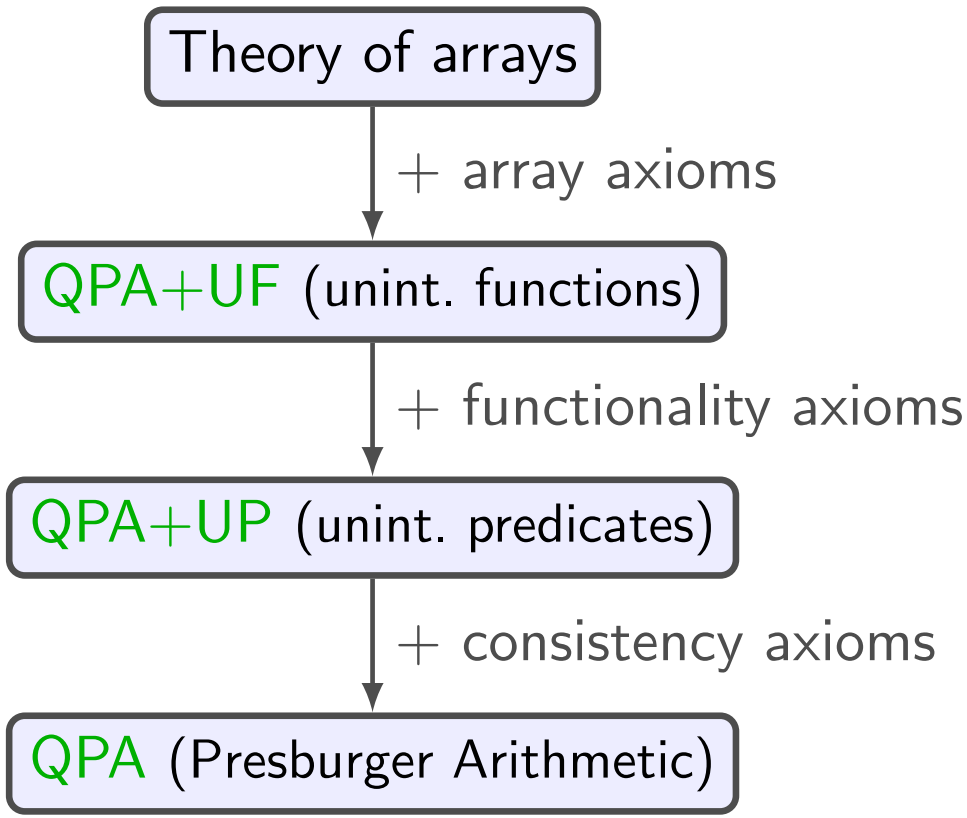
Theorem (extension of PA+UP)

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

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

(Similarly for PA+UF.)

Decidability by stack of encodings



$\Rightarrow$ Decision proc. for PA+AR, PA+UF, PA+UP

$\Rightarrow$ "Similar" to standard implementation in SMT solvers

Interpolation by stack of encodings

PA+* calculus = PA calculus
+ (ground) quantifier handling
+ theory axioms

Interpolation by stack of encodings

PA+* calculus = PA calculus
+ (ground) quantifier handling
+ theory axioms

} Interpolating version in VMCAI paper

Interpolants from proofs

- recursively annotate proof with partial interpolants, according to interpolating rules
- "total" interpolants extracted from closed proofs
- proves existence of QPA+* interpolants.

How to close PA+UP under interpolation

How to close PA+UP under interpolation

Need ability to use witness for $\alpha \mid t$ in terms:

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

with strongest interpolant

$$\exists x.\, (2x = y \ \wedge \ p(x))$$

How to close PA+UP under interpolation

Need ability to use witness for $\alpha \mid t$ in terms:

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

with strongest interpolant

$$\exists x.\,(2x = y \ \wedge \ p(x))$$

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

Need ability to use witness for $\alpha \mid t$ in terms:

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

with strongest interpolant

$$\exists x.\,(2x = y \ \wedge \ p(x))$$

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

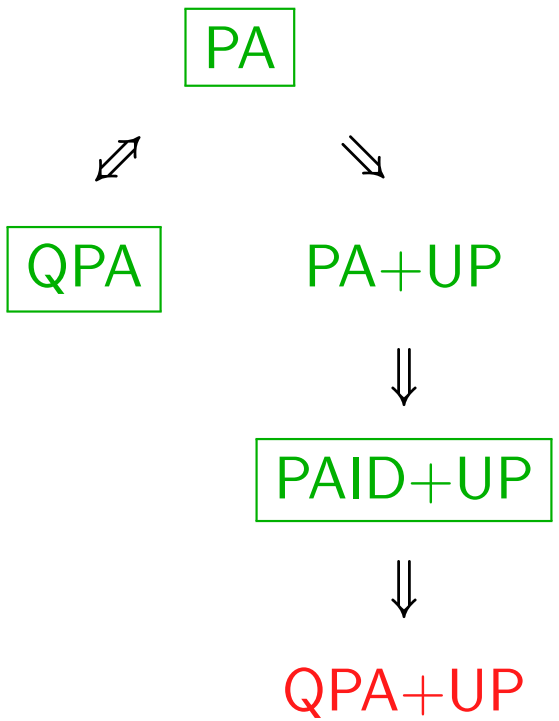
Expressiveness of guarded quantifiers

They encode integer division:

$$\exists x.(\alpha x = t \wedge \ \phi\ ) \quad \equiv \quad (\alpha \mid t) \wedge \phi[x \to (t \div \alpha)]$$

$$\underbrace{\exists x.(\alpha x = t \wedge \mathit{true})}_{\alpha \mid t} \quad \equiv \quad \alpha \cdot (t \div \alpha) = t$$

PA+UP can be closed!



Legend:

decidable

undecidable

$\square$ = closed under interpolation

$\Downarrow$ = 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

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.