Extending Proof Tree Preserving Interpolation to Sequences and Trees (Work in Progress)

Jürgen Christ Jochen Hoenicke

University of Freiburg

July 8, 2013

Extending
Proof Tree Preserving Interpolation
to
Proof Tree Preserving Tree Interpolation

Outline

Uses of Tree Interpolation

$$
\begin{tikzcd}
n = 0 \arrow[d] \\
n \leq 0 \arrow[d] \\
res = 0 \arrow[d] \\
res < n
\end{tikzcd}
$$

- Hoare-style program verification [Henzinger 04]

procedure $f(n)$ returns res
if $(n <= 0)$
    $res := 0$

assert $res >= n$

Uses of Tree Interpolation

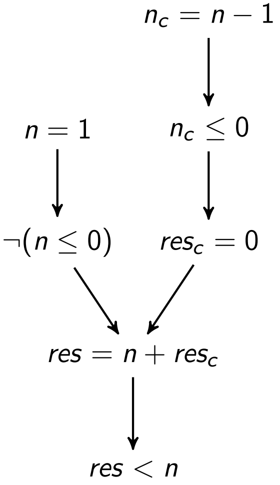
$$
\begin{array}{c}
n = 0 \\
\downarrow \top \\
n \leq 0 \\
\downarrow \top \\
res = 0 \\
\downarrow n \leq res \\
res < n
\end{array}
$$

- Hoare-style program verification [Henzinger 04]

procedure $f(n)$ returns res
if $(n <= 0)$
    $res := 0$

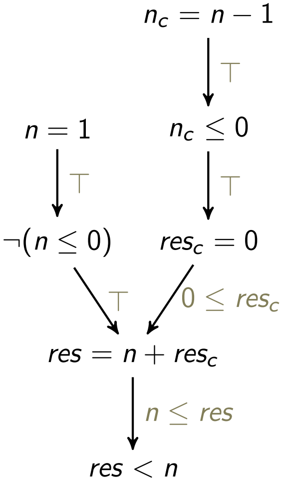
assert $res >= n$

Uses of Tree Interpolation



- Hoare-style program verification [Henzinger 04, Heizmann 10]

procedure $f(n)$ returns res
if $(n <= 0)$
    $res := 0$
else
    $res := n + \textbf{call } f(n-1)$
assert $res >= n$



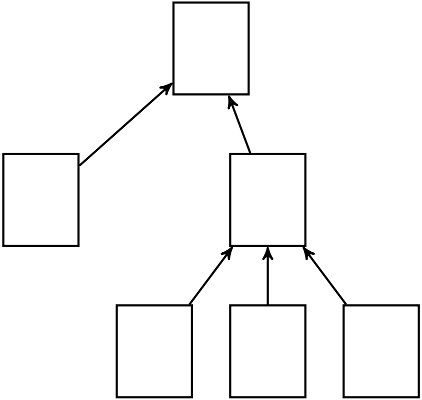
- Hoare-style program verification [Henzinger 04, Heizmann 10]

procedure $f(n)$ returns res
if $(n <= 0)$
  $res := 0$
else
  $res := n +$ **call** $f(n-1)$
assert $res >= n$

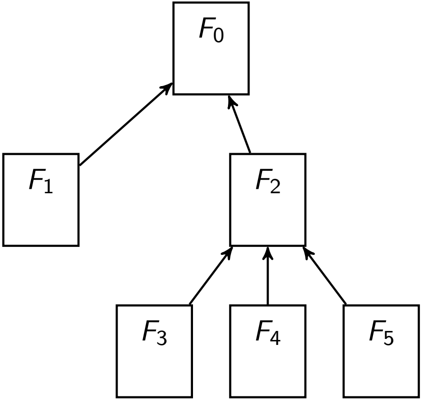
Uses of Tree Interpolation

- Hoare-style program verification [Henzinger 04, Heizmann 10]
- Verification of multi-threaded programs and higher order programs [Rybalchenko 12]
- Incremental update checking [Sery 11]
- Solving non-recursive Horn clauses [Rybalchenko 11]
- Inductive Dataflow Graphs [Podelski 13]
- ...

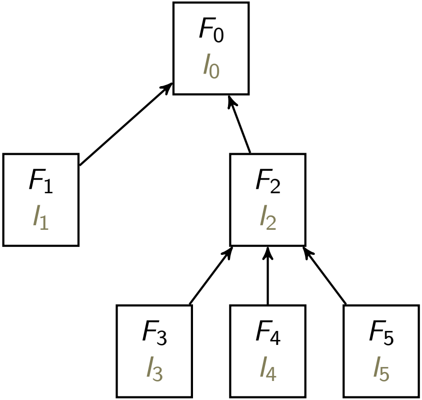
Tree Interpolation Problem



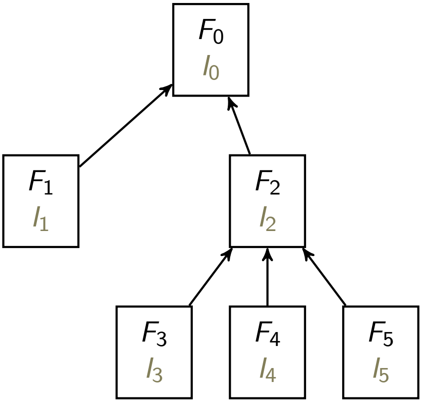
Tree Interpolation Problem



$\bigwedge F_i$ is unsatisfiable



$\bigwedge F_i$ is unsatisfiable

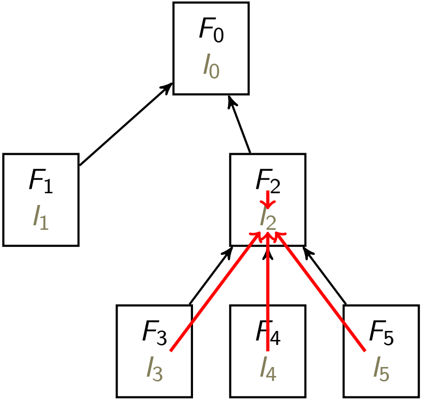


$\bigwedge F_i$ is unsatisfiable

Tree Inductivity:

- $I_0 \equiv \bot$

Tree Interpolation Problem

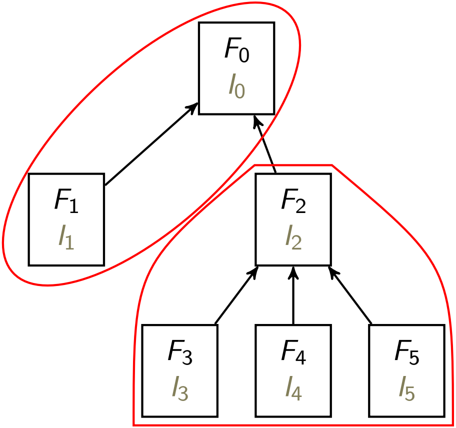


$\bigwedge F_i$ is unsatisfiable

Tree Inductivity:

- $I_0 \equiv \bot$
- Child interpolants and parent imply parent interpolant

Tree Interpolation Problem



$\bigwedge F_i$ is unsatisfiable

Tree Inductivity:

- $I_0 \equiv \bot$
- Child interpolants and parent imply parent interpolant
- Interpolant only contains symbols occurring inside and outside the current subtree

Outline

For $A \wedge B \models_{\mathcal{T}} \bot$:

- $A \models_{\mathcal{T}} I$,
- $B \wedge I \models_{\mathcal{T}} \bot$,
- $symb(I) \subseteq symb(A) \cap symb(B)$

Binary Interpolation

For $A \land B \models_{\mathcal{T}} \bot$:

- $A \models_{\mathcal{T}} I$,
- $B \land I \models_{\mathcal{T}} \bot$,
- $symb(I) \subseteq symb(A) \cap symb(B)$

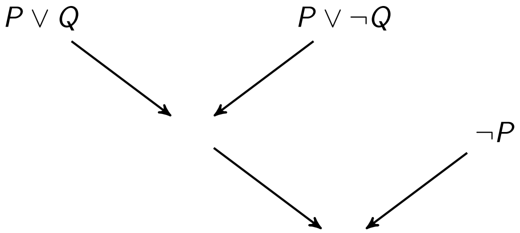
$$\begin{tikzcd} B \\ A \arrow[u] \end{tikzcd}$$

Outline

Resolution Refutation

Proof consists of

- leaves representing input clauses,



Proof consists of

- leaves representing input clauses,
- inner nodes derived by resolution

$$\frac{C_1 \vee \ell \qquad C_2 \vee \neg\ell}{C_1 \vee C_2}$$

$$\begin{tikzcd}
P \vee Q \arrow[dr] & & P \vee \neg Q \arrow[dl] & \\
& P \arrow[dr] & & \neg P \arrow[dl] \\
& & {} &
\end{tikzcd}$$

Proof consists of

- leaves representing input clauses,
- inner nodes derived by resolution

$$\frac{C_1 \lor \ell \qquad C_2 \lor \neg\ell}{C_1 \lor C_2}$$

- the root node representing the empty clause.

$$\begin{tikzcd}
P \lor Q \arrow[dr] & & P \lor \neg Q \arrow[dl] & \\
& P \arrow[dr] & & \neg P \arrow[dl] \\
& & \bot &
\end{tikzcd}$$

Labelled Resolution Refutation

Label each clause in the resolution refutation with *partial interpolant*

$$\begin{tikzcd}
P \lor Q \ : \ I_{P \lor Q} \arrow[dr] & & P \lor \neg Q \ : \ I_{P \lor \neg Q} \arrow[dl] & \\
& P \ : \ I_P \arrow[dr] & & \neg P \ : \ I_{\neg P} \arrow[dl] \\
& & \bot \ : \ I_\bot &
\end{tikzcd}$$

Labelled Resolution Refutation

Label each clause in the resolution refutation with *partial interpolant*

$$
\begin{tikzcd}
P \lor Q \ : \ I_{P \lor Q} \arrow[dr] & & P \lor \neg Q \ : \ I_{P \lor \neg Q} \arrow[dl] & \\
& P \ : \ I_P \arrow[dr] & & \neg P \ : \ I_{\neg P} \arrow[dl] \\
& & \perp \ : \ I_{\perp} &
\end{tikzcd}
$$

- Syntactic rules for leaves

Labelled Resolution Refutation

Label each clause in the resolution refutation with *partial interpolant*

$$\ell \in A \; \frac{C_1 \vee \ell \ : \ I_1 \qquad C_2 \vee \neg\ell \ : \ I_2}{C_1 \vee C_2 \ : \ I_1 \vee I_2}$$

$$\ell \in B \; \frac{C_1 \vee \ell \ : \ I_1 \qquad C_2 \vee \neg\ell \ : \ I_2}{C_1 \vee C_2 \ : \ I_1 \wedge I_2}$$

$$\begin{tikzcd}
P \vee Q \ : \ I_{P \vee Q} \arrow[dr] & & P \vee \neg Q \ : \ I_{P \vee \neg Q} \arrow[dl] & \\
& P \ : \ I_P \arrow[dr] & & \neg P \ : \ I_{\neg P} \arrow[dl] \\
& & \bot \ : \ I_\bot &
\end{tikzcd}$$

- Syntactic rules for leaves
- Interpolant of resolved based on interpolants of antecedents and pivot

Labelled Resolution Refutation

Label each clause in the resolution refutation with *partial interpolant*

$$\ell \in {\color{blue}A}\ \frac{\begin{array}{c} C_1 \vee \ell \ : \ I_1 \\ C_2 \vee \neg\ell \ : \ I_2 \end{array}}{C_1 \vee C_2 \ : \ I_1 \vee I_2}$$

$$\ell \in {\color{red}B}\ \frac{\begin{array}{c} C_1 \vee \ell \ : \ I_1 \\ C_2 \vee \neg\ell \ : \ I_2 \end{array}}{C_1 \vee C_2 \ : \ I_1 \wedge I_2}$$

$$\begin{tikzcd}
P \vee Q \ : \ I_{P\vee Q} \arrow[dr] & & P \vee \neg Q \ : \ I_{P\vee\neg Q} \arrow[dl] & \\
& P \ : \ I_P \arrow[dr] & & \neg P \ : \ I_{\neg P} \arrow[dl] \\
& & \bot \ : \ I_\bot &
\end{tikzcd}$$

- Syntactic rules for leaves
- Interpolant of resolved based on interpolants of antecedents and pivot

$I_\bot$ is desired interpolant.

Partial Interpolants

Partial interpolant I_C of clause C is interpolant of

$$A \wedge B \wedge \neg C$$

Partial interpolant I_C of clause C is interpolant of

$$A \land B \land \neg C$$

How to split $\neg C$?

Partial Interpolants

Partial interpolant I_C of clause C is interpolant of

$$A \wedge B \wedge \neg C$$

Define $\neg C \downharpoonleft A$ and $\neg C \downharpoonleft B$ such that

- $symb(\neg C \downharpoonleft A) \subseteq symb(A)$
- $symb(\neg C \downharpoonleft B) \subseteq symb(B)$
- $\neg C \leftrightarrow \neg C \downharpoonleft A \wedge \neg C \downharpoonleft B$

Partial Interpolants

Partial interpolant I_C of clause C is interpolant of

$$A \land B \land \neg C$$

Define $\neg C \downharpoonleft A$ and $\neg C \downharpoonleft B$ such that

- $symb(\neg C \downharpoonleft A) \subseteq symb(A)$
- $symb(\neg C \downharpoonleft B) \subseteq symb(B)$
- $\neg C \leftrightarrow \neg C \downharpoonleft A \land \neg C \downharpoonleft B$

Partial interpolant I_C is interpolant of
$A \land ((\neg C) \downharpoonleft A)$ and $B \land ((\neg C) \downharpoonleft B)$.

Outline

Additional Leaves

- Theory lemmas
- Theory combination lemmas

$$x \leq y \vee x \neq y$$
$$x \geq y \vee x \neq y$$
$$x < y \vee x > y \vee x = y$$

Additional Leaves

- Theory lemmas
- Theory combination lemmas

$$x \leq y \vee x \neq y$$
$$x \geq y \vee x \neq y$$
$$x < y \vee x > y \vee x = y$$

might contain literals that are not in the input formulas

Mixed Literals

- literals that contain symbols only in A and symbols only in B: $a = b$

Mixed Literals

- literals that contain symbols only in A and symbols only in B: $a = b$
- literals do not occur in input formulas

Mixed Literals

- literals that contain symbols only in A and symbols only in B: $a = b$
- literals do not occur in input formulas
- created by
  - theory combination (Nelson-Oppen, Ackermannization),
  - cuts and extended branches used to solve integer arithmetic,
  - ...

Mixed Literals

- literals that contain symbols only in A and symbols only in B: $a = b$
- literals do not occur in input formulas
- created by
  - theory combination (Nelson-Oppen, Ackermannization),
  - cuts and extended branches used to solve integer arithmetic,
  - ...

What is $a = b \downharpoonleft A$ and $a = b \downharpoonleft B$?

Interpolation and Mixed Literals

Purification:
replace $a \leq b$ by $a \leq x \land x \leq b$
similar to purification in Nelson-Oppen

Purification:
replace $a \leq b$ by $a \leq x \wedge x \leq b$
similar to purification in Nelson-Oppen

Interpolation:
Remove purification variable on resolution:

$$\frac{C_1 \vee a \leq b \;:\; I_1(x_1) \qquad C_2 \vee \neg(a \leq b) \;:\; I_2(x_2)}{C_1 \vee C_2 \;:\; I_3}$$

Purification:
replace $a \leq b$ by $a \leq x \wedge x \leq b$
similar to purification in Nelson-Oppen

Interpolation:
Remove purification variable on resolution:

$$\frac{C_1 \vee a \leq b \;:\; I_1(x_1) \qquad C_2 \vee \neg(a \leq b) \;:\; I_2(x_2)}{C_1 \vee C_2 \;:\; I_3}$$

Rules for uninterpreted functions and linear arithmetic [TACAS 2013]

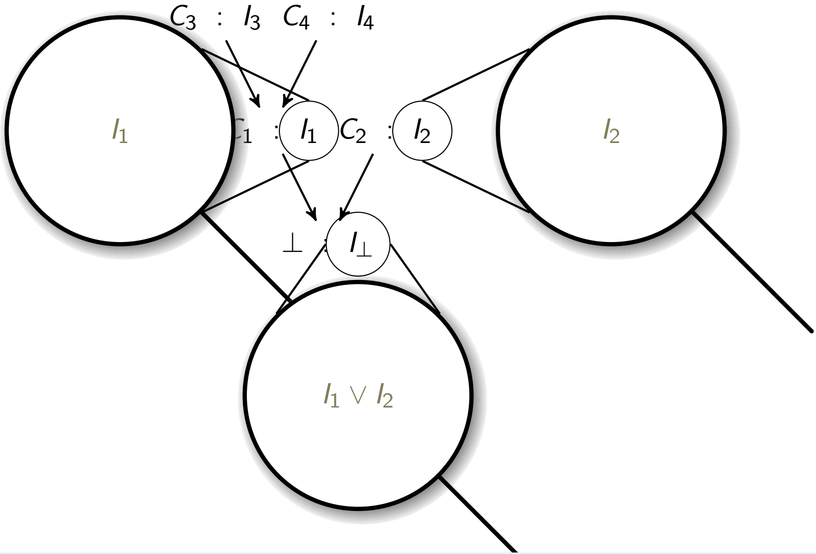
Outline

Idea

Binary Interpolation:

$$
\begin{tikzcd}
C_3 : I_3 \arrow[dr] & & C_4 : I_4 \arrow[dl] & \\
& C_1 : I_1 \arrow[dr] & & C_2 : I_2 \arrow[dl] \\
& & \bot : I_\bot &
\end{tikzcd}
$$

Binary Interpolation:

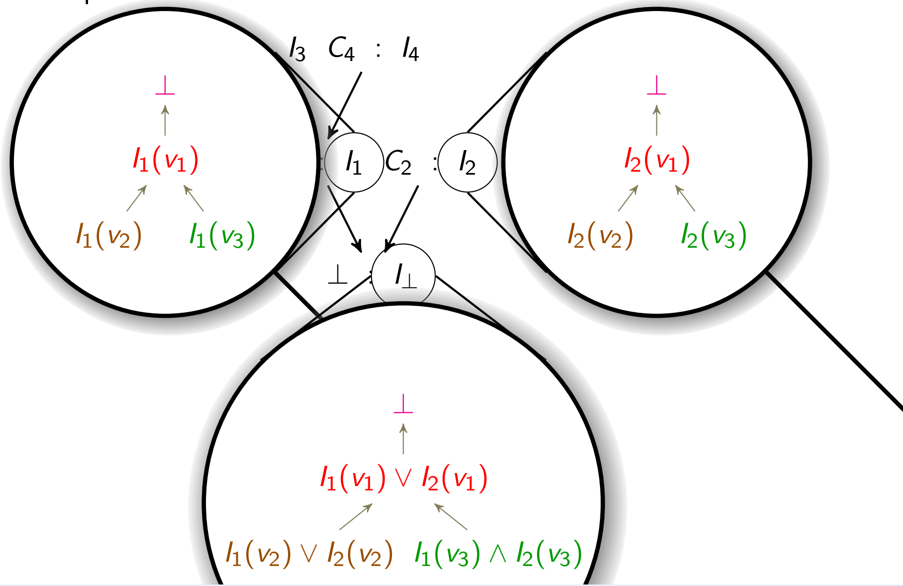


Idea

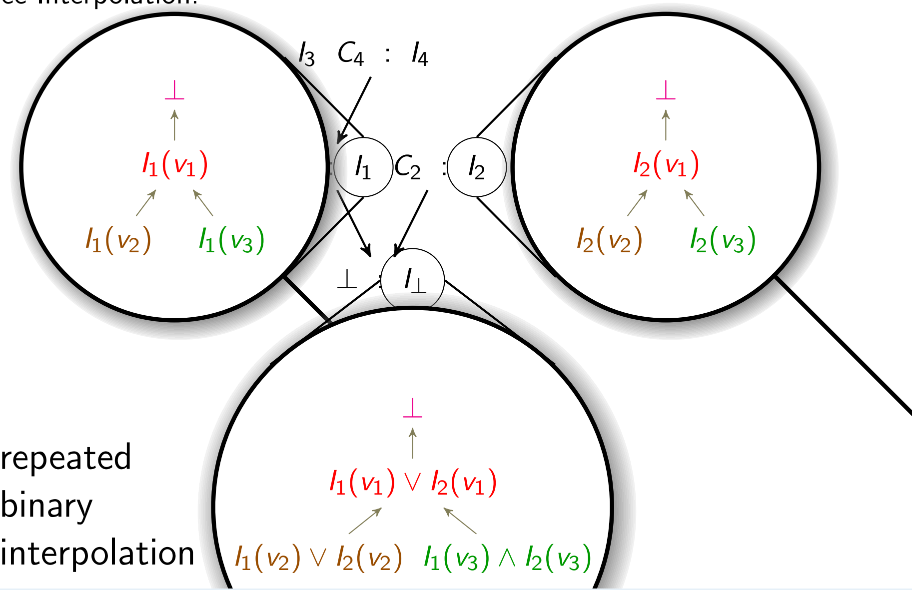
Tree Interpolation:

$$
\begin{tikzcd}
C_3 : I_3 \arrow[dr] & & C_4 : I_4 \arrow[dl] & \\
& C_1 : I_1 \arrow[dr] & & C_2 : I_2 \arrow[dl] \\
& & \bot : I_\bot &
\end{tikzcd}
$$

Tree Interpolation:



Tree Interpolation:



repeated binary interpolation

Partial tree interpolant I_C for clause C is tree interpolant of

$$\begin{tikzcd} & F_0 & & \\ & F_1 \arrow[u] & & \land \neg C \\ F_2 \arrow[ur] & & F_3 \arrow[ul] & \end{tikzcd}$$

How to split $\neg C$?

Partial Tree Interpolants

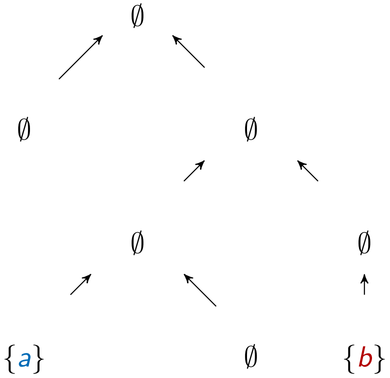
Partial tree interpolant I_C for clause C is tree interpolant of

$$\begin{tikzcd}[column sep=small]
 & F_0 \land ((\neg C) \downharpoonleft v_0) & \\
 & F_1 \land ((\neg C) \downharpoonleft v_1) \arrow[u] & \\
F_2 \land ((\neg C) \downharpoonleft v_2) \arrow[ur] & & F_3 \land ((\neg C) \downharpoonleft v_3) \arrow[ul]
\end{tikzcd}$$

- One purification function per node
- $\ell \leftrightarrow \exists \overline{x}.\ \bigwedge_v \ell \downharpoonleft v$

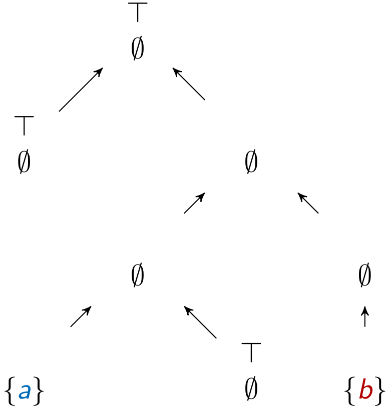
Projection of Mixed Literals

- one auxiliary variable for every node in which literal is mixed
- projection of $a = b$:



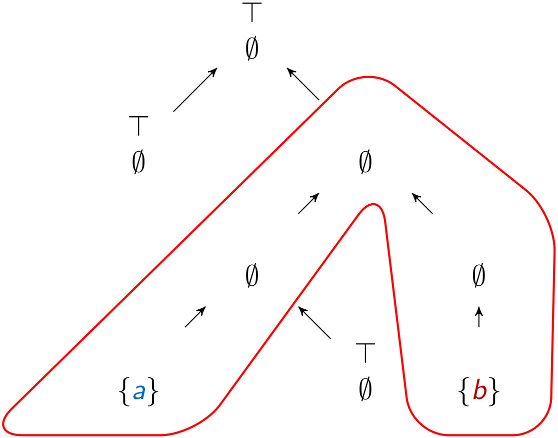
Projection of Mixed Literals

- one auxiliary variable for every node in which literal is mixed
- projection of $a = b$:

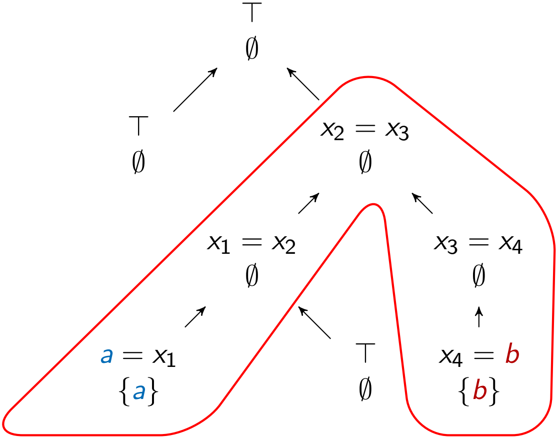


Projection of Mixed Literals

- one auxiliary variable for every node in which literal is mixed
- projection of $a = b$:



- one auxiliary variable for every node in which literal is mixed
- projection of $a = b$:



Outline

$$
\begin{tikzcd}
& \begin{array}{c}\{q, r\}\\ q \neq r\end{array} & \\
\begin{array}{c}\{c, d\}\\ c = d\end{array} \arrow[ur] & & \begin{array}{c}\{b, d, r, f(\cdot)\}\\ d = b \wedge f(b) = r\end{array} \arrow[ul] \\
\begin{array}{c}\{a, c, q, f(\cdot)\}\\ a = c \wedge q = f(a)\end{array} \arrow[u] & &
\end{tikzcd}
$$

$$
\frac{a = b \vee a \neq c \vee c \neq d \vee d \neq b \qquad a \neq b \vee q \neq f(a) \vee f(b) \neq r \vee q = r}{a \neq c \vee c \neq d \vee d \neq b \vee q \neq f(a) \vee f(b) \neq r \vee q = r}
$$

Interpolation Problem and Proof Excerpt

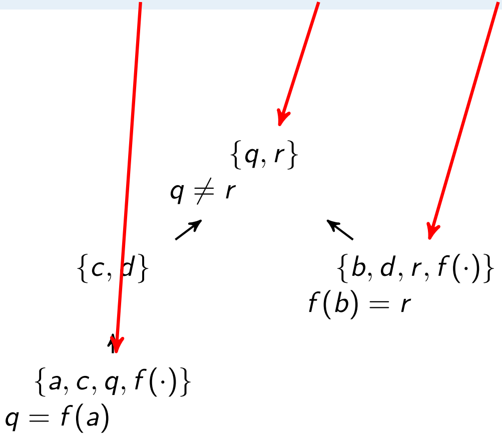
$$\begin{tikzcd}[column sep=small]
& \begin{matrix}\{q, r\} \\ q \neq r\end{matrix} & \\
\begin{matrix}\{c, d\} \\ c = d\end{matrix} \arrow[ur] & & \begin{matrix}\{b, d, r, f(\cdot)\} \\ d = b \wedge f(b) = r\end{matrix} \arrow[ul] \\
\begin{matrix}\{a, c, q, f(\cdot)\} \\ a = c \wedge q = f(a)\end{matrix} \arrow[u] & &
\end{tikzcd}$$

$$\frac{a = b \vee a \neq c \vee c \neq d \vee d \neq b \qquad a \neq b \vee q \neq f(a) \vee f(b) \neq r \vee q = r}{a \neq c \vee c \neq d \vee d \neq b \vee q \neq f(a) \vee f(b) \neq r \vee q = r}$$

Projection: $a = b \wedge q = f(a) \wedge q \neq r \wedge f(b) = r$

$$
\begin{tikzcd}
 & \{q, r\} & \\
\{c, d\} \arrow[ur] & & \{b, d, r, f(\cdot)\} \arrow[ul] \\
\{a, c, q, f(\cdot)\} \arrow[u] & &
\end{tikzcd}
$$

Projection: $a = b \wedge q = f(a) \wedge q \neq r \wedge f(b) = r$



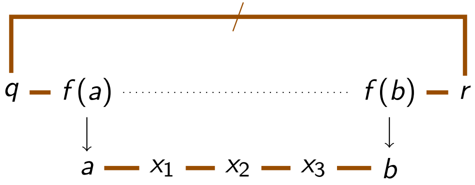
$$\begin{array}{ccc}
 & \{q, r\} & \\
 & q \neq r \wedge x_2 = x_3 & \\
\nearrow & & \nwarrow \\
\{c, d\} & & \{b, d, r, f(\cdot)\} \\
x_1 = x_2 & & f(b) = r \wedge x_3 = b \\
\uparrow & & \\
\{a, c, q, f(\cdot)\} & & \\
q = f(a) \wedge a = x_1 & &
\end{array}$$

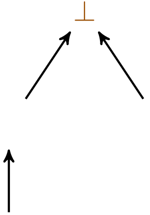
Interpolation: $a = b \land q = f(a) \land f(b) = r \land q \neq r$

$q \neq r \land x_2 = x_3$

$x_1 = x_2$ $\quad$ $f(b) = r \land x_3 = b$

$q = f(a) \land a = x_1$

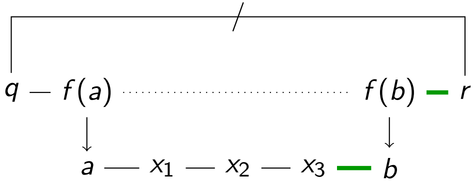


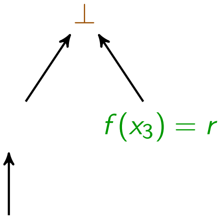


$q \neq r \wedge x_2 = x_3$

$x_1 = x_2$ $\quad$ $f(b) = r \wedge x_3 = b$

$q = f(a) \wedge a = x_1$

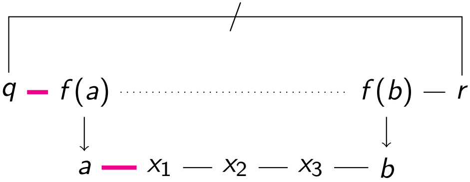




Interpolation: $a = b \wedge q = f(a) \wedge f(b) = r \wedge q \neq r$

$$\begin{tikzcd}
& q \neq r \wedge x_2 = x_3 & \\
x_1 = x_2 \arrow[ur] & & f(b) = r \wedge x_3 = b \arrow[ul] \\
\color{magenta}{q = f(a) \wedge a = x_1} \arrow[u] & &
\end{tikzcd}$$

$$\begin{tikzcd}
& \bot & \\
{} \arrow[ur] & & \color{green}{f(x_3) = r} \arrow[ul] \\
\color{magenta}{q = f(x_1)} \arrow[u] & &
\end{tikzcd}$$



$$q \neq r \wedge x_2 = x_3$$

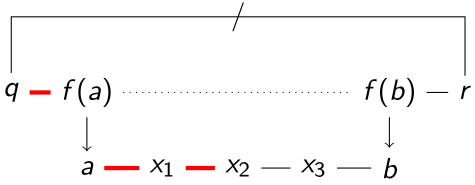
$$x_1 = x_2 \qquad f(b) = r \wedge x_3 = b$$

$$q = f(a) \wedge a = x_1$$

$$\perp$$

$$q = f(x_2) \qquad f(x_3) = r$$

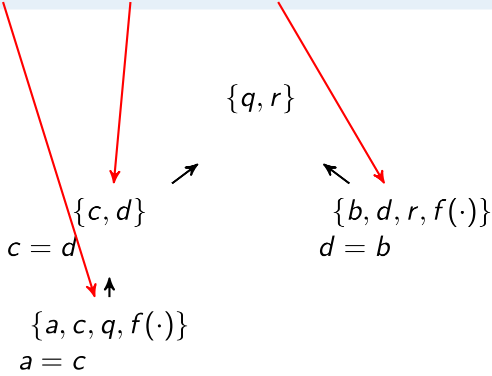
$$q = f(x_1)$$



Projection: $a = c \land c = d \land d = b \land a \neq b$

$$
\begin{tikzcd}
 & \{q, r\} & \\
\{c, d\} \arrow[ur] & & \{b, d, r, f(\cdot)\} \arrow[ul] \\
\{a, c, q, f(\cdot)\} \arrow[u] & &
\end{tikzcd}
$$

Projection: $a = c \wedge c = d \wedge d = b \wedge a \neq b$



Projection: $a = c \wedge c = d \wedge d = b \wedge a \neq b$

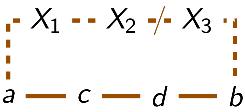
$$
\begin{tikzcd}
 & \begin{array}{c} \{q, r\} \\ X_2 \cap X_3 = \emptyset \end{array} & \\
\begin{array}{c} \{c, d\} \\ c = d \wedge X_1 \subseteq X_2 \end{array} \arrow[ur] & & \begin{array}{c} \{b, d, r, f(\cdot)\} \\ d = b \wedge b \in X_3 \end{array} \arrow[ul] \\
\begin{array}{c} \{a, c, q, f(\cdot)\} \\ a = c \wedge a \in X_1 \end{array} \arrow[u] & &
\end{tikzcd}
$$

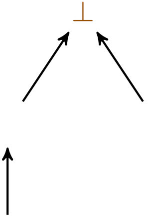
- X_1, X_2, X_3 set-valued
- X_i separates a and b
- No reasoning about sets required in the solver

$$X_2 \cap X_3 = \emptyset$$

$$c = d \wedge X_1 \subseteq X_2 \qquad d = b \wedge b \in X_3$$

$$a = c \wedge a \in X_1$$





$$X_2 \cap X_3 = \emptyset$$

$$c = d \land X_1 \subseteq X_2 \qquad d = b \land b \in X_3$$

$$a = c \land a \in X_1$$

$$X_1 \; \text{--} \; X_2 \; \not\text{--} \; X_3$$

$$a \;\text{—}\; c \;\text{—}\; d \;\text{—}\; b$$

$$\perp$$

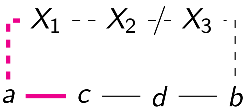
$$d \in X_3$$

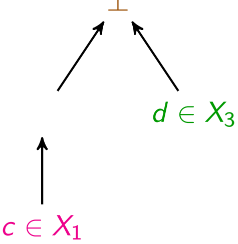
Interpolation: $a = c \wedge c = d \wedge d = b \wedge a \neq b$

$X_2 \cap X_3 = \emptyset$

$c = d \wedge X_1 \subseteq X_2 \quad d = b \wedge b \in X_3$

$a = c \wedge a \in X_1$

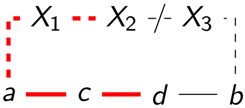


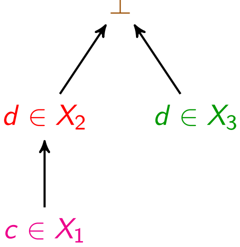


$$X_2 \cap X_3 = \emptyset$$

$$c = d \wedge X_1 \subseteq X_2 \qquad d = b \wedge b \in X_3$$

$$a = c \wedge a \in X_1$$





Magic Rule for Resolution on Mixed Equalities

- partial interpolant for $C_1 \vee a = b$ has form $I_1[s \in X]$
  "If $s \in X$ holds, then $s = a$ resp. $s = b$ (whichever is in the subtree)"
- partial interpolant for $C_2 \vee a \neq b$ has form $I_2(x)$
  "$I_2(x)$ holds for a resp. b (whichever is in the subtree)"

Magic Rule for Resolution on Mixed Equalities

- partial interpolant for $C_1 \vee a = b$ has form $I_1[s \in X]$
  "If $s \in X$ holds, then $s = a$ resp. $s = b$ (whichever is in the subtree)"
- partial interpolant for $C_2 \vee a \neq b$ has form $I_2(x)$
  "$I_2(x)$ holds for a resp. b (whichever is in the subtree)"
- partial interpolant for the resolvent $C_1 \vee C_2$

$$I_1[I_2(s)]$$

Interpolating the Resolution Step

$$\bot \qquad\qquad \bot$$

$$C_1 \lor a = b \;:\; d \in X_2 \;\; d \in X_3 \qquad\qquad C_2 \lor a \neq b \;:\; q = f(x_2) \;\; f(x_3) = r$$

$$c \in X_1 \qquad\qquad q = f(x_1)$$

$$C_1 \lor C_2 \;:$$

Interpolating the Resolution Step

$$\begin{tikzcd}[column sep=small]
 & & \bot & \\
C_1 \lor a = b \; : & d \in X_2 \arrow[ur] & & d \in X_3 \arrow[ul] \\
 & c \in X_1 \arrow[u] & &
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small]
 & & \bot & \\
C_2 \lor a \neq b \; : & q = f(x_2) \arrow[ur] & & f(x_3) = r \arrow[ul] \\
 & q = f(x_1) \arrow[u] & &
\end{tikzcd}$$

$$\begin{tikzcd}[column sep=small]
 & & \bot & \\
C_1 \lor C_2 \; : & q = f(d) \arrow[ur] & & f(d) = r \arrow[ul] \\
 & q = f(c) \arrow[u] & &
\end{tikzcd}$$

Outline

Conclusion

- We extended our interpolation scheme to sequence and tree interpolation.
- Tree interpolation is repeated binary interpolation.
- Scheme computes quantifier-free interpolants in the combination of UF and LA, in particular in QF_UFLIA.
- No need to manipulate resolution proof.
- Independent of the solver or proof search.
- Correctness proofs still work in progress.

Conclusion

- We extended our interpolation scheme to sequence and tree interpolation.
- Tree interpolation is repeated binary interpolation.
- Scheme computes quantifier-free interpolants in the combination of UF and LA, in particular in QF_UFLIA.
- No need to manipulate resolution proof.
- Independent of the solver or proof search.
- Correctness proofs still work in progress.
- Scheme is implemented in SMTInterpol.

`http://ultimate.informatik.uni-freiburg.de/smtinterpol`

Thanks for your attention