Efficient Interpolation for the Theory of Arrays

Tanja Schindler Jochen Hoenicke

University of Freiburg

July 23, 2017

Motivation

Interpolation for the Theory of Arrays

- model checkers use interpolants to automatically generate invariants
- theory of arrays represents the data type arrays, memory, parallel processes

Motivation

Interpolation for the Theory of Arrays

- model checkers use interpolants to automatically generate invariants
- theory of arrays represents the data type arrays, memory, parallel processes

Existing work

[Bruttomesso et al. 2011], [Totla&Wies 2013]

- the solver needs to know the interpolation problem in advance
- need to re-run the solver for each partitioning to compute sequence or tree interpolants

Outline

Motivation

Background
- Proof Tree Preserving Interpolation
- Theory of Arrays
- Weakly Equivalent Arrays

Interpolants for the Theory of Arrays
- Read-Over-Weakeq Lemmas
- Weakeq-Ext Lemmas

Conclusion

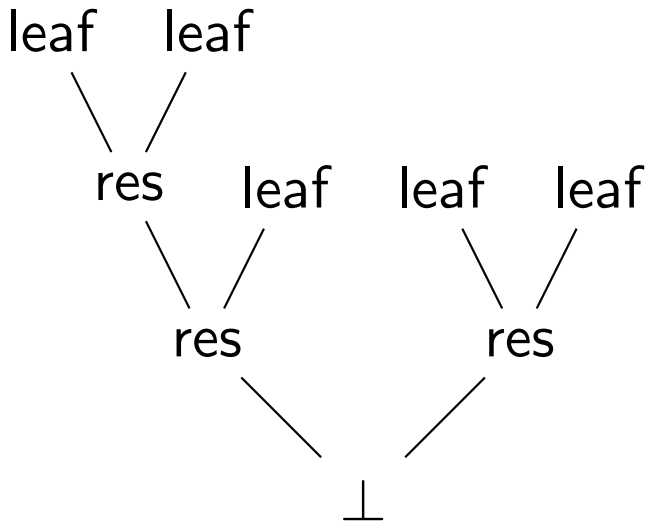
Craig Interpolant

Given formulas A, B such that $A \wedge B$ is unsatisfiable,
a Craig interpolant is a formula I such that

- A implies I
- $I \wedge B$ is unsatisfiable
- I contains only symbols that occur in both A and B.

Interpolants from Proofs

Resolution Proof for F

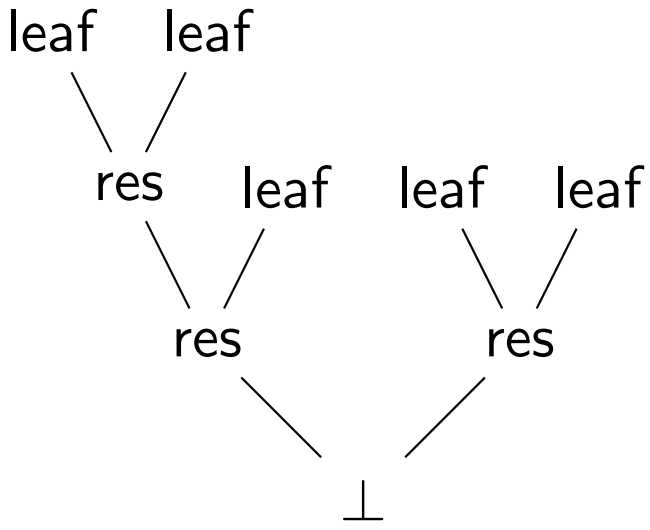


Derive Interpolants

- compute partial interpolants for leaf nodes
  - theory lemmas : need special procedures
- combine them at resolution nodes
- the partial interpolant of $\bot$ is the interpolant for $F : A \wedge B$

Interpolants from Proofs

Resolution Proof for F



Derive Interpolants

- compute partial interpolants for leaf nodes
  - theory lemmas : need special procedures
- combine them at resolution nodes
- the partial interpolant of $\bot$ is the interpolant for $F : A \wedge B$

Proof Tree Preserving Interpolation

- can be applied to proofs unaware of the interpolation problem
- can handle mixed (in)equalities
- can not handle mixed terms

[Christ et al. 2013]

Proof Tree Preserving Interpolation

- Mixed equalities: $A \wedge B \wedge a = b$

  Interpolate $A \wedge a = x$ and $B \wedge x = b$ for fresh x

  Interpolant contains x

Proof Tree Preserving Interpolation

- Mixed equalities: $A \wedge B \wedge a = b$

  Interpolate $A \wedge a = x$ and $B \wedge x = b$ for fresh x

  Interpolant contains x

- Mixed disequalities: $A \wedge B \wedge a \neq b$

  Interpolate $A \wedge \mathrm{EQ}(x, a)$ and $B \wedge \neg\mathrm{EQ}(x, b)$ with uninterpreted predicate EQ

  Interpolant contains $\mathrm{EQ}(x, s)$,

Proof Tree Preserving Interpolation

- Mixed equalities: $A \wedge B \wedge a = b$

  Interpolate $A \wedge a = x$ and $B \wedge x = b$ for fresh x

  Interpolant contains x

- Mixed disequalities: $A \wedge B \wedge a \neq b$

  Interpolate $A \wedge \mathrm{EQ}(x, a)$ and $B \wedge \neg\mathrm{EQ}(x, b)$ with uninterpreted predicate EQ

  Interpolant contains $\mathrm{EQ}(x, s)$,

  and $A \wedge B \longrightarrow a = s \wedge s = b$ (equality-interpolating theories)

The Axioms of the Theory of Arrays

$$a\langle i \triangleleft v\rangle[i] = v \qquad \text{(idx)}$$

$$i \neq k \longrightarrow a\langle k \triangleleft v\rangle[i] = a[i] \qquad \text{(read-over-write)}$$

$$(\forall\ i.\ a[i] = b[i]) \longrightarrow a = b \qquad \text{(ext)}$$

The Axioms of the Theory of Arrays

$$a\langle i \triangleleft v\rangle[i] = v \qquad \text{(idx)}$$

$$i \neq k \longrightarrow a\langle k \triangleleft v\rangle[i] = a[i] \qquad \text{(read-over-write)}$$

$$a[\mathrm{diff}(a,b)] = b[\mathrm{diff}(a,b)] \longrightarrow a = b \qquad \text{(ext-diff)}$$

[Bruttomesso et al. 2011]

Weakly Equivalent Arrays

Problem

Proof Tree Preserving Interpolation cannot handle mixed terms.blue

Example

Proof for interpolation problem

$$A{:=}\ s_1\langle k \triangleleft v\rangle = s_2 \wedge f(k) = 0$$
$$B{:=}\ s_1[i] \neq s_2[i] \wedge f(i) = 1$$

Instantiating read-over-write

$$i \neq k \longrightarrow s_1\langle k \triangleleft v\rangle[i] = s_1[i]$$

Weakly Equivalent Arrays

Problem

Proof Tree Preserving Interpolation cannot handle mixed terms.blue

Example

Proof for interpolation problem

$$A{:=}\ s_1\langle k \lhd v\rangle = s_2 \wedge f(k) = 0$$
$$B{:=}\ s_1[i] \neq s_2[i] \wedge f(i) = 1$$

Instantiating read-over-write

$$i \neq k \longrightarrow s_1\langle k \lhd v\rangle[i] = s_1[i]$$

Solution

Avoid creating mixed terms:
Weakly Equivalent Arrays [Christ&Hoenicke 2015]

Read-Over-Weakeq

Extended Array Lemma

$$i \neq k \longrightarrow a\langle k \triangleleft v \rangle[i] = a[i] \qquad \text{(read-over-write)}$$

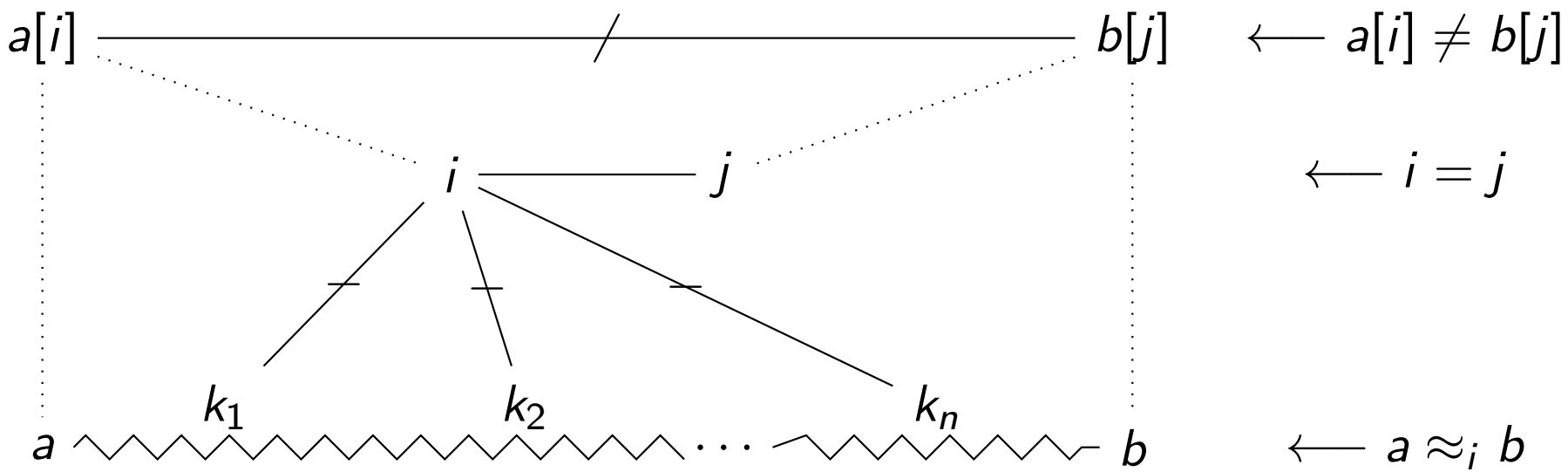
Read-Over-Weakeq

Extended Array Lemma

$$a \approx_i b \wedge i = j \longrightarrow a[i] = b[j] \qquad \text{(read-over-weakeq)}$$

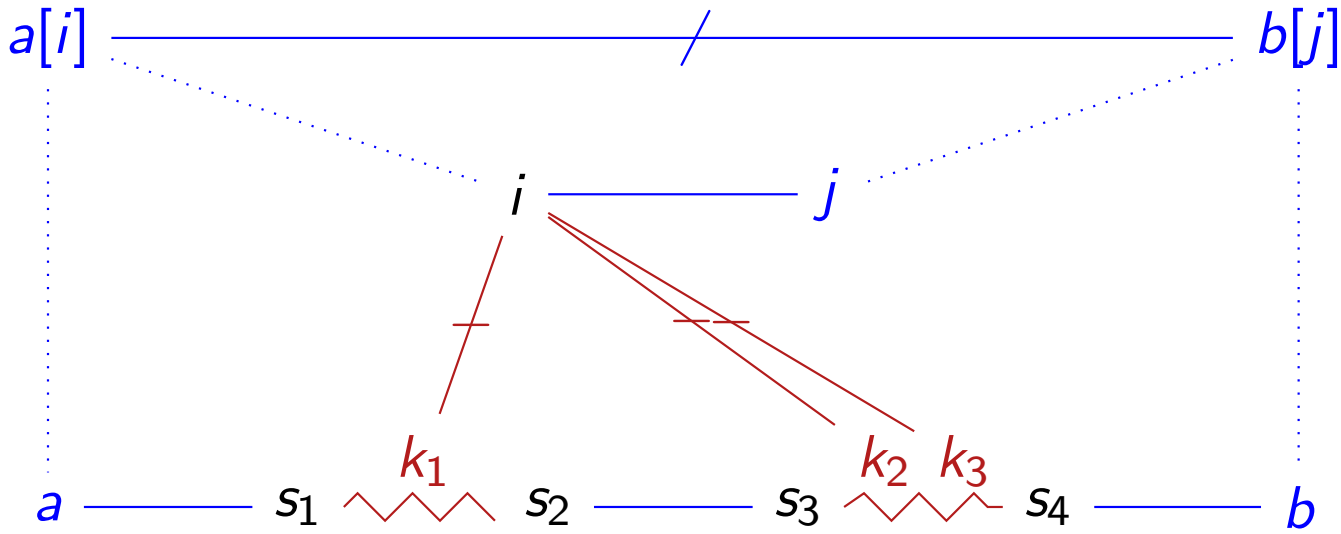
($a \approx_i b$: connected by path of array equalities and writes not on i)

Visualization of the corresponding conflict



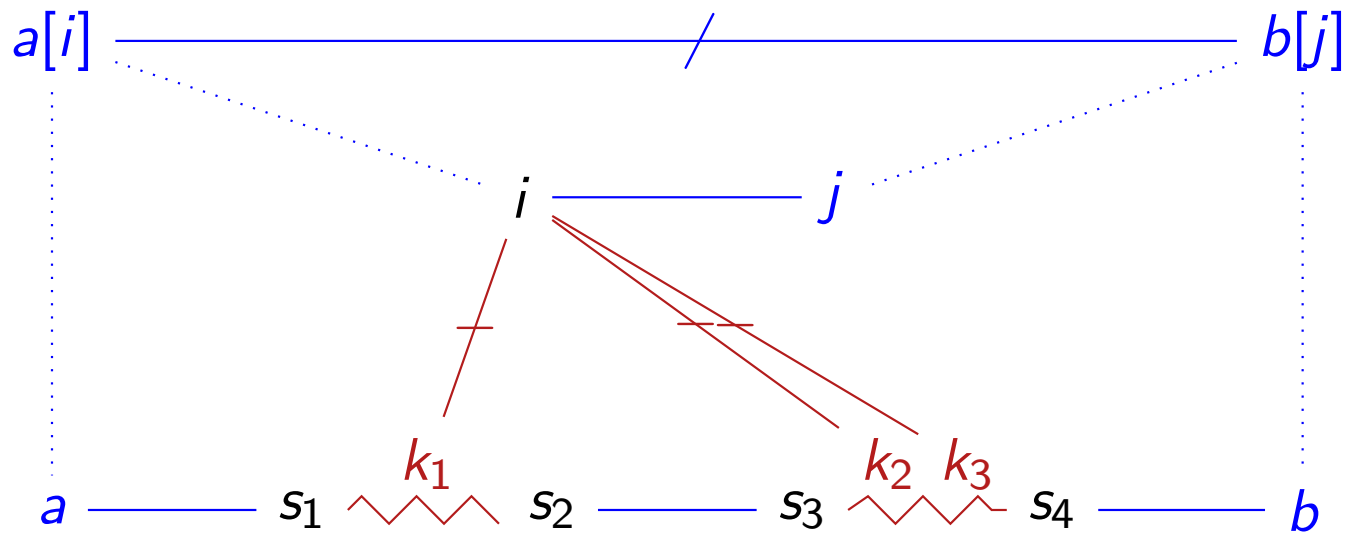
Interpolants for Read-Over-Weakeq

. . . with i in both A and B



Interpolants for Read-Over-Weakeq

. . . with i in both A and B

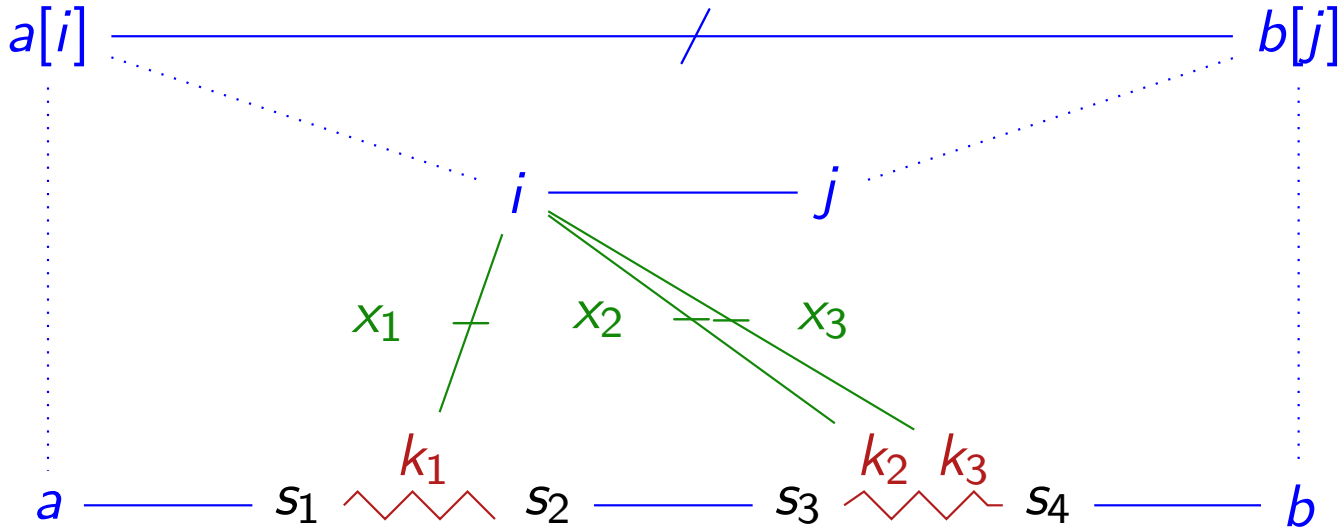


Interpolant

$I : s_1[i] = s_2[i] \wedge s_3[i] = s_4[i]$

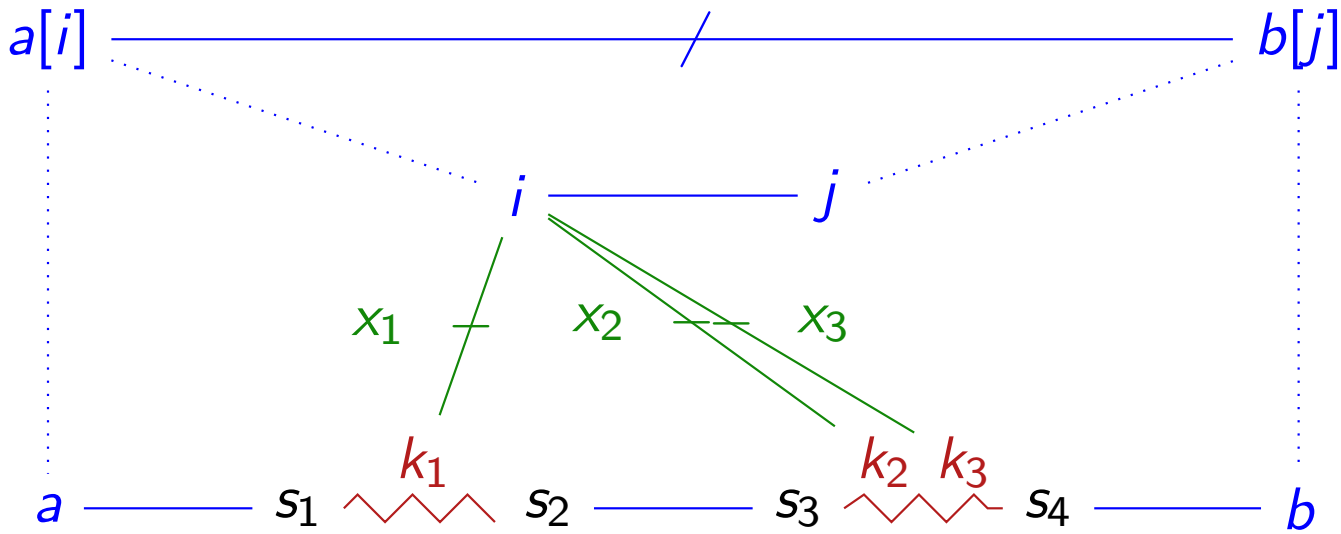
Interpolants for Read-Over-Weakeq

. . . with i and j in B



Interpolants for Read-Over-Weakeq

. . . with i and j in B



Interpolant

$I : \mathrm{weq}(s_1, s_2, 1, \mathrm{EQ}(x_1, \cdot)) \wedge \mathrm{weq}(s_3, s_4, 2, \mathrm{EQ}(x_2, \cdot) \vee \mathrm{EQ}(x_3, \cdot))$

$$\mathrm{weq}(a, b, 0, F(\cdot)) :\equiv a = b$$

$$\mathrm{weq}(a, b, m+1, F(\cdot)) :\equiv (a = b \vee F(\mathrm{diff}(a, b))) \wedge \mathrm{weq}(\ a\langle \mathrm{diff}(a, b) \lhd b[\mathrm{diff}(a, b)]\rangle,\ \ b, m, F(\cdot))$$

Weakeq-Ext Lemmas

Extended Array Lemma

$$(\forall\ i.\ a[i] = b[i]) \longrightarrow a = b \qquad \text{(ext)}$$

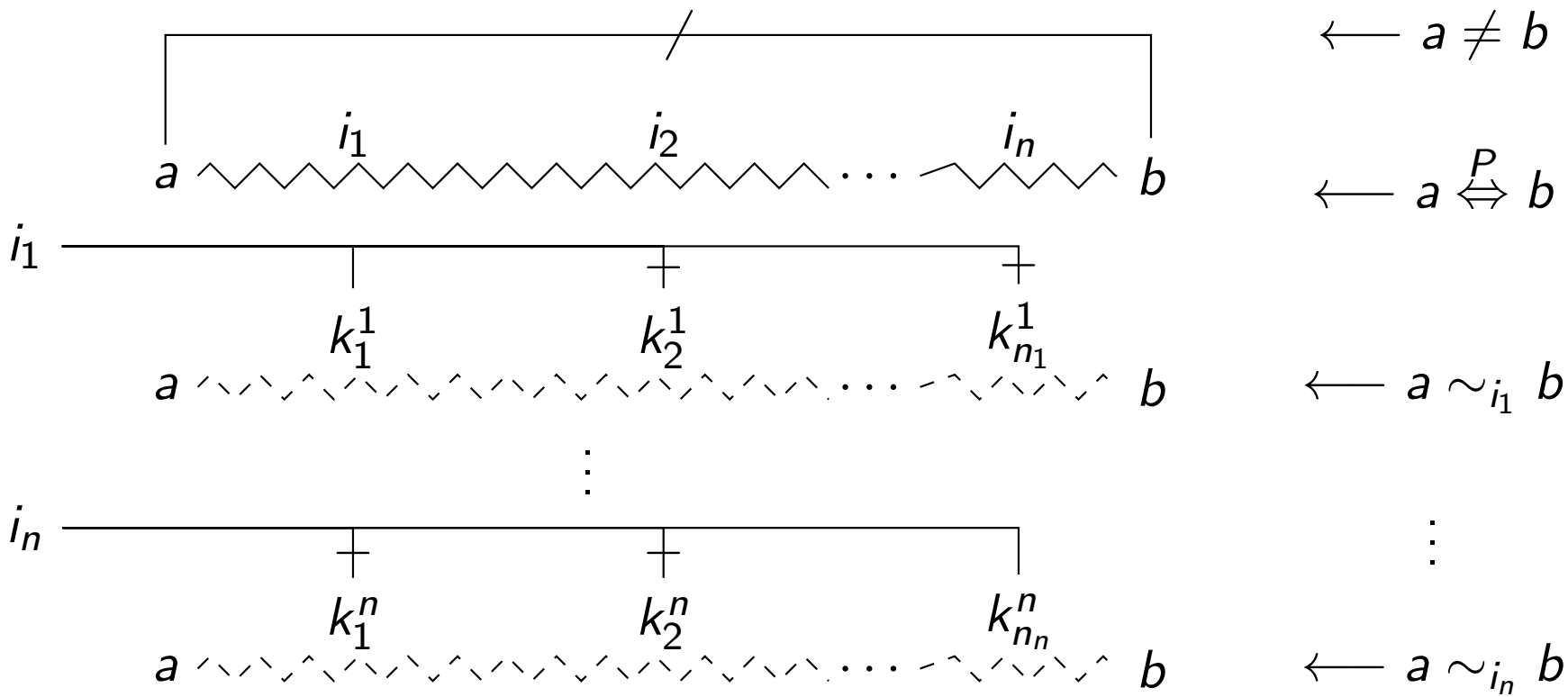
Weakeq-Ext Lemmas

Extended Array Lemma

$$a \overset{P}{\Leftrightarrow} b \wedge (\forall i \in \mathsf{Stores}(P).\ a \sim_i b) \longrightarrow a = b \qquad \text{(weakeq-ext)}$$

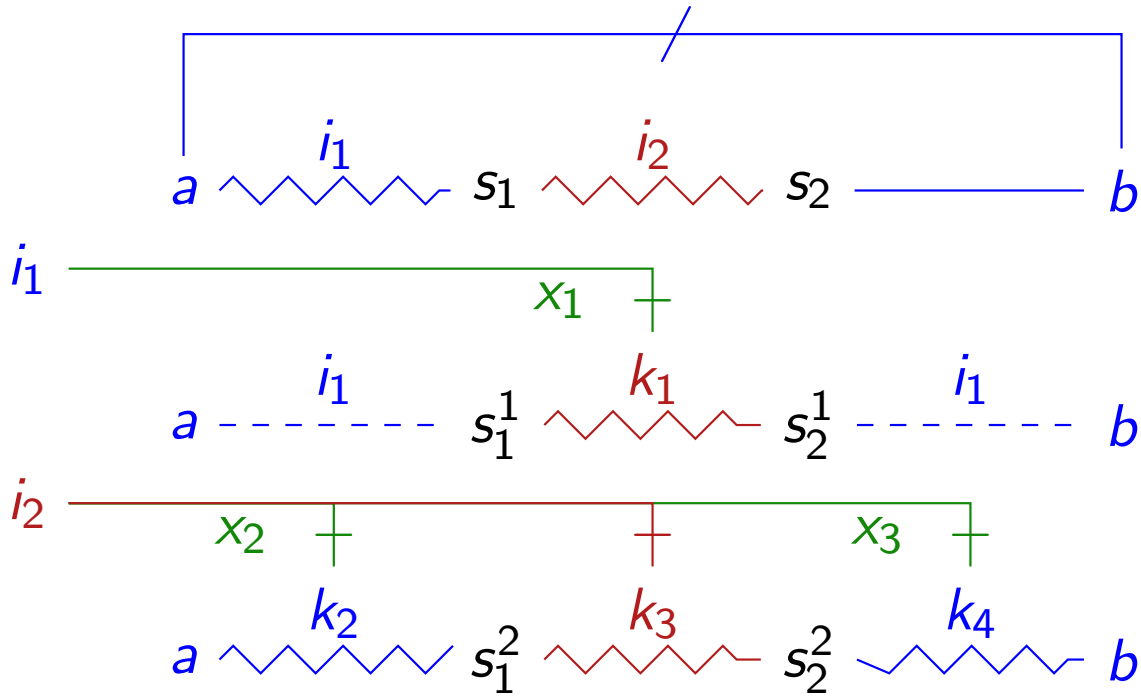
$a \overset{P}{\Leftrightarrow} b$: connected by a path of array equalities and writes
$a \sim_i b$: connected by a path of subpaths $s \approx_i s'$ and read equalities on i

Visualization of the corresponding conflict



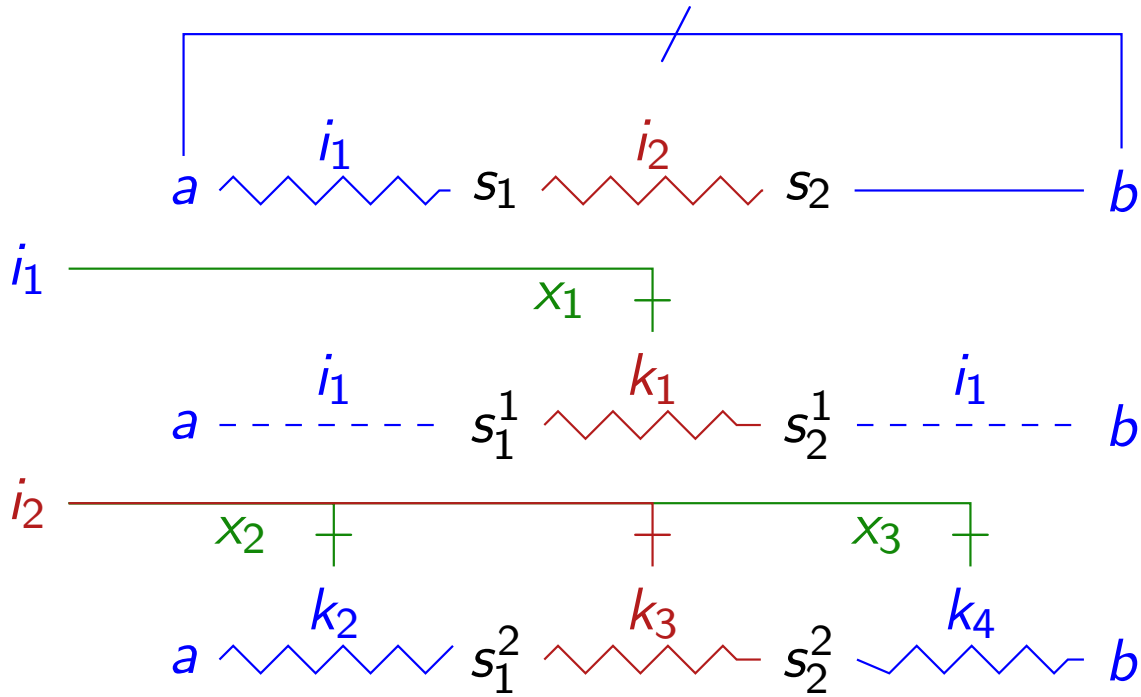
Interpolants for Weakeq-Ext

. . . with a and b in B



Interpolants for Weakeq-Ext

. . . with a and b in B



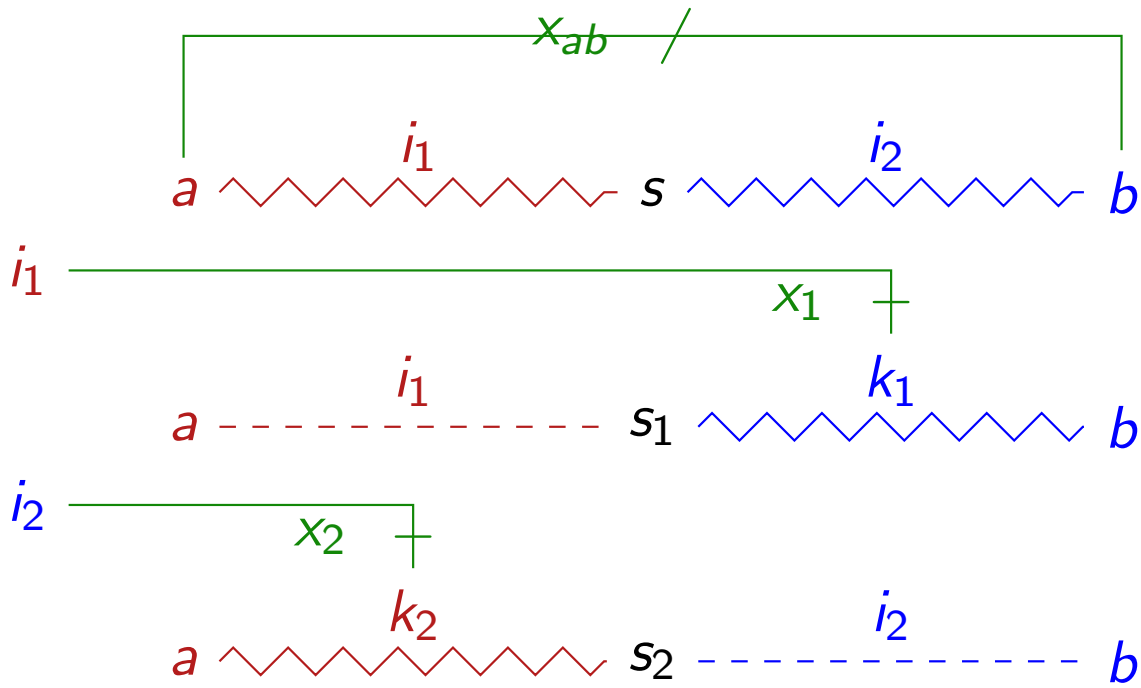
Interpolant

$$I : \mathrm{weq}(s_1^1, s_2^1, 1, \mathrm{EQ}(x_1, \cdot))$$

$$\wedge\, \mathrm{weq}(s_1, s_2, 1, \underbrace{\mathrm{EQ}(x_2, \cdot) \wedge s_1^2[\cdot] = s_2^2[\cdot] \wedge \mathrm{EQ}(x_4, \cdot)}_{\text{compare with read-over-weakeq}})$$

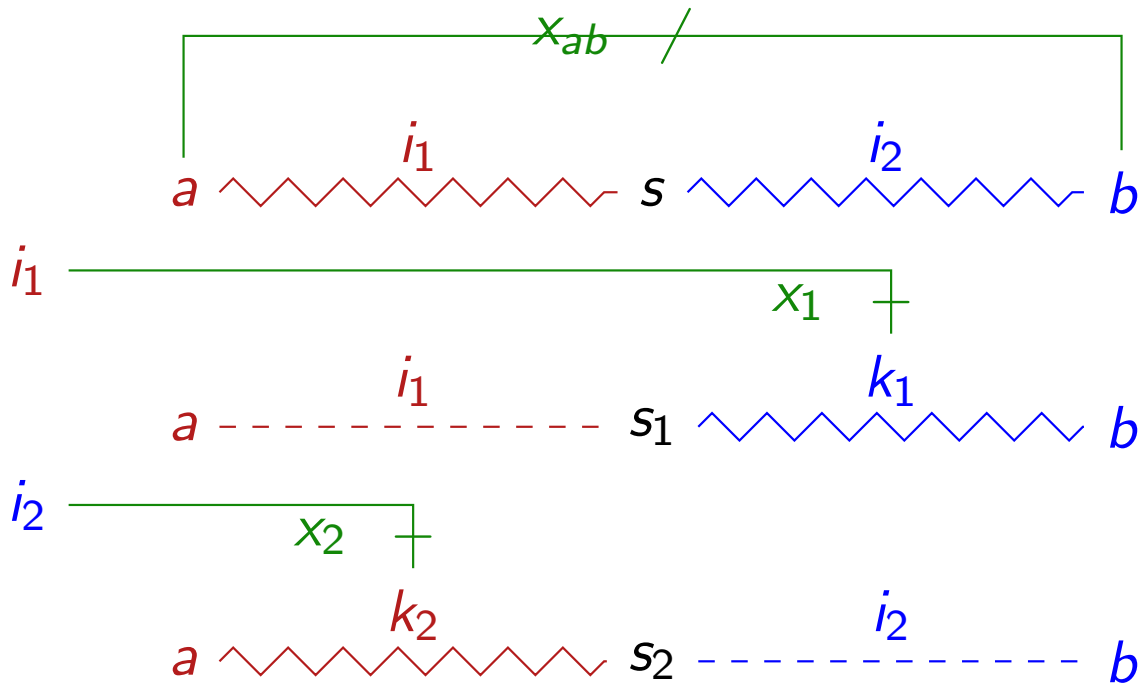
Interpolants for Weakeq-Ext

. . . with a in A and b in B



Interpolants for Weakeq-Ext

. . . with a in A and b in B



Interpolant

$$I : \mathrm{EQ}(x_{ab}, s) \wedge \mathrm{weq}(s, s_2, 1, \mathrm{EQ}(x_2, \cdot))$$

$$\vee\ \mathrm{nweq}\Big(s, s_1, 2, \underbrace{\mathrm{EQ}(x_{ab}, s\langle \cdot \triangleleft s_1[\cdot]\rangle)) \wedge \mathrm{weq}(s\langle \cdot \triangleleft s_1[\cdot]\rangle, s_2, 1, \mathrm{EQ}(x_2, :))}_{\text{compare with first line}} \wedge \mathrm{EQ}(x_1, \cdot)\Big)$$

Conclusion

Our interpolation method for the theory of arrays

- produces quantifier-free interpolants
- is designed for proof tree preserving interpolation
- reuses a graph produced during the proof
  based on weak equivalence between arrays
- is efficient for sequence and tree interpolants

Implementation in SMTInterpol is ongoing work.

Thank you for your attention!