Decision Procedures

Jochen Hoenicke

Software Engineering
Albert-Ludwigs-University Freiburg

Winter Term 2016/17

Craig Interpolation

Given an unsatisfiable formula of the form:

$$F \wedge G$$

Can we find a "smaller" formula that explains the conflict?

I.e., a formula implied by F that is inconsistent with G?

Under certain conditions, there is an interpolant I with

- $F \Rightarrow I$.
- $I \wedge G$ is unsatisfiable.
- I contains only symbols common to F and G.

Craig Interpolation

A craig interpolant I for an unsatisfiable formula $F \wedge G$ is

- $F \Rightarrow I$.
- $I \wedge G$ is unsatisfiable.
- I contains only symbols common to F and G.

Craig interpolants exists in many theories and fragments:

- First-order logic.
- Quantifier-free FOL.
- Quantifier-free fragment of T_{E}.
- Quantifier-free fragment of $T_{\mathbb{Q}}$.
- Quantifier-free fragment of $\widehat{T_{\mathbb{Z}}}$ (augmented with divisibility).

However, QF fragment of $T_{\mathbb{Z}}$ does not allow Craig interpolation.

Program correctness

Consider this path through LinearSearch:

$@\mathrm{pre}\ 0 \leq \ell \wedge u < |a|$

$i := \ell$

`assume` $i \leq u$

`assume` $a[i] \neq e$

$i := i + 1$

`assume` $i \leq u$

$@\ 0 \leq i \wedge i < |a|$

Single Static Assignment (SSA) replaces assignments by assumes:

$@\mathrm{pre}\ 0 \leq \ell \wedge u < |a|$

`assume` $i_1 = \ell$

`assume` $i_1 \leq u$

`assume` $a[i_1] \neq e$

`assume` $i_2 = i_1 + 1$

`assume` $i_2 \leq u$

$@\ 0 \leq i_2 \wedge i_2 < |a|$

Program correctness and Interpolants

If program contains only assumes, the VC looks like

$$VC\ :\ P \rightarrow (F_1 \rightarrow (F_2 \rightarrow (F_3 \rightarrow \ldots (F_n \rightarrow Q) \ldots)))$$

Using $\neg(F \rightarrow G) \Leftrightarrow F \land \neg G$ compute negation:

$$\neg VC\ :\ P \land F_1 \land F_2 \land F_3 \land \cdots \land F_n \land \neg Q$$

If verification condition is valid $\neg VC$ is unsatisfiable. We can compute interpolants for any program point, e.g. for

$$\color{red}{P \land F_1 \land F_2} \land \color{blue}{F_3 \land \cdots \land F_n \land \neg Q}$$

Consider the path through LinearSearch:

@pre $0 \leq \ell \wedge u < |a|$

assume $i_1 = \ell$

assume $i_1 \leq u$

assume $a[i_1] \neq e$

assume $i_2 = i_1 + 1$

assume $i_2 \leq u$

@ $0 \leq i_2 \wedge i_2 < |a|$

The negated VC is unsatisfiable:

$0 \leq \ell \wedge u < |a| \wedge i_1 = \ell$

$\wedge\ i_1 \leq u \wedge a[i_1] \neq e \wedge i_2 = i_1 + 1$

$\wedge\ i_2 \leq u \wedge (0 > i_2 \vee i_2 \geq |a|)$

The interpolant I for the red and blue part is

$$i_1 \geq 0 \wedge u < |a|$$

This is actually the loop invariant needed to prove the assertion.

Computing Interpolants

Suppose $F_1 \wedge F_n \wedge G_1 \wedge G_n$
How can we compute an interpolant?

- The algorithm is dependent on the theory and the fragment.
- We will show an algorithm for
  - Quantifier-free conjunctive fragment of T_{E}.
  - Quantifier-free conjunctive fragment of $T_{\mathbb{Q}}$.

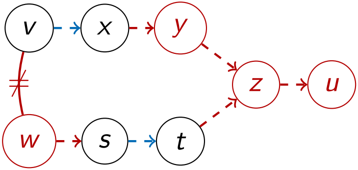
$$F_1 \land \cdots \land F_n \land G_1 \land \cdots \land G_n \text{ is unsat}$$

Let us first consider the case without function symbols.
The congruence closure algorithm returns unsat. Hence,

- there is a disequality $v \neq w$ and
- v, w have the same representative.

Example:

$v \neq w \land x = y \land y = z \land z = u \land w = s \land t = z \land s = t \land v = x$



The Interpolant "summarizes" the red edges: $I : v \neq s \land x = t$

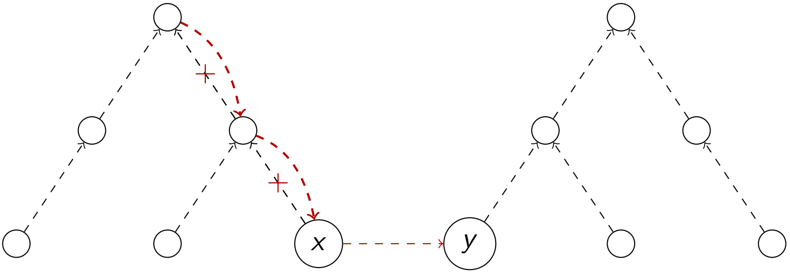
Edges in Congruence Closure Graph

Problem: Congruence closure graph draws edges between representatives instead of the equal terms. This makes finding the paths harder.

Solution: Change merge algorithm:

- Make one of the terms the representative by inverting edges to root
- Draw outgoing edge from the new representative to the equal term

Every term still has only one outgoing equality edge.



Given conjunctive formula:

$$F_1 \wedge \cdots \wedge F_n \wedge G_1 \wedge \cdots \wedge G_m$$

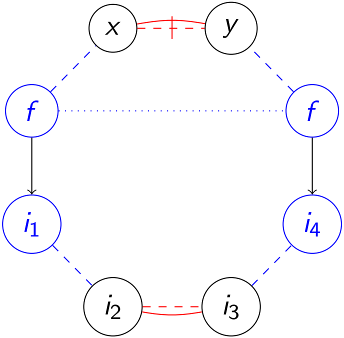
The following algorithm can be used unless there is a congruence edge:

- Build the congruence closure graph. Edges F_i are colored red, Edges G_j are colored blue.
- Add (colored) disequality edge. Find circle and remove all other edges.
- Combine maximal red paths, remove blue paths.
- The F paths start and end at shared symbols.
  Interpolant is the conjunction of the corresponding equalities.

Handling Congruence Edges (Case 1)

Both side of the congruence edge belong to G.

$$i_3 = i_2 \land x \neq y \land f(i_1) = x \land f(i_4) = y \land i_1 = i_2 \land i_3 = i_4$$



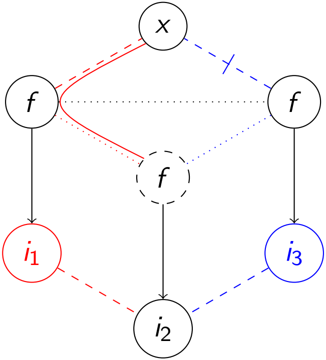
- Follow the path that connects the arguments.
- Also add summarized edges for that path.
- Treat the congruence edge as blue edge (ignore it).
- Interpolant is conjunction of all summarized paths.

Interpolant:
$i_2 = i_3 \land x \neq y$

Handling Congruence Edges (Case 2)

Both side of the congruence edge belong to different formulas.

$$f(i_1) = x \wedge i_2 = i_1 \wedge i_3 = i_2 \wedge f(i_3) \neq x$$

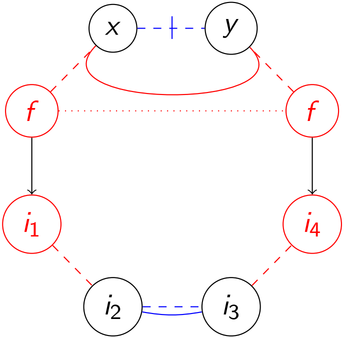


Interpolant: $x = f(i_2)$.

- Function symbol a must be shared.
- Follow the path that connects the arguments.
- Find first change from red to blue.
- Lift function application on that term.
- Summarize $x = f(i_1) \wedge i_1 = i_2$ by $x = f(i_2)$.
- Compute remaining interpolant as usual.

Both side of the congruence edge belong to F.

$$f(i_1) = x \wedge f(i_4) = y \wedge i_1 = i_2 \wedge i_3 = i_4 \wedge i_3 = i_2 \wedge x \neq y$$



Interpolant:
$i_2 = i_3 \rightarrow x = y$

- Follow the path that connects the arguments.
- Find the first and last terms i_2, i_3 where color changes.
- Treat congruence edge as red edge and summarize path.
- The summary only holds under $i_2 = i_3$,i.e., add $i_2 = i_3 \rightarrow x = y$ to interpolants.
- Summarize remaining path segments as usual.

Computing Interpolants for $T_{\mathbb{Q}}$

First apply Dutertre/de Moura algorithm.

- Non-basic variables $x_1, \dots, x_n$.
- Basic variables $y_1, \dots, y_m$.
- $y_i = \sum a_{ij} x_j$
- Conjunctive formula
  $y_1 \leq b_1 \dots y_{m'} \leq b_{m'} \wedge y_{m'+1} \leq b_{m'+1} \dots y_m \leq b_m$.

The algorithm returns unsatisfiable if and only if there is a line:

	x	$\cdots$	x	y	$\cdots$	y	y	$\cdots$	y
$\vdots$									
y_i/y_i	0	$\cdots$	0	$-/0$	$\cdots$	$-/0$	$-/0$	$\cdots$	$-/0$
$\vdots$									

$y_i = \sum -a'_k y_k$, $a'_k \geq 0$ and $\sum -a'_k b_k > b_i$
(the constraint $y_i \leq b_i$ is not satisfied)

Computing Interpolants for $T_{\mathbb{Q}}$

The conflict is:

$$b_i \geq y_i = \sum -a'_k y_k \geq \sum -a'_k b_k > b_i$$

or

$$0 = y_i + \sum a'_k y_k \leq b_i + \sum a'_k b_k < 0$$

We split the y variables into blue and red ones:

$$0 = \sum_{k=1}^{m'} a_{ik} y_k + \sum_{k=m'+1}^{m} a_{ik} y_k \leq \sum_{k=1}^{m'} a_{ik} b_k + \sum_{k=m'+1}^{m} a_{ik} b_k < 0$$

where $a'_k \geq 0, (a'_i = 1)$. The interpolant I is the red part:

$$\sum_{k=1}^{m'} a_{ik} y_k \leq \sum_{k=1}^{m'} a_{ik} b_k$$

where the basic variables y_k are replaced by their definition.

Example

$$x_1 + x_2 \leq 3 \land x_1 - x_2 \leq 1 \land x_3 - x_1 \leq 1 \land x_3 \geq 4$$

$$y_1 := x_1 + x_2 \qquad b_1 := 3 \qquad y_3 := -x_1 + x_3 \qquad b_3 := 1$$
$$y_2 := x_1 - x_2 \qquad b_1 := 1 \qquad y_4 := -x_3 \qquad b_4 := -4$$

Algorithm ends with the tableaux

	1	1	-4	
	y_2	y_3	y_4	β
y_1	-1	-2	-2	5
x_1	0	-1	-1	3
x_2	-1	-1	-1	2
x_3	0	0	-1	4

Conflict is $0 = y_1 + y_2 + 2y_3 + 2y_4 \leq 3 + 1 + 2 - 8 = -2$.
Interpolant is: $y_1 + y_2 \leq 3 + 1$
or (substituting non-basic vars): $2x_1 \leq 4$.

Correctness

$F_k : y_k := \sum_{j=0}^{n} a_{kj} x_j \leq b_k, (k=1,\dots,m)$ $\qquad$ $G_k : y_k := \sum_{j=0}^{n} a_{kj} x_j \leq b_k, (k=m',\dots,m)$

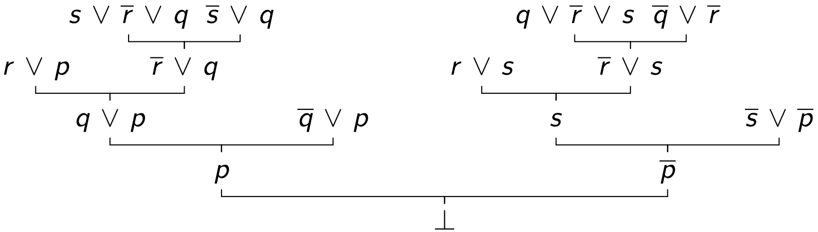
Conflict is $0 = \sum_{k=1}^{m'} a'_k y_k + \sum_{k=m'+1}^{m} a'_k y_k \leq \sum_{k=1}^{m'} a'_k b_k + \sum_{k=m'+1}^{m} a'_k b_k < 0$

After substitution the red part $\sum_{k=1}^{m'} a'_k y_k \leq \sum_{k=1}^{m'} a'_k b_k$ becomes

$$I : \sum_{j=1}^{n} \left(\sum_{k=1}^{m'} a'_k a_{kj} \right) x_j \leq \sum_{k=1}^{m'} a'_k b_k .$$

- $F \Rightarrow I$ (sum up the inequalities in F with factors a'_k).
- $I \wedge G \Rightarrow \bot$ (sum up I and G with factors a'_k to get $0 \leq \sum_{k=1}^{m} a'_k b_k < 0$).
- Only shared symbols in I: $0 = \sum_{k=1}^{m'} a_{kj} a'_k x_j + \sum_{k=m'+1}^{m} a_{kj} a'_k x_j$.
  If the left sum is not zero, the right sum is not zero and x_j appears in F and G.

A proof of unsatisfiability is a resolution tree:



where each node is generated by the rule

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

- The leaves are (trivial) consequences of $F \wedge G$.
- Therefore, every node is a consequence.
- Therefore, the root node $\perp$ is a consequence.

Interpolants for Conflict Clauses

Key Idea: Compute Interpolants for conflict clauses:
Split C into C_F and C_G (if literal appear in F and G put it in C_G).

The conflict clause follows from the original formula:

$$F \land G \Rightarrow C_F \lor C_G$$

Hence, the following formula is unsatisfiable.

$$F \land \neg C_F \land G \land \neg C_G$$

An interpolant I_C for C is the interpolant of the above formula. I_C contains only symbols shared between F and G.

McMillan's algorithm

Assign all literals to either F or G.

$s, \overline{r}, q : I_1 \quad \overline{s}, q : \top$

$q, \overline{r}, \overline{s} : I_1 \quad \overline{q}, \overline{r} : I_3$

$r, p : \bot \quad \overline{r}, q : I_1$

$r, s : s \quad \overline{r}, s : I_1 \wedge I_3$

$q, p : I_1 \quad \overline{q}, p : I_2$

$s : s \vee (I_1 \wedge I_3) \quad \overline{s}, \overline{p} : \overline{s}$

$p : I_1 \wedge I_2 \quad \overline{p} : I_1 \wedge I_3 \wedge \overline{s}$

$\top : (I_1 \wedge I_2) \vee (I_1 \wedge I_3 \wedge \overline{s})$

Compute interpolants for the leaves.
Then, for every resolution step compute interpolant as

$$\frac{\overline{\ell}_F \wedge \overline{C_1} : I_1 \qquad \ell_F \wedge \overline{C_2} : I_2}{\overline{C_1} \wedge \overline{C_2} : I_1 \vee I_2} \qquad \frac{\overline{\ell}_G \wedge \overline{C_1} : I_1 \qquad \ell_G \wedge \overline{C_2} : I_2}{\overline{C_1} \wedge \overline{C_2} : I_1 \wedge I_2}$$

Computing Interpolants for Conflict Clauses

There are several points where conflict clauses are returned:

- Conflict clauses is returned by TCHECK.
  Then theory must give an interpolant.
- Conflict clauses comes from F.
  Then $F \Rightarrow C_F \lor C_G$.
  Hence, $(F \land \neg C_F) \Rightarrow C_G$. Also, $C_G \land G \land \neg C_G$ is unsatisfiable
  Interpolant is C_G.
- Conflict clauses comes from G.
  Then $C_G = C$, $G \Rightarrow C_G$.
  Hence, $(G \land \neg C_G)$ is unsatisfiable. Interpolant is $\top$.
- Conflict clause comes from resolution on ℓ.
  Then there is a unit clause $U = \ell \lor U'$ with interpolant I_U
  and conflict clause $C = \neg\ell \lor C'$ with interpolant I_C.

  If $\ell \in F$, set $I_{U' \lor C'} = I_U \lor I_C$
  If $\ell \in G$, set $I_{U' \lor C'} = I_U \land I_C$

Computing Interpolants for DPLL(T)

The previous algorithm can compute interpolant for each conflict clause.
The final conflict clause returned is $\bot$.
$I_\bot$ is an interpolant of $F \wedge G$.

Unfortunately, it is not that easy...
... because equalities shared by Nelson-Oppen can contain red and blue symbols simultaneously.

Example:

$$F\colon t \le 2a \land 2a \le s \land f(a) = q$$
$$G\colon s \le 2b \land 2b \le t \land f(b) \ne q$$

Nelson-Oppen proof

Purifying the example gives:

$$\Gamma_E : f(a) = q \land f(b) \neq q$$
$$\Gamma_{\mathbb{Q}} : t \leq 2a \land 2a \leq s \land s \leq 2b \land 2b \leq t$$

Shared variables $V = \{a, b\}$
Nelson-Oppen proceeds as follows

1. $\Gamma_{\mathbb{Q}}$ propagates $a = b$.
2. $\Gamma_E \cup a = b$ is unsatisfiable.

$$\Gamma_E : f(a) = q \wedge f(b) \neq q$$
$$\Gamma_{\mathbb{Q}} : t \leq 2a \wedge 2a \leq s \wedge s \leq 2b \wedge 2b \leq t$$

N-O introduces three literals: $a = b$, $a \leq b$, $a \geq b$.
Theory conflicts:

$$2b \leq t \wedge t \leq 2a \wedge \neg(b \leq a)$$
$$2a \leq s \wedge s \leq 2b \wedge \neg(a \leq b)$$
$$a \leq b \wedge b \leq a \wedge a \neq b$$
$$a = b \wedge f(a) = q \wedge f(b) \neq q$$

How can we compute interpolants for the conflicts?

Interpolant with $a = b$

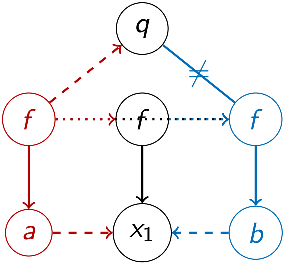
What is an interpolant of $a = b \wedge f(a) = q \wedge f(b) \neq q$?

Key Idea: Split

$$a = b$$

into

$$a = x_1 \wedge x_1 = b \text{ where } x_1 \text{ shared}$$



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

Interpolant: $f(x_1) = q$

Interpolant with $a \neq b$

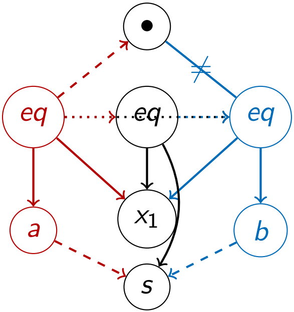
What is an interpolant of $a \neq b \wedge a = s \wedge b = s$?

Key Idea: Split

$$a \neq b$$

into

$$eq(x_1, a) \wedge \neg eq(x_1, b) \text{ where } x_1 \text{ shared, } eq \text{ a predicate}$$



$eq(x_1, a) = \bullet \wedge a = s \wedge$
$eq(x_1, b) \neq \bullet \wedge b = s$

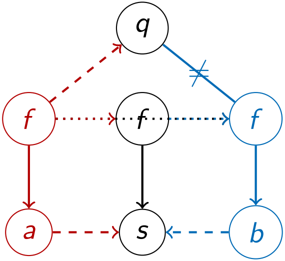
Interpolant: $eq(x_1, s)$

Resolving on $a = b$

Consider the resolution step

$$\frac{a = b \lor a \neq s \lor b \neq s \qquad a \neq b \lor f(a) \neq q \lor f(b) = q}{f(a) \neq q \lor f(b) = q \lor a \neq s \lor b \neq s}$$

How to combine the interpolants $eq(x_1, s)$ and $f(x_1) = q$?



$f(a) = q \land a = s \land$
$f(b) \neq q \land s = b$

Interpolant: $f(s) = q$

$eq(x_1, s)$ indicates that x_1 should be replaced by s.

Resolution rule for $a = b$

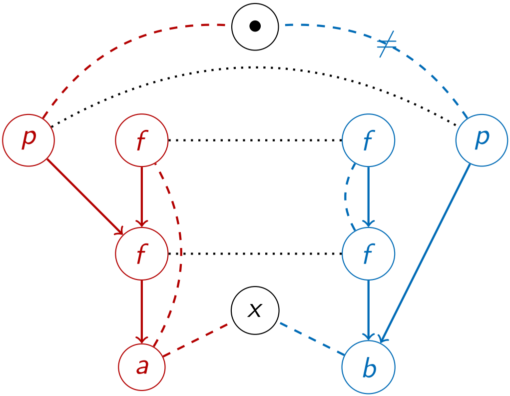
The interpolation rule is

$$\frac{a = b \lor C_1 : I_1[eq(x, s_1)] \ldots [eq(x, s_n)] \qquad a \neq b \lor C_2 : I_2(x)}{C_1 \lor C_2 : I_1[I_2(s_1)] \ldots [I_2(s_n)]}$$

In our example

$$\frac{\begin{array}{c}\neg(a \neq b \land a = s \land b = s) : eq(x_1, s) \\ \neg(a = b \land f(a) = q \land f(b) \neq q) : q = f(x_1)\end{array}}{\neg(f(a) = q \land f(b) \neq q \land a = s \land b = s) : q = f(s)}$$

$$a = f(f(a)) \land a = x \land p(f(a)) \land b = x \land f(b) = f(f(b)) \land \neg p(b)$$



$$a = f(f(a)) \land a = x \land p(f(a)) \land b = x \land f(b) = f(f(b)) \land \neg p(b)$$

Prove using the following lemmas:

$$
\begin{array}{rl}
F_1 : & a = x \land x = b \rightarrow f(a) =_{x_1} f(b) : eq(x_1, f(x)) \\
F_2 : & f(a) =_{x_1} f(b) \rightarrow f(f(a)) =_{x_2} f(f(b)) : eq(x_2, f(x_1)) \\
F_3 : & f(a) =_{x_1} f(b) = f(f(b)) =_{x_2} \\
& f(f(a)) = a = x = b \rightarrow f(a) =_{x_3} b : eq(x_3, x_1) \land x_2 = x \\
F_4 : & f(a) =_{x_3} b \land p(f(a)) \rightarrow p(b) : p(x_3)
\end{array}
$$

Example: Annotating Proof with Interpolants

$$F_1 : \quad a = x \land x = b \rightarrow f(a) =_{x_1} f(b) \colon eq(x_1, f(x))$$

$$F_2 : \quad f(a) =_{x_1} f(b) \rightarrow f(f(a)) =_{x_2} f(f(b)) \colon eq(x_2, f(x_1))$$

$$F_3 : \quad f(a) =_{x_1} f(b) = f(f(b)) =_{x_2} f(f(a)) = a = x = b \rightarrow f(a) =_{x_3} b \colon eq(x_3, x_1) \land x_2 = x$$

$$F_4 : \quad f(a) =_{x_3} b \land p(f(a)) \rightarrow p(b) \colon p(x_3)$$

$$F_2 : eq(x_2, f(x_1)) \qquad F_3 : eq(x_3, x_1) \land x_2 = x$$

$$F_1 : eq(x_1, f(x)) \qquad eq(x_3, x_1) \land f(x_1) = x$$

$$eq(x_3, f(x)) \land f(f(x)) = x \qquad F_4 : p(x_3)$$

$$p(f(x)) \land f(f(x)) = x$$

Example: Checking Interpolants

$$a = f(f(a)) \land a = x \land p(f(a)) \land b = x \land f(b) = f(f(b)) \land \neg p(b)$$

Interpolant: $p(f(x)) \land f(f(x)) = x$

- $F \rightarrow I$: Substitute $a = x$ into other atoms.
- $I \land G \rightarrow \bot$: $b = x \land f(f(x)) = x \land \neg p(b)$ implies $\neg p(f(f(x)))$.
  With $b = x$, $f(b) = f(f(b))$ this implies $\neg p(f(x))$.
  This contradicts $p(f(x))$.
- Symbol condition: p, f, x are shared.

Back to the Nelson–Oppen Example

$$\Gamma_E : f(a) = q \wedge f(b) \neq q$$
$$\Gamma_{\mathbb{Q}} : t \leq 2a \wedge 2a \leq s \wedge s \leq 2b \wedge 2b \leq t$$

Theory conflicts:

$$2b \leq t \wedge t \leq 2a \wedge \neg(b \leq a)$$
$$2a \leq s \wedge s \leq 2b \wedge \neg(a \leq b)$$
$$a \leq b \wedge b \leq a \wedge a \neq b$$
$$a = b \wedge f(a) = q \wedge f(b) \neq q$$

How can we compute interpolants for the conflicts?

Interpolant with $a > b$

What is an interpolant of $2a \leq s \wedge s \leq 2b \wedge a > b$

Split

$$a > b$$

into

$$a \geq x_1 \wedge x_1 > a \text{ where } x_1 \text{ shared}$$

$$\begin{array}{rcll} 2a - s & \leq & 0 & \cdot 1 \\ s - 2b & \leq & 0 & \cdot 1 \\ x_1 - a & \leq & 0 & \cdot 2 \\ b - x_1 & < & 0 & \cdot 2 \\ \hline 0 & < & 0 & \end{array} \qquad \begin{array}{rcll} 2a - s & \leq & 0 & \cdot 1 \\ x_1 - a & \leq & 0 & \cdot 2 \\ \hline 2x_1 - s & \leq & 0 & \end{array}$$

Interpolant: $2x_1 - s \leq 0$.
We need the term $2x_1 - s$ later; we write interpolant as:

$$LA(2x_1 - s, 2x_1 - s \leq 0)$$

Interpolant with $a < b$

What is an interpolant of $t \leq 2a \wedge 2b \leq t \wedge a < b$

Split

$$a < b$$

into

$$a \leq x_2 \wedge x_2 < b \text{ where } x_2 \text{ shared}$$

$$\begin{array}{rcll} t - 2a & \leq & 0 & \cdot 1 \\ 2b - t & \leq & 0 & \cdot 1 \\ a - x_2 & \leq & 0 & \cdot 2 \\ x_2 - b & < & 0 & \cdot 2 \\ \hline 0 & < & 0 & \end{array} \qquad \begin{array}{rcll} t - 2a & \leq & 0 & \cdot 1 \\ a - x_2 & \leq & 0 & \cdot 2 \\ \hline t - 2x_2 & \leq & 0 & \end{array}$$

Interpolant: $t - 2x_2 \leq 0$.
We need the term $t - 2x_2$ later; we write interpolant as:

$$LA(t - 2x_2, t - 2x_2 \leq 0)$$

What is an interpolant of $a \le b \wedge b \le a \wedge a \neq b$

$$a \le x_1 \wedge x_2 \le a \wedge eq(x_3, a) \wedge x_1 \le b \wedge b \le x_2 \wedge \neg eq(x_3, b)$$

Manually we find the interpolant

$$x_2 - x_1 < 0 \vee (x_2 - x_1 \le 0 \wedge eq(x_3, x_2))$$

Here $x_2 - x_1$ is the "critical term"; Interpolant:

$$LA(x_2 - x_1, x_2 - x_1 < 0 \vee (x_2 - x_1 \le 0 \wedge eq(x_3, x_2)))$$

Magic rule:

$$\frac{a \leq b \vee C_1 : LA(s_1 + c_1 x_1, F_1(x_1)) \quad a > b \vee C_2 : LA(s_2 - c_2 x_1, F_2(x_2))}{C_1 \vee C_2 : LA(c_2 s_1 + c_1 s_2, c_2 s_1 + c_1 s_2 < 0 \vee (F_1(s_2/c_2) \wedge F_2(s_2/c_2)))}$$

Example:

$$\frac{\begin{array}{r} a \leq b \vee 2a > s \vee s > 2b : LA(2x_1 - s, 2x_1 - s \leq 0) \\ a > b \vee a < b \vee a = b : LA(x_2 - x_1, x_2 - x_1 < 0 \vee \\ (x_2 - x_1 \leq 0 \wedge eq(x_3, x_2))) \end{array}}{a < b \vee a = b \vee 2a > s \vee s > 2b : I_3}$$

$$I_3 : LA(2x_2 - s, 2x_2 - s < 0 \vee (2x_2 - s \leq 0 \wedge eq(x_3, x_2)))`$$

(simplifying $x_2 < x_2$ to $\perp$ and $x_2 \leq x_2$ to $\top$).

Magic rule:

$$\frac{a \leq b \vee C_1 : LA(s_1 + c_1 x_1, F_1(x_1)) \quad a > b \vee C_2 : LA(s_2 - c_2 x_1, F_2(x_2))}{C_1 \vee C_2 : LA(c_2 s_1 + c_1 s_2, c_2 s_1 + c_1 s_2 < 0 \vee (F_1(s_2/c_2) \wedge F_2(s_2/c_2)))}$$

$$\frac{\begin{array}{r} a < b \vee a = b \vee 2a > s \vee s > 2b : LA(2x_2 - s, 2x_2 - s < 0 \vee \\ (2x_2 - s \leq 0 \wedge eq(x_3, x_2))) \\ a \geq b \vee t < 2a \vee 2b < s : LA(t - 2x_1, t - 2x_1 \leq 0) \end{array}}{\begin{array}{r} a = b \vee 2a > s \vee s > 2b \\ \vee t > 2a \vee t > 2b : I_4 \end{array}}$$

$$I_4 : LA(t - s, t - s < 0 \vee (t - s \leq 0 \wedge eq(x_3, t/2)))$$

The critical term $t - s$ does not contain an auxiliary and can be removed.

$$I_4 : t - s < 0 \vee (t - s \leq 0 \wedge eq(x_3, t/2))$$

Example continued (with equality)

$$\frac{\begin{array}{rcl} a = b \lor 2a > s \lor s > 2b \\ \lor\, t > 2a \lor t > 2b & : & \begin{array}{l} t - s < 0 \lor \\ (t - s \leq 0 \land eq(x_3, t/2)) \end{array} \\ a \neq b \lor f(a) \neq q \lor f(b) = q & : & q = f(x_3) \end{array}}{\begin{array}{rcl} 2a > s \lor s > 2b \\ \lor\, t > 2a \lor t > 2b & : & \begin{array}{l} t - s < 0 \lor \\ (t - s \leq 0 \land q = f(t/2)) \end{array} \\ \lor\, f(a) \neq q \lor f(b) = q \end{array}}$$

The interpolant of

$$2a \leq s \land t \leq 2a \land f(a) = q \land s \leq 2b \land 2b \leq t \land f(b) \neq q$$

is

$$t - s < 0 \lor (t - s \leq 0 \land q = f(t/2))$$