Decision Procedures

Jochen Hoenicke

Software Engineering
Albert-Ludwigs-University Freiburg

Winter Term 2019/2020

Craig Interpolation

A decision procedure for satisfiability has two possible outcomes:

- satisfiable, model: valuation for uninterpreted symbols
- unsatisfiable, with proof

Is there something simpler than a proof?

Introduction

Given an unsatisfiable conjunction of two formulas:

$$F \wedge G \text{ is unsatisfiable,}$$

i.e.,

$$F \Rightarrow \neg G$$

Can we find a "small" formula that explains this?

A formula implied by F that implies $\neg G$?

Under certain conditions, there is an interpolant I with

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

A Craig interpolant I for an unsatisfiable formula $F \wedge G$ is a formula s.t.

- $F \Rightarrow I$.
- $I \wedge G$ 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).
- Quantifier-free fragment of $\widehat{T_{\mathsf{A}}^{=}}$ (augmented with difference).

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

Motivation: Program correctness

Consider this path through LinearSearch:

$$@\text{pre } 0 \le \ell \land u < |a|$$
$$i := \ell$$
$$\texttt{assume } i \le u$$
$$\texttt{assume } a[i] \ne e$$
$$i := i + 1$$
$$\texttt{assume } i \le u$$
$$@\ 0 \le i \land i < |a|$$

Single Static Assignment (SSA) replaces assignments by assumes:

$$@\text{pre } 0 \le \ell \land u < |a|$$
$$\texttt{assume } i_1 = \ell$$
$$\texttt{assume } i_1 \le u$$
$$\texttt{assume } a[i_1] \ne e$$
$$\texttt{assume } i_2 = i_1 + 1$$
$$\texttt{assume } i_2 \le u$$
$$@\ 0 \le i_2 \land i_2 < |a|$$

The program contains only assumes. Therefore, the VC is

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

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

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

Verification Condition and Interpolants

Consider the path through LinearSearch:

$$
\begin{array}{l}
\texttt{@pre } 0 \leq \ell \land u < |a| \\
\texttt{assume } i_1 = \ell \\
\texttt{assume } i_1 \leq u \\
\texttt{assume } a[i_1] \neq e \\
\texttt{assume } i_2 = i_1 + 1 \\
\texttt{assume } i_2 \leq u \\
\texttt{@ } 0 \leq i_2 \land i_2 < |a|
\end{array}
$$

The negated VC is unsatisfiable:

$$
\begin{array}{l}
0 \leq \ell \land u < |a| \land i_1 = \ell \\
\land\ i_1 \leq u \land a[i_1] \neq e \land i_2 = i_1 + 1 \\
\land\ i_2 \leq u \land (0 > i_2 \lor i_2 \geq |a|)
\end{array}
$$

The interpolant I for the red and blue part is

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

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

Computing Interpolants

Given an unsatisfiable conjunction $F_1 \wedge F_n \wedge G_1 \wedge G_n$.

How can we compute an interpolant?
Answer: it depends on the theory fragment.

We will show an algorithm for

- Quantifier-free conjunctive fragment of T_{E}.
- Quantifier-free conjunctive fragment of $T_{\mathbb{Q}}$.

Computing Interpolants for T_E

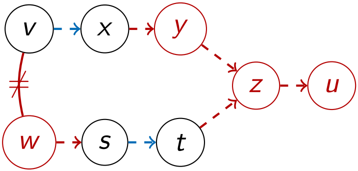
$$F_1 \wedge \cdots \wedge F_n \wedge G_1 \wedge \cdots \wedge G_n \text{ is unsat}$$

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 are connected by equality or congruence edges.

Example

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



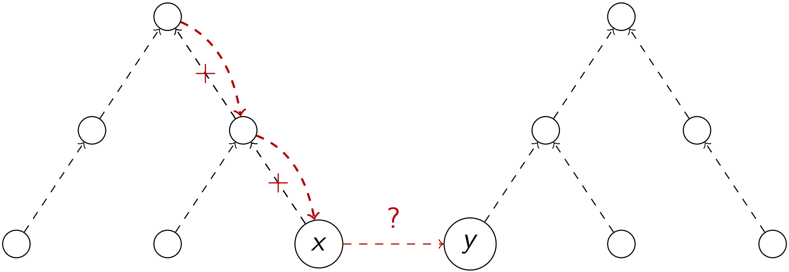
Disequality: $v \neq w$
Equality chain:

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

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 directly 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:

- Build the congruence closure graph.
- Find the disequality $s \neq t$ that contradicts equalities.
- Find the path from s to t in the equality graph and add a disequality edge from s to t to close circle.
- For each congruence, find the path between the arguments.
- Color edges from F_i red, and edges from G_j blue.
  Color congruence red if it connects two terms from $F_1 \ldots F_n$.
- Remove all blue paths.
- Summarize each of the remaining red components.
- Interpolant is the conjunction of summaries.

Summarize Components

A sequence of equalities is summarized by a single equality between end points:

$$x = y = z = t \text{ has summary } x = t$$

If a sequence contains the single disequality, the summary is a disequality

$$s = w \neq v \text{ has summary } s \neq v$$

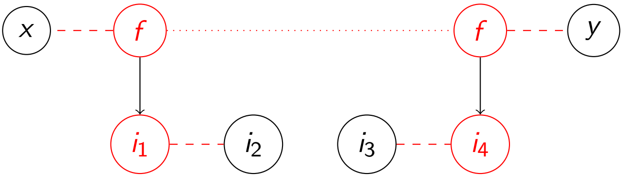
If the whole cycle is in A, the summary is $\perp$.

$$s = w \neq v = x = y = z = t = s \text{ has summary } \perp$$

Case 1: The congruence edge is colored red.

- For each argument path with a gap, add a disequality between the endpoints (where the gap is).
- Summarize the component as if the congruence was an equality.
- The summary is the disjunction of the above formulas.

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

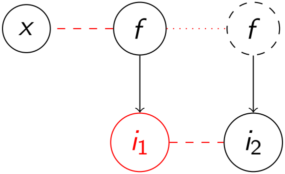


Summary:
$i_2 \neq i_3 \lor x = y$

Case 2: The congruence edge is not colored red.

- For each argument, find the endpoint of the corresponding path.
- Apply the function to the end point and connect with a red edge.
- Summarize as usual, ignoring the partial argument paths.

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



Summary: $x = f(i_2)$.

Example

$$F : f(g(x)) = y \land x \neq y$$
$$G : x = f(z) \land x = f(g(f(z)))$$

Example

$$F : y = x \wedge x \neq f(g(y)) \wedge f(x) = w$$
$$G : x = w \wedge x = z \wedge g(z) = x$$

First apply Dutertre/de Moura algorithm.

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

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

$$
\begin{array}{c|ccccccccc}
 & x & \cdots & x & y & \cdots & y & y & \cdots & y \\
\hline
\vdots & & & & & & & & & \\
y_i/y_i & 0 & \cdots & 0 & -/0 & \cdots & -/0 & -/0 & \cdots & -/0 \\
\vdots & & & & & & & & &
\end{array}
$$

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

The conflict is:

$$b_i \geq y_i = \sum -c_k y_k \geq \sum -c_k b_k > b_i$$

or

$$0 = y_i + \sum c_k y_k \leq b_i + \sum c_k b_k < 0$$

We split the y variables into blue and red ones:

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

where $c_k \geq 0, (c_i = 1)$. The interpolant I is the red part:

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

where the basic variables y_k are replaced by their definition.

Example

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

$$\begin{array}{llll} y_1 := x_1 + x_2 & b_1 := 3 & y_3 := -x_1 + x_3 & b_3 := 1 \\ y_2 := x_1 - x_2 & b_1 := 1 & y_4 := -x_3 & b_4 := -4 \end{array}$$

Algorithm ends with the tableau

	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

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

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

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

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

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

Given the input:

$$F : (p \lor r) \land (\overline{p} \lor q)$$
$$G : (\overline{q} \lor r \lor s) \land (\overline{r} \lor s) \land (\overline{q} \lor \overline{s}) \land (q \lor \overline{r} \lor \overline{s})$$

$\langle \epsilon, F \land G, \top \rangle \xrightarrow{\text{Decide}} \langle r^{\square}, F \land G, \top \rangle \xrightarrow{\text{Propagate}} \langle r^{\square} s^{\overline{r} \lor s}, F \land G, \top \rangle \xrightarrow{\text{Propagate}}$
$\langle r^{\square} s^{\overline{r} \lor s} \overline{q}^{\overline{q} \lor \overline{s}}, F \land G, \top \rangle \xrightarrow{\text{Conflict}} \langle r^{\square} s^{\overline{r} \lor s} \overline{q}^{\overline{q} \lor \overline{s}}, F \land G, q \lor \overline{r} \lor \overline{s} \rangle \xrightarrow{\text{Explain}}$
$\langle r^{\square} s^{\overline{r} \lor s} \overline{q}^{\overline{q} \lor \overline{s}}, F \land G, \overline{r} \lor \overline{s} \rangle \xrightarrow{\text{Explain}} \langle r^{\square} s^{\overline{r} \lor s} \overline{q}^{\overline{q} \lor \overline{s}}, F \land G, \overline{r} \rangle \xrightarrow{\text{Learn}}$
$\langle r^{\square} s^{\overline{r} \lor s} \overline{q}^{\overline{q} \lor \overline{s}}, F \land G \land \overline{r}, \overline{r} \rangle \xrightarrow{\text{Back}} \langle \overline{r}^{\overline{r}}, F \land G \land \overline{r}, \top \rangle \xrightarrow{\text{Propagate}}$
$\langle \overline{r}^{\overline{r}} p^{p \lor r}, F \land G \land \overline{r}, \top \rangle \xrightarrow{\text{Propagate}} \langle \overline{r}^{\overline{r}} p^{p \lor r} q^{\overline{p} \lor q}, F \land G \land \overline{r}, \top \rangle \xrightarrow{\text{Propagate}}$
$\langle \overline{r}^{\overline{r}} p^{p \lor r} q^{\overline{p} \lor q} s^{\overline{q} \lor r \lor s}, F \land G \land \overline{r}, \top \rangle \xrightarrow{\text{Conflict}} \langle \overline{r}^{\overline{r}} p^{p \lor r} q^{\overline{p} \lor q} s^{\overline{q} \lor r \lor s}, F \land G \land$
$\overline{r}, \overline{q} \lor \overline{s} \rangle \xrightarrow{\text{Explain}} \langle \overline{r}^{\overline{r}} p^{p \lor r} q^{\overline{p} \lor q} s^{\overline{q} \lor r \lor s}, F \land G \land \overline{r}, \overline{q} \lor r \rangle \xrightarrow{\text{Explain}}$
$\langle \overline{r}^{\overline{r}} p^{p \lor r} q^{\overline{p} \lor q} s^{\overline{q} \lor r \lor s}, F \land G \land \overline{r}, \overline{p} \lor r \rangle \xrightarrow{\text{Explain}}$
$\langle \overline{r}^{\overline{r}} p^{p \lor r} q^{\overline{p} \lor q} s^{\overline{q} \lor r \lor s}, F \land G \land \overline{r}, r \rangle \xrightarrow{\text{Explain}} \langle \overline{r}^{\overline{r}} p^{p \lor r} q^{\overline{p} \lor q} s^{\overline{q} \lor r \lor s}, F \land G \land \overline{r}, \bot \rangle$

Computing Interpolants for DPLL(T)

Every explain step is a resolution step

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

These can be put together to a proof tree:

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{q} \lor \overline{s} \quad \overline{q} \lor r \lor s}{\overline{q} \lor r} \qquad \overline{p} \lor q}{\overline{p} \lor r} \qquad p \lor r}{r} \qquad \dfrac{\dfrac{q \lor \overline{r} \lor \overline{s} \qquad \overline{q} \lor \overline{s}}{\overline{r} \lor \overline{s}} \qquad \overline{r} \lor s}{\overline{r}}}{\bot}$$

- The leaves are input clauses from $F \land G$.
- Every clause is a consequence of $F \land G$.
- The root node $\bot$ shows that $F \land G$ is unsat.

Interpolants for Intermediate Clauses

Key Idea: Compute interpolants for an intermediate clause C:
Split C into C_F and C_G (if literal appears in F and G put it in C_G).

The conflict clause follows from the original formula:

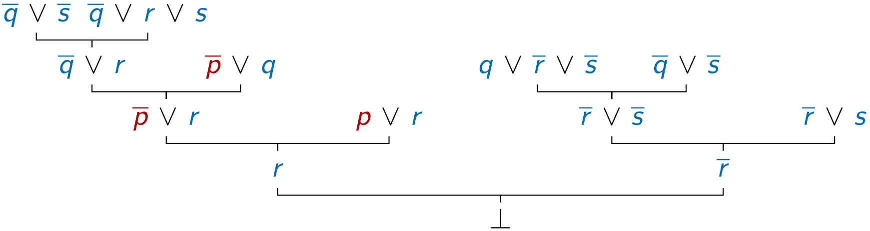
$$F \land G \Rightarrow C_F \lor C_G$$

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.

Color literals occuring only in F red and all others blue.



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 Leafs

- Clause comes from F.
  Then $F \Rightarrow C_F \vee C_G$.
  Hence, $(F \wedge \neg C_F) \Rightarrow C_G$. Also, $C_G \wedge G \wedge \neg C_G$ is unsatisfiable
  Interpolant is C_G.
- Clause comes from G.
  Then $C_G = C$, $G \Rightarrow C_G$.
  Hence, $(G \wedge \neg C_G)$ is unsatisfiable. Interpolant is $\top$.
- Clause is generated by TConflict.
  Then theory must give an interpolant.

Example: McMillan's algorithm

Interpolation for resolution rule:

$$\frac{\overline{\ell_F} \land \overline{C_1} \,:\, I_1 \qquad \ell_F \land \overline{C_2} \,:\, I_2}{\overline{C_1} \land \overline{C_2} \,:\, I_1 \lor I_2} \qquad\qquad \frac{\overline{\ell_G} \land \overline{C_1} \,:\, I_1 \qquad \ell_G \land \overline{C_2} \,:\, I_2}{\overline{C_1} \land \overline{C_2} \,:\, I_1 \land I_2}$$

Interpolation Example:

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{q}, \overline{s} : \top \quad \overline{q}, r, s : \top}{\overline{q}, r : \top} \quad \overline{p}, q : q}{\overline{p}, r : q} \quad p, r : r}{r : q \lor r} \qquad \dfrac{\dfrac{q, \overline{r}, \overline{s} : \top \quad \overline{q}, \overline{s} : \top}{\overline{r}, \overline{s} : \top} \quad \overline{r}, s : \top}{\overline{r} : \top}}{\bot : q \lor r}$$