Verification of linear hybrid systems: Symbolic representations using simple interpolants

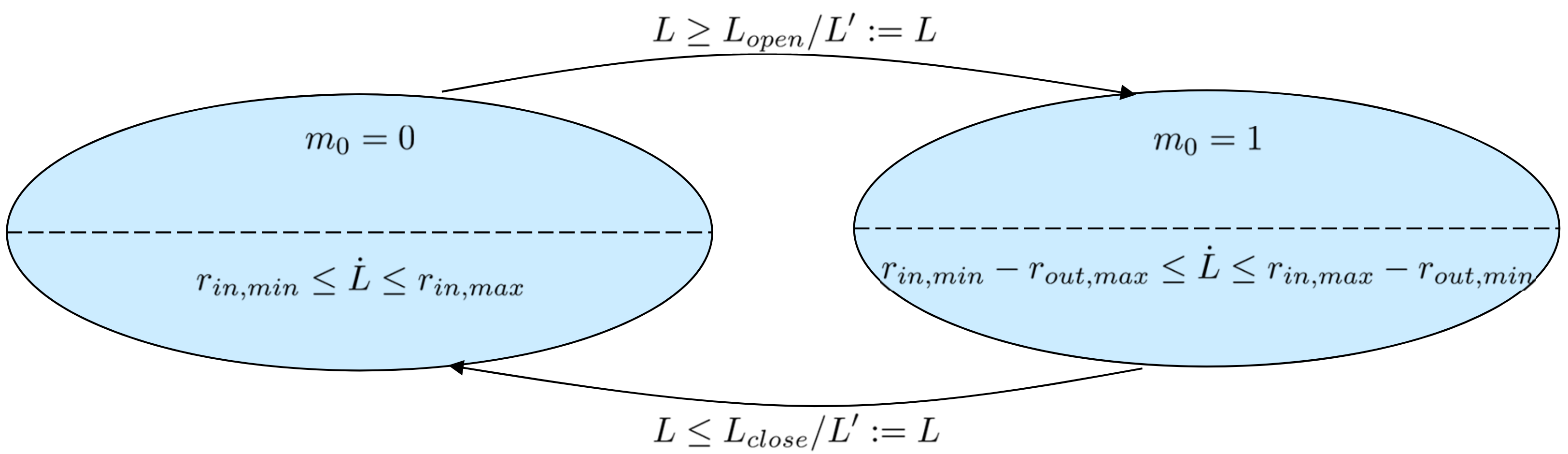
Christoph Scholl

Albert-Ludwigs-University Freiburg

Thanks to Florian Pigorsch, Stefan Disch, Ernst Althaus, Werner Damm, Uwe Waldmann, …

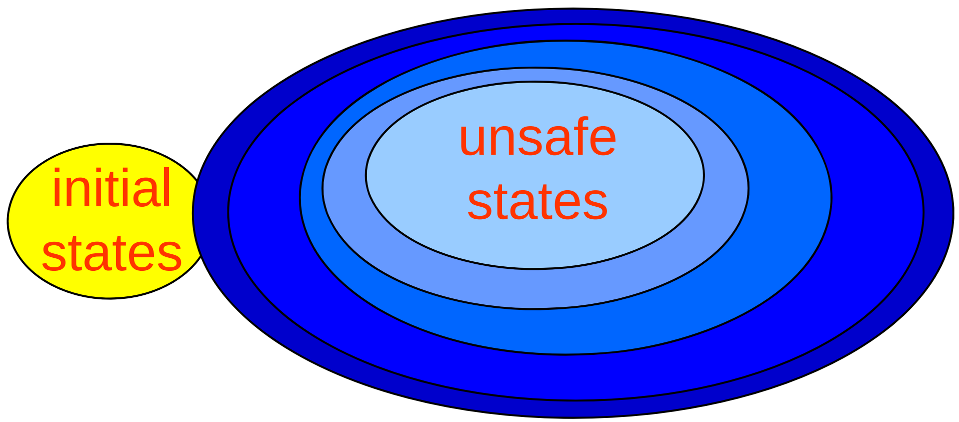
Background: LinAIG Based Model Checking

- Given:
  - Hybrid system with dynamics restricted to differential inclusions
  - Intended application domain: Hybrid systems with a large number of discrete states
  - Safety specification
  - Initial states



Background: LinAIG Based Model Checking

- Approach:
  - Backward model checking from unsafe states
  - Symbolic representation of sets of states by LinAIGs
    (= AND-Inverter-Graphs with linear constraints)
  - Preimage computation until initial states or fixed point reached

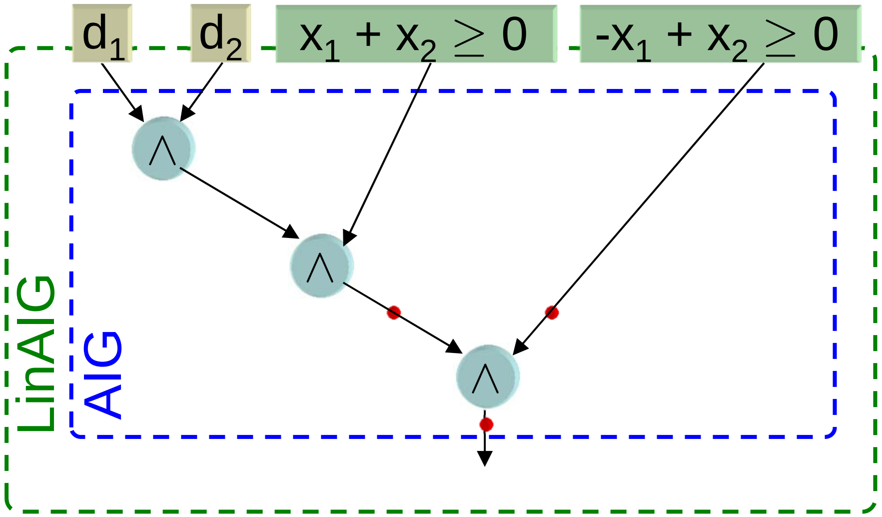


Background: LinAIGs

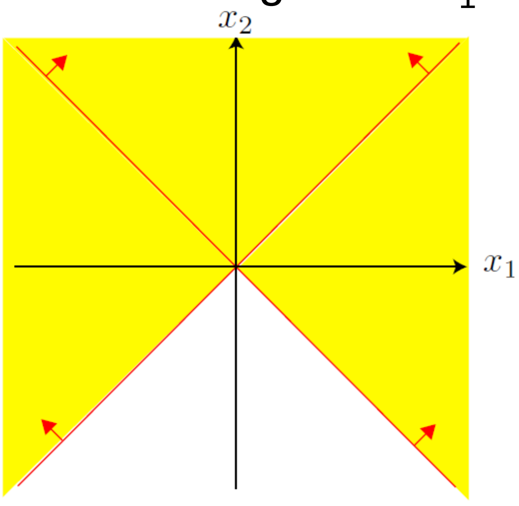
- Sets of states are represented by
  - Arbitrary Boolean combinations of Boolean variables $d_1,\ldots, d_n$ and linear constraints over real-valued variables $x_1,\ldots, x_m$

- **Example**: $(d_1 \wedge d_2) \wedge (x_1 + x_2 \geq 0) \vee (- x_1 + x_2 \geq 0)$

LinAIG:

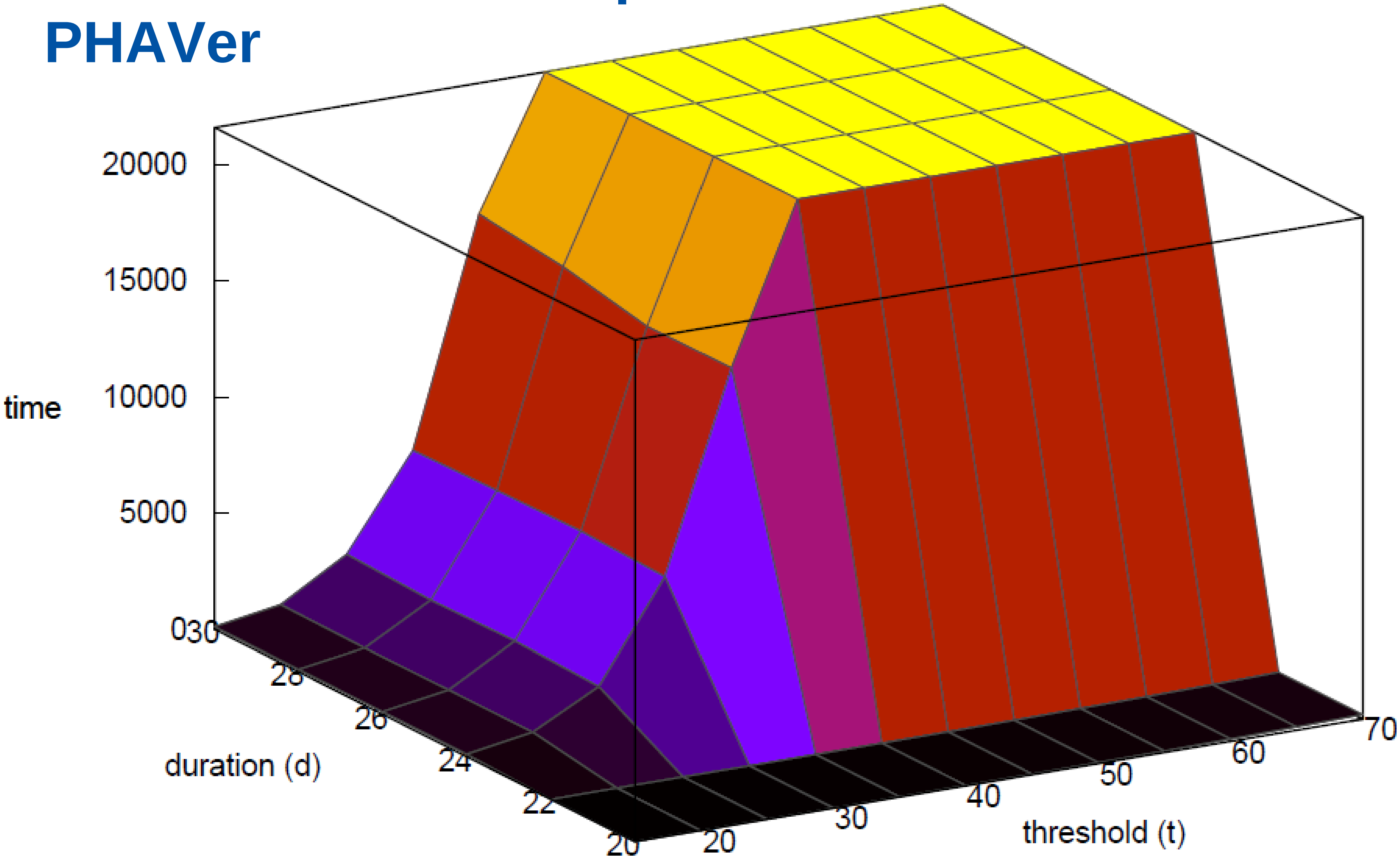


Represented region for $d_1 = d_2 = 0$:

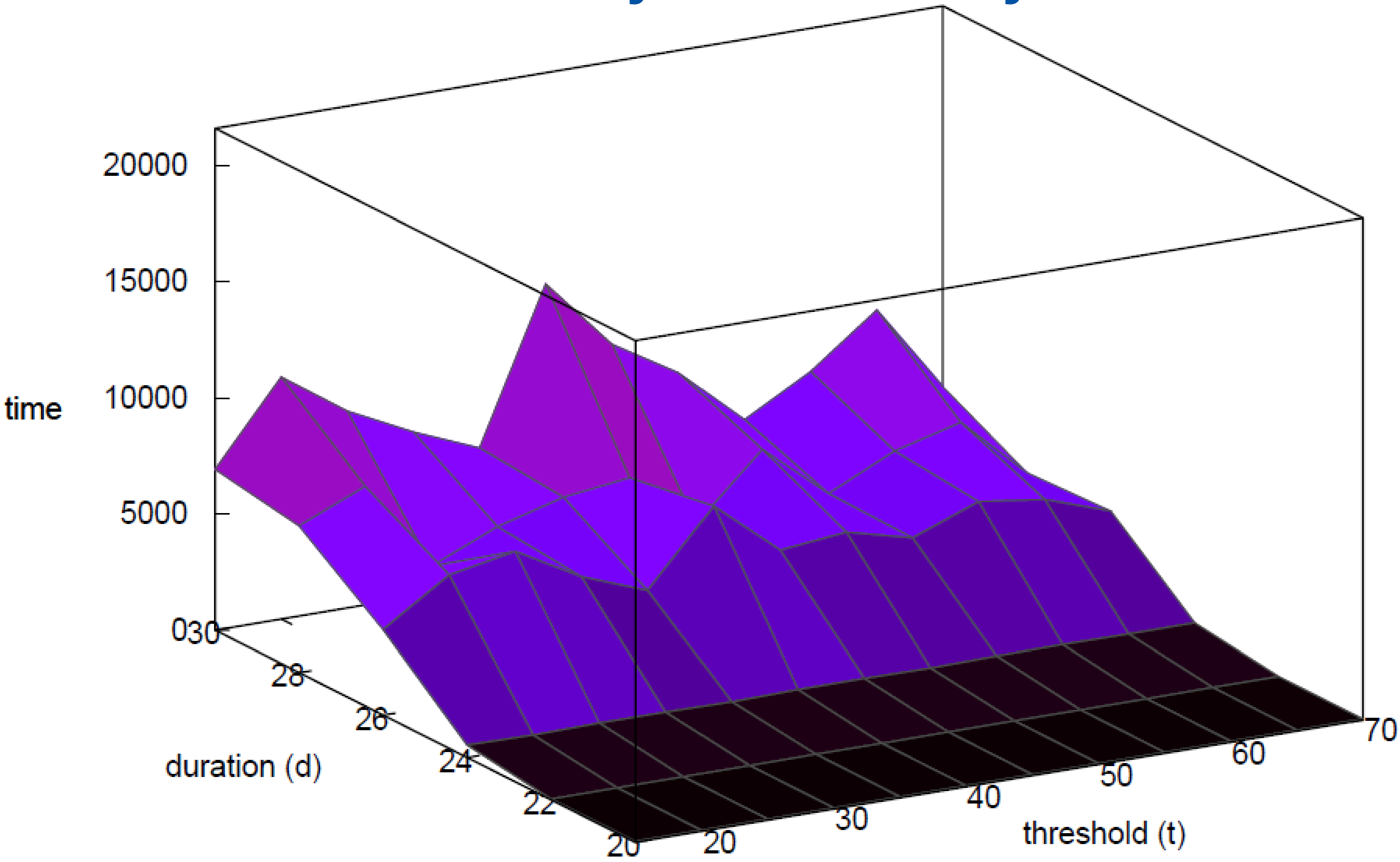


- Representations may be optimized by several techniques including „Redundancy Removal", „Constraint Minimization"

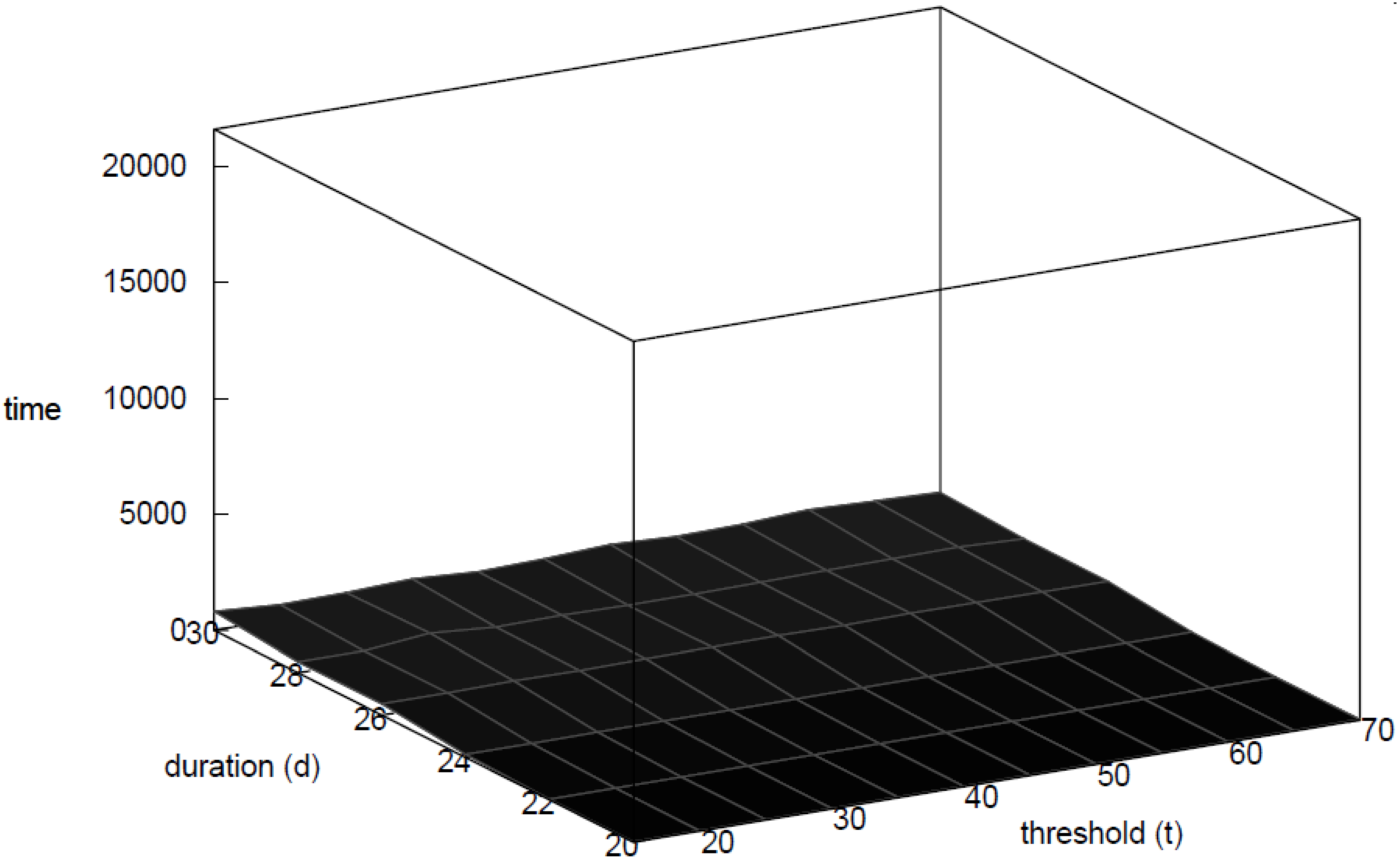
Parameterized Example: Dam
PHAVer



FOMC with redudancy removal only



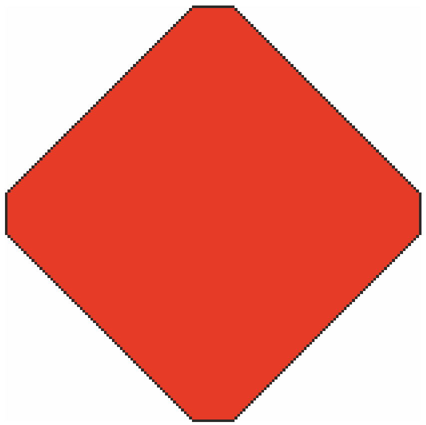
FOMC with constraint minimization

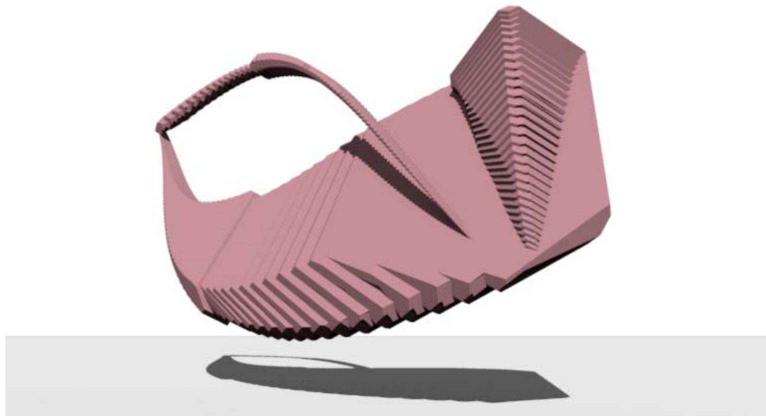


Motivation (1)

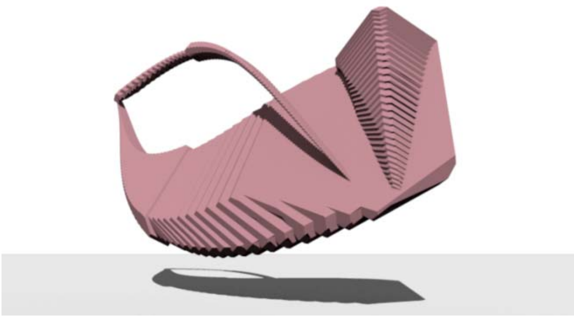
- Our current state set compaction techniques
  - Do not change the computed sets of unsafe states
    $\Rightarrow$ exact model checking
  - Make use only of already existing linear constraints for state set representation

- Problem: Sometimes the boundary of the represented region is really complicated





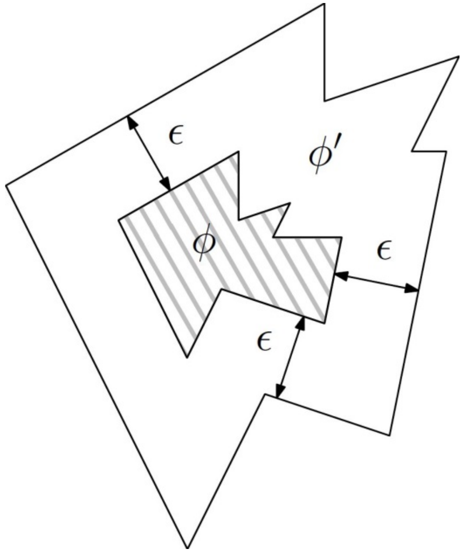
Motivation (2)



- Goal:
  - Replace complicated state sets by „smoother“ representations
  - Introduce (restricted) over-approximations
- It is important to have the complete picture (i.e. the complete state set) to be able to judge which over-approximation makes sense.
- As usual:
  - If safety can be proved using over-approximations, everything is fine.
  - Otherwise: Counterexample-guided abstraction refinement

Method

- Allow the state set to expand into an ϵ-environment of the current state set

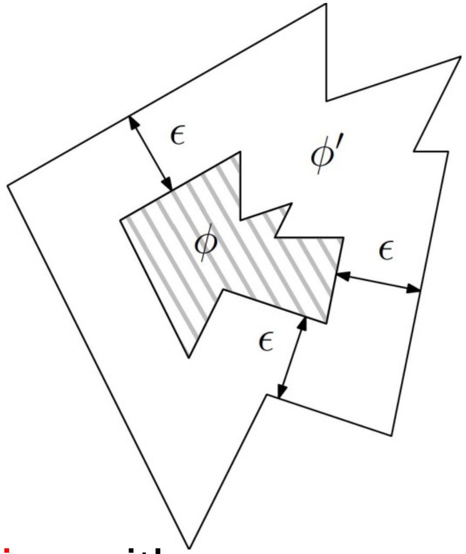


Craig Interpolation

- A Craig Interpolant for two formulas A and B with $A \wedge B = 0$ is a formula I with
  - $A \Rightarrow I$
  - $I \wedge B = 0$
  - The uninterpreted symbols in I occur both in A and B as well as the free variables in I occur freely both in A and B

Method

- Allow the state set to expand into an ϵ-environment of the current state set

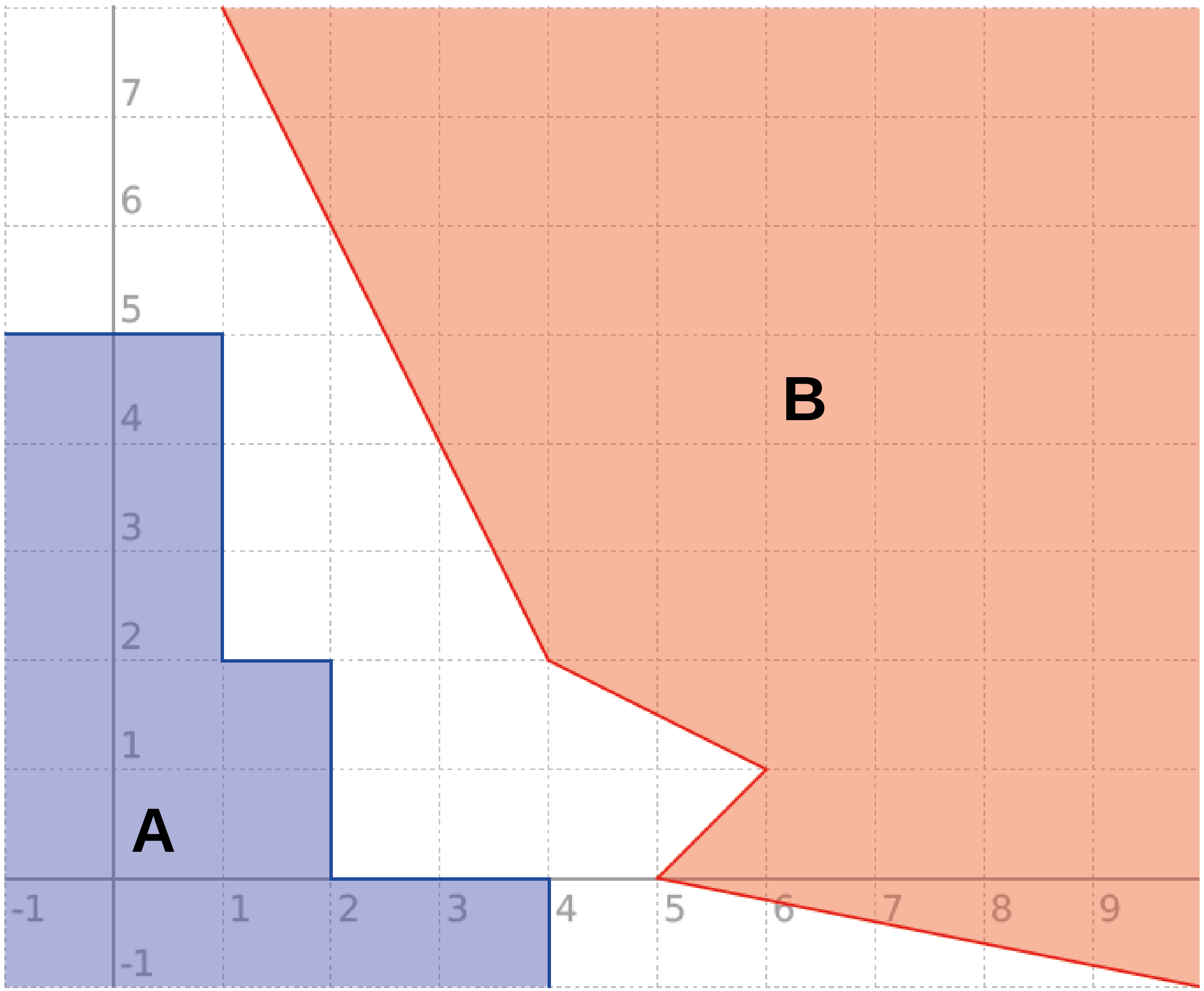


- $\Rightarrow$ Craig Interpolation with
  - Current state set as A
  - Negation of (current state set + ϵ-environment + other „don‘t cares“) as B
  - $A \wedge B = 0$

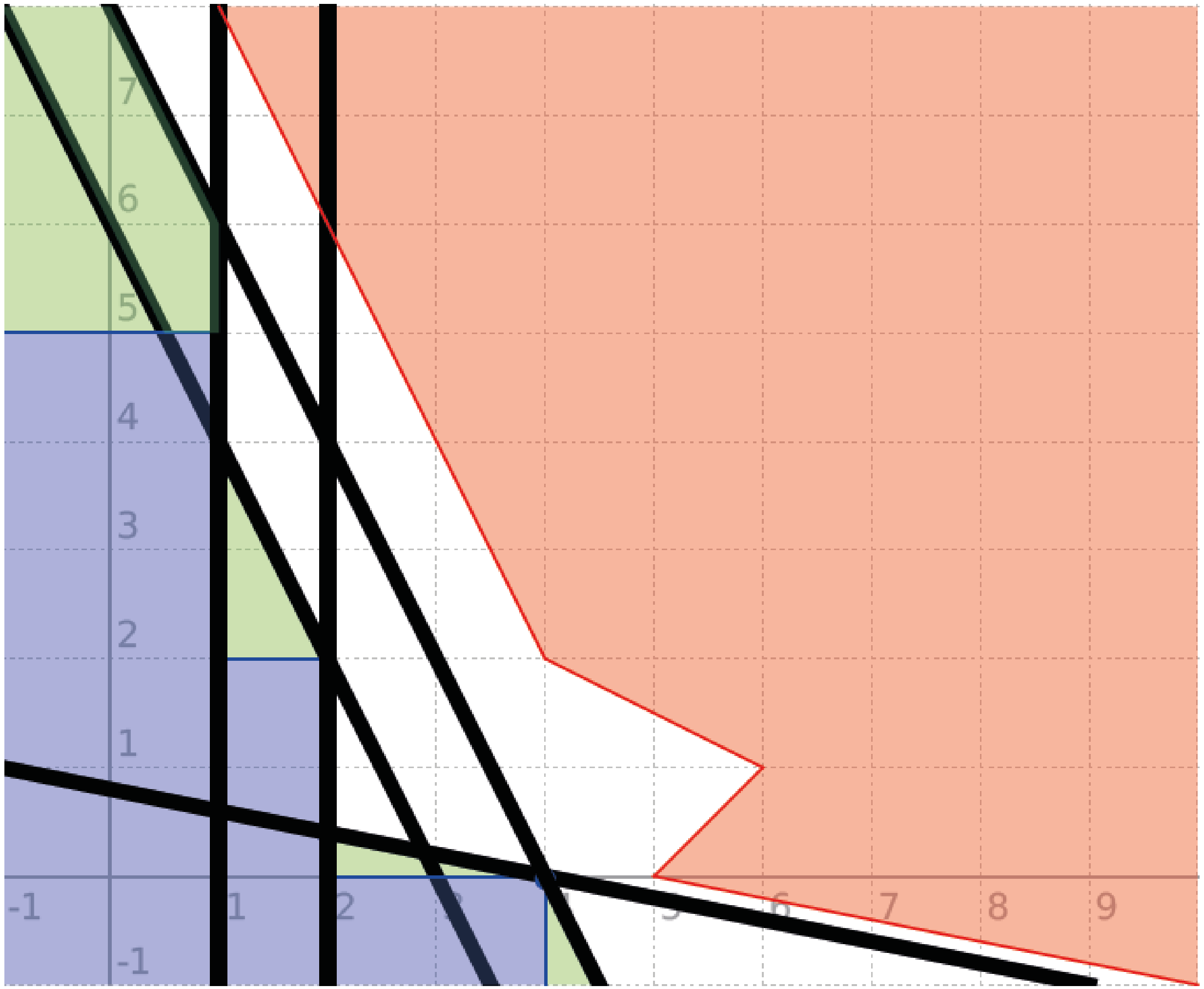
$\Rightarrow$ Craig interpolant I with $A \Rightarrow I$, $I \wedge B = 0$

- Thus we need simple interpolants!

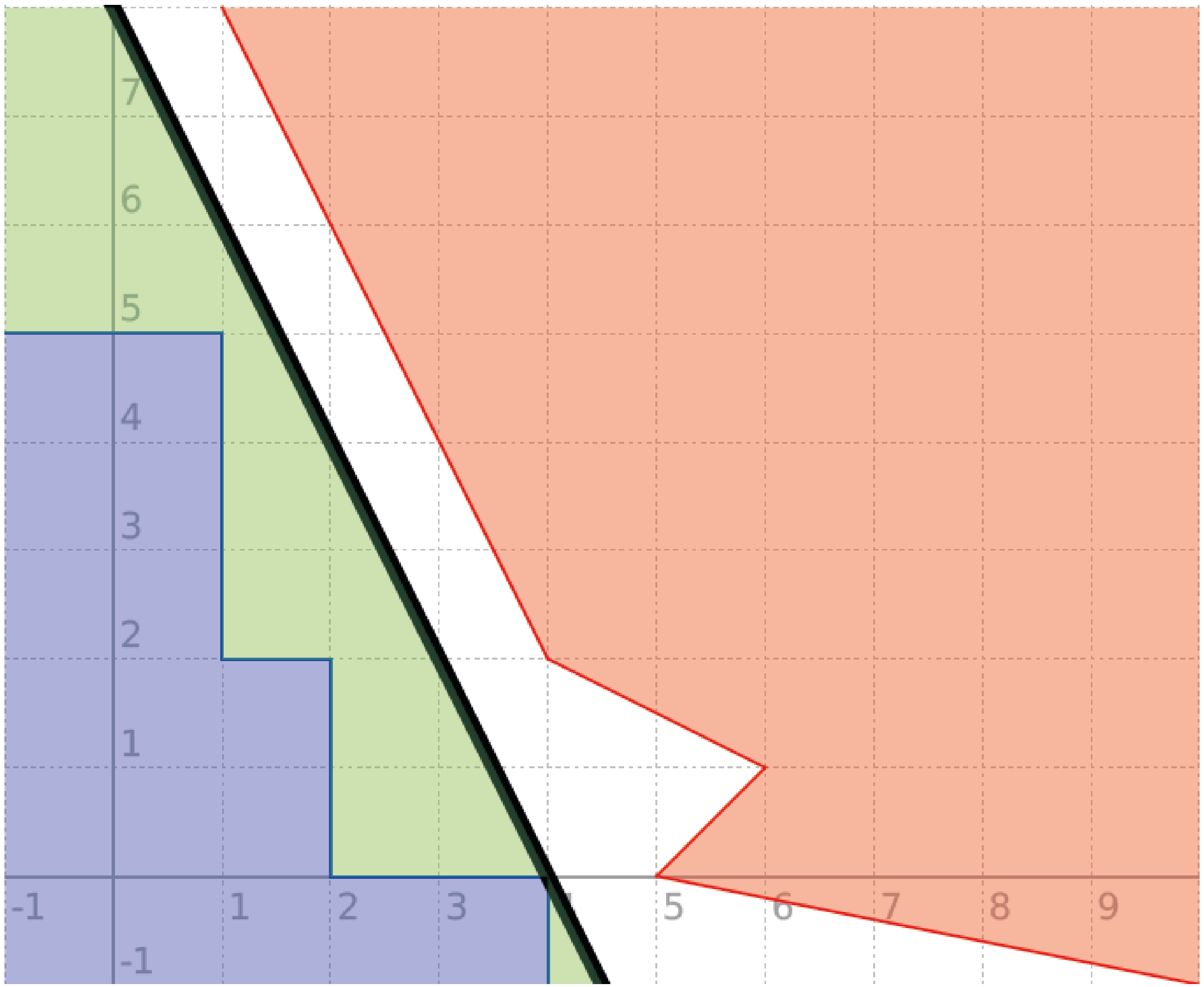
Interpolation example computed by MathSAT



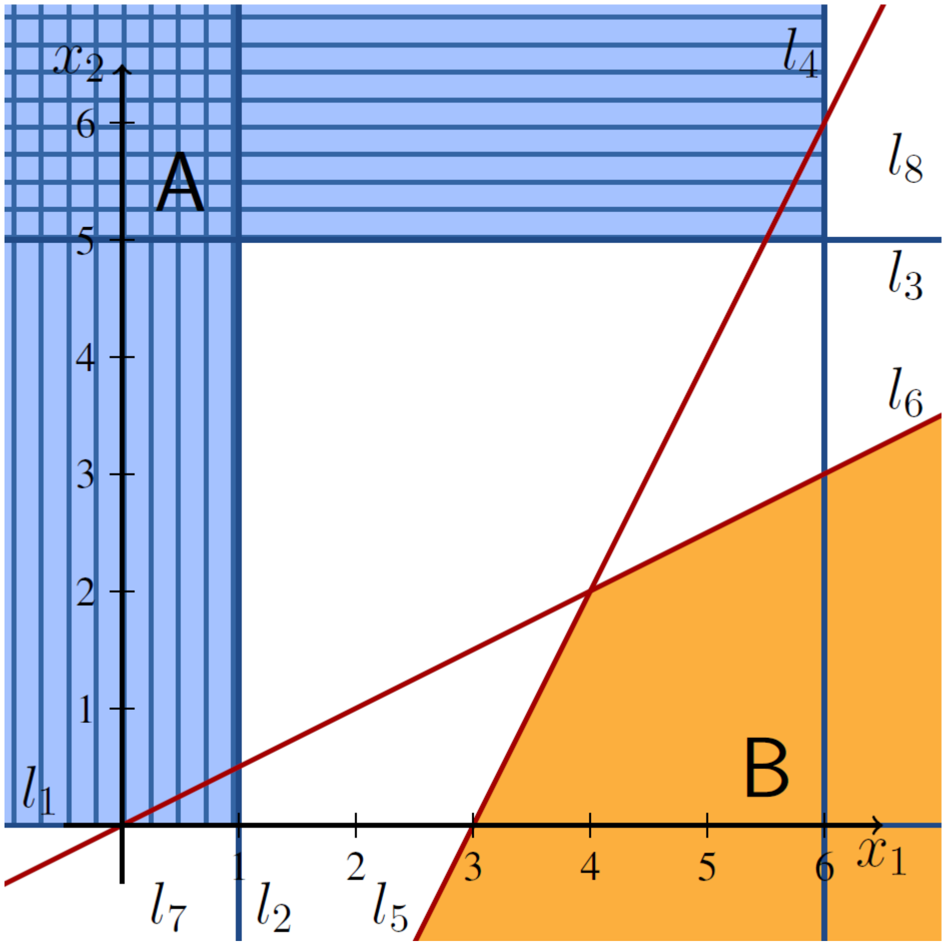
Interpolation example computed by MathSAT



Another possible solution …



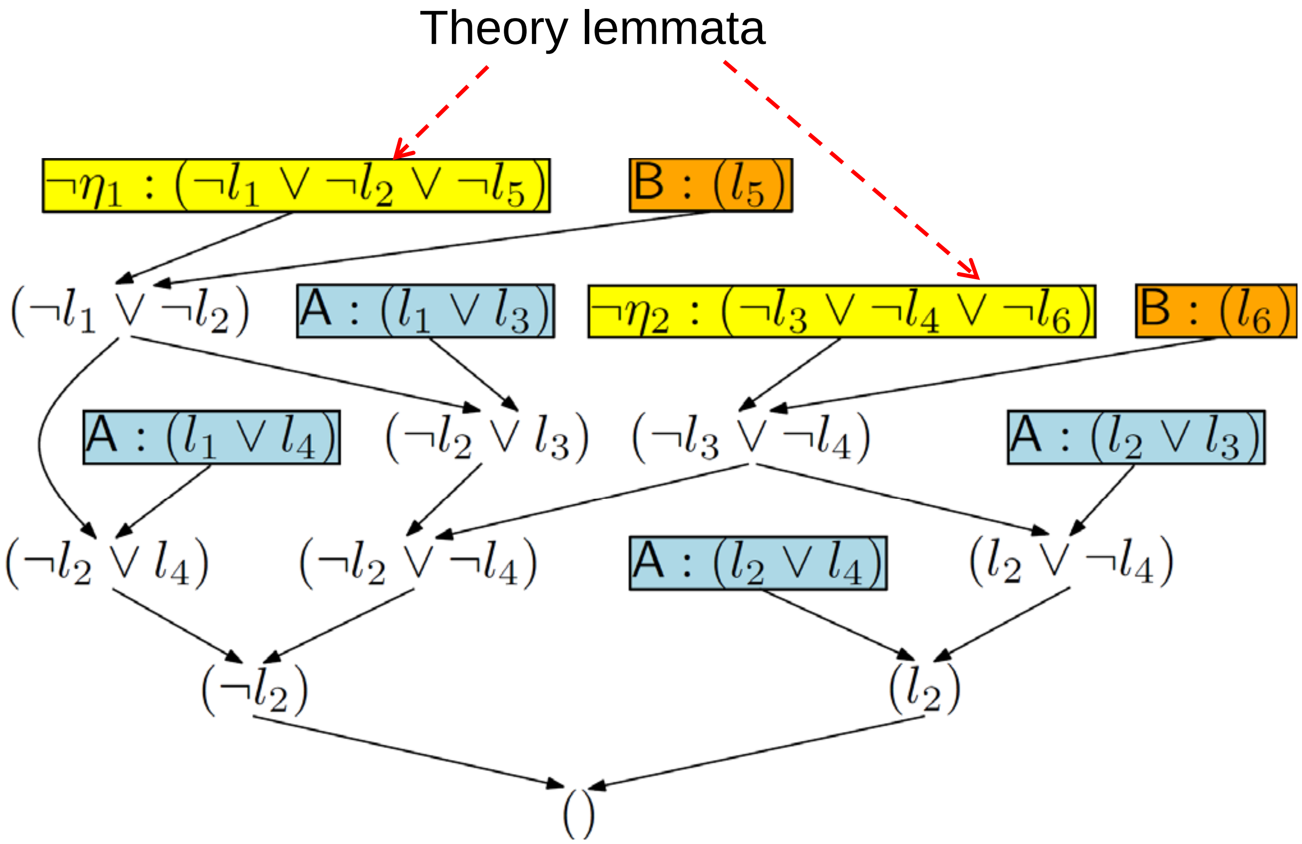
Closer look at interpolation procedure: Running example



$$
\begin{aligned}
l_1 &= (-x_2 \le 0),\\
l_2 &= (x_1 \le 1),\\
l_3 &= (-x_2 \le -5),\\
l_4 &= (x_1 \le 6),\\
l_5 &= (-2x_1 + x_2 \le -6),\\
l_6 &= (-x_1 + 2x_2 \le 0)
\end{aligned}
$$

$$
\begin{aligned}
\text{A} &= (l_1 \wedge l_2) \vee (l_3 \wedge l_4)\\
&= (l_1 \vee l_3) \wedge (l_1 \vee l_4)\\
&\quad \wedge (l_2 \vee l_3) \wedge (l_2 \vee l_4)\\
\text{B} &= (l_5 \wedge l_6)
\end{aligned}
$$

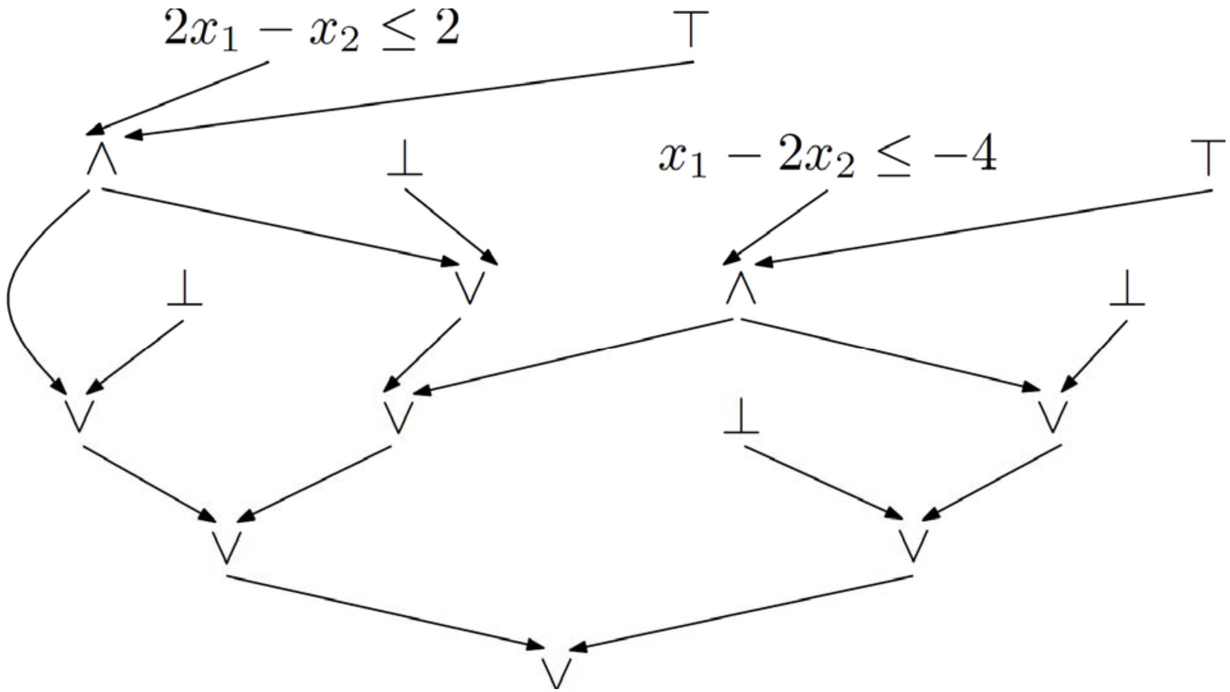
Proof of unsatisfiability



How to construct an interpolant?
(see McMillan 2005)

- Leaves:
  - Remove all atoms not occuring in B from A-clauses
  - Replace B-clauses by 1
  - Replace theory lemmata by single linear constraint, the „theory interpolant“
- Internal nodes:
  - Replace by OR, if pivot is not in B
  - Replace by AND, if pivot is in B

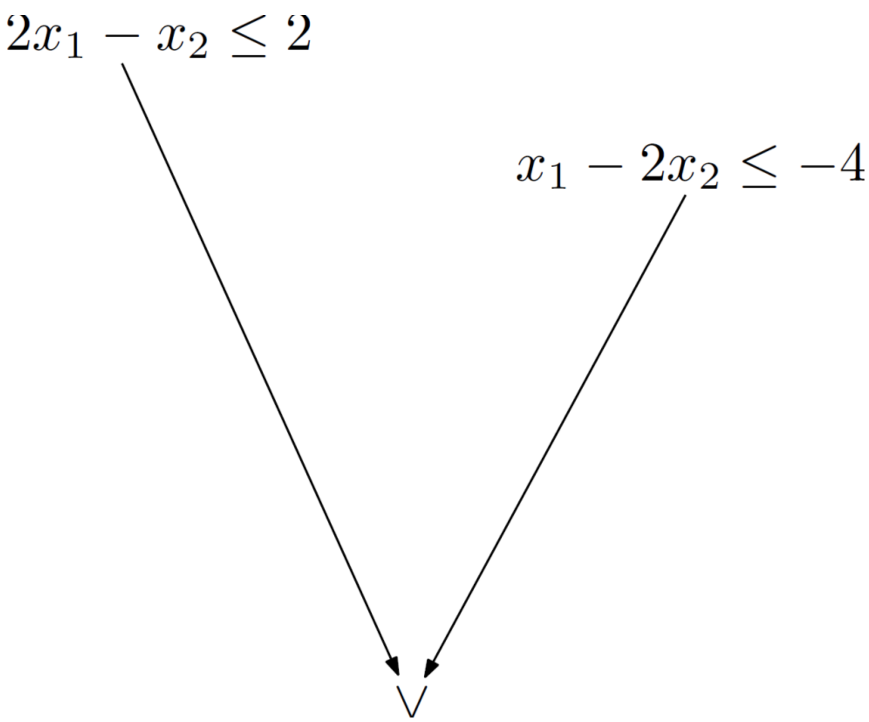
Interpolant



How to construct an interpolant?
(see McMillan 2005)

- Leaves:
  - Remove all atoms not occuring in B from A-clauses
  - Replace B-clauses by 1
  - Replace theory lemmata by single linear constraint, the „theory interpolant“
- Internal nodes:
  - Replace by OR, if pivot is not in B
  - Replace by AND, if pivot is in B

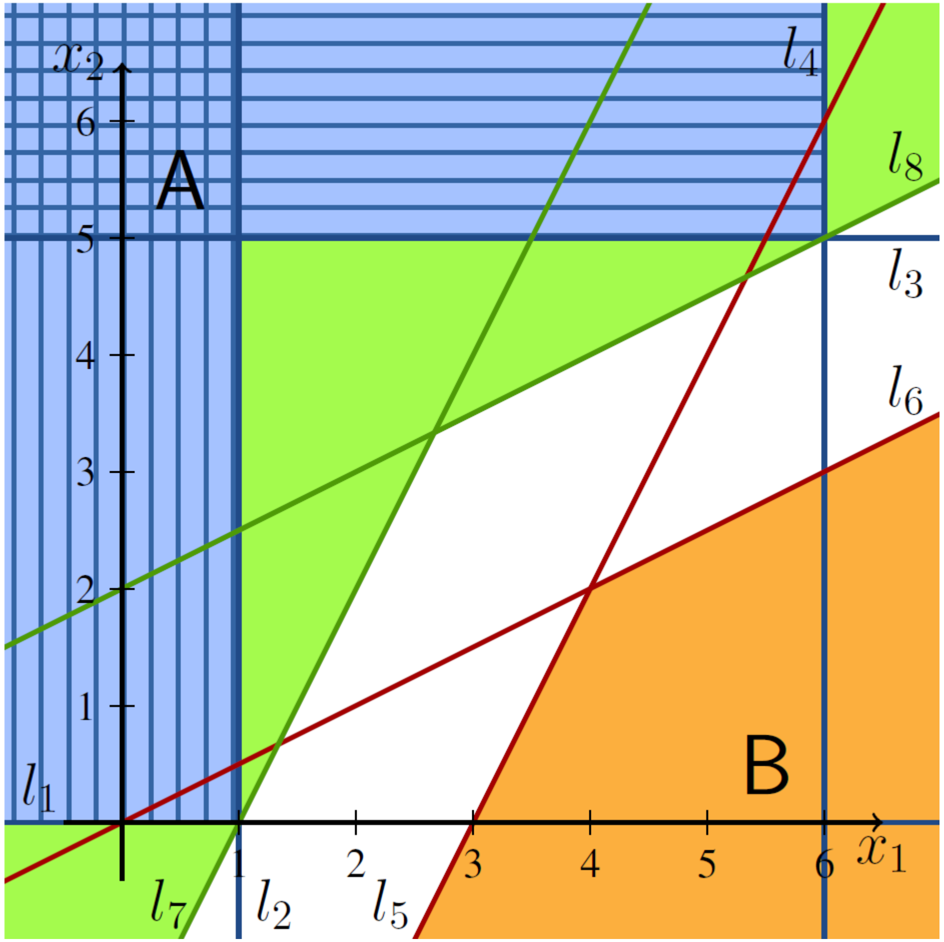
Interpolant



How to construct an interpolant?
(see McMillan 2005)

- Leaves:
  - Remove all atoms not occuring in B from A-clauses
  - Replace B-clauses by 1
  - Replace theory lemmata by single linear constraint, the „theory interpolant“
- Internal nodes:
  - Replace by OR, if pivot is not in B
  - Replace by AND, if pivot is in B

Interpolant



How to compute Theory Interpolants?

- Theory interpolants are computed for each theory lemma, e.g. $(\neg l_1 \vee \neg l_2 \vee \neg l_5)$
- The theory lemma says that $(l_1 \wedge l_2 \wedge l_5)$ is inconsistent.
- A theory interpolant is itself an interpolant of the „A-part“$(l_1 \wedge l_2)$ and the „B-part“ l_5.
- Proof of unsatisfiability for „A-part“ $\wedge$ „B-part“:
  - Non-negative linear combination leading to contradiction (e.g. 0 $\leq$ -4)

$$
\begin{array}{rrcr|l}
 & -x_2 & \leq & 0 & \cdot 1 \\
x_1 & & \leq & 1 & \cdot 2 \\
-2x_1 + & x_2 & \leq & -6 & \cdot 1 \\
\hline
0x_1 + & 0x_2 & \leq & -4 &
\end{array}
$$

How to compute Theory Interpolants?

- Theory interpolants are computed for each theory lemma, e.g. $(\neg l_1 \vee \neg l_2 \vee \neg l_5)$
- The theory lemma says that $(l_1 \wedge l_2 \wedge l_5)$ is inconsistent.
- A theory interpolant is itself an interpolant of the „A-part“ $(l_1 \wedge l_2)$ and the „B-part“ l_5.
- Interpolant I_t for „A-part“ $\wedge$ „B-part“:
  - First part of the proof belonging to the „A-part“

$$\begin{array}{rrrr|l} & -x_2 & \leq & 0 & \cdot 1 \\ x_1 & & \leq & 1 & \cdot 2 \\ -2x_1 + & x_2 & \leq & -6 & \cdot 1 \\ \hline 0x_1 + & 0x_2 & \leq & -4 & \end{array} \qquad \begin{array}{rrrr|l} & -x_2 & \leq & 0 & \cdot 1 \\ x_1 & & \leq & 1 & \cdot 2 \\ \hline \boxed{2x_1 -} & \boxed{x_2} & \boxed{\leq} & \boxed{2} & \end{array} \qquad \begin{array}{rrrr|l} 2x_1 - & x_2 & \leq & 2 & \cdot 1 \\ -2x_1 + & x_2 & \leq & -6 & \cdot 1 \\ \hline 0x_1 + & 0x_2 & \leq & -4 & \end{array}$$

Theory Interpolant I_t

Proof that $I_t \wedge$ „B-part“ = 0

Computing Theory Interpolants

- Theory interpolants can be computed by linear programming (Rybalchenko, Sofronie-Stokkermans 2007):

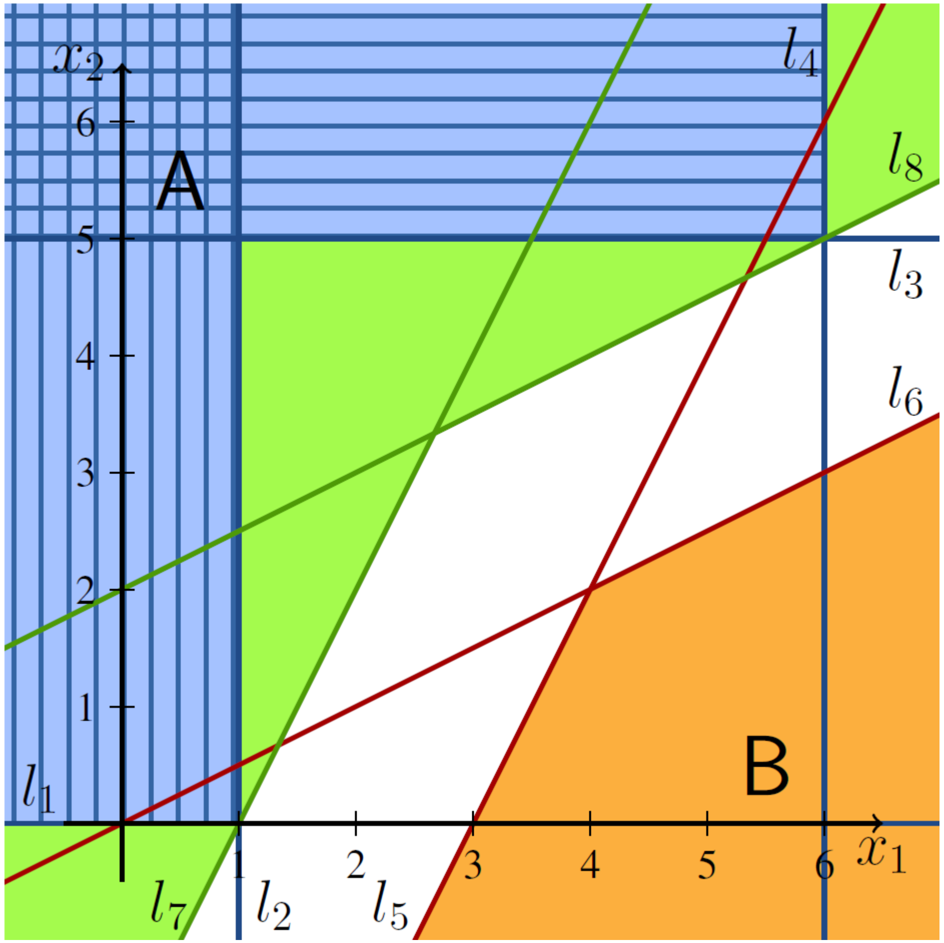
$$
\begin{array}{rcrl}
 & -x_2 \leq & 0 & | \cdot \lambda_1 \\
x_1 & \leq & 1 & | \cdot \lambda_2 \\
-2x_1 + & x_2 \leq & -6 & | \cdot \mu_1 \\
\hline
0x_1 + & 0x_2 \leq & -1 &
\end{array}
$$

- Suitable values for may be found by linear programming.
- The computed interpolant is a linear constraint $i_1 x_1 + i_2 x_2 \leq \delta$ with

$$
\boxed{
\begin{array}{rcl}
\lambda_1, \lambda_2, \mu_1 \geq 0 & & \\
\\
\lambda_2 - 2\mu_1 & = & 0 \\
-\lambda_1 \quad + \mu_1 & = & 0 \\
\lambda_2 - 6\mu_1 & \leq & -1 \\
\\
\lambda_2 & = & i_1 \\
-\lambda_1 & = & i_2 \\
\lambda_2 & = & \delta
\end{array}
}
$$

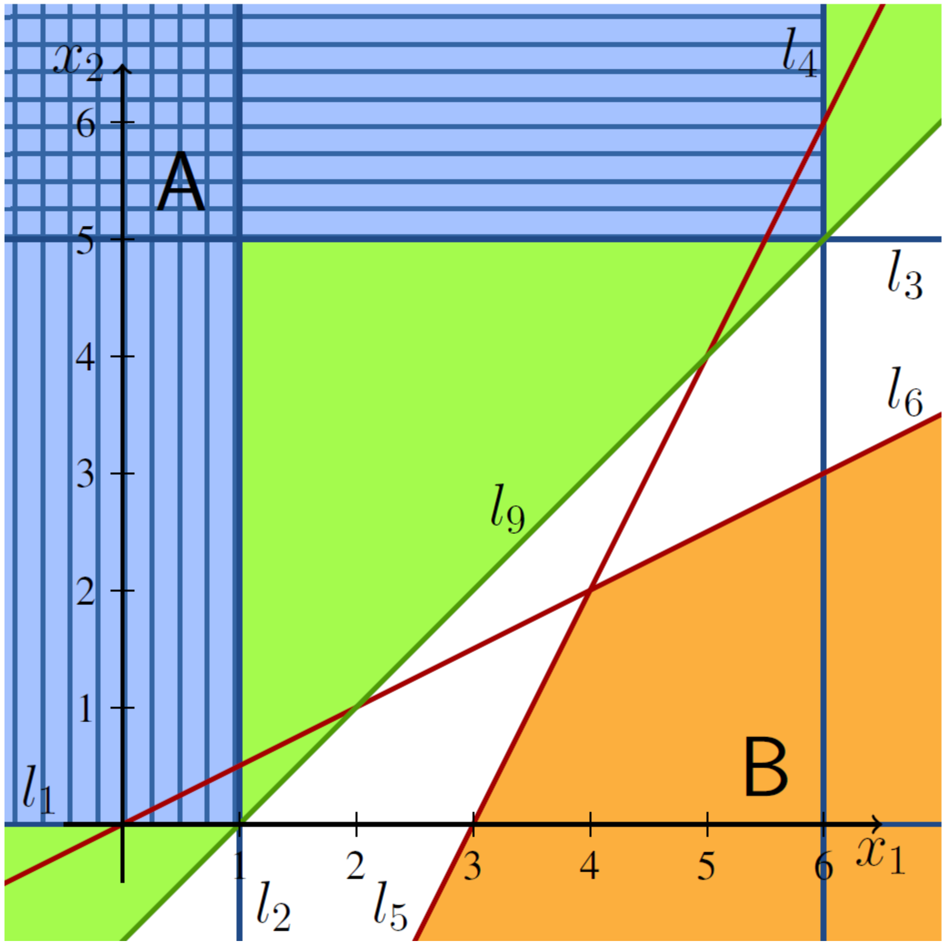
Running example

- This method results in exactly the following interpolant with one linear constraint for each theory lemma:



Running example

- However, there is an interpolant with a single linear constraint:



Shared Interpolants for Several Theory Lemmata

- Just an extension to the RS-2007-method:

$$\begin{array}{rcrl} & -x_2 \le & 0 & | \cdot \lambda_{1,1} \\ x_1 & \le & 1 & | \cdot \lambda_{1,2} \\ -2x_1 + & x_2 \le & -6 & | \cdot \mu_{1,1} \\ \hline 0x_1 + & 0x_2 \le & -1 & \end{array} \qquad \begin{array}{rcrl} & -x_2 \le & -5 & | \cdot \lambda_{2,1} \\ x_1 & \le & 6 & | \cdot \lambda_{2,2} \\ -x_1 + & 2x_2 \le & 0 & | \cdot \mu_{2,1} \\ \hline 0x_1 + & 0x_2 \le & -1 & \end{array}$$

$$\boxed{\begin{array}{rrcr} \multicolumn{4}{c}{\lambda_{1,1}, \lambda_{1,2}, \mu_{1,1} \ge 0} \\ & \lambda_{1,2} - 2\mu_{1,1} & = & 0 \\ -\lambda_{1,1} & + \mu_{1,1} & = & 0 \\ & \lambda_{1,2} - 6\mu_{1,1} & \le & -1 \\ & \lambda_{1,2} & = & i_1 \\ -\lambda_{1,1} & & = & i_2 \\ & \lambda_{1,2} & = & \delta \end{array}} \qquad \boxed{\begin{array}{rrcr} \multicolumn{4}{c}{\lambda_{2,1}, \lambda_{2,2}, \mu_{2,1} \ge 0} \\ & \lambda_{2,2} - \mu_{2,1} & = & 0 \\ -\lambda_{2,1} & + 2\mu_{2,1} & = & 0 \\ -5\lambda_{2,1} + 6\lambda_{2,2} & & \le & -1 \\ & \lambda_{2,2} & = & i_1 \\ -\lambda_{2,1} & & = & i_2 \\ -5\lambda_{2,1} + 6\lambda_{2,2} & & = & \delta \end{array}}$$

- Shared theory interpolant $i_1x_1 + i_2x_2 \le \delta$ for two theory lemmata?
- … can be computed by linear programming as well.

Shared Interpolants for Several Theory Lemmata

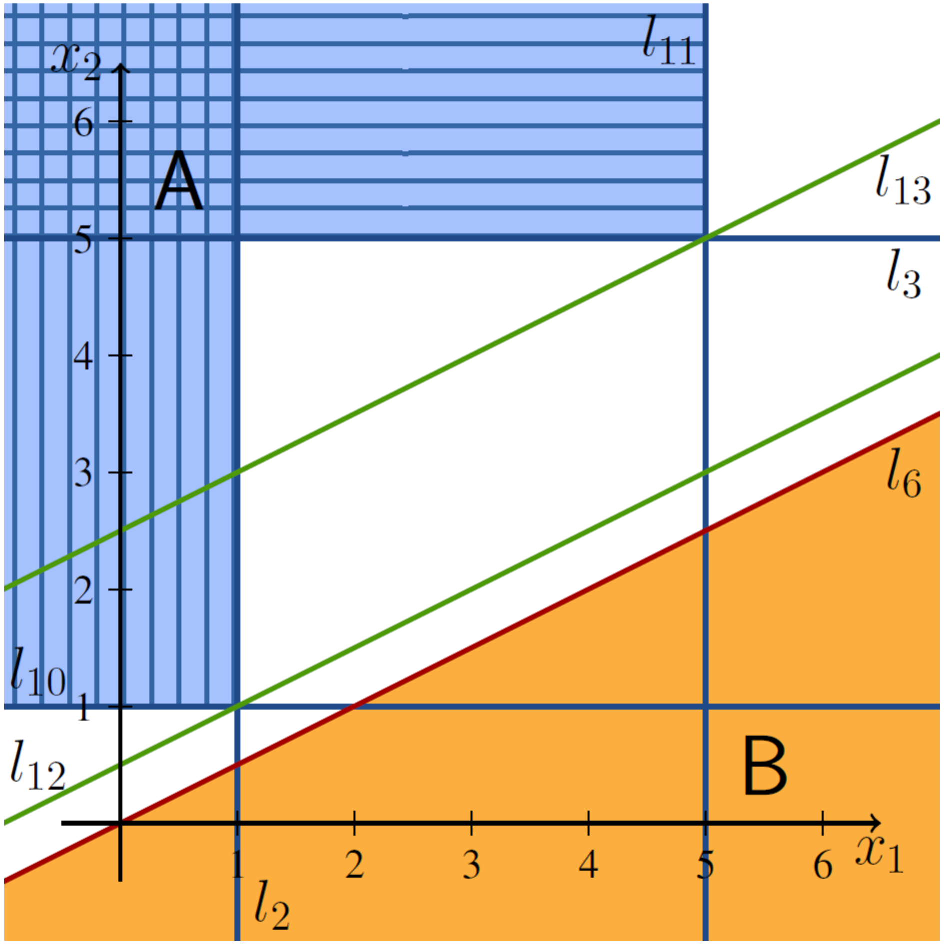
- Unfortunately, first results showed that this does not work!
- The potential to find shared interpolants for several theory lemmata is not high enough.

- More degrees of freedom are needed to enable a larger number of shared interpolants …

- **1st approach: Relaxing constraints**
- **Lemma**: The RS-2007-method only computes theory interpolants which touch the A-part of the theory conflict (as long as the theory conflict is minimized, and both A- and B-part are not empty).

$\Rightarrow$ Relax constraints to remove this restriction

Relaxing constraints



Relaxing constraints

$$
\begin{array}{rrcrl}
 & -x_2 & \leq & 0 & | \cdot \lambda_{1,1} \\
x_1 & & \leq & 1 & | \cdot \lambda_{1,2} \\
-2x_1 + & x_2 & \leq & -6 & | \cdot \mu_{1,1} \\
\hline
0x_1 + & 0x_2 & \leq & -1 &
\end{array}
\qquad
\begin{array}{rrcrl}
 & -x_2 & \leq & -5 & | \cdot \lambda_{2,1} \\
x_1 & & \leq & 6 & | \cdot \lambda_{2,2} \\
-x_1 + & 2x_2 & \leq & 0 & | \cdot \mu_{2,1} \\
\hline
0x_1 + & 0x_2 & \leq & -1 &
\end{array}
$$

$$
\boxed{
\begin{array}{l}
\lambda_{1,1}, \lambda_{1,2}, \mu_{1,1} \geq 0 \\
\\
\begin{array}{rrrcr}
 & \lambda_{1,2} & - 2\mu_{1,1} & = & 0 \\
-\lambda_{1,1} & & + \mu_{1,1} & = & 0 \\
 & \lambda_{1,2} & - 6\mu_{1,1} & \leq & -1 \\
\\
 & \lambda_{1,2} & & = & i_1 \\
-\lambda_{1,1} & & & = & i_2 \\
 & \lambda_{1,2} & & = & \delta
\end{array}
\end{array}
}
\qquad
\boxed{
\begin{array}{l}
\lambda_{2,1}, \lambda_{2,2}, \mu_{2,1} \geq 0 \\
\\
\begin{array}{rrrcr}
 & \lambda_{2,2} & - \mu_{2,1} & = & 0 \\
-\lambda_{2,1} & & + 2\mu_{2,1} & = & 0 \\
-5\lambda_{2,1} & + 6\lambda_{2,2} & & \leq & -1 \\
\\
 & \lambda_{2,2} & & = & i_1 \\
-\lambda_{2,1} & & & = & i_2 \\
-5\lambda_{2,1} & + 6\lambda_{2,2} & & = & \delta
\end{array}
\end{array}
}
$$

- Shared interpolant $i_1 x_1 + i_2 x_2 \leq \delta$

Relaxing constraints

$$\begin{array}{rrcr|l}
 & -x_2 & \leq & 0 & \cdot \lambda_{1,1} \\
x_1 & & \leq & 1 & \cdot \lambda_{1,2} \\
-2x_1 + & x_2 & \leq & -6 & \cdot \mu_{1,1} \\
\hline
0x_1 + & 0x_2 & \leq & -1 &
\end{array}
\qquad
\begin{array}{rrcr|l}
 & -x_2 & \leq & -5 & \cdot \lambda_{2,1} \\
x_1 & & \leq & 6 & \cdot \lambda_{2,2} \\
-x_1 + & 2x_2 & \leq & 0 & \cdot \mu_{2,1} \\
\hline
0x_1 + & 0x_2 & \leq & -1 &
\end{array}$$

$$\boxed{\begin{array}{c}
\lambda_{1,1}, \lambda_{1,2}, \mu_{1,1} \geq 0 \\[1em]
\begin{array}{rrcr}
 & \lambda_{1,2} - 2\mu_{1,1} & = & 0 \\
-\lambda_{1,1} & + \mu_{1,1} & = & 0 \\
 & \lambda_{1,2} - 6\mu_{1,1} & \leq & -1 \\[1em]
 & \lambda_{1,2} & = & i_1 \\
-\lambda_{1,1} & & = & i_2 \\
 & \lambda_{1,2} & \leq & \delta
\end{array}
\end{array}}
\qquad
\boxed{\begin{array}{c}
\lambda_{2,1}, \lambda_{2,2}, \mu_{2,1} \geq 0 \\[1em]
\begin{array}{rrcr}
 & \lambda_{2,2} - \mu_{2,1} & = & 0 \\
-\lambda_{2,1} & + 2\mu_{2,1} & = & 0 \\
-5\lambda_{2,1} + 6\lambda_{2,2} & & \leq & -1 \\[1em]
 & \lambda_{2,2} & = & i_1 \\
-\lambda_{2,1} & & = & i_2 \\
-5\lambda_{2,1} + 6\lambda_{2,2} & & \leq & \delta
\end{array}
\end{array}}$$

- Shared interpolant $i_1 x_1 + i_2 x_2 \leq \delta$

Relaxing constraints

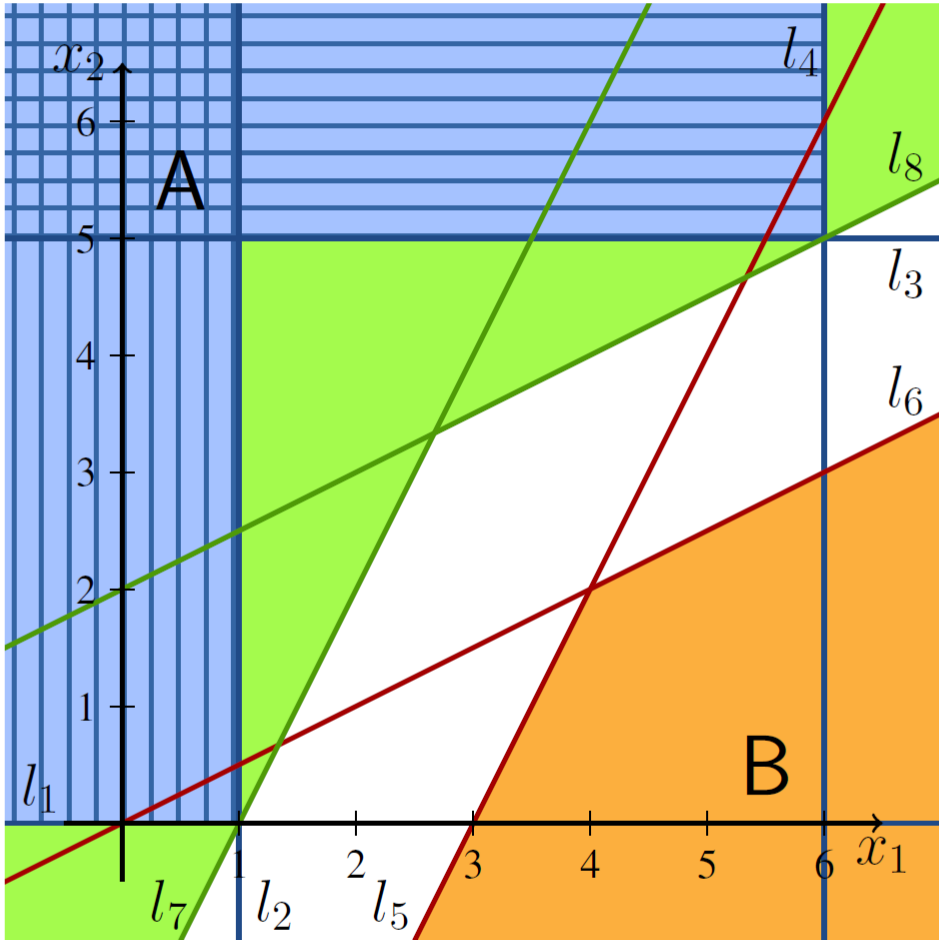
$$
\begin{array}{rcrl}
 & -x_2 \leq & 0 & \mid \cdot \lambda_{1,1} \\
x_1 & \leq & 1 & \mid \cdot \lambda_{1,2} \\
-2x_1 + & x_2 \leq & -6 & \mid \cdot \mu_{1,1} \\
\hline
0x_1 + & 0x_2 \leq & -1 &
\end{array}
\qquad
\begin{array}{rcrl}
 & -x_2 \leq & -5 & \mid \cdot \lambda_{2,1} \\
x_1 & \leq & 6 & \mid \cdot \lambda_{2,2} \\
-x_1 + & 2x_2 \leq & 0 & \mid \cdot \mu_{2,1} \\
\hline
0x_1 + & 0x_2 \leq & -1 &
\end{array}
$$

$$
\boxed{
\begin{array}{c}
\lambda_{1,1}, \lambda_{1,2}, \mu_{1,1} \geq 0 \\
\\
\begin{array}{rrrcr}
 & \lambda_{1,2} & - 2\mu_{1,1} & = & 0 \\
-\lambda_{1,1} & & + \ \mu_{1,1} & = & 0 \\
\delta & & - 6\mu_{1,1} & \leq & -1 \\
\\
 & \lambda_{1,2} & & = & i_1 \\
-\lambda_{1,1} & & & = & i_2 \\
 & \lambda_{1,2} & & \leq & \delta
\end{array}
\end{array}
}
\qquad
\boxed{
\begin{array}{c}
\lambda_{2,1}, \lambda_{2,2}, \mu_{2,1} \geq 0 \\
\\
\begin{array}{rrrcr}
 & \lambda_{2,2} & - \ \mu_{2,1} & = & 0 \\
-\lambda_{2,1} & & + 2\mu_{2,1} & = & 0 \\
\delta & & & \leq & -1 \\
\\
 & \lambda_{2,2} & & = & i_1 \\
-\lambda_{2,1} & & & = & i_2 \\
-5\lambda_{2,1} + & 6\lambda_{2,2} & & \leq & \delta
\end{array}
\end{array}
}
$$

- Shared interpolant $i_1 x_1 + i_2 x_2 \leq \delta$

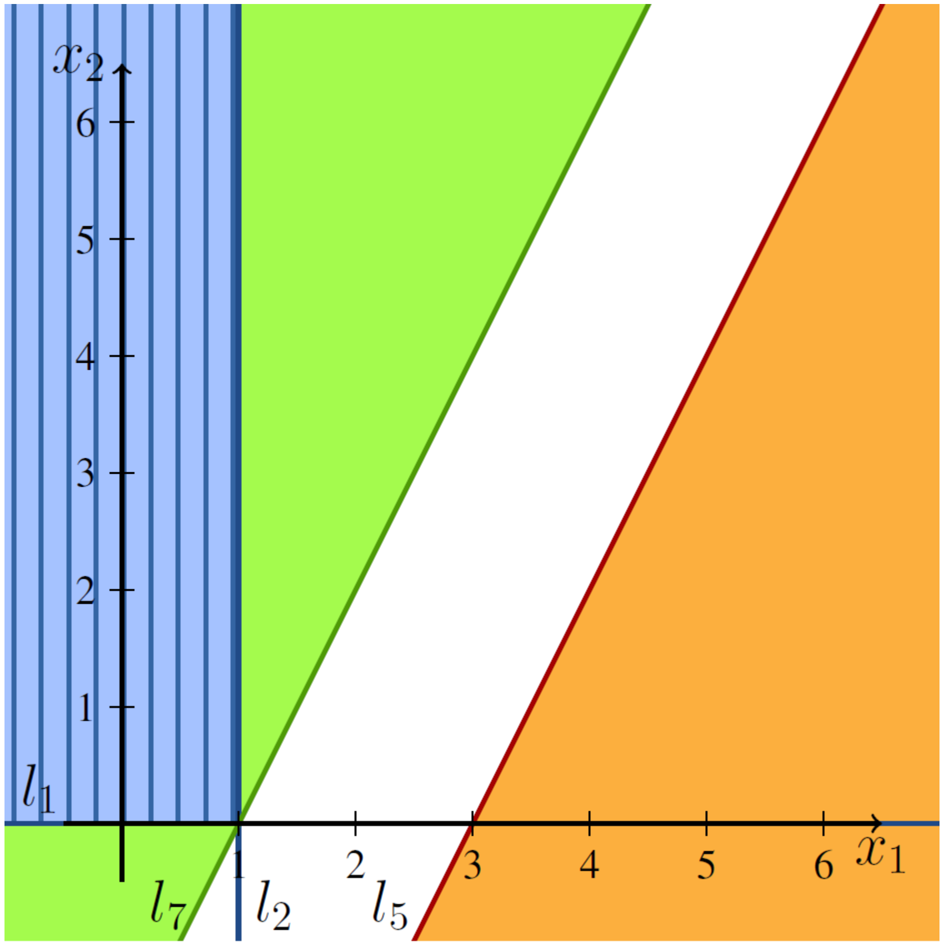
Shared Interpolants for Several Theory Lemmata

- Unfortunately, this still does not work for our example:



Shared Interpolants for Several Theory Lemmata

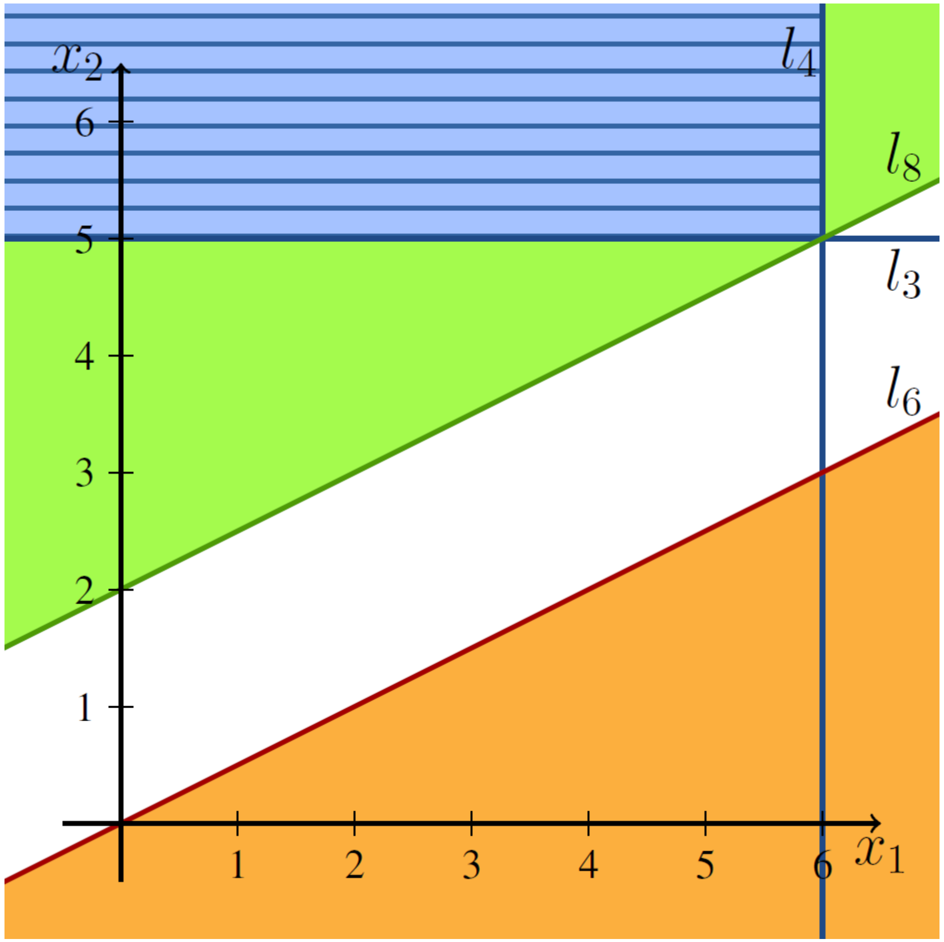
- Unfortunately, this still does not work for our example: 1st theory lemma



Direction of l_7 is fixed!

Shared Interpolants for Several Theory Lemmata

- Unfortunately, this still does not work for our example: 2nd theory lemma

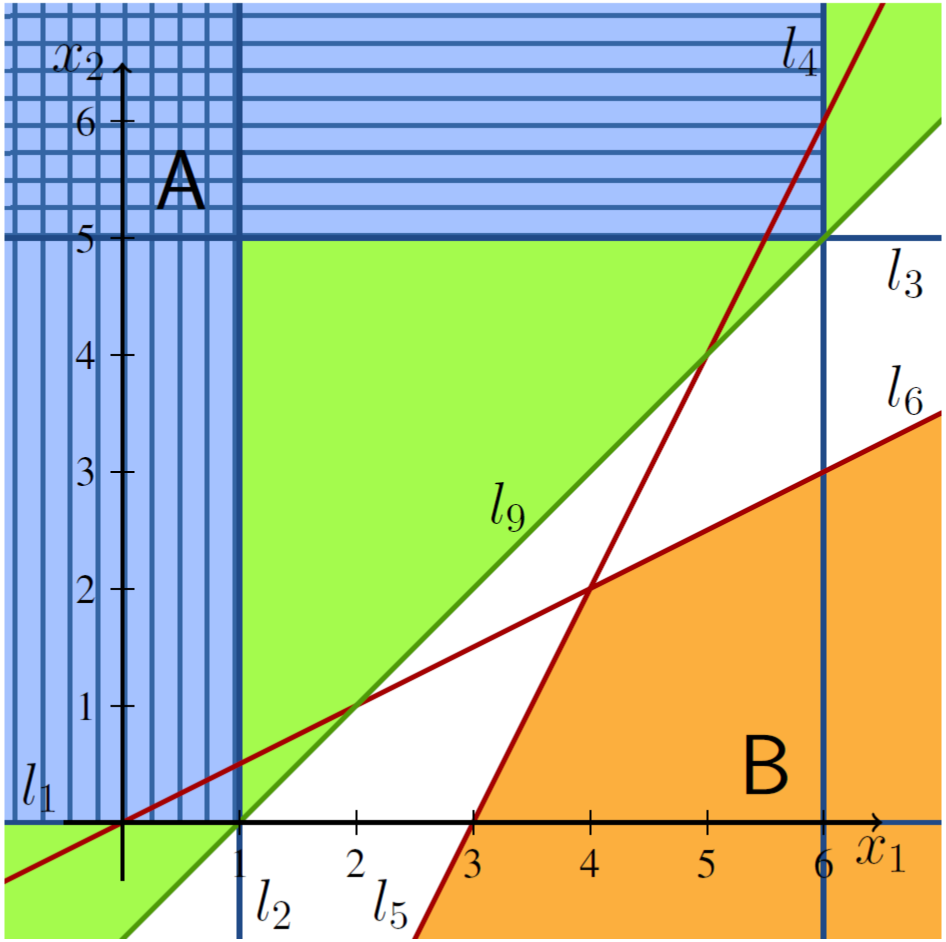


Direction of l_8 is fixed!

Shared Interpolants for Several Theory Lemmata

- **Lemma**: If a theory conflict is minimized (and neither A-part nor B-part are empty), then the direction vector of the theory interpolant is fixed.
- **However**: Modern SMT solvers **minimize** theory conflicts in order to prune the search space as much as possible!
- **Idea**: Extend theory lemmata by additional inequations in a way that
  - the SMT proof is not destroyed,
  - or at least: The interpolant computed as before is still an interpolant.
- **Note**: Of course an inconsistent set of constraints remains inconsistent, if extended by additional constraints.

Running example



- If $(l_1 \wedge l_2 \wedge l_5)$ is extended to $(l_1 \wedge l_2 \wedge l_5 \wedge l_6)$ and $(l_3 \wedge l_4 \wedge l_6)$ is extended to $(l_3 \wedge l_4 \wedge l_5 \wedge l_6)$, then l_9 is a shared interpolant for both theory lemmata.

Extending Theory Lemmata, Method 1

- 1st method: Push-up operation

$$\begin{tikzcd}[column sep=small] (\neg a \lor c) \arrow[dr] & & \boxed{(a \lor b)} \arrow[dl] \\ & (b \lor c) & \end{tikzcd}$$

(Pigorsch / Scholl, DATE 2013)

Extending Theory Lemmata, Method 1

- 1st method: Push-up operation

$$
\begin{tikzcd}
(\neg b \lor c) \arrow[dr] & & (\neg a \lor c) \arrow[dr] & & \boxed{(a \lor b \lor c)} \arrow[dlll] \arrow[dl] \arrow[dr] & & (\neg b \lor c \lor e) \arrow[dl] \\
& (a \lor c) & & (b \lor c) & & (a \lor c \lor e) &
\end{tikzcd}
$$

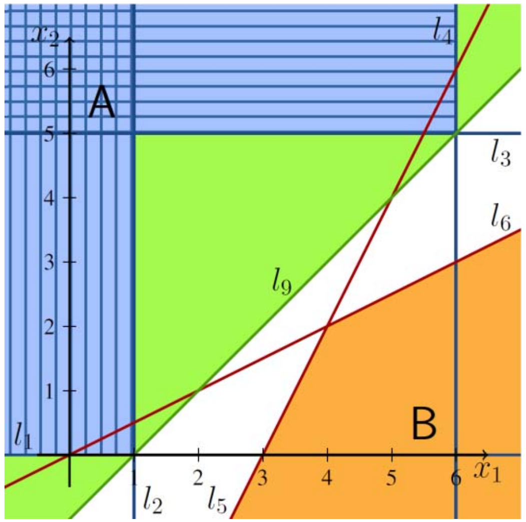
- Resolution proof remains valid after push-up of a literal *c* into clause *n*, if
  - *c* is in the intersection of all of its children's clauses,
  - *c* is not *n*'s pivot.

- Extend theory lemmata by literals pushed into them …
- After push-up operations, the SMT proof remains valid.

(Pigorsch / Scholl, DATE 2013)

Extending Theory Lemmata, Method 2

- 2nd method: Implied literals
- A literal l is
  - implied for B, iff $\mathsf{B} \Rightarrow l$,
  - implied for A, iff $\mathsf{A} \Rightarrow l$ and l does not occur in B.
- **Lemma:** Adding the negation of implied literals to theory lemmata in an SMT proof and using the interpolation construction according to [McMillan 2005] leads to a valid interpolant.



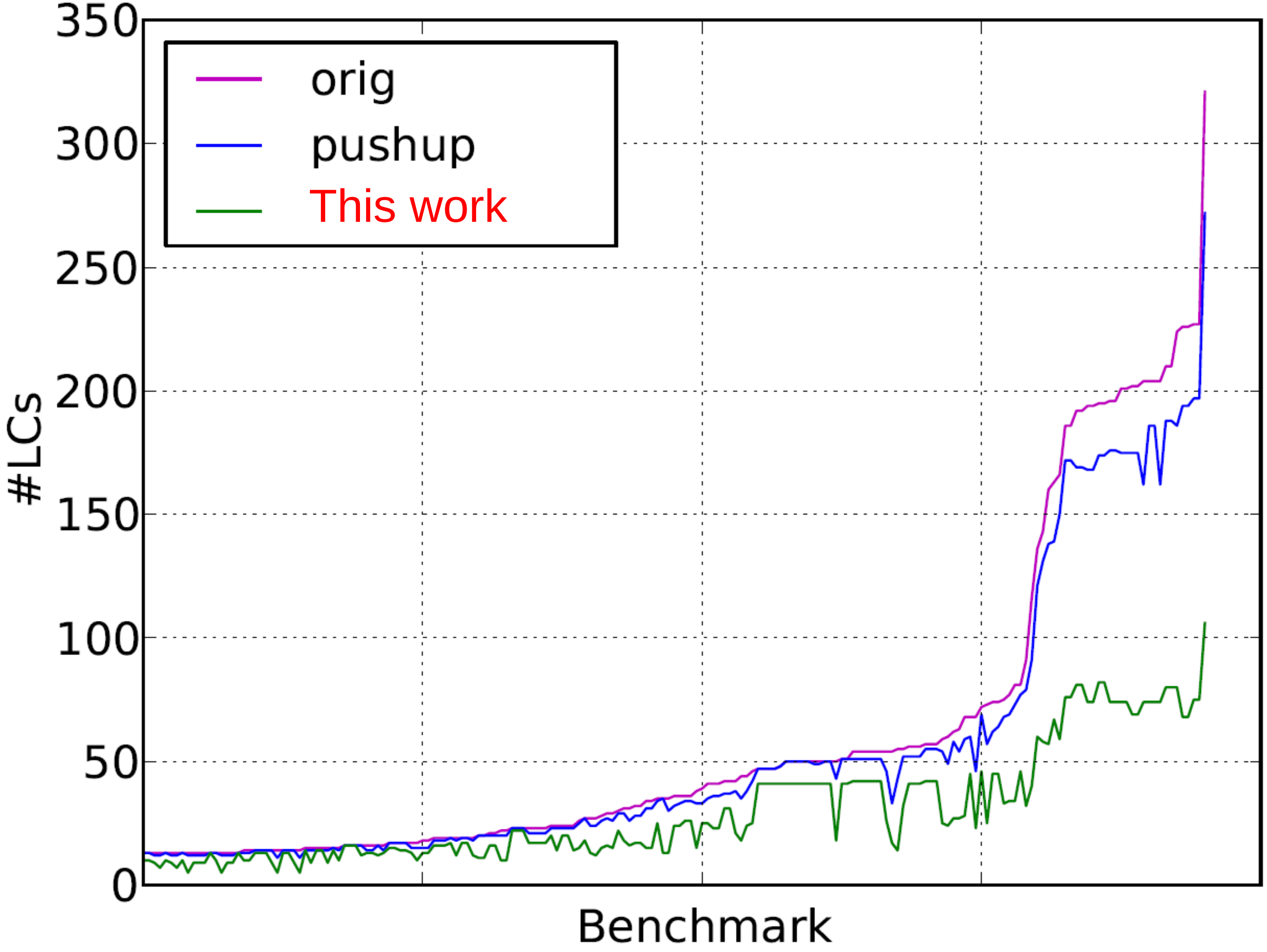
- **Running example**:
  - l_1 and l_4 are implied for A.
  - l_5 and l_6 are implied for B.
  - Theory lemma $(\neg l_1 \vee \neg l_2 \vee \neg l_5)$ may be extended to $(\neg l_1 \vee \neg l_2 \vee \neg l_4 \vee \neg l_5 \vee \neg l_6)$
  - Theory lemma $(\neg l_3 \vee \neg l_4 \vee \neg l_6)$ may be extended to $(\neg l_1 \vee \neg l_3 \vee \neg l_4 \vee \neg l_5 \vee \neg l_6)$
  - This leads to the shared theory interpolant as depicted.

(Scholl, Pigorsch, Disch, Althaus, DATE 2014)

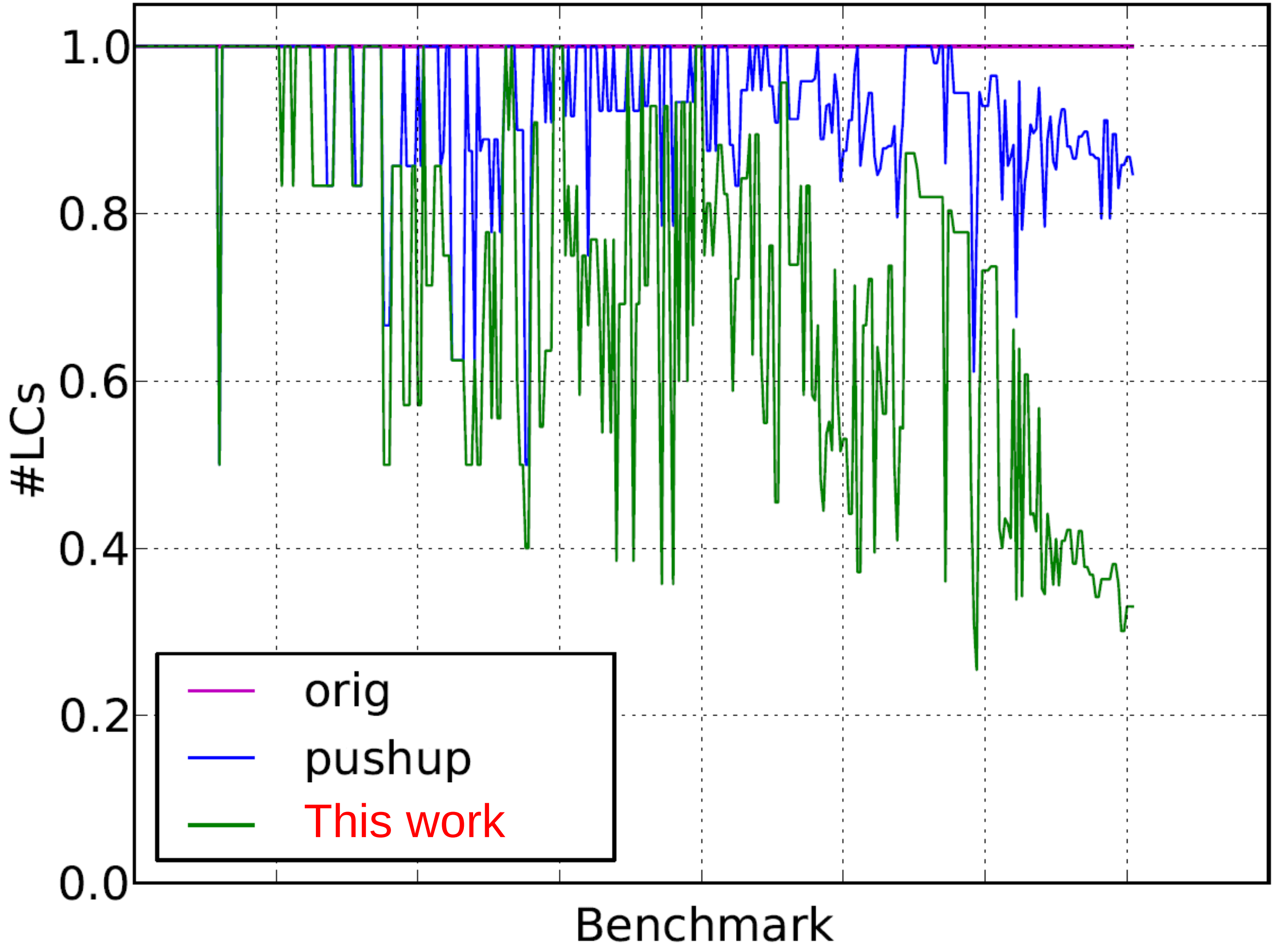
Experiments

- > 200 intermediate state sets produced by our hybrid model checker (representing A).
- ϵ-bloating of state sets represents $\neg B$.
- Formulas representing A and B contain up to 7 rational variables, up to 1,380 inequations, up to 18,915 Boolean variables, and up to 56,721 clauses.

First Results



First Results



Conclusions and Future Work

- Interpolants based on proofs of unsatisfiability may be simplified to a great extent by shared interpolants.
- Key to successful simplification: Preprocessing proofs to increase degrees of freedom in the selection of theory interpolants.
- Existing LP solvers / SMT solvers may be used.

- Generalization to other theories?
- To do: Full integration into model checking procedure with abstraction refinement.