Generic theory combination for model-constructing satisfiability (MCSAT)

Stéphane Graham-Lengrand
CNRS - SRI International

joint work with Maria Paola Bonacina and Natarajan Shankar

Formal Methods Seminar, 14th February 2017

(Ground) Sat-Modulo-Theories problems

Trying to determine whether a collection of formulæ has a model (sat) or not (unsat).
Formulae are here

- built without quantifiers
- defined as terms of sort bool

…terms being those of multi-sorted first-order logic, i.e. built with (free) variables and symbols declared with input and output sorts, e.g.

$$
\begin{aligned}
&f : s_1 \to s_2\\
&+, \times : (\mathrm{Q} \times \mathrm{Q}) \to \mathrm{Q}\\
&\mathrm{is\_prime} : \mathrm{N} \to \mathrm{bool}\\
&=_s : (s \times s) \to \mathrm{bool}\\
&\leq : (\mathrm{Q} \times \mathrm{Q}) \to \mathrm{bool}\\
&\vee, \wedge : (\mathrm{bool} \times \mathrm{bool}) \to \mathrm{bool}\\
&\ldots
\end{aligned}
$$

(Ground) Sat-Modulo-Theories problems

Trying to determine whether a collection of formulæ has a model (sat) or not (unsat).

Formulae are here

- built without quantifiers
- defined as terms of sort bool

The question of satisfiability is asked respectively to a range of theories $\mathcal{T}_1, \ldots, \mathcal{T}_k$, which may impose or restrict the way each sort and each symbol is interpreted:

For instance,

- the Boolean theory imposes that sort Bool be interpreted as $\{\text{true}, \text{false}\}$ and $\vee, \wedge$ be interpreted with the usual truth tables, etc.
- Linear Rational Arithmetic imposes that $+$ be interpreted in the intuitive way, but does not know anything about $\times$, etc

Traditional approaches

When the only theory involved is the Boolean one,
then this is SAT-solving.
Can be addressed by (clausification$+$) DPLL/CDCL.

Traditional approaches

When the only theory involved is the Boolean one,
then this is SAT-solving.
Can be addressed by (clausification+) DPLL/CDCL.
In presence of other theories, a popular architecture is
DPLL($\bigcup_{i=1}^{n} \mathcal{T}_i$), where

- a front-end is a SAT-solver running DPLL/CDCL;
- it is interfaced with a backend that combines decision procedures for the theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$
(usually by the Nelson-Oppen combination technique)

What is MC-Sat?

What is MC-Sat?

Introduced in [dMJ13, JBdM13]

What is MC-Sat?

Introduced in [dMJ13, JBdM13]
2 things:

- a template for theory-specific decision procedures
- a sound and complete way of combining theories following the template

What is MC-Sat?

Introduced in [dMJ13, JBdM13]
2 things:

- a template for theory-specific decision procedures
- a sound and complete way of combining theories following the template

The template is an abstraction of how DPLL/CDCL works, which becomes one instance of the scheme.

What is MC-Sat?

Introduced in [dMJ13, JBdM13]
2 things:

- a template for theory-specific decision procedures
- a sound and complete way of combining theories following the template

The template is an abstraction of how DPLL/CDCL works, which becomes one instance of the scheme.
Consequence: Bool. theory has the same status as other theories.
(differs from traditional SMT-architecture)

What is MC-Sat?

Introduced in [dMJ13, JBdM13]
2 things:

- a template for theory-specific decision procedures
- a sound and complete way of combining theories following the template

The template is an abstraction of how DPLL/CDCL works, which becomes one instance of the scheme.
Consequence: Bool. theory has the same status as other theories.
(differs from traditional SMT-architecture)

Other differences with traditional approaches:

- terms and literals are exchanged that do not belong to the original problem;
- parts that are really specific to the theories can consist of much smaller steps.

1. A glance at MC-Sat

An example in Linear Rational Arithmetic

$l_0 : (-2{\cdot}x - y < 0)$, $l_1 : (x + y < 0)$, $l_2 : (x < -1)$

unsatisfiable in LRA.

An example in Linear Rational Arithmetic

$l_0 : (-2{\cdot}x - y < 0)$, $l_1 : (x + y < 0)$, $l_2 : (x < -1)$

unsatisfiable in LRA. Example of MCSAT run:

- Guess a value, e.g. $y \leftarrow 0$

An example in Linear Rational Arithmetic

$l_0 : (-2{\cdot}x - y < 0), \qquad l_1 : (x + y < 0), \qquad l_2 : (x < -1)$

unsatisfiable in LRA. Example of MCSAT run:

- Guess a value, e.g. $y \leftarrow 0$
  Then l_0 yields lower bound $x > 0$

An example in Linear Rational Arithmetic

$l_0 : (-2{\cdot}x - y < 0)$, $l_1 : (x + y < 0)$, $l_2 : (x < -1)$

unsatisfiable in LRA. Example of MCSAT run:

- Guess a value, e.g. $y \leftarrow 0$
  Then l_0 yields lower bound $x > 0$
  Together with l_2, space of possible values for x is empty
  What to do? just undo $y \leftarrow 0$?

An example in Linear Rational Arithmetic

$l_0 : (-2{\cdot}x-y < 0)$, $l_1 : (x+y < 0)$, $l_2 : (x < -1)$

unsatisfiable in LRA. Example of MCSAT run:

- Guess a value, e.g. $y{\leftarrow}0$
  Then l_0 yields lower bound $x > 0$
  Together with l_2, space of possible values for x is empty
  What to do? just undo $y{\leftarrow}0$? No:
- Clash of bounds suggests to infer $l_0 + 2l_2$, i.e. $l_3 : (-y < -2)$
  indeed violated by the guess $y{\leftarrow}0$

An example in Linear Rational Arithmetic

$l_0 : (-2{\cdot}x - y < 0)$, $l_1 : (x + y < 0)$, $l_2 : (x < -1)$

unsatisfiable in LRA. Example of MCSAT run:

- Guess a value, e.g. $y \leftarrow 0$
  Then l_0 yields lower bound $x > 0$
  Together with l_2, space of possible values for x is empty
  What to do? just undo $y \leftarrow 0$? No:
- Clash of bounds suggests to infer $l_0 + 2l_2$, i.e. $l_3 : (-y < -2)$
  indeed violated by the guess $y \leftarrow 0$
- Now undo the guess but keep l_3.

An example in Linear Rational Arithmetic

$l_0 : (-2{\cdot}x - y < 0)$, $l_1 : (x + y < 0)$, $l_2 : (x < -1)$

unsatisfiable in LRA. Example of MCSAT run:

- Guess a value, e.g. $y \leftarrow 0$
  Then l_0 yields lower bound $x > 0$
  Together with l_2, space of possible values for x is empty
  What to do? just undo $y \leftarrow 0$? No:
- Clash of bounds suggests to infer $l_0 + 2l_2$, i.e. $l_3 : (-y < -2)$
  indeed violated by the guess $y \leftarrow 0$
- Now undo the guess but keep l_3.
- Try new guess, say $y \leftarrow 4$

An example in Linear Rational Arithmetic

$l_0 : (-2{\cdot}x - y < 0)$, $l_1 : (x + y < 0)$, $l_2 : (x < -1)$

unsatisfiable in LRA. Example of MCSAT run:

- Guess a value, e.g. $y \leftarrow 0$
  Then l_0 yields lower bound $x > 0$
  Together with l_2, space of possible values for x is empty
  What to do? just undo $y \leftarrow 0$? No:
- Clash of bounds suggests to infer $l_0 + 2l_2$, i.e. $l_3 : (-y < -2)$
  indeed violated by the guess $y \leftarrow 0$
- Now undo the guess but keep l_3.
- Try new guess, say $y \leftarrow 4$
  l_1 yields upper bound $x < -4$, l_0 yields lower bound $x > -2$

An example in Linear Rational Arithmetic

$l_0 : (-2{\cdot}x - y < 0)$, $l_1 : (x + y < 0)$, $l_2 : (x < -1)$

unsatisfiable in LRA. Example of MCSAT run:

- Guess a value, e.g. $y \leftarrow 0$
  Then l_0 yields lower bound $x > 0$
  Together with l_2, space of possible values for x is empty
  What to do? just undo $y \leftarrow 0$? No:
- Clash of bounds suggests to infer $l_0 + 2l_2$, i.e. $l_3 : (-y < -2)$
  indeed violated by the guess $y \leftarrow 0$
- Now undo the guess but keep l_3.
- Try new guess, say $y \leftarrow 4$
  l_1 yields upper bound $x < -4$, l_0 yields lower bound $x > -2$
- Clash of bounds suggests to infer $l_0 + 2l_1$, i.e. $l_4 : (y < 0)$
  indeed violated by the guess $y \leftarrow 4$

An example in Linear Rational Arithmetic

$l_0 : (-2{\cdot}x - y < 0)$, $l_1 : (x + y < 0)$, $l_2 : (x < -1)$

unsatisfiable in LRA. Example of MCSAT run:

- Guess a value, e.g. $y \leftarrow 0$
  Then l_0 yields lower bound $x > 0$
  Together with l_2, space of possible values for x is empty
  What to do? just undo $y \leftarrow 0$? No:
- Clash of bounds suggests to infer $l_0 + 2l_2$, i.e. $l_3 : (-y < -2)$
  indeed violated by the guess $y \leftarrow 0$
- Now undo the guess but keep l_3.
- Try new guess, say $y \leftarrow 4$
  l_1 yields upper bound $x < -4$, l_0 yields lower bound $x > -2$
- Clash of bounds suggests to infer $l_0 + 2l_1$, i.e. $l_4 : (y < 0)$
  indeed violated by the guess $y \leftarrow 4$
- Undo guess, keep l_4

An example in Linear Rational Arithmetic

$l_0 : (-2{\cdot}x - y < 0)$, $l_1 : (x + y < 0)$, $l_2 : (x < -1)$

unsatisfiable in LRA. Example of MCSAT run:

- Guess a value, e.g. $y \leftarrow 0$
  Then l_0 yields lower bound $x > 0$
  Together with l_2, space of possible values for x is empty
  What to do? just undo $y \leftarrow 0$? No:
- Clash of bounds suggests to infer $l_0 + 2l_2$, i.e. $l_3 : (-y < -2)$
  indeed violated by the guess $y \leftarrow 0$
- Now undo the guess but keep l_3.
- Try new guess, say $y \leftarrow 4$
  l_1 yields upper bound $x < -4$, l_0 yields lower bound $x > -2$
- Clash of bounds suggests to infer $l_0 + 2l_1$, i.e. $l_4 : (y < 0)$
  indeed violated by the guess $y \leftarrow 4$
- Undo guess, keep l_4
  l_3 and l_4 give clash of bounds for y

An example in Linear Rational Arithmetic

$l_0 : (-2{\cdot}x - y < 0)$, $l_1 : (x + y < 0)$, $l_2 : (x < -1)$

unsatisfiable in LRA. Example of MCSAT run:

- Guess a value, e.g. $y \leftarrow 0$
  Then l_0 yields lower bound $x > 0$
  Together with l_2, space of possible values for x is empty
  What to do? just undo $y \leftarrow 0$? No:
- Clash of bounds suggests to infer $l_0 + 2l_2$, i.e. $l_3 : (-y < -2)$
  indeed violated by the guess $y \leftarrow 0$
- Now undo the guess but keep l_3.
- Try new guess, say $y \leftarrow 4$
  l_1 yields upper bound $x < -4$, l_0 yields lower bound $x > -2$
- Clash of bounds suggests to infer $l_0 + 2l_1$, i.e. $l_4 : (y < 0)$
  indeed violated by the guess $y \leftarrow 4$
- Undo guess, keep l_4
  l_3 and l_4 give clash of bounds for y
- Suggests to infer $l_3 + l_4$, i.e. $l_5 : 0 < -2$

An example in Linear Rational Arithmetic

$l_0 : (-2\cdot x - y < 0)$, $l_1 : (x + y < 0)$, $l_2 : (x < -1)$

unsatisfiable in LRA. Example of MCSAT run:

- Guess a value, e.g. $y \leftarrow 0$
  Then l_0 yields lower bound $x > 0$
  Together with l_2, space of possible values for x is empty
  What to do? just undo $y \leftarrow 0$? No:
- Clash of bounds suggests to infer $l_0 + 2l_2$, i.e. $l_3 : (-y < -2)$
  indeed violated by the guess $y \leftarrow 0$
- Now undo the guess but keep l_3.
- Try new guess, say $y \leftarrow 4$
  l_1 yields upper bound $x < -4$, l_0 yields lower bound $x > -2$
- Clash of bounds suggests to infer $l_0 + 2l_1$, i.e. $l_4 : (y < 0)$
  indeed violated by the guess $y \leftarrow 4$
- Undo guess, keep l_4
  l_3 and l_4 give clash of bounds for y
- Suggests to infer $l_3 + l_4$, i.e. $l_5 : 0 < -2$
  No guess to undo, problem is UNSAT

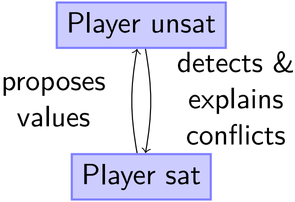
An example in Linear Rational Arithmetic

$l_0 : (-2{\cdot}x - y < 0)$, $l_1 : (x + y < 0)$, $l_2 : (x < -1)$

unsatisfiable in LRA. Example of MCSAT run:

- **Guess a value**, e.g. $y \leftarrow 0$
  Then l_0 yields lower bound $x > 0$
  Together with l_2, space of possible values for x is empty
  What to do? just undo $y \leftarrow 0$? No:
- Clash of bounds suggests to **infer** $l_0 + 2l_2$, i.e. $l_3 : (-y < -2)$
  indeed violated by the guess $y \leftarrow 0$
- Now **undo the guess** but keep l_3.
- Try new **guess**, say $y \leftarrow 4$
  l_1 yields upper bound $x < -4$, l_0 yields lower bound $x > -2$
- Clash of bounds suggests to **infer** $l_0 + 2l_1$, i.e. $l_4 : (y < 0)$
  indeed violated by the guess $y \leftarrow 4$
- **Undo guess**, keep l_4
  l_3 and l_4 give clash of bounds for y
- Suggests to **infer** $l_3 + l_4$, i.e. $l_5 : 0 < -2$
  No guess to undo, problem is UNSAT

Key ingredients of MC-Sat calculi



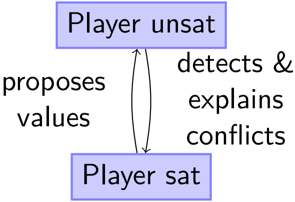
Player unsat

proposes values

detects & explains conflicts

Player sat

Key ingredients of MC-Sat calculi



Player unsat

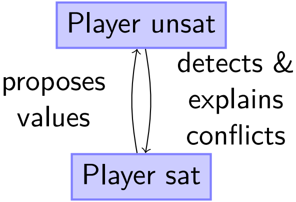
proposes values

detects & explains conflicts

Player sat

- Ability to make guesses
  that do not “immediately violate” currently known constraints.
  Here we can make such guesses up until there is a bound clash

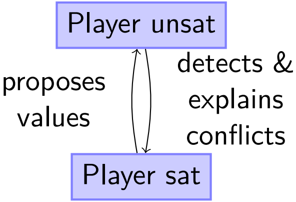
Key ingredients of MC-Sat calculi



- Ability to make guesses
  that do not "immediately violate" currently known constraints.
  Here we can make such guesses up until there is a bound clash
- When undoing a guess, "something new" must be learnt that at least prevents the same guess from being made again.
  With infinite domains (e.g. $\mathbb{Q}$) the "something new" must definitely reject more than 1 value.
  Here we used Fourier-Motzkin resolutions:

$$(e_1 < x), (x < e_2) \vdash_{\mathsf{LRA}} (e_1 < e_2)$$

Key ingredients of MC-Sat calculi



Player unsat

proposes values

detects & explains conflicts

Player sat

- Ability to make guesses that do not “immediately violate” currently known constraints. Here we can make such guesses up until there is a bound clash
- When undoing a guess, “something new” must be learnt that at least prevents the same guess from being made again. With infinite domains (e.g. $\mathbb{Q}$) the “something new” must definitely reject more than 1 value. Here we used Fourier-Motzkin resolutions:

$$(e_1 < x), (x < e_2) \vdash_{\mathsf{LRA}} (e_1 < e_2)$$

- Some generic mechanism to expand trails and analyse conflicts

Subtleties

- New literals are introduced during a run (here l_3 and l_4 by FM-resolutions)

$$
\begin{array}{lll}
l_0 : & -2{\cdot}x - y < 0 & \\
l_1 : & x + y < 0 & \\
l_2 : & x < -1 & \\
l_3 : & -y < -2 & (l_0 + 2l_2) \\
l_4 : & x < -2 & (l_1 + l_3)
\end{array}
$$

Subtleties

- New literals are introduced during a run (here l_3 and l_4 by FM-resolutions)

  ☠

  This opens the door of non-termination (infinitely many new things can be learnt)

$$
\begin{array}{lll}
l_0: & -2{\cdot}x - y < 0 & \\
l_1: & x + y < 0 & \\
l_2: & x < -1 & \\
l_3: & -y < -2 & (l_0 + 2l_2) \\
l_4: & x < -2 & (l_1 + l_3)
\end{array}
$$

Subtleties

- New literals are introduced during a run (here l_3 and l_4 by FM-resolutions)

  ☠

  This opens the door of non-termination (infinitely many new things can be learnt)

$$\begin{array}{lrl}
l_0 : & -2\cdot x - y < 0 & \\
l_1 : & x + y < 0 & \\
l_2 : & x < -1 & \\
l_3 : & -y < -2 & (l_0 + 2l_2) \\
l_4 : & x < -2 & (l_1 + l_3) \\
l_5 : & -y < -4 & (l_0 + 2l_4)
\end{array}$$

Subtleties

- New literals are introduced during a run (here l_3 and l_4 by FM-resolutions)

  ☠

  This opens the door of non-termination (infinitely many new things can be learnt)

$$
\begin{array}{lll}
l_0 : & -2{\cdot}x - y < 0 & \\
l_1 : & x + y < 0 & \\
l_2 : & x < -1 & \\
l_3 : & -y < -2 & (l_0 + 2l_2) \\
l_4 : & x < -2 & (l_1 + l_3) \\
l_5 : & -y < -4 & (l_0 + 2l_4) \\
l_6 : & x < -4 & (l_1 + l_5)
\end{array}
$$

Subtleties

- New literals are introduced during a run (here l_3 and l_4 by FM-resolutions)

  ☠

  This opens the door of non-termination (infinitely many new things can be learnt)

$$
\begin{array}{lll}
l_0 : & -2{\cdot}x - y < 0 & \\
l_1 : & x + y < 0 & \\
l_2 : & x < -1 & \\
l_3 : & -y < -2 & (l_0 + 2l_2) \\
l_4 : & x < -2 & (l_1 + l_3) \\
l_5 : & -y < -4 & (l_0 + 2l_4) \\
l_6 : & x < -4 & (l_1 + l_5) \\
l_7 : & -y < -8 & (l_0 + 2l_6) \\
\dots & &
\end{array}
$$

Subtleties

- New literals are introduced during a run (here l_3 and l_4 by FM-resolutions)

  ☠

  This opens the door of non-termination (infinitely many new things can be learnt)

$$
\begin{array}{lll}
l_0: & -2\cdot x - y < 0 & \\
l_1: & x + y < 0 & \\
l_2: & x < -1 & \\
l_3: & -y < -2 & (l_0 + 2l_2) \\
l_4: & x < -2 & (l_1 + l_3) \\
l_5: & -y < -4 & (l_0 + 2l_4) \\
l_6: & x < -4 & (l_1 + l_5) \\
l_7: & -y < -8 & (l_0 + 2l_6) \\
\dots & &
\end{array}
$$

- Even if non-termination is avoided, introducing new material should we done with parsimony

Subtleties

- New literals are introduced during a run (here l_3 and l_4 by FM-resolutions)

  ☠

  This opens the door of non-termination (infinitely many new things can be learnt)

$$
\begin{array}{lll}
l_0 : & -2{\cdot}x - y < 0 & \\
l_1 : & x + y < 0 & \\
l_2 : & x < -1 & \\
l_3 : & -y < -2 & (l_0 + 2l_2) \\
l_4 : & x < -2 & (l_1 + l_3) \\
l_5 : & -y < -4 & (l_0 + 2l_4) \\
l_6 : & x < -4 & (l_1 + l_5) \\
l_7 : & -y < -8 & (l_0 + 2l_6) \\
\ldots & &
\end{array}
$$

- Even if non-termination is avoided,
  introducing new material should we done with parsimony
  Important aspect of MC-Sat is laziness.
  FM-resolution only introduced to learn something from bound clashes

Subtleties

- New literals are introduced during a run (here l_3 and l_4 by FM-resolutions)

  ☠

  This opens the door of non-termination (infinitely many new things can be learnt)

$$\begin{array}{lll}
l_0 : & -2 \cdot x - y < 0 & \\
l_1 : & x + y < 0 & \\
l_2 : & x < -1 & \\
l_3 : & -y < -2 & (l_0 + 2 l_2) \\
l_4 : & x < -2 & (l_1 + l_3) \\
l_5 : & -y < -4 & (l_0 + 2 l_4) \\
l_6 : & x < -4 & (l_1 + l_5) \\
l_7 : & -y < -8 & (l_0 + 2 l_6) \\
\dots & &
\end{array}$$

- Even if non-termination is avoided, introducing new material should we done with parsimony

  Important aspect of MC-Sat is laziness.

  FM-resolution only introduced to learn something from bound clashes

  More generally, Player Unsat can afford being lazy, and react only when sufficiently many terms have been assigned semantics.

Subtleties

- New literals are introduced during a run (here l_3 and l_4 by FM-resolutions)

  ☠

  This opens the door of non-termination (infinitely many new things can be learnt)

$$\begin{array}{lll}
l_0 : & -2{\cdot}x - y < 0 & \\
l_1 : & x + y < 0 & \\
l_2 : & x < -1 & \\
l_3 : & -y < -2 & (l_0 + 2l_2) \\
l_4 : & x < -2 & (l_1 + l_3) \\
l_5 : & -y < -4 & (l_0 + 2l_4) \\
l_6 : & x < -4 & (l_1 + l_5) \\
l_7 : & -y < -8 & (l_0 + 2l_6) \\
\ldots & &
\end{array}$$

- Even if non-termination is avoided, introducing new material should we done with parsimony Important aspect of MC-Sat is laziness. FM-resolution only introduced to learn something from bound clashes More generally, Player Unsat can afford being lazy, and react only when sufficiently many terms have been assigned semantics.

  DPLL's 2-watched literals technique (detecting when to apply Boolean propagation) generalises to n-watched literals & can be used in each theory.

Contributions

- In [dMJ13], Boolean logic + one abstract theory $\mathcal{T}$ (mimicking DPLL($\mathcal{T}$))

Contributions

- In [dMJ13], Boolean logic + one abstract theory $\mathcal{T}$ (mimicking DPLL($\mathcal{T}$))
- In [JBdM13]: Boolean+LRA+EUF

Contributions

- In [dMJ13], Boolean logic + one abstract theory $\mathcal{T}$ (mimicking DPLL($\mathcal{T}$))
- In [JBdM13]: Boolean+LRA+EUF
- Other contributions, for Bit vectors [ZWR16], Nonlinear Real Arithmetic [JdM12], Nonlinear Integer Arithmetic [Jov17], ...

Contributions

- In [dMJ13], Boolean logic + one abstract theory $\mathcal{T}$ (mimicking DPLL($\mathcal{T}$))
- In [JBdM13]: Boolean+LRA+EUF
- Other contributions, for Bit vectors [ZWR16], Nonlinear Real Arithmetic [JdM12], Nonlinear Integer Arithmetic [Jov17], ...

This raises the questions:

- Is there a generic way to combine à la MCSAT several abstract theories? Which requirements should the theory reasoning mechanisms satisfy for the combined system to be sound, complete, and terminating?

Contributions

- In [dMJ13], Boolean logic + one abstract theory $\mathcal{T}$ (mimicking DPLL($\mathcal{T}$))
- In [JBdM13]: Boolean+LRA+EUF
- Other contributions, for Bit vectors [ZWR16], Nonlinear Real Arithmetic [JdM12], Nonlinear Integer Arithmetic [Jov17], ...

This raises the questions:

- Is there a generic way to combine à la MCSAT several abstract theories? Which requirements should the theory reasoning mechanisms satisfy for the combined system to be sound, complete, and terminating?
- Is there a way to integrate or generalize both MCSAT and Nelson-Oppen scheme (equality sharing)?

MP Bonacina, N Shankar and SGL address this for disjoint theories in [BGLS16]

2. MC-Sat mechanisms in our formal framework

Same example formalized in our formal framework

Trail = stack of assignments $(t \leftarrow v)$ + "explanation function", initialized with input problem

$(l \leftarrow \text{true})$ abbrev. as l

Empty explanation for input problem

id	trail items	expl.
0	$-2 \cdot x - y < 0$	$\{\}$
1	$x + y < 0$	$\{\}$
2	$x < -1$	$\{\}$

Same example formalized in our formal framework

Trail = stack of assignments $(t \leftarrow v)$ + "explanation function", initialized with input problem

$(l \leftarrow \text{true})$ abbrev. as l

Empty explanation for input problem

Level:
greatest decision involved

id	trail items	expl.	lev.
0	$-2 \cdot x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$y \leftarrow 0$		1

Same example formalized in our formal framework

Trail $=$ stack of assignments $(t \leftarrow v)$ $+$ “explanation function”, initialized with input problem

$(l \leftarrow \text{true})$ abbrev. as l

Empty explanation for input problem

Level:

greatest decision involved

id	trail items	expl.	lev.
0	$-2 \cdot x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$y \leftarrow 0$		1
4	$-y < -2$	$\{0, 2\}$	0

Same example formalized in our formal framework

Trail = stack of assignments $(t \leftarrow v)$ + "explanation function", initialized with input problem

$(l \leftarrow \text{true})$ abbrev. as l

Empty explanation for input problem

Level:
greatest decision involved

If conflict is of level 0...
...problem is unsat

id	trail items	expl.	lev.
0	$-2 \cdot x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$\mathbf{y \leftarrow 0}$		1
4	$-\mathbf{y} < -\mathbf{2}$	$\{0, 2\}$	0
	conflict E^1: $\{3, 4\}$		1

Same example formalized in our formal framework

Trail = stack of assignments $(t \leftarrow v)$ + “explanation function”, initialized with input problem

$(l \leftarrow \text{true})$ abbrev. as l
Empty explanation for input problem

Level:
greatest decision involved

If conflict is of level 0...
...problem is unsat

Phase 1

id	trail items	expl.	lev.
0	$-2 \cdot x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$\mathbf{y \leftarrow 0}$		1
4	$\mathbf{-y < -2}$	$\{0, 2\}$	0
	conflict E^1: $\{3, 4\}$		1

Phase 2

id	trail items	expl.	lev.
0	$-2 \cdot x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$-y < -2$	$\{0, 2\}$	0

Same example formalized in our formal framework

Trail = stack of assignments $(t \leftarrow v)$ + "explanation function", initialized with input problem

$(l \leftarrow \text{true})$ abbrev. as l

Empty explanation for input problem

Level:

greatest decision involved

If conflict is of level 0...

...problem is unsat

Phase 1

id	trail items	expl.	lev.
0	$-2 \cdot x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$\mathbf{y \leftarrow 0}$		1
4	$\mathbf{-y < -2}$	$\{0, 2\}$	0
	conflict E^1: $\{3, 4\}$		1

Phase 2

id	trail items	expl.	lev.
0	$-2 \cdot x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$-y < -2$	$\{0, 2\}$	0
4	$y \leftarrow 4$		1

Same example formalized in our formal framework

Trail = stack of assignments $(t \leftarrow v)$ + "explanation function", initialized with input problem

$(l \leftarrow \text{true})$ abbrev. as l

Empty explanation for input problem

Level:

greatest decision involved

If conflict is of level 0…

…problem is unsat

Phase 1

id	trail items	expl.	lev.
0	$-2 \cdot x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$\mathbf{y \leftarrow 0}$		1
4	$\mathbf{-y < -2}$	$\{0, 2\}$	0
	conflict E^1: $\{3, 4\}$		1

Phase 2

id	trail items	expl.	lev.
0	$-2 \cdot x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$-y < -2$	$\{0, 2\}$	0
4	$y \leftarrow 4$		1
5	$y < 0$	$\{0, 1\}$	0

Same example formalized in our formal framework

Trail = stack of assignments $(t \leftarrow v)$ + "explanation function", initialized with input problem

$(l \leftarrow \text{true})$ abbrev. as l

Empty explanation for input problem

Level:

greatest decision involved

If conflict is of level 0...

...problem is unsat

Phase 1

id	trail items	expl.	lev.
0	$-2{\cdot}x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$\mathbf{y \leftarrow 0}$		1
4	$\mathbf{-y < -2}$	$\{0, 2\}$	0
	conflict E^1: $\{3, 4\}$		1

Phase 2

id	trail items	expl.	lev.
0	$-2{\cdot}x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$-y < -2$	$\{0, 2\}$	0
4	$\mathbf{y \leftarrow 4}$		1
5	$\mathbf{y < 0}$	$\{0, 1\}$	0
	conflict E^2: $\{4, 5\}$		1

Same example formalized in our formal framework

Trail = stack of assignments $(t \leftarrow v)$ + “explanation function”, initialized with input problem

$(l \leftarrow \text{true})$ abbrev. as l
Empty explanation for input problem

Level:
greatest decision involved

If conflict is of level 0…
…problem is unsat

Phase 1

id	trail items	expl.	lev.
0	$-2{\cdot}x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$\mathbf{y \leftarrow 0}$		1
4	$\mathbf{-y < -2}$	$\{0, 2\}$	0
	conflict E^1: $\{3, 4\}$		1

Phase 2

id	trail items	expl.	lev.
0	$-2{\cdot}x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$-y < -2$	$\{0, 2\}$	0
4	$\mathbf{y \leftarrow 4}$		1
5	$\mathbf{y < 0}$	$\{0, 1\}$	0
	conflict E^2: $\{4, 5\}$		1

Phase 3

id	trail items	expl.	lev.
0	$-2{\cdot}x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$-y < -2$	$\{0, 2\}$	0
4	$y < 0$	$\{0, 1\}$	0

Same example formalized in our formal framework

Trail = stack of assignments $(t \leftarrow v)$ + "explanation function", initialized with input problem

$(l \leftarrow \text{true})$ abbrev. as l

Empty explanation for input problem

Level:

greatest decision involved

If conflict is of level 0. . .

. . . problem is unsat

Phase 1

id	trail items	expl.	lev.
0	$-2 \cdot x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$\mathbf{y \leftarrow 0}$		1
4	$\mathbf{-y < -2}$	$\{0, 2\}$	0
	conflict E^1: $\{3, 4\}$		1

Phase 2

id	trail items	expl.	lev.
0	$-2 \cdot x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$-y < -2$	$\{0, 2\}$	0
4	$\mathbf{y \leftarrow 4}$		1
5	$\mathbf{y < 0}$	$\{0, 1\}$	0
	conflict E^2: $\{4, 5\}$		1

Phase 3

id	trail items	expl.	lev.
0	$-2 \cdot x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$-y < -2$	$\{0, 2\}$	0
4	$y < 0$	$\{0, 1\}$	0
5	$0 < -2$	$\{3, 4\}$	0

Same example formalized in our formal framework

Trail = stack of assignments $(t \leftarrow v)$ + "explanation function", initialized with input problem

$(l \leftarrow \text{true})$ abbrev. as l

Empty explanation for input problem

Level:

greatest decision involved

If conflict is of level 0…

…problem is unsat

Phase 1

id	trail items	expl.	lev.
0	$-2{\cdot}x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$\mathbf{y \leftarrow 0}$		1
4	$\mathbf{-y < -2}$	$\{0, 2\}$	0
	conflict E^1: $\{3, 4\}$		1

Phase 2

id	trail items	expl.	lev.
0	$-2{\cdot}x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$-y < -2$	$\{0, 2\}$	0
4	$\mathbf{y \leftarrow 4}$		1
5	$\mathbf{y < 0}$	$\{0, 1\}$	0
	conflict E^2: $\{4, 5\}$		1

Phase 3

id	trail items	expl.	lev.
0	$-2{\cdot}x - y < 0$	$\{\}$	0
1	$x + y < 0$	$\{\}$	0
2	$x < -1$	$\{\}$	0
3	$-y < -2$	$\{0, 2\}$	0
4	$y < 0$	$\{0, 1\}$	0
5	$\mathbf{0 < -2}$	$\{3, 4\}$	0
	conflict E^3: $\{5\}$		0

An example with arithmetic, arrays, congruence

$$f(a[i{:=}\,v][j]) \simeq w\ ,\ w-2 \simeq f(u)\ ,\ i \simeq j\ ,\ u \simeq v$$

Phase 1

id	trail items	expl.	lev.
0	$f(x) \simeq w$	{}	0
1	$y \simeq f(u)$	{}	0
2	$w-2 \simeq y$	{}	0
3	$a[i{:=}\,v][j] \simeq x$	{}	0
4	$i \simeq j$	{}	0
5	$u \simeq v$	{}	0

An example with arithmetic, arrays, congruence

$$f(a[i{:=}\,v][j]) \simeq w\ ,\ \ w-2 \simeq f(u)\ ,\ \ i \simeq j\ ,\ \ u \simeq v$$

Phase 1

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w-2 \simeq y$	$\{\}$	0
3	$a[i{:=}\,v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v{\leftarrow}\mathfrak{c}$		1

An example with arithmetic, arrays, congruence

$$f(a[i{:=}\ v][j]) \simeq w\ ,\ \ w-2 \simeq f(u)\ ,\ \ i \simeq j\ ,\ \ u \simeq v$$

Phase 1

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w-2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \leftarrow \mathfrak{c}$		1
7	$a[i{:=}\ v][j] \leftarrow \mathfrak{d}$		2

An example with arithmetic, arrays, congruence

$$f(a[i{:=}\ v][j]) \simeq w\ ,\ \ w - 2 \simeq f(u)\ ,\ \ i \simeq j\ ,\ \ u \simeq v$$

	Phase 1		
id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v{\leftarrow}\mathfrak{c}$		1
7	$a[i{:=}\ v][j]{\leftarrow}\mathfrak{d}$		2
8	$v \not\simeq a[i{:=}\ v][j]$	$\{6, 7\}$	2

An example with arithmetic, arrays, congruence

$$f(a[i:=v][j]) \simeq w\ ,\ w-2 \simeq f(u)\ ,\ i \simeq j\ ,\ u \simeq v$$

	Phase 1		
id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w-2 \simeq y$	$\{\}$	0
3	$a[i:=v][j] \simeq x$	$\{\}$	0
4	$\mathbf{i} \simeq \mathbf{j}$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \leftarrow \mathfrak{c}$		1
7	$a[i:=v][j] \leftarrow \mathfrak{d}$		2
8	$\mathbf{v} \not\simeq \mathbf{a}[\mathbf{i}:=\mathbf{v}][\mathbf{j}]$	$\{6,7\}$	2
	conflict E^1: $\{4,8\}$		2

An example with arithmetic, arrays, congruence

$$f(a[i{:=}\ v][j]) \simeq w\ ,\ \ w - 2 \simeq f(u)\ ,\ \ i \simeq j\ ,\ \ u \simeq v$$

Phase 1

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$\mathbf{i} \simeq \mathbf{j}$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \leftarrow \mathfrak{c}$		1
7	$a[i{:=}\ v][j] \leftarrow \mathfrak{d}$		2
8	$\mathbf{v} \not\simeq \mathbf{a}[\mathbf{i}{:=}\ \mathbf{v}][\mathbf{j}]$	$\{6, 7\}$	2
	conflict E^1: $\{4, 8\}$		2

Phase 2

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0

An example with arithmetic, arrays, congruence

Phase 2

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0

An example with arithmetic, arrays, congruence

Phase 2

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u \leftarrow \mathfrak{c}$		1

An example with arithmetic, arrays, congruence

Phase 2

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u \leftarrow \mathfrak{c}$		1
8	$x \leftarrow \mathfrak{c}$		2

An example with arithmetic, arrays, congruence

Phase 2

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u {\leftarrow} \mathfrak{c}$		1
8	$x {\leftarrow} \mathfrak{c}$		2
9	$w {\leftarrow} 0$		3

An example with arithmetic, arrays, congruence

Phase 2

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u \leftarrow \mathfrak{c}$		1
8	$x \leftarrow \mathfrak{c}$		2
9	$w \leftarrow 0$		3
10	$y \leftarrow -2$		4

An example with arithmetic, arrays, congruence

Phase 2

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u \leftarrow \mathfrak{c}$		1
8	$x \leftarrow \mathfrak{c}$		2
9	$w \leftarrow 0$		3
10	$y \leftarrow -2$		4
11	$y \not\simeq w$	$\{9, 10\}$	4

An example with arithmetic, arrays, congruence

Phase 2

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u{\leftarrow}\mathfrak{c}$		1
8	$x{\leftarrow}\mathfrak{c}$		2
9	$w{\leftarrow}0$		3
10	$y{\leftarrow} -2$		4
11	$y \not\simeq w$	$\{9, 10\}$	4
12	$u \simeq x$	$\{7, 8\}$	2

An example with arithmetic, arrays, congruence

Phase 2

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u \leftarrow \mathfrak{c}$		1
8	$x \leftarrow \mathfrak{c}$		2
9	$w \leftarrow 0$		3
10	$y \leftarrow -2$		4
11	$y \not\simeq w$	$\{9, 10\}$	4
12	$u \simeq x$	$\{7, 8\}$	2
13	$f(u) \simeq f(x)$	$\{12\}$	2

An example with arithmetic, arrays, congruence

Phase 2

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u{\leftarrow}\mathfrak{c}$		1
8	$x{\leftarrow}\mathfrak{c}$		2
9	$w{\leftarrow}0$		3
10	$y{\leftarrow} -2$		4
11	$y \not\simeq w$	$\{9, 10\}$	4
12	$u \simeq x$	$\{7, 8\}$	2
13	$f(u) \simeq f(x)$	$\{12\}$	2
14	$f(u) \simeq w$	$\{0, 13\}$	2

An example with arithmetic, arrays, congruence

Phase 2

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$\mathbf{y} \simeq \mathbf{f}(\mathbf{u})$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u {\leftarrow} \mathfrak{c}$		1
8	$x {\leftarrow} \mathfrak{c}$		2
9	$w {\leftarrow} 0$		3
10	$y {\leftarrow} -2$		4
11	$\mathbf{y} \not\simeq \mathbf{w}$	$\{9, 10\}$	4
12	$u \simeq x$	$\{7, 8\}$	2
13	$f(u) \simeq f(x)$	$\{12\}$	2
14	$\mathbf{f}(\mathbf{u}) \simeq \mathbf{w}$	$\{0, 13\}$	2
	conflict E^2: $\{1, 11, 14\}$		4

An example with arithmetic, arrays, congruence

Phase 2

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$\mathbf{y} \simeq \mathbf{f}(\mathbf{u})$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u{\leftarrow}\mathfrak{c}$		1
8	$x{\leftarrow}\mathfrak{c}$		2
9	$w{\leftarrow}0$		3
10	$y{\leftarrow}-2$		4
11	$\mathbf{y} \not\simeq \mathbf{w}$	$\{9, 10\}$	4
12	$u \simeq x$	$\{7, 8\}$	2
13	$f(u) \simeq f(x)$	$\{12\}$	2
14	$\mathbf{f}(\mathbf{u}) \simeq \mathbf{w}$	$\{0, 13\}$	2
	conflict E^2: $\{1, 11, 14\}$		4

Phase 3

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u{\leftarrow}\mathfrak{c}$		1
8	$x{\leftarrow}\mathfrak{c}$		2
9	$u \simeq x$	$\{7, 8\}$	2
10	$f(u) \simeq f(x)$	$\{9\}$	2
11	$f(u) \simeq w$	$\{0, 10\}$	2
12	$y \simeq w$	$\{1, 11\}$	2

An example with arithmetic, arrays, congruence

Phase 3

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u{\leftarrow}\mathfrak{c}$		1
8	$x{\leftarrow}\mathfrak{c}$		2
9	$u \simeq x$	$\{7, 8\}$	2
10	$f(u) \simeq f(x)$	$\{9\}$	2
11	$f(u) \simeq w$	$\{0, 10\}$	2
12	$y \simeq w$	$\{1, 11\}$	2

An example with arithmetic, arrays, congruence

Phase 3

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u{\leftarrow}\mathfrak{c}$		1
8	$x{\leftarrow}\mathfrak{c}$		2
9	$u \simeq x$	$\{7, 8\}$	2
10	$f(u) \simeq f(x)$	$\{9\}$	2
11	$f(u) \simeq w$	$\{0, 10\}$	2
12	$y \simeq w$	$\{1, 11\}$	2
13	$w - 2 \simeq w$	$\{2, 12\}$	2

An example with arithmetic, arrays, congruence

Phase 3

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u \leftarrow \mathfrak{c}$		1
8	$x \leftarrow \mathfrak{c}$		2
9	$u \simeq x$	$\{7, 8\}$	2
10	$f(u) \simeq f(x)$	$\{9\}$	2
11	$f(u) \simeq w$	$\{0, 10\}$	2
12	$y \simeq w$	$\{1, 11\}$	2
13	$\mathbf{w - 2 \simeq w}$	$\{2, 12\}$	2
	conflict E_1^3: $\{13\}$		2

An example with arithmetic, arrays, congruence

Phase 3

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$\mathbf{w - 2 \simeq y}$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u{\leftarrow}\mathfrak{c}$		1
8	$x{\leftarrow}\mathfrak{c}$		2
9	$u \simeq x$	$\{7, 8\}$	2
10	$f(u) \simeq f(x)$	$\{9\}$	2
11	$f(u) \simeq w$	$\{0, 10\}$	2
12	$\mathbf{y \simeq w}$	$\{1, 11\}$	2
13	$w - 2 \simeq w$	$\{2, 12\}$	2
	conflict E_1^3: $\{13\}$		2
	conflict E_2^3: $\{2, 12\}$		2

An example with arithmetic, arrays, congruence

Phase 3

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$\mathbf{y \simeq f(u)}$	$\{\}$	0
2	$\mathbf{w - 2 \simeq y}$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u \leftarrow \mathfrak{c}$		1
8	$x \leftarrow \mathfrak{c}$		2
9	$u \simeq x$	$\{7, 8\}$	2
10	$f(u) \simeq f(x)$	$\{9\}$	2
11	$\mathbf{f(u) \simeq w}$	$\{0, 10\}$	2
12	$y \simeq w$	$\{1, 11\}$	2
13	$w - 2 \simeq w$	$\{2, 12\}$	2
	conflict E_1^3: $\{13\}$		2
	conflict E_2^3: $\{2, 12\}$		2
	conflict E_3^3: $\{1, 2, 11\}$		2

An example with arithmetic, arrays, congruence

Phase 3

id	trail items	expl.	lev.
0	$\mathbf{f(x) \simeq w}$	$\{\}$	0
1	$\mathbf{y \simeq f(u)}$	$\{\}$	0
2	$\mathbf{w - 2 \simeq y}$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u{\leftarrow}\mathfrak{c}$		1
8	$x{\leftarrow}\mathfrak{c}$		2
9	$u \simeq x$	$\{7, 8\}$	2
10	$\mathbf{f(u) \simeq f(x)}$	$\{9\}$	2
11	$f(u) \simeq w$	$\{0, 10\}$	2
12	$y \simeq w$	$\{1, 11\}$	2
13	$w - 2 \simeq w$	$\{2, 12\}$	2
	conflict E_1^3: $\{13\}$		2
	conflict E_2^3: $\{2, 12\}$		2
	conflict E_3^3: $\{1, 2, 11\}$		2
	conflict E_4^3: $\{0, 1, 2, 10\}$		2

An example with arithmetic, arrays, congruence

Phase 3

id	trail items	expl.	lev.
0	$\mathbf{f(x) \simeq w}$	$\{\}$	0
1	$\mathbf{y \simeq f(u)}$	$\{\}$	0
2	$\mathbf{w - 2 \simeq y}$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u{\leftarrow}\mathfrak{c}$		1
8	$x{\leftarrow}\mathfrak{c}$		2
9	$\mathbf{u \simeq x}$	$\{7,8\}$	2
10	$f(u) \simeq f(x)$	$\{9\}$	2
11	$f(u) \simeq w$	$\{0,10\}$	2
12	$y \simeq w$	$\{1,11\}$	2
13	$w - 2 \simeq w$	$\{2,12\}$	2
	conflict E_1^3: $\{13\}$		2
	conflict E_2^3: $\{2,12\}$		2
	conflict E_3^3: $\{1,2,11\}$		2
	conflict E_4^3: $\{0,1,2,10\}$		2
	conflict E_5^3: $\{0,1,2,9\}$		2

An example with arithmetic, arrays, congruence

Phase

id	trail items	expl.	lev.
0	$\mathbf{f(x) \simeq w}$	$\{\}$	0
1	$\mathbf{y \simeq f(u)}$	$\{\}$	0
2	$\mathbf{w - 2 \simeq y}$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u{\leftarrow}\mathfrak{c}$		1
8	$x{\leftarrow}\mathfrak{c}$		2
9	$\mathbf{u \simeq x}$	$\{7, 8\}$	2
10	$f(u) \simeq f(x)$	$\{9\}$	2
11	$f(u) \simeq w$	$\{0, 10\}$	2
12	$y \simeq w$	$\{1, 11\}$	2
13	$w - 2 \simeq w$	$\{2, 12\}$	2
	conflict E^3_1: $\{13\}$		2
	conflict E^3_2: $\{2, 12\}$		2
	conflict E^3_3: $\{1, 2, 11\}$		2
	conflict E^3_4: $\{0, 1, 2, 10\}$		2
	conflict E^3_5: $\{0, 1, 2, 9\}$		2

Phase 4

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u \not\simeq x$	$\{0, 1, 2\}$	0

An example with arithmetic, arrays, congruence

Phase

id	trail items	expl.	lev.
0	$\mathbf{f(x) \simeq w}$	$\{\}$	0
1	$\mathbf{y \simeq f(u)}$	$\{\}$	0
2	$\mathbf{w - 2 \simeq y}$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u \leftarrow \mathfrak{c}$		1
8	$x \leftarrow \mathfrak{c}$		2
9	$\mathbf{u \simeq x}$	$\{7, 8\}$	2
10	$f(u) \simeq f(x)$	$\{9\}$	2
11	$f(u) \simeq w$	$\{0, 10\}$	2
12	$y \simeq w$	$\{1, 11\}$	2
13	$w - 2 \simeq w$	$\{2, 12\}$	2
	conflict E_1^3: $\{13\}$		2
	conflict E_2^3: $\{2, 12\}$		2
	conflict E_3^3: $\{1, 2, 11\}$		2
	conflict E_4^3: $\{0, 1, 2, 10\}$		2
	conflict E_5^3: $\{0, 1, 2, 9\}$		2

Phase 4

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u \not\simeq x$	$\{0, 1, 2\}$	0
8	$v \simeq x$	$\{3, 6\}$	0

An example with arithmetic, arrays, congruence

Phase

id	trail items	expl.	lev.
0	$\mathbf{f(x) \simeq w}$	$\{\}$	0
1	$\mathbf{y \simeq f(u)}$	$\{\}$	0
2	$\mathbf{w - 2 \simeq y}$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$u \simeq v$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$u \leftarrow \mathfrak{c}$		1
8	$x \leftarrow \mathfrak{c}$		2
9	$\mathbf{u \simeq x}$	$\{7, 8\}$	2
10	$f(u) \simeq f(x)$	$\{9\}$	2
11	$f(u) \simeq w$	$\{0, 10\}$	2
12	$y \simeq w$	$\{1, 11\}$	2
13	$w - 2 \simeq w$	$\{2, 12\}$	2
	conflict E_1^3: $\{13\}$		2
	conflict E_2^3: $\{2, 12\}$		2
	conflict E_3^3: $\{1, 2, 11\}$		2
	conflict E_4^3: $\{0, 1, 2, 10\}$		2
	conflict E_5^3: $\{0, 1, 2, 9\}$		2

Phase 4

id	trail items	expl.	lev.
0	$f(x) \simeq w$	$\{\}$	0
1	$y \simeq f(u)$	$\{\}$	0
2	$w - 2 \simeq y$	$\{\}$	0
3	$a[i{:=}\ v][j] \simeq x$	$\{\}$	0
4	$i \simeq j$	$\{\}$	0
5	$\mathbf{u \simeq v}$	$\{\}$	0
6	$v \simeq a[i{:=}\ v][j]$	$\{4\}$	0
7	$\mathbf{u \not\simeq x}$	$\{0, 1, 2\}$	0
8	$\mathbf{v \simeq x}$	$\{3, 6\}$	0
	conflict E^4: $\{5, 7, 8\}$		0

Theory-specific ingredients: $\mathcal{T}$-modules

Given a theory $\mathcal{T}$, a module for $\mathcal{T}$ identifies:

- A collection of sorts for which it will propose values; e.g. sort Q for LRA. These sorts are $\mathcal{T}$-public.

Theory-specific ingredients: $\mathcal{T}$-modules

Given a theory $\mathcal{T}$, a module for $\mathcal{T}$ identifies:

- A collection of sorts for which it will propose values; e.g. sort Q for LRA. These sorts are $\mathcal{T}$-public.
- A collection of values for those sorts, and an extension $\mathcal{T}^+$ of theory $\mathcal{T}$ on extended signature e.g. to specify, when writing $x \leftarrow \sqrt{2}$, that $\sqrt{2} \times \sqrt{2} = 1 + 1$.

Theory-specific ingredients: $\mathcal{T}$-modules

Given a theory $\mathcal{T}$, a module for $\mathcal{T}$ identifies:

- A collection of sorts for which it will propose values; e.g. sort Q for LRA. These sorts are $\mathcal{T}$-public.
- A collection of values for those sorts, and an extension $\mathcal{T}^+$ of theory $\mathcal{T}$ on extended signature e.g. to specify, when writing $x{\leftarrow}\sqrt{2}$, that $\sqrt{2}\times\sqrt{2} = 1 + 1$.
- A collection of $\mathcal{T}$-inferences of the form $J \vdash_{\mathcal{T}} L$, where J is made of Boolean or non-Boolean assignments, and L is a Boolean assignment.

Theory-specific ingredients: $\mathcal{T}$-modules

Given a theory $\mathcal{T}$, a module for $\mathcal{T}$ identifies:

- A collection of sorts for which it will propose values; e.g. sort $\mathbb{Q}$ for LRA. These sorts are $\mathcal{T}$-public.
- A collection of values for those sorts, and an extension $\mathcal{T}^+$ of theory $\mathcal{T}$ on extended signature e.g. to specify, when writing $x \leftarrow \sqrt{2}$, that $\sqrt{2} \times \sqrt{2} = 1 + 1$.
- A collection of $\mathcal{T}$-inferences of the form $J \vdash_{\mathcal{T}} L$, where J is made of Boolean or non-Boolean assignments, and L is a Boolean assignment.

We add to these inference equality inferences

$$\begin{array}{ll} (t_1 \leftarrow v_1), (t_2 \leftarrow v_2) \vdash t_1 \simeq_s t_2 & \text{if } v_1 \text{ and } v_2 \text{ are the same} \\ (t_1 \leftarrow v_1), (t_2 \leftarrow v_2) \vdash t_1 \not\simeq_s t_2 & \text{if } v_1 \text{ and } v_2 \text{ are different} \end{array}$$

+ reflexivity, symmetry, transitivity.

Example for LRA

LRA-public sorts: just Q.

Example for LRA

LRA-public sorts: just Q. LRA-values: $\mathbb{Q}$. LRA^{+}: trivial

Example for LRA

LRA-public sorts: just Q. LRA-values: $\mathbb{Q}$. LRA^{+}: trivial

(Some) LRA-inferences:

- Evaluations:

$$t_1 \leftarrow q_1, \ldots, t_n \leftarrow q_n \vdash_{\mathsf{LRA}} l \leftarrow b$$

  where l evaluates to b under the assignments

Example for LRA

LRA-public sorts: just Q. LRA-values: $\mathbb{Q}$. LRA$^+$: trivial
(Some) LRA-inferences:

- Evaluations:

$$t_1 \leftarrow q_1, \ldots, t_n \leftarrow q_n \vdash_{\mathsf{LRA}} l \leftarrow b$$

where l evaluates to b under the assignments

- FM-resolutions:

$$(e_1 \lessdot_1 x), (x \lessdot_2 e_2) \vdash_{\mathsf{LRA}} (e_1 \lessdot_3 e_2)$$

where $\lessdot$ is $<$ or $\leq \ldots$
(triggered only where e_1 and e_2 have been assigned values)

Example for LRA

LRA-public sorts: just Q. LRA-values: $\mathbb{Q}$. LRA$^+$: trivial
(Some) LRA-inferences:

- Evaluations:

  $$t_1 \leftarrow q_1, \ldots, t_n \leftarrow q_n \vdash_{\mathsf{LRA}} l \leftarrow b$$

  where l evaluates to b under the assignments

- FM-resolutions:

  $$(e_1 \lessdot_1 x), (x \lessdot_2 e_2) \vdash_{\mathsf{LRA}} (e_1 \lessdot_3 e_2)$$

  where $\lessdot$ is $<$ or $\leq \ldots$
  (triggered only where e_1 and e_2 have been assigned values)

- Treatment of disequality:

  $$(e_1 \leq x), (x \leq e_2), (e_1 \simeq e_0), (e_2 \simeq e_0), (\overline{x \simeq e_0}) \vdash_{\mathsf{LRA}} \bot$$

  (triggered only where e_0, e_1 and e_2 have been assigned values)

Design choices

Why make the notion of $\mathcal{T}$-inferences central?

- Rather minimalistic, with derived notions such as: Non-Boolean assignment $(t \leftarrow v)$ "immediately violates" J if there is an inference $J, (t \leftarrow v) \vdash_{\mathcal{T}} L$ with $\overline{L} \in J$

Design choices

Why make the notion of $\mathcal{T}$-inferences central?

- Rather minimalistic, with derived notions such as: Non-Boolean assignment $(t \leftarrow v)$ “immediately violates” J if there is an inference $J, (t \leftarrow v) \vdash_{\mathcal{T}} L$ with $\overline{L} \in J$
- $(t \leftarrow v)$ can be understood as a proxy for “all literals that immediately follow from this assignment” (compact representation for a group of simultaneously made Boolean decisions)

Design choices

Why make the notion of $\mathcal{T}$-inferences central?

- Rather minimalistic, with derived notions such as:
  Non-Boolean assignment $(t{\leftarrow}v)$ "immediately violates" J
  if there is an inference $J, (t{\leftarrow}v) \vdash_{\mathcal{T}} L$ with $\overline{L} \in J$
- $(t{\leftarrow}v)$ can be understood as a proxy for
  "all literals that immediately follow from this assignment"
  (compact representation for a group of simultaneously made Boolean decisions)
- Directional (as opposed to, say, a theory lemma):
  premisses of inferences have to be present in the problem,
  conclusion can introduce new material

Design choices

Why make the notion of $\mathcal{T}$-inferences central?

- Rather minimalistic, with derived notions such as:
  Non-Boolean assignment $(t \leftarrow v)$ "immediately violates" J
  if there is an inference $J, (t \leftarrow v) \vdash_{\mathcal{T}} L$ with $\overline{L} \in J$
- $(t \leftarrow v)$ can be understood as a proxy for
  "all literals that immediately follow from this assignment"
  (compact representation for a group of simultaneously made Boolean decisions)
- Directional (as opposed to, say, a theory lemma):
  premisses of inferences have to be present in the problem,
  conclusion can introduce new material
- Identifies the grains of theory-specific reasoning.
  An MC-Sat derivation of unsat almost explicitly constructs an aggregation of theory inferences
  that can be taken as a proof object (cf. example)

Generic calculus: Search rules

Parameterized by finite set of terms $\mathcal{B}$ called global basis
Let $\mathcal{T}$ be a theory with a specific $\mathcal{T}$-module.

If assignment $t \leftarrow v$ (in $\mathcal{T}$-public sort) does not immediately violate Γ
Decide

$$\Gamma \quad \longrightarrow \quad \Gamma, (t \leftarrow v)$$

If $J \vdash_{\mathcal{T}^=} L$,
with J already in Γ and L is for a formula in $\mathcal{B}$
Propagate

$$\Gamma \quad \longrightarrow \quad \Gamma, (J \vdash L) \quad \text{if } \overline{L} \text{ not in } \Gamma$$

Conflict

$$\Gamma \quad \longrightarrow \quad \Gamma' \qquad \begin{array}{l} \text{if } \overline{L} \text{ in } \Gamma, \\ \text{level}_\Gamma(J, \overline{L}) > 0 \\ \text{and analysing conflict } \langle \Gamma; J, \overline{L} \rangle \text{ gives } \Gamma' \end{array}$$

Fail

$$\Gamma \quad \longrightarrow \quad \text{unsat} \qquad \text{if } \overline{L} \text{ in } \Gamma \text{ and level}_\Gamma(J, \overline{L}) = 0$$

Generic calculus: Conflict analysis rules

Resolve

$\langle \Gamma; E, A \rangle \implies \langle \Gamma; E \cup J \rangle$ if $\mathrm{explain}_\Gamma(A) = J$
& greatest decision in J,
if any, is Boolean

UIPBackjump

$\langle \Gamma; E, L \rangle \implies \Gamma_{\leq \mathrm{level}_\Gamma(E)}, (E \vdash \overline{L})$ if $\mathrm{level}_\Gamma(E) < \mathrm{level}_\Gamma(L)$

SemSplit

$\langle \Gamma; E, L \rangle \implies \Gamma_{\leq \mathrm{level}_\Gamma(L)-1}, \overline{L}$ if $\mathrm{level}_\Gamma(L) = \mathrm{level}_\Gamma(E)$
& there is a decision in $\mathrm{explain}_\Gamma(L)$
& the greatest one is non-Boolean

Undo

$\langle \Gamma; E, A \rangle \implies \Gamma_{\leq \mathrm{level}_\Gamma(A)-1}$ if A is a non-Boolean decision
and $\mathrm{level}_\Gamma(E) < \mathrm{level}_\Gamma(A)$

3. Properties of the calculus

Termination and Soundness

Termination:

If for each theory module $\mathcal{T}$ involved,
there is a local basis $X \mapsto \mathsf{basis}_{\mathcal{T}}(X)$ satisfying some properties,

then it is possible to define a global finite basis for the combination of the theories

Termination and Soundness

Termination:

If for each theory module $\mathcal{T}$ involved,
there is a local basis $X \mapsto \text{basis}_{\mathcal{T}}(X)$ satisfying some properties,

then it is possible to define a global finite basis for the combination of the theories

. . . and termination of the calculus follows.

Termination and Soundness

Termination:

If for each theory module $\mathcal{T}$ involved,
there is a local basis $X \mapsto \mathsf{basis}_{\mathcal{T}}(X)$ satisfying some properties,

then it is possible to define a global finite basis for the combination of the theories

. . . and termination of the calculus follows.

(This relies on the fact that the theories are disjoint)

Termination and Soundness

Termination:

If for each theory module $\mathcal{T}$ involved,
there is a local basis $X \mapsto \mathsf{basis}_{\mathcal{T}}(X)$ satisfying some properties,

then it is possible to define a global finite basis for the combination of the theories

. . . and termination of the calculus follows.

(This relies on the fact that the theories are disjoint)

Soundness:

If for each theory module $\mathcal{T}$ involved the $\mathcal{T}$-inferences are sound (i.e. any model endorsing the premisses endorses the conclusion), then if the calculus ends with unsat, then the input was unsat

What happens if we never get unsat?

Do we have a model?

What happens if we never get unsat?

Do we have a model?

This relies on a completeness condition for theory modules:
For any Γ,

- Either any model of Γ in the equality theory (where each sort different from bool is interpreted as an infinite countable set) can be extended into a $\mathcal{T}^+$-model of Γ
- Or a $\mathcal{T}$-decision can be made (not immediately violating Γ)
- Or a $\mathcal{T}$-inference can infer a new assignment (for a term in the local basis)

What happens if we never get unsat?

Do we have a model?

This relies on a completeness condition for theory modules:
For any Γ,

- Either any model of Γ in the equality theory (where each sort different from bool is interpreted as an infinite countable set) can be extended into a $\mathcal{T}^+$-model of Γ
- Or a $\mathcal{T}$-decision can be made (not immediately violating Γ)
- Or a $\mathcal{T}$-inference can infer a new assignment (for a term in the local basis)

Theorem: If all theory modules satisfy the completeness condition, and if the calculus cannot make any further transitions, then the state describes a model.

What happens if we never get unsat?

Do we have a model?

This relies on a completeness condition for theory modules:
For any Γ,

- Either any model of Γ in the equality theory (where each sort different from bool is interpreted as an infinite countable set) can be extended into a $\mathcal{T}^+$-model of Γ
- Or a $\mathcal{T}$-decision can be made (not immediately violating Γ)
- Or a $\mathcal{T}$-inference can infer a new assignment (for a term in the local basis)

Theorem: If all theory modules satisfy the completeness condition, and if the calculus cannot make any further transitions, then the state describes a model.

Proof adapts Nelson-Oppen

Theories for which we provided such theory modules

- LRA(careful with the local basis)

Theories for which we provided such theory modules

- LRA(careful with the local basis)
- EUF

$$\begin{aligned}
(t_i \simeq u_i)_{i=1\dots n}, (f(t_1,\dots,t_n) \not\simeq f(u_1,\dots,u_n)) &\vdash_{\mathsf{EUF}} \quad \bot \\
(t_i \simeq u_i)_{i=1\dots n} &\vdash_{\mathsf{EUF}} \quad (f(t_1,\dots,t_n) \simeq f(u_1,\dots,u_n)) \\
(t_i \simeq u_i)_{i=1\dots n, i\neq i_0}, f(t_1,\dots,t_n) \not\simeq f(u_1,\dots,u_n) &\vdash_{\mathsf{EUF}} \quad t_{i_0} \not\simeq u_{i_0}
\end{aligned}$$

Theories for which we provided such theory modules

- LRA(careful with the local basis)
- EUF

$$\begin{aligned}
(t_i \simeq u_i)_{i=1\dots n}, (f(t_1,\dots,t_n) \not\simeq f(u_1,\dots,u_n)) &\vdash_{\mathsf{EUF}} \quad \bot \\
(t_i \simeq u_i)_{i=1\dots n} &\vdash_{\mathsf{EUF}} \quad (f(t_1,\dots,t_n) \simeq f(u_1,\dots,u_n)) \\
(t_i \simeq u_i)_{i=1\dots n, i\neq i_0}, f(t_1,\dots,t_n) \not\simeq f(u_1,\dots,u_n) &\vdash_{\mathsf{EUF}} \quad t_{i_0} \not\simeq u_{i_0}
\end{aligned}$$

- Arrays

$$\begin{aligned}
(t \simeq t'), (i \simeq i'), (t[i] \not\simeq t'[i']) &\vdash_{\mathsf{Arr}} \quad \bot \\
(t \simeq t'), (i \simeq i'), (u \simeq u'), (t[i{:=}\, u] \not\simeq t'[i'{:=}\, u']) &\vdash_{\mathsf{Arr}} \quad \bot \\
(t \simeq t'), (u \simeq u'), (\mathsf{diff}(t,u) \not\simeq \mathsf{diff}(t',u')) &\vdash_{\mathsf{Arr}} \quad \bot \\
(t' \simeq t[i{:=}\, u]), (i \simeq j), (u \not\simeq t'[j]) &\vdash_{\mathsf{Arr}} \quad \bot \\
(t' \simeq t[i{:=}\, u]), (i \not\simeq j), (j \simeq j'), (t[j] \not\simeq t'[j']) &\vdash_{\mathsf{Arr}} \quad \bot \\
(t \not\simeq u) &\vdash_{\mathsf{Arr}} \quad (t[\mathsf{diff}(t,u)] \not\simeq u[\mathsf{diff}(t,u)])
\end{aligned}$$

Theories for which we provided such theory modules

- LRA(careful with the local basis)
- EUF

$$\begin{array}{rl} (t_i \simeq u_i)_{i=1\dots n}, (f(t_1, \dots, t_n) \not\simeq f(u_1, \dots, u_n)) \vdash_{\mathsf{EUF}} & \bot \\ (t_i \simeq u_i)_{i=1\dots n} \vdash_{\mathsf{EUF}} & (f(t_1, \dots, t_n) \simeq f(u_1, \dots, u_n)) \\ (t_i \simeq u_i)_{i=1\dots n, i\neq i_0}, f(t_1, \dots, t_n) \not\simeq f(u_1, \dots, u_n) \vdash_{\mathsf{EUF}} & t_{i_0} \not\simeq u_{i_0} \end{array}$$

- Arrays

$$\begin{array}{rl} (t \simeq t'), (i \simeq i'), (t[i] \not\simeq t'[i']) \vdash_{\mathsf{Arr}} & \bot \\ (t \simeq t'), (i \simeq i'), (u \simeq u'), (t[i{:=}\, u] \not\simeq t'[i'{:=}\, u']) \vdash_{\mathsf{Arr}} & \bot \\ (t \simeq t'), (u \simeq u'), (\mathsf{diff}(t, u) \not\simeq \mathsf{diff}(t', u')) \vdash_{\mathsf{Arr}} & \bot \\ (t' \simeq t[i{:=}\, u]), (i \simeq j), (u \not\simeq t'[j]) \vdash_{\mathsf{Arr}} & \bot \\ (t' \simeq t[i{:=}\, u]), (i \not\simeq j), (j \simeq j'), (t[j] \not\simeq t'[j']) \vdash_{\mathsf{Arr}} & \bot \\ (t \not\simeq u) \vdash_{\mathsf{Arr}} & (t[\mathsf{diff}(t, u)] \not\simeq u[\mathsf{diff}(t, u)]) \end{array}$$

- Black box procedure (coarse-grain inferences)

$$l_1 \leftarrow b_1, \dots, l_n \leftarrow b_n \vdash_{\mathcal{T}} \bot$$

where $l_1, \dots, l_n$ are formulæ, and the conjunction of the literals corresponding to the Boolean assignments $l_1 \leftarrow b_1, \dots, l_n \leftarrow b_n$ is $\mathcal{T}$-unsatisfiable

(as detected by e.g. the decision procedure)

Conclusion

In [BGLS16],
We do not assume purification
& let every theory module see every other theory term assignment.

Consequence: We can remove the stably infinite condition

Conclusion

In [BGLS16],

We do not assume purification

& let every theory module see every other theory term assignment.

Consequence: We can remove the stably infinite condition

But we still need to use a reference theory $\mathcal{T}_0$ to make theories agree of sorts cardinalities

Completeness condition for theory modules is now dependent on $\mathcal{T}_0$ ($\mathcal{T}_0$-completeness)

Conclusion

In [BGLS16],

We do not assume purification

& let every theory module see every other theory term assignment.

Consequence: We can remove the stably infinite condition

But we still need to use a reference theory $\mathcal{T}_0$ to make theories agree of sorts cardinalities

Completeness condition for theory modules is now dependent on $\mathcal{T}_0$ ($\mathcal{T}_0$-completeness)

Further work:

- non-disjoint theories?

Conclusion

In [BGLS16],
We do not assume purification
& let every theory module see every other theory term assignment.

Consequence: We can remove the stably infinite condition
But we still need to use a reference theory $\mathcal{T}_0$ to make theories agree of sorts cardinalities
Completeness condition for theory modules is now dependent on $\mathcal{T}_0$ ($\mathcal{T}_0$-completeness)

Further work:

- non-disjoint theories?
- how to handle quantifiers?

Conclusion

In [BGLS16],
We do not assume purification
& let every theory module see every other theory term assignment.

Consequence: We can remove the stably infinite condition
But we still need to use a reference theory $\mathcal{T}_0$ to make theories agree of sorts cardinalities
Completeness condition for theory modules is now dependent on $\mathcal{T}_0$ ($\mathcal{T}_0$-completeness)

Further work:

- non-disjoint theories?
- how to handle quantifiers?
- From proof production to “proved correct” implementation:
  If implementation of each inference is correct and state transitions are correct, then answer is correct
  Separates a kernel that is critical for correctness
  from strategies that is critical for efficiency

M. P. Bonacina, S. Graham-Lengrand, and N. Shankar.
A model-constructing framework for theory combination.
Technical Report RR-99/2016, Università degli Studi di Verona - SRI International - CNRS, 2016.
Available at
http://hal.archives-ouvertes.fr/hal-01425305

N. Bjorner and M. Janota.
Playing with quantified satisfaction.
In M. Davis, A. Fehnker, A. McIver, and A. Voronkov, editors, *Proc. of the the 20th Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'15)*, volume 9450 of *LNCS*. Springer-Verlag, 2015.

L. M. de Moura and D. Jovanovic.
A model-constructing satisfiability calculus.
In R. Giacobazzi, J. Berdine, and I. Mastroeni, editors, *Proc. of the 14th Int. Conf. on Verification, Model Checking, and*

Abstract Interpretation (VMCAI'13), volume 7737 of *LNCS*, pages 1–12. Springer-Verlag, 2013.

D. Jovanović, C. Barrett, and L. de Moura.
The design and implementation of the model constructing satisfiability calculus.
In *Proc. of the 13th Int. Conf. on Formal Methods In Computer-Aided Design (FMCAD '13)*. FMCAD Inc., 2013. Portland, Oregon

D. Jovanović and L. de Moura.
Solving non-linear arithmetic.
In B. Gramlich, D. Miller, and U. Sattler, editors, *Proc. of the 6th Int. Joint Conf. on Automated Reasoning (IJCAR)*, volume 7364 of *LNAI*, pages 339–354. Springer, 2012.

D. Jovanovic.
Solving nonlinear integer arithmetic with MCSAT.

In A. Bouajjani and D. Monniaux, editors, *Verification, Model Checking, and Abstract Interpretation - 18th International Conference, VMCAI 2017, Paris, France, January 15-17, 2017, Proceedings*, volume 10145 of *LNCS*, pages 330–346. Springer-Verlag, 2017.

A. Zeljic, C. M. Wintersteiger, and P. Rümmer.
Deciding bit-vector formulas with mcsat.
In N. Creignou and D. L. Berre, editors, *Theory and Applications of Satisfiability Testing - SAT 2016 - 19th International Conference, Bordeaux, France, July 5-8, 2016, Proceedings*, volume 9710 of *Lecture Notes in Computer Science*, pages 249–266. Springer, 2016.

Messages and provability primitives

Implementing this in PSYCHE; each theory $\mathcal{T}$ can emit messages:

`type _ message =`	
`| Unsat : set -> unsat message`	$\Gamma \vdash_{\mathcal{T}} \bot$
`| Infer : set -> form -> infer message`	$\Gamma \vdash_{\mathcal{T}} A$
`| Sat   : set -> sat message`	"$\mathcal{T}$ checks Γ"

Messages and provability primitives

Implementing this in PSYCHE; each theory $\mathcal{T}$ can emit messages:

`type _ message =`	
`| Unsat : set -> unsat message`	$\Gamma \vdash_{\mathcal{T}} \bot$
`| Infer : set -> form -> infer message`	$\Gamma \vdash_{\mathcal{T}} A$
`| Sat   : set -> sat message`	"$\mathcal{T}$ checks Γ"

```
module type Combo = sig
type 'b ans = [...]
val oracle  : 'b message -> 'b ans
val resolve : infer ans -> unsat ans -> unsat ans
val curryfy : set -> unsat ans -> infer ans
[...]
end
```

Messages and provability primitives

Implementing this in PSYCHE; each theory $\mathcal{T}$ can emit messages:

```
type _ message =
| Unsat : set -> unsat message
| Infer : set -> form -> infer message
| Sat   : set -> sat message
```

$\Gamma \vdash_{\mathcal{T}} \bot$

$\Gamma \vdash_{\mathcal{T}} A$

"$\mathcal{T}$ checks Γ"

```
module type Combo = sig
type 'b ans = [...]
val oracle  : 'b message -> 'b ans
val resolve : infer ans -> unsat ans -> unsat ans
val curryfy : set -> unsat ans -> infer ans
[...]
end
```

oracle

$$\Gamma \vdash_{\mathcal{T}} \bot \qquad \rightsquigarrow \qquad \frac{}{\Gamma \vdash \bot}$$

Messages and provability primitives

Implementing this in PSYCHE; each theory $\mathcal{T}$ can emit messages:

```
type _ message =
```

`| Unsat : set -> unsat message`	$\Gamma \vdash_{\mathcal{T}} \bot$
`| Infer : set -> form -> infer message`	$\Gamma \vdash_{\mathcal{T}} A$
`| Sat   : set -> sat message`	“$\mathcal{T}$ checks Γ”

```
module type Combo = sig
type 'b ans = [...]
val oracle  : 'b message -> 'b ans
val resolve : infer ans -> unsat ans -> unsat ans
val curryfy : set -> unsat ans -> infer ans
[...]
end
```

oracle

$$\Gamma \vdash_{\mathcal{T}} A \qquad \leadsto \qquad \Gamma \vdash A$$

Messages and provability primitives

Implementing this in Psyche; each theory $\mathcal{T}$ can emit messages:

```
type _ message =
| Unsat : set -> unsat message
| Infer : set -> form -> infer message
| Sat   : set -> sat message
```

$\Gamma \vdash_{\mathcal{T}} \bot$

$\Gamma \vdash_{\mathcal{T}} A$

"$\mathcal{T}$ checks Γ"

```
module type Combo = sig
type 'b ans = [...]
val oracle  : 'b message -> 'b ans
val resolve : infer ans -> unsat ans -> unsat ans
val curryfy : set -> unsat ans -> infer ans
[...]
end
```

resolve

$$\begin{array}{c}\Delta \subseteq \Gamma \\ \Gamma' \subseteq \Gamma\end{array} \qquad \frac{\Delta, l \vdash \bot}{\Gamma \vdash^{?} \bot}\ \Gamma' \vdash l \qquad \rightsquigarrow \qquad \frac{\Delta, l \vdash \bot}{\Delta \cup \Gamma' \vdash \bot}$$

Messages and provability primitives

Implementing this in PSYCHE; each theory $\mathcal{T}$ can emit messages:

```
type _ message =
```

`| Unsat : set -> unsat message`	$\Gamma \vdash_{\mathcal{T}} \bot$
`| Infer : set -> form -> infer message`	$\Gamma \vdash_{\mathcal{T}} A$
`| Sat   : set -> sat message`	“$\mathcal{T}$ checks Γ”

```
module type Combo = sig
type 'b ans = [...]
val oracle  : 'b message -> 'b ans
val resolve : infer ans -> unsat ans -> unsat ans
val curryfy : set -> unsat ans -> infer ans
[...]
end
```

curryfy

$$\Gamma, A \vdash \bot \quad \rightsquigarrow \quad \Gamma \vdash \neg A$$

Satisfiability primitives

```
 [...]
val sat_init : set -> sat ans
val sat_combo: sat ans -> sat ans -> sat ans
```

Satisfiability primitives

```
[...]
val sat_init : set -> sat ans
val sat_combo: sat ans -> sat ans -> sat ans
```

`sat_init` Γ

records that satisfiability of Γ needs to be checked by all theories

Satisfiability primitives

```
[...]
val sat_init : set -> sat ans
val sat_combo: sat ans -> sat ans -> sat ans
```

`sat_init Γ`
records that satisfiability of Γ needs to be checked by all theories

`sat_combo t1 t2`
checks that the Γ in `t1` and `t2` match, then
theories that still need to check it
= intersection of those in `t1` and `t2`

Satisfiability primitives

```
[...]
val sat_init : set -> sat ans
val sat_combo: sat ans -> sat ans -> sat ans
```

`sat_init` Γ
records that satisfiability of Γ needs to be checked by all theories

`sat_combo t1 t2`
checks that the Γ in `t1` and `t2` match, then
theories that still need to check it
= intersection of those in `t1` and `t2`

Here, "$\mathcal{T}$ checks Γ" means **more** than "Γ is $\mathcal{T}$-satisfiable".
It means "Γ entirely describes the $\mathcal{T}$-model".

Satisfiability primitives

```
[...]
val sat_init : set -> sat ans
val sat_combo: sat ans -> sat ans -> sat ans
```

`sat_init` Γ
records that satisfiability of Γ needs to be checked by all theories

`sat_combo t1 t2`
checks that the Γ in `t1` and `t2` match, then
theories that still need to check it
= intersection of those in `t1` and `t2`

Here, "$\mathcal{T}$ checks Γ" means **more** than "Γ is $\mathcal{T}$-satisfiable".
It means "Γ entirely describes the $\mathcal{T}$-model".

When no more theories have to check satisfiability of Γ, we stop:
all theories have agreed on model

Trust

```
module type Combo = sig
type 'b ans = [...]
[...]
end
```

Type `'b ans` is private to module `Combo`…
…like type `theorem` of the LCF architecture for theorem proving.

Trust

```
module type Combo = sig
type 'b ans = [...]
[...]
end
```

Type `'b ans` is private to module `Combo`...

...like type `theorem` of the LCF architecture for theorem proving.

This guarantees correctness of answers...

1. ...if module `Combo` is trusted
2. ...if messages from the theories are trusted,

and regardless of the strategies used to drive the search.

Trust

```
module type Combo = sig
type 'b ans = [...]
[...]
end
```

Type `'b ans` is private to module `Combo`…
…like type `theorem` of the LCF architecture for theorem proving.

This guarantees correctness of answers…

1. …if module `Combo` is trusted
2. …if messages from the theories are trusted,

and regardless of the strategies used to drive the search.

Here, small steps for theory messages are highly desirable for (2.):
Easier to trust (or prove correct)
the code producing message $(e < x), (x < e') \vdash_{\mathsf{LRA}} (e < e')$
than a full simplex code.