Linear Integer Arithmetic Revisited

Martin Bromberger, Thomas Sturm, Christoph Weidenbach

Max-Planck-Institute for Informatics

July 17, 2015

Linear Integer Arithmetic

Input:

$$C = \bigwedge_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \leq b_i \right), \text{ where } a_{ij}, b_i \in \mathbb{Z}$$

Linear Integer Arithmetic

Input:

$$C = \bigwedge_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \leq b_i \right), \text{ where } a_{ij}, b_i \in \mathbb{Z}$$

Output: A solution $x^* \in \mathbb{Z}^n$ or proof of unsatisfiability

Linear Integer Arithmetic

Input:

$$C = \bigwedge_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \leq b_i \right), \text{ where } a_{ij}, b_i \in \mathbb{Z}$$

Output: A solution $x^* \in \mathbb{Z}^n$ or proof of unsatisfiability

Example: $3x - 2y \leq 1 \quad \wedge \quad z \geq 0 \quad \wedge \quad x + 2y - 7z \geq 2$

Linear Integer Arithmetic

Input:

$$C = \bigwedge_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \leq b_i \right), \text{ where } a_{ij}, b_i \in \mathbb{Z}$$

Output: A solution $x^* \in \mathbb{Z}^n$ or proof of unsatisfiability

Example: $3x - 2y \leq 1 \quad \wedge \quad z \geq 0 \quad \wedge \quad x + 2y - 7z \geq 2$

Solution: $x = 0,\ y = 1,\ z = 0$

Linear Integer Arithmetic

Input:

$$C = \bigwedge_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \leq b_i \right), \text{ where } a_{ij}, b_i \in \mathbb{Z}$$

Output: A solution $x^* \in \mathbb{Z}^n$ or proof of unsatisfiability

Example: $3 \cdot 0 - 2 \cdot 1 \leq 1 \ \wedge \quad 0 \geq 0 \quad \wedge \ 0 + 2 \cdot 1 - 7 \cdot 0 \geq 2$

Solution: $x = 0,\ y = 1,\ z = 0$

Linear Integer Arithmetic

Input:

$$C = \bigwedge_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \leq b_i \right), \text{ where } a_{ij}, b_i \in \mathbb{Z}$$

Output: A solution $x^* \in \mathbb{Z}^n$ or proof of unsatisfiability

Example: $-2 \leq 1 \quad \wedge \quad 0 \geq 0 \quad \wedge \quad 2 \geq 2$

Solution: $x = 0, y = 1, z = 0$

Difficulty

Restricting solutions to Integers vs Reals:

Difficulty

Restricting solutions to Integers vs Reals:

$$y \geq 4x \quad \wedge \quad 3x \geq z$$

Difficulty

Restricting solutions to Integers vs Reals:

$$y \geq 4x \quad \wedge \quad 3x \geq z \qquad \equiv \qquad \frac{y}{4} \geq x \geq \frac{z}{3}$$

Difficulty

Restricting solutions to Integers vs Reals:

$$y \geq 4x \quad \wedge \quad 3x \geq z \qquad \equiv \qquad \frac{y}{4} \geq \; x \; \geq \; \frac{z}{3}$$

Real lower bound: $x := \frac{z}{3}$

Difficulty

Restricting solutions to Integers vs Reals:

$$y \geq 4x \quad \wedge \quad 3x \geq z \qquad \equiv \qquad \frac{y}{4} \geq x \geq \frac{z}{3}$$

Real lower bound: $x := \frac{z}{3} \qquad \Rightarrow \qquad 3y \geq 4z \quad \wedge \quad z \geq z$

Difficulty

Restricting solutions to Integers vs Reals:

$$y \geq 4x \quad \wedge \quad 3x \geq z \qquad \equiv \qquad \frac{y}{4} \geq x \geq \frac{z}{3}$$

Real lower bound: $x := \frac{z}{3} \qquad \Rightarrow \qquad 3y \geq 4z \quad \wedge \quad z \geq z$

Not a solution for Integers! E.g., set $y := 3$, $z := 1$, then

$$3 \cdot 3 \geq 4 \cdot 1 \quad \wedge \quad 1 \geq 1 \quad \text{but} \quad \frac{3}{4} \geq x \geq \frac{1}{3}$$

Difficulty

Restricting solutions to Integers vs Reals:

$$y \geq 4x \quad \wedge \quad 3x \geq z \qquad \equiv \qquad \frac{y}{4} \geq x \geq \frac{z}{3}$$

Real lower bound: $x := \frac{z}{3} \qquad \Rightarrow \qquad 3y \geq 4z \quad \wedge \quad z \geq z$

Not a solution for Integers! E.g., set $y := 3$, $z := 1$, then

$$3 \cdot 3 \geq 4 \cdot 1 \quad \wedge \quad 1 \geq 1 \quad \text{but} \quad \frac{3}{4} \geq x \geq \frac{1}{3}$$

Integer lower bound: $x := \left\lceil \frac{z}{3} \right\rceil$

Difficulty

Restricting solutions to Integers vs Reals:

$$y \geq 4x \quad \wedge \quad 3x \geq z \qquad \equiv \qquad \frac{y}{4} \geq x \geq \frac{z}{3}$$

Real lower bound: $x := \frac{z}{3} \qquad \Rightarrow \qquad 3y \geq 4z \quad \wedge \quad z \geq z$

Not a solution for Integers! E.g., set $y := 3$, $z := 1$, then

$$3 \cdot 3 \geq 4 \cdot 1 \quad \wedge \quad 1 \geq 1 \quad \text{but} \quad \frac{3}{4} \geq x \geq \frac{1}{3}$$

Integer lower bound: $x := \left\lceil \frac{z}{3} \right\rceil \in \left\{ \frac{z}{3}, \frac{z+1}{3}, \frac{z+2}{3} \right\}$

Difficulty

Restricting solutions to Integers vs Reals:

$$y \geq 4x \quad \wedge \quad 3x \geq z \qquad \equiv \qquad \frac{y}{4} \geq x \geq \frac{z}{3}$$

Real lower bound: $x := \frac{z}{3} \qquad \Rightarrow \qquad 3y \geq 4z \quad \wedge \quad z \geq z$

Not a solution for Integers! E.g., set $y := 3$, $z := 1$, then

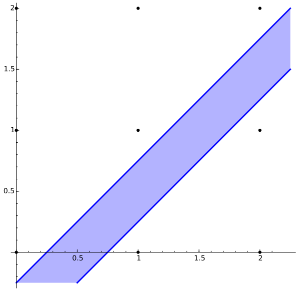
$$3 \cdot 3 \geq 4 \cdot 1 \quad \wedge \quad 1 \geq 1 \quad \text{but} \quad \frac{3}{4} \geq x \geq \frac{1}{3}$$

Integer lower bound: $x := \left\lceil \frac{z}{3} \right\rceil \in \left\{ \frac{z}{3}, \frac{z+1}{3}, \frac{z+2}{3} \right\}$

$$\Rightarrow \bigvee_{k=0}^{2} 3 \mid z + k \quad \wedge \quad 3y \geq 4z + 4k \quad \wedge \quad z + k \geq z$$

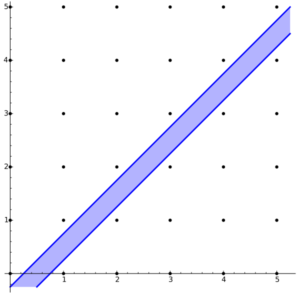
Unbounded Problems

$$4x - 4y \geq 1$$
$$\wedge$$
$$4x - 4y \leq 3$$



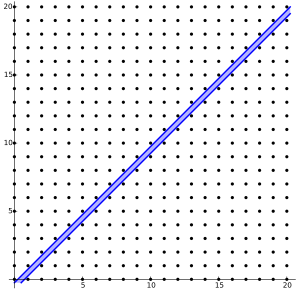
Unbounded Problems

$$4x - 4y \geq 1$$
$$\wedge$$
$$4x - 4y \leq 3$$



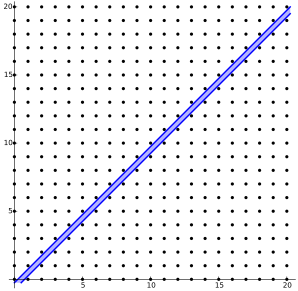
Unbounded Problems

$$4x - 4y \geq 1$$
$$\wedge$$
$$4x - 4y \leq 3$$



Unbounded Problems

$$4x - 4y \geq 1$$
$$\wedge$$
$$4x - 4y \leq 3$$

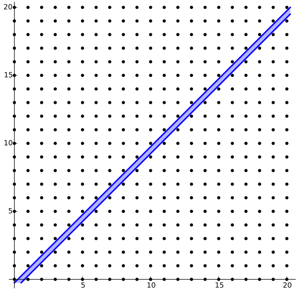


A Priori Bounds:

- $\forall i. \; -n(ma)^{2m+1} \leq x_i \leq n(ma)^{2m+1}$
- finite search space: solve via enumeration
- for the example: $-65536 \leq x, y \leq 65536$
- in practice: too large!
- but: state-of-the-art!

Unbounded Problems

$$4x - 4y \geq 1$$
$$\wedge$$
$$4x - 4y \leq 3$$

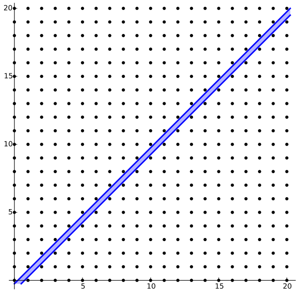


A Priori Bounds:

- $\forall i. \; -n(ma)^{2m+1} \leq x_i \leq n(ma)^{2m+1}$
- finite search space: solve via enumeration
- for the example: $-65536 \leq x, y \leq 65536$
- in practice: too large!
- but: state-of-the-art!

Unbounded Problems

$$4x - 4y \geq 1$$
$$\wedge$$
$$4x - 4y \leq 3$$

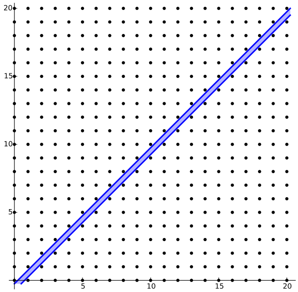


A Priori Bounds:

- $\forall i. \; -n(ma)^{2m+1} \leq x_i \leq n(ma)^{2m+1}$
- finite search space: solve via enumeration
- for the example: $-65536 \leq x, y \leq 65536$
- in practice: too large!
- but: state-of-the-art!

Unbounded Problems

$$4x - 4y \geq 1$$
$$\wedge$$
$$4x - 4y \leq 3$$

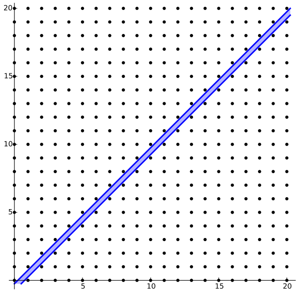


A Priori Bounds:

- $\forall i. \; -n(ma)^{2m+1} \leq x_i \leq n(ma)^{2m+1}$
- finite search space: solve via enumeration
- for the example: $-65536 \leq x, y \leq 65536$
- in practice: too large!
- but: state-of-the-art!

Unbounded Problems

$$4x - 4y \geq 1$$
$$\wedge$$
$$4x - 4y \leq 3$$



A Priori Bounds:

- $\forall i. \; -n(ma)^{2m+1} \leq x_i \leq n(ma)^{2m+1}$
- finite search space: solve via enumeration
- for the example: $-65536 \leq x, y \leq 65536$
- in practice: too large!
- but: state-of-the-art!

Cutting to the Chase

by Dejan Jovanovic and Leonardo M. de Moura (2011, 2013)

A CDCL based LIA solver (CUTSAT)

- termination oriented on problem structure
- termination is dynamic; not by a priori bounds
- termination based on quantifier elimination
  - reducing unbounded variables to bounded variables
  - requires divisibility constraints, e.g., $6 \mid 3x + 2y$

Cutting to the Chase

by Dejan Jovanovic and Leonardo M. de Moura (2011, 2013)

A CDCL based LIA solver (CUTSAT)

Contributions:

- termination oriented on problem structure
- termination is dynamic; not by a priori bounds
- termination based on quantifier elimination
  - reducing unbounded variables to bounded variables
  - requires divisibility constraints, e.g., $6 \mid 3x + 2y$

Cutting to the Chase

by Dejan Jovanovic and Leonardo M. de Moura (2011, 2013)

A CDCL based LIA solver (CUTSAT)

Contributions:

- termination oriented on problem structure
- termination is dynamic; not by a priori bounds
- termination based on quantifier elimination
  - reducing unbounded variables to bounded variables
  - requires divisibility constraints, e.g., $6 \mid 3x + 2y$

Cutting to the Chase

by Dejan Jovanovic and Leonardo M. de Moura (2011, 2013)

A CDCL based LIA solver (CUTSAT)

Contributions:

- termination oriented on problem structure
- termination is dynamic; not by a priori bounds
- termination based on quantifier elimination
  - reducing unbounded variables to bounded variables
  - requires divisibility constraints, e.g., $6 \mid 3x + 2y$

Cutting to the Chase

by Dejan Jovanovic and Leonardo M. de Moura (2011, 2013)

A CDCL based LIA solver (CUTSAT)

Contributions:

- termination oriented on problem structure
- termination is dynamic; not by a priori bounds
- termination based on quantifier elimination
  - reducing unbounded variables to bounded variables
  - requires divisibility constraints, e.g., $6 \mid 3x + 2y$

Cutting to the Chase

by Dejan Jovanovic and Leonardo M. de Moura (2011, 2013)

A CDCL based LIA solver (CUTSAT)

Contributions:

- termination oriented on problem structure
- termination is dynamic; not by a priori bounds
- termination based on quantifier elimination
  - reducing unbounded variables to bounded variables
  - requires divisibility constraints, e.g., $6 \mid 3x + 2y$

Cutting to the Chase

by Dejan Jovanovic and Leonardo M. de Moura (2011, 2013)

A CDCL based LIA solver (CUTSAT)

Contributions:

- termination oriented on problem structure
- termination is dynamic; not by a priori bounds
- termination based on quantifier elimination
  - reducing unbounded variables to bounded variables
  - requires divisibility constraints, e.g., $6 \mid 3x + 2y$

Example

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ z \geq 0 \wedge \quad z \geq y + 1 \quad \wedge \quad z \leq x + y$$

$$M := [\![]\!]$$

Propagate

Propagate inequalities with one variable!

Example

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ z \geq 0 \wedge \quad z \geq y + 1 \quad \wedge \quad z \leq x + y$$

$$M := [\![]\!]$$

Propagate

Propagate inequalities with one variable!

Example

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ z \geq 0 \ \wedge \quad z \geq y + 1 \quad \wedge \quad z \leq x + y$$

$$M := [\![x \geq 0, y \geq 0, z \geq 0]\!]$$

Propagate

Propagate inequalities with one variable!

Example

$$C := x \geq 0 \ \land \ y \geq 0 \ \land \ z \geq 0 \land \quad z \geq y + 1 \quad \land \quad z \leq x + y$$

$$M := [\![x \geq 0, y \geq 0, z \geq 0]\!]$$

Propagate

Use model to propagate more bounds!

Example

$$z \geq 0 + 1$$

$$C := x \geq 0 \;\land\; y \geq 0 \;\land\; z \geq 0 \land \quad z \geq y + 1 \quad \land \quad z \leq x + y$$

$$M := [\![x \geq 0, y \geq 0, z \geq 0]\!]$$

Propagate

Use model to propagate more bounds!

Example

$$C := x \geq 0 \;\wedge\; y \geq 0 \;\wedge\; z \geq 0 \;\wedge\; \frac{z \geq 0 + 1}{z \geq y + 1} \;\wedge\; z \leq x + y$$

$$M := [\![x \geq 0, y \geq 0, z \geq 0, z \geq 1]\!]$$

Propagate

Use model to propagate more bounds!

Example

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ z \geq 0 \wedge \quad z \geq y + 1 \quad \wedge \quad z \leq x + y$$

$$M := [\![x \geq 0, y \geq 0, z \geq 0, z \geq 1]\!]$$

Decision

Assign variables with a decision!

Example

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ z \geq 0 \wedge \quad z \geq y + 1 \quad \wedge \quad z \leq x + y$$

$$M := [\![x \geq 0, y \geq 0, z \geq 0, z \geq 1, x \leq_{\mathbf{d}} 0]\!]$$

Decision

Assign variables with a decision!

Example

$$C := x \geq 0 \;\wedge\; y \geq 0 \;\wedge\; z \geq 0 \;\wedge\; z \geq y + 1 \;\wedge\; z \leq x + y$$

$$M := [\![x \geq 0, y \geq 0, z \geq 0, z \geq 1, x \leq_{\mathbf{d}} 0, y \leq_{\mathbf{d}} 0]\!]$$

Decision

Assign variables with a decision!

Example

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ z \geq 0 \wedge \quad z \geq y + 1 \quad \wedge \quad z \leq x + y$$

$$M := [\![x \geq 0, y \geq 0, z \geq 0, z \geq 1, x \leq_{\mathbf{d}} 0, y \leq_{\mathbf{d}} 0]\!]$$

Conflict

Find responsible constraints

Example

$$C := x \geq 0 \;\wedge\; y \geq 0 \;\wedge\; z \geq 0 \;\wedge\; \underset{z \geq y + 1}{z \geq 0 + 1} \;\wedge\; \underset{z \leq x + y}{z \leq 0 + 0}$$

$$M := [\![x \geq 0, y \geq 0, z \geq 0, z \geq 1, x \leq_{\mathbf{d}} 0, y \leq_{\mathbf{d}} 0]\!]$$

Conflict

Find responsible constraints

Example

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ z \geq 0 \wedge \quad \begin{matrix} z \geq 0 + 1 \\ z \geq y + 1 \end{matrix} \quad \wedge \quad \begin{matrix} z \leq 0 + 0 \\ z \leq x + y \end{matrix}$$

$$M := [\![x \geq 0, y \geq 0, z \geq 0, z \geq 1, x \leq_{\mathbf{d}} 0, y \leq_{\mathbf{d}} 0]\!]$$

Conflict

Backtrack and block conflict with resolvent!

Example

$$C := x \geq 0 \;\wedge\; y \geq 0 \;\wedge\; z \geq 0 \;\wedge\; \begin{matrix} z \geq 0 + 1 \\ z \geq y + 1 \end{matrix} \;\wedge\; \begin{matrix} z \leq 0 + 0 \\ z \leq x + y \end{matrix}$$

$$M := [\![x \geq 0, y \geq 0, z \geq 0, z \geq 1]\!]$$

Conflict

Backtrack and block conflict with resolvent!

Example

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ z \geq 0 \ \wedge \quad \overset{z \geq 0 + 1}{z \geq y + 1} \quad \wedge \quad \overset{z \leq 0 + 0}{z \leq x + y}$$

$$\wedge \ x \geq 1$$

$$M := [\![x \geq 0, y \geq 0, z \geq 0, z \geq 1]\!]$$

Conflict

Backtrack and block conflict with resolvent!

Example

$$C := x \geq 0 \;\wedge\; y \geq 0 \;\wedge\; z \geq 0 \wedge \quad z \geq y + 1 \quad \wedge \quad z \leq x + y \;\wedge\; x \geq 1$$

$$M := [\![x \geq 0, y \geq 0, z \geq 0, z \geq 1]\!]$$

Conflict

Backtrack and block conflict with resolvent!

Example

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ z \geq 0 \wedge \quad z \geq y + 1 \quad \wedge \quad z \leq x + y \ \wedge \ x \geq 1$$

$$M := [\![x \geq 0, y \geq 0, z \geq 0, z \geq 1]\!]$$

Propagate

Use model to propagate more bounds!

Example

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ z \geq 0 \wedge \quad z \geq y + 1 \quad \wedge \quad z \leq x + y \ \wedge \ x \geq 1$$

$$M := [\![x \geq 0, y \geq 0, z \geq 0, z \geq 1, x \geq 1]\!]$$

Propagate

Use model to propagate more bounds!

Example

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ z \geq 0 \wedge \quad z \geq y + 1 \quad \wedge \quad z \leq x + y \quad \wedge \ x \geq 1$$

$$M := [\![x \geq 0, y \geq 0, z \geq 0, z \geq 1, x \geq 1]\!]$$

Decision

Assign variables with a decision!

Example

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ z \geq 0 \wedge \quad z \geq y + 1 \quad \wedge \quad z \leq x + y \ \wedge x \geq 1$$

$$M := [\![x \geq 0, y \geq 0, z \geq 0, z \geq 1, x \geq 1, x \leq_{\mathbf{d}} 1, y \leq_{\mathbf{d}} 0, z \leq_{\mathbf{d}} 1]\!]$$

Decision

Assign variables with a decision!

Example

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ z \geq 0 \ \wedge \quad \begin{matrix} z \geq 0 + 1 \\ z \geq y + 1 \end{matrix} \quad \wedge \quad \begin{matrix} z \leq 1 + 0 \\ z \leq x + y \end{matrix}$$
$$\wedge \ x \geq 1$$

$$M := [\![x \geq 0, y \geq 0, z \geq 0, z \geq 1, x \geq 1, x \leq_{\mathbf{d}} 1, y \leq_{\mathbf{d}} 0, z \leq_{\mathbf{d}} 1]\!]$$

Done

Complete model: $x := 1, y := 0, z := 1$

Cutting to the Chase

by Dejan Jovanovic and Leonardo M. de Moura (2011, 2013)

Claims:

- terminates with bounds
- terminates without bounds;

Cutting to the Chase

by Dejan Jovanovic and Leonardo M. de Moura (2011, 2013)

Claims:

- terminates with bounds ✓
- terminates without bounds;

Cutting to the Chase

by Dejan Jovanovic and Leonardo M. de Moura (2011, 2013)

Claims:

- terminates with bounds ✓
- terminates without bounds;

Cutting to the Chase

by Dejan Jovanovic and Leonardo M. de Moura (2011, 2013)

Claims:

- terminates with bounds ✓
- terminates without bounds; BUT: stuck states

Example

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ y \leq 1 \ \wedge \quad \begin{matrix} 2 \mid 2 \cdot 0 + 1 \\ 2 \mid 2x + y \end{matrix}$$

$$M := [\![x \geq 0, y \geq 0, y \leq 1, y \geq_{\mathbf{d}} 1, x \leq_{\mathbf{d}} 0]\!]$$

Conflict

Find responsible constraints under Assignment $x = 0$, $y = 1$;

Example

$$2 \mid 2 \cdot 0 + 1$$

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ y \leq 1 \ \wedge \quad 2 \mid 2x + y$$

$$M := [\![x \geq 0, y \geq 0, y \leq 1, y \geq_{\mathbf{d}} 1, x \leq_{\mathbf{d}} 0]\!]$$

Conflict

CUTSAT is stuck! But not at a solution, e.g., $x = 0$, $y = 0$

Finding Gaps

Find and fill gaps by turning:

dynamic termination strategy

$\Downarrow$

static quantifier elimination

Finding Gaps

Find and fill gaps by turning:

dynamic termination strategy

$$\Downarrow$$

static quantifier elimination

Result: New QE procedure (Weak Cooper Elimination)

Weak Cooper Elimination

Input: a conjunction $C(x)$ and a variable x to eliminate

Weak Cooper Elimination

Input: a conjunction $C(x)$ and a variable x to eliminate

Output:
- a conjunction C' without x;
- $\exists K.C' \equiv \exists x.C(x)$

Weak Cooper Elimination

Input: a conjunction $C(x)$ and a variable x to eliminate

Output:
- a conjunction C' without x;
- $\exists K.C' \equiv \exists x.C(x)$

Drawback: introduce new but bounded variables

Weak Cooper Elimination

Input:	a conjunction $C(x)$ and a variable x to eliminate
Output:	- a conjunction C' without x; - $\exists K.C' \equiv \exists x.C(x)$
Drawback:	introduce new but bounded variables
Termination:	eliminate all unbounded variables

Cores:

Divide constraints into cores:

If: an assignment ν is not a solution for $C(x)$

Then: there exists a responsible core that is conflicting under ν

Cores:

Divide constraints into cores:

If: an assignment ν is not a solution for $C(x)$

Then: there exists a responsible core that is conflicting under ν

Type	$ax \geq p$	$bx \leq q$	$d \mid cx + r$	conflicting

Cores:

Divide constraints into cores:

If: an assignment ν is not a solution for $C(x)$

Then: there exists a responsible core that is conflicting under ν

Type	$ax \geq p$	$bx \leq q$	$d \mid cx + r$	conflicting
1.Interval:				

Cores:

Divide constraints into cores:

If: an assignment ν is not a solution for $C(x)$

Then: there exists a responsible core that is conflicting under ν

Type	$ax \geq p$	$bx \leq q$	$d \mid cx + r$	conflicting
1.Interval:	1	1	0	

Cores:

Divide constraints into cores:

If: an assignment ν is not a solution for $C(x)$

Then: there exists a responsible core that is conflicting under ν

Type	$ax \geq p$	$bx \leq q$	$d \mid cx + r$	conflicting
1.Interval:	1	1	0	$0 \geq x \quad \wedge \quad x \geq 1$

Cores:

Divide constraints into cores:

If: an assignment ν is not a solution for $C(x)$

Then: there exists a responsible core that is conflicting under ν

Type	$ax \geq p$	$bx \leq q$	$d \mid cx + r$	conflicting
1.Interval:	1	1	0	$0 \geq x \quad \wedge \quad x \geq 1$
2.Divisibility:	1	1	1	

Cores:

Divide constraints into cores:

If: an assignment ν is not a solution for $C(x)$

Then: there exists a responsible core that is conflicting under ν

Type	$ax \geq p$	$bx \leq q$	$d \mid cx + r$	conflicting
1.Interval:	1	1	0	$0 \geq x \quad \wedge \quad x \geq 1$
2.Divisibility:	1	1	1	$2 \geq x \quad \wedge \quad x \geq 1 \quad \wedge \quad 3 \mid x$

Cores:

Divide constraints into cores:

If: an assignment ν is not a solution for $C(x)$

Then: there exists a responsible core that is conflicting under ν

Type	$ax \geq p$	$bx \leq q$	$d \mid cx + r$	conflicting
1.Interval:	1	1	0	$0 \geq x \quad \wedge \quad x \geq 1$
2.Divisibility:	1	1	1	$2 \geq x \quad \wedge \quad x \geq 1 \quad \wedge \quad 3 \mid x$
3.Diophantine:	0	0		

Cores:

Divide constraints into cores:

If: an assignment ν is not a solution for $C(x)$

Then: there exists a responsible core that is conflicting under ν

Type	$ax \geq p$	$bx \leq q$	$d \mid cx + r$	conflicting
1.Interval:	1	1	0	$0 \geq x \quad \wedge \quad x \geq 1$
2.Divisibility:	1	1	1	$2 \geq x \quad \wedge \quad x \geq 1 \quad \wedge \quad 3 \mid x$
3.Diophantine:	0	0	1	$2 \mid 2x + 1$

Cores:

Divide constraints into cores:

If: an assignment ν is not a solution for $C(x)$

Then: there exists a responsible core that is conflicting under ν

Type	$ax \geq p$	$bx \leq q$	$d \mid cx + r$	conflicting
1.Interval:	1	1	0	$0 \geq x \quad \wedge \quad x \geq 1$
2.Divisibility:	1	1	1	$2 \geq x \quad \wedge \quad x \geq 1 \quad \wedge \quad 3 \mid x$
3.Diophantine:	0	0	1	$2 \mid 2x + 1$

Note: Cores are not necessarily conflicting, e.g., $y \geq x \quad \wedge \quad x \geq z$

Weak Cooper Elimination

$$\exists x.C(x) \equiv \exists k_1, \ldots, k_{n'}. \left[\left(\bigwedge_{\text{all } I \text{ in } C(x) \text{ w/o } x} I \right) \wedge \left(\bigwedge_{\text{for each core } C^*} R_{C^*} \right) \right]$$

Resolvents R_{C^*}:

- R_{C^*} free of x
- R_{C^*} describes C^*'s solutions for x
- $k_1, \ldots, k_{n'}$ are new bounded variables

Weak Cooper Elimination

$$\exists x. C(x) \equiv \exists k_1, \ldots, k_{n'}. \left[\left(\bigwedge_{\text{all } I \text{ in } C(x) \text{ w/o } x} I \right) \wedge \left(\bigwedge_{\text{for each core } C^*} R_{C^*} \right) \right]$$

Resolvents R_{C^*}:

- R_{C^*} free of x
- R_{C^*} describes C^*'s solutions for x
- $k_1, \ldots, k_{n'}$ are new bounded variables

Weak Cooper Elimination

$$\exists x.C(x) \equiv \exists k_1, \ldots, k_{n'}. \left[\left(\bigwedge_{\text{all } I \text{ in } C(x) \text{ w/o } x} I \right) \wedge \left(\bigwedge_{\text{for each core } C^*} R_{C^*} \right) \right]$$

Resolvents R_{C^*}:

- R_{C^*} free of x
- R_{C^*} describes C^*'s solutions for x
- $k_1, \ldots, k_{n'}$ are new bounded variables

Weak Cooper Elimination

$$\exists x.C(x) \equiv \exists k_1, \ldots, k_{n'}. \left[\left(\bigwedge_{\text{all } I \text{ in } C(x) \text{ w/o } x} I \right) \wedge \left(\bigwedge_{\text{for each core } C^*} R_{C^*} \right) \right]$$

Resolvents R_{C^*}:

- R_{C^*} free of x
- R_{C^*} describes C^*'s solutions for x
- $k_1, \ldots, k_{n'}$ are new bounded variables

Example 1

Input:

$$3x \geq y \quad \wedge \quad x \leq 4z$$

After eliminating x:

$$\exists k.(k \geq 0 \wedge k \leq 2 \quad \wedge \quad 3 \mid y + k \quad \wedge \quad y + k \leq 12z)$$

Example 1

Input:

$$3x \geq y \quad \wedge \quad x \leq 4z$$

After eliminating x:

$$\exists k.(k \geq 0 \wedge k \leq 2 \quad \wedge \quad 3 \mid y + k \quad \wedge \quad y + k \leq 12z)$$

Solution for x from resolvent:

Example 1

Input:

$$3x \geq y \quad \wedge \quad x \leq 4z$$

After eliminating x:

$$\exists k.(k \geq 0 \wedge k \leq 2 \quad \wedge \quad 3 \mid y + k \quad \wedge \quad y + k \leq 12z)$$

Solution for x from resolvent:

$$x := \frac{y+k}{3} \in \mathbb{Z}$$

Example 1

Input:

$$3x \geq y \quad \wedge \quad x \leq 4z$$

After eliminating x:

$$\exists k.(k \geq 0 \wedge k \leq 2 \quad \wedge \quad 3 \mid y + k \quad \wedge \quad y + k \leq 12z)$$

Solution for x from resolvent:

$$x := \frac{y+k}{3} \in \mathbb{Z} \qquad \frac{y}{3} \leq \frac{y+k}{3} \leq 4z$$

Example 2

Input:

$$3x \geq y \quad \wedge \quad 2x \geq 2y - z \quad \wedge \quad x \leq 4z$$

After eliminating x:

$$\exists k_1.(k_1 \geq 0 \wedge k_1 \leq 2 \quad \wedge \quad 3 \mid y + k_1 \quad \wedge \quad y + k_1 \leq 12z)$$

$$\bigwedge$$

$$\exists k_2.(k_2 \geq 0 \wedge k_2 \leq 1 \quad \wedge \quad 2 \mid 2y - z + k_2 \quad \wedge \quad 2y - z + k_2 \leq 8z)$$

Example 3

Input:

$$x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ y \leq 1 \ \wedge \ 2 \mid 2x + y$$

After eliminating x:

Example 3

Input:

$$x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ y \leq 1 \ \wedge \ 2 \mid 2x + y$$

After eliminating x:

$$y \geq 0 \ \wedge \ y \leq 1 \ \wedge \ 2 \mid y$$

Example 3

Input:

$$x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ y \leq 1 \ \wedge \ 2 \mid 2x + y$$

After eliminating x:

$$y \geq 0 \ \wedge \ y \leq 1 \ \wedge \ 2 \mid y$$

Solution for x from resolvent:

Example 3

Input:

$$x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ y \leq 1 \ \wedge \ 2 \mid 2x + y$$

After eliminating x:

$$y \geq 0 \ \wedge \ y \leq 1 \ \wedge \ 2 \mid y$$

Solution for x from resolvent:

as y is even any $x \geq 0$

CUTSAT++

Idea:

- describe conflicts as cores
- learn resolvents to prevent a core from reverting into a conflict

CUTSAT++

Idea:

- describe conflicts as cores
- learn resolvents to prevent a core from reverting into a conflict

CUTSAT++

Idea:

- describe conflicts as cores
- learn resolvents to prevent a core from reverting into a conflict

CUTSAT++

Idea:

- describe conflicts as cores
- learn resolvents to prevent a core from reverting into a conflict

CUTSAT vs CUTSAT++:

- CUTSAT: only first two types of cores $\Rightarrow$ Stuck
- CUTSAT++: all types of cores

CUTSAT++

Idea:

- describe conflicts as cores
- learn resolvents to prevent a core from reverting into a conflict

CUTSAT vs CUTSAT++:

- CUTSAT: only first two types of cores $\Rightarrow$ Stuck
- CUTSAT++: all types of cores

CUTSAT++

Idea:

- describe conflicts as cores
- learn resolvents to prevent a core from reverting into a conflict

CUTSAT vs CUTSAT++:

- CUTSAT: only first two types of cores $\Rightarrow$ Stuck
- CUTSAT++: all types of cores

CUTSAT++

Idea:

- describe conflicts as cores
- learn resolvents to prevent a core from reverting into a conflict

CUTSAT vs CUTSAT++:

- CUTSAT: only first two types of cores $\Rightarrow$ Stuck
- CUTSAT++: all types of cores

Conclusions:

- CUTSAT++ integrates complete Weak Cooper Elimination
- CUTSAT++ always terminates

CUTSAT++

Idea:

- describe conflicts as cores
- learn resolvents to prevent a core from reverting into a conflict

CUTSAT vs CUTSAT++:

- CUTSAT: only first two types of cores $\Rightarrow$ Stuck
- CUTSAT++: all types of cores

Conclusions:

- CUTSAT++ integrates complete Weak Cooper Elimination
- CUTSAT++ always terminates

CUTSAT++

Idea:

- describe conflicts as cores
- learn resolvents to prevent a core from reverting into a conflict

CUTSAT vs CUTSAT++:

- CUTSAT: only first two types of cores $\Rightarrow$ Stuck
- CUTSAT++: all types of cores

Conclusions:

- CUTSAT++ integrates complete Weak Cooper Elimination
- CUTSAT++ always terminates

Example

$$C := x \geq 0 \;\wedge\; y \geq 0 \;\wedge\; y \leq 1 \;\wedge\; \begin{array}{l} 2 \mid 2 \cdot 0 + 1 \\ 2 \mid 2x + y \end{array}$$

$$M := [\![x \geq 0, y \geq 0, y \leq 1, y \geq_{\mathbf{d}} 1, x \leq_{\mathbf{d}} 0]\!]$$

Conflict

CUTSAT finds no core! CUTSAT is stuck!

Example

$$2 \mid 2 \cdot 0 + 1$$

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ y \leq 1 \ \wedge \quad 2 \mid 2x + y$$

$$M := [\![x \geq 0, y \geq 0, y \leq 1, y \geq_{\mathbf{d}} 1, x \leq_{\mathbf{d}} 0]\!]$$

Conflict

But CUTSAT++ finds a diophantine core!

Example

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ y \leq 1 \wedge \quad \begin{matrix} 2 \mid 2 \cdot 0 + 1 \\ 2 \mid 2x + y \end{matrix}$$

$$M := [\![x \geq 0, y \geq 0, y \leq 1, y \geq_{\mathbf{d}} 1, x \leq_{\mathbf{d}} 0]\!]$$

Conflict

Backtrack and block conflict with resolvent!

Example

$$C := x \geq 0 \;\wedge\; y \geq 0 \;\wedge\; y \leq 1 \;\wedge\quad 2 \mid 2x + y$$

$$M := [\![x \geq 0, y \geq 0, y \leq 1]\!]$$

Conflict

Backtrack and block conflict with resolvent!

Example

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ y \leq 1 \wedge \quad 2 \mid 2x + y \quad \wedge\, 2 \mid y$$

$$M := [\![x \geq 0, y \geq 0, y \leq 1]\!]$$

Conflict

Backtrack and block conflict with resolvent!

Example

$$C := x \geq 0 \;\wedge\; y \geq 0 \;\wedge\; y \leq 1 \wedge \quad 2 \mid 2x + y \quad \wedge\, 2 \mid y$$

$$M := [\![x \geq 0, y \geq 0, y \leq 1]\!]$$

Propagate

Propagate divisibility constraint!

Example

$$C := x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ y \leq 1 \wedge \quad 2 \mid 2x + y \quad \wedge\, 2 \mid y$$

$$M := [\![x \geq 0, y \geq 0, y \leq 1, y \leq 0]\!]$$

Propagate

Propagate divisibility constraint!

Example

$$C := x \geq 0 \;\wedge\; y \geq 0 \;\wedge\; y \leq 1 \;\wedge\quad 2 \mid 2x + y \quad \wedge\, 2 \mid y$$

$$M := [\![x \geq 0, y \geq 0, y \leq 1, y \leq 0]\!]$$

Decision

Assign variables with a decision!

Example

$$C := x \geq 0 \;\wedge\; y \geq 0 \;\wedge\; y \leq 1 \wedge \quad 2 \mid 2x + y \quad \wedge 2 \mid y$$

$$M := [\![x \geq 0, y \geq 0, y \leq 1, y \leq 0, x \leq_{\mathbf{d}} 0]\!]$$

Decision

Assign variables with a decision!

Example

$$C := x \geq 0 \;\wedge\; y \geq 0 \;\wedge\; y \leq 1 \;\wedge\; \begin{array}{l} 2 \mid 2 \cdot 0 + 0 \\ 2 \mid 2x + y \end{array} \;\wedge\; 2 \mid y$$

$$M := [\![x \geq 0, y \geq 0, y \leq 1, y \leq 0, x \leq_{\mathbf{d}} 0]\!]$$

Done

Complete model: $x := 0, y := 0$

Conclusions

CUTSAT++:

- fills termination gaps of CUTSAT
- by integrating Weak Cooper Elimination
- termination still oriented on problem structure

Conclusions

CUTSAT++:

- fills termination gaps of CUTSAT
- by integrating Weak Cooper Elimination
- termination still oriented on problem structure

Conclusions

CUTSAT++:

- fills termination gaps of CUTSAT
- by integrating Weak Cooper Elimination
- termination still oriented on problem structure

Conclusions

CUTSAT++:

- fills termination gaps of CUTSAT
- by integrating Weak Cooper Elimination
- termination still oriented on problem structure

Conclusions

CUTSAT++:

- fills termination gaps of CUTSAT
- by integrating Weak Cooper Elimination
- termination still oriented on problem structure

Future Research:

- a CUTSAT++ implementation
- benchmark results
- improvements to the procedure

Conclusions

CUTSAT++:

- fills termination gaps of CUTSAT
- by integrating Weak Cooper Elimination
- termination still oriented on problem structure

Future Research:

- a CUTSAT++ implementation
- benchmark results
- improvements to the procedure

Conclusions

CUTSAT++:

- fills termination gaps of CUTSAT
- by integrating Weak Cooper Elimination
- termination still oriented on problem structure

Future Research:

- a CUTSAT++ implementation
- benchmark results
- improvements to the procedure

Conclusions

CUTSAT++:

- fills termination gaps of CUTSAT
- by integrating Weak Cooper Elimination
- termination still oriented on problem structure

Future Research:

- a CUTSAT++ implementation
- benchmark results
- improvements to the procedure