TU Delft

Certificates and Proof Systems

Modern Constraint Programming (ESSAI'26)

Emir Demirović

Conflict Analysis

Algorithm 1 Conflict Analysis

Input: conflict explanation $(A_1 \wedge \cdots \wedge A_n \implies \bot)$
Output: asserting nogood

```
N ← conflict explanation
while N is not asserting do
    Let A be the most recently assigned atomic constraint in N
    Let R ⟹ A be the explanation of A
    Replace A in N by R
end while
return N =0
```

Propagation-based rewriting

Conflict preservation

Soundness

Termination

$$\langle x \geq 5 \rangle \land \langle y \geq 2 \rangle \land \langle z \geq 1 \rangle \land \langle z \leq 3 \rangle \Rightarrow \bot$$

$$\langle x \geq 5 \rangle \land \langle y \geq 2 \rangle \Rightarrow \langle z \neq 1 \rangle \land \langle z \neq 2 \rangle \land \langle z \neq 3 \rangle$$

$\langle x \geq 5 \rangle = \top$

$\langle y \geq 3 \rangle = \top$

$\langle z \geq 1 \rangle = ?$

$\langle z \leq 3 \rangle = ?$

We can propagate!

"extended nogood propagation"

Conflict Analysis with Extended Nogood Propagation

Algorithm 1 Conflict Analysis

Input: conflict explanation $(A_1 \wedge \cdots \wedge A_n \implies \bot)$

Output: ~~asserting nogood~~ propagating nogood

$N \leftarrow$ conflict explanation

while ~~N is not asserting~~ **do** N is not propagating

Let A be the most recently assigned atomic constraint in N

Let $R \implies A$ be the explanation of A

Replace A in N by R

end while

return N

More general learning!

Problem with two constraints

$$\sum w_i x_i \geq k \quad \wedge \quad \sum w_i x_i \leq k-1$$

Infeasible!

Easy problem, but exponential time for our search with conflict analysis!

Hypercube Linear Constraints

$$A_1 \wedge A_2 \wedge \cdots \wedge A_n \Rightarrow \sum w_i x_i \geq k$$

Generalises nogoods

$$A_1 \wedge A_2 \wedge \cdots \wedge A_n \Rightarrow \bot$$

Generalises linear inequalities

$$\top \Rightarrow \sum w_i x_i \geq k$$

In this lecture…

Certificates and Proof Systems

Justification for infeasibility/optimality claims about the instance

Discussion on recent approaches

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

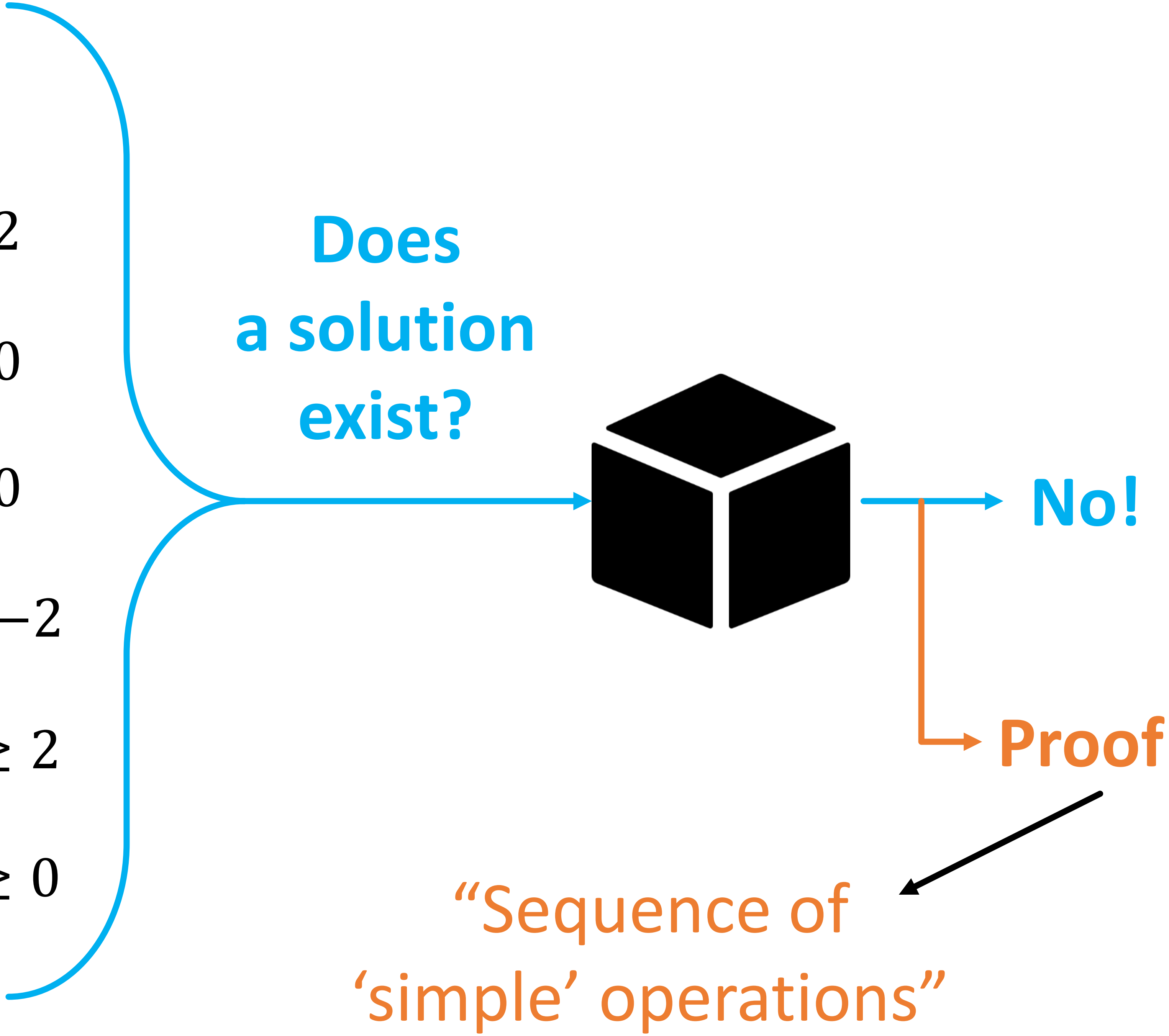
$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$



$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

Example Proof Sketch

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \geq 1]$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \geq 1]$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \leq 0]$

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \geq 1]$
$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \leq 0]$

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \geq 1]$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \leq 0]$

$[x \leq 0] \wedge [y \geq 2] \rightarrow \bot$

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1 : 2x + y + 2z \geq 2$

$c_2 : 2x + y - 2z \geq 0$

$c_3 : 2x - y + 2z \geq 0$

$c_4 : 2x - y - 2z \geq -2$

$c_5 : -2x + y + 2z \geq 2$

$c_6 : -2x + y - 2z \geq 0$

$[x \leq 0] \land [y \geq 2] \rightarrow [z \geq 1]$

$[x \leq 0] \land [y \geq 2] \rightarrow [z \leq 0]$

$[x \leq 0] \land [y \geq 2] \rightarrow \bot$

$[x \leq 0] \land [y \leq 1] \rightarrow [z \geq 1]$

$[x \leq 0] \land [y \leq 1] \rightarrow [z \leq 0]$

$[x \leq 0] \land [y \leq 1] \rightarrow \bot$

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

$[x \leq 0] \land [y \geq 2] \rightarrow [z \geq 1]$

$[x \leq 0] \land [y \geq 2] \rightarrow [z \leq 0]$

$[x \leq 0] \land [y \geq 2] \rightarrow \bot$

$[x \leq 0] \land [y \leq 1] \rightarrow [z \geq 1]$

$[x \leq 0] \land [y \leq 1] \rightarrow [z \leq 0]$

$[x \leq 0] \land [y \leq 1] \rightarrow \bot$

$[x \leq 0] \rightarrow \bot$

$x, z \in \{0,1\}$
$y \in \{0,1,2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \geq 1]$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \leq 0]$

$[x \leq 0] \wedge [y \geq 2] \rightarrow \bot$

$[x \leq 0] \wedge [y \leq 1] \rightarrow [z \geq 1]$

$[x \leq 0] \wedge [y \leq 1] \rightarrow [z \leq 0]$

$[x \leq 0] \wedge [y \leq 1] \rightarrow \bot$

$[x \leq 0] \rightarrow \bot$

$[x \geq 1] \rightarrow [z \geq 1]$

$[x \geq 1] \rightarrow [z \leq 0]$

$[x \geq 1] \rightarrow \bot$

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \geq 1]$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \leq 0]$

$[x \leq 0] \wedge [y \geq 2] \rightarrow \bot$

$[x \leq 0] \wedge [y \leq 1] \rightarrow [z \geq 1]$

$[x \leq 0] \wedge [y \leq 1] \rightarrow [z \leq 0]$

$[x \leq 0] \wedge [y \leq 1] \rightarrow \bot$

$[x \leq 0] \rightarrow \bot$

$[x \geq 1] \rightarrow [z \geq 1]$

$[x \geq 1] \rightarrow [z \leq 0]$

$[x \geq 1] \rightarrow \bot$

$\bot$

Certificate complete!

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \geq 1]$
$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \leq 0]$
$[x \leq 0] \wedge [y \geq 2] \rightarrow \bot$
$[x \leq 0] \wedge [y \leq 1] \rightarrow [z \geq 1]$
$[x \leq 0] \wedge [y \leq 1] \rightarrow [z \leq 0]$
$[x \leq 0] \wedge [y \leq 1] \rightarrow \bot$
$[x \leq 0] \rightarrow \bot$
$[x \geq 1] \rightarrow [z \geq 1]$
$[x \geq 1] \rightarrow [z \leq 0]$
$[x \geq 1] \rightarrow \bot$
$\bot$

Certificate traces solver behaviour

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

Arbitrary constraints

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: (|x| + 1) \cdot y = 2z$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \geq 1]$
$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \leq 0]$
$[x \leq 0] \wedge [y \geq 2] \rightarrow \perp$
$[x \leq 0] \wedge [y \leq 1] \rightarrow [z \geq 1]$
$[x \leq 0] \wedge [y \leq 1] \rightarrow [z \leq 0]$
$[x \leq 0] \wedge [y \leq 1] \rightarrow \perp$
$[x \leq 0] \rightarrow \perp$
$[x \geq 1] \rightarrow [z \geq 1]$
$[x \geq 1] \rightarrow [z \leq 0]$
$[x \geq 1] \rightarrow \perp$
$\perp$

(Same proof, but different constraints. Explanations are key.)

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

Use checker (Day 1)!

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \geq 1]$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \leq 0]$

$[x \leq 0] \wedge [y \geq 2] \rightarrow \bot$

$[x \leq 0] \wedge [y \leq 1] \rightarrow [z \geq 1]$

$[x \leq 0] \wedge [y \leq 1] \rightarrow [z \leq 0]$

$[x \leq 0] \wedge [y \leq 1] \rightarrow \bot$

$[x \leq 0] \rightarrow \bot$

$[x \geq 1] \rightarrow [z \geq 1]$

$[x \geq 1] \rightarrow [z \leq 0]$

$[x \geq 1] \rightarrow \bot$

$\bot$

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

Use checker (Day 3)!

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \geq 1]$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \leq 0]$

$[x \leq 0] \wedge [y \geq 2] \rightarrow \bot$

$[x \leq 0] \wedge [y \leq 1] \rightarrow [z \geq 1]$

$[x \leq 0] \wedge [y \leq 1] \rightarrow [z \leq 0]$

$[x \leq 0] \wedge [y \leq 1] \rightarrow \bot$

$[x \leq 0] \rightarrow \bot$

$[x \geq 1] \rightarrow [z \geq 1]$

$[x \geq 1] \rightarrow [z \leq 0]$

$[x \geq 1] \rightarrow \bot$

$\bot$

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1 : 2x + y + 2z \geq 2$

$c_2 : 2x + y - 2z \geq 0$

$c_3 : 2x - y$

$c_4 : 2x - y - 2z \geq -2$

$c_5 : -2x + y + 2z \geq 2$

$c_6 : -2x + y - 2z \geq 0$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \geq 1]$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \leq 0]$

$[x \leq 0] \wedge [y \geq 2] \rightarrow \bot$

$[x \leq 0] \wedge [y \leq 1] \rightarrow [z \geq 1]$

$z \leq 0]$

$[x \leq 0] \wedge [y \leq 1] \rightarrow \bot$

$[x \leq 0] \rightarrow \bot$

$[x \geq 1] \rightarrow [z \geq 1]$

$[x \geq 1] \rightarrow [z \leq 0]$

$[x \geq 1] \rightarrow \bot$

$\bot$

We can verify the proof independently from the solver!

Another Example

$$c_1: 5x + 3y \leq 6$$

$$c_2: x + y + z \geq 3$$

$$x \in \{0, 1, 2\}$$
$$y, z \in \{0, 1\}$$

Construct a certificate of infeasibility!

$c_1: 5x + 3y \leq 6$

$c_2: x + y + z \geq 3$

$x \in \{0, 1, 2\}$
$y, z \in \{0, 1\}$

$$\langle x \geq 1 \rangle \wedge \langle y \geq 0 \rangle \wedge \langle z \leq 1 \rangle \implies \bot$$

Claim

this nogood is implied by the constraints

ID	Derived constraint
c_3	$c_1 \models \langle y \geq 0 \rangle \implies \langle x \leq 1 \rangle$
c_4	$c_1 \models \langle x \geq 1 \rangle \implies \langle y \leq 0 \rangle$
c_5	$c_2 \models \langle x \leq 1 \rangle \wedge \langle z \leq 1 \rangle \implies \langle y \geq 1 \rangle$
c_6	$c_3 \wedge c_4 \wedge c_5 \models \langle x \geq 1 \rangle \wedge \langle y \geq 0 \rangle \wedge \langle z \leq 1 \rangle \implies \bot$

c_1: $5x + 3y \leq 6$

c_2: $x + y + z \geq 3$

$x \in \{0, 1, 2\}$
$y, z \in \{0, 1\}$

Proof of the claim

(similar as when verifying learned nogoods)

$c_1: 5x + 3y \leq 6$

$c_2: x + y + z \geq 3$

$x \in \{0, 1, 2\}$
$y, z \in \{0, 1\}$

$\langle y \geq 0 \rangle \wedge \langle y \leq 1 \rangle \wedge \langle z \leq 1 \rangle \implies \bot$

New claim

$c_1: 5x + 3y \leq 6$

$c_2: x + y + z \geq 3$

$x \in \{0, 1, 2\}$
$y, z \in \{0, 1\}$

c_6	$c_3 \wedge c_4 \wedge c_5$	$\models$	$\langle x \geq 1 \rangle \wedge \langle y \geq 0 \rangle \wedge \langle z \leq 1 \rangle \implies \bot$
c_7	c_6	$\models$	$\langle y \geq 0 \rangle \wedge \langle z \leq 1 \rangle \implies \langle x \leq 0 \rangle$
c_8	c_2	$\models$	$\langle y \leq 1 \rangle \wedge \langle z \leq 1 \rangle \implies \langle x \geq 1 \rangle$
c_9	$c_7 \wedge c_8$	$\models$	$\langle y \geq 0 \rangle \wedge \langle y \leq 1 \rangle \wedge \langle z \leq 1 \rangle \implies \bot$

Proof of the claim

$c_1: 5x + 3y \leq 6$

$c_2: x + y + z \geq 3$

$x \in \{0, 1, 2\}$
$y, z \in \{0, 1\}$

c_6	$c_3 \wedge c_4 \wedge c_5$	$\models$	$\langle x \geq 1 \rangle \wedge \langle y \geq 0 \rangle \wedge \langle z \leq 1 \rangle \implies \bot$
c_7	c_6	$\models$	$\langle y \geq 0 \rangle \wedge \langle z \leq 1 \rangle \implies \langle x \leq 0 \rangle$
c_8	c_2	$\models$	$\langle y \leq 1 \rangle \wedge \langle z \leq 1 \rangle \implies \langle x \geq 1 \rangle$
c_9	$c_7 \wedge c_8$	$\models$	$\langle y \geq 0 \rangle \wedge \langle y \leq 1 \rangle \wedge \langle z \leq 1 \rangle \implies \bot$

Previously proven claim used

$c_1 : 5x + 3y \leq 6$

$c_2 : x + y + z \geq 3$

$x \in \{0, 1, 2\}$
$y, z \in \{0, 1\}$

Refers to the bounds of variables

(not used so far)

$\top \models \top \Longrightarrow \langle y \geq 0 \rangle$

$c_1: 5x + 3y \leq 6$

$c_2: x + y + z \geq 3$

$x \in \{0, 1, 2\}$
$y, z \in \{0, 1\}$

Final claim of infeasibility

$\top \implies \bot$

Proof of the final claim

$c_1: 5x + 3y \leq 6$

$c_2: x + y + z \geq 3$

$x \in \{0, 1, 2\}$
$y, z \in \{0, 1\}$

c_9	$c_7 \wedge c_8$	$\models$	$\langle y \geq 0 \rangle \wedge \langle y \leq 1 \rangle \wedge \langle z \leq 1 \rangle \implies \bot$
c_{10}	$\top$	$\models$	$\top \implies \langle y \geq 0 \rangle$
c_{11}	$\top$	$\models$	$\top \implies \langle y \leq 1 \rangle$
c_{12}	$\top$	$\models$	$\top \implies \langle z \leq 1 \rangle$
c_{13}	c_9	$\models$	$\langle y \geq 0 \rangle \wedge \langle y \leq 1 \rangle \wedge \langle z \leq 1 \rangle \implies \bot$
c_{14}	$c_{10} \wedge c_{11} \wedge c_{12} \wedge c_{13}$	$\models$	$\top \implies \bot$

Certificate of infeasibility

$c_1: 5x + 3y \leq 6$

$c_2: x + y + z \geq 3$

$x \in \{0, 1, 2\}$
$y, z \in \{0, 1\}$

ID			Derived constraint
c_3	c_1	$\models$	$\langle y \geq 0 \rangle \implies \langle x \leq 1 \rangle$
c_4	c_1	$\models$	$\langle x \geq 1 \rangle \implies \langle y \leq 0 \rangle$
c_5	c_2	$\models$	$\langle x \leq 1 \rangle \wedge \langle z \leq 1 \rangle \implies \langle y \geq 1 \rangle$
c_6	$c_3 \wedge c_4 \wedge c_5$	$\models$	$\langle x \geq 1 \rangle \wedge \langle y \geq 0 \rangle \wedge \langle z \leq 1 \rangle \implies \bot$
c_7	c_6	$\models$	$\langle y \geq 0 \rangle \wedge \langle z \leq 1 \rangle \implies \langle x \leq 0 \rangle$
c_8	c_2	$\models$	$\langle y \leq 1 \rangle \wedge \langle z \leq 1 \rangle \implies \langle x \geq 1 \rangle$
c_9	$c_7 \wedge c_8$	$\models$	$\langle y \geq 0 \rangle \wedge \langle y \leq 1 \rangle \wedge \langle z \leq 1 \rangle \implies \bot$
c_{10}	$\top$	$\models$	$\top \implies \langle y \geq 0 \rangle$
c_{11}	$\top$	$\models$	$\top \implies \langle y \leq 1 \rangle$
c_{12}	$\top$	$\models$	$\top \implies \langle z \leq 1 \rangle$
c_{13}	c_9	$\models$	$\langle y \geq 0 \rangle \wedge \langle y \leq 1 \rangle \wedge \langle z \leq 1 \rangle \implies \bot$
c_{14}	$c_{10} \wedge c_{11} \wedge c_{12} \wedge c_{13}$	$\models$	$\top \implies \bot$

$c_1: 5x + 3y \leq 6$

$c_2: x + y + z \geq 3$

$x \in \{0, 1, 2\}$
$y, z \in \{0, 1\}$

ID	Derived constraint		
c_3	c_1	$\models$	$\langle y \geq 0 \rangle \implies \langle x \leq 1 \rangle$
c_4	c_1	$\models$	$\langle x \geq 1 \rangle \implies \langle y \leq 0 \rangle$
c_5	c_2	$\models$	$\langle x \leq 1 \rangle \land \langle z \leq 1 \rangle \implies \langle y \geq 1 \rangle$
c_6	$c_3 \land c_4 \land c_5$	$\models$	$\langle x \geq 1 \rangle \land \langle y \geq 0 \rangle \land \langle z \leq 1 \rangle \implies \bot$
c_7	c_6	$\models$	$\langle y \geq 0 \rangle \land \langle z \leq 1 \rangle \implies \langle x \leq 0 \rangle$
c_8	c_2	$\models$	$\langle y \leq 1 \rangle \land \langle z \leq 1 \rangle \implies \langle x \geq 1 \rangle$
c_9	$c_7 \land c_8$	$\models$	$\langle y \geq 0 \rangle \land \langle y \leq 1 \rangle \land \langle z \leq 1 \rangle \implies \bot$
c_{10}	$\top$	$\models$	$\top \implies \langle y \geq 0 \rangle$
c_{11}	$\top$	$\models$	$\top \implies \langle y \leq 1 \rangle$
c_{12}	$\top$	$\models$	$\top \implies \langle z \leq 1 \rangle$
c_{13}	c_9	$\models$	$\langle y \geq 0 \rangle \land \langle y \leq 1 \rangle \land \langle z \leq 1 \rangle \implies \bot$
c_{14}	$c_{10} \land c_{11} \land c_{12} \land c_{13}$	$\models$	$\top \implies \bot$

Claim about the instance, not the solver!

(Practical)
Proof Systems

Formal language to specify constraints

(Practical)
Proof Systems

Formal language to specify constraints

Operations to derive new constraints
(polytime)

(Practical)
Proof Systems

Formal language to specify constraints

Operations to derive new constraints
(polytime)

Easy to verify proofs
(polytime)

(Practical)
Proof Systems

Not tied to a solver!

Formal language to specify constraints

Operations to derive new constraints (polytime)

Easy to verify proofs (polytime)

(Proof systems designed to mimic solver behaviour)

CP Proof System

Formal language to specify constraints

Constraint Programming

CP Proof System

Formal language to specify constraints

Constraint Programming

Variables

$\boldsymbol{x_i \in D_i \subset Z}$

$x \in \{0, 1, 2\}$

$y, z \in \{0, 1\}$

CP Proof System

Formal language to specify constraints

Constraint Programming

Variables

$\boldsymbol{x_i \in D_i \subset Z}$

$x \in \{0, 1, 2\}$

$y, z \in \{0, 1\}$

Constraints

$\boldsymbol{c}: \boldsymbol{Z^n} \rightarrow \{\top, \bot\}$

$c_1: 5x + 3y \leq 6$

$c_2: x + y + z \geq 3$

CP Proof System

Operations to derive new constraints

"combine constraints"

CP Proof System

Operations to derive new constraints

"combine constraints"

Inferences

$$c_1: 5x + 3y \leq 6$$

$$c_1 \vDash \langle y \geq 0 \rangle \Rightarrow \langle x \leq 1 \rangle$$

CP Proof System

Operations to derive new constraints

“combine constraints”

Inferences

c_5: All-Different(x, y, z)

$$c_5 \models \langle x \geq 1 \rangle \land \langle x \leq 2 \rangle \land \langle y \geq 1 \rangle \land \langle y \leq 2 \rangle \Rightarrow \langle z \neq 1 \rangle$$

CP Proof System

Operations to derive new constraints

“combine constraints”

c_3	c_1	$\models$	$\langle y \geq 0 \rangle \implies \langle x \leq 1 \rangle$
c_4	c_1	$\models$	$\langle x \geq 1 \rangle \implies \langle y \leq 0 \rangle$
c_5	c_2	$\models$	$\langle x \leq 1 \rangle \land \langle z \leq 1 \rangle \implies \langle y \geq 1 \rangle$
c_6	$c_3 \land c_4 \land c_5$	$\models$	$\langle x \geq 1 \rangle \land \langle y \geq 0 \rangle \land \langle z \leq 1 \rangle \implies \bot$

Proof Stage

derives a constraint

CP Proof System

Operations to derive new constraints

"combine constraints"

Inferences → Proof Stage

↓

new constraint

CP Proof System

Easy to verify proofs

Checkers

Inference Checkers (Day 1)

Deduction Checkers (Day 3)

CP Proof System

Constraint Programming Language

Inferences and Proof Stages

Checkers

Up next...

Certificate Generation

Discussion on recent approaches

How to generate certificates?

Explicitly record solver behaviour

Multi-stage framework

Certificate generation: log everything

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

$[x \leq 0] \land [y \geq 2] \rightarrow [z \geq 1]$

$[x \leq 0] \land [y \geq 2] \rightarrow [z \leq 0]$

$[x \leq 0] \land [y \geq 2] \rightarrow \bot$

$[x \leq 0] \land [y \leq 1] \rightarrow [z \geq 1]$

$[x \leq 0] \land [y \leq 1] \rightarrow [z \leq 0]$

$[x \leq 0] \land [y \leq 1] \rightarrow \bot$

$[x \leq 0] \rightarrow \bot$

$[x \geq 1] \rightarrow [z \geq 1]$

$[x \geq 1] \rightarrow [z \leq 0]$

$[x \geq 1] \rightarrow \bot$

$\bot$

Certificate generation: log everything

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

Trail

$x \leq 0 \ @ \ 1$

$y \geq 2 \ @ \ 2$

$c_3 \rightarrow z \geq 1$

$c_4 \rightarrow \perp \ (z \leq 0)$

Certificate generation: log everything

$x, z \in \{0,1\}$
$y \in \{0,1,2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

Trail

$x \leq 0 \; @ \; 1$

$y \geq 2 \; @ \; 2$

$c_3 \rightarrow z \geq 1$

$c_4 \rightarrow \bot \; (z \leq 0)$

Translating into the proof system…

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \geq 1]$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \leq 0]$

$[x \leq 0] \wedge [y \geq 2] \rightarrow \bot$

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

Translating every solver step into the proof format

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

$[x \leq 0] \land [y \geq 2] \rightarrow [z \geq 1]$

$[x \leq 0] \land [y \geq 2] \rightarrow [z \leq 0]$

$[x \leq 0] \land [y \geq 2] \rightarrow \bot$

$[x \leq 0] \land [y \leq 1] \rightarrow [z \geq 1]$

$[x \leq 0] \land [y \leq 1] \rightarrow [z \leq 0]$

$[x \leq 0] \land [y \leq 1] \rightarrow \bot$

$[x \leq 0] \rightarrow \bot$

$[x \geq 1] \rightarrow [z \geq 1]$

$[x \geq 1] \rightarrow [z \leq 0]$

$[x \geq 1] \rightarrow \bot$

$\bot$

(Similarly for the other parts)

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

Translating every solver step into the proof format

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

Expensive in practice

$c_6: -2x + y - 2z \geq 0$

$[x \leq 0] \land [y \geq 2] \rightarrow [z \geq 1]$

$[x \leq 0] \land [y \geq 2] \rightarrow [z \leq 0]$

$[x \leq 0] \land [y \geq 2] \rightarrow \bot$

$[x \leq 0] \land [y \leq 1] \rightarrow [z \geq 1]$

$[x \leq 0] \land [y \leq 1] \rightarrow [z \leq 0]$

$[x \leq 0] \land [y \leq 1] \rightarrow \bot$

$[x \leq 0] \rightarrow \bot$

$[x \geq 1] \rightarrow [z \geq 1]$

$[x \geq 1] \rightarrow [z \leq 0]$

$[x \geq 1] \rightarrow \bot$

$\bot$

(Similarly for the other parts)

Generate certificates in stages

Record a proof scaffold → inexpensive

Trim redundancies

Expand into a full certificate

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

Scaffold: list of nogoods

$c_1: 2x + y + 2z$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \geq 1]$

$[x \leq 0] \wedge [y \geq 2] \rightarrow [z \leq 0]$

$[x \leq 0] \wedge [y \geq 2] \rightarrow \perp$

$[x \leq 0] \wedge [y \leq 1] \rightarrow [z \geq 1]$

$[x \leq 0] \wedge [y \leq 1] \rightarrow [z \leq 0]$

$[x \leq 0] \wedge [y \leq 1] \rightarrow \perp$

$[x \leq 0] \rightarrow \perp$

$[x \geq 1] \rightarrow [z \geq 1]$

$[x \geq 1] \rightarrow [z \geq 0]$

$[x \geq 1] \rightarrow \perp$

$\perp$

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1 \colon 2x + y + 2z$

$c_2 \colon 2x + y - 2z \geq 0$

$c_3 \colon 2x - y + 2z \geq 0$

$c_4 \colon 2x - y - 2z \geq -2$

$c_5 \colon -2x + y + 2z \geq 2$

$c_6 \colon -2x + y - 2z \geq 0$

Scaffold: list of nogoods

$[x \leq 0] \wedge [y \geq 2] \rightarrow \bot$

$[x \leq 0] \rightarrow \bot$

$\bot$

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x$

$c_4: 2x$

Scaffold can contain redundancy

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

$[x \leq 0] \wedge [y \geq 2] \rightarrow \bot$

$[y \leq 0] \wedge [z \geq 1] \rightarrow \bot$

$[x \leq 0] \rightarrow \bot$

$[y \leq 0] \wedge [z \leq 0] \rightarrow \bot$

$\bot$

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x$

Simplify!
(trim)

$c_4: 2x$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

$[x \leq 0] \land [y \geq 2] \rightarrow \bot$

~~$[y \leq 0] \land [z \geq 1] \rightarrow \bot$~~

$[x \leq 0] \rightarrow \bot$

~~$[y \leq 0] \land [z \leq 0] \rightarrow \bot$~~

$\bot$

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z$

Expand!

$[x \leq 0] \wedge [y \geq 2] \rightarrow \bot$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$[x \leq 0] \rightarrow \bot$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

$\bot$

$x, z \in \{0,1\}$
$y \in \{0, 1, 2\}$

$c_1: 2x + y + 2z \geq 2$

$c_2: 2x + y - 2z \geq 0$

$c_3: 2x - y + 2z \geq 0$

$c_4: 2x - y - 2z \geq -2$

$c_5: -2x + y + 2z \geq 2$

$c_6: -2x + y - 2z \geq 0$

Full certificate

$[x \leq 0] \land [y \geq 2] \rightarrow [z \geq 1]$

$[x \leq 0] \land [y \geq 2] \rightarrow [z \leq 0]$

$[x \leq 0] \land [y \geq 2] \rightarrow \bot$

$[x \leq 0] \land [y \leq 1] \rightarrow [z \geq 1]$

$[x \leq 0] \land [y \leq 1] \rightarrow [z \leq 0]$

$[x \leq 0] \land [y \leq 1] \rightarrow \bot$

$[x \leq 0] \rightarrow \bot$

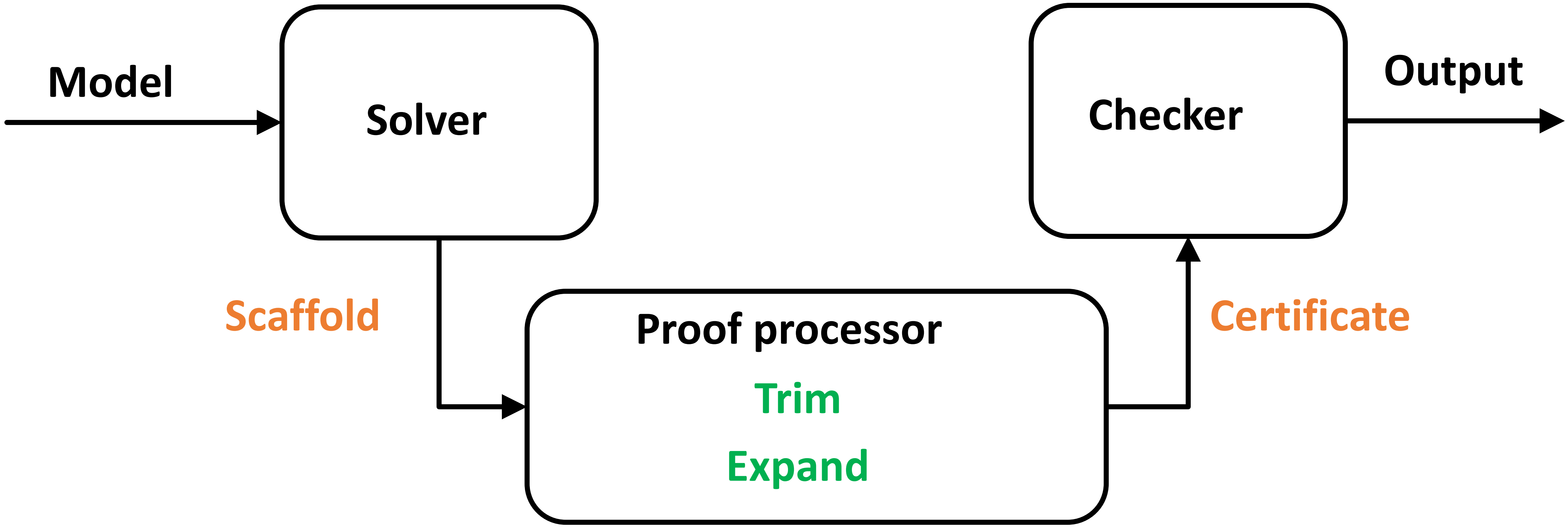
$[x \geq 1] \rightarrow [z \geq 1]$

$[x \geq 1] \rightarrow [z \geq 0]$

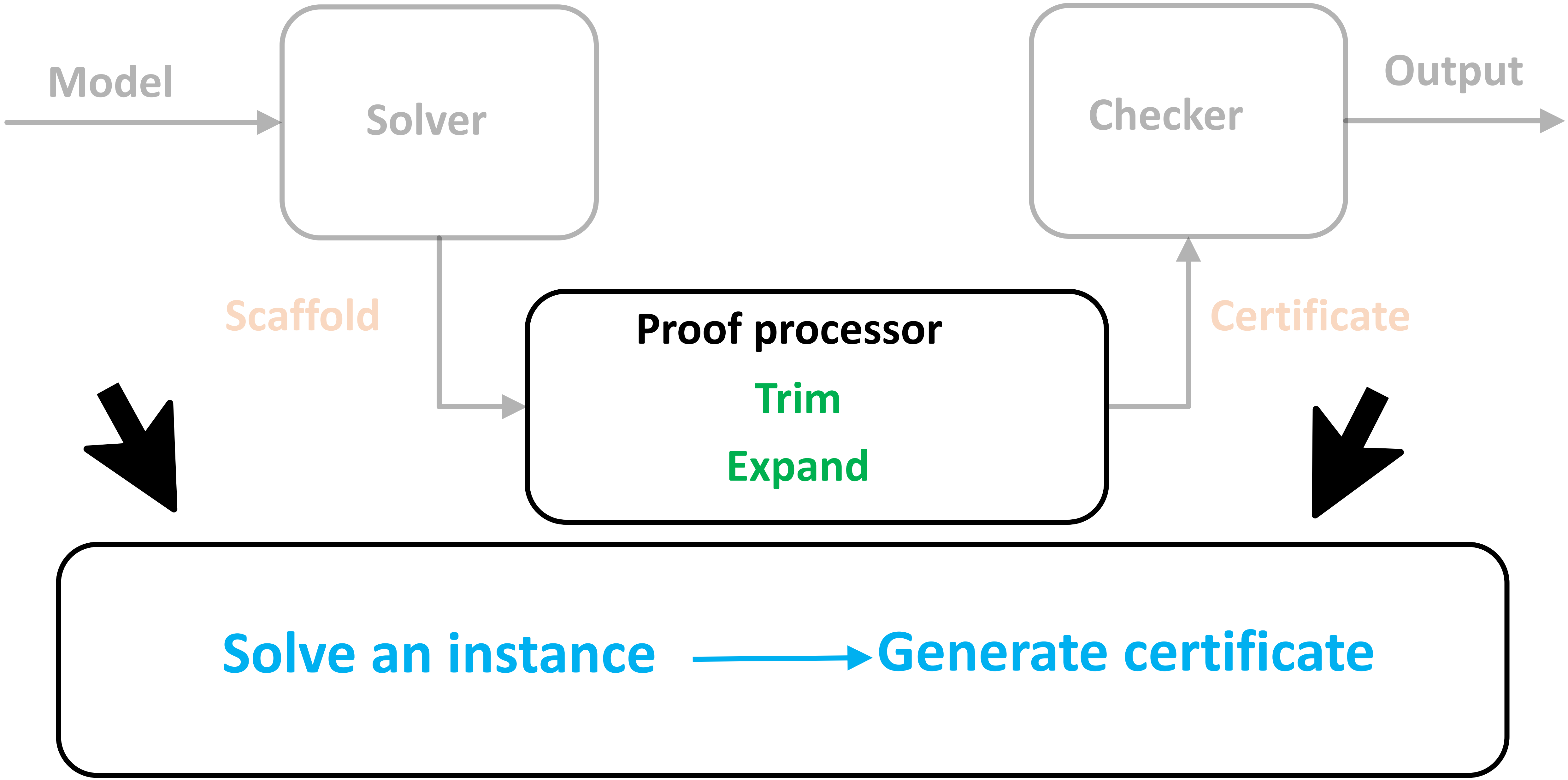
$[x \geq 1] \rightarrow \bot$

$\bot$

Our multi-stage workflow



Our multi-stage workflow



Summary: Modern Constraint Programming

Summary: Modern Constraint Programming

Day 1

Search,
Propagation, Checkers

Summary: Modern Constraint Programming

Day 1

Search,
Propagation, Checkers

Day 2

Propagation
(All-Different, Cumulative)

Summary: Modern Constraint Programming

Day 1
Search,
Propagation, Checkers

Day 2
Propagation
(All-Different, Cumulative)

Day 3
Conflict Analysis
(Extended) Nogoods
Hypercubes

Summary: Modern Constraint Programming

Day 1
Search,
Propagation, Checkers

Day 2
Propagation
(All-Different, Cumulative)

Day 3
Conflict Analysis
(Extended) Nogoods
Hypercubes

Day 4
Certification
and Proof Systems

Recent developments

Disclaimer:

Very biased selection of works related to this course!

Recent developments

Conflict analysis for SAT

“GRASP: A search algorithm for propositional satisfiability”

Marques-Silva, Sakallah; 1996

Recent developments

Conflict analysis for SAT

"GRASP: A search algorithm for propositional satisfiability"

Marques-Silva, Sakallah; 1996

Many related works in between…

SAT-CP conflict analysis

"Propagation via lazy clause generation"

Ohrimenko, Stuckey; 2009

"Lazy clause generation reengineered"

Feydy, Stuckey; 2009

Recent developments

Conflict analysis for SAT

"GRASP: A search algorithm for propositional satisfiability"

Marques-Silva, Sakallah; 1996

How to learn from conflicts?

orks

in between…

SAT-CP conflict analysis

"Propagation via lazy clause generation"

Ohrimenko, Stuckey; 2009

"Lazy clause generation reengineered"

Feydy, Stuckey; 2009

Recent developments

“From Literals to Atomic Constraints: Generalising Conflict-Driven Clause Learning for Constraint Programming”

Marijnissen, Flippo, Demirović; CP 2026

~~Hybrid SAT-CP~~

<u>Native CP CDCL</u>

Viewpoint used in this course!

Extended nogood propagation

Recent developments

Cutting planes
0-1 Linear Inequalities

“Divide and Conquer: Towards Faster Pseudo-Boolean Solving”

Elffers, Nordström; IJCAI 2018

Theoretically
more powerful than
SAT conflict analysis

0-1 problems

Recent developments

Cutting planes
0-1 Linear Inequalities

"Divide and Conquer: Towards Faster Pseudo-Boolean Solving"

Elffers, Nordström; IJCAI 2018

Theoretically
more powerful than
SAT conflict analysis

0-1 problems

Cutting planes
for CP

"Conflict Analysis Based on Cutting-Planes for Constraint Programming"

Baauw, Flippo, Demirović; CP 2025

Incomplete
but can be more powerful

Integer problems

Recent developments

Cutting planes
0-1 Linear Inequalities

"Divide and Conquer: Towards Faster Pseudo-Boolean Solving"

Elffers, Nordström; IJCAI 2018

Cutting planes
for CP

"Conflict Analysis Based on Cutting-Planes for Constraint Programming"

Baauw, Flippo, Demirović; CP 2025

Hypercube Linear
Conflict Analysis for CP

"Resolution Meets Cutting Planes: Introducing Hypercube Linear Resolution"

Flippo, Stuckey, Demirović; CPAIOR 2026

Complete!
Generalises cutting planes for CP

Recent developments

Cutting planes
0-1 Linear Inequalities

“Divide and Conquer: Towards Faster Pseudo-Boolean Solving”

Elffers, N... IJCAI

Cutting planes
for CP

“Conflict Analysis Based on Cutting-Planes for Constraint Programming”

...ć; CP 2025

How to combine more expressive constraints?

Hypercube Linear
Conflict Analysis for CP

“Resolution Meets Cutting Planes: Introducing Hypercube Linear Resolution”

Flippo, Stuckey, Demirović; CPAIOR 2026

Complete!
Generalises cutting planes for CP

Recent developments

Certificates for CP
as 0-1 Linear Inequalities

"An Auditable Constraint Programming Solver"

Gocht, McCreesh, Nordström; CP 2022

Recent developments

Certificates for CP as 0-1 Linear Inequalities

"An Auditable Constraint Programming Solver"

Gocht, McCreesh, Nordström; CP 2022

Formally verified Native CP Proof System

"Formally Verified Certification of Constraint Programming Proofs"

Flippo, Sidorov, Ten Brink, Pit-Claudel, Demirović; CP 2026

Practical generation of certificates

"A Multi-Stage Proof Logging Framework to Certify the Correctness of CP Solvers"

Flippo, Sidorov, Marijnissen, Smits, Demirović; CP 2024

(used in this course!)

Recent developments

Certificates for CP
as 0-1 Linear Inequalities

"An Auditable Constraint Programming Solver"

Gocht, McCreesh, Nordström; CP 2022

How to certify claims of CP solvers?

Formally verified Native CP Proof System

"Formally Verified Certification of Constraint Programming Proofs"

Flippo, Sidorov, Ten Brink, Pit-Claudel, Demirović; CP 2026

Practical generation of certificates

"A Multi-Stage Proof Logging Framework

to Certify the Correctness of CP Solvers"

Flippo, Sidorov, Marijnissen, Smits, Demirović; CP 2024

(used in this course!)

Recent developments

Make certificates more human-understandable

"Using Certifying Constraint Solvers

for Generating Step-wise Explanations"

Bleukx, Flippo, Bogaerts, Demirović, Guns; AAAI 2026

Recent developments

Make certificates more human-understandable

"Using Certifying Constraint Solvers

for Generating Step-wise Explanations"

Bleukx, Flippo, Bogaerts, Demirović, Guns; AAAI 2026

Transfer CP propagation to Dynamic Programming

"Domain-Independent Dynamic Programming

with Constraint Propagation"

Marijnissen, Beck, Demirović, Kuroiwa; ICAPS 2026

Recent developments

Make certificates more human-understandable

"Using Certifying Constraint Solvers

for Generating Step-wise Explanations"

Bleukx, Flippo, Bogaerts, Demirović, Guns; AAAI 2026

Transfer CP propagation to Dynamic Programming

"Domain-Independent Dynamic Programming

with Constraint Propagation"

Marijnissen, Beck, Demirović, Kuroiwa; ICAPS 2026

Pumpkin (CP Solver)

Rust

Research vessel

Implements techniques from the course

Summary: Modern Constraint Programming

Day 1
Search,
Propagation, Checkers

Day 2
Propagation
(All-Different, Cumulative)

Day 3
Conflict Analysis
(Extended) Nogoods
Hypercubes

Day 4
Certification
and Proof Systems