Proof Logging MaxCDCL and MDD-encodings

Dieter Vandesande
Joint work with Bart Bogaerts and Jordi Coll

May 23, 2024

ARTIFICIAL INTELLIGENCE RESEARCH GROUP

OUTLINE OF THIS PRESENTATION

- What is MaxSAT and how to certify it?
- Proof logging the B&B solver MaxCDCL
- Proof logging additional techniques in MaxCDCL
  - Hardening
  - Literal Unlocking
- Proof logging BDD PB-to-CNF encoding
- Future work & Conclusions

OUTLINE OF THIS PRESENTATION

- What is MaxSAT and how to certify it?
- Proof logging the B&B solver MaxCDCL
- Proof logging additional techniques in MaxCDCL
  - Hardening
  - Literal Unlocking
- Proof logging BDD PB-to-CNF encoding
- Future work & Conclusions

THE MAXIMUM SATISFIABILITY PROBLEM

Example:

$$F = \{x_1 \lor x_2,\ x_2 \lor x_3,\ x_1 \lor \overline{x_2} \lor x_3\}$$
$$\mathcal{O} = x_1 + x_2 + x_3$$

Optimization variant of Satisfiability Problem.

A MaxSAT-instance is a tuple $(F, \mathcal{O})$ with:

- F a propositional formula
- $\mathcal{O}$ an integer linear objective over Boolean variables

THE MAXIMUM SATISFIABILITY PROBLEM

Example:

$F = \{\boldsymbol{x_1} \vee x_2,\ x_2 \vee \boldsymbol{x_3},\ \boldsymbol{x_1} \vee \overline{\boldsymbol{x_2}} \vee \boldsymbol{x_3}\}$

$\mathcal{O} = \boldsymbol{x_1} + x_2 + \boldsymbol{x_3}$

Solution: $\alpha = \{\boldsymbol{x_1} \mapsto 1, x_2 \mapsto 0, \boldsymbol{x_3} \mapsto 1\}$

Optimization variant of Satisfiability Problem.

A MaxSAT-instance is a tuple $(F, \mathcal{O})$ with:

- F a propositional formula
- $\mathcal{O}$ an integer linear objective over Boolean variables

A solution is an assignment for all variables such that:

- The formula F is satisfied
- No other satisfying assignment has lower objective value

PROOF SYSTEMS FOR MAXSAT REASONING

PROOF SYSTEMS FOR MAXSAT REASONING

Proof systems for MaxSAT are studied theoretically for proof complexity

- MaxSAT resolution [LH05, HL06, BLM06, BLM07]
- Tableaux reasoning [LMS16, LCH+22, LM22]
- Cost-aware redundancy notions [BMM13, BJ19, IBJ22]

PROOF SYSTEMS FOR MAXSAT REASONING

Proof systems for MaxSAT are studied theoretically for proof complexity

- MaxSAT resolution [LH05, HL06, BLM06, BLM07]
- Tableaux reasoning [LMS16, LCH^{+}22, LM22]
- Cost-aware redundancy notions [BMM13, BJ19, IBJ22]

Solvers specifically designed for emitting proofs

- MaxSAT resolution [PCH21, PCH22]
- Cost Resolution [LNOR11]

PROOF SYSTEMS FOR MAXSAT REASONING

Proof systems for MaxSAT are studied theoretically for proof complexity

- MaxSAT resolution [LH05, HL06, BLM06, BLM07]
- Tableaux reasoning [LMS16, LCH^{+}22, LM22]
- Cost-aware redundancy notions [BMM13, BJ19, IBJ22]

Solvers specifically designed for emitting proofs

- MaxSAT resolution [PCH21, PCH22]
- Cost Resolution [LNOR11]

No certified state-of-the-art MaxSAT solver using native proof system!

MAXSAT SOLVERS

Four main categories:

- Model-Improving
- Core-Guided
- Implicit Hitting Set
- Branch-and-Bound

MAXSAT SOLVERS

Four main categories:

- Model-Improving
  - SAT-based
- Core-Guided
  - SAT-based
- Implicit Hitting Set
  - SAT-based
- Branch-and-Bound

MAXSAT SOLVERS

Four main categories:

- Model-Improving
  - SAT-based
  - Use PB-to-CNF encodings to encode model-improving constraint
- Core-Guided
  - SAT-based
  - Use PB-to-CNF encodings to relax unsat cores
- Implicit Hitting Set
  - SAT-based
  - Use a closed-source MIP solver to guide search
- Branch-and-Bound

MAXSAT SOLVERS

Four main categories:

- Model-Improving
  - SAT-based
  - Use PB-to-CNF encodings to encode model-improving constraint
- Core-Guided
  - SAT-based
  - Use PB-to-CNF encodings to relax unsat cores
- Implicit Hitting Set
  - SAT-based
  - Use a closed-source MIP solver to guide search
- Branch-and-Bound
  - Solves MaxSAT “natively”

MAXSAT SOLVERS

Four main categories:

- Model-Improving
  - SAT-based
  - Use PB-to-CNF encodings to encode model-improving constraint
- Core-Guided
  - SAT-based
  - Use PB-to-CNF encodings to relax unsat cores
- Implicit Hitting Set
  - SAT-based
  - Use a closed-source MIP solver to guide search
- Branch-and-Bound
  - Solves MaxSAT “natively”

Different reasoning techniques!

CERTIFIED MAXSAT SOLVERS

Idea (Does not work):

- Utilize one of SAT's proof systems

CERTIFIED MAXSAT SOLVERS

Idea (Does not work):

- Utilize one of SAT's proof systems

  Inherently not able to reason about optimality

CERTIFIED MAXSAT SOLVERS

Idea (Does not work):

- Utilize one of SAT's proof systems
  Inherently not able to reason about optimality

Idea (Does not work):

- Obtain solution α with $\mathcal{O}(\alpha) = v^*$ for $(F, \mathcal{O})$ by running MaxSAT solver

CERTIFIED MAXSAT SOLVERS

Idea (Does not work):

- Utilize one of SAT's proof systems

  Inherently not able to reason about optimality

Idea (Does not work):

- Obtain solution α with $\mathcal{O}(\alpha) = v^*$ for $(F, \mathcal{O})$ by running MaxSAT solver
- Check solution to be satisfying assignment

CERTIFIED MAXSAT SOLVERS

Idea (Does not work):

- Utilize one of SAT's proof systems

  Inherently not able to reason about optimality

Idea (Does not work):

- Obtain solution α with $\mathcal{O}(\alpha) = v^*$ for $(F, \mathcal{O})$ by running MaxSAT solver
- Check solution to be satisfying assignment
- Create formula $F' = F \wedge \quad \mathcal{O} < v^*$

CERTIFIED MAXSAT SOLVERS

Idea (Does not work):

- Utilize one of SAT's proof systems
  Inherently not able to reason about optimality

Idea (Does not work):

- Obtain solution α with $\mathcal{O}(\alpha) = v^*$ for $(F, \mathcal{O})$ by running MaxSAT solver
- Check solution to be satisfying assignment
- Create formula $F' = F \wedge \quad \mathcal{O} < v^*$
- Run SAT solver with standard proof logging to obtain certificate of UNSAT for F'

CERTIFIED MAXSAT SOLVERS

Idea (Does not work):

- Utilize one of SAT's proof systems
  Inherently not able to reason about optimality

Idea (Does not work):

- Obtain solution α with $\mathcal{O}(\alpha) = v^*$ for $(F, \mathcal{O})$ by running MaxSAT solver
- Check solution to be satisfying assignment
  Easy to check!
- Create formula $F' = F \wedge \quad \mathcal{O} < v^*$
- Run SAT solver with standard proof logging to obtain certificate of UNSAT for F'

CERTIFIED MAXSAT SOLVERS

Idea (Does not work):

- Utilize one of SAT's proof systems
  Inherently not able to reason about optimality

Idea (Does not work):

- Obtain solution α with $\mathcal{O}(\alpha) = v^*$ for $(F, \mathcal{O})$ by running MaxSAT solver
- Check solution to be satisfying assignment
  Easy to check!
- Create formula $F' = F \wedge \mathrm{CNF}(\mathcal{O} < v^*)$
  Requires proof logging – Not possible with state-of-the-art proof systems for SAT
- Run SAT solver with standard proof logging to obtain certificate of UNSAT for F'

CERTIFIED MAXSAT SOLVERS

Idea (Does not work):

- Utilize one of SAT's proof systems
  Inherently not able to reason about optimality

Idea (Does not work):

- Obtain solution α with $\mathcal{O}(\alpha) = v^*$ for $(F, \mathcal{O})$ by running MaxSAT solver
- Check solution to be satisfying assignment
  Easy to check!
- Create formula $F' = F \wedge \mathrm{CNF}(\mathcal{O} < v^*)$
  Requires proof logging – Not possible with state-of-the-art proof systems for SAT
- Run SAT solver with standard proof logging to obtain certificate of UNSAT for F'
  Causes serious overhead

CERTIFIED MAXSAT SOLVERS

Idea (Does not work):

- Utilize one of SAT's proof systems
  Inherently not able to reason about optimality

Idea (Does not work):

- Obtain solution α with $\mathcal{O}(\alpha) = v^*$ for $(F, \mathcal{O})$ by running MaxSAT solver
- Check solution to be satisfying assignment
  Easy to check!
- Create formula $F' = F \wedge \mathrm{CNF}(\mathcal{O} < v^*)$
  Requires proof logging – Not possible with state-of-the-art proof systems for SAT
- Run SAT solver with standard proof logging to obtain certificate of UNSAT for F'
  Causes serious overhead

Only proves answer correct, not reasoning within solver!

CERTIFIED MAXSAT SOLVERS

Idea:

- Express the solver's reasoning in a more general proof system

CERTIFIED MAXSAT SOLVERS

Idea:

- Express the solver's reasoning in a more general proof system
  VeriPB!

CERTIFIED MAXSAT SOLVERS

Idea:

- Express the solver's reasoning in a more general proof system
  VeriPB!

A small and recent history of VeriPB MaxSAT proof logging:

CERTIFIED MAXSAT SOLVERS

Idea:

- Express the solver's reasoning in a more general proof system
  VeriPB!

A small and recent history of VeriPB MaxSAT proof logging:

- QMaxSAT: Solution Improving Search [Van23, VDB22]
  - Focus on certifying PB-to-CNF encodings

CERTIFIED MAXSAT SOLVERS

Idea:

- Express the solver's reasoning in a more general proof system
  VeriPB!

A small and recent history of VeriPB MaxSAT proof logging:

- QMaxSAT: Solution Improving Search [Van23, VDB22]
  - Focus on certifying PB-to-CNF encodings
- RC2 and CGSS: Core-Guided Search [BBN+23]
  - Including techniques such as stratification, hardening, intrinsic-at-most-ones constraints, ...

CERTIFIED MAXSAT SOLVERS

Idea:

- Express the solver's reasoning in a more general proof system
  VeriPB!

A small and recent history of VeriPB MaxSAT proof logging:

- QMaxSAT: Solution Improving Search [Van23, VDB22]
  - Focus on certifying PB-to-CNF encodings
- RC2 and CGSS: Core-Guided Search [BBN+23]
  - Including techniques such as stratification, hardening, intrinsic-at-most-ones constraints, ...
- Pacose (Coming Soon): Solution Improving Search Revisited
  - Challenge: Intricate without-loss-of-generality reasoning in the DPW encoding

CERTIFIED MAXSAT SOLVERS

Idea:

- Express the solver's reasoning in a more general proof system
  VeriPB!

A small and recent history of VeriPB MaxSAT proof logging:

- QMaxSAT: Solution Improving Search [Van23, VDB22]
  - Focus on certifying PB-to-CNF encodings
- RC2 and CGSS: Core-Guided Search [BBN+23]
  - Including techniques such as stratification, hardening, intrinsic-at-most-ones constraints, ...
- Pacose (Coming Soon): Solution Improving Search Revisited
  - Challenge: Intricate without-loss-of-generality reasoning in the DPW encoding

This talk:

- Branch-and-Bound

CERTIFIED MAXSAT SOLVERS

Idea:

- Express the solver's reasoning in a more general proof system
  VeriPB!

A small and recent history of VeriPB MaxSAT proof logging:

- QMaxSAT: Solution Improving Search [Van23, VDB22]
  - Focus on certifying PB-to-CNF encodings
- RC2 and CGSS: Core-Guided Search [BBN+23]
  - Including techniques such as stratification, hardening, intrinsic-at-most-ones constraints, ...
- Pacose (Coming Soon): Solution Improving Search Revisited
  - Challenge: Intricate without-loss-of-generality reasoning in the DPW encoding

This talk:

- Branch-and-Bound (and a little bit of Solution-Improving Search)

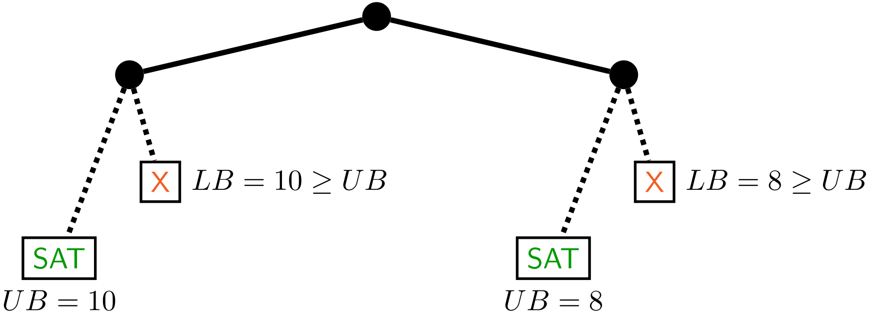
OUTLINE OF THIS PRESENTATION

- What is MaxSAT and how to certify it?
- Proof logging the B&B solver MaxCDCL
- Proof logging additional techniques in MaxCDCL
  - Hardening
  - Literal Unlocking
- Proof logging BDD PB-to-CNF encoding
- Future work & Conclusions

BRANCH AND BOUND

Branch and Bound:

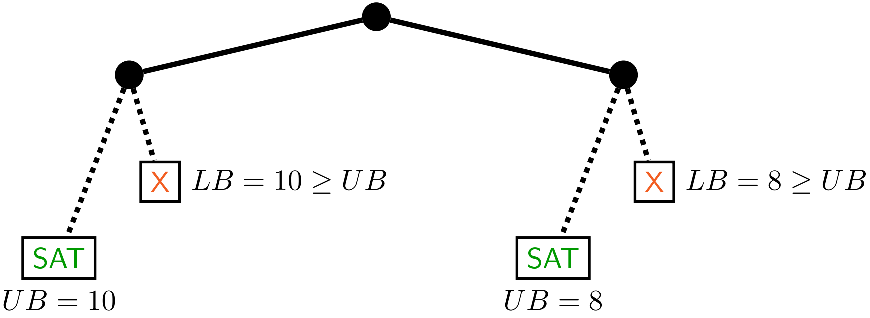
- Explore the search tree looking for optimal solutions
- Update Upper Bound UB when solution with better cost is found
- Underestimate LB of the cost at every node
- Prune branch when conflict found or when $LB \geq UB$



MAXCDCL AS BRANCH AND BOUND

Branch and Bound in MaxCDCL:

- Explore the search tree **(CDCL)** looking for satisfiable assignments
- Update Upper Bound UB when solution with better cost is found
- Underestimate LB of the cost at every node **using lookahead with UP**
- Prune branch when conflict found or when $LB \geq UB$ **and learn a clause**



MAXCDCL AS CDCL GENERALIZATION

MaxCDCL conflicts:

- **Hard conflict:**
  - A clause is falsified
- **Soft conflict:**
  - (underestimated) LB $\geq$ UB

MAXCDCL AS CDCL GENERALIZATION

MaxCDCL conflicts:

- **Hard conflict:**
  - A clause is falsified
- **Soft conflict:**
  - (underestimated) LB $\geq$ UB

In both cases: conflict analysis for learning new clause (CDCL)

LOOKAHEAD: LB UNDERESTIMATION (UNWEIGHTED CASE)

Lookahead with UP for underestimating LB:

1. Assume unassigned objective literals false and apply UP until:
   - A hard clause is falsified
   - Or a not yet assigned objective literal is assigned 1
2. We have found a **local** unsatisfiable core
3. Since unweighted case: Each **disjoint** core increases the LB by 1
4. When $LB \geq UB$, a soft conflict is found

SOFT CONFLICT DETECTION BY EXAMPLE (UNWEIGHTED CASE)

$\mathcal{O}^t = y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8$ $\quad \mathbf{UB} = 3$

Trail: $x_1^d \; \overline{x_2}^p \; x_3^p \; \overline{x_4}^d \; x_5^p \; x_6^p \; x_7^p$

SOFT CONFLICT DETECTION BY EXAMPLE (UNWEIGHTED CASE)

$\mathcal{O}^t = \cancel{y_1} + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8 \quad \textbf{UB} = 3$

Trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

Find one core:

$x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p\ \overline{y}_1^a\ x_9^p\ x_{10}^p$

SOFT CONFLICT DETECTION BY EXAMPLE (UNWEIGHTED CASE)

$\mathcal{O}^t = \cancel{y_1} + \cancel{y_2} + y_3 + y_4 + y_5 + y_6 + y_7 + y_8 \qquad \mathbf{UB} = 3$

Trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

Find one core:

$x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p\ \overline{y_1}^a\ x_9^p\ x_{10}^p\ \overline{y_2}^a\ \overline{x_{11}}^p$

SOFT CONFLICT DETECTION BY EXAMPLE (UNWEIGHTED CASE)

$\mathcal{O}^t = \cancel{y_1} + \cancel{y_2} + \cancel{y_3} + y_4 + y_5 + y_6 + y_7 + y_8 \quad \mathbf{UB} = 3$

Trail: $x_1^d \; \overline{x_2}^p \; x_3^p \; \overline{x_4}^d \; x_5^p \; x_6^p \; x_7^p$

Find one core:

$x_1^d \; \overline{x_2}^p \; x_3^p \; \overline{x_4}^d \; x_5^p \; x_6^p \; x_7^p \; \overline{y_1}^a \; x_9^p \; x_{10}^p \; \overline{y_2}^a \; \overline{x_{11}}^p \; \overline{y_3}^a$

SOFT CONFLICT DETECTION BY EXAMPLE (UNWEIGHTED CASE)

$\mathcal{O}^t = \cancel{y_1} + \cancel{y_2} + \cancel{y_3} + \cancel{y_4} + y_5 + y_6 + y_7 + y_8 \qquad \mathbf{UB} = 3$

Trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

Find one core:

$x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p\ \overline{y}_1^a\ x_9^p\ x_{10}^p\ \overline{y}_2^a\ \overline{x_{11}}^p\ \overline{y}_3^a\ \overline{y}_4^a\ x_{12}^p$ ($\overline{x_{12}} \vee x_{11} \in F$ falsified)

SOFT CONFLICT DETECTION BY EXAMPLE (UNWEIGHTED CASE)

$\mathcal{O}^t = \cancel{y_1} + \cancel{y_2} + \cancel{y_3} + \cancel{y_4} + y_5 + y_6 + y_7 + y_8 \qquad \mathbf{UB} = 3$

Trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

Find one core:

$x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p\ \overline{y_1}^a\ x_9^p\ x_{10}^p\ \overline{y_2}^a\ \overline{x_{11}}^p\ \overline{y_3}^a\ \overline{y_4}^a\ x_{12}^p$ ($\overline{x_{12}} \vee x_{11} \in F$ falsified)

$x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p\ \overline{y_1}^a\ \overline{y_2}^a\ \overline{y_3}^a\ \overline{y_4}^a$ (Assumptions suffice)

SOFT CONFLICT DETECTION BY EXAMPLE (UNWEIGHTED CASE)

$\mathcal{O}^t = \cancel{y_1} + y_2 + y_3 + \cancel{y_4} + y_5 + y_6 + y_7 + y_8 \qquad \mathbf{UB} = 3$

Trail: $x_1^d \; \overline{x_2}^p \; x_3^p \; \overline{x_4}^d \; x_5^p \; x_6^p \; x_7^p$

Find one core:

$x_1^d \; \overline{x_2}^p \; x_3^p \; \overline{x_4}^d \; x_5^p \; x_6^p \; x_7^p \; \overline{y_1}^a \; x_9^p \; x_{10}^p \; \overline{y_2}^a \; \overline{x_{11}}^p \; \overline{y_3}^a \; \overline{y_4}^a \; x_{12}^p$ ($\overline{x_{12}} \vee x_{11} \in F$ falsified)

$x_1^d \; \overline{x_2}^p \; x_3^p \; \overline{x_4}^d \; x_5^p \; x_6^p \; x_7^p \; \overline{y_1}^a \; \overline{y_2}^a \; \overline{y_3}^a \; \overline{y_4}^a$ (Assumptions suffice)

$\overline{x_2}^p \; \overline{x_4}^d \; \overline{y_1}^a \; \overline{y_4}^a$ (Conflict analysis)

SOFT CONFLICT DETECTION BY EXAMPLE (UNWEIGHTED CASE)

$\mathcal{O}^t = \cancel{y_1} + y_2 + y_3 + \cancel{y_4} + y_5 + y_6 + y_7 + y_8$ $\quad$ $\mathbf{UB} = 3$

Trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

Find one core:

$x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p\ \overline{y}_1^a\ x_9^p\ x_{10}^p\ \overline{y}_2^a\ \overline{x_{11}}^p\ \overline{y}_3^a\ \overline{y}_4^a\ x_{12}^p$ ($\overline{x_{12}} \vee x_{11} \in F$ falsified)

$x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p\ \overline{y}_1^a\ \overline{y}_2^a\ \overline{y}_3^a\ \overline{y}_4^a$ (Assumptions suffice)

$\overline{x_2}^p\ \overline{x_4}^d\ \overline{y}_1^a\ \overline{y}_4^a$ (Conflict analysis)

Local core:

$\overline{x_2} \wedge \overline{x_4} \wedge \overline{y}_1 \wedge \overline{y}_4 \rightarrow \square$

$\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_4$ (Reasons $\rightarrow$ Core)

SOFT CONFLICT DETECTION BY EXAMPLE (UNWEIGHTED CASE)

$\mathcal{O}^t = \cancel{y_1} + y_2 + y_3 + \cancel{y_4} + y_5 + y_6 + y_7 + y_8 \quad \mathbf{UB} = 3$

Trail: $x_1^d \; \overline{x_2}^p \; x_3^p \; \overline{x_4}^d \; x_5^p \; x_6^p \; x_7^p$

SOFT CONFLICT DETECTION BY EXAMPLE (UNWEIGHTED CASE)

$\mathcal{O}^t = \cancel{y_1} + \cancel{y_2} + y_3 + \cancel{y_4} + y_5 + y_6 + y_7 + y_8 \qquad \mathbf{UB} = 3$

Trail: $x_1^d \; \overline{x_2}^p \; x_3^p \; \overline{x_4}^d \; x_5^p \; x_6^p \; x_7^p$

Find next core:

$x_1^d \; \overline{x_2}^p \; x_3^p \; \overline{x_4}^d \; x_5^p \; x_6^p \; x_7^p \; \overline{y_2}^a$

SOFT CONFLICT DETECTION BY EXAMPLE (UNWEIGHTED CASE)

$\mathcal{O}^t = \cancel{y_1} + \cancel{y_2} + \cancel{y_3} + \cancel{y_4} + y_5 + y_6 + y_7 + y_8 \quad \mathbf{UB} = 3$

Trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

Find next core:

$x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p\ \overline{y}_2^a\ \overline{y}_3^a\ {y_5}^p$ (Propagate y_5 true)

SOFT CONFLICT DETECTION BY EXAMPLE (UNWEIGHTED CASE)

$\mathcal{O}^t = \cancel{y_1} + \cancel{y_2} + \cancel{y_3} + \cancel{y_4} + \cancel{y_5} + y_6 + y_7 + y_8 \quad \mathbf{UB} = 3$

Trail: $x_1^d \; \overline{x_2}^p \; x_3^p \; \overline{x_4}^d \; x_5^p \; x_6^p \; x_7^p$

Find next core:

$x_1^d \; \overline{x_2}^p \; x_3^p \; \overline{x_4}^d \; x_5^p \; x_6^p \; x_7^p \; \overline{y}_2^a \; \overline{y}_3^a \; {y_5}^p$ (Propagate y_5 true)

$\overline{x_2}^p \; x_7^p \; \overline{y}_2^a \; \overline{y}_3^a \; {y_5}^p$ (Conflict analysis)

SOFT CONFLICT DETECTION BY EXAMPLE (UNWEIGHTED CASE)

$\mathcal{O}^t = \cancel{y_1} + \cancel{y_2} + \cancel{y_3} + \cancel{y_4} + \cancel{y_5} + y_6 + y_7 + y_8 \qquad \mathbf{UB} = 3$

Trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

Find next core:

$x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p\ \overline{y}_2^a\ \overline{y}_3^a\ y_5{}^p$ (Propagate y_5 true)

$\overline{x_2}^p\ x_7^p\ \overline{y}_2^a\ \overline{y}_3^a\ y_5{}^p$ (Conflict analysis)

Local core:

$\overline{x_2} \wedge x_7 \wedge \overline{y}_2 \wedge \overline{y}_3 \wedge \overline{y}_5 \rightarrow \square$

$\overline{x_2} \wedge x_7 \rightarrow y_2 \vee y_3 \vee y_5$ (Reasons $\rightarrow$ Core)

SOFT CONFLICT DETECTION: FULL EXAMPLE (UNWEIGHTED CASE)

$\mathcal{O} = y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8 \qquad \mathbf{UB} = 3$

Trail: $x_1^d \; \overline{x_2}^p \; x_3^p \; \overline{x_4}^d \; x_5^p \; x_6^p \; x_7^p$

Found disjoint local cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_4$

Core 2: $\overline{x_2} \wedge x_7 \rightarrow y_2 \vee y_3 \vee y_5$

Core 3: $x_1 \wedge \overline{x_4} \wedge x_7 \rightarrow y_6 \vee y_7$

SOFT CONFLICT DETECTION: FULL EXAMPLE (UNWEIGHTED CASE)

$\mathcal{O} = y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8 \quad \mathbf{UB} = 3$

Trail: $x_1^d \; \overline{x_2}^p \; x_3^p \; \overline{x_4}^d \; x_5^p \; x_6^p \; x_7^p$

Found disjoint local cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_4$

Core 2: $\overline{x_2} \wedge x_7 \rightarrow y_2 \vee y_3 \vee y_5$

Core 3: $x_1 \wedge \overline{x_4} \wedge x_7 \rightarrow y_6 \vee y_7$

$$x_1 \wedge \overline{x_2} \wedge \overline{x_4} \wedge x_7 \rightarrow (y_1 \vee y_4) \wedge (y_2 \vee y_3 \vee y_5) \wedge (y_6 \vee y_7)$$

SOFT CONFLICT DETECTION: FULL EXAMPLE (UNWEIGHTED CASE)

$\mathcal{O} = y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8 \qquad \mathbf{UB} = 3$

Trail: $x_1^d \; \overline{x_2}^p \; x_3^p \; \overline{x_4}^d \; x_5^p \; x_6^p \; x_7^p$

Found disjoint local cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_4$

Core 2: $\overline{x_2} \wedge x_7 \rightarrow y_2 \vee y_3 \vee y_5$

Core 3: $x_1 \wedge \overline{x_4} \wedge x_7 \rightarrow y_6 \vee y_7$

$x_1 \wedge \overline{x_2} \wedge \overline{x_4} \wedge x_7 \rightarrow (y_1 \vee y_4) \wedge (y_2 \vee y_3 \vee y_5) \wedge (y_6 \vee y_7)$

$x_1 \wedge \overline{x_2} \wedge \overline{x_4} \wedge x_7 \rightarrow LB = 3 \geq 3 = UB$

SOFT CONFLICT DETECTION: FULL EXAMPLE (UNWEIGHTED CASE)

$\mathcal{O} = y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8 \qquad \mathbf{UB} = 3$

Trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

Found disjoint local cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_4$

Core 2: $\overline{x_2} \wedge x_7 \rightarrow y_2 \vee y_3 \vee y_5$

Core 3: $x_1 \wedge \overline{x_4} \wedge x_7 \rightarrow y_6 \vee y_7$

$x_1 \wedge \overline{x_2} \wedge \overline{x_4} \wedge x_7 \rightarrow (y_1 \vee y_4) \wedge (y_2 \vee y_3 \vee y_5) \wedge (y_6 \vee y_7)$

$x_1 \wedge \overline{x_2} \wedge \overline{x_4} \wedge x_7 \rightarrow LB = 3 \geq 3 = UB$

Soft conflict:

$x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p,\qquad$ Conflict $\overline{x_1} \vee x_2 \vee x_4 \vee \overline{x_7}$ **(soft conflict)**

SOFT CONFLICT DETECTION (WEIGHTED CASE)

Weighted MaxCDCL

- Weight of Local Core $\mathcal{K}$ = smallest weight of objective literals in $\mathcal{K}$
- Each objective literal can contribute to many cores
- The total contribution of a literal cannot exceed its weight

SOFT CONFLICT DETECTION (WEIGHTED CASE)

Weighted MaxCDCL

- Weight of Local Core $\mathcal{K}$ = smallest weight of objective literals in $\mathcal{K}$
- Each objective literal can contribute to many cores
- The total contribution of a literal cannot exceed its weight

$\mathcal{O}^t = 7y_1 + 2y_2 + 1y_3 + 1y_4 + 1y_5 + 4y_6 + 1y_7 + 3y_8 \quad \mathbf{UB} = 4$

Trail: $x_1^d \; \overline{x_2}^p \; x_3^p \; \overline{x_4}^d \; x_5^p \; x_6^p \; x_7^p$

Found local cores

SOFT CONFLICT DETECTION (WEIGHTED CASE)

Weighted MaxCDCL

- Weight of Local Core $\mathcal{K}$ = smallest weight of objective literals in $\mathcal{K}$
- Each objective literal can contribute to many cores
- The total contribution of a literal cannot exceed its weight

$\mathcal{O}^t = 7y_1 + 2y_2 + 1y_3 + 1y_4 + 1y_5 + 4y_6 + 1y_7 + 3y_8 \quad \mathbf{UB} = 4$

Trail: $x_1^d \; \overline{x_2}^p \; x_3^p \; \overline{x_4}^d \; x_5^p \; x_6^p \; x_7^p$

Found local cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (weight 2)

SOFT CONFLICT DETECTION (WEIGHTED CASE)

Weighted MaxCDCL

- Weight of Local Core $\mathcal{K}$ = smallest weight of objective literals in $\mathcal{K}$
- Each objective literal can contribute to many cores
- The total contribution of a literal cannot exceed its weight

$\mathcal{O}^t = \not{7}\ 5y_1 + \not{2}\ 0y_2 + 1y_3 + 1y_4 + 1y_5 + 4y_6 + 1y_7 + 3y_8 \quad \mathbf{UB} = 4$

Trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

Found local cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (weight 2)

SOFT CONFLICT DETECTION (WEIGHTED CASE)

Weighted MaxCDCL

- Weight of Local Core $\mathcal{K}$ = smallest weight of objective literals in $\mathcal{K}$
- Each objective literal can contribute to many cores
- The total contribution of a literal cannot exceed its weight

$\mathcal{O}^t = \cancel{7}\ 5y_1 + \cancel{2}\ 0y_2 + 1y_3 + 1y_4 + 1y_5 + 4y_6 + 1y_7 + 3y_8$ $\quad \mathbf{UB} = 4$

Trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

Found local cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (weight 2)

Core 2: $x_3 \wedge \overline{x_4} \rightarrow y_1 \vee y_5$ (weight 1)

SOFT CONFLICT DETECTION (WEIGHTED CASE)

Weighted MaxCDCL

- Weight of Local Core $\mathcal{K}$ = smallest weight of objective literals in $\mathcal{K}$
- Each objective literal can contribute to many cores
- The total contribution of a literal cannot exceed its weight

$\mathcal{O}^t = \not{7}\ \not{5}\ 4y_1 + \not{2}\ 0y_2 + 1y_3 + 1y_4 + \not{1}\ 0y_5 + 4y_6 + 1y_7 + 3y_8 \quad \mathbf{UB} = 4$

Trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

Found local cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (weight 2)

Core 2: $x_3 \wedge \overline{x_4} \rightarrow y_1 \vee y_5$ (weight 1)

SOFT CONFLICT DETECTION (WEIGHTED CASE)

Weighted MaxCDCL

- Weight of Local Core $\mathcal{K}$ = smallest weight of objective literals in $\mathcal{K}$
- Each objective literal can contribute to many cores
- The total contribution of a literal cannot exceed its weight

$\mathcal{O}^t = \not{7}\ \not{5}\ \not{4}\ 1y_1 + \not{2}\ 0y_2 + 1y_3 + 1y_4 + \not{1}\ 0y_5 + \not{4}\ 1y_6 + 1y_7 + \not{3}\ 0y_8 \quad \mathbf{UB} = 4$

Trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

Found local cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (weight 2)
Core 2: $x_3 \wedge \overline{x_4} \rightarrow y_1 \vee y_5$ (weight 1)
Core 3: $x_1 \rightarrow y_1 \vee y_6 \vee y_8$ (weight 3)

SOFT CONFLICT DETECTION (WEIGHTED CASE)

Weighted MaxCDCL

- Weight of Local Core $\mathcal{K}$ = smallest weight of objective literals in $\mathcal{K}$
- Each objective literal can contribute to many cores
- The total contribution of a literal cannot exceed its weight

$\mathcal{O}^t = \not7\ \not5\ \not4\ 1\not7\ \not5\ 2y_1 + \not2\ 0y_2 + 1y_3 + 1y_4 + 1y_5 + \not4\ 1y_6 + 1y_7 + \not3\ 0y_8 \quad \mathbf{UB} = 4$

Trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

Found local cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (weight 2)

~~Core 2: $x_3 \wedge \overline{x_4} \rightarrow y_1 \vee y_5$ (weight 1)~~

Core 3: $x_1 \rightarrow y_1 \vee y_6 \vee y_8$ (weight 3)

SOFT CONFLICT DETECTION (WEIGHTED CASE)

Weighted MaxCDCL

- Weight of Local Core $\mathcal{K}$ = smallest weight of objective literals in $\mathcal{K}$
- Each objective literal can contribute to many cores
- The total contribution of a literal cannot exceed its weight

$\mathcal{O}^t = \not7\ \not5\ \not4\ 1\not7\ \not5\ 2y_1 + \not2\ 0y_2 + 1y_3 + 1y_4 + 1y_5 + \not4\ 1y_6 + 1y_7 + \not3\ 0y_8 \quad \mathbf{UB} = 4$

Trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

Found local cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (weight 2)

Core 3: $x_1 \rightarrow y_1 \vee y_6 \vee y_8$ (weight 3)

SOFT CONFLICT DETECTION (WEIGHTED CASE)

Weighted MaxCDCL

- Weight of Local Core $\mathcal{K}$ = smallest weight of objective literals in $\mathcal{K}$
- Each objective literal can contribute to many cores
- The total contribution of a literal cannot exceed its weight

$$\mathcal{O}^t = \cancel{7}\ \cancel{5}\ \cancel{4}\ 1\cancel{7}\ \cancel{5}\ 2y_1 + \cancel{2}\ 0y_2 + 1y_3 + 1y_4 + 1y_5 + \cancel{4}\ 1y_6 + 1y_7 + \cancel{3}\ 0y_8 \qquad \mathbf{UB} = 4$$

Trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

Found local cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (weight 2)

Core 3: $x_1 \rightarrow y_1 \vee y_6 \vee y_8$ (weight 3)

Conclusion: $x_1 \wedge \overline{x_2} \wedge \overline{x_4} \rightarrow LB = 5 \geq 4 = UB$ **Soft Conflict clause**: $\overline{x}_1 \vee x_2 \vee x_4$

PROOF LOGGING SOFT CONFLICTS

To Derive: $\overline{x}_1 + x_2 + x_4 \geq 1 \quad \mathbf{UB} = 4$

PROOF LOGGING SOFT CONFLICTS

To Derive: $\overline{x}_1 + x_2 + x_4 \geq 1 \quad \mathbf{UB} = 4$

Found "disjoint" cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (2)

Core 2: $x_1 \rightarrow y_1 \vee y_6 \vee y_8$ (3)

PROOF LOGGING SOFT CONFLICTS

To Derive: $\overline{x}_1 + x_2 + x_4 \geq 1$ $\quad$ $\mathbf{UB} = 4$

Found “disjoint” cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (2)

PB: $x_2 + x_4 + y_1 + y_2 \geq 1$

Core 2: $x_1 \rightarrow y_1 \vee y_6 \vee y_8$ (3)

PB: $\overline{x}_1 + y_1 + y_6 + y_8 \geq 1$

PROOF LOGGING SOFT CONFLICTS

To Derive: $\overline{x}_1 + x_2 + x_4 \geq 1 \quad \mathbf{UB} = 4$

Found "disjoint" cores (RUP)

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (2)

PB: $x_2 + x_4 + y_1 + y_2 \geq 1$

Core 2: $x_1 \rightarrow y_1 \vee y_6 \vee y_8$ (3)

PB: $\overline{x}_1 + y_1 + y_6 + y_8 \geq 1$

PROOF LOGGING SOFT CONFLICTS

To Derive: $\overline{x}_1 + x_2 + x_4 \geq 1$ $\quad \mathbf{UB} = 4$

Found “disjoint” cores (RUP)

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (2)

PB: $2x_2 + 2x_4 + 2y_1 + 2y_2 \geq 2$ ~~1~~

Core 2: $x_1 \rightarrow y_1 \vee y_6 \vee y_8$ (3)

PB: $3\overline{x}_1 + 3y_1 + 3y_6 + 3y_8 \geq 3$ ~~1~~

Multiplication by their weight

PROOF LOGGING SOFT CONFLICTS

To Derive: $\overline{x}_1 + x_2 + x_4 \geq 1$ $\quad \mathbf{UB} = 4$

Found "disjoint" cores (RUP)

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (2)

PB: $2x_2 + 2x_4 + 2y_1 + 2y_2 \geq 2$ ~~1~~

Core 2: $x_1 \rightarrow y_1 \vee y_6 \vee y_8$ (3)

PB: $3\overline{x}_1 + 3y_1 + 3y_6 + 3y_8 \geq 3$ ~~1~~

Multiplication by their weight and addition:

$3\overline{x}_1 + 2x_2 + 2x_4 + 5y_1 + 2y_2 + 3y_6 + 3y_8 \geq 5$

PROOF LOGGING SOFT CONFLICTS

To Derive: $\overline{x}_1 + x_2 + x_4 \geq 1 \quad \mathbf{UB} = 4$

Found "disjoint" cores (RUP)

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (2)

PB: $2x_2 + 2x_4 + 2y_1 + 2y_2 \geq 2 \not{1}$

Core 2: $x_1 \rightarrow y_1 \vee y_6 \vee y_8$ (3)

PB: $3\overline{x}_1 + 3y_1 + 3y_6 + 3y_8 \geq 3 \not{1}$

Multiplication by their weight and addition:

$3\overline{x}_1 + 2x_2 + 2x_4 + 5y_1 + 2y_2 + 3y_6 + 3y_8 \geq 5$

Model improving constraint

$7y_1 + 2y_2 + 1y_3 + 1y_4 + 1y_5 + 4y_6 + 1y_7 + 3y_8 \leq 3$

PROOF LOGGING SOFT CONFLICTS

To Derive: $\overline{x}_1 + x_2 + x_4 \geq 1 \quad \mathbf{UB} = 4$

Found "disjoint" cores (RUP)

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (2)

PB: $2x_2 + 2x_4 + 2y_1 + 2y_2 \geq 2 \not{1}$

Core 2: $x_1 \rightarrow y_1 \vee y_6 \vee y_8$ (3)

PB: $3\overline{x}_1 + 3y_1 + 3y_6 + 3y_8 \geq 3 \not{1}$

Multiplication by their weight and addition:

$3\overline{x}_1+2x_2+2x_4+5y_1+2y_2+3y_6+3y_8 \geq 5$

Model improving constraint

$7y_1 + 2y_2 + 1y_3 + 1y_4 + 1y_5 + 4y_6 + 1y_7 + 3y_8 \leq 3$

In normalized form:

$7\overline{y}_1+2\overline{y}_2+1\overline{y}_3+1\overline{y}_4+1\overline{y}_5+4\overline{y}_6+1\overline{y}_7+3\overline{y}_8 \geq 20-3$

PROOF LOGGING SOFT CONFLICTS

To Derive: $\overline{x}_1 + x_2 + x_4 \geq 1$ $\quad \mathbf{UB} = 4$

Found "disjoint" cores (RUP)

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (2)

PB: $2x_2 + 2x_4 + 2y_1 + 2y_2 \geq 2$ ~~1~~

Core 2: $x_1 \rightarrow y_1 \vee y_6 \vee y_8$ (3)

PB: $3\overline{x}_1 + 3y_1 + 3y_6 + 3y_8 \geq 3$ ~~1~~

Multiplication by their weight and addition:

$3\overline{x}_1 + 2x_2 + 2x_4 + 5y_1 + 2y_2 + 3y_6 + 3y_8 \geq 5$

Model improving constraint

$7y_1 + 2y_2 + 1y_3 + 1y_4 + 1y_5 + 4y_6 + 1y_7 + 3y_8 \leq 3$

In normalized form:

$7\overline{y}_1 + 2\overline{y}_2 + 1\overline{y}_3 + 1\overline{y}_4 + 1\overline{y}_5 + 4\overline{y}_6 + 1\overline{y}_7 + 3\overline{y}_8 \geq 20 - 3$

Weakening:

$5\overline{y}_1 + 2\overline{y}_2 + 3\overline{y}_6 + 3\overline{y}_8 \geq 13 - 3$

PROOF LOGGING SOFT CONFLICTS

To Derive: $\overline{x}_1 + x_2 + x_4 \geq 1$ $\quad \mathbf{UB} = 4$

Found "disjoint" cores (RUP)

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (2)

PB: $2x_2 + 2x_4 + 2y_1 + 2y_2 \geq 2$ ~~1~~

Core 2: $x_1 \rightarrow y_1 \vee y_6 \vee y_8$ (3)

PB: $3\overline{x}_1 + 3y_1 + 3y_6 + 3y_8 \geq 3$ ~~1~~

Multiplication by their weight and addition:

$$3\overline{x}_1 + 2x_2 + 2x_4 + 5y_1 + 2y_2 + 3y_6 + 3y_8 \geq 5$$

Model improving constraint

$$7y_1 + 2y_2 + 1y_3 + 1y_4 + 1y_5 + 4y_6 + 1y_7 + 3y_8 \leq 3$$

In normalized form:

$$7\overline{y}_1 + 2\overline{y}_2 + 1\overline{y}_3 + 1\overline{y}_4 + 1\overline{y}_5 + 4\overline{y}_6 + 1\overline{y}_7 + 3\overline{y}_8 \geq 20 - 3$$

Weakening:

$$5\overline{y}_1 + 2\overline{y}_2 + 3\overline{y}_6 + 3\overline{y}_8 \geq 13 - 3$$

Addition:

$$3\overline{x}_1 + 2x_2 + 2x_4 + 5y_1 + 5\overline{y}_1 + 2y_2 + 2\overline{y}_2 + 3y_6 + 3\overline{y}_6 + 3y_8 + 3\overline{y}_8 \geq 13 + 5 - 3$$

PROOF LOGGING SOFT CONFLICTS

To Derive: $\overline{x}_1 + x_2 + x_4 \geq 1$ $\quad$ $\mathbf{UB} = 4$

Found "disjoint" cores (RUP)

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (2)

PB: $2x_2 + 2x_4 + 2y_1 + 2y_2 \geq 2\ \not{1}$

Core 2: $x_1 \rightarrow y_1 \vee y_6 \vee y_8$ (3)

PB: $3\overline{x}_1 + 3y_1 + 3y_6 + 3y_8 \geq 3\ \not{1}$

Multiplication by their weight and addition:

$3\overline{x}_1 + 2x_2 + 2x_4 + 5y_1 + 2y_2 + 3y_6 + 3y_8 \geq 5$

Model improving constraint

$7y_1 + 2y_2 + 1y_3 + 1y_4 + 1y_5 + 4y_6 + 1y_7 + 3y_8 \leq 3$

In normalized form:

$7\overline{y}_1 + 2\overline{y}_2 + 1\overline{y}_3 + 1\overline{y}_4 + 1\overline{y}_5 + 4\overline{y}_6 + 1\overline{y}_7 + 3\overline{y}_8 \geq 20 - 3$

Weakening:

$5\overline{y}_1 + 2\overline{y}_2 + 3\overline{y}_6 + 3\overline{y}_8 \geq 13 - 3$

Addition:

$3\overline{x}_1 + 2x_2 + 2x_4 + \cancel{5y_1 + 5\overline{y}_1 + 2y_2 + 2\overline{y}_2 + 3y_6 + 3\overline{y}_6 + 3y_8 + 3\overline{y}_8} \geq \cancel{13} + 5 - 3$

PROOF LOGGING SOFT CONFLICTS

To Derive: $\overline{x}_1 + x_2 + x_4 \geq 1 \quad \mathbf{UB} = 4$

Found "disjoint" cores (RUP)

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (2)

PB: $2x_2 + 2x_4 + 2y_1 + 2y_2 \geq 2\ \cancel{1}$

Core 2: $x_1 \rightarrow y_1 \vee y_6 \vee y_8$ (3)

PB: $3\overline{x}_1 + 3y_1 + 3y_6 + 3y_8 \geq 3\ \cancel{1}$

Multiplication by their weight and addition:

$3\overline{x}_1 + 2x_2 + 2x_4 + 5y_1 + 2y_2 + 3y_6 + 3y_8 \geq 5$

Model improving constraint

$7y_1 + 2y_2 + 1y_3 + 1y_4 + 1y_5 + 4y_6 + 1y_7 + 3y_8 \leq 3$

In normalized form:

$7\overline{y}_1 + 2\overline{y}_2 + 1\overline{y}_3 + 1\overline{y}_4 + 1\overline{y}_5 + 4\overline{y}_6 + 1\overline{y}_7 + 3\overline{y}_8 \geq 20 - 3$

Weakening:

$5\overline{y}_1 + 2\overline{y}_2 + 3\overline{y}_6 + 3\overline{y}_8 \geq 13 - 3$

Addition:

$3\overline{x}_1 + 2x_2 + 2x_4 + \cancel{5y_1 + 5\overline{y}_1 + 2y_2 + 2\overline{y}_2 + 3y_6 + 3\overline{y}_6 + 3y_8 + 3\overline{y}_8} \geq \cancel{13} + 5 - 3$

Division by $5 - 3$ and Saturation: $\overline{x}_1 + x_2 + x_4 \geq 1$

OUTLINE OF THIS PRESENTATION

- What is MaxSAT and how to certify it?
- Proof logging the B&B solver MaxCDCL
- Proof logging additional techniques in MaxCDCL
  - Hardening
  - Literal Unlocking
- Proof logging BDD PB-to-CNF encoding
- Future work & Conclusions

OUTLINE OF THIS PRESENTATION

- What is MaxSAT and how to certify it?
- Proof logging the B&B solver MaxCDCL
- Proof logging additional techniques in MaxCDCL
  - Hardening
  - Literal Unlocking
- Proof logging BDD PB-to-CNF encoding
- Future work & Conclusions

HARDENING

$$\mathcal{O}^t = \not{7}\ \not{5}\ 4y_1 + \not{2}\ 0y_2 + 1y_3 + 1y_4 + \not{1}\ 0y_5 + \not{4}\ 3y_6 + 1y_7 + 3y_8 \qquad \mathbf{UB} = 5$$

Found local cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (weight 2)

Core 2: $x_3 \wedge \overline{x_4} \rightarrow y_1 \vee y_5 \vee y_6$ (weight 1)

HARDENING

$\mathcal{O}^t = \not{7}\ \not{5}\ 4y_1 + \not{2}\ 0y_2 + 1y_3 + 1y_4 + \not{1}\ 0y_5 + \not{4}\ 3y_6 + 1y_7 + 3y_8 \quad \mathbf{UB} = 5$

Found local cores

Core 1: $\overline{x_2} \land \overline{x_4} \rightarrow y_1 \lor y_2$ (weight 2)

Core 2: $x_3 \land \overline{x_4} \rightarrow y_1 \lor y_5 \lor y_6$ (weight 1)

Conclusion: $\overline{x_2} \land x_3 \land \overline{x_4} \land y_6 \rightarrow LB = 6 \geq 5 = UB$

HARDENING

$\mathcal{O}^t = \not{7}\ \not{5}\ 4y_1 + \not{2}\ 0y_2 + 1y_3 + 1y_4 + \not{1}\ 0y_5 + \not{4}\ 3y_6 + 1y_7 + 3y_8 \qquad \mathbf{UB} = 5$

Found local cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \to y_1 \vee y_2$ (weight 2)

Core 2: $x_3 \wedge \overline{x_4} \to y_1 \vee y_5 \vee y_6$ (weight 1)

Conclusion: $\overline{x_2} \wedge x_3 \wedge \overline{x_4} \wedge y_6 \to LB = 6 \geq 5 = UB$

$$\overline{x_2} \wedge x_3 \wedge \overline{x_4} \to \overline{y_6}$$

HARDENING

$$\mathcal{O}^t = \not{7}\ \not{5}\ 4y_1 + \not{2}\ 0y_2 + 1y_3 + 1y_4 + \not{1}\ 0y_5 + \not{4}\ 3y_6 + 1y_7 + 3y_8 \qquad \textbf{UB} = 5$$

Found local cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (weight 2)

Core 2: $x_3 \wedge \overline{x_4} \rightarrow y_1 \vee y_5 \vee y_6$ (weight 1)

Conclusion: $\overline{x_2} \wedge x_3 \wedge \overline{x_4} \wedge y_6 \rightarrow LB = 6 \geq 5 = UB$

$$\overline{x_2} \wedge x_3 \wedge \overline{x_4} \rightarrow \overline{y_6}$$

Clauses Learned: $x_2 \vee \overline{x_3} \vee x_4 \vee \overline{y_i} \quad (i \in \{1, 6, 8\})$

PROOF LOGGING HARDENING

To Derive: $x_2 \vee \overline{x_3} \vee x_4 \vee y_1$

Found “disjoint” cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (weight 2)

PB: $x_2 + x_4 + y_1 + y_2 \geq 1$

Core 2: $x_3 \wedge \overline{x_4} \rightarrow y_1 \vee y_5 \vee y_6$ (weight 1)

PB: $\overline{x}_3 + x_4 + y_1 + y_6 + y_8 \geq 1$

PROOF LOGGING HARDENING

To Derive: $x_2 \lor \overline{x_3} \lor x_4 \lor y_1$

Found "disjoint" cores

Core 1: $\overline{x_2} \land \overline{x_4} \rightarrow y_1 \lor y_2$ (weight 2)

PB: $x_2 + x_4 + y_1 + y_2 \geq 1$

Core 2: $x_3 \land \overline{x_4} \rightarrow y_1 \lor y_5 \lor y_6$ (weight 1)

PB: $\overline{x}_3 + x_4 + y_1 + y_6 + y_8 \geq 1$

Model improving constraint

$$7y_1 + 2y_2 + 1y_3 + 1y_4 + 1y_5 + 4y_6 + 1y_7 + 3y_8 \leq 3$$

In normalized form:

$$7\overline{y}_1 + 2\overline{y}_2 + 1\overline{y}_3 + 1\overline{y}_4 + 1\overline{y}_5 + 4\overline{y}_6 + 1\overline{y}_7 + 3\overline{y}_8 \geq 20 - 3$$

PROOF LOGGING HARDENING

To Derive: $x_2 \vee \overline{x_3} \vee x_4 \vee y_1$

Found "disjoint" cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (weight 2)

PB: $2x_2 + 2x_4 + 2y_1 + 2y_2 \geq 2 \not{1}$

Core 2: $x_3 \wedge \overline{x_4} \rightarrow y_1 \vee y_5 \vee y_6$ (weight 1)

PB: $1\overline{x}_3 + 1x_4 + 1y_1 + 1y_6 + 1y_8 \geq 1$

Model improving constraint

$7y_1 + 2y_2 + 1y_3 + 1y_4 + 1y_5 + 4y_6 + 1y_7 + 3y_8 \leq 3$

In normalized form:

$7\overline{y}_1 + 2\overline{y}_2 + 1\overline{y}_3 + 1\overline{y}_4 + 1\overline{y}_5 + 4\overline{y}_6 + 1\overline{y}_7 + 3\overline{y}_8 \geq 20 - 3$

Multiplying cores by their weight and addition with Model-Improving Constraint:

$2x_2 + 1\overline{x}_3 + 3x_4 + 4\overline{y}_1 + 1\overline{y}_3 + 1\overline{y}_4 + 1\overline{y}_5 + 3\overline{y}_6 + 1\overline{y}_7 + 2\overline{y}_8 \geq 13 + 3 - 3$

PROOF LOGGING HARDENING

To Derive: $x_2 \lor \overline{x_3} \lor x_4 \lor y_1$

Found "disjoint" cores

Core 1: $\overline{x_2} \land \overline{x_4} \rightarrow y_1 \lor y_2$ (weight 2)

PB: $2x_2 + 2x_4 + 2y_1 + 2y_2 \geq 2\ \not{1}$

Core 2: $x_3 \land \overline{x_4} \rightarrow y_1 \lor y_5 \lor y_6$ (weight 1)

PB: $1\overline{x}_3 + 1x_4 + 1y_1 + 1y_6 + 1y_8 \geq 1$

Model improving constraint

$7y_1 + 2y_2 + 1y_3 + 1y_4 + 1y_5 + 4y_6 + 1y_7 + 3y_8 \leq 3$

In normalized form:

$7\overline{y}_1 + 2\overline{y}_2 + 1\overline{y}_3 + 1\overline{y}_4 + 1\overline{y}_5 + 4\overline{y}_6 + 1\overline{y}_7 + 3\overline{y}_8 \geq 20 - 3$

Multiplying cores by their weight and addition with Model-Improving Constraint:

$2x_2 + 1\overline{x}_3 + 3x_4 + 4\overline{y}_1 + 1\overline{y}_3 + 1\overline{y}_4 + 1\overline{y}_5 + 3\overline{y}_6 + 1\overline{y}_7 + 2\overline{y}_8 \geq 13 + 3 - 3$

Weakening all y_i with $i \in \{1, 3, 4, 5, 7, 8\}$:

$2x_2 + 1\overline{x}_3 + 3x_4 + 3\overline{y}_6 \geq 3$

PROOF LOGGING HARDENING

To Derive: $x_2 \vee \overline{x_3} \vee x_4 \vee y_1$

Found "disjoint" cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_1 \vee y_2$ (weight 2)

PB: $2x_2 + 2x_4 + 2y_1 + 2y_2 \geq 2\ \not{1}$

Core 2: $x_3 \wedge \overline{x_4} \rightarrow y_1 \vee y_5 \vee y_6$ (weight 1)

PB: $1\overline{x}_3 + 1x_4 + 1y_1 + 1y_6 + 1y_8 \geq 1$

Model improving constraint

$7y_1 + 2y_2 + 1y_3 + 1y_4 + 1y_5 + 4y_6 + 1y_7 + 3y_8 \leq 3$

In normalized form:

$7\overline{y}_1 + 2\overline{y}_2 + 1\overline{y}_3 + 1\overline{y}_4 + 1\overline{y}_5 + 4\overline{y}_6 + 1\overline{y}_7 + 3\overline{y}_8 \geq 20 - 3$

Multiplying cores by their weight and addition with Model-Improving Constraint:

$2x_2 + 1\overline{x}_3 + 3x_4 + 4\overline{y}_1 + 1\overline{y}_3 + 1\overline{y}_4 + 1\overline{y}_5 + 3\overline{y}_6 + 1\overline{y}_7 + 2\overline{y}_8 \geq 13 + 3 - 3$

Weakening all y_i with $i \in \{1, 3, 4, 5, 7, 8\}$:

$2x_2 + 1\overline{x}_3 + 3x_4 + 3\overline{y}_6 \geq 3$

Division by 3 and saturation: $x_2 + \overline{x}_3 + x_4 + \overline{y}_6 \geq 1$

OUTLINE OF THIS PRESENTATION

- What is MaxSAT and how to certify it?
- Proof logging the B&B solver MaxCDCL
- Proof logging additional techniques in MaxCDCL
  - Hardening
  - Literal Unlocking
- Proof logging BDD PB-to-CNF encoding
- Future work & Conclusions

UNWEIGHTED MAXCDCL REVISITED

Unweighted MaxCDCL searches for set $\mathcal{L}$ of tuples (b, L) such that

1. Each L is a set of objective literals
2. For each (b, L) in $\mathcal{L}$, it holds that $F \wedge \alpha \models \sum_{\ell \in L} \ell \geq b$.
3. For each pair (b, L) and (b', L') in $\mathcal{L}$, $L \cap L' = \emptyset$.
4. The total weight exceeds the current upper bound: $\sum_{(b,L) \in \mathcal{L}} b \geq \mathbf{UB}$.

UNWEIGHTED MAXCDCL REVISITED

Unweighted MaxCDCL searches for set $\mathcal{L}$ of tuples (b, L) such that

1. Each L is a set of objective literals
2. For each (b, L) in $\mathcal{L}$, it holds that $F \wedge \alpha \models \sum_{\ell \in L} \ell \geq b$.
3. For each pair (b, L) and (b', L') in $\mathcal{L}$, $L \cap L' = \emptyset$.
4. The total weight exceeds the current upper bound: $\sum_{(b,L) \in \mathcal{L}} b \geq \mathbf{UB}$.

$\mathcal{O} = y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8 + y_9 + \ldots \quad \mathbf{UB} = 4$

Found disjoint local "cores"

Core 1: $\overline{x}_2 \wedge \overline{x}_4 \rightarrow y_1 + y_3 + y_5 + y_8 \geq 3$

Core 2: $x_4 \wedge \overline{x}_7 \wedge x_9 \rightarrow y_2 + y_4 + y_6 \geq 2$

UNWEIGHTED MAXCDCL REVISITED

Unweighted MaxCDCL searches for set $\mathcal{L}$ of tuples (b, L) such that

1. Each L is a set of objective literals
2. For each (b, L) in $\mathcal{L}$, it holds that $F \wedge \alpha \models \sum_{\ell \in L} \ell \geq b$.
3. For each pair (b, L) and (b', L') in $\mathcal{L}$, $L \cap L' = \emptyset$.
4. The total weight exceeds the current upper bound: $\sum_{(b,L) \in \mathcal{L}} b \geq \mathbf{UB}$.

$\mathcal{O} = y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8 + y_9 + \ldots \qquad \mathbf{UB} = 4$

Found disjoint local "cores"

Core 1: $\overline{x}_2 \wedge \overline{x}_4 \rightarrow y_1 + y_3 + y_5 + y_8 \geq 3$

Core 2: $x_4 \wedge \overline{x}_7 \wedge x_9 \rightarrow y_2 + y_4 + y_6 \geq 2$

$\overline{x}_2 \wedge \overline{x}_4 \wedge \overline{x}_7 \wedge x_9 \rightarrow LB = 5 \geq 4 = UB$ **Soft conflict clause:** $x_2 \vee x_4 \vee x_7 \vee \overline{x}_9$

LOOKAHEAD WITH LITERAL UNLOCKING (BY EXAMPLE)

$\mathcal{O}^t = \cancel{y_1} + \cancel{y_2} + \cancel{y_3} + \cancel{y_4} + \cancel{y_5} + \cancel{y_6} + \cancel{y_7} + \cancel{y_8} + y_9 + \ldots$

Trail: $x_1^d \; \overline{x_2}^d \; x_3^p \; \overline{x_4}^d \; x_5^p$

Found disjoint local "cores"

Core 1: $\overline{x_2} \land \overline{x_4} \rightarrow y_3 + y_5 + y_6 \geq 1$

Core 2: $x_1 \land \overline{x_2} \rightarrow y_1 + y_2 + y_4 + y_7 + y_8 \geq 2$

LOOKAHEAD WITH LITERAL UNLOCKING (BY EXAMPLE)

$\mathcal{O}^t = \not{y_1} + \not{y_2} + \not{y_3} + \not{y_4} + \not{y_5} + \not{y_6} + \not{y_7} + \not{y_8} + y_9 + \ldots$

Trail: $x_1^d\ \overline{x_2}^d\ x_3^p\ \overline{x_4}^d\ x_5^p\ \overline{y_9}^a\ y_1^p\ y_3^p$

Found disjoint local “cores”

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_3 + y_5 + y_6 \geq 1$

Core 2: $x_1 \wedge \overline{x_2} \rightarrow y_1 + y_2 + y_4 + y_7 + y_8 \geq 2$

LOOKAHEAD WITH LITERAL UNLOCKING (BY EXAMPLE)

$\mathcal{O}^t = \cancel{y_1} + \cancel{y_2} + \cancel{y_3} + \cancel{y_4} + y_5 + y_6 + \cancel{y_7} + \cancel{y_8} + y_9 + \ldots$

Trail: $x_1^d\ \overline{x_2}^d\ x_3^p\ \overline{x_4}^d\ x_5^p\ \overline{y_9}^a\ y_1^p\ y_3^p$

Found disjoint local "cores"

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_3 + y_5 + y_6 \geq 1$ "$\{y_9\}$ unlocks Core 1 on $\{y_3\}$"

Core 2: $x_1 \wedge \overline{x_2} \rightarrow y_1 + y_2 + y_4 + y_7 + y_8 \geq 2$

LOOKAHEAD WITH LITERAL UNLOCKING (BY EXAMPLE)

$\mathcal{O}^t = \not{y_1} + \not{y_2} + \not{y_3} + \not{y_4} + \not{y_5} + \not{y_6} + \not{y_7} + \not{y_8} + y_9 + \ldots$

Trail: $x_1^d\ \overline{x_2}^d\ x_3^p\ \overline{x_4}^d\ x_5^p\ \overline{y_9}^a\ y_1^p\ y_3^p\ \overline{y_5}^a\ \overline{y_6}^a\ y_7^p$

Found disjoint local “cores”

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_3 + y_5 + y_6 \geq 1$ “$\{y_9\}$ unlocks Core 1 on $\{y_3\}$”

Core 2: $x_1 \wedge \overline{x_2} \rightarrow y_1 + y_2 + y_4 + y_7 + y_8 \geq 2$ “$\{y_9, y_5, y_6\}$ unlocks Core 2 on $\{y_1, y_7\}$”

LOOKAHEAD WITH LITERAL UNLOCKING (BY EXAMPLE)

$\mathcal{O}^t = \cancel{y_1} + y_2 + \cancel{y_3} + y_4 + \cancel{y_5} + \cancel{y_6} + \cancel{y_7} + y_8 + y_9 + \ldots$

Trail: $x_1^d \ \overline{x_2}^d \ x_3^p \ \overline{x_4}^d \ x_5^p \ \overline{y_9}^a \ y_1^p \ y_3^p \ \overline{y_5}^a \ \overline{y_6}^a \ y_7^p$

Found disjoint local "cores"

Core 1: $\overline{x_2} \land \overline{x_4} \rightarrow y_3 + y_5 + y_6 \geq 1$ "$\{y_9\}$ unlocks Core 1 on $\{y_3\}$"

Core 2: $x_1 \land \overline{x_2} \rightarrow y_1 + y_2 + y_4 + y_7 + y_8 \geq 2$ "$\{y_9, y_5, y_6\}$ unlocks Core 2 on $\{y_1, y_7\}$"

LOOKAHEAD WITH LITERAL UNLOCKING (BY EXAMPLE)

$\mathcal{O}^t = \not{y_1} + \not{y_2} + \not{y_3} + y_4 + \not{y_5} + \not{y_6} + \not{y_7} + y_8 + y_9 + \ldots$

Trail: $x_1^d\ \overline{x_2}^d\ x_3^p\ \overline{x_4}^d\ x_5^p\ \overline{y_9}^a\ y_1^p\ y_3^p\ \overline{y_5}^a\ \overline{y_6}^a\ y_7^p\ \overline{y_2}^a\ \bot$

Found disjoint local "cores"

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_3 + y_5 + y_6 \geq 1$ "$\{y_9\}$ unlocks Core 1 on $\{y_3\}$"

Core 2: $x_1 \wedge \overline{x_2} \rightarrow y_1 + y_2 + y_4 + y_7 + y_8 \geq 2$ "$\{y_9, y_5, y_6\}$ unlocks Core 2 on $\{y_1, y_7\}$"

LOOKAHEAD WITH LITERAL UNLOCKING (BY EXAMPLE)

$\mathcal{O}^t = \cancel{y_1} + \cancel{y_2} + \cancel{y_3} + y_4 + \cancel{y_5} + \cancel{y_6} + \cancel{y_7} + y_8 + y_9 + \ldots$

Trail: $x_1^d \ \overline{x_2}^d \ x_3^p \ \overline{x_4}^d \ x_5^p \ \overline{y_9}^a \ y_1^p \ y_3^p \ \overline{y_5}^a \ \overline{y_6}^a \ y_7^p \ \overline{y_2}^a \ \bot$

Found disjoint local "cores"

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_3 + y_5 + y_6 \geq 1$ "$\{y_9\}$ unlocks Core 1 on $\{y_3\}$"

Core 2: $x_1 \wedge \overline{x_2} \rightarrow y_1 + y_2 + y_4 + y_7 + y_8 \geq 2$ "$\{y_9, y_5, y_6\}$ unlocks Core 2 on $\{y_1, y_7\}$"

New core: $x_1 \wedge \overline{x_4} \rightarrow y_9 + y_5 + y_6 + y_2 \geq 1$

LOOKAHEAD WITH LITERAL UNLOCKING (BY EXAMPLE)

$\mathcal{O}^t = \cancel{y_1} + \cancel{y_2} + \cancel{y_3} + \cancel{y_4} + \cancel{y_5} + \cancel{y_6} + \cancel{y_7} + y_8 + y_9 + \ldots$

Trail: $x_1^d\ \overline{x_2}^d\ x_3^p\ \overline{x_4}^d\ x_5^p\ \overline{y_9}^a\ y_1^p\ y_3^p\ \overline{y_5}^a\ \overline{y_6}^a\ y_7^p\ \overline{y_2}^a\ \bot$

Found disjoint local "cores"

Core 1: $\overline{x_2} \wedge \overline{x_4} \to y_3 + y_5 + y_6 \geq 1$ "$\{y_9\}$ unlocks Core 1 on $\{y_3\}$"

Core 2: $x_1 \wedge \overline{x_2} \to y_1 + y_2 + y_4 + y_7 + y_8 \geq 2$ "$\{y_9, y_5, y_6\}$ unlocks Core 2 on $\{y_1, y_7\}$"

New core: $x_1 \wedge \overline{x_4} \to y_9 + y_5 + y_6 + y_2 \geq 1$

Addition of cores: $x_1 \wedge \overline{x_2} \wedge \overline{x_4} \to y_1 + 2y_2 + y_3 + y_4 + 2y_5 + 2y_6 + y_7 + y_8 + y_9 \geq 4$

LOOKAHEAD WITH LITERAL UNLOCKING (BY EXAMPLE)

$\mathcal{O}^t = \not{y_1} + \not{y_2} + \not{y_3} + y_4 + \not{y_5} + \not{y_6} + \not{y_7} + y_8 + y_9 + \ldots$

Trail: $x_1^d \; \overline{x_2}^d \; x_3^p \; \overline{x_4}^d \; x_5^p \; \overline{y_9}^a \; y_1^p \; y_3^p \; \overline{y_5}^a \; \overline{y_6}^a \; y_7^p \; \overline{y_2}^a \; \bot$

Found disjoint local "cores"

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_3 + y_5 + y_6 \geq 1$ "$\{y_9\}$ unlocks Core 1 on $\{y_3\}$"

Core 2: $x_1 \wedge \overline{x_2} \rightarrow y_1 + y_2 + y_4 + y_7 + y_8 \geq 2$ "$\{y_9, y_5, y_6\}$ unlocks Core 2 on $\{y_1, y_7\}$"

New core: $x_1 \wedge \overline{x_4} \rightarrow y_9 + y_5 + y_6 + y_2 \geq 1$

~~Addition of cores: $x_1 \wedge \overline{x_2} \wedge \overline{x_4} \rightarrow y_1 + 2y_2 + y_3 + y_4 + 2y_5 + 2y_6 + y_7 + y_8 + y_9 \geq 4$~~

LOOKAHEAD WITH LITERAL UNLOCKING (BY EXAMPLE)

$\mathcal{O}^t = \cancel{y_1} + \cancel{y_2} + \cancel{y_3} + y_4 + \cancel{y_5} + \cancel{y_6} + \cancel{y_7} + \cancel{y_8} + \cancel{y_9} + \ldots$

Trail: $x_1^d\ \overline{x_2}^d\ x_3^p\ \overline{x_4}^d\ x_5^p\ \overline{y_9}^a\ y_1^p\ y_3^p\ \overline{y_5}^a\ \overline{y_6}^a\ y_7^p\ \overline{y_2}^a\ \bot$

Found disjoint local "cores"

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow y_3 + y_5 + y_6 \geq 1$ "$\{y_9\}$ unlocks Core 1 on $\{y_3\}$"

Core 2: $x_1 \wedge \overline{x_2} \rightarrow y_1 + y_2 + y_4 + y_7 + y_8 \geq 2$ "$\{y_9, y_5, y_6\}$ unlocks Core 2 on $\{y_1, y_7\}$"

New core: $x_1 \wedge \overline{x_4} \rightarrow y_9 + y_5 + y_6 + y_2 \geq 1$

~~Addition of cores: $x_1 \wedge \overline{x_2} \wedge \overline{x_4} \rightarrow y_1 + 2y_2 + y_3 + y_4 + 2y_5 + 2y_6 + y_7 + y_8 + y_9 \geq 4$~~

Conclusion $x_1 \wedge \overline{x}_2 \wedge \overline{x}_4 \rightarrow \sum_{i=1}^{9} y_i \geq 4$ **?**

LOOKAHEAD WITH LITERAL UNLOCKING (BY EXAMPLE)

$\mathcal{O}^t = \cancel{y_1} + \cancel{y_2} + \cancel{y_3} + y_4 + \cancel{y_5} + \cancel{y_6} + \cancel{y_7} + \cancel{y_8} + \cancel{y_9} + \dots$

Trail: $x_1^d\ \overline{x_2}^d\ x_3^p\ \overline{x_4}^d\ x_5^p\ \overline{y_9}^a\ y_1^p\ y_3^p\ \overline{y_5}^a\ \overline{y_6}^a\ y_7^p\ \overline{y_2}^a\ \bot$

Found disjoint local "cores"

Core 1: $\overline{x_2} \wedge \overline{x_4} \wedge \overline{y_9} \rightarrow y_3 + \cancel{y_5} + \cancel{y_6} \geq 1$ "$\{y_9\}$ unlocks Core 1 on $\{y_3\}$"

Core 2: $x_1 \wedge \overline{x_2} \rightarrow y_1 + y_2 + y_4 + y_7 + y_8 \geq 2$ "$\{y_9, y_5, y_6\}$ unlocks Core 2 on $\{y_1, y_7\}$"

New core: $x_1 \wedge \overline{x_4} \rightarrow \cancel{y_9} + y_5 + y_6 + y_2 \geq 1$

~~Addition of cores: $x_1 \wedge \overline{x_2} \wedge \overline{x_4} \rightarrow y_1 + 2y_2 + y_3 + y_4 + 2y_5 + 2y_6 + y_7 + y_8 + y_9 \geq 4$~~

Conclusion $x_1 \wedge \overline{x}_2 \wedge \overline{x}_4 \rightarrow \sum_{i=1}^{9} y_i \geq 4$ **?**

LOOKAHEAD WITH LITERAL UNLOCKING (BY EXAMPLE)

$\mathcal{O}^t = \cancel{y_1} + \cancel{y_2} + \cancel{y_3} + y_4 + \cancel{y_5} + \cancel{y_6} + \cancel{y_7} + \cancel{y_8} + \cancel{y_9} + \dots$

Trail: $x_1^d\ \overline{x_2}^d\ x_3^p\ \overline{x_4}^d\ x_5^p\ \overline{y_9}^a\ y_1^p\ y_3^p\ \overline{y_5}^a\ \overline{y_6}^a\ y_7^p\ \overline{y_2}^a\ \bot$

Found disjoint local "cores"

Core 1: $\overline{x_2} \wedge \overline{x_4} \wedge \overline{y_9} \rightarrow y_3 + \cancel{y_5} + \cancel{y_6} \geq 1$ "$\{y_9\}$ unlocks Core 1 on $\{y_3\}$"

Core 2: $x_1 \wedge \overline{x_2} \wedge \overline{y_5} + \overline{y_6} \rightarrow y_1 + \cancel{y_2} + \cancel{y_4} + y_7 + \cancel{y_8} \geq 2$ "$\{y_9, y_5, y_6\}$ unlocks Core 2 on $\{y_1, y_7\}$"

New core: $x_1 \wedge \overline{x_4} \rightarrow \cancel{y_9} + \cancel{y_5} + \cancel{y_6} + y_2 \geq 1$

~~Addition of cores: $x_1 \wedge \overline{x_2} \wedge \overline{x_4} \rightarrow y_1 + 2y_2 + y_3 + y_4 + 2y_5 + 2y_6 + y_7 + y_8 + y_9 \geq 4$~~

Conclusion $x_1 \wedge \overline{x}_2 \wedge \overline{x}_4 \rightarrow \sum_{i=1}^{9} y_i \geq 4$ **?**

LOOKAHEAD WITH LITERAL UNLOCKING (BY EXAMPLE)

$\mathcal{O}^t = \cancel{y_1} + \cancel{y_2} + \cancel{y_3} + y_4 + \cancel{y_5} + \cancel{y_6} + \cancel{y_7} + \cancel{y_8} + \cancel{y_9} + \ldots$

Trail: $x_1^d\ \overline{x_2}^d\ x_3^p\ \overline{x_4}^d\ x_5^p\ \overline{y_9}^a\ y_1^p\ y_3^p\ \overline{y_5}^a\ \overline{y_6}^a\ y_7^p\ \overline{y_2}^a\ \bot$

Found disjoint local "cores"

Core 1: $\overline{x_2} \land \overline{x_4} \land \overline{y_9} \rightarrow y_3 + \cancel{y_5} + \cancel{y_6} \geq 1$ "$\{y_9\}$ unlocks Core 1 on $\{y_3\}$"

Core 2: $x_1 \land \overline{x_2} \land \overline{y_5} + \overline{y_6} \rightarrow y_1 + \cancel{y_2} + \cancel{y_4} + y_7 + \cancel{y_8} \geq 2$ "$\{y_9, y_5, y_6\}$ unlocks Core 2 on $\{y_1, y_7\}$"

New core: $x_1 \land \overline{x_4} \rightarrow \cancel{y_9} + \cancel{y_5} + \cancel{y_6} + y_2 \geq 1$

~~Addition of cores: $x_1 \land \overline{x_2} \land \overline{x_4} \rightarrow y_1 + 2y_2 + y_3 + y_4 + 2y_5 + 2y_6 + y_7 + y_8 + y_9 \geq 4$~~

Conclusion $x_1 \land \overline{x}_2 \land \overline{x}_4 \rightarrow \sum_{i=1}^{9} y_i \geq 4$

CERTIFYING LITERAL UNLOCKING

Proposition

Let $L_i|_{1\leq i\leq k}$ *and* L *be pairwise disjoint sets of objective literals and* $b_i|_{1\leq i\leq k}$ *natural numbers. Assume* $U_i \subseteq L_i$ *with* $|U_i| = b_i$ *for each* i *and write* R_i *for* $L_i \setminus U_i$*. From the constraints*

$$L_i \geq b_i \ (\forall 1 \leq i \leq k), \qquad L + \sum_{j<i} R_j + \ell \geq 1 \ (\forall 1 \leq i \leq k, \ell \in U_i), \qquad L + \sum_j R_j \geq 1$$

there is a cutting planes derivation that derives

$$L + \sum_{j\geq i} U_j + \sum_j R_j \geq 1 + \sum_{j\geq i} b_j \tag{1}$$

for each $i \in \{1, \ldots, k+1\}$*.*

OUTLINE OF THIS PRESENTATION

- What is MaxSAT and how to certify it?
- Proof logging the B&B solver MaxCDCL
- Proof logging additional techniques in MaxCDCL
  - Hardening
  - Literal Unlocking
- Proof logging BDD PB-to-CNF encoding
- Future work & Conclusions

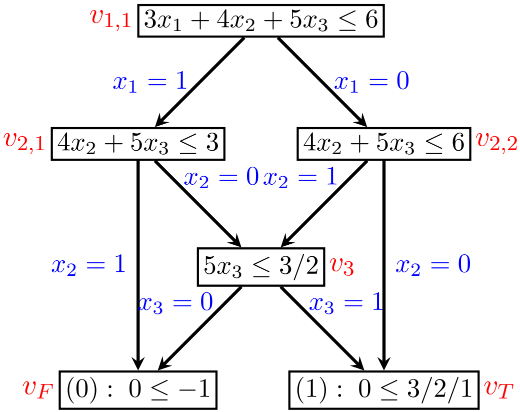
MAXCDCL'S USAGE OF BDDS

MaxCDCL $\cup$ Solution-Improving: MaxCDCL encodes model-improving constraint to enhance propagation.

MAXCDCL'S USAGE OF BDDS

MaxCDCL ∪ Solution-Improving: MaxCDCL encodes model-improving constraint to enhance propagation.

Binary Decision Diagram:

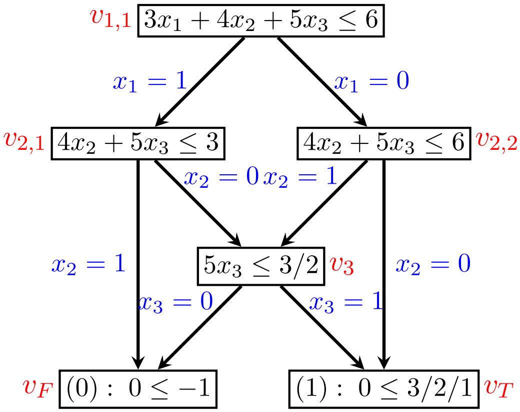


MAXCDCL'S USAGE OF BDDS

MaxCDCL ∪ Solution-Improving: MaxCDCL encodes model-improving constraint to enhance propagation.

Binary Decision Diagram:

- Every node corresponds with part of the original PB constraint and,

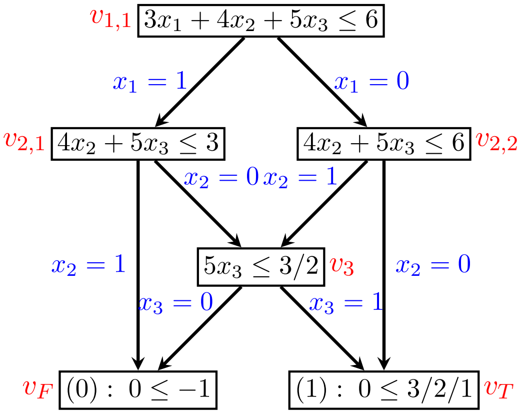


MAXCDCL'S USAGE OF BDDS

MaxCDCL ∪ Solution-Improving: MaxCDCL encodes model-improving constraint to enhance propagation.

Binary Decision Diagram:

- Every node corresponds with part of the original PB constraint and,
- Every node propagates based on one decision literal.

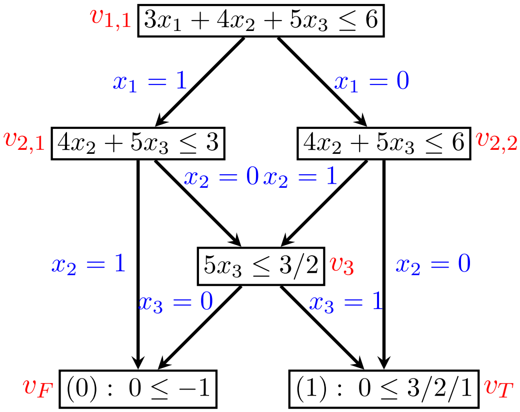


MAXCDCL'S USAGE OF BDDS

MaxCDCL ∪ Solution-Improving: MaxCDCL encodes model-improving constraint to enhance propagation.

Binary Decision Diagram:

- Every node corresponds with part of the original PB constraint and,
- Every node propagates based on one decision literal.
- If v_F node is propagated true, then constraint in root is falsified.

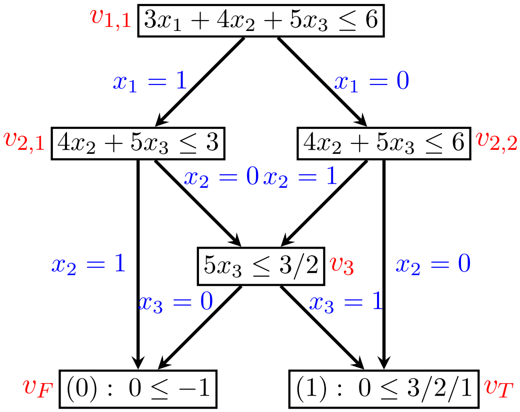


MAXCDCL'S USAGE OF BDDS

MaxCDCL ∪ Solution-Improving: MaxCDCL encodes model-improving constraint to enhance propagation.

Introducing reification variables for each node:

- E.g., $v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 6$

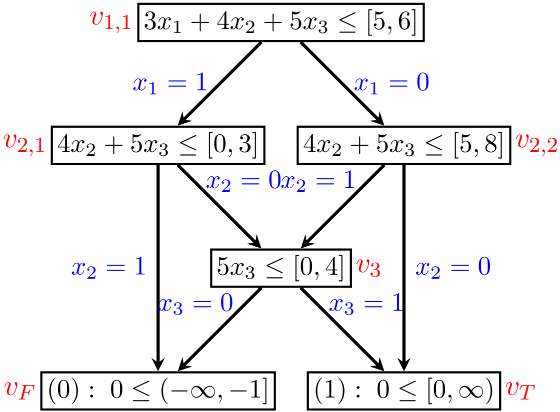


MAXCDCL'S USAGE OF BDDS

MaxCDCL ∪ Solution-Improving: MaxCDCL encodes model-improving constraint to enhance propagation.

Introducing reification variables for each node:

- E.g., $v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 6$
- But also $v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 7$

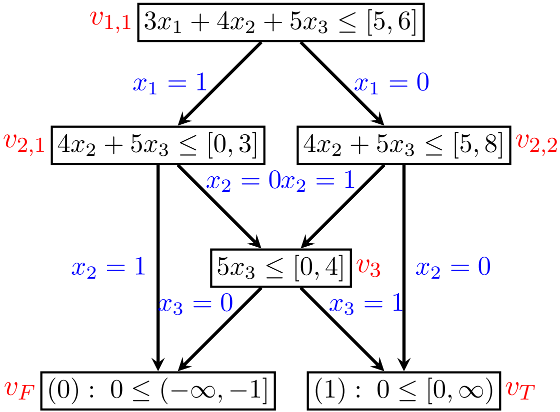


MAXCDCL'S USAGE OF BDDS

MaxCDCL ∪ Solution-Improving: MaxCDCL encodes model-improving constraint to enhance propagation.

Introducing reification variables for each node:

- E.g., $v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 6$
- But also $v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 7$
- Hence, $v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq [5,8]$



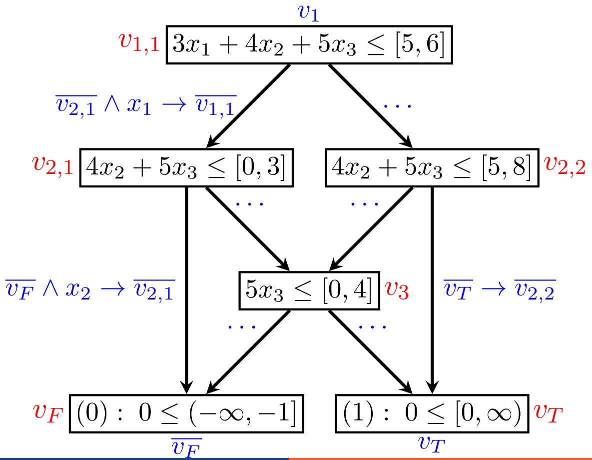
MAXCDCL'S USAGE OF BDDS

MaxCDCL $\cup$ Solution-Improving: MaxCDCL encodes model-improving constraint to enhance propagation.

Introducing reification variables for each node:

- E.g., $v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 6$
- But also $v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 7$
- Hence, $v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq [5, 8]$

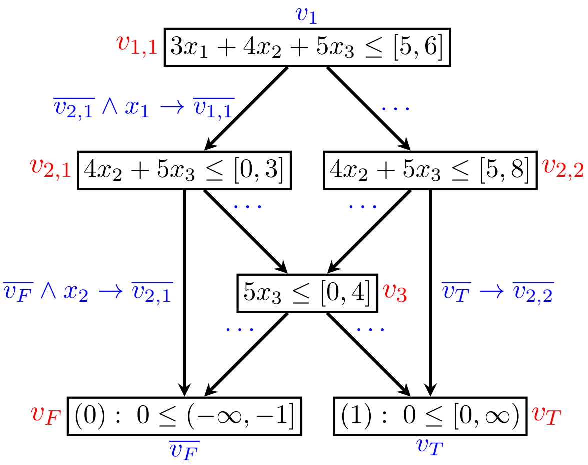
After introducing the reification variables, clauses are added to the solver.



HOW TO CERTIFY BDDS?

Step 1: Derive reification of node variables. E.g.,

- $v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq [5, 8]$
  - $v_{2,2} \rightarrow 4x_2 + 5x_3 \leq 5$
  - $v_{2,2} \leftarrow 4x_2 + 5x_3 \leq 8$



HOW TO CERTIFY BDDS?

Step 1: Derive reification of node variables. E.g.,

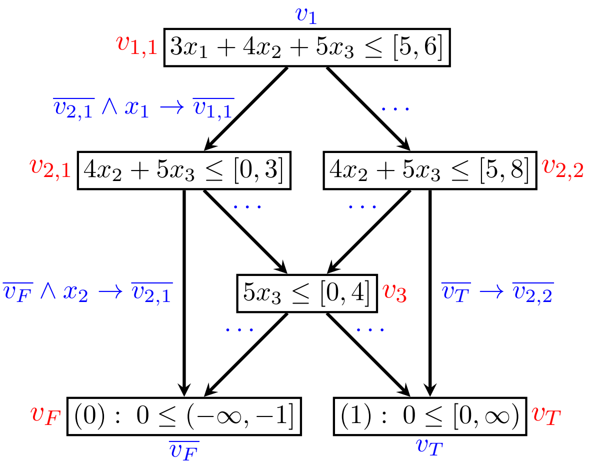
- $v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq [5,8]$
  - $v_{2,2} \rightarrow 4x_2 + 5x_3 \leq 5$
  - $v_{2,2} \leftarrow 4x_2 + 5x_3 \leq 8$

by introducing

- $v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 5$
- $v'_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 8$ (only in proof)

and deriving

- $v'_{2,2} \rightarrow v_{2,2}$



HOW TO CERTIFY BDDS?

Step 1: Derive reification of node variables. E.g.,

- $v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq [5, 8]$
  - $v_{2,2} \rightarrow 4x_2 + 5x_3 \leq 5$
  - $v_{2,2} \leftarrow 4x_2 + 5x_3 \leq 8$

by introducing

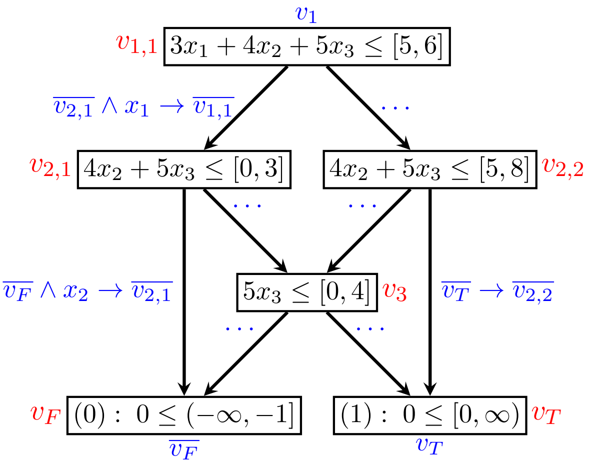
- $v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 5$
- $v'_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 8$ (only in proof)

and deriving

- $v'_{2,2} \rightarrow v_{2,2}$

Step 2: Derive clauses.

- Straight-forward cutting planes derivation.



INTERMEZZO: PROOF BY CONTRADICTION

Remember definition of Redundance-Based Strengthening:

Definition

A constraint C is redundant with respect to the pseudo-Boolean formula F if and only if there exists a substitution ω, called a witness, such that

$$F \wedge \neg C \models F|_\omega \wedge C|_\omega$$

INTERMEZZO: PROOF BY CONTRADICTION

Remember definition of Redundance-Based Strengthening:

Definition

A constraint C is redundant with respect to the pseudo-Boolean formula F if and only if there exists a substitution ω, called a witness, such that

$$F \wedge \neg C \models F|_\omega \wedge C|_\omega$$

Proof by contradiction — Take empty witness.

Condition to prove RBS becomes:

$$F \wedge \neg C \models F \wedge C$$

Only one non-trivial proof goal:

$$F \wedge \neg C \wedge \neg C \vdash 0 \geq 1$$

INTERMEZZO: PROOF BY CASE SPLITTING

Suppose we have derived two constraints:

$$a \cdot x + \sum_i b_i l_i \geq B \qquad\qquad a \cdot \overline{x} + \sum_i b_i l_i \geq B$$

We want to derive the constraint

$$\sum_i b_i l_i \geq B$$

INTERMEZZO: PROOF BY CASE SPLITTING

Suppose we have derived two constraints:

$$a \cdot x + \sum_i b_i l_i \geq B \qquad\qquad a \cdot \overline{x} + \sum_i b_i l_i \geq B$$

We want to derive the constraint

$$\sum_i b_i l_i \geq B$$

Following completeness of Cutting Planes: Should be possible.

INTERMEZZO: PROOF BY CASE SPLITTING

Suppose we have derived two constraints:

$$a \cdot x + \sum_i b_i l_i \geq B \qquad\qquad a \cdot \overline{x} + \sum_i b_i l_i \geq B$$

We want to derive the constraint

$$\sum_i b_i l_i \geq B$$

Following completeness of Cutting Planes: Should be possible.

Unfortunately, we don't know how to do this using cutting planes derivation [BN21].

INTERMEZZO: PROOF BY CASE SPLITTING

Suppose we have derived two constraints:

$$a \cdot x + \sum_i b_i l_i \geq B \qquad\qquad a \cdot \overline{x} + \sum_i b_i l_i \geq B$$

We want to derive the constraint

$$\sum_i b_i l_i \geq B$$

Following completeness of Cutting Planes: Should be possible.

Unfortunately, we don't know how to do this using cutting planes derivation [BN21].

Luckily, possible by proof by contradiction [Van23].

PROVING REIFICATION OF NODE VARIABLES

We have

- $v_{2,2} \rightarrow 4x_2 + 5x_3 \leq 5$
- $v'_{2,2} \leftarrow 4x_2 + 5x_3 \leq 8$

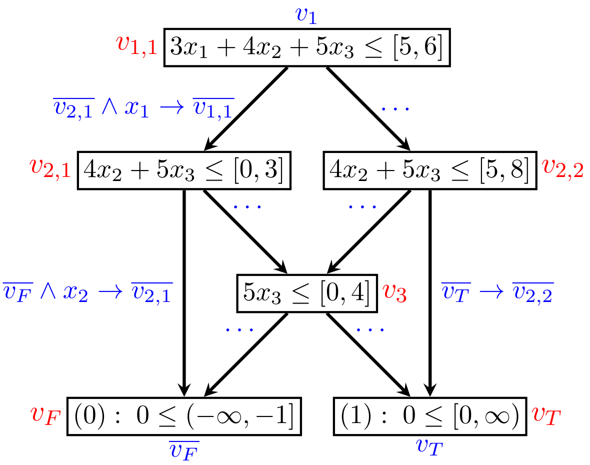
and we want to derive

- $v'_{2,2} \rightarrow v_{2,2}$

If we can prove

- $\overline{x}_2 + \overline{v}'_{2,2} + v_{2,2} \geq 1$
- $x_2 + \overline{v}'_{2,2} + v_{2,2} \geq 1$

then by case splitting $\overline{v}'_{2,2} + v_{2,2} \geq 1$ follows.



PROVING REIFICATION OF NODE VARIABLES

To derive:

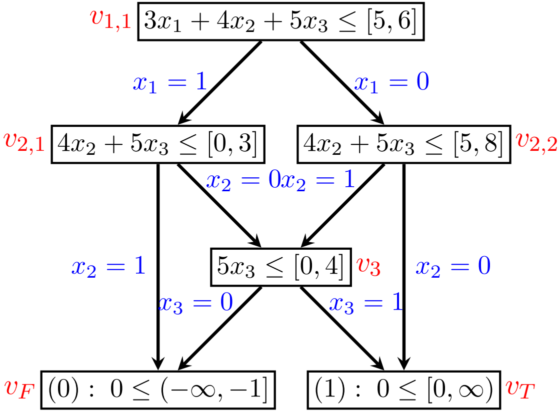
- $\overline{x}_2 + \overline{v}'_{2,2} + v_{2,2} \geq 1$

We have for node $v_{2,2}$:

- $v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 5$
- $v'_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 8$

For node v_3:

- $v_3 \rightarrow 5x_3 \leq 0$
- $v_3 \leftarrow 5x_3 \leq 4$



PROVING REIFICATION OF NODE VARIABLES (BY CONTRADICTION)

To Derive: $\overline{x}_2 + \overline{v}'_{2,2} + v_{2,2} \geq 1$. We assume the negation, i.e.,

$$x_2 \geq 1, \qquad v'_{2,2} \geq 1, \qquad \overline{v}_{2,2} \geq 1$$

PROVING REIFICATION OF NODE VARIABLES (BY CONTRADICTION)

To Derive: $\overline{x}_2 + \overline{v}'_{2,2} + v_{2,2} \geq 1$. We assume the negation, i.e.,

$$x_2 \geq 1, \qquad v'_{2,2} \geq 1, \qquad \overline{v}_{2,2} \geq 1$$

Constraints already derived:

$$v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 5 \qquad v'_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 8$$

$$v_3 \rightarrow 5x_3 \leq 0 \qquad v_3 \leftarrow 5x_3 \leq 4$$

PROVING REIFICATION OF NODE VARIABLES (BY CONTRADICTION)

To Derive: $\overline{x}_2 + \overline{v}'_{2,2} + v_{2,2} \geq 1$. We assume the negation, i.e.,

$$x_2 \geq 1, \qquad v'_{2,2} \geq 1, \qquad \overline{v}_{2,2} \geq 1$$

Constraints already derived:

$$v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 5 \qquad v'_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 8$$

$$v_3 \rightarrow 5x_3 \leq 0 \qquad v_3 \leftarrow 5x_3 \leq 4$$

From $v'_{2,2} \geq 1$: $4x_2 + 5x_3 \leq 8$

Using $x_2 \geq 1$: $5x_3 \leq 4$

Using definition of v_3: $v_3 \geq 1$

PROVING REIFICATION OF NODE VARIABLES (BY CONTRADICTION)

To Derive: $\overline{x}_2 + \overline{v}'_{2,2} + v_{2,2} \geq 1$. We assume the negation, i.e.,

$$x_2 \geq 1, \qquad v'_{2,2} \geq 1, \qquad \overline{v}_{2,2} \geq 1$$

Constraints already derived:

$$v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 5 \qquad v'_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 8$$

$$v_3 \rightarrow 5x_3 \leq 0 \qquad v_3 \leftarrow 5x_3 \leq 4$$

From $v'_{2,2} \geq 1$: $4x_2 + 5x_3 \leq 8$	From $\overline{v}_{2,2} \geq 1$: $4x_2 + 5x_3 \geq 5 + 1$
Using $x_2 \geq 1$: $5x_3 \leq 4$	Weakening x_2: $5x_3 \geq 2$
Using definition of v_3: $v_3 \geq 1$	Using definition of v_3: $\overline{v}_3 \geq 1$

PROVING REIFICATION OF NODE VARIABLES (BY CONTRADICTION)

To Derive: $\overline{x}_2 + \overline{v}'_{2,2} + v_{2,2} \geq 1$. We assume the negation, i.e.,

$$x_2 \geq 1, \qquad v'_{2,2} \geq 1, \qquad \overline{v}_{2,2} \geq 1$$

Constraints already derived:

$$v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 5 \qquad v'_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 8$$

$$v_3 \rightarrow 5x_3 \leq 0 \qquad v_3 \leftarrow 5x_3 \leq 4$$

From $v'_{2,2} \geq 1$: $4x_2 + 5x_3 \leq 8$

Using $x_2 \geq 1$: $5x_3 \leq 4$

Using definition of v_3: $v_3 \geq 1$

From $\overline{v}_{2,2} \geq 1$: $4x_2 + 5x_3 \geq 5 + 1$

Weakening x_2: $5x_3 \geq 2$

Using definition of v_3: $\overline{v}_3 \geq 1$

Contradiction.

PROVING REIFICATION OF NODE VARIABLES (BY CONTRADICTION)

To Derive: $\overline{x}_2 + \overline{v}'_{2,2} + v_{2,2} \geq 1$. We assume the negation, i.e.,

$$x_2 \geq 1, \qquad v'_{2,2} \geq 1, \qquad \overline{v}_{2,2} \geq 1$$

Constraints already derived:

$$v_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 5 \qquad v'_{2,2} \leftrightarrow 4x_2 + 5x_3 \leq 8$$

$$v_3 \rightarrow 5x_3 \leq 0 \qquad v_3 \leftarrow 5x_3 \leq 4$$

From $v'_{2,2} \geq 1$: $4x_2 + 5x_3 \leq 8$

Using $x_2 \geq 1$: $5x_3 \leq 4$

Using definition of v_3: $v_3 \geq 1$

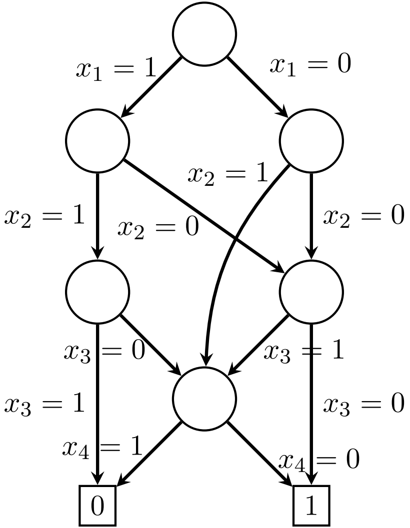
From $\overline{v}_{2,2} \geq 1$: $4x_2 + 5x_3 \geq 5 + 1$

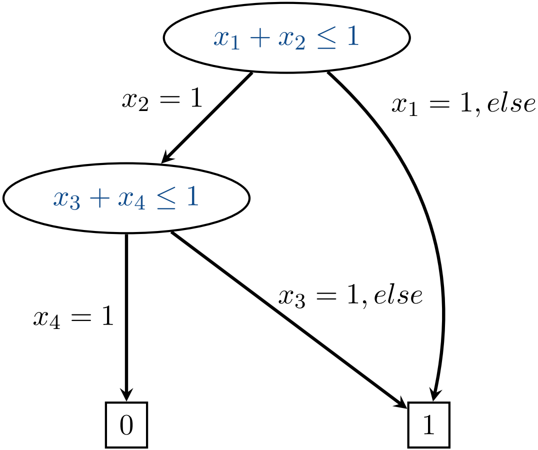
Weakening x_2: $5x_3 \geq 2$

Using definition of v_3: $\overline{v}_3 \geq 1$

Contradiction. Same reasoning to obtain $x_2 + \overline{v}'_{2,2} + v_{2,2} \geq 1$.

MULTI-VALUED DECISION DIAGRAM (MDD)





OUTLINE OF THIS PRESENTATION

- What is MaxSAT and how to certify it?
- Proof logging the B&B solver MaxCDCL
- Proof logging additional techniques in MaxCDCL
  - Hardening
  - Literal Unlocking
- Proof logging BDD PB-to-CNF encoding
- Future work & Conclusions

WRAPPING UP

This talk:

- MaxCDCL
  - MaxSAT solving by combining Branch-and-Bound and CDCL
  - Encoding the model-improving constraint using MDD encoding

WRAPPING UP

This talk:

- MaxCDCL
  - MaxSAT solving by combining Branch-and-Bound and CDCL
  - Encoding the model-improving constraint using MDD encoding
- Proof logging is possible with VeriPB!!
- Work in progress paper submitted to Pragmatics of SAT

WRAPPING UP

This talk:

- MaxCDCL
  - MaxSAT solving by combining Branch-and-Bound and CDCL
  - Encoding the model-improving constraint using MDD encoding
- Proof logging is possible with VeriPB!!
- Work in progress paper submitted to Pragmatics of SAT

Future work:

- Implementation & Experiments
- Implicit Hitting Set solvers

WRAPPING UP

This talk:

- MaxCDCL
  - MaxSAT solving by combining Branch-and-Bound and CDCL
  - Encoding the model-improving constraint using MDD encoding
- Proof logging is possible with VeriPB!!
- Work in progress paper submitted to Pragmatics of SAT

Future work:

- Implementation & Experiments
- Implicit Hitting Set solvers
- Certified track in MaxSAT competition?

WRAPPING UP

This talk:

- MaxCDCL
  - MaxSAT solving by combining Branch-and-Bound and CDCL
  - Encoding the model-improving constraint using MDD encoding
- Proof logging is possible with VeriPB!!
- Work in progress paper submitted to Pragmatics of SAT

Future work:

- Implementation & Experiments
- Implicit Hitting Set solvers
- Certified track in MaxSAT competition?
- Other fields of combinatorial solving

WRAPPING UP

This talk:

- MaxCDCL
  - MaxSAT solving by combining Branch-and-Bound and CDCL
  - Encoding the model-improving constraint using MDD encoding
- Proof logging is possible with VeriPB!!
- Work in progress paper submitted to Pragmatics of SAT

Future work:

- Implementation & Experiments
- Implicit Hitting Set solvers
- Certified track in MaxSAT competition?
- Other fields of combinatorial solving — Interesting things happening!

WRAPPING UP

This talk:

- MaxCDCL
  - MaxSAT solving by combining Branch-and-Bound and CDCL
  - Encoding the model-improving constraint using MDD encoding
- Proof logging is possible with VeriPB!!
- Work in progress paper submitted to Pragmatics of SAT

Future work:

- Implementation & Experiments
- Implicit Hitting Set solvers
- Certified track in MaxSAT competition?
- Other fields of combinatorial solving — Interesting things happening!

Thank you for your attention!

REFERENCES

[BBN⁺23] Jeremias Berg, Bart Bogaerts, Jakob Nordström, Andy Oertel, and Dieter Vandesande. Certified core-guided MaxSAT solving. In Brigitte Pientka and Cesare Tinelli, editors, *Automated Deduction - CADE 29 - 29th International Conference on Automated Deduction, Rome, Italy, July 1-4, 2023, Proceedings*, volume 14132 of *Lecture Notes in Computer Science*, pages 1–22. Springer, 2023.

[BJ19] Jeremias Berg and Matti Järvisalo. Unifying reasoning and core-guided search for maximum satisfiability. In Francesco Calimeri, Nicola Leone, and Marco Manna, editors, *Logics in Artificial Intelligence - 16th European Conference, JELIA 2019, Rende, Italy, May 7-11, 2019, Proceedings*, volume 11468 of *Lecture Notes in Computer Science*, pages 287–303. Springer, 2019.

[BLM06] Maria Luisa Bonet, Jordi Levy, and Felip Manyà. A complete calculus for max-sat. In Armin Biere and Carla P. Gomes, editors, *Theory and Applications of Satisfiability Testing - SAT 2006, 9th International Conference, Seattle, WA, USA, August 12-15, 2006, Proceedings*, volume 4121 of *Lecture Notes in Computer Science*, pages 240–251. Springer, 2006.

[BLM07] Maria Luisa Bonet, Jordi Levy, and Felip Manyà. Resolution for max-sat. *Artif. Intell.*, 171(8-9):606–618, 2007.

REFERENCES

[BMM13] Anton Belov, António Morgado, and João Marques-Silva. Sat-based preprocessing for maxsat. In Kenneth L. McMillan, Aart Middeldorp, and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning - 19th International Conference, LPAR-19, Stellenbosch, South Africa, December 14-19, 2013. Proceedings*, volume 8312 of *Lecture Notes in Computer Science*, pages 96–111. Springer, 2013.

[BN21] Samuel R. Buss and Jakob Nordström. Proof complexity and SAT solving. In Armin Biere, Marijn J. H. Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, volume 336 of *Frontiers in Artificial Intelligence and Applications*, chapter 7, pages 233–350. IOS Press, 2nd edition, February 2021.

[HL06] Federico Heras and Javier Larrosa. New inference rules for efficient max-sat solving. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 68–73. AAAI Press, 2006.

[IBJ22] Hannes Ihalainen, Jeremias Berg, and Matti Järvisalo. Clause redundancy and preprocessing in maximum satisfiability. In Jasmin Blanchette, Laura Kovács, and Dirk Pattinson, editors, *Automated Reasoning - 11th International Joint Conference, IJCAR 2022, Haifa, Israel, August 8-10, 2022, Proceedings*, volume 13385 of *Lecture Notes in Computer Science*, pages 75–94. Springer, 2022.

REFERENCES

[LCH+22] Shoulin Li, Jordi Coll, Djamal Habet, Chu-Min Li, and Felip Manyà. A tableau calculus for maxsat based on resolution. In Atia Cortés, Francisco Grimaldo, and Tommaso Flaminio, editors, *Artificial Intelligence Research and Development - Proceedings of the 24th International Conference of the Catalan Association for Artificial Intelligence, CCIA 2022, Sitges, Spain, 19-21 October 2022*, volume 356 of *Frontiers in Artificial Intelligence and Applications*, pages 35–44. IOS Press, 2022.

[LH05] Javier Larrosa and Federico Heras. Resolution in max-sat and its relation to local consistency in weighted csps. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, pages 193–198. Professional Book Center, 2005.

[LM22] Chu Min Li and Felip Manyà. Inference in maxsat and minsat. In Wolfgang Ahrendt, Bernhard Beckert, Richard Bubel, and Einar Broch Johnsen, editors, *The Logic of Software. A Tasting Menu of Formal Methods - Essays Dedicated to Reiner Hähnle on the Occasion of His 60th Birthday*, volume 13360 of *Lecture Notes in Computer Science*, pages 350–369. Springer, 2022.

REFERENCES

[LMS16] Chu Min Li, Felip Manyà, and Joan Ramon Soler. A clause tableau calculus for minsat. In Àngela Nebot, Xavier Binefa, and Ramón López de Mántaras, editors, *Artificial Intelligence Research and Development - Proceedings of the 19th International Conference of the Catalan Association for Artificial Intelligence, Barcelona, Catalonia, Spain, October 19-21, 2016*, volume 288 of *Frontiers in Artificial Intelligence and Applications*, pages 88–97. IOS Press, 2016.

[LNOR11] Javier Larrosa, Robert Nieuwenhuis, Albert Oliveras, and Enric Rodríguez-Carbonell. A framework for certified Boolean branch-and-bound optimization. *J. Autom. Reason.*, 46(1):81–102, 2011.

[PCH21] Matthieu Py, Mohamed Sami Cherif, and Djamal Habet. A proof builder for max-sat. In Chu-Min Li and Felip Manyà, editors, *Theory and Applications of Satisfiability Testing - SAT 2021 - 24th International Conference, Barcelona, Spain, July 5-9, 2021, Proceedings*, volume 12831 of *Lecture Notes in Computer Science*, pages 488–498. Springer, 2021.

[PCH22] Matthieu Py, Mohamed Sami Cherif, and Djamal Habet. Proofs and certificates for max-sat. *J. Artif. Intell. Res.*, 75:1373–1400, 2022.

[Van23] Dieter Vandesande. Towards certified MaxSAT solving: Certified MaxSAT solving with SAT oracles and encodings of pseudo-Boolean constraints. Master's thesis, Vrije Universiteit Brussel (VUB), 2023.

REFERENCES

[VDB22] Dieter Vandesande, Wolf De Wulf, and Bart Bogaerts. QMaxSATpb: A certified MaxSAT solver. In Georg Gottlob, Daniela Inclezan, and Marco Maratea, editors, *Logic Programming and Nonmonotonic Reasoning - 16th International Conference, LPNMR 2022, Genova, Italy, September 5-9, 2022, Proceedings*, volume 13416 of *Lecture Notes in Computer Science*, pages 429–442. Springer, 2022.

HOW TO FIND SUCH CORES?

Definition

Let (b, L) be a cardinality constraint, $U \subsetneq L$, and L' a set of objective literals disjoint from L. L' unlocks (b, L) on U if $|U| \geq b$ and $F \wedge \alpha \wedge \bigwedge_{\ell \in L'} \overline{\ell} \models \ell'$ for each $\ell' \in U$.

Notation: (b, L) represents the cardinality constraint $\sum_{\ell \in L} l \geq b$.

Example:

$\mathcal{O} = y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8 + y_9 + y_{10}$
Local Core: $y_1 + y_2 + y_3 + y_4 + y_5 + y_6 \geq 3$

If assigning $y_7 = y_8 = 0$ propagates literals $y_1 \wedge y_3 \wedge y_6$,
then $L' = \{y_6, y_7\}$ unlocks $y_1 + y_2 + y_3 + y_4 + y_5 + y_6 \geq 3$ on $U = \{y_1, y_3, y_6\}$.

CERTIFYING LITERAL UNLOCKING

Proposition

Let $L_i|_{1\leq i\leq k}$ and L be pairwise disjoint sets of objective literals and $b_i|_{1\leq i\leq k}$ natural numbers. Assume $U_i \subseteq L_i$ with $|U_i| = b_i$ for each i and write R_i for $L_i \setminus U_i$. From the constraints

$$L_i \geq b_i \ (\forall 1 \leq i \leq k), \qquad L + \sum_{j<i} R_j + \ell \geq 1 \ (\forall 1 \leq i \leq k, \ell \in U_i), \qquad L + \sum_{j} R_j \geq 1$$

there is a cutting planes derivation that derives

$$L + \sum_{j\geq i} U_j + \sum_{j} R_j \geq 1 + \sum_{j\geq i} b_j \tag{2}$$

for each $i \in \{1, \ldots, k+1\}$.

CERTIFYING LITERAL UNLOCKING

To Derive: $L + \sum_{j \geq i} U_j + \sum_j R_j \geq 1 + \sum_{j \geq i} b_j$.

CERTIFYING LITERAL UNLOCKING

To Derive: $L + \sum_{j \geq i} U_j + \sum_j R_j \geq 1 + \sum_{j \geq i} b_j$. By **induction** on i.

CERTIFYING LITERAL UNLOCKING

To Derive: $L + \sum_{j \geq i} U_j + \sum_j R_j \geq 1 + \sum_{j \geq i} b_j$. By **induction** on i.

For $i = k + 1 : L + \sum_j R_j \geq 1$.

CERTIFYING LITERAL UNLOCKING

To Derive: $L + \sum_{j \geq i} U_j + \sum_j R_j \geq 1 + \sum_{j \geq i} b_j$. By **induction** on i.

For i between 1 and k (assuming already derived for $i+1$):

Step 1. Addition of $L + \sum_{j<i} R_j + \ell \geq 1$ for every $\ell \in U_i$ results in

$$b_i L + b_i \sum_{j<i} R_j + U_i \geq b_i$$

CERTIFYING LITERAL UNLOCKING

To Derive: $L + \sum_{j \geq i} U_j + \sum_j R_j \geq 1 + \sum_{j \geq i} b_j$. By **induction** on i.

For i between 1 and k (assuming already derived for $i+1$):

Step 1. Addition of $L + \sum_{j<i} R_j + \ell \geq 1$ for every $\ell \in U_i$ results in

$$b_i L + b_i \sum_{j<i} R_j + U_i \geq b_i$$

Step 2. Addition with IH gives:

$$((b_{i+1} + 1) \cdot L + \sum_{j \geq i} U_j + (b_{i+1} + 1) \sum_{j<i} R_j + \sum_{j \geq i} R_j \geq 1 + \sum_{j \geq i} b_j$$

CERTIFYING LITERAL UNLOCKING

To Derive: $L + \sum_{j \geq i} U_j + \sum_j R_j \geq 1 + \sum_{j \geq i} b_j$. By **induction** on i.

For i between 1 and k (assuming already derived for $i + 1$):

Step 2. Addition with IH gives:

$$((b_{i+1} + 1) \cdot L + \sum_{j \geq i} U_j + (b_{i+1} + 1) \sum_{j < i} R_j + \sum_{j \geq i} R_j \geq 1 + \sum_{j \geq i} b_j$$

Step 3. Multiplying all constraints $L_j \geq b_j$ for $j \geq i$ with b_{i+1} gives:

$$b_{i+1} \sum_{j \geq i} U_j + b_{i+1} \sum_{j \geq i} R_j \geq b_{i+1} \sum_{j \geq i} b_j$$

CERTIFYING LITERAL UNLOCKING

To Derive: $L + \sum_{j \geq i} U_j + \sum_j R_j \geq 1 + \sum_{j \geq i} b_j$. By **induction** on i.

For i between 1 and k (assuming already derived for $i+1$):

Step 2. Addition with IH gives:

$$((b_{i+1}+1) \cdot L + \sum_{j \geq i} U_j + (b_{i+1}+1) \sum_{j<i} R_j + \sum_{j \geq i} R_j \geq 1 + \sum_{j \geq i} b_j$$

Step 3. Multiplying all constraints $L_j \geq b_j$ for $j \geq i$ with b_{i+1} gives:

$$b_{i+1} \sum_{j \geq i} U_j + b_{i+1} \sum_{j \geq i} R_j \geq b_{i+1} \sum_{j \geq i} b_j$$

Step 4. Addition of constraints from *Step 2* and *Step 3*:

$$(b_{i+1}+1) \cdot L + (b_{i+1}+1) \sum_{j} R_j + (b_{i+1}+1) \sum_{j \geq i} R_j \geq 1 + (b_{i+1}+1) \sum_{j > i} b_j$$

CERTIFYING LITERAL UNLOCKING

To Derive: $L + \sum_{j \geq i} U_j + \sum_j R_j \geq 1 + \sum_{j \geq i} b_j$. By **induction** on i.

For i between 1 and k (assuming already derived for $i+1$):

Step 4. Addition of constraints from *Step 2* and *Step 3*:

$$(b_{i+1} + 1) \cdot L + (b_{i+1} + 1) \sum_j R_j + (b_{i+1} + 1) \sum_{j \geq i} R_j \geq 1 + (b_{i+1} + 1) \sum_{j > i} b_j$$

Step 5. Dividing this by $b_{i+1} + 1$ (and rounding the righthand-side up) yields

$$L + \sum_j R_j + \sum_{j \geq i} R_j \geq 1 + \sum_{j > i} b_j$$

INTERMEZZO: PROOF BY CASE SPLITTING

Suppose we have derived two constraints:

$$a \cdot x + \sum_i b_i l_i \geq B \qquad\qquad a \cdot \overline{x} + \sum_i b_i l_i \geq B$$

And we want to derive the constraint

$$\sum_i b_i l_i \geq B$$

INTERMEZZO: PROOF BY CASE SPLITTING

Suppose we have derived two constraints:

$$a \cdot x + \sum_i b_i l_i \geq B \qquad\qquad a \cdot \overline{x} + \sum_i b_i l_i \geq B$$

And we want to derive the constraint

$$\sum_i b_i l_i \geq B$$

By contradiction. Needed: CP derivation that shows

$$(a \cdot x + \sum_i b_i l_i \geq B) \wedge (a \cdot \overline{x} + \sum_i b_i l_i \geq B) \wedge \neg(\sum_i b_i l_i \geq B) \vdash 0 \geq 1$$

INTERMEZZO: PROOF BY CASE SPLITTING

Suppose we have derived two constraints:

$$a \cdot x + \sum_i b_i l_i \geq B \qquad\qquad a \cdot \overline{x} + \sum_i b_i l_i \geq B$$

And we want to derive the constraint

$$\sum_i b_i l_i \geq B$$

By contradiction. Needed: CP derivation that shows

$$(a \cdot x + \sum_i b_i l_i \geq B) \wedge (a \cdot \overline{x} + \sum_i b_i l_i \geq B) \wedge \neg(\sum_i b_i l_i \geq B) \vdash 0 \geq 1$$

After normalization:

$$(a \cdot x + \sum_i b_i l_i \geq B) \wedge (a \cdot \overline{x} + \sum_i b_i l_i \geq B) \wedge (\sum_i b_i l_i \geq \sum_i b_i - B + 1) \vdash 0 \geq 1$$

INTERMEZZO: PROOF BY CASE SPLITTING

To show:

$$(a \cdot x + \sum_i b_i l_i \geq B) \wedge (a \cdot \overline{x} + \sum_i b_i l_i \geq B) \wedge (\sum_i b_i \bar{l}_i \geq \sum_i b_i - B + 1) \vdash 0 \geq 1$$

Addition of $(a \cdot x + \sum_i b_i l_i \geq B)$ with $(\sum_i b_i \bar{l}_i \geq \sum_i b_i - B + 1)$ gives

$$a \cdot x + \sum_i b_i l_i + \sum_i b_i \bar{l}_i \geq B + \sum_i b_i - B + 1$$

which is equal to

$$a \cdot x \geq 1$$

After saturation: $x \geq 1$.
Similarly, addition of $(a \cdot \overline{x} + \sum_i b_i l_i \geq B)$ and $(\sum_i b_i \bar{l}_i \geq \sum_i b_i - B + 1)$ and saturation gives

$$\overline{x} \geq 1$$

which is clearly contradiction with $x \geq 1$.