

Computational Logic and Satisfiability

IN4077

Conflict-driven SAT solvers

Marijn J.H. Heule

September 23th 2008

Conflict-driven: Overview

- Most succesful architecture

Conflict-driven: Overview

- Most successful architecture
- Superior on industrial benchmarks

Conflict-driven: Overview

- Most successful architecture
- Superior on industrial benchmarks
- Brute-force?
 - Addition conflict clauses
 - Fast unit propagation

Conflict-driven: Overview

- Most successful architecture
- Superior on industrial benchmarks
- Brute-force?
 - Addition conflict clauses
 - Fast unit propagation
- State-of-the-art solvers: zChaff [MMZZ01], minisat [ES03], Rsat[PD07]

Conflict-driven: Analysis

$$x_1 \vee x_4$$

$$x_3 \vee \neg x_4 \vee \neg x_5$$

$$\neg x_3 \vee \neg x_2 \vee \neg x_4$$

$$\mathcal{F}_{\text{extra}}$$

①

Conflict-driven: Analysis

$$x_1 \vee x_4$$

$$x_3 \vee \neg x_4 \vee \neg x_5$$

$$\neg x_3 \vee \neg x_2 \vee \neg x_4$$

$\mathcal{F}_{\text{extra}}$

$$x_5 = 1 \begin{array}{c} \textcircled{0} \\ | \\ \textcircled{1} \end{array}$$

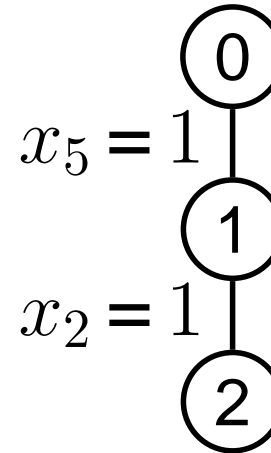
Conflict-driven: Analysis

$$x_1 \vee x_4$$

$$x_3 \vee \neg x_4 \vee \neg x_5$$

$$\neg x_3 \vee \neg x_2 \vee \neg x_4$$

$\mathcal{F}_{\text{extra}}$



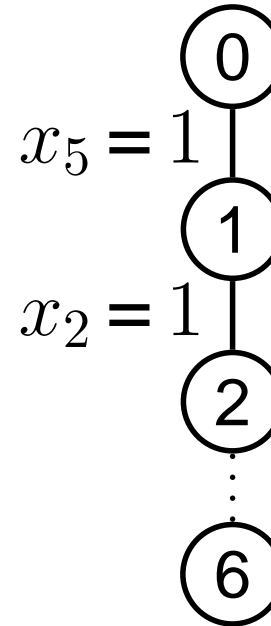
Conflict-driven: Analysis

$$x_1 \vee x_4$$

$$x_3 \vee \neg x_4 \vee \neg x_5$$

$$\neg x_3 \vee \neg x_2 \vee \neg x_4$$

$\mathcal{F}_{\text{extra}}$



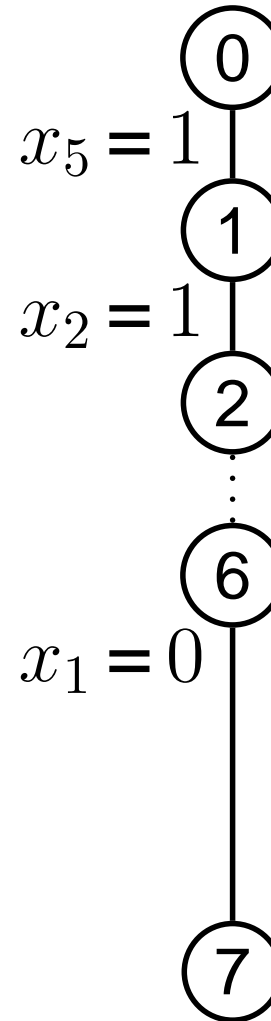
Conflict-driven: Analysis

$$x_1 \vee x_4$$

$$x_3 \vee \neg x_4 \vee \neg x_5$$

$$\neg x_3 \vee \neg x_2 \vee \neg x_4$$

$\mathcal{F}_{\text{extra}}$



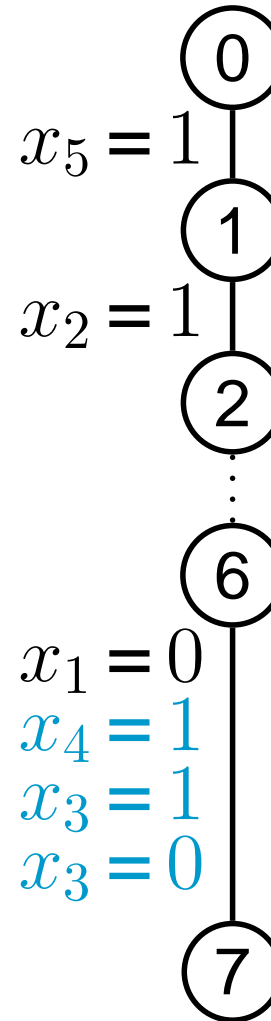
Conflict-driven: Analysis

$$x_1 \vee x_4$$

$$x_3 \vee \neg x_4 \vee \neg x_5$$

$$\neg x_3 \vee \neg x_2 \vee \neg x_4$$

$\mathcal{F}_{\text{extra}}$



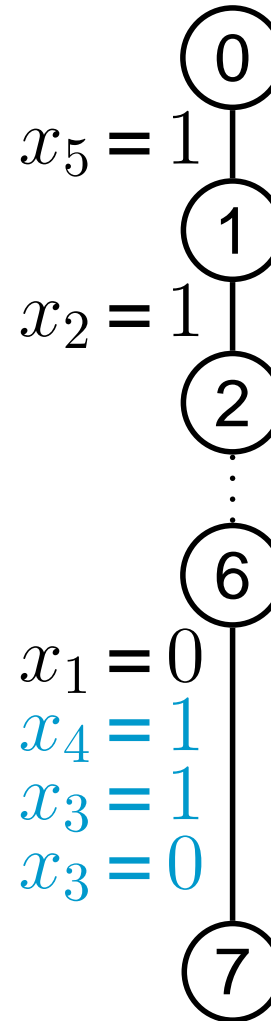
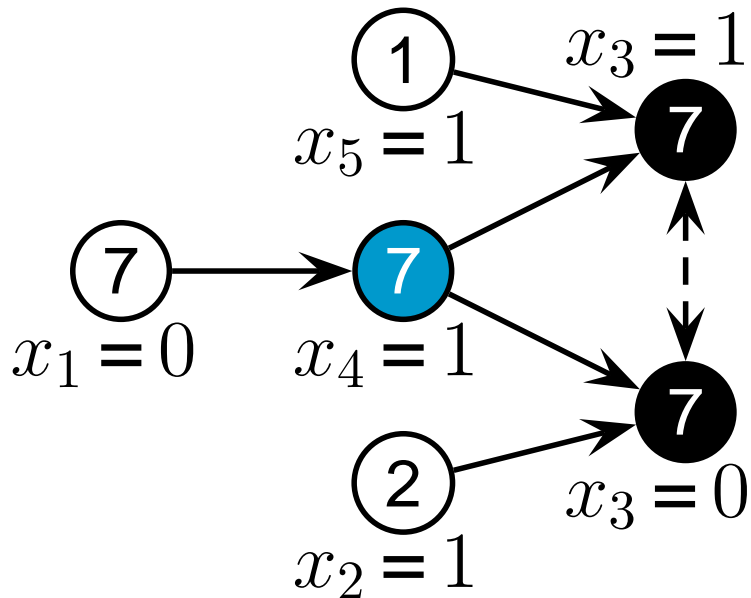
Conflict-driven: Analysis

$$x_1 \vee x_4$$

$$x_3 \vee \neg x_4 \vee \neg x_5$$

$$\neg x_3 \vee \neg x_2 \vee \neg x_4$$

$\mathcal{F}_{\text{extra}}$



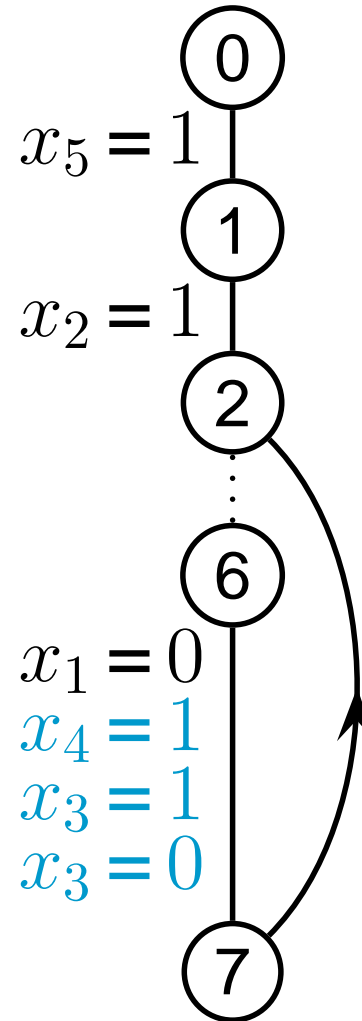
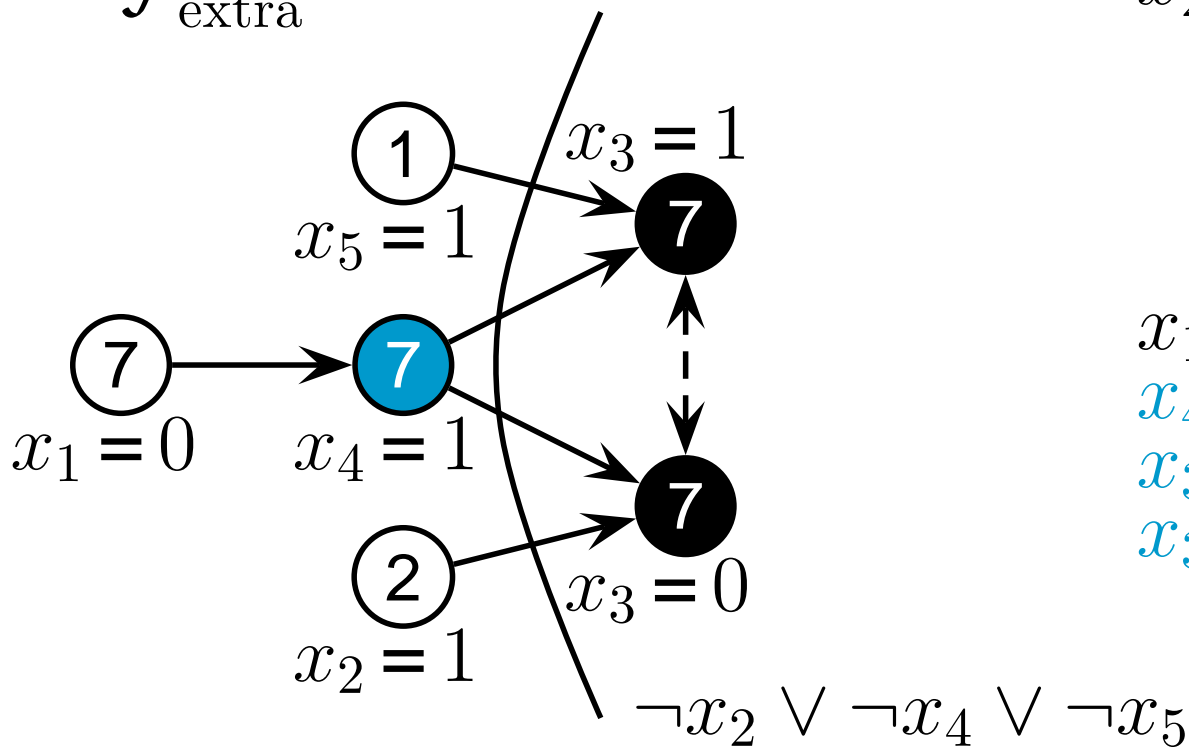
Conflict-driven: Analysis

$$x_1 \vee x_4$$

$$x_3 \vee \neg x_4 \vee \neg x_5$$

$$\neg x_3 \vee \neg x_2 \vee \neg x_4$$

$\mathcal{F}_{\text{extra}}$



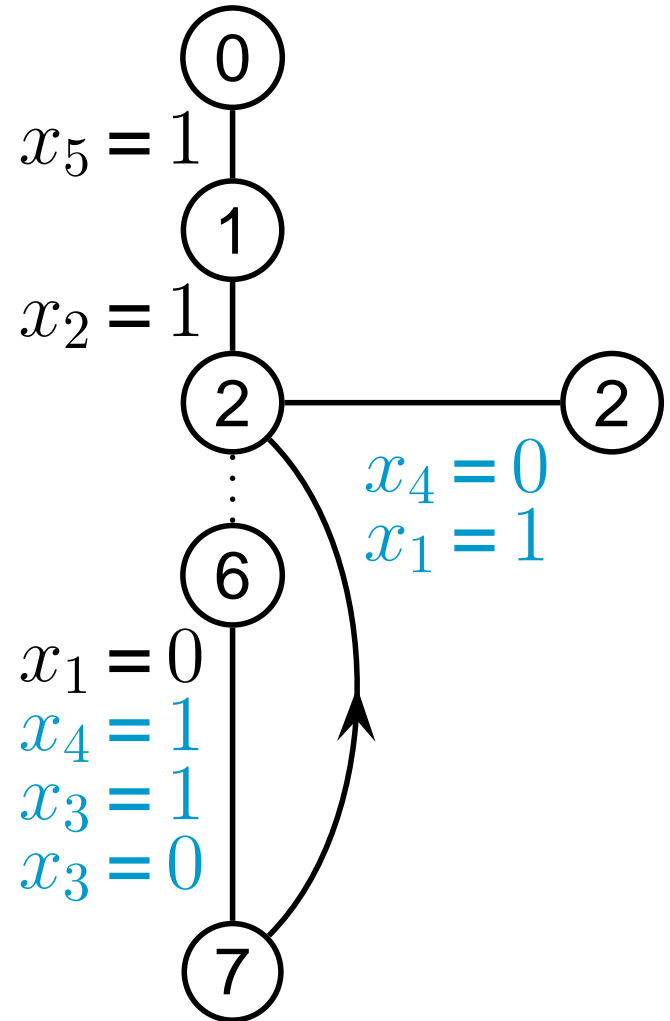
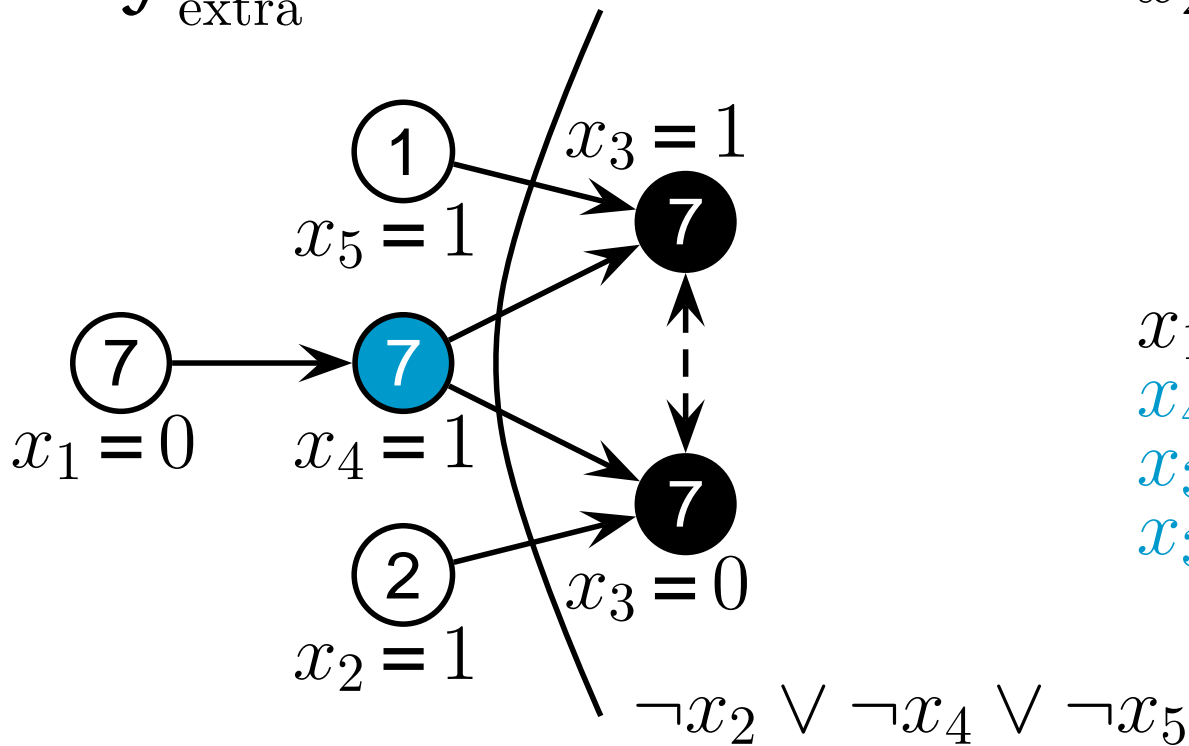
Conflict-driven: Analysis

$$x_1 \vee x_4$$

$$x_3 \vee \neg x_4 \vee \neg x_5$$

$$\neg x_3 \vee \neg x_2 \vee \neg x_4$$

$\mathcal{F}_{\text{extra}}$



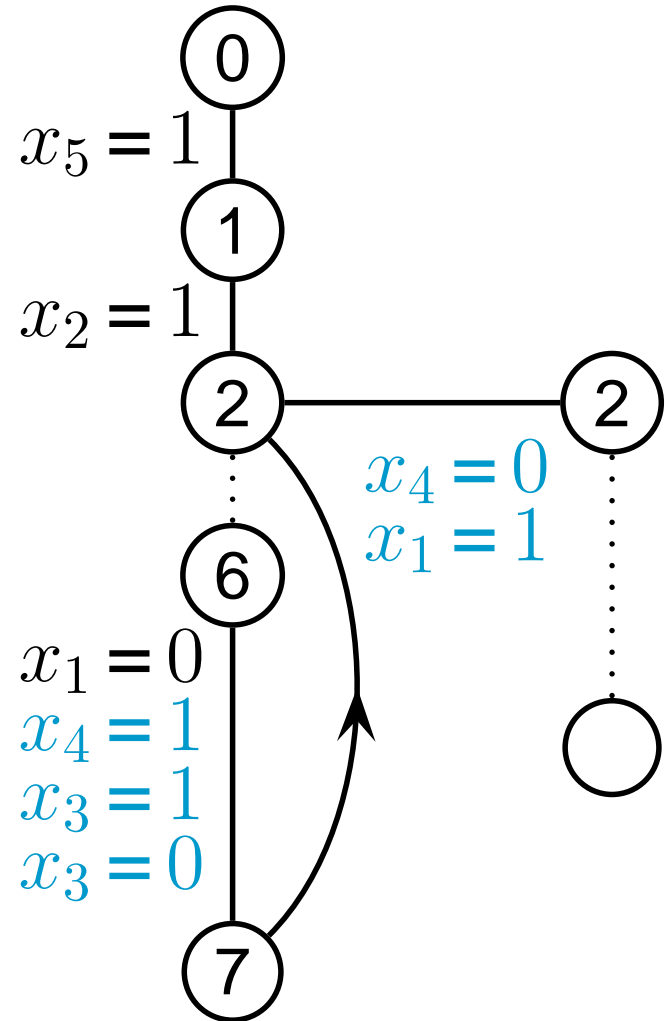
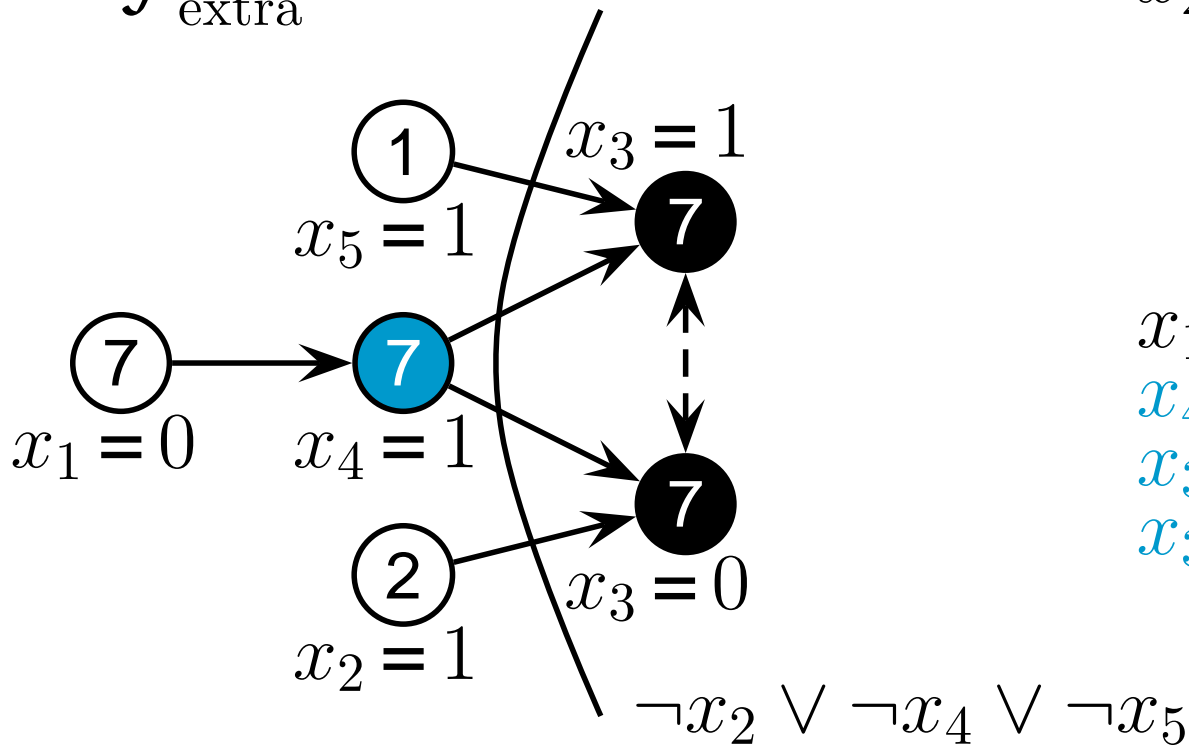
Conflict-driven: Analysis

$$x_1 \vee x_4$$

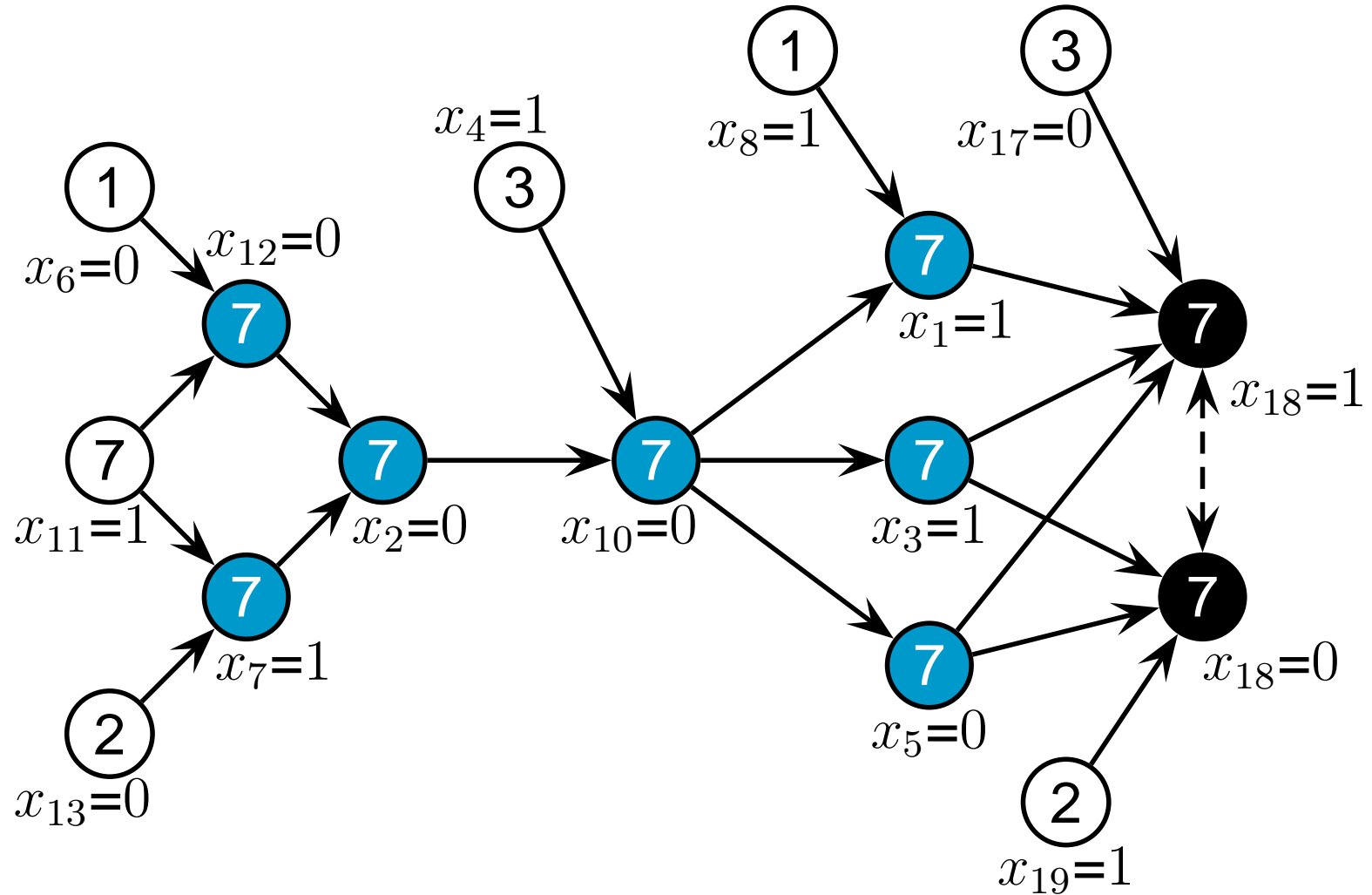
$$x_3 \vee \neg x_4 \vee \neg x_5$$

$$\neg x_3 \vee \neg x_2 \vee \neg x_4$$

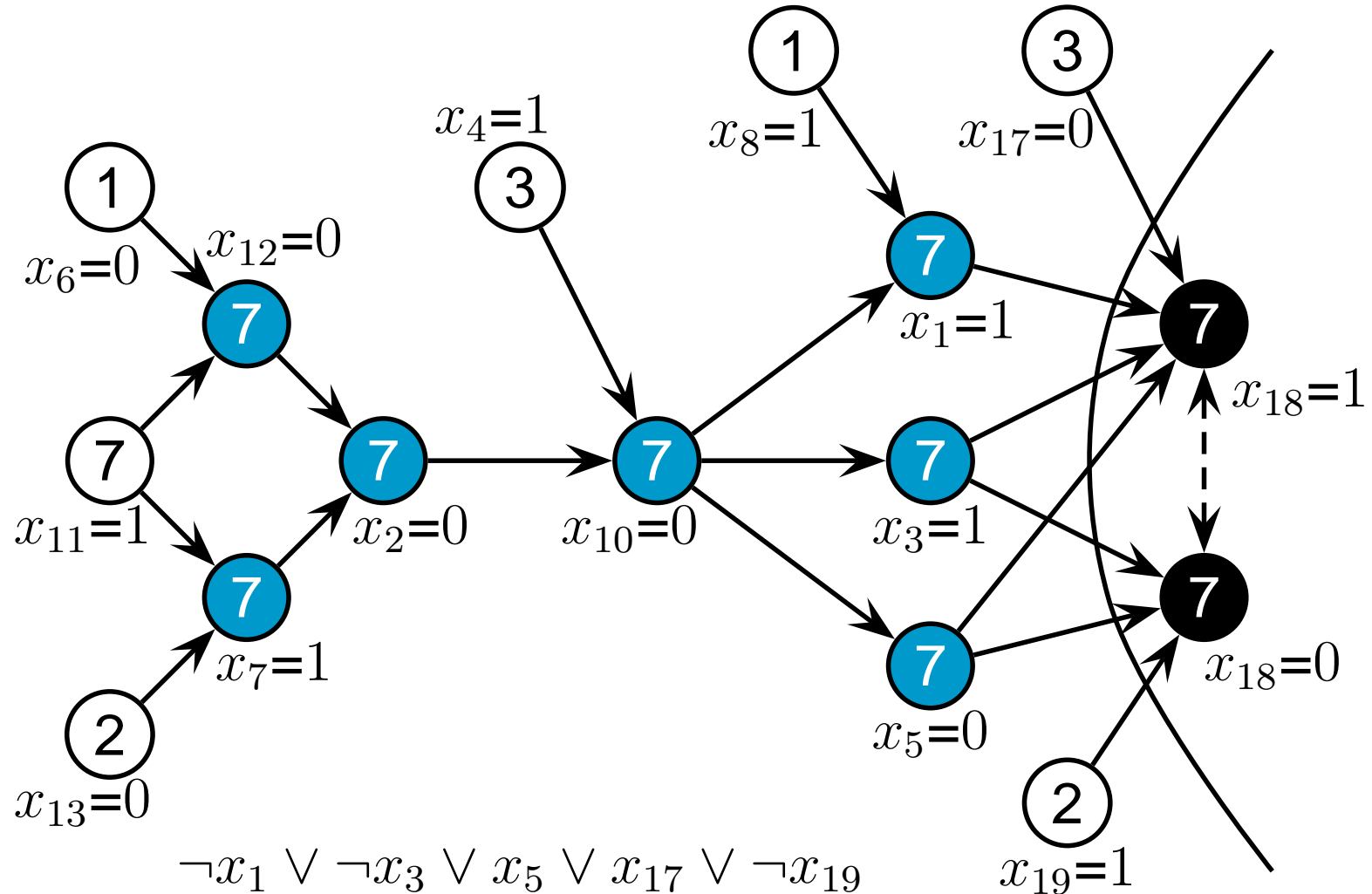
$\mathcal{F}_{\text{extra}}$



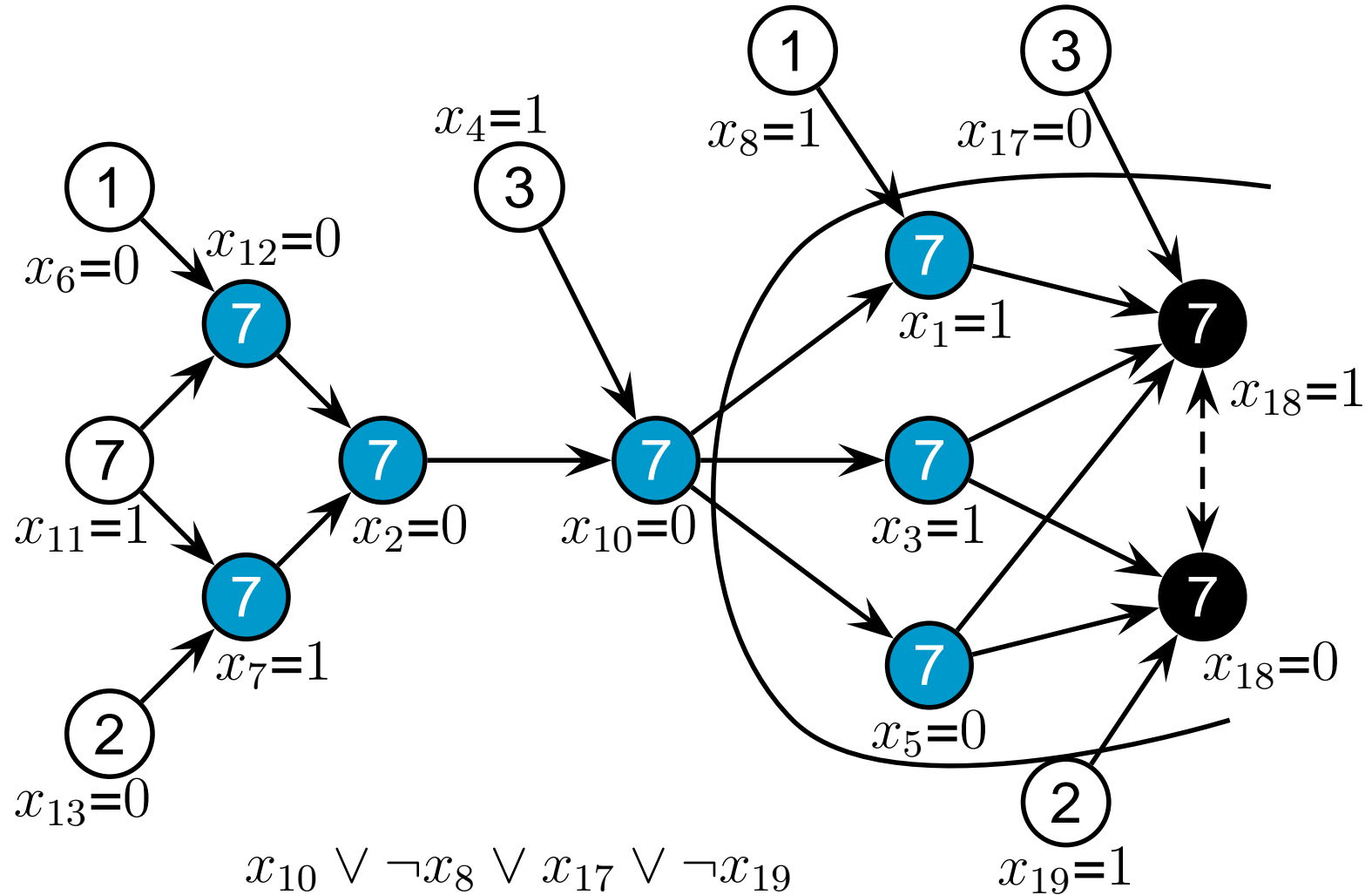
Conflict-driven: Implication graph



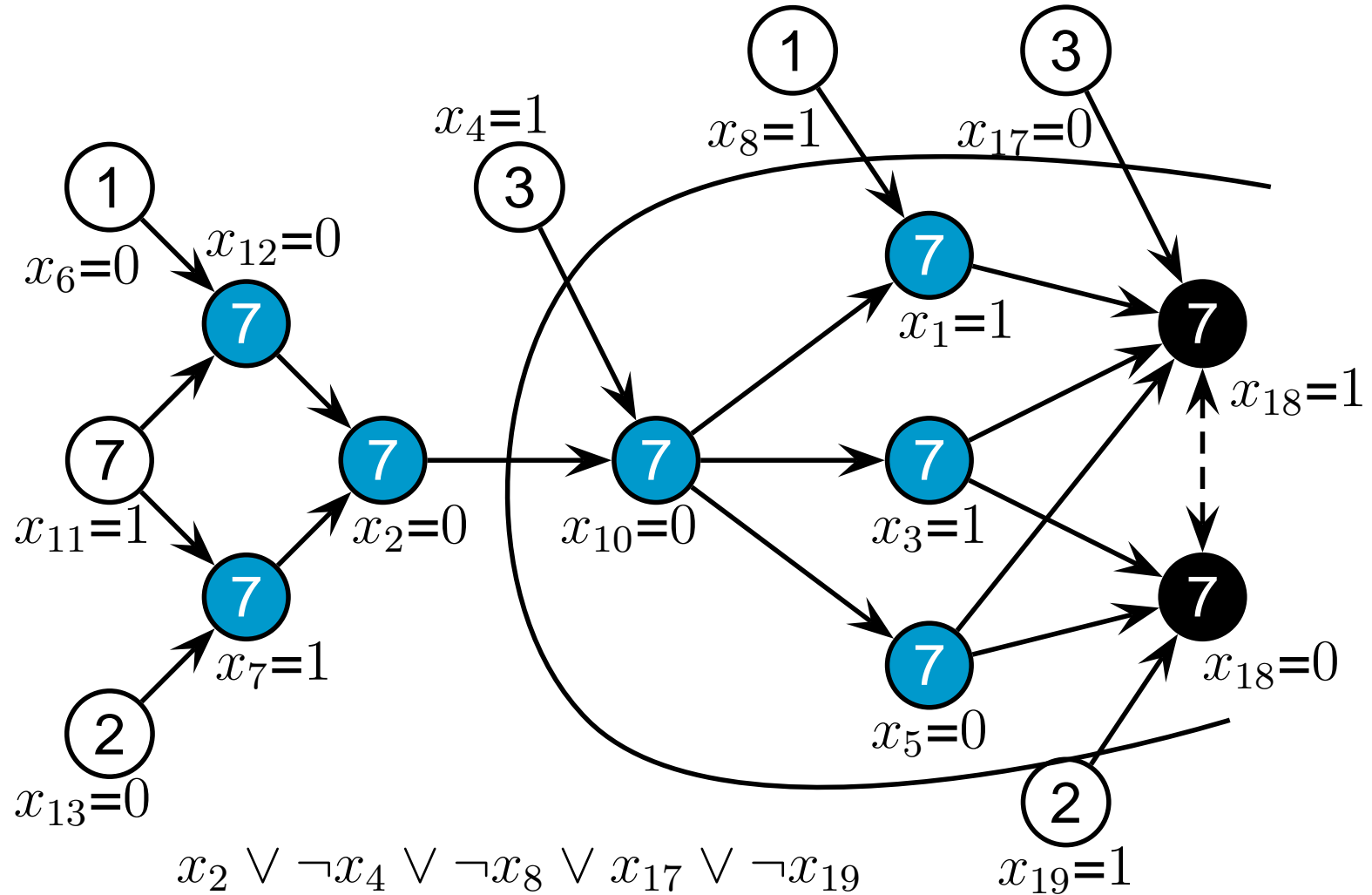
Conflict-driven: Implication graph



Conflict-driven: Implication graph



Conflict-driven: Implication graph



Conflict-driven: Watch pointers (1)

$$\varphi = \{x_1 = *, x_2 = *, x_3 = *, x_4 = *, x_5 = *, x_6 = *\}$$

$\neg x_1$	x_2	$\neg x_3$	$\neg x_5$	x_6
------------	-------	------------	------------	-------

x_1	$\neg x_3$	x_4	$\neg x_5$	$\neg x_6$
-------	------------	-------	------------	------------

Conflict-driven: Watch pointers (1)

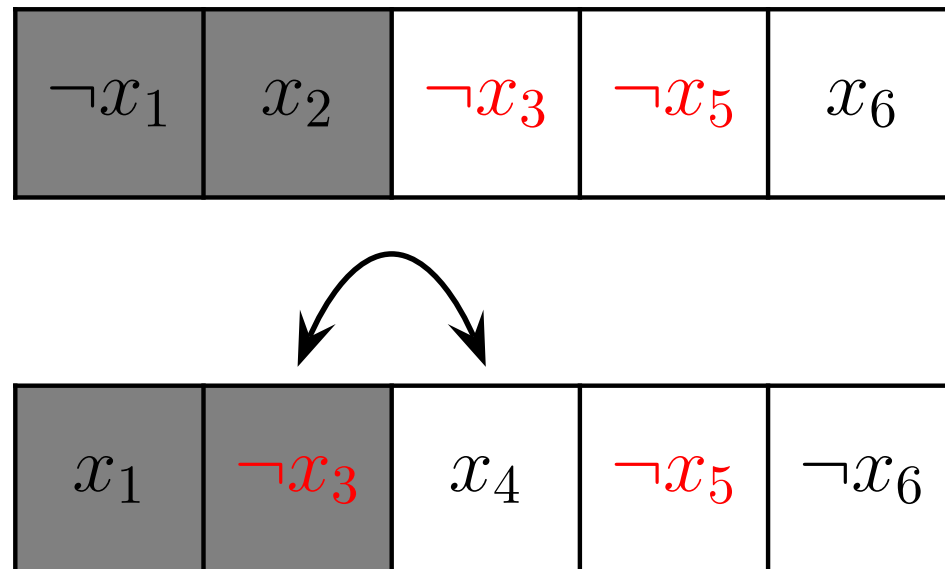
$$\varphi = \{x_1 = *, x_2 = *, x_3 = *, x_4 = *, x_5 = \mathbf{1}, x_6 = *\}$$

$\neg x_1$	x_2	$\neg x_3$	$\neg x_5$	x_6
------------	-------	------------	------------	-------

x_1	$\neg x_3$	x_4	$\neg x_5$	$\neg x_6$
-------	------------	-------	------------	------------

Conflict-driven: Watch pointers (1)

$$\varphi = \{x_1 = *, x_2 = *, x_3 = \mathbf{1}, x_4 = *, x_5 = 1, x_6 = *\}$$



Conflict-driven: Watch pointers (1)

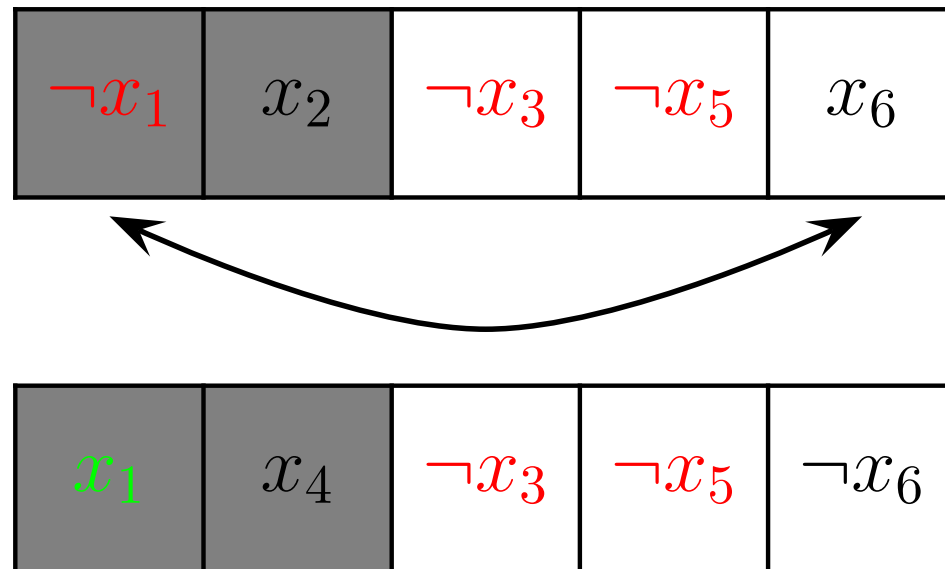
$$\varphi = \{x_1 = *, x_2 = *, x_3 = 1, x_4 = *, x_5 = 1, x_6 = *\}$$

$\neg x_1$	x_2	$\neg x_3$	$\neg x_5$	x_6
------------	-------	------------	------------	-------

x_1	x_4	$\neg x_3$	$\neg x_5$	$\neg x_6$
-------	-------	------------	------------	------------

Conflict-driven: Watch pointers (1)

$$\varphi = \{x_1 = \mathbf{1}, x_2 = *, x_3 = 1, x_4 = *, x_5 = 1, x_6 = *\}$$



Conflict-driven: Watch pointers (1)

$$\varphi = \{x_1 = 1, x_2 = *, x_3 = 1, x_4 = *, x_5 = 1, x_6 = *\}$$

x_6	x_2	$\neg x_3$	$\neg x_5$	$\neg x_1$
-------	-------	------------	------------	------------

x_1	x_4	$\neg x_3$	$\neg x_5$	$\neg x_6$
-------	-------	------------	------------	------------

Conflict-driven: Watch pointers (1)

$$\varphi = \{x_1 = 1, x_2 = *, x_3 = 1, x_4 = \mathbf{0}, x_5 = 1, x_6 = *\}$$

x_6	x_2	$\neg x_3$	$\neg x_5$	$\neg x_1$
-------	-------	------------	------------	------------

x_1	x_4	$\neg x_3$	$\neg x_5$	$\neg x_6$
-------	-------	------------	------------	------------

Conflict-driven: Watch pointers (1)

$$\varphi = \{x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 1, x_6 = *\}$$

x_6	x_2	$\neg x_3$	$\neg x_5$	$\neg x_1$
-------	-------	------------	------------	------------

x_1	x_4	$\neg x_3$	$\neg x_5$	$\neg x_6$
-------	-------	------------	------------	------------

Conflict-driven: Watch pointers (1)

$$\varphi = \{x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 1, x_6 = 1\}$$

x_6	x_2	$\neg x_3$	$\neg x_5$	$\neg x_1$
-------	-------	------------	------------	------------

x_1	x_4	$\neg x_3$	$\neg x_5$	$\neg x_6$
-------	-------	------------	------------	------------

Conflict-driven: Watch pointers (1)

$$\varphi = \{x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 1, x_6 = 1\}$$

x_6	x_2	$\neg x_3$	$\neg x_5$	$\neg x_1$
-------	-------	------------	------------	------------

x_1	x_4	$\neg x_3$	$\neg x_5$	$\neg x_6$
-------	-------	------------	------------	------------

Conflict-driven: Watch pointers (2)

- Only examine a clause when both
 - A watch pointer gets falsified
 - The other one is not satisfied

Conflict-driven: Watch pointers (2)

- Only examine a clause when both
 - A watch pointer gets falsified
 - The other one is not satisfied
- While backjumping, just unassign variables

Conflict-driven: Watch pointers (2)

- Only examine a clause when both
 - A watch pointer gets falsified
 - The other one is not satisfied
- While backjumping, just unassign variables
- Conflict clauses \rightarrow watch pointers

Conflict-driven: Watch pointers (2)

- Only examine a clause when both
 - A watch pointer gets falsified
 - The other one is not satisfied
- While backjumping, just unassign variables
- Conflict clauses \rightarrow watch pointers
- No detailed information available

Conflict-driven: Watch pointers (2)

- Only examine a clause when both
 - A watch pointer gets falsified
 - The other one is not satisfied
- While backjumping, just unassign variables
- Conflict clauses → watch pointers
- No detailed information available
- Not used for binary clauses

Conflict-driven: Heuristics & Restarts

Heuristics:

- Select a variable occurring in recent conflicts

Conflict-driven: Heuristics & Restarts

Heuristics:

- Select a variable occurring in recent conflicts
- Negative branching (minisat)

Conflict-driven: Heuristics & Restarts

Heuristics:

- Select a variable occurring in recent conflicts
- Negative branching (minisat)
- Phase saving (rsat, picosat)

Conflict-driven: Heuristics & Restarts

Heuristics:

- Select a variable occurring in recent conflicts
- Negative branching (minisat)
- Phase saving (rsat, picosat)

Restart strategies

- Incremental restarts (minisat)

Conflict-driven: Heuristics & Restarts

Heuristics:

- Select a variable occurring in recent conflicts
- Negative branching (minisat)
- Phase saving (rsat, picosat)

Restart strategies

- Incremental restarts (minisat)
- Rapid restarts (rsat, picosat)

Conflict-driven: Heuristics & Restarts

Heuristics:

- Select a variable occurring in recent conflicts
- Negative branching (minisat)
- Phase saving (rsat, picosat)

Restart strategies

- Incremental restarts (minisat)
- Rapid restarts (rsat, picosat)
- While restarting, remove least used conflict clauses from database

Conflict-driven: Pseudo code

```
1: while TRUE do  
2:    $l_{\text{decision}} := \text{GETDECISIONLITERAL}()$   
3:   if no  $l_{\text{decision}}$  then return satisfiable  
4:    $\mathcal{F} := \text{SIMPLIFY}(\mathcal{F}(l_{\text{decision}} \leftarrow 1))$   
5:   while empty clause  $\emptyset \in \mathcal{F}$  do  
6:      $C_{\text{conflict}} := \text{ANALYZECONFLICTS}()$   
7:     if  $C_{\text{conflict}} = \emptyset$  then return unsatisfiable  
8:      $\text{BACKTRACK}(C_{\text{conflict}})$   
9:      $\mathcal{F} := \text{SIMPLIFY}(\mathcal{F} \cup C_{\text{conflict}})$   
10:  end while  
11: end while
```

Computational Logic and Satisfiability

IN4077

Conflict-driven SAT solvers

Marijn J.H. Heule

September 23th 2008