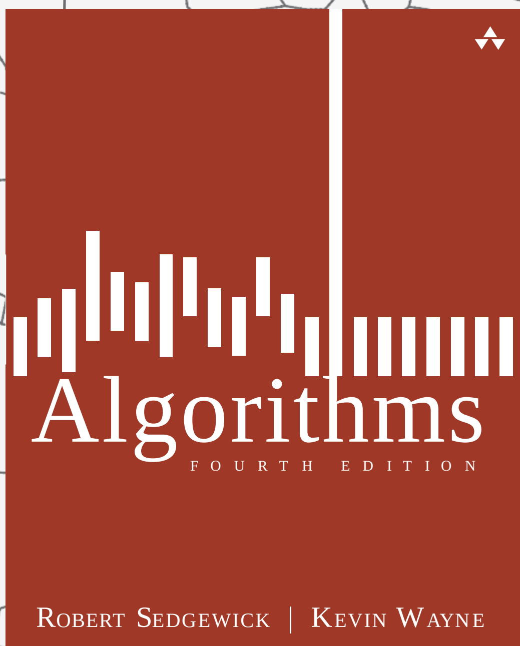
Reiknirit

Magnús M. Halldórsson



http://algs4.cs.princeton.edu

Satisfiability : DPLL heuristic and 2-SAT

- *DPLL heuristic*
- *2-SAT*

- *Dynamic programming*

Objectives for this week

What else should a computer scientist know about algorithms?

- For many, this may be the last course on algorithms and complexity

Next lecture: The limits of computing

- There are things that computers will not be able to do
- The most important are the „NP-complete“ problems

This lecture: Heuristics

- When we don‘t have algorithms with nice time complexity
- The „science of brute force“
- Recent progress, extremely successful

Common intersection: The Satisfiability Problem (SAT)

- Hugely practical „meta-problem“
- Fundamental theoretical importance

Today's lecture

Introduce the Satisfiability Problem (SAT)

- Applications in AI, planning, circuit design, verifiable programs
- Boolean formulas in CNF (conjunctive normal form)

Heuristic search algorithms

- Methods with poor worst case behavior, that often work well.
- Generation of all bitstrings
- Backtracking: Examining no more than necessary

Satisfiability solutions method

- Backtracking
- Unit clause propagation and DPLL

Algorithms

Robert Sedgewick | Kevin Wayne

http://algs4.cs.princeton.edu

Satisfiability and NP-completeness

- *Backtracking*
- *DPLL*

Davis-Putnam-Logeman-Loveland (DPLL) procedure

Transformations that preserve satisfiability

- Change formula φ to a *simpler* formula φ'
- φ is satisfiable iff φ' is satisfiable

Simplifying transformations:

1. Unit clause propagation
2. Pure literal elimination

Davis-Putnam-Logeman-Loveland (DPLL) procedure

Transformations that preserve satisfiability

- Change formula φ to a *simpler* formula φ'
- φ is satisfiable iff φ' is satisfiable

Unit clause propagation

- Suppose there is a clause consisting of a single literal
- Example: $(x_1) \wedge (\neg x_1 \vee x_2)$
- We can replace that literal by the necessary truth value

Power of the method: Recursive application

- Suppose there is a clause consisting of a single literal
- Example:
  $(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_1 \vee x_2) \wedge (\neg x_2 \vee \neg x_3) \wedge (x_1 \vee x_2 \vee x_5)$
  $\wedge (\neg x_1 \vee x_2) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_4 \vee \neg x_5) \wedge (x_5)$

Davis-Putnam-Logeman-Loveland (DPLL) procedure

Transformations that preserve satisfiability

- Change formula φ to a *simpler* formula φ'
- φ is satisfiable iff φ' is satisfiable

Pure-literal elimination

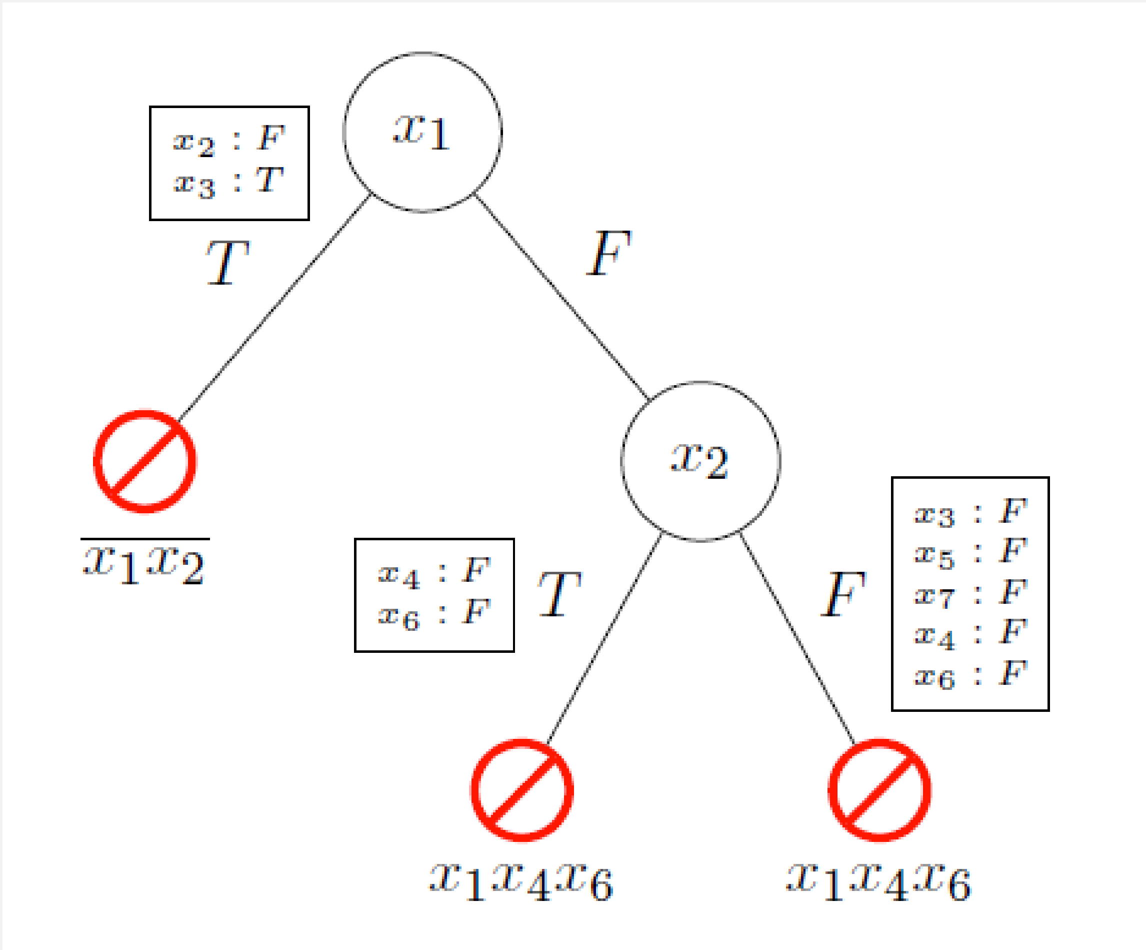
- Suppose a variable appears only negated (or only unnegated).
- It is then safe to assign it $False$ ($True$)
- Example:
  $(x_1 \lor \neg x_2 \lor \neg x_3) \land (x_1 \lor x_2) \land (\neg x_2 \lor \neg x_3) \land (x_1 \lor x_2 \lor x_5)$
  $\land (\neg x_1 \lor x_2) \land (\neg x_3 \lor \neg x_4) \land (x_4 \lor \neg x_5) \land (x_1 \lor x_4 \lor x_5)$

Example

$$\mathcal{F} : \overline{x_1},\ x_1\overline{x_2},\ x_1x_2x_3,\ \ \overline{x_3}x_4,\ x_1\overline{x_3x_4}$$

Example

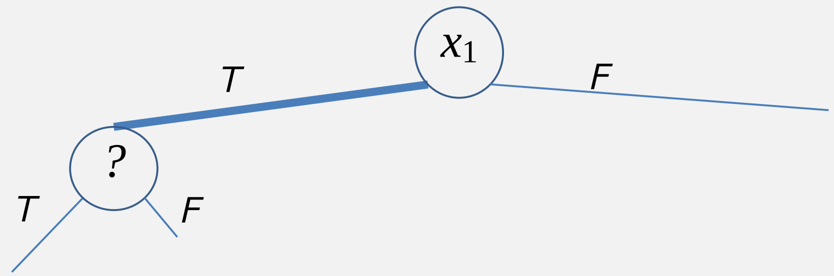
$\mathcal{F} : \overline{x_1},\ x_1\overline{x_2},\ x_1x_2x_3,\ \overline{x_3}x_4,\ x_1\overline{x_3x_4}$



Applying unit-clause propagation

$x_1 \lor x_2 \lor x_3$
$\neg x_1 \lor \neg x_2$
$\neg x_1 \lor \neg x_3$
$\neg x_2 \lor \neg x_3$
$x_4 \lor x_5 \lor x_6$
$\neg x_4 \lor \neg x_5$
$\neg x_4 \lor \neg x_6$
$\neg x_5 \lor \neg x_6$
$x_7 \lor x_8 \lor x_9$
$\neg x_7 \lor \neg x_8$
$\neg x_7 \lor \neg x_9$
$\neg x_8 \lor \neg x_9$
$x_{10} \lor x_{11} \lor x_{12}$
$\neg x_{10} \lor \neg x_{11}$
$\neg x_{10} \lor \neg x_{12}$
$\neg x_{11} \lor \neg x_{12}$
$\neg x_8 \lor \neg x_{11}$

$\neg x_1 \lor \neg x_4$
$\neg x_2 \lor \neg x_5$
$\neg x_3 \lor \neg x_6$
$\neg x_1 \lor \neg x_7$
$\neg x_2 \lor \neg x_8$
$\neg x_3 \lor \neg x_9$
$\neg x_1 \lor \neg x_{10}$
$\neg x_2 \lor \neg x_{11}$
$\neg x_3 \lor \neg x_{12}$
$\neg x_4 \lor \neg x_7$
$\neg x_5 \lor \neg x_8$
$\neg x_6 \lor \neg x_9$
$\neg x_4 \lor \neg x_{10}$
$\neg x_5 \lor \neg x_{11}$
$\neg x_6 \lor \neg x_{12}$
$\neg x_7 \lor \neg x_{10}$
$\neg x_9 \lor \neg x_{12}$

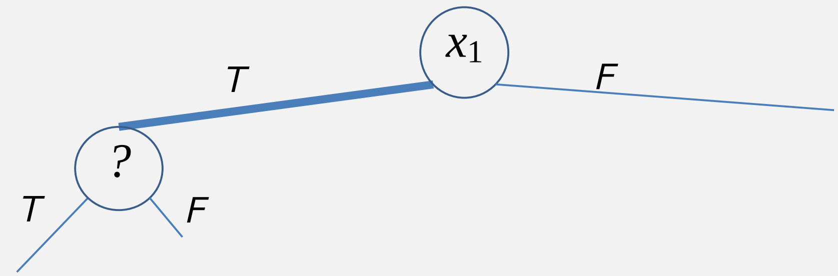


In this example, we consider the variables in the natural order, starting with x_1 .
We start with exploring the case with x_1 true.

Applying unit-clause propagation

$\cancel{x_1 \lor x_2 \lor x_3}$
$\cancel{\neg x_1} \lor \neg x_2$
$\neg x_1 \lor \neg x_3$
$\neg x_2 \lor \neg x_3$
$x_4 \lor x_5 \lor x_6$
$\neg x_4 \lor \neg x_5$
$\neg x_4 \lor \neg x_6$
$\neg x_5 \lor \neg x_6$
$x_7 \lor x_8 \lor x_9$
$\neg x_7 \lor \neg x_8$
$\neg x_7 \lor \neg x_9$
$\neg x_8 \lor \neg x_9$
$x_{10} \lor x_{11} \lor x_{12}$
$\neg x_{10} \lor \neg x_{11}$
$\neg x_{10} \lor \neg x_{12}$
$\neg x_{11} \lor \neg x_{12}$
$\neg x_8 \lor \neg x_{11}$

$\neg x_1 \lor \neg x_4$
$\neg x_2 \lor \neg x_5$
$\neg x_3 \lor \neg x_6$
$\neg x_1 \lor \neg x_7$
$\neg x_2 \lor \neg x_8$
$\neg x_3 \lor \neg x_9$
$\neg x_1 \lor \neg x_{10}$
$\neg x_2 \lor \neg x_{11}$
$\neg x_3 \lor \neg x_{12}$
$\neg x_4 \lor \neg x_7$
$\neg x_5 \lor \neg x_8$
$\neg x_6 \lor \neg x_9$
$\neg x_4 \lor \neg x_{10}$
$\neg x_5 \lor \neg x_{11}$
$\neg x_6 \lor \neg x_{12}$
$\neg x_7 \lor \neg x_{10}$
$\neg x_9 \lor \neg x_{12}$



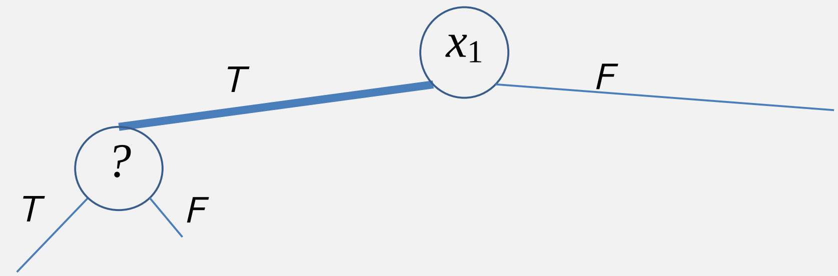
When x_1 receives the value T (= true), then x_2 must necessarily receive the value F(= false), because of the clause $\neg x_1 \lor \neg x_2$.

Clause already satisfied:
Literal that is false

Applying unit-clause propagation

$\cancel{x_1 \vee x_2 \vee x_3}$
$\cancel{\neg x_1} \vee \neg x_2$
$\cancel{\neg x_1} \vee \neg x_3$
$\neg x_2 \vee \neg x_3$
$x_4 \vee x_5 \vee x_6$
$\neg x_4 \vee \neg x_5$
$\neg x_4 \vee \neg x_6$
$\neg x_5 \vee \neg x_6$
$x_7 \vee x_8 \vee x_9$
$\neg x_7 \vee \neg x_8$
$\neg x_7 \vee \neg x_9$
$\neg x_8 \vee \neg x_9$
$x_{10} \vee x_{11} \vee x_{12}$
$\neg x_{10} \vee \neg x_{11}$
$\neg x_{10} \vee \neg x_{12}$
$\neg x_{11} \vee \neg x_{12}$
$\neg x_8 \vee \neg x_{11}$

$\cancel{\neg x_1} \vee \neg x_4$
$\neg x_2 \vee \neg x_5$
$\neg x_3 \vee \neg x_6$
$\cancel{\neg x_1} \vee \neg x_7$
$\neg x_2 \vee \neg x_8$
$\neg x_3 \vee \neg x_9$
$\cancel{\neg x_1} \vee \neg x_{10}$
$\neg x_2 \vee \neg x_{11}$
$\neg x_3 \vee \neg x_{12}$
$\neg x_4 \vee \neg x_7$
$\neg x_5 \vee \neg x_8$
$\neg x_6 \vee \neg x_9$
$\neg x_4 \vee \neg x_{10}$
$\neg x_5 \vee \neg x_{11}$
$\neg x_6 \vee \neg x_{12}$
$\neg x_7 \vee \neg x_{10}$
$\neg x_9 \vee \neg x_{12}$



When x_1 receives the value T (= true), then x_2 must necessarily receive the value F(= false), because of the clause $\neg x_1 \vee \neg x_2$.

Similarly x_3, x_4, x_7, x_{10} must also be false.

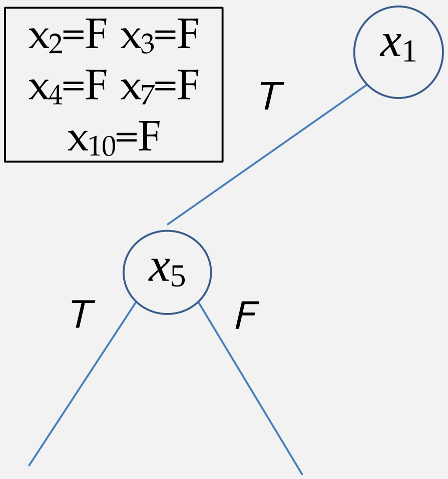
We label the edge from x_1 to the left with the variables that become assigned because of unit clause propagation.

DPLL

$\cancel{x_1 \vee x_2 \vee x_3}$
$\cancel{\neg x_1 \vee \neg x_2}$
$\cancel{\neg x_1 \vee \neg x_3}$
$\cancel{\neg x_2 \vee \neg x_3}$
$x_4 \vee x_5 \vee x_6$
$\cancel{\neg x_4 \vee \neg x_5}$
$\cancel{\neg x_4 \vee \neg x_6}$
$\neg x_5 \vee \neg x_6$
$x_7 \vee x_8 \vee x_9$
$\cancel{\neg x_7 \vee \neg x_8}$
$\cancel{\neg x_7 \vee \neg x_9}$
$\neg x_8 \vee \neg x_9$
$x_{10} \vee x_{11} \vee x_{12}$
$\cancel{\neg x_{10} \vee \neg x_{11}}$
$\cancel{\neg x_{10} \vee \neg x_{12}}$
$\neg x_{11} \vee \neg x_{12}$
$\neg x_8 \vee \neg x_{11}$

$\cancel{\neg x_1 \vee \neg x_4}$
$\cancel{\neg x_2 \vee \neg x_5}$
$\cancel{\neg x_3 \vee \neg x_6}$
$\cancel{\neg x_1 \vee \neg x_7}$
$\cancel{\neg x_2 \vee \neg x_8}$
$\cancel{\neg x_3 \vee \neg x_9}$
$\cancel{\neg x_1 \vee \neg x_{10}}$
$\cancel{\neg x_2 \vee \neg x_{11}}$
$\cancel{\neg x_3 \vee \neg x_{12}}$
$\cancel{\neg x_4 \vee \neg x_7}$
$\neg x_5 \vee \neg x_8$
$\neg x_6 \vee \neg x_9$
$\cancel{\neg x_4 \vee \neg x_{10}}$
$\neg x_5 \vee \neg x_{11}$
$\neg x_6 \vee \neg x_{12}$
$\cancel{\neg x_7 \vee \neg x_{10}}$
$\neg x_9 \vee \neg x_{12}$

x_2=F x_3=F
x_4=F x_7=F
x_{10}=F



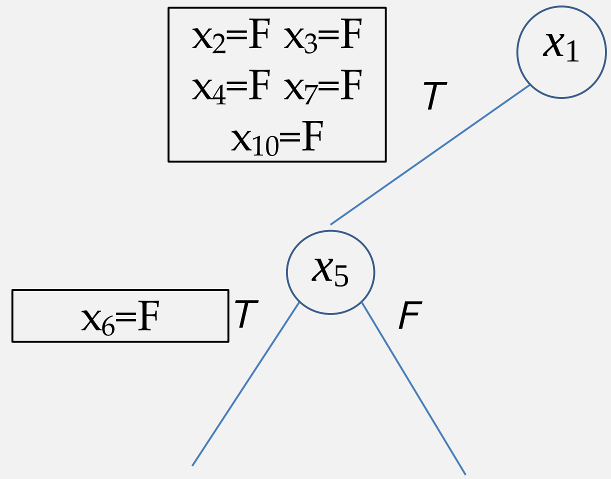
We label the edge from x_1 to the left with the variables that become assigned because of unit clause propagation.

Clause already satisfied:
Literal that is false

DPLL

$\cancel{x_1 \vee x_2 \vee x_3}$
$\cancel{\neg x_1 \vee \neg x_2}$
$\cancel{\cancel{\neg x_1} \vee \neg x_3}$
$\cancel{\neg x_2 \vee \neg x_3}$
$\cancel{x_4} \vee x_5 \vee x_6$
$\cancel{\neg x_4 \vee \neg x_5}$
$\cancel{\neg x_4 \vee \neg x_6}$
$\cancel{\neg x_5} \vee \neg x_6$
$\cancel{x_7} \vee x_8 \vee x_9$
$\cancel{\neg x_7 \vee \neg x_8}$
$\cancel{\neg x_7 \vee \neg x_9}$
$\neg x_8 \vee \neg x_9$
$\cancel{x_{10}} \vee x_{11} \vee x_{12}$
$\cancel{\neg x_{10} \vee \neg x_{11}}$
$\cancel{\neg x_{10} \vee \neg x_{12}}$
$\neg x_{11} \vee \neg x_{12}$
$\neg x_8 \vee \neg x_{11}$

$\cancel{\neg x_1 \vee \neg x_4}$
$\cancel{\neg x_2 \vee \neg x_5}$
$\cancel{\neg x_3 \vee \neg x_6}$
$\cancel{\neg x_1 \vee \neg x_7}$
$\cancel{\neg x_2 \vee \neg x_8}$
$\cancel{\neg x_3 \vee \neg x_9}$
$\cancel{\neg x_1 \vee \neg x_{10}}$
$\cancel{\neg x_2 \vee \neg x_{11}}$
$\cancel{\neg x_3 \vee \neg x_{12}}$
$\cancel{\neg x_4 \vee \neg x_7}$
$\cancel{\neg x_5} \vee \neg x_8$
$\neg x_6 \vee \neg x_9$
$\cancel{\neg x_4 \vee \neg x_{10}}$
$\cancel{\neg x_5} \vee \neg x_{11}$
$\neg x_6 \vee \neg x_{12}$
$\cancel{\neg x_7 \vee \neg x_{10}}$
$\neg x_9 \vee \neg x_{12}$

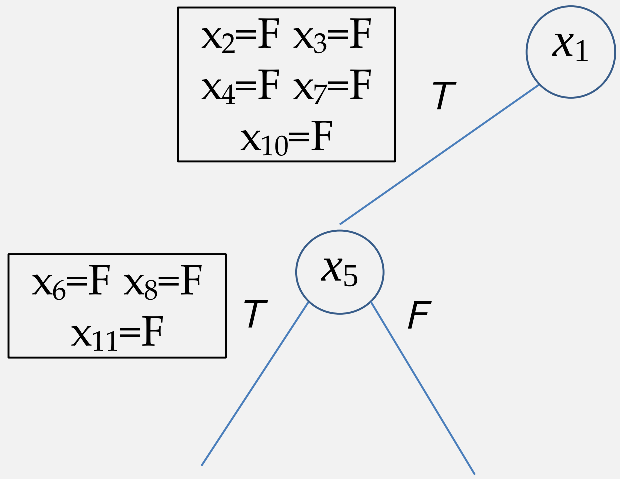


When x_5 is assigned T (=true), then x_6 must be assigned F (= false), because of the clause $\neg x_5 \vee \neg x_6$.

DPLL

$\cancel{x_1 \lor x_2 \lor x_3}$
$\cancel{\neg x_1 \lor \neg x_2}$
$\cancel{\neg x_1 \lor \neg x_3}$
$\cancel{\neg x_2 \lor \neg x_3}$
$x_4 \lor x_5 \lor x_6$
$\cancel{\neg x_4 \lor \neg x_5}$
$\cancel{\neg x_4 \lor \neg x_6}$
$\neg x_5 \lor \neg x_6$
$x_7 \lor x_8 \lor x_9$
$\cancel{\neg x_7 \lor \neg x_8}$
$\cancel{\neg x_7 \lor \neg x_9}$
$\neg x_8 \lor \neg x_9$
$x_{10} \lor x_{11} \lor x_{12}$
$\cancel{\neg x_{10} \lor \neg x_{11}}$
$\cancel{\neg x_{10} \lor \neg x_{12}}$
$\neg x_{11} \lor \neg x_{12}$
$\neg x_8 \lor \neg x_{11}$

$\cancel{\neg x_1 \lor \neg x_4}$
$\cancel{\neg x_2 \lor \neg x_5}$
$\cancel{\neg x_3 \lor \neg x_6}$
$\cancel{\neg x_1 \lor \neg x_7}$
$\cancel{\neg x_2 \lor \neg x_8}$
$\cancel{\neg x_3 \lor \neg x_9}$
$\cancel{\neg x_1 \lor \neg x_{10}}$
$\cancel{\neg x_2 \lor \neg x_{11}}$
$\cancel{\neg x_3 \lor \neg x_{12}}$
$\cancel{\neg x_4 \lor \neg x_7}$
$\neg x_5 \lor \neg x_8$
$\neg x_6 \lor \neg x_9$
$\cancel{\neg x_4 \lor \neg x_{10}}$
$\neg x_5 \lor \neg x_{11}$
$\neg x_6 \lor \neg x_{12}$
$\cancel{\neg x_7 \lor \neg x_{10}}$
$\neg x_9 \lor \neg x_{12}$

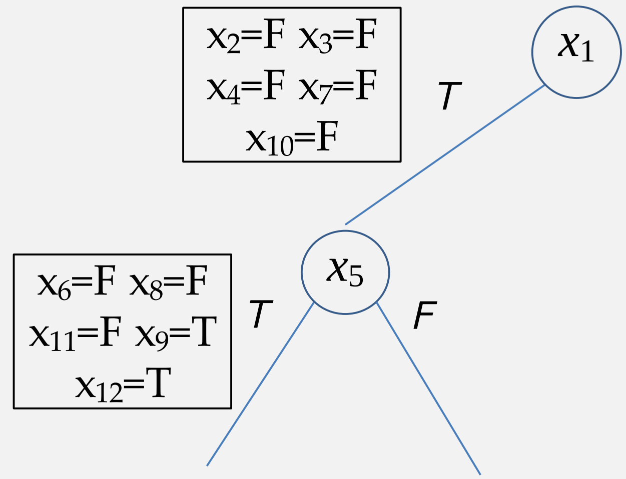


When x_5 is assigned T (=true), then x_6 must be assigned F (= false), because of the clause $\neg x_5 \lor \neg x_6$.
Same for x_8, x_{11}.

DPLL

$x_1 \vee x_2 \vee x_3$
$\neg x_1 \vee \neg x_2$
$\neg x_1 \vee \neg x_3$
$\neg x_2 \vee \neg x_3$
$x_4 \vee x_5 \vee x_6$
$\neg x_4 \vee \neg x_5$
$\neg x_4 \vee \neg x_6$
$\neg x_5 \vee \neg x_6$
$x_7 \vee x_8 \vee x_9$
$\neg x_7 \vee \neg x_8$
$\neg x_7 \vee \neg x_9$
$\neg x_8 \vee \neg x_9$
$x_{10} \vee x_{11} \vee x_{12}$
$\neg x_{10} \vee \neg x_{11}$
$\neg x_{10} \vee \neg x_{12}$
$\neg x_{11} \vee \neg x_{12}$
$\neg x_8 \vee \neg x_{11}$

$\neg x_1 \vee \neg x_4$
$\neg x_2 \vee \neg x_5$
$\neg x_3 \vee \neg x_6$
$\neg x_1 \vee \neg x_7$
$\neg x_2 \vee \neg x_8$
$\neg x_3 \vee \neg x_9$
$\neg x_1 \vee \neg x_{10}$
$\neg x_2 \vee \neg x_{11}$
$\neg x_3 \vee \neg x_{12}$
$\neg x_4 \vee \neg x_7$
$\neg x_5 \vee \neg x_8$
$\neg x_6 \vee \neg x_9$
$\neg x_4 \vee \neg x_{10}$
$\neg x_5 \vee \neg x_{11}$
$\neg x_6 \vee \neg x_{12}$
$\neg x_7 \vee \neg x_{10}$
$\neg x_9 \vee \neg x_{12}$

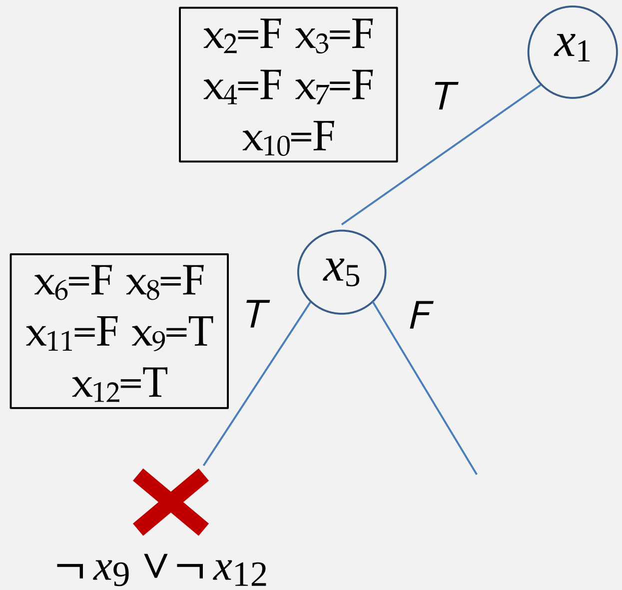


When x_5 is assigned T (=true), then x_6 must be assigned F (= false), because of the clause $\neg x_5 \vee \neg x_6$.
Same for x_8, x_{11}.
Then, x_9, x_{12} must become T (because of $x_7 \vee x_8 \vee x_9$ and $x_{10} \vee x_{11} \vee x_{12}$).

DPLL

$x_1 \vee x_2 \vee x_3$
$\neg x_1 \vee \neg x_2$
$\neg x_1 \vee \neg x_3$
$\neg x_2 \vee \neg x_3$
$x_4 \vee x_5 \vee x_6$
$\neg x_4 \vee \neg x_5$
$\neg x_4 \vee \neg x_6$
$\neg x_5 \vee \neg x_6$
$x_7 \vee x_8 \vee x_9$
$\neg x_7 \vee \neg x_8$
$\neg x_7 \vee \neg x_9$
$\neg x_8 \vee \neg x_9$
$x_{10} \vee x_{11} \vee x_{12}$
$\neg x_{10} \vee \neg x_{11}$
$\neg x_{10} \vee \neg x_{12}$
$\neg x_{11} \vee \neg x_{12}$
$\neg x_8 \vee \neg x_{11}$

$\neg x_1 \vee \neg x_4$
$\neg x_2 \vee \neg x_5$
$\neg x_3 \vee \neg x_6$
$\neg x_1 \vee \neg x_7$
$\neg x_2 \vee \neg x_8$
$\neg x_3 \vee \neg x_9$
$\neg x_1 \vee \neg x_{10}$
$\neg x_2 \vee \neg x_{11}$
$\neg x_3 \vee \neg x_{12}$
$\neg x_4 \vee \neg x_7$
$\neg x_5 \vee \neg x_8$
$\neg x_6 \vee \neg x_9$
$\neg x_4 \vee \neg x_{10}$
$\neg x_5 \vee \neg x_{11}$
$\neg x_6 \vee \neg x_{12}$
$\neg x_7 \vee \neg x_{10}$
$\neg x_9 \vee \neg x_{12}$

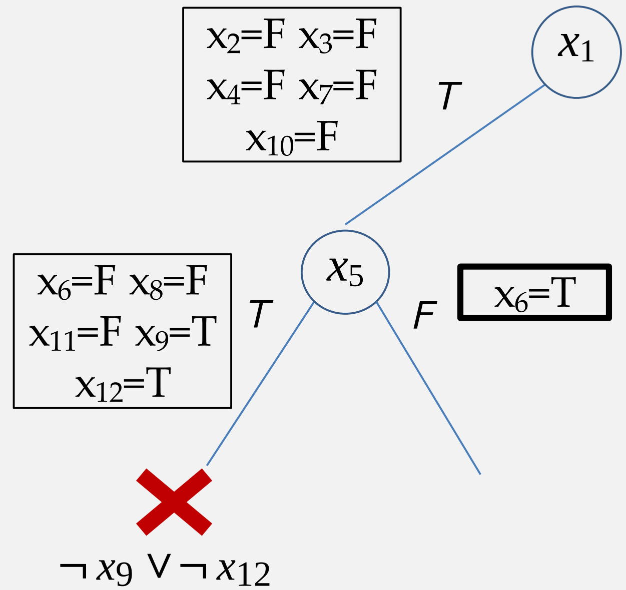


When x_5 is assigned T (=true), then x_6 must be assigned F (= false), because of the clause $\neg x_5 \vee \neg x_6$.
Same for x_8, x_{11}.
Then, x_9, x_{12} must become T (because of $x_7 \vee x_8 \vee x_9$ and $x_{10} \vee x_{11} \vee x_{12}$).
That causes conflict in the clause $\neg x_9 \vee \neg x_{12}$.
Hence, this branch is a dead-end, and shown as red X.

DPLL

~~$x_1 \lor x_2 \lor x_3$~~
~~$\neg x_1 \lor \neg x_2$~~
~~$\neg x_1 \lor \neg x_3$~~
~~$\neg x_2 \lor \neg x_3$~~
$x_4 \lor x_5 \lor x_6$
~~$\neg x_4 \lor \neg x_5$~~
~~$\neg x_4 \lor \neg x_6$~~
~~$\neg x_5 \lor \neg x_6$~~
$x_7 \lor x_8 \lor x_9$
~~$\neg x_7 \lor \neg x_8$~~
~~$\neg x_7 \lor \neg x_9$~~
$\neg x_8 \lor \neg x_9$
$x_{10} \lor x_{11} \lor x_{12}$
~~$\neg x_{10} \lor \neg x_{11}$~~
~~$\neg x_{10} \lor \neg x_{12}$~~
$\neg x_{11} \lor \neg x_{12}$
$\neg x_8 \lor \neg x_{11}$

~~$\neg x_1 \lor \neg x_4$~~
~~$\neg x_2 \lor \neg x_5$~~
~~$\neg x_3 \lor \neg x_6$~~
~~$\neg x_1 \lor \neg x_7$~~
~~$\neg x_2 \lor \neg x_8$~~
~~$\neg x_3 \lor \neg x_9$~~
~~$\neg x_1 \lor \neg x_{10}$~~
~~$\neg x_2 \lor \neg x_{11}$~~
~~$\neg x_3 \lor \neg x_{12}$~~
~~$\neg x_4 \lor \neg x_7$~~
~~$\neg x_5 \lor \neg x_8$~~
$\neg x_6 \lor \neg x_9$
~~$\neg x_4 \lor \neg x_{10}$~~
~~$\neg x_5 \lor \neg x_{11}$~~
$\neg x_6 \lor \neg x_{12}$
~~$\neg x_7 \lor \neg x_{10}$~~
$\neg x_9 \lor \neg x_{12}$

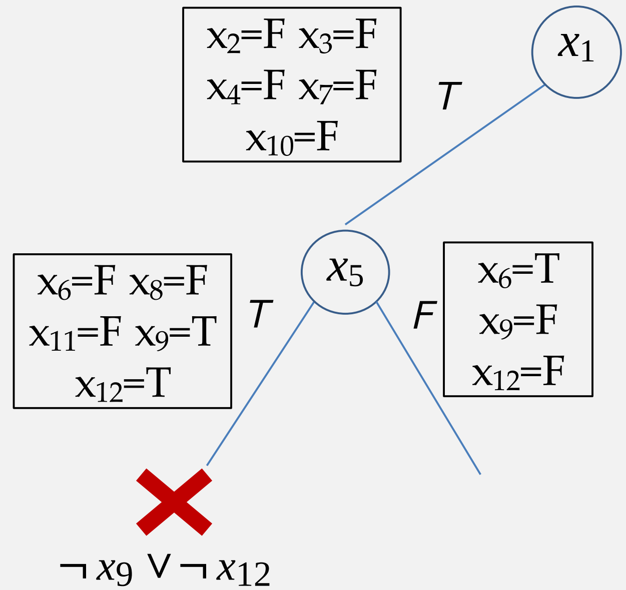


When x_5 is assigned F (=false), then x_6 must be assigned T (= true), because of the clause $x_4 \lor x_5 \lor x_6$.

DPLL

$x_1 \vee x_2 \vee x_3$
$\neg x_1 \vee \neg x_2$
$\neg x_1 \vee \neg x_3$
$\neg x_2 \vee \neg x_3$
$x_4 \vee x_5 \vee x_6$
$\neg x_4 \vee \neg x_5$
$\neg x_4 \vee \neg x_6$
$\neg x_5 \vee \neg x_6$
$x_7 \vee x_8 \vee x_9$
$\neg x_7 \vee \neg x_8$
$\neg x_7 \vee \neg x_9$
$\neg x_8 \vee \neg x_9$
$x_{10} \vee x_{11} \vee x_{12}$
$\neg x_{10} \vee \neg x_{11}$
$\neg x_{10} \vee \neg x_{12}$
$\neg x_{11} \vee \neg x_{12}$
$\neg x_8 \vee \neg x_{11}$

$\neg x_1 \vee \neg x_4$
$\neg x_2 \vee \neg x_5$
$\neg x_3 \vee \neg x_6$
$\neg x_1 \vee \neg x_7$
$\neg x_2 \vee \neg x_8$
$\neg x_3 \vee \neg x_9$
$\neg x_1 \vee \neg x_{10}$
$\neg x_2 \vee \neg x_{11}$
$\neg x_3 \vee \neg x_{12}$
$\neg x_4 \vee \neg x_7$
$\neg x_5 \vee \neg x_8$
$\neg x_6 \vee \neg x_9$
$\neg x_4 \vee \neg x_{10}$
$\neg x_5 \vee \neg x_{11}$
$\neg x_6 \vee \neg x_{12}$
$\neg x_7 \vee \neg x_{10}$
$\neg x_9 \vee \neg x_{12}$

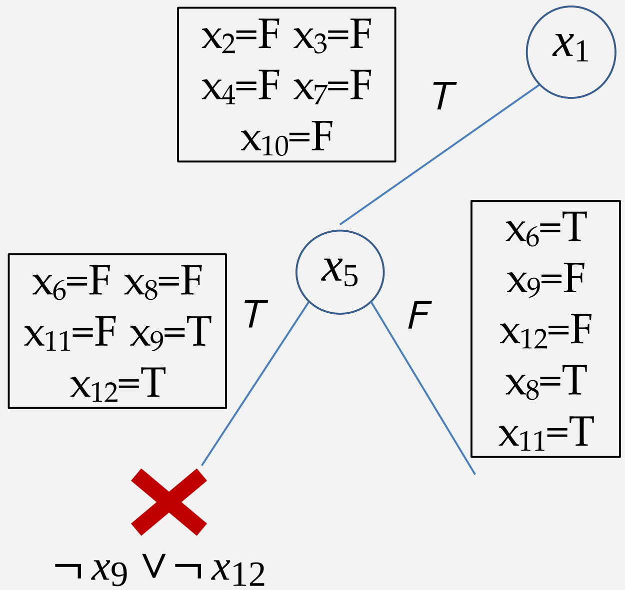


When x_5 is assigned F (=false), then x_6 must be assigned T (= true), because of the clause $x_4 \vee x_5 \vee x_6$.
Unit-clauses formed mean that that x_9, x_{12} must be F.

DPLL

$x_1 \vee x_2 \vee x_3$
$\neg x_1 \vee \neg x_2$
$\neg x_1 \vee \neg x_3$
$\neg x_2 \vee \neg x_3$
$x_4 \vee x_5 \vee x_6$
$\neg x_4 \vee \neg x_5$
$\neg x_4 \vee \neg x_6$
$\neg x_5 \vee \neg x_6$
$x_7 \vee x_8 \vee x_9$
$\neg x_7 \vee \neg x_8$
$\neg x_7 \vee \neg x_9$
$\neg x_8 \vee \neg x_9$
$x_{10} \vee x_{11} \vee x_{12}$
$\neg x_{10} \vee \neg x_{11}$
$\neg x_{10} \vee \neg x_{12}$
$\neg x_{11} \vee \neg x_{12}$
$\neg x_8 \vee \neg x_{11}$

$\neg x_1 \vee \neg x_4$
$\neg x_2 \vee \neg x_5$
$\neg x_3 \vee \neg x_6$
$\neg x_1 \vee \neg x_7$
$\neg x_2 \vee \neg x_8$
$\neg x_3 \vee \neg x_9$
$\neg x_1 \vee \neg x_{10}$
$\neg x_2 \vee \neg x_{11}$
$\neg x_3 \vee \neg x_{12}$
$\neg x_4 \vee \neg x_7$
$\neg x_5 \vee \neg x_8$
$\neg x_6 \vee \neg x_9$
$\neg x_4 \vee \neg x_{10}$
$\neg x_5 \vee \neg x_{11}$
$\neg x_6 \vee \neg x_{12}$
$\neg x_7 \vee \neg x_{10}$
$\neg x_9 \vee \neg x_{12}$

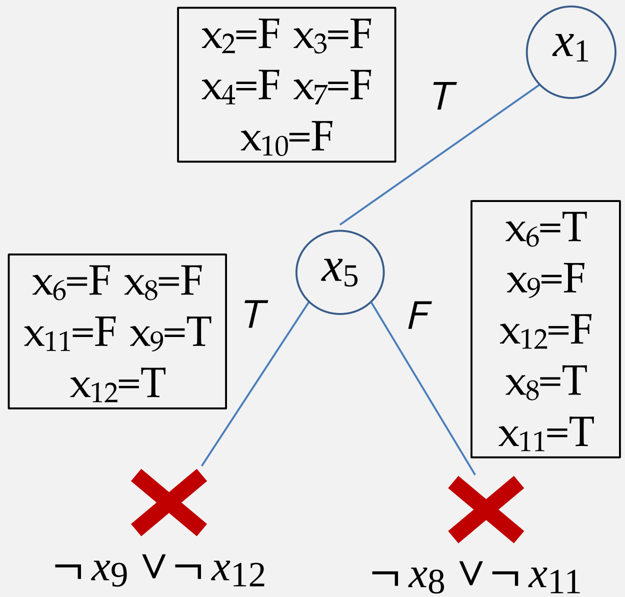


When x_5 is assigned F (=false), then x_6 must be assigned T (= true), because of the clause $x_4 \vee x_5 \vee x_6$.
Unit-clauses formed mean that that x_9, x_{12} must be F.
Further unit-clause propagation give that x_8, x_{11} must be T.

DPLL

$x_1 \lor x_2 \lor x_3$
$\neg x_1 \lor \neg x_2$
$\neg x_1 \lor \neg x_3$
$\neg x_2 \lor \neg x_3$
$x_4 \lor x_5 \lor x_6$
$\neg x_4 \lor \neg x_5$
$\neg x_4 \lor \neg x_6$
$\neg x_5 \lor \neg x_6$
$x_7 \lor x_8 \lor x_9$
$\neg x_7 \lor \neg x_8$
$\neg x_7 \lor \neg x_9$
$\neg x_8 \lor \neg x_9$
$x_{10} \lor x_{11} \lor x_{12}$
$\neg x_{10} \lor \neg x_{11}$
$\neg x_{10} \lor \neg x_{12}$
$\neg x_{11} \lor \neg x_{12}$
$\neg x_8 \lor \neg x_{11}$

$\neg x_1 \lor \neg x_4$
$\neg x_2 \lor \neg x_5$
$\neg x_3 \lor \neg x_6$
$\neg x_1 \lor \neg x_7$
$\neg x_2 \lor \neg x_8$
$\neg x_3 \lor \neg x_9$
$\neg x_1 \lor \neg x_{10}$
$\neg x_2 \lor \neg x_{11}$
$\neg x_3 \lor \neg x_{12}$
$\neg x_4 \lor \neg x_7$
$\neg x_5 \lor \neg x_8$
$\neg x_6 \lor \neg x_9$
$\neg x_4 \lor \neg x_{10}$
$\neg x_5 \lor \neg x_{11}$
$\neg x_6 \lor \neg x_{12}$
$\neg x_7 \lor \neg x_{10}$
$\neg x_9 \lor \neg x_{12}$



When x_5 is assigned F (=false), then x_6 must be assigned T (= true), because of the clause $x_4 \lor x_5 \lor x_6$.
Unit-clauses formed mean that that x_9, x_{12} must be F.
Further unit-clause propagation give that x_8, x_{11} must be T.
That causes conflict in the clause $\neg x_8 \lor \neg x_{11}$.
Hence, this branch is a dead-end, and shown as red X.

http://algs4.cs.princeton.edu

Satisfiability and NP-completeness

- *Backtracking*
- *DPLL*
- *Implementation*

API: Solvers

public class	Solvers	
	public Solvers()	*New instance*
boolean	naiveSatisfiability()	*Naive solver, returning satisfiability*
boolean	bbSatisfiability()	*Backtracking solver, returning satisfiability*
boolean	dpSatisfiability()	*DPLL solver, returning satisfiability*
Asgmt	satAsgmt()	*Return the satisfying assignment*
int	nStates()	*Return the number of solver states*
Asgmt	unitLits(Formula F, Asgmt asg)	*Perform unit clause propagation*
Int	nextVariable(Formula F, Asgmt asg)	*Find the next variable to branch on*
String	toString()	*string representation*

API: Boolean formula

public class	Fmla	
	public Fmla(In in)	*Read formula from file*
Iterable<Clause>	clauses()	*Return all the clauses*
int	nClauses();	*Return number of clauses*
void	addClause(Clause)	*Adding clauses*
boolean	isSatisfied(Assignment a)	*Does the partial assignment satisfy the formula?*
boolean	isValuated(Assignment a)	*Does the partial assignment result in a valued formula?*
String	toString()	*string representation*

Library method:

```
public static Formula RandomCNF(int n, int m, int k)
```

Generate random k-CNF formula with n variables and m clauses

Formula class implementation

```
public class Formula
{
 private final int n;              // # of variables
 private Bag<Clauses> clauses;
}
```

Formula I/O

```
public Formula(In in) {
  int n = in.readInt(), m = in.readInt();
  this.n = n; this.m = m;
  clauses = new Bag<Clauses>();
  in.readLine();
  for (int i = 0; i < m; i++) {
    String[] tokens = in.readLine().split(",");
    Clause newClause = new Clause();
    for (String lit : tokens) {
     int var = Math.abs(Integer.parseInt(lit));
     boolean sign = (lit.charAt(0) != '-');
     newClause.addLiteral(var,sign);
   }
   clauses.add(newClause);
   }
}
```

f1.txt

```
2
1
0,-1
```

$x_0 \vee \neg x_1$

f2.txt

```
2
4
0,-1
0,1
-0,1
-0,-1
```

$(x_0 \vee \neg x_1) \wedge (x_0 \vee x_1)$
$\wedge (\neg x_0 \vee x_1) \wedge (\neg x_0 \vee \neg x_1)$

API: Assignment

`public class`	`Assignment`	
	`public Assignment()`	*Constructor*
`Iterable<Integer>`	`vars()`	*Return all the variables*
`int`	`size()`	*Return no. of literals*
`boolean`	`isSet(int var)`	*Is the variable assigned?*
`boolean`	`getValue(int var)`	*Return the value of the variable*
`void`	`setValue(int var)`	*Set the value of the variable*
`void`	`unset(int var)`	*Remove the value for variable*
`void`	`joinAsgmts(Assignment asg2)`	*Combine the two assignments*
`void`	`removeAsgmts(Assignment a)`	*Symmetric difference of asgmts*
`String`	`toString()`	*string representation*

Satisfiability: Backtracking with unit propagation

```
public boolean dpSatisfiability(Formula F, Asgmt asg) {
   nStates++;
   Asgmt unitVars = F.unitLits(asg);
   if (F.isValuated(asg)) {
      boolean result = F.isSatisfied(asg);
      if (!result) asg.unset(unitVars);
      return result;
   }
   int v = chooseVariable(asg);
   asg.add(v, true);
   if (dpSatisfiability(asg)) return true;
   asg.add(v, false);
   if (dpSatisfiability(asg)) return true;
   asg.removeAsgt(asg,unitVars);
   asg.remove(v);
   return false;
}
```

Algorithms

Robert Sedgewick | Kevin Wayne

http://algs4.cs.princeton.edu

Satisfiability and NP-completeness

- *Backtracking*
- *DPLL*
- *Further improvements*

Further improvements

Variable selection rule

- Can add heuristics to select the next variable to branch on
  - Random; Most frequent in unsatisfied clauses

Conflict-driven learning

- Learn (and remember) new clauses that can be added
- Means failures are discovered earlier

Solvable special cases

- 2-SAT : SAT where each clause has (at most) 2 literals
- Solvable in linear time

Other modern improvements

- Efficient implementation of the unit-clause rule
- Formula preprocessing

Boolean Schur Triples Problem

Mathematical problem (related to Ramsey theory):

- Does there exists a red/blue coloring of the numbers 1 to n, such that there is no monochromatic solution of $a + b = c$ with $a < b < c \leq n$.

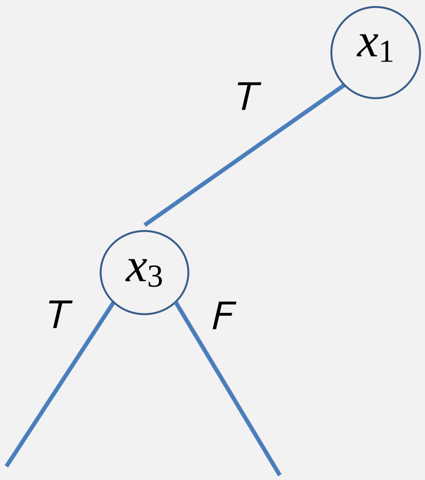
$$
\begin{aligned}
&(x_1 \lor x_2 \lor x_3) \land (\overline{x}_1 \lor \overline{x}_2 \lor \overline{x}_3) \land (x_1 \lor x_3 \lor x_4) \land (\overline{x}_1 \lor \overline{x}_3 \lor \overline{x}_4) \land \\
&(x_1 \lor x_4 \lor x_5) \land (\overline{x}_1 \lor \overline{x}_4 \lor \overline{x}_5) \land (x_2 \lor x_3 \lor x_5) \land (\overline{x}_2 \lor \overline{x}_3 \lor \overline{x}_5) \land \\
&(x_1 \lor x_5 \lor x_6) \land (\overline{x}_1 \lor \overline{x}_5 \lor \overline{x}_6) \land (x_2 \lor x_4 \lor x_6) \land (\overline{x}_2 \lor x_4 \lor \overline{x}_6) \land \\
&(x_1 \lor x_6 \lor x_7) \land (\overline{x}_1 \lor \overline{x}_6 \lor \overline{x}_7) \land (x_2 \lor x_5 \lor x_7) \land (\overline{x}_2 \lor \overline{x}_5 \lor \overline{x}_7) \land \\
&(x_3 \lor x_4 \lor x_7) \land (\overline{x}_3 \lor \overline{x}_4 \lor \overline{x}_7) \land (x_1 \lor x_7 \lor x_8) \land (\overline{x}_1 \lor \overline{x}_7 \lor \overline{x}_8) \land \\
&(x_2 \lor x_6 \lor x_8) \land (\overline{x}_2 \lor \overline{x}_6 \lor \overline{x}_8) \land (x_3 \lor x_5 \lor x_8) \land (\overline{x}_3 \lor \overline{x}_5 \lor \overline{x}_8) \land \\
&(x_1 \lor x_8 \lor x_9) \land (\overline{x}_1 \lor \overline{x}_8 \lor \overline{x}_9) \land (x_2 \lor x_7 \lor x_9) \land (\overline{x}_2 \lor \overline{x}_7 \lor \overline{x}_9) \land \\
&(x_3 \lor x_6 \lor x_9) \land (\overline{x}_3 \lor \overline{x}_6 \lor \overline{x}_9) \land (x_4 \lor x_5 \lor x_9) \land (\overline{x}_4 \lor \overline{x}_5 \lor \overline{x}_9)
\end{aligned}
$$

sch.txt

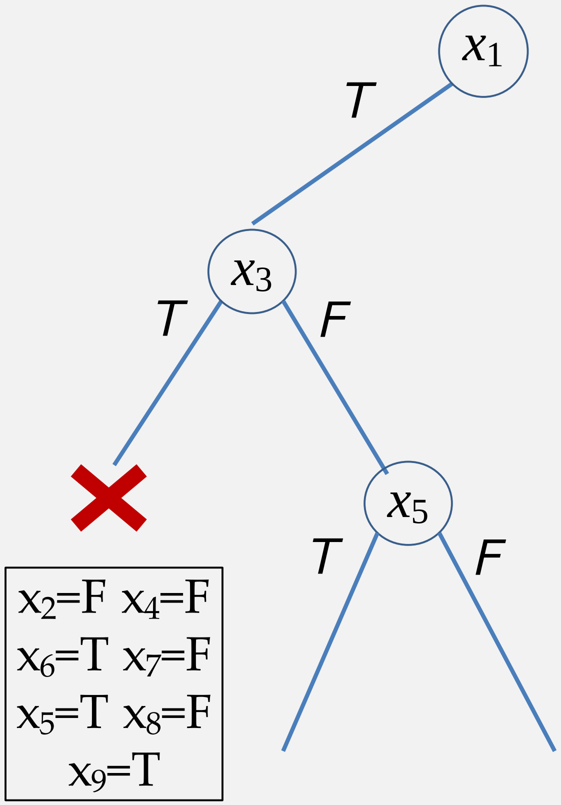
Source:
- Heule, Kullman, "The science of brute force", Communications of the ACM, Aug 2017.

Boolean Schur Triples Problem

$$(x_1 \lor x_2 \lor x_3) \land (\overline{x}_1 \lor \overline{x}_2 \lor \overline{x}_3) \land (x_1 \lor x_3 \lor x_4) \land (\overline{x}_1 \lor \overline{x}_3 \lor \overline{x}_4) \land$$
$$(x_1 \lor x_4 \lor x_5) \land (\overline{x}_1 \lor \overline{x}_4 \lor \overline{x}_5) \land (x_2 \lor x_3 \lor x_5) \land (\overline{x}_2 \lor \overline{x}_3 \lor \overline{x}_5) \land$$
$$(x_1 \lor x_5 \lor x_6) \land (\overline{x}_1 \lor \overline{x}_5 \lor \overline{x}_6) \land (x_2 \lor x_4 \lor x_6) \land (\overline{x}_2 \lor x_4 \lor \overline{x}_6) \land$$
$$(x_1 \lor x_6 \lor x_7) \land (\overline{x}_1 \lor \overline{x}_6 \lor \overline{x}_7) \land (x_2 \lor x_5 \lor x_7) \land (\overline{x}_2 \lor \overline{x}_5 \lor \overline{x}_7) \land$$
$$(x_3 \lor x_4 \lor x_7) \land (\overline{x}_3 \lor \overline{x}_4 \lor \overline{x}_7) \land (x_1 \lor x_7 \lor x_8) \land (\overline{x}_1 \lor \overline{x}_7 \lor \overline{x}_8) \land$$
$$(x_2 \lor x_6 \lor x_8) \land (\overline{x}_2 \lor \overline{x}_6 \lor \overline{x}_8) \land (x_3 \lor x_5 \lor x_8) \land (\overline{x}_3 \lor \overline{x}_5 \lor \overline{x}_8) \land$$
$$(x_1 \lor x_8 \lor x_9) \land (\overline{x}_1 \lor \overline{x}_8 \lor \overline{x}_9) \land (x_2 \lor x_7 \lor x_9) \land (\overline{x}_2 \lor \overline{x}_7 \lor \overline{x}_9) \land$$
$$(x_3 \lor x_6 \lor x_9) \land (\overline{x}_3 \lor \overline{x}_6 \lor \overline{x}_9) \land (x_4 \lor x_5 \lor x_9) \land (\overline{x}_4 \lor \overline{x}_5 \lor \overline{x}_9)$$



Boolean Schur Triples Problem



$$
\begin{aligned}
&(x_1 \vee x_2 \vee x_3) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3) \wedge (x_1 \vee x_3 \vee x_4) \wedge (\overline{x}_1 \vee \overline{x}_3 \vee \overline{x}_4) \wedge \\
&(x_1 \vee x_4 \vee x_5) \wedge (\overline{x}_1 \vee \overline{x}_4 \vee \overline{x}_5) \wedge (x_2 \vee x_3 \vee x_5) \wedge (\overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_5) \wedge \\
&(x_1 \vee x_5 \vee x_6) \wedge (\overline{x}_1 \vee \overline{x}_5 \vee \overline{x}_6) \wedge (x_2 \vee x_4 \vee x_6) \wedge (\overline{x}_2 \vee x_4 \vee \overline{x}_6) \wedge \\
&(x_1 \vee x_6 \vee x_7) \wedge (\overline{x}_1 \vee \overline{x}_6 \vee \overline{x}_7) \wedge (x_2 \vee x_5 \vee x_7) \wedge (\overline{x}_2 \vee \overline{x}_5 \vee \overline{x}_7) \wedge \\
&(x_3 \vee x_4 \vee x_7) \wedge (\overline{x}_3 \vee \overline{x}_4 \vee \overline{x}_7) \wedge (x_1 \vee x_7 \vee x_8) \wedge (\overline{x}_1 \vee \overline{x}_7 \vee \overline{x}_8) \wedge \\
&(x_2 \vee x_6 \vee x_8) \wedge (\overline{x}_2 \vee \overline{x}_6 \vee \overline{x}_8) \wedge (x_3 \vee x_5 \vee x_8) \wedge (\overline{x}_3 \vee \overline{x}_5 \vee \overline{x}_8) \wedge \\
&(x_1 \vee x_8 \vee x_9) \wedge (\overline{x}_1 \vee \overline{x}_8 \vee \overline{x}_9) \wedge (x_2 \vee x_7 \vee x_9) \wedge (\overline{x}_2 \vee \overline{x}_7 \vee \overline{x}_9) \wedge \\
&(x_3 \vee x_6 \vee x_9) \wedge (\overline{x}_3 \vee \overline{x}_6 \vee \overline{x}_9) \wedge (x_4 \vee x_5 \vee x_9) \wedge (\overline{x}_4 \vee \overline{x}_5 \vee \overline{x}_9)
\end{aligned}
$$

Satisfiability and NP-completeness

- *Backtracking*
- *DPLL*
- *Further improvements*
- *2-Satisfiability*

Algorithms

Robert Sedgewick | Kevin Wayne

http://algs4.cs.princeton.edu

2-Satisfiability

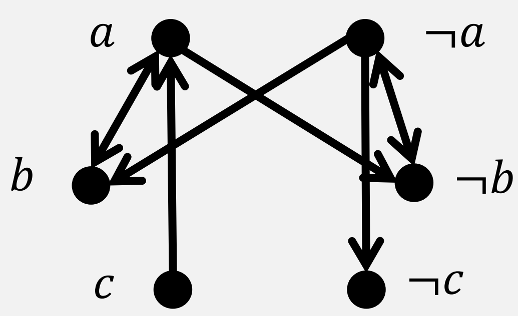
2-Satisfiability problem Satisfiability when all clauses have at most 2 literals.

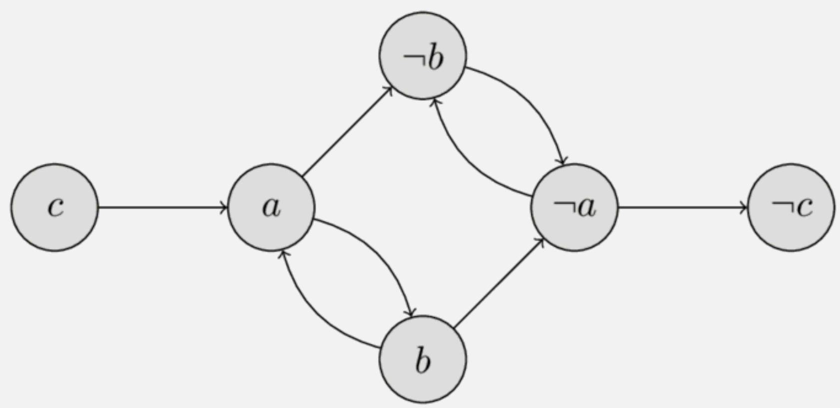
Example $(a \lor \neg b) \land (\neg a \lor b) \land (\neg a \lor \neg b) \land (a \lor \neg c)$

Equivalent implications

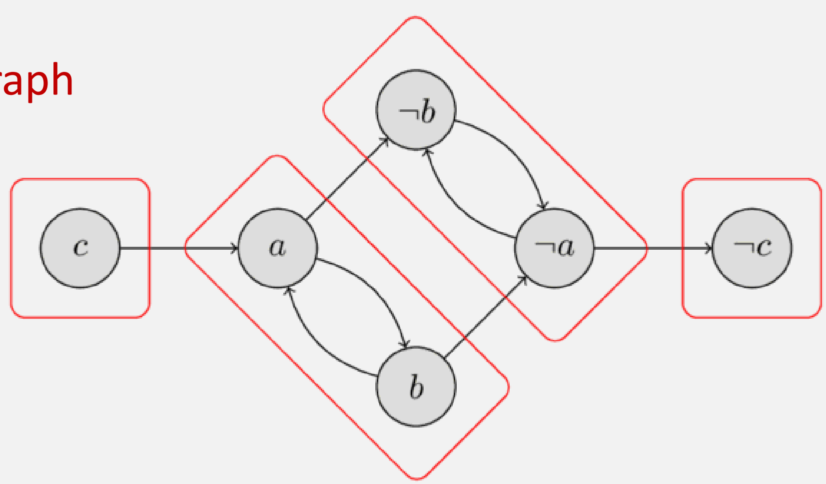
$\neg a \Rightarrow \neg b$	$a \Rightarrow b$	$a \Rightarrow \neg b$	$\neg a \Rightarrow \neg c$
$b \Rightarrow a$	$\neg b \Rightarrow \neg a$	$b \Rightarrow \neg a$	$c \Rightarrow a$

Implication graph





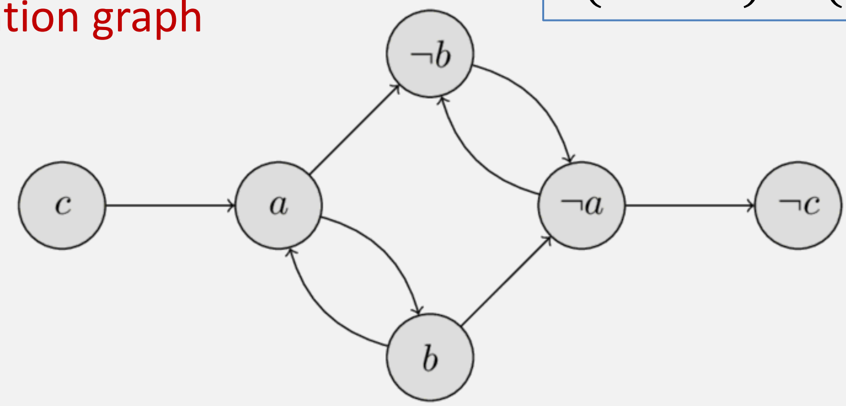
Condensation graph



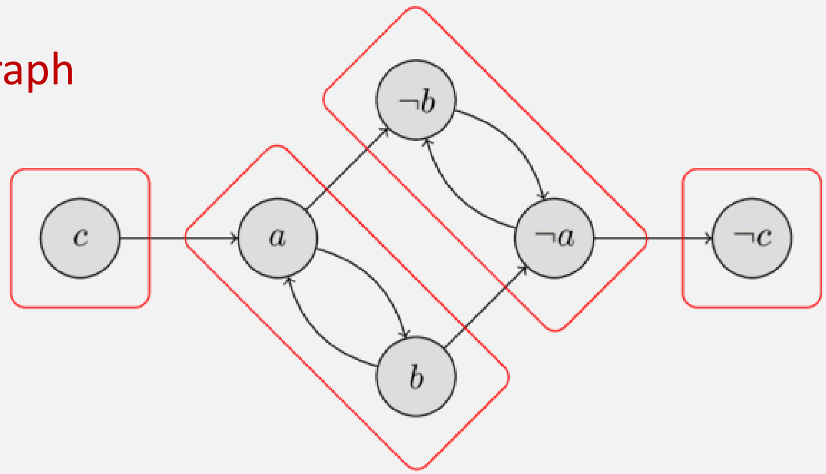
2-Satisfiability : DFS-based algorithm (Aspvall, Plath, Tarjan)

$$(a \vee \neg b) \wedge (\neg a \vee b) \wedge (\neg a \vee \neg b) \wedge (a \vee \neg c)$$

Implication graph



Condensation graph



Algorithm

- Form condensation graph H, by shrinking each strong component to a single node
- Process the strong components of the condensation graph in reverse topological order
  - Set all the literals in the component to be true, if possible; otherwise, false

References

Attributions

- Joao Marques-Silva, Southampton
  - „Practical applications of Boolean Satisfiability“, talk at WODES’08
- Ari K. Jónsson, rektor
  - Presentation, „Practical Planning I“
- Federico Pecora, Örebro University
  - Lecture in Advanced AI, DT4019
- David Dill, Stanford
  - Lecture 2: Practical SAT solving, CS 357