Conflict analysis in SAT solvers

Bjørnar Luteberget

2024-05-14

Background

Why look at SAT solvers for real-world optimization problems?

- SAT solvers have state-of-the-art perf. on *some* optimization problems.
  - Example: "small-domain" but hard resource-constrained project scheduling.
  - Example: dynamic time-indexed models (DDD)
- SAT solvers have state-of-the-art perf. on some *feasibility (sub)problems*
  - Example: railway bound-to-deadlock detection
  - Example: geographical decomposition in railway scheduling problems

Today's topic:

- SAT solvers are built on **tree search** with "*deduction*/*propagation*/*presolving*"
  - Like MILP solvers and other optimization solvers
  - What is special about SAT solvers that MILP solvers don't have?

Today's topic

Today's topic:

- SAT solvers are built on **tree search** with "*deduction*/*propagation*/*presolving*"
  - Like MILP solvers and other optimization solvers
  - What is special about SAT solvers that MILP solvers don't have?
- Answer: **conflict-driven clause learning**
  - Marques-Silva, J. P., & Sakallah, K. A. (1999). GRASP: A search algorithm for propositional satisfiability. IEEE Transactions on Computers, 48(5), 506-521.
- No original research today – text-book lecture based on
  - Biere, A., Heule, M., & van Maaren, H. (Eds.). (2009). Handbook of satisfiability (Vol. 185). IOS press.

Outline

Systematic (complete) solvers for discrete “variables and constraints” problems:

- Splitting the search space: tree of assignments
- Simplifying the problem under a partial assignment
- Backtracking when infeasible

Outline

Systematic (complete) solvers for discrete "variables and constraints" problems:

- Splitting the search space: tree of assignments
  - **SAT**: activity-based variable ordering
  - **MILP**: pseudo-cost branching variable ordering
- Simplifying the problem under a partial assignment
- Backtracking when infeasible

Outline

Systematic (complete) solvers for discrete “variables and constraints” problems:

- Splitting the search space: tree of assignments
  - **SAT**: activity-based variable ordering
  - **MILP**: pseudo-cost branching variable ordering
- Simplifying the problem under a partial assignment
  - **SAT**: unit propagation
  - **MILP**: node pre-solving (+cutting planes)
- Backtracking when infeasible

Outline

Systematic (complete) solvers for discrete “variables and constraints” problems:

- Splitting the search space: tree of assignments
  - **SAT**: activity-based variable ordering
  - **MILP**: pseudo-cost branching variable ordering
- Simplifying the problem under a partial assignment
  - **SAT**: unit propagation
  - **MILP**: node pre-solving (+cutting planes)
- Backtracking when infeasible
  - **SAT**: *conflict analysis, clause learning*, backtracking
  - **MILP**: best-bound node queue

Two prerequisites

- SAT problem description: Conjunctive normal form
- SAT problem simplification: Unit propagation

Conjunctive normal form

- Most SAT solvers take only formulas given in conjunctive normal form

$$(x \vee \bar{y} \vee \ldots) \wedge (z \vee \overline{w} \vee \ldots) \wedge \ldots$$

- Variables and negated variables are called *literals*.
- Disjunctions of literals are called *clauses*.

$$x \vee y \vee \bar{z} \qquad (x + y + (1 - z) \geq 1)$$

- CNF is a conjunction of clauses.
- By adding some variables, we can transform any propositional logic formula to CNF (with worst-case linear increase in size).

Unit propagation

A simple and efficient preprocessing/deduction technique:

- Any clauses that...
  - are not already satisfied by the current assignment, and ...
  - have only one remaining unassigned variable ...

  ... forces the assignment of the remaining variable.

- Example: $a \vee \overline{b} \vee c$, under the partial assignment, $a = \perp, c = \perp$, propagates

$$b = \perp.$$

- (Special case of domain propagation for linear constraints.)

Worked example

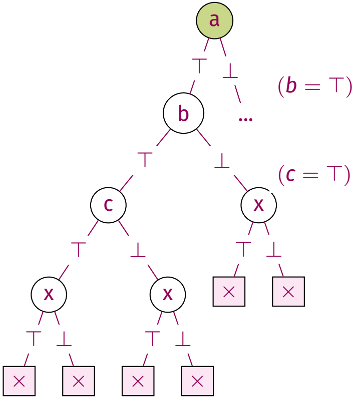
Consider the CNF problem:

$$
\begin{aligned}
&(a \vee b) \wedge \\
&(b \vee c) \wedge \\
&(\overline{a} \vee \overline{x} \vee y) \wedge \\
&(\overline{a} \vee x \vee z) \wedge \\
&(\overline{a} \vee \overline{y} \vee z) \wedge \\
&(\overline{a} \vee x \vee \overline{z}) \wedge \\
&(\overline{a} \vee \overline{y} \vee \overline{z})
\end{aligned}
$$

... and fix the variable ordering to a, b, c, x, y, z.

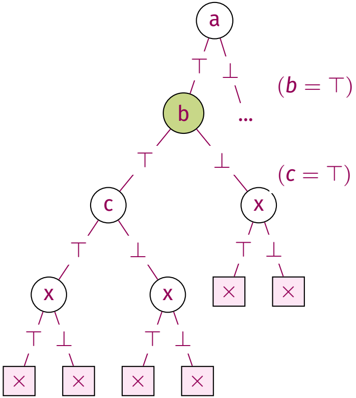
Tree search

$$(a \vee b) \wedge$$
$$(b \vee c) \wedge$$
$$(\overline{a} \vee \overline{x} \vee y) \wedge$$
$$(\overline{a} \vee x \vee z) \wedge$$
$$(\overline{a} \vee \overline{y} \vee z) \wedge$$
$$(\overline{a} \vee x \vee \overline{z}) \wedge$$
$$(\overline{a} \vee \overline{y} \vee \overline{z})$$



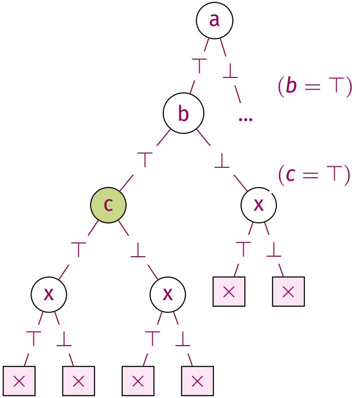
Tree search

$$
\begin{aligned}
\cancel{(a \vee b)} \wedge \\
(b \vee c) \wedge \\
(\overline{a} \vee \overline{x} \vee y) \wedge \\
(\overline{a} \vee x \vee z) \wedge \\
(\overline{a} \vee \overline{y} \vee z) \wedge \\
(\overline{a} \vee x \vee \overline{z}) \wedge \\
(\overline{a} \vee \overline{y} \vee \overline{z})
\end{aligned}
$$



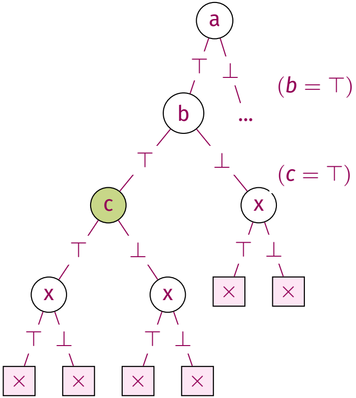
Tree search

$$\cancel{(a \vee b)} \wedge$$
$$\cancel{(b \vee c)} \wedge$$
$$(\overline{a} \vee \overline{x} \vee y) \wedge$$
$$(\overline{a} \vee x \vee z) \wedge$$
$$(\overline{a} \vee \overline{y} \vee z) \wedge$$
$$(\overline{a} \vee x \vee \overline{z}) \wedge$$
$$(\overline{a} \vee \overline{y} \vee \overline{z})$$



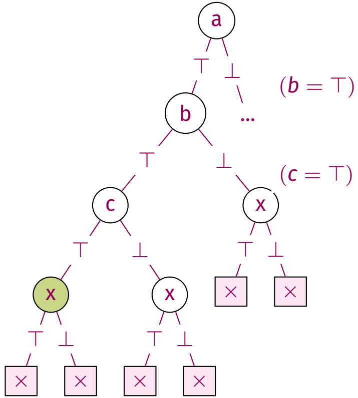
Tree search

$$\cancel{(a \vee b)} \wedge$$
$$\cancel{(b \vee c)} \wedge$$
$$(\bar{a} \vee \bar{x} \vee y) \wedge$$
$$(\bar{a} \vee x \vee z) \wedge$$
$$(\bar{a} \vee \bar{y} \vee z) \wedge$$
$$(\bar{a} \vee x \vee \bar{z}) \wedge$$
$$(\bar{a} \vee \bar{y} \vee \bar{z})$$



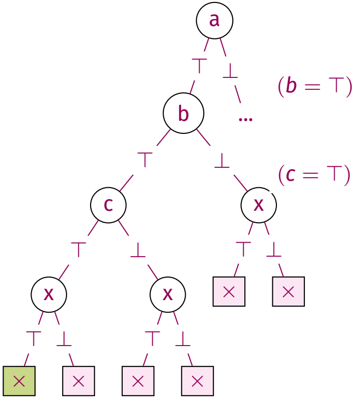
Tree search

$$\cancel{(a \vee b)} \wedge$$
$$\cancel{(b \vee c)} \wedge$$
$$(\overline{a} \vee \overline{x} \vee y) \wedge$$
$$(\overline{a} \vee x \vee z) \wedge$$
$$(\overline{a} \vee \overline{y} \vee z) \wedge$$
$$(\overline{a} \vee x \vee \overline{z}) \wedge$$
$$(\overline{a} \vee \overline{y} \vee \overline{z})$$



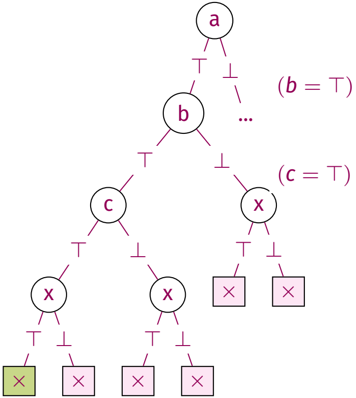
Tree search

$$\cancel{(a \vee b)} \wedge$$
$$\cancel{(b \vee c)} \wedge$$
$$(\bar{a} \vee \bar{x} \vee y) \wedge$$
$$(\bar{a} \vee x \vee z) \wedge$$
$$(\bar{a} \vee \bar{y} \vee z) \wedge$$
$$(\bar{a} \vee x \vee \bar{z}) \wedge$$
$$(\bar{a} \vee \bar{y} \vee \bar{z})$$



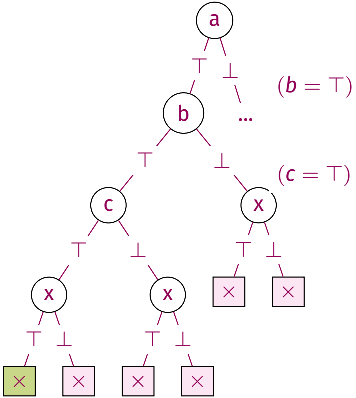
Tree search

$$
\begin{aligned}
&\cancel{(a \lor b)} \land \\
&\cancel{(b \lor c)} \land \\
&(\bar{a} \lor \bar{x} \lor y) \land \\
&\cancel{(\bar{a} \lor x \lor z)} \land \\
&(\bar{a} \lor \bar{y} \lor z) \land \\
&\cancel{(\bar{a} \lor x \lor \bar{z})} \land \\
&(\bar{a} \lor \bar{y} \lor \bar{z})
\end{aligned}
$$



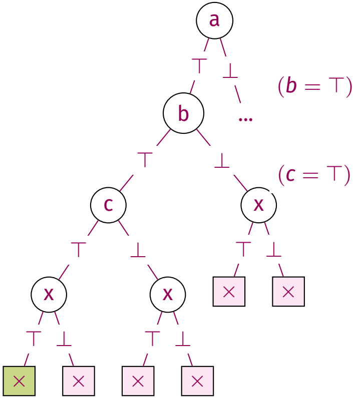
Tree search

$$\begin{aligned}
\cancel{(a \vee b)} &\wedge \\
\cancel{(b \vee c)} &\wedge \\
(\bar{a} \vee \bar{x} \vee y) &\wedge \\
\cancel{(\bar{a} \vee x \vee z)} &\wedge \\
(\bar{a} \vee \bar{y} \vee z) &\wedge \\
\cancel{(\bar{a} \vee x \vee \bar{z})} &\wedge \\
(\bar{a} \vee \bar{y} \vee \bar{z}) &
\end{aligned}$$



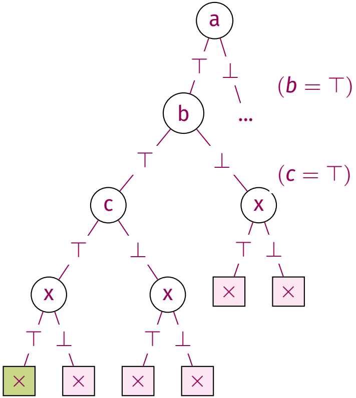
Tree search

$\cancel{(a \vee b)} \wedge$
$\cancel{(b \vee c)} \wedge$
$\cancel{(\bar{a} \vee \bar{x} \vee y)} \wedge$
$\cancel{(\bar{a} \vee x \vee z)} \wedge$
$(\bar{a} \vee \bar{y} \vee z) \wedge$
$\cancel{(\bar{a} \vee x \vee \bar{z})} \wedge$
$(\bar{a} \vee \bar{y} \vee \bar{z})$



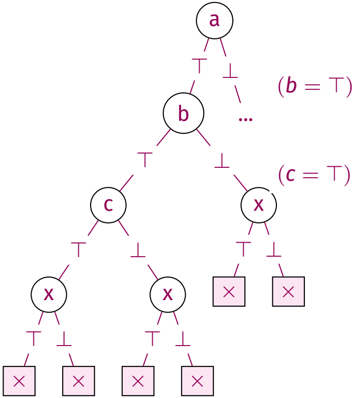
Tree search

$$\cancel{(a \vee b)} \wedge$$
$$\cancel{(b \vee c)} \wedge$$
$$\cancel{(\overline{a} \vee \overline{x} \vee y)} \wedge$$
$$\cancel{(\overline{a} \vee x \vee z)} \wedge$$
$$\cancel{(\overline{a} \vee \overline{y} \vee z)} \wedge$$
$$\cancel{(\overline{a} \vee x \vee \overline{z})} \wedge$$
$$(\overline{a} \vee \overline{y} \vee \overline{z})$$



Tree search

$$
\begin{aligned}
(a \vee b) \wedge \\
(b \vee c) \wedge \\
(\overline{a} \vee \overline{x} \vee y) \wedge \\
(\overline{a} \vee x \vee z) \wedge \\
(\overline{a} \vee \overline{y} \vee z) \wedge \\
(\overline{a} \vee x \vee \overline{z}) \wedge \\
(\overline{a} \vee \overline{y} \vee \overline{z})
\end{aligned}
$$



Conflict analysis

- Can we do something better than simple backtracking when reaching a conflict?

Conflict analysis

Definition (Implication graph)

Let π be a path in the search tree. The Implication graph G_π is the directed acyclic graph (V, E) where

- $l \in V$ for each variable assignment (both decisions and unit propagations)
- For any unit propagation along π, derived from clause $C = (l_1 \vee \ldots \vee l_k \vee l)$, E contains the edges $(\overline{l_1}, l)$, . . ., $(\overline{l_k}, l)$.
- If π falsifies a clause, G_π contains the special *conflict vertex* $\bot$ and edges $(\overline{l_1}, \bot), \ldots, (\overline{l_k}, \bot)$ for exactly one falsified clause $(l_1 \vee \ldots \vee l_k)$.

Implication graph

$$\begin{aligned}(a \vee b) \wedge \\ (b \vee c) \wedge \\ (\overline{a} \vee \overline{x} \vee y) \wedge \\ (\overline{a} \vee x \vee z) \wedge \\ (\overline{a} \vee \overline{y} \vee z) \wedge \\ (\overline{a} \vee x \vee \overline{z}) \wedge \\ (\overline{a} \vee \overline{y} \vee \overline{z})\end{aligned}$$

$\boxed{1 : a = \top}$

Implication graph

$$
\begin{aligned}
(a \vee b) &\wedge \\
(b \vee c) &\wedge \\
(\overline{a} \vee \overline{x} \vee y) &\wedge \\
(\overline{a} \vee x \vee z) &\wedge \\
(\overline{a} \vee \overline{y} \vee z) &\wedge \\
(\overline{a} \vee x \vee \overline{z}) &\wedge \\
(\overline{a} \vee \overline{y} \vee \overline{z}) &
\end{aligned}
$$

$\boxed{1 : a = \top}$

$\boxed{2 : b = \top}$

Implication graph

$$
\begin{aligned}
&(a \lor b) \land \\
&(b \lor c) \land \\
&(\overline{a} \lor \overline{x} \lor y) \land \\
&(\overline{a} \lor x \lor z) \land \\
&(\overline{a} \lor \overline{y} \lor z) \land \\
&(\overline{a} \lor x \lor \overline{z}) \land \\
&(\overline{a} \lor \overline{y} \lor \overline{z})
\end{aligned}
$$

$\boxed{1 : a = \top}$

$\boxed{2 : b = \top}$

$\boxed{3 : c = \top}$

Implication graph

$$
\begin{aligned}
&(a \lor b) \land \\
&(b \lor c) \land \\
&(\overline{a} \lor \overline{x} \lor y) \land \\
&(\overline{a} \lor x \lor z) \land \\
&(\overline{a} \lor \overline{y} \lor z) \land \\
&(\overline{a} \lor x \lor \overline{z}) \land \\
&(\overline{a} \lor \overline{y} \lor \overline{z})
\end{aligned}
$$

$\boxed{1 : a = \top}$

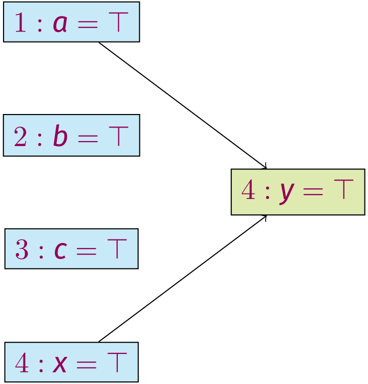
$\boxed{2 : b = \top}$

$\boxed{3 : c = \top}$

$\boxed{4 : x = \top}$

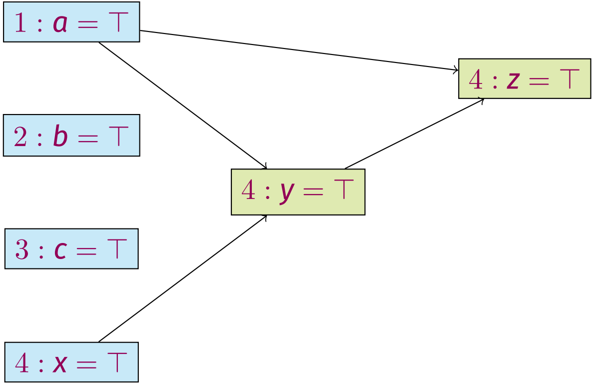
Implication graph

$$
\begin{aligned}
&(a \vee b) \wedge \\
&(b \vee c) \wedge \\
&(\overline{a} \vee \overline{x} \vee y) \wedge \\
&(\overline{a} \vee x \vee z) \wedge \\
&(\overline{a} \vee \overline{y} \vee z) \wedge \\
&(\overline{a} \vee x \vee \overline{z}) \wedge \\
&(\overline{a} \vee \overline{y} \vee \overline{z})
\end{aligned}
$$



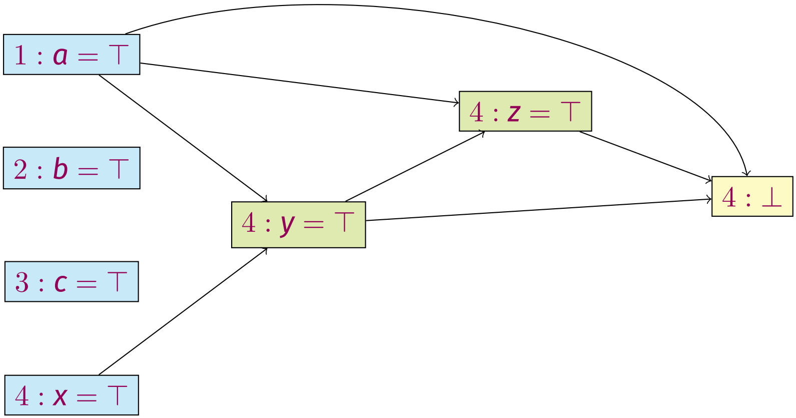
Implication graph

$$
\begin{aligned}
(a \vee b) &\wedge \\
(b \vee c) &\wedge \\
(\overline{a} \vee \overline{x} \vee y) &\wedge \\
(\overline{a} \vee x \vee z) &\wedge \\
(\overline{a} \vee \overline{y} \vee z) &\wedge \\
(\overline{a} \vee x \vee \overline{z}) &\wedge \\
(\overline{a} \vee \overline{y} \vee \overline{z})&
\end{aligned}
$$



Implication graph

$$
\begin{aligned}
(a \vee b) \wedge \\
(b \vee c) \wedge \\
(\bar{a} \vee \bar{x} \vee y) \wedge \\
(\bar{a} \vee x \vee z) \wedge \\
(\bar{a} \vee \bar{y} \vee z) \wedge \\
(\bar{a} \vee x \vee \bar{z}) \wedge \\
(\bar{a} \vee \bar{y} \vee \bar{z})
\end{aligned}
$$



Conflict cut

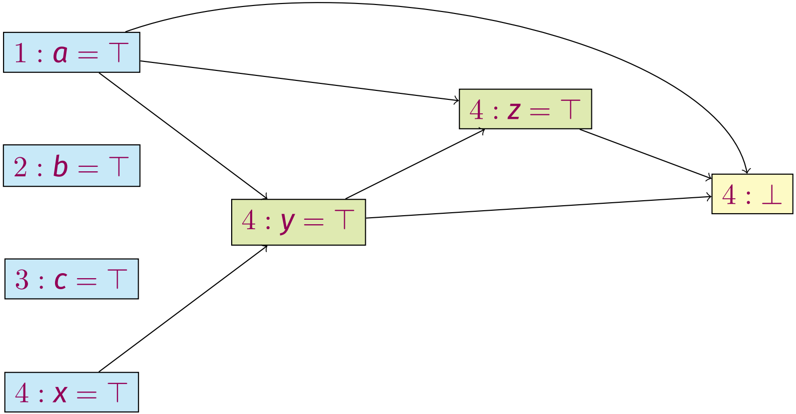
Definition (Conflict cut)

For an implication graph containing the conflict vertex $\perp$, a *conflict cut* is a partition $W = (A, B)$ where:

- all decisions belong to A
- the conflict vertex $\perp$ belongs to B

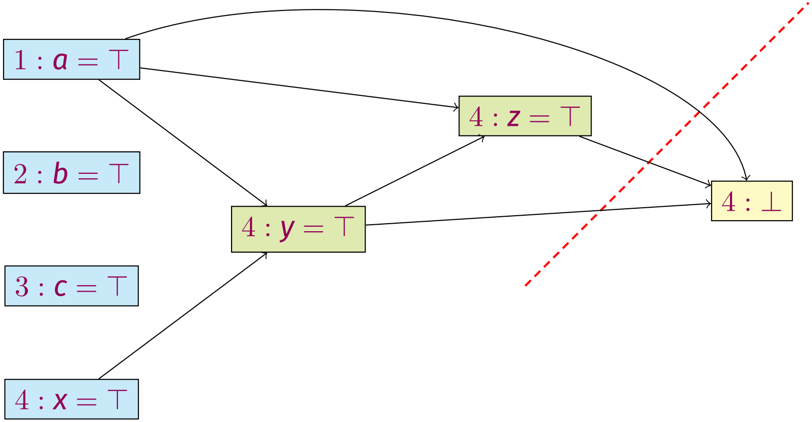
Implication graph

$$
\begin{aligned}
(a \lor b) &\land \\
(b \lor c) &\land \\
(\overline{a} \lor \overline{x} \lor y) &\land \\
(\overline{a} \lor x \lor z) &\land \\
(\overline{a} \lor \overline{y} \lor z) &\land \\
(\overline{a} \lor x \lor \overline{z}) &\land \\
(\overline{a} \lor \overline{y} \lor \overline{z}) &
\end{aligned}
$$



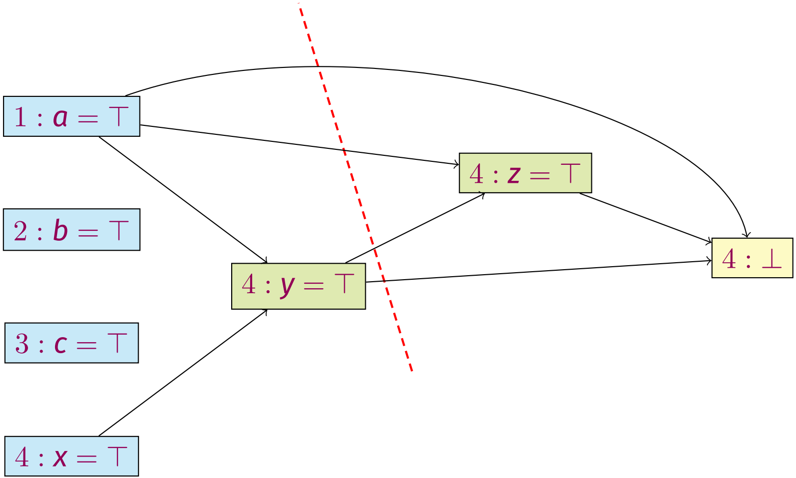
Implication graph

$$(a \vee b) \wedge$$
$$(b \vee c) \wedge$$
$$(\overline{a} \vee \overline{x} \vee y) \wedge$$
$$(\overline{a} \vee x \vee z) \wedge$$
$$(\overline{a} \vee \overline{y} \vee z) \wedge$$
$$(\overline{a} \vee x \vee \overline{z}) \wedge$$
$$(\overline{a} \vee \overline{y} \vee \overline{z})$$



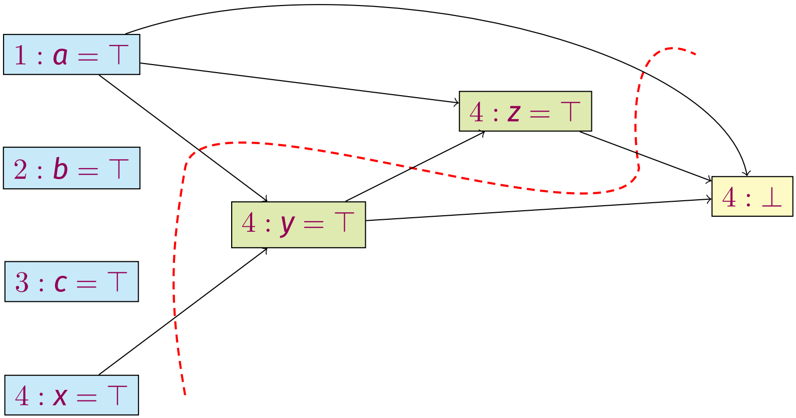
Implication graph

$$
\begin{aligned}
&(a \vee b) \wedge \\
&(b \vee c) \wedge \\
&(\bar{a} \vee \bar{x} \vee y) \wedge \\
&(\bar{a} \vee x \vee z) \wedge \\
&(\bar{a} \vee \bar{y} \vee z) \wedge \\
&(\bar{a} \vee x \vee \bar{z}) \wedge \\
&(\bar{a} \vee \bar{y} \vee \bar{z})
\end{aligned}
$$



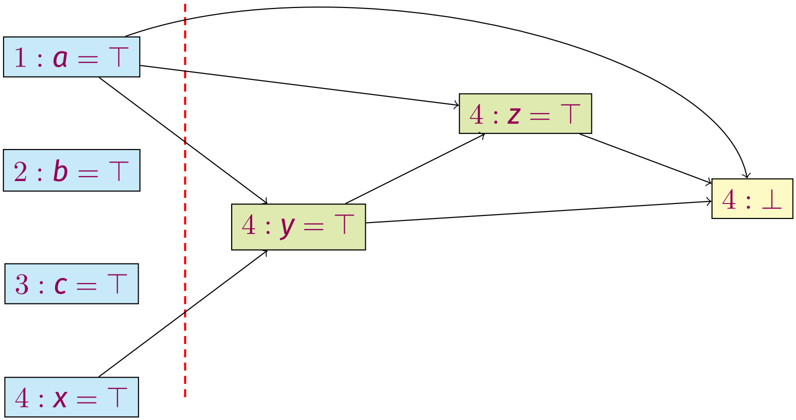
Implication graph

$$
\begin{aligned}
(a \vee b) \wedge \\
(b \vee c) \wedge \\
(\overline{a} \vee \overline{x} \vee y) \wedge \\
(\overline{a} \vee x \vee z) \wedge \\
(\overline{a} \vee \overline{y} \vee z) \wedge \\
(\overline{a} \vee x \vee \overline{z}) \wedge \\
(\overline{a} \vee \overline{y} \vee \overline{z})
\end{aligned}
$$



Implication graph

$$
\begin{aligned}
(a \vee b) &\wedge \\
(b \vee c) &\wedge \\
(\overline{a} \vee \overline{x} \vee y) &\wedge \\
(\overline{a} \vee x \vee z) &\wedge \\
(\overline{a} \vee \overline{y} \vee z) &\wedge \\
(\overline{a} \vee x \vee \overline{z}) &\wedge \\
(\overline{a} \vee \overline{y} \vee \overline{z}) &
\end{aligned}
$$



Learnt clause

Definition (Reason set)

The *reason set* of a conflict cut W is $R = \{l \in A \mid \exists l' \in B : (l, l') \in E\}$.
(the nodes in A that have edges to B)

Definition (Learnt clause)

The *learnt clause* corresponding to W is $\bigvee_{l \in R} \bar{l}$.

Theorem

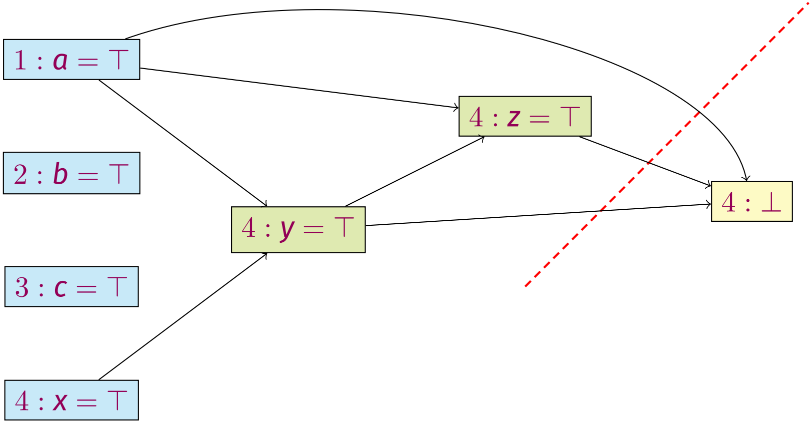
Learnt clauses are valid, i.e. they are implied by the formula.

Setting all R-literals to true and performing unit propagation will produce a conflict. Therefore, $\bigvee_{l \in R} \bar{l}$ is a logical consequence of the original formula.

$$\bigwedge_{l \in R} l \quad \text{is contradictory, so} \quad \neg(\bigwedge_{l \in R} l) = \bigvee_{l \in R} \bar{l} \quad \text{is valid.}$$

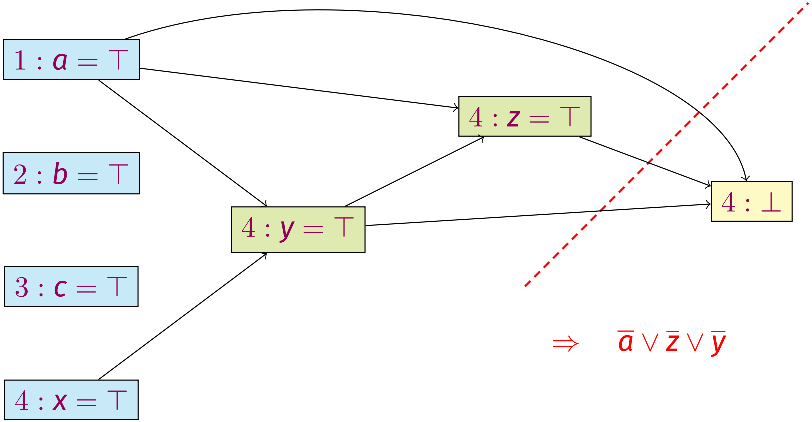
Implication graph

$$
\begin{aligned}
&(a \vee b) \wedge \\
&(b \vee c) \wedge \\
&(\overline{a} \vee \overline{x} \vee y) \wedge \\
&(\overline{a} \vee x \vee z) \wedge \\
&(\overline{a} \vee \overline{y} \vee z) \wedge \\
&(\overline{a} \vee x \vee \overline{z}) \wedge \\
&(\overline{a} \vee \overline{y} \vee \overline{z})
\end{aligned}
$$



Implication graph

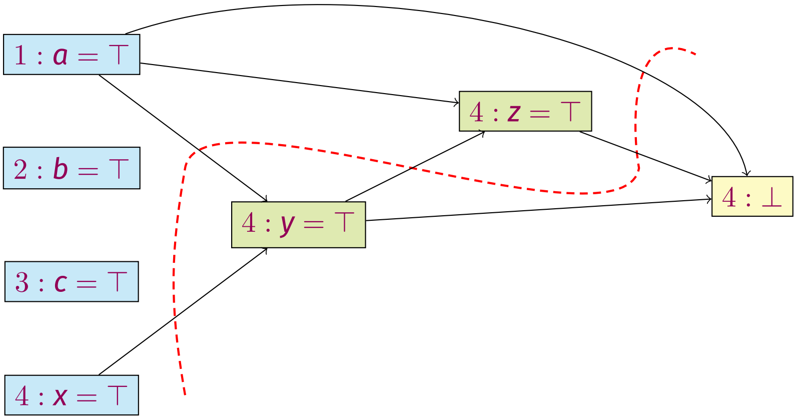
$$
\begin{aligned}
&(a \vee b) \wedge \\
&(b \vee c) \wedge \\
&(\overline{a} \vee \overline{x} \vee y) \wedge \\
&(\overline{a} \vee x \vee z) \wedge \\
&(\overline{a} \vee \overline{y} \vee z) \wedge \\
&(\overline{a} \vee x \vee \overline{z}) \wedge \\
&(\overline{a} \vee \overline{y} \vee \overline{z})
\end{aligned}
$$



$$\Rightarrow \quad \overline{a} \vee \overline{z} \vee \overline{y}$$

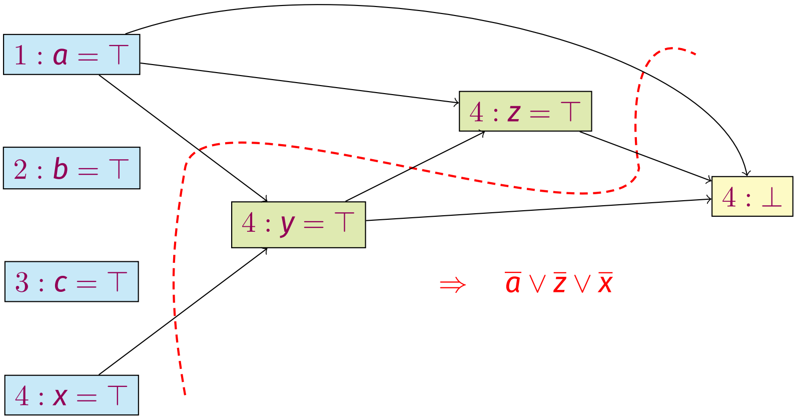
Implication graph

$$
\begin{aligned}
(a \lor b) &\land \\
(b \lor c) &\land \\
(\overline{a} \lor \overline{x} \lor y) &\land \\
(\overline{a} \lor x \lor z) &\land \\
(\overline{a} \lor \overline{y} \lor z) &\land \\
(\overline{a} \lor x \lor \overline{z}) &\land \\
(\overline{a} \lor \overline{y} \lor \overline{z})&
\end{aligned}
$$



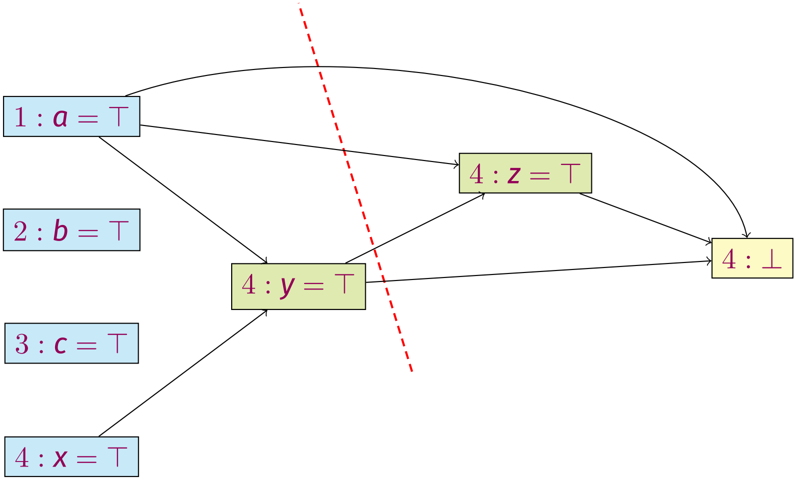
Implication graph

$$
\begin{aligned}
(a \lor b) &\land \\
(b \lor c) &\land \\
(\overline{a} \lor \overline{x} \lor y) &\land \\
(\overline{a} \lor x \lor z) &\land \\
(\overline{a} \lor \overline{y} \lor z) &\land \\
(\overline{a} \lor x \lor \overline{z}) &\land \\
(\overline{a} \lor \overline{y} \lor \overline{z}) &
\end{aligned}
$$



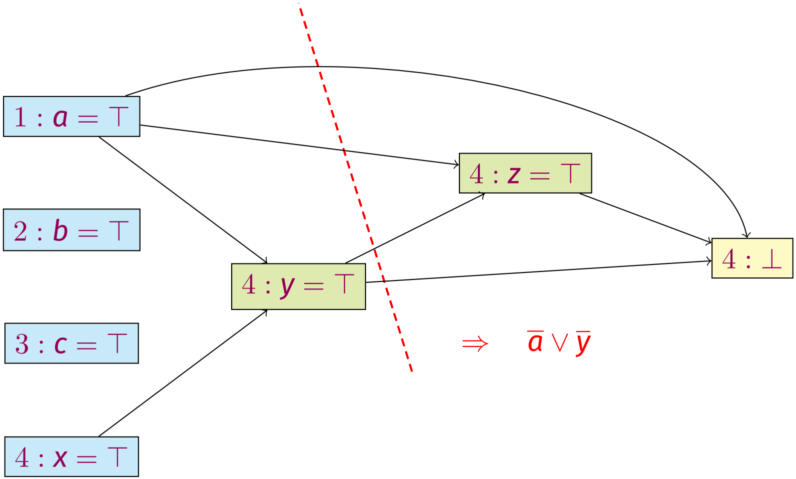
Implication graph

$$
\begin{aligned}
(a \vee b) &\wedge \\
(b \vee c) &\wedge \\
(\bar{a} \vee \bar{x} \vee y) &\wedge \\
(\bar{a} \vee x \vee z) &\wedge \\
(\bar{a} \vee \bar{y} \vee z) &\wedge \\
(\bar{a} \vee x \vee \bar{z}) &\wedge \\
(\bar{a} \vee \bar{y} \vee \bar{z})&
\end{aligned}
$$



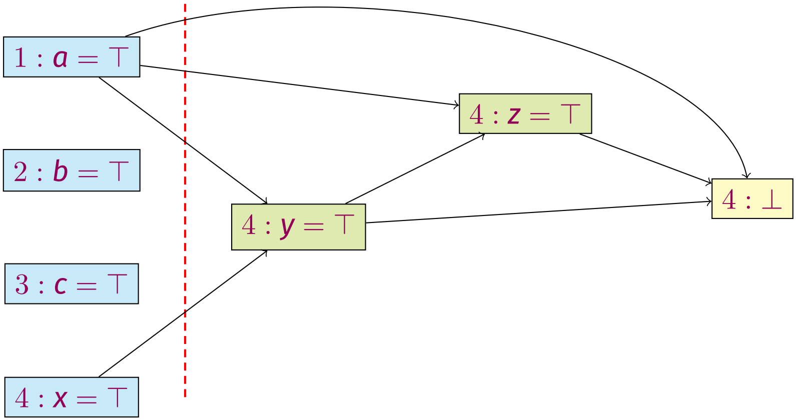
Implication graph

$$(a \vee b) \wedge$$
$$(b \vee c) \wedge$$
$$(\bar{a} \vee \bar{x} \vee y) \wedge$$
$$(\bar{a} \vee x \vee z) \wedge$$
$$(\bar{a} \vee \bar{y} \vee z) \wedge$$
$$(\bar{a} \vee x \vee \bar{z}) \wedge$$
$$(\bar{a} \vee \bar{y} \vee \bar{z})$$



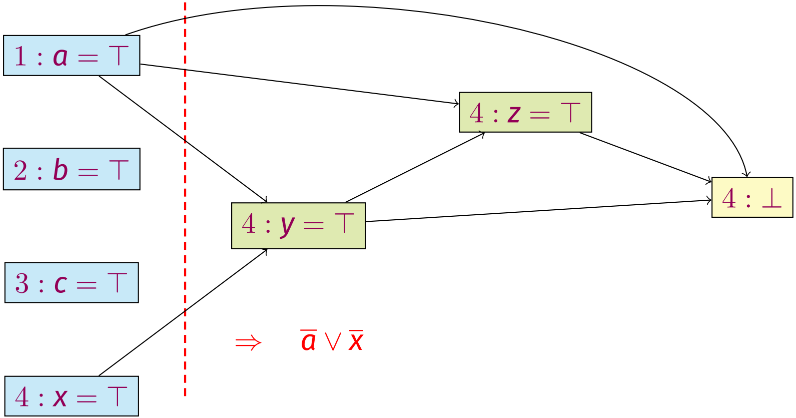
Implication graph

$$
\begin{aligned}
&(a \lor b) \land \\
&(b \lor c) \land \\
&(\overline{a} \lor \overline{x} \lor y) \land \\
&(\overline{a} \lor x \lor z) \land \\
&(\overline{a} \lor \overline{y} \lor z) \land \\
&(\overline{a} \lor x \lor \overline{z}) \land \\
&(\overline{a} \lor \overline{y} \lor \overline{z})
\end{aligned}
$$



Implication graph

$$
\begin{aligned}
(a \vee b) &\wedge \\
(b \vee c) &\wedge \\
(\overline{a} \vee \overline{x} \vee y) &\wedge \\
(\overline{a} \vee x \vee z) &\wedge \\
(\overline{a} \vee \overline{y} \vee z) &\wedge \\
(\overline{a} \vee x \vee \overline{z}) &\wedge \\
(\overline{a} \vee \overline{y} \vee \overline{z})&
\end{aligned}
$$



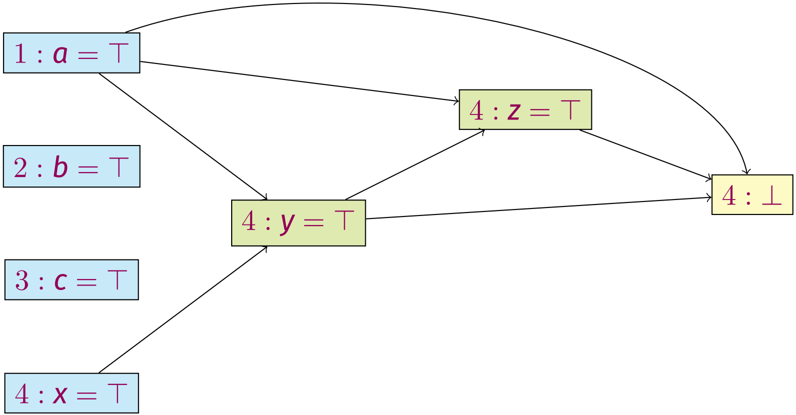
Unique Implication Point

Definition (UIP)

A vertex l in the implication graph is a Unique Implication Point (UIP) if all the paths from the ***last* decision** to **the conflict vertex** $\perp$ go through l.

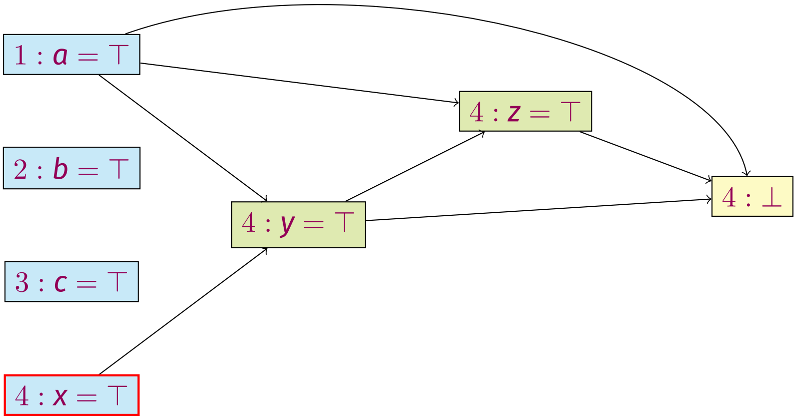
Implication graph

$$
\begin{aligned}
(a \vee b) &\wedge \\
(b \vee c) &\wedge \\
(\bar{a} \vee \bar{x} \vee y) &\wedge \\
(\bar{a} \vee x \vee z) &\wedge \\
(\bar{a} \vee \bar{y} \vee z) &\wedge \\
(\bar{a} \vee x \vee \bar{z}) &\wedge \\
(\bar{a} \vee \bar{y} \vee \bar{z}) &
\end{aligned}
$$



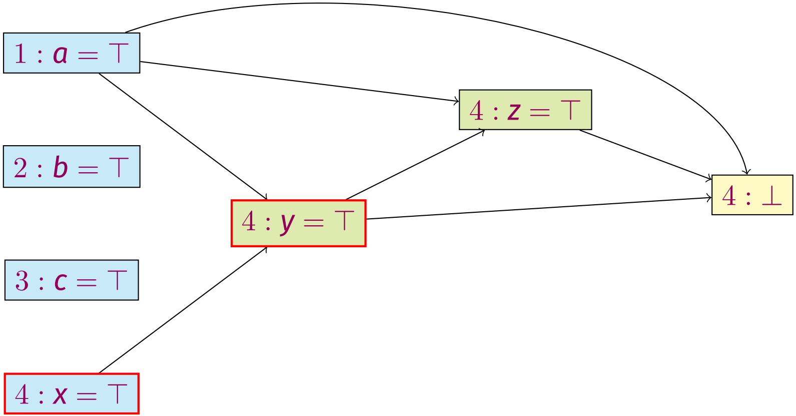
Implication graph

$$(a \vee b) \wedge$$
$$(b \vee c) \wedge$$
$$(\overline{a} \vee \overline{x} \vee y) \wedge$$
$$(\overline{a} \vee x \vee z) \wedge$$
$$(\overline{a} \vee \overline{y} \vee z) \wedge$$
$$(\overline{a} \vee x \vee \overline{z}) \wedge$$
$$(\overline{a} \vee \overline{y} \vee \overline{z})$$



Implication graph

$$
\begin{aligned}
&(a \vee b) \wedge \\
&(b \vee c) \wedge \\
&(\overline{a} \vee \overline{x} \vee y) \wedge \\
&(\overline{a} \vee x \vee z) \wedge \\
&(\overline{a} \vee \overline{y} \vee z) \wedge \\
&(\overline{a} \vee x \vee \overline{z}) \wedge \\
&(\overline{a} \vee \overline{y} \vee \overline{z})
\end{aligned}
$$



Unique Implication Point

Definition (UIP)

A vertex l in the implication graph is a Unique Implication Point (UIP) if all the paths from the ***last* decision** to **the conflict vertex** $\perp$ go through l.

Unique Implication Point

Definition (UIP)

A vertex l in the implication graph is a Unique Implication Point (UIP) if all the paths from the ***last* decision** to **the conflict vertex** $\perp$ go through l.

Definition (UIP cut)

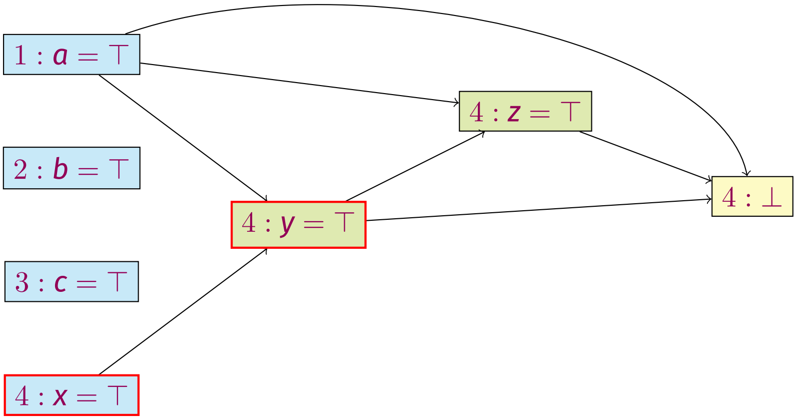
If l is a UIP, the corresponding **UIP cut** is $W = (A, B)$ where B contains the successors of l from which there is a path to $\perp$.

Note that:

- A UIP always exists: the last decision variable is itself a UIP.

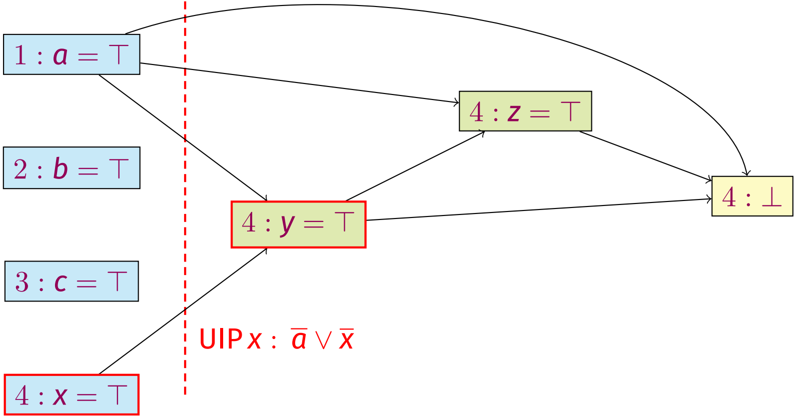
Implication graph

$$
\begin{aligned}
(a \vee b) \wedge \\
(b \vee c) \wedge \\
(\overline{a} \vee \overline{x} \vee y) \wedge \\
(\overline{a} \vee x \vee z) \wedge \\
(\overline{a} \vee \overline{y} \vee z) \wedge \\
(\overline{a} \vee x \vee \overline{z}) \wedge \\
(\overline{a} \vee \overline{y} \vee \overline{z})
\end{aligned}
$$



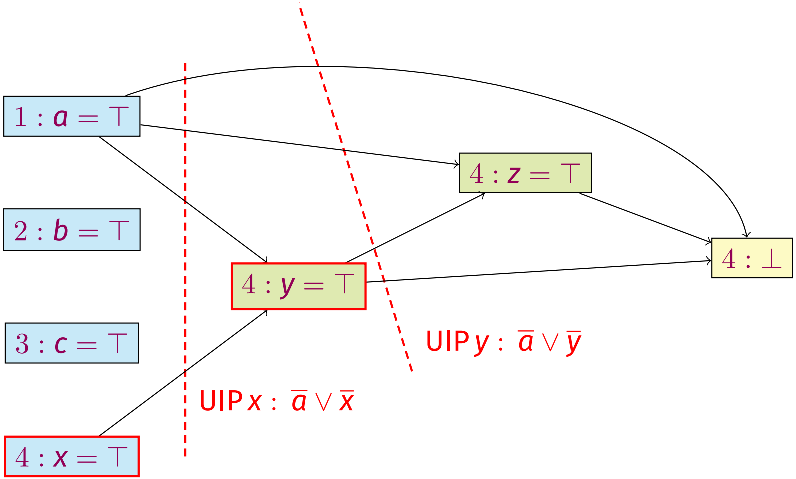
Implication graph

$$
\begin{aligned}
(a \vee b) \wedge \\
(b \vee c) \wedge \\
(\overline{a} \vee \overline{x} \vee y) \wedge \\
(\overline{a} \vee x \vee z) \wedge \\
(\overline{a} \vee \overline{y} \vee z) \wedge \\
(\overline{a} \vee x \vee \overline{z}) \wedge \\
(\overline{a} \vee \overline{y} \vee \overline{z})
\end{aligned}
$$



Implication graph

$$
\begin{aligned}
(a \vee b) &\wedge \\
(b \vee c) &\wedge \\
(\overline{a} \vee \overline{x} \vee y) &\wedge \\
(\overline{a} \vee x \vee z) &\wedge \\
(\overline{a} \vee \overline{y} \vee z) &\wedge \\
(\overline{a} \vee x \vee \overline{z}) &\wedge \\
(\overline{a} \vee \overline{y} \vee \overline{z})&
\end{aligned}
$$

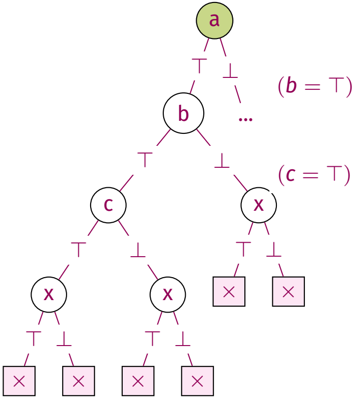


Backjumping

- Add the clause from the UIP cut to the original formula.
- Backtrack to the *second-highest* decision level of literals used in the clause.
- Since UIP has only one literal at the highest decision level, unit propagation will happen.

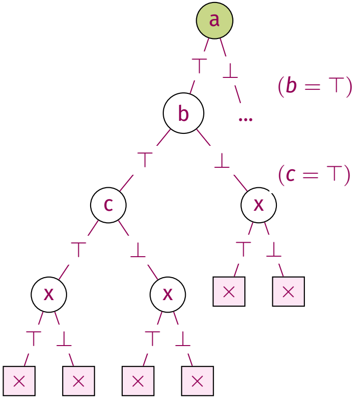
Tree search

$$(a \vee b) \wedge$$
$$(b \vee c) \wedge$$
$$(\overline{a} \vee \overline{x} \vee y) \wedge$$
$$(\overline{a} \vee x \vee z) \wedge$$
$$(\overline{a} \vee \overline{y} \vee z) \wedge$$
$$(\overline{a} \vee x \vee \overline{z}) \wedge$$
$$(\overline{a} \vee \overline{y} \vee \overline{z})$$



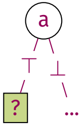
Tree search

$$
\begin{aligned}
&(a \vee b) \wedge \\
&(b \vee c) \wedge \\
&(\overline{a} \vee \overline{x} \vee y) \wedge \\
&(\overline{a} \vee x \vee z) \wedge \\
&(\overline{a} \vee \overline{y} \vee z) \wedge \\
&(\overline{a} \vee x \vee \overline{z}) \wedge \\
&(\overline{a} \vee \overline{y} \vee \overline{z}) \wedge \\
&(\overline{a} \vee \overline{y})
\end{aligned}
$$



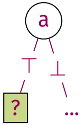
Tree search

$$
\begin{aligned}
&(a \vee b) \wedge \\
&(b \vee c) \wedge \\
&(\overline{a} \vee \overline{x} \vee y) \wedge \\
&(\overline{a} \vee x \vee z) \wedge \\
&(\overline{a} \vee \overline{y} \vee z) \wedge \\
&(\overline{a} \vee x \vee \overline{z}) \wedge \\
&(\overline{a} \vee \overline{y} \vee \overline{z}) \wedge \\
&(\overline{a} \vee \overline{y})
\end{aligned}
$$



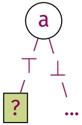
Tree search

$$\cancel{(a \vee b)} \wedge$$
$$(b \vee c) \wedge$$
$$(\overline{a} \vee \overline{x} \vee y) \wedge$$
$$(\overline{a} \vee x \vee z) \wedge$$
$$(\overline{a} \vee \overline{y} \vee z) \wedge$$
$$(\overline{a} \vee x \vee \overline{z}) \wedge$$
$$(\overline{a} \vee \overline{y} \vee \overline{z}) \wedge$$
$$(\overline{a} \vee \overline{y})$$



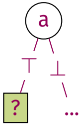
Tree search

$$
\begin{aligned}
&\cancel{(a \lor b)} \land \\
&(b \lor c) \land \\
&(\overline{a} \lor \overline{x} \lor y) \land \\
&(\overline{a} \lor x \lor z) \land \\
&\cancel{(\overline{a} \lor \overline{y} \lor z)} \land \\
&(\overline{a} \lor x \lor \overline{z}) \land \\
&\cancel{(\overline{a} \lor \overline{y} \lor \overline{z})} \land \\
&\cancel{(\overline{a} \lor \overline{y})}
\end{aligned}
$$



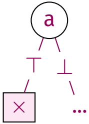
Tree search

$$
\begin{aligned}
\cancel{(a \vee b)} \wedge \\
(b \vee c) \wedge \\
\cancel{(\overline{a} \vee \overline{x} \vee \overline{y})} \wedge \\
(\overline{a} \vee x \vee z) \wedge \\
\cancel{(\overline{a} \vee \overline{y} \vee z)} \wedge \\
(\overline{a} \vee x \vee \overline{z}) \wedge \\
\cancel{(\overline{a} \vee \overline{y} \vee \overline{z})} \wedge \\
\cancel{(\overline{a} \vee \overline{y})}
\end{aligned}
$$



Tree search

$$
\begin{aligned}
&\cancel{(a \vee b)} \wedge \\
&(b \vee c) \wedge \\
&\cancel{(\bar{a} \vee \bar{x} \vee y)} \wedge \\
&(\bar{a} \vee x \vee z) \wedge \\
&\cancel{(\bar{a} \vee \bar{y} \vee z)} \wedge \\
&(\bar{a} \vee x \vee \bar{z}) \wedge \\
&\cancel{(\bar{a} \vee \bar{y} \vee \bar{z})} \wedge \\
&\cancel{(\bar{a} \vee \bar{y})}
\end{aligned}
$$



Implication graph

$$
\begin{aligned}
(a \vee b) \wedge \\
(b \vee c) \wedge \\
(\overline{a} \vee \overline{x} \vee y) \wedge \\
(\overline{a} \vee x \vee z) \wedge \\
(\overline{a} \vee \overline{y} \vee z) \wedge \\
(\overline{a} \vee x \vee \overline{z}) \wedge \\
(\overline{a} \vee \overline{y} \vee \overline{z}) \wedge \\
(\overline{a} \vee \overline{y})
\end{aligned}
$$

$\boxed{1 : a = \top}$

Implication graph

$$
\begin{aligned}
(a \vee b) &\wedge \\
(b \vee c) &\wedge \\
(\overline{a} \vee \overline{x} \vee y) &\wedge \\
(\overline{a} \vee x \vee z) &\wedge \\
(\overline{a} \vee \overline{y} \vee z) &\wedge \\
(\overline{a} \vee x \vee \overline{z}) &\wedge \\
(\overline{a} \vee \overline{y} \vee \overline{z}) &\wedge \\
(\overline{a} \vee \overline{y}) &
\end{aligned}
$$

$$\boxed{1 : a = \top} \longrightarrow \boxed{1 : y = \bot}$$

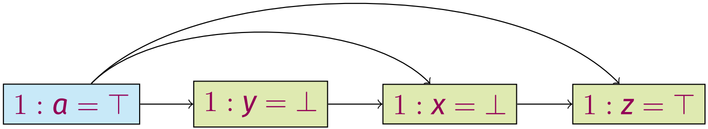
Implication graph

$$
\begin{aligned}
(a \vee b) &\wedge \\
(b \vee c) &\wedge \\
(\overline{a} \vee \overline{x} \vee y) &\wedge \\
(\overline{a} \vee x \vee z) &\wedge \\
(\overline{a} \vee \overline{y} \vee z) &\wedge \\
(\overline{a} \vee x \vee \overline{z}) &\wedge \\
(\overline{a} \vee \overline{y} \vee \overline{z}) &\wedge \\
\color{red}(\overline{a} \vee \overline{y})&
\end{aligned}
$$

$$
\begin{tikzcd}
\boxed{1 : a = \top} \arrow[r] \arrow[rr, bend left] & \boxed{1 : y = \bot} \arrow[r] & \boxed{1 : x = \bot}
\end{tikzcd}
$$

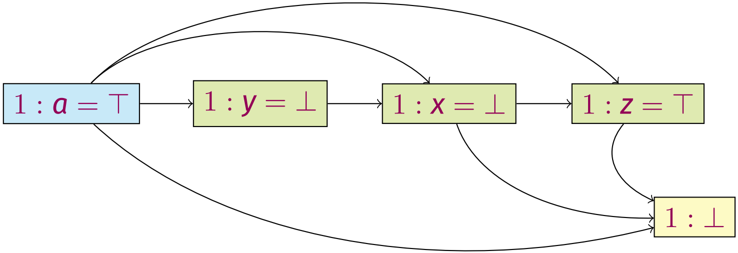
Implication graph

$$
\begin{aligned}
&(a \vee b) \wedge \\
&(b \vee c) \wedge \\
&(\overline{a} \vee \overline{x} \vee y) \wedge \\
&(\overline{a} \vee x \vee z) \wedge \\
&(\overline{a} \vee \overline{y} \vee z) \wedge \\
&(\overline{a} \vee x \vee \overline{z}) \wedge \\
&(\overline{a} \vee \overline{y} \vee \overline{z}) \wedge \\
&(\overline{a} \vee \overline{y})
\end{aligned}
$$



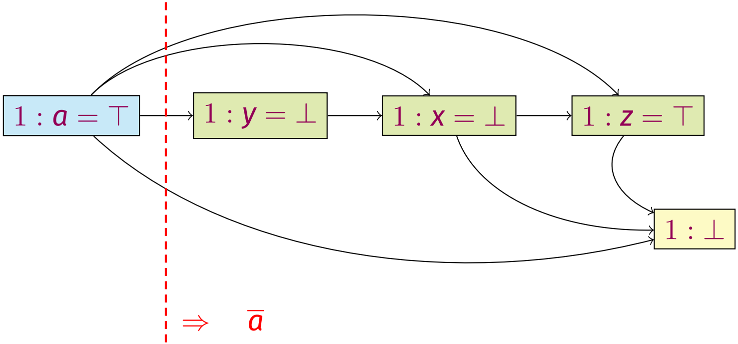
Implication graph

$$(a \vee b) \wedge$$
$$(b \vee c) \wedge$$
$$(\overline{a} \vee \overline{x} \vee y) \wedge$$
$$(\overline{a} \vee x \vee z) \wedge$$
$$(\overline{a} \vee \overline{y} \vee z) \wedge$$
$$(\overline{a} \vee x \vee \overline{z}) \wedge$$
$$(\overline{a} \vee \overline{y} \vee \overline{z}) \wedge$$
$$(\overline{a} \vee \overline{y})$$



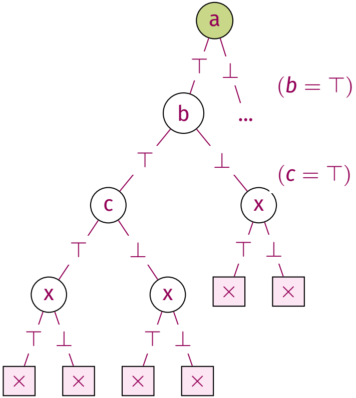
Implication graph

$$
\begin{aligned}
&(a \vee b) \wedge \\
&(b \vee c) \wedge \\
&(\overline{a} \vee \overline{x} \vee y) \wedge \\
&(\overline{a} \vee x \vee z) \wedge \\
&(\overline{a} \vee \overline{y} \vee z) \wedge \\
&(\overline{a} \vee x \vee \overline{z}) \wedge \\
&(\overline{a} \vee \overline{y} \vee \overline{z}) \wedge \\
&(\overline{a} \vee \overline{y})
\end{aligned}
$$



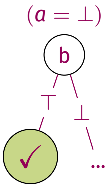
Tree search

$$(a \vee b) \wedge$$
$$(b \vee c) \wedge$$
$$(\overline{a} \vee \overline{x} \vee y) \wedge$$
$$(\overline{a} \vee x \vee z) \wedge$$
$$(\overline{a} \vee \overline{y} \vee z) \wedge$$
$$(\overline{a} \vee x \vee \overline{z}) \wedge$$
$$(\overline{a} \vee \overline{y} \vee \overline{z})$$



Tree search

$$
\begin{aligned}
(a \vee b) \wedge \\
(b \vee c) \wedge \\
(\overline{a} \vee \overline{x} \vee y) \wedge \\
(\overline{a} \vee x \vee z) \wedge \\
(\overline{a} \vee \overline{y} \vee z) \wedge \\
(\overline{a} \vee x \vee \overline{z}) \wedge \\
(\overline{a} \vee \overline{y} \vee \overline{z}) \wedge \\
\color{red}{(\overline{a} \vee \overline{y})} \wedge \\
\color{red}{(\overline{a})}
\end{aligned}
$$



Nice properties

- Adding a UIP clause will cause new unit propagation to happen in the search tree (the search tree becomes smaller).
- The same assignment and implication graph can never be reached again.
- … even if you re-start the search tree, change the branching.

Conflict analysis for MILPs?

- Has been tried in SCIP.
  - Achterberg, T. (2007). Conflict analysis in mixed integer programming. Discrete Optimization, 4(1), 4-20.
  - Sandholm, T., & Shields, R. (2006). Nogood learning for mixed integer programming. In Workshop on Hybrid Methods and Branching Rules in Combinatorial Optimization, Montréal.
  - Witzig, J., Berthold, T., & Heinz, S. (2017). Experiments with conflict analysis in mixed integer programming. In Integration of AI and OR Techniques in Constraint Programming (CPAIOR 2017).
- Logical conflict analysis can be generalized to incompatible combinations of bound changes for MILPs, but linearization of combinations of bound changes involving general integer and continuous variables might be weak.
- Not sure if it's much used in commercial solvers.