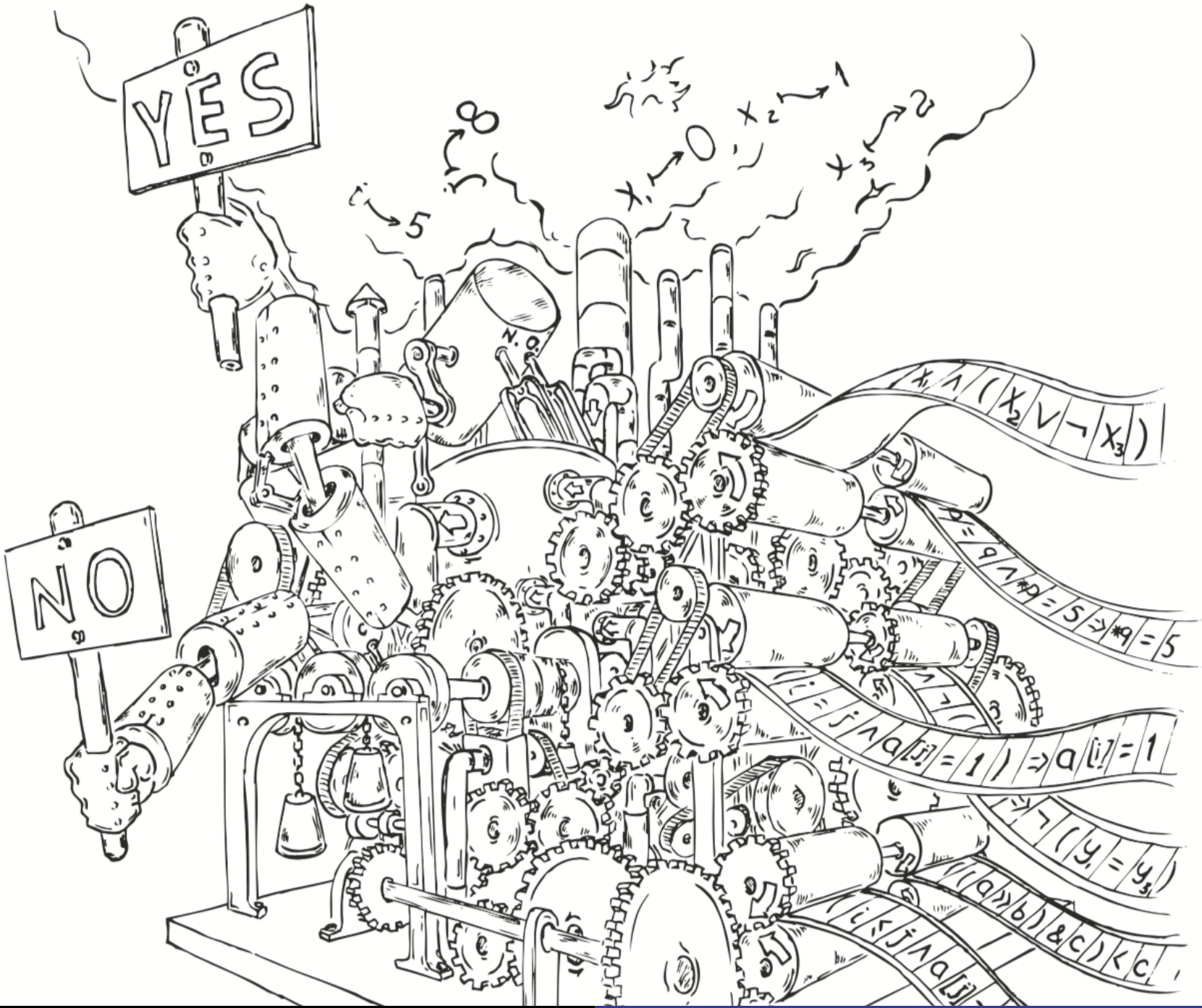

YES
NO

SAT/SMT solvers
2. Conflict-Driven Clause Learning algorithm

Roman Kholin

Lalambda

Tbilisi, 2023

Obvious algorithm

- Brute force
- $O(2^n)$

Definitions

State of a clause under an assignment

A clause is satisfied if one or more of its literals are satisfied, conflicting if all of its literals are assigned but not satisfied, unit if it is not satisfied and all but one of its literals are assigned, and unresolved otherwise.

$$\{x_1 \mapsto 1,\ x_2 \mapsto 0,\ x_4 \mapsto 1\}\ ,$$

$(x_1 \vee x_3 \vee \neg x_4)$	is satisfied,
$(\neg x_1 \vee x_2)$	is conflicting,
$(\neg x_1 \vee \neg x_4 \vee x_3)$	is unit,
$(\neg x_1 \vee x_3 \vee x_5)$	is unresolved.

The clause $C := (\neg x_1 \vee \neg x_4 \vee x_3)$ and the partial assignment $\{x_1 \to 1, x_4 \to 1\}$ imply the assignment x_3 and $\mathit{Antecedent}(x_3) = C$

Definitions

Implication graph

An implication graph is a labeled directed acyclic graph G(V, E), where:

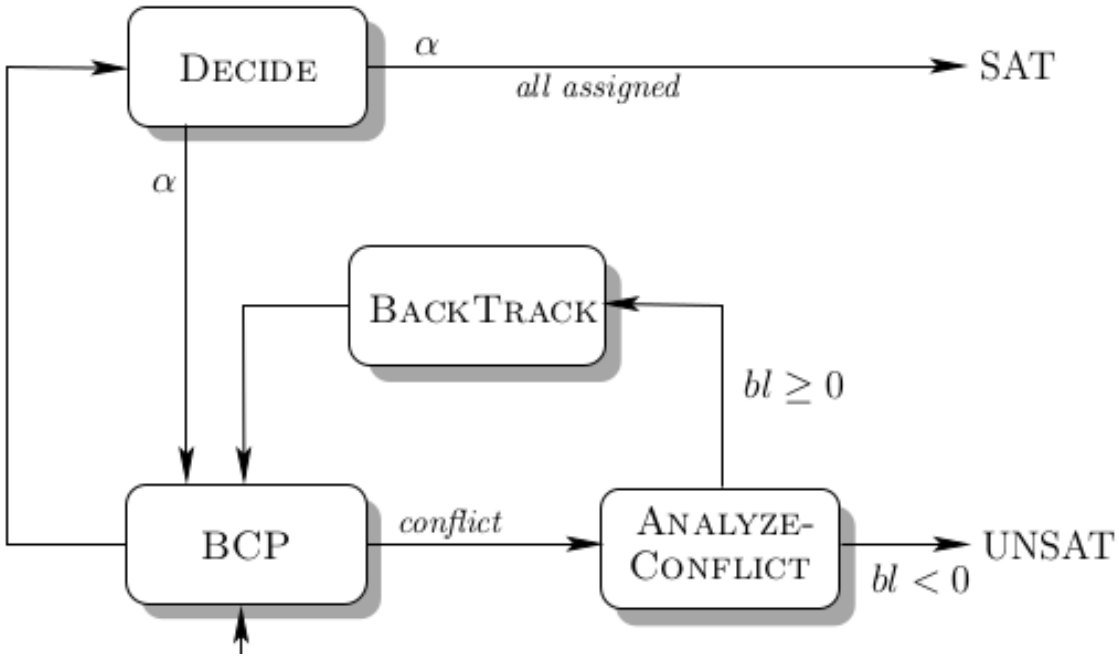
- V represents the literals of the current partial assignment (we refer to a node and the literal that it represents interchangeably). Each node is labeled with the literal that it represents and the decision level at which it entered the partial assignment
- E with $E = \{(v_i, v_j) | v_i, v_j \in V, \neg v_i \neg Antecedent(v_j)\}$ denotes the set of directed edges where each edge (v_i, v_j) is labeled with Antecedent(v_j)
- G can also contain a single conflict node labeled with k and incoming edges $\{(v, k) | \neg v \in c\}$ labeled with c for some conflicting clause c

Example

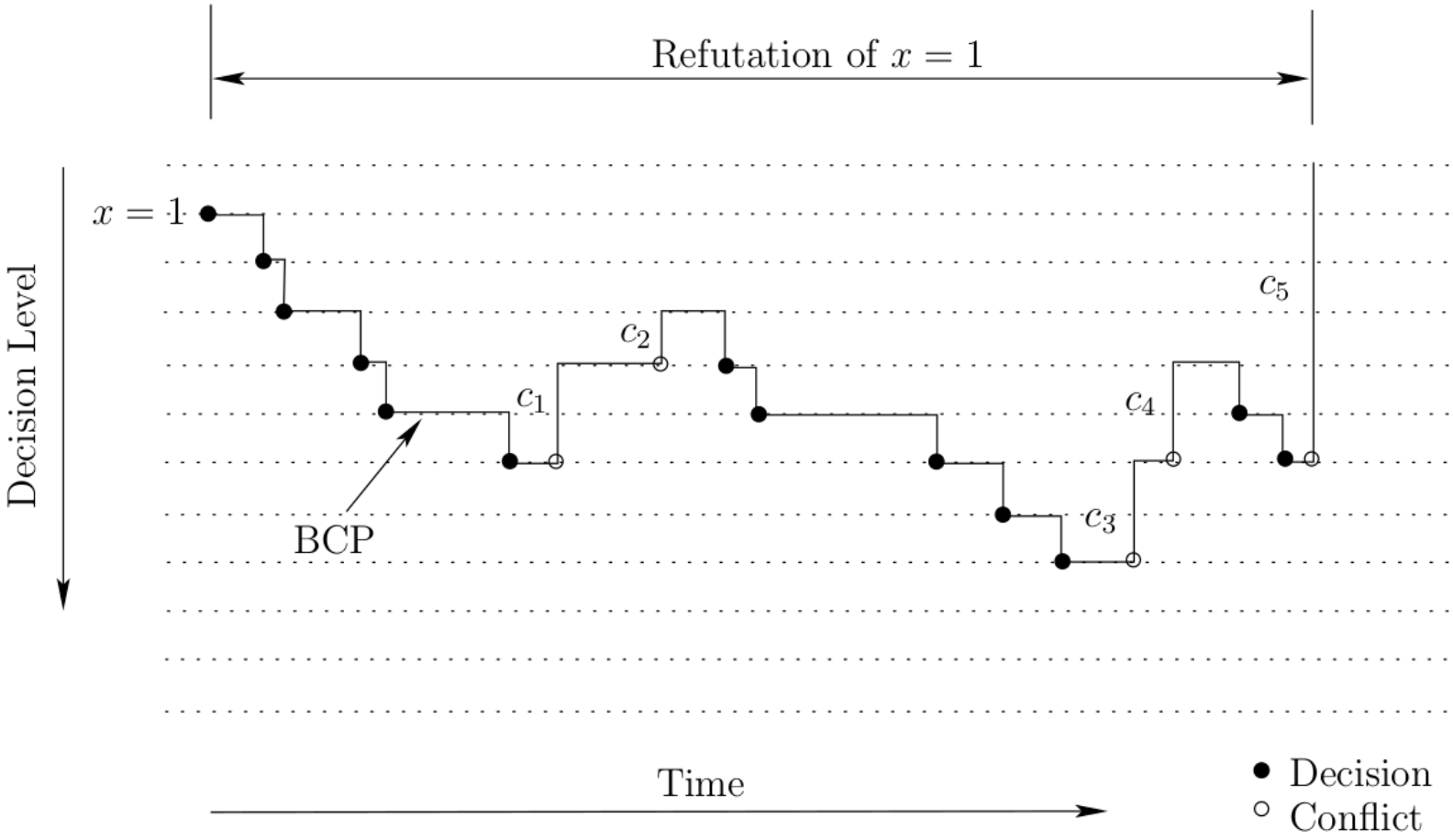
$c_1 = (\neg x_1 \vee x_2)$
$c_2 = (\neg x_1 \vee x_3 \vee x_5)$
$c_3 = (\neg x_2 \vee x_4)$
$c_4 = (\neg x_3 \vee \neg x_4)$
$c_5 = (x_1 \vee x_5 \vee \neg x_2)$
$c_6 = (x_2 \vee x_3)$
$c_7 = (x_2 \vee \neg x_3)$
$c_8 = (x_6 \vee \neg x_5)$
Let x_1@6 and $\neg x_5$@3

Conflict-driven clause learning

```
1. function CDCL
2.     while (TRUE) do
3.         while (BCP() = "conflict") do
4.             backtrack-level := ANALYZE-CONFLICT();
5.             if backtrack-level < 0 then return "Unsatisfiable";
6.             BackTrack(backtrack-level);
7.         if ¬DECIDE() then return "Satisfiable";
```



It is never the case that the solver enters decision level dl again with the same partial assignment


Refutation of $x = 1$
$x = 1$
Decision Level
c_1
c_2
c_3
c_4
c_5
BCP
Time
Decision
Conflict

UIP

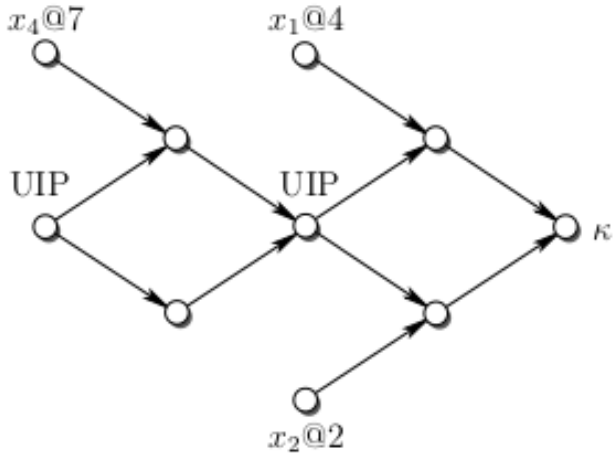
Unique Implication Point (UIP)

Given a partial conflict graph corresponding to the decision level of the conflict, a unique implication point (UIP) is any node other than the conflict node that is on all paths from the decision node to the conflict node

First UIP

A first UIP is a UIP that is closest to the conflict node

UIP



Binary resolution and related terms

$$\frac{(a_1 \vee \ldots \vee a_n \vee \beta) \qquad (b_1 \vee \ldots \vee b_m \vee \neg\beta)}{(a_1 \vee \ldots \vee a_n \vee b_1 \vee \ldots \vee b_m)}$$

where

$a_1, \ldots, a_n, b_1, \ldots, b_m$ are literals and β is a variable. The variable β is called the resolution variable. The clauses $(a_1 \vee \cdots \vee a_n \vee \beta)$ and $(b_1 \vee \cdots \vee b_m \vee (\neg\beta))$ are the resolving clauses, and $(a_1 \vee \cdots \vee a_n \vee b_1 \cdots \vee b_m)$ is the resolvent clause

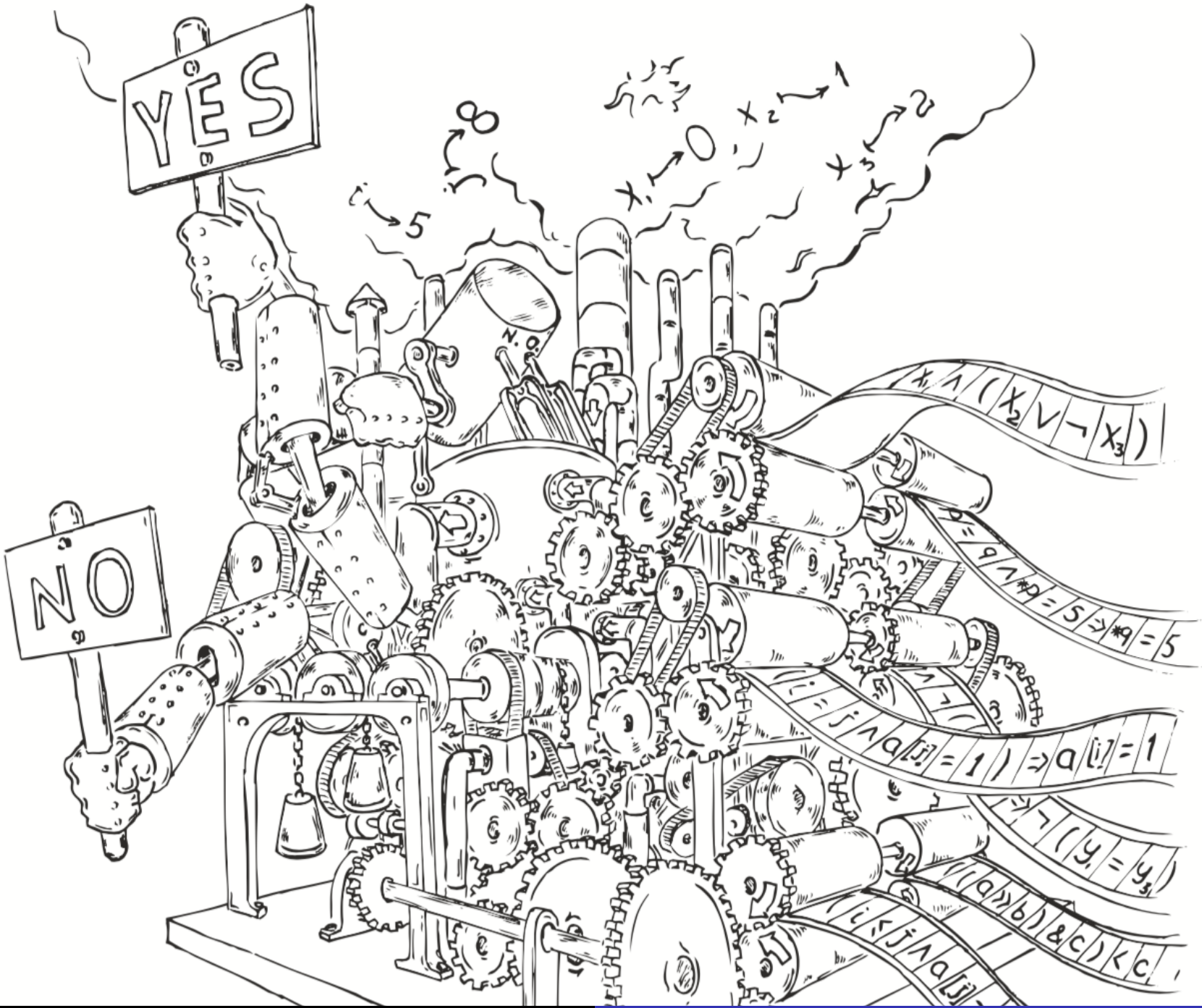
Analyze conflict

1. **if** *current-decision-level* $= 0$ **then return** -1;
2. $cl :=$ *current-conflicting-clause*;
3. **while** $(\neg$STOP-CRITERION-MET$(cl))$ **do**
4. $\quad lit :=$ LAST-ASSIGNED-LITERAL(cl);
5. $\quad var :=$ VARIABLE-OF-LITERAL(lit);
6. $\quad ante :=$ ANTECEDENT(lit);
7. $\quad cl :=$ RESOLVE$(cl, ante, var)$;
8. add-clause-to-database(cl);
9. **return** clause-asserting-level(cl); ▷ 2nd highest decision level in cl

$c_1 = (\neg x_4 \vee x_2 \vee x_5)$
$c_2 = (\neg x_4 \vee x_{10} \vee x_6)$
$c_3 = (\neg x_5 \vee \neg x_6 \vee \neg x_7)$
$c_4 = (\neg x_6 \vee x_7)$

Decision Heuristics

- Jeroslow–Wang: compute for each literal l $J(l) = \Sigma_{w \in B, l \in w} 2^{-|w|}$
- Dynamic Largest Individual Sum (DLIS): at each decision level, choose the unassigned literal that satisfies the largest number of currently unsatisfied clauses
- Variable State Independent Decaying Sum (VSIDS): A new conflict clause, adds 1 to the score of each literal that appears in it. Periodically divide all scores by 2.


YES
NO