The Overall Picture

<table>
<thead>
<tr>
<th>Application</th>
<th>System + Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Algorithm + Implementation</td>
</tr>
<tr>
<td>Algorithm</td>
<td>Calculus + Strategy</td>
</tr>
<tr>
<td>Calculus</td>
<td>Logic + States + Rules</td>
</tr>
<tr>
<td>Logic</td>
<td>Syntax + Semantics</td>
</tr>
</tbody>
</table>
Conflict Driven Clause Learning (CDCL)

The CDCL calculus tests satisfiability of a finite set $N$ of propositional clauses.

I assume that $\perp \notin N$ and that the clauses in $N$ do not contain duplicate literal occurrences. Furthermore, duplicate literal occurrences are always silently removed during rule applications of the calculus. (Exhaustive Condensation.)
The CDCL calculus explicitly builds a candidate model for a clause set. If such a sequence of literals $L_1, \ldots, L_n$ satisfies the clause set $N$, it is done. If not, there is a false clause $C \in N$ with respect to $L_1, \ldots, L_n$.

Now instead of just backtracking through the literals $L_1, \ldots, L_n$, CDCL generates in addition a new clause, called *learned clause* via resolution, that actually guarantees that the subsequence of $L_1, \ldots, L_n$ that caused $C$ to be false will not be generated anymore.

This causes CDCL to be exponentially more powerful in proof length than its predecessor DPLL or Tableau.
A CDCL problem state is a five-tuple \((M; N; U; k; D)\) where

- \(M\) is a sequence of annotated literals, called a *trail*,
- \(N\) and \(U\) are sets of clauses,
- \(k \in \mathbb{N}\), and
- \(D\) is a non-empty clause or \(\top\) or \(\bot\), called the *mode* of the state.

The set \(N\) is initialized by the problem clauses where the set \(U\) contains all newly learned clauses that are consequences of clauses from \(N\) derived by resolution.
Modes of CDCL States

\[(\epsilon; N; \emptyset; 0; \top)\] is the start state for some clause set \(N\)

\[(M; N; U; k; \top)\] is a final state, if \(M \models N\) and all literals from \(N\) are defined in \(M\)

\[(M; N; U; k; \bot)\] is a final state, where \(N\) has no model

\[(M; N; U; k; \top)\] is an intermediate model search state if \(M \not\models N\)

\[(M; N; U; k; D)\] is a backtracking state if \(D \not\in \{\top, \bot\}\)
The Role of Levels

Literals in $L \in M$ are either annotated with a number, a level, i.e., they have the form $L^k$ meaning that $L$ is the $k^{th}$ guessed decision literal, or they are annotated with a clause that forced the literal to become true.

A literal $L$ is of level $k$ with respect to a problem state $(M; N; U; j; C)$ if $L$ or $\text{comp}(L)$ occurs in $M$ and the first decision literal left from $L$ ($\text{comp}(L)$) in $M$ is annotated with $k$. If there is no such decision literal then $k = 0$.

A clause $D$ is of level $k$ with respect to a problem state $(M; N; U; j; C)$ if $k$ is the maximal level of a literal in $D$. 
CDCL Rules

**Propagate** \((M; N; U; k; \top) \Rightarrow_{\text{CDCL}} (ML^{C \lor L}; N; U; k; \top)\)

provided \(C \lor L \in (N \cup U)\), \(M \models \neg C\), and \(L\) is undefined in \(M\)

**Decide** \((M; N; U; k; \top) \Rightarrow_{\text{CDCL}} (ML^{k+1}; N; U; k + 1; \top)\)

provided \(L\) is undefined in \(M\)

**Conflict** \((M; N; U; k; \top) \Rightarrow_{\text{CDCL}} (M; N; U; k; D)\)

provided \(D \in (N \cup U)\) and \(M \models \neg D\)
Skip \[ (ML^{C \lor L}; N; U; k; D) \Rightarrow_{CDCL} (M; N; U; k; D) \]
provided \( D \not\in \{\top, \bot\} \) and \( \text{comp}(L) \) does not occur in \( D \)

Resolve \[ (ML^{C \lor L}; N; U; k; D \lor \text{comp}(L)) \Rightarrow_{CDCL} (M; N; U; k; D \lor C) \]
provided \( D \) is of level \( k \)

Backtrack \[ (M_1 K^{i+1} M_2; N; U; k; D \lor L) \Rightarrow_{CDCL} (M_1 L^{D \lor L}; N; U \cup \{D \lor L\}; i; \top) \]
provided \( L \) is of level \( k \) and \( D \) is of level \( i \).

Restart \[ (M; N; U; k; \top) \Rightarrow_{CDCL} (\epsilon; N; U; 0; \top) \]
provided \( M \not\models N \)

Forget \[ (M; N; U \cup \{C\}; k; \top) \Rightarrow_{CDCL} (M; N; U; k; \top) \]
provided \( M \not\models N \)