Clause representation of CNF formulas

- Clause: set of literals (disjunction).

$$\{A, B\} \text{ stands for } (A \vee B).$$

- Formula: set of clauses (conjunction).

$$\{\{A, B\}, \{\neg A, B\}\} \text{ stands for } ((A \vee B) \wedge (\neg A \vee B)).$$

- Block: set of formulas (disjunction).

$$\{F, G\} \text{ stands for } (F \vee G).$$

The empty clause stands for **false**.
The empty formula stands for **true**.
The empty block stands for **false**.

The DPLL algorithm

- Developed by Davis, Putman, Loveland und Logemann
- Basis for the most efficient of today's solvers.
- Rules for transforming of blocks.
- If block B is transformed into B' then:

$$B \text{ is satisfiable (contains a satisfiable formula)}$$
$$\Longleftrightarrow$$
$$B' \text{ is satisfiable (contains a satisfiable formula)}$$

- Using the rules we construct a sequence of blocks called a derivation.
- The first blocks contains only the formula to be checked.
- The formula is satisfiable iff the derivation ends with a block containing the empty formula.
- The formula is unsatisfiable iff the derivation ends with a block in which every formula contains the empty clause.

The rules

Simplification rules

- Reduce the number of clauses.
- Diverse variants of the algorithm which eliminate some rules or add others.
- The simplification rules are not compliting (i.e., there are formulas for which they alone cannot decide satisfiability).

Splitting rule

- Increases the number of formulas.
- Guarantees completeness.

Simplification rules

- One-literal rule:
  Pick a formula of the form $F = F' \cup \{L\}$.
  Remove all clauses of F containing L.
  Remove all occurrences of $\overline{L}$ in the remaining clauses.
- Pure-literal rule:
  Pick a formula such that L does not occur in any clause of F.
  Remove all clauses containing $\overline{L}$.
- Subsumption rule:
  Pick a formula containing two clauses C, C' such that $C \subseteq C'$.
  Remove C'.
- Clean-up rule:
  Remove all clauses of the form $C \cup \{L, \overline{L}\}$.

Splitting rule

Pick an atomic formula A occurring in some formula F.
Replace F by $F \cup \{A\}$ und $F \cup \{\neg A\}$.

Notice that the rule increases the number of formulas by 1.

Derivations

- A derivation (from F) is a sequence $\{F\}, B_1, B_2 \ldots$ of blocks constructed using the rules.
- A derivation is maximal if it is infinite or cannot be extended with a new block, i.e., no rule can be applied to its last block.
- A derivation is successful if it ends with a block containing the empty formula.
- A derivation is unsuccessful if it ends with a block in which every formula contains the empty clause.

Correctness

Lemma: Let $B_0, B_1, \ldots, B_n$ be a derivation.
For every $0 \leq i \leq n-1$ the block B_i contains a satisfiable formula iff B_{i+1} contains a satisfiable formula.

Lemma: Every maximal derivation is finite.

Lemma: If F is satisfiable, then every maximal derivation from F is successful.

Lemma: If F is unsatisfiable, then every maximal derivation from F is unsuccessful.