Algorithm DPLL
Input: A set of clauses $\Phi$.
Output: A Truth Value.

function DPLL($\Phi$)

    if $\Phi$ is a consistent set of literals
        then return true;
    if $\Phi$ contains an empty clause
        then return false;
    for every unit clause $l$ in $\Phi$
        $\Phi \leftarrow$ unit-propagate($l$, $\Phi$);
    for every literal $l$ that occurs pure in $\Phi$
        $\Phi \leftarrow$ pure-literal-assign($l$, $\Phi$);
    $l \leftarrow$ choose-literal($\Phi$);
    return DPLL($\Phi \land l$) or DPLL($\Phi \land \neg l$);

Davis–Putnam–Logemann–Loveland (DPLL) algorithm

DPLL is a complete, backtracking-based search algorithm for deciding the satisfiability of propositional logic formulae in conjunctive normal form, i.e. for solving the CNF-SAT problem.
Where:

- `unit-propagate(l, \Phi)` and `pure-literal-assign(l, \Phi)` are functions that return the result of applying unit propagation and the pure literal rule, respectively, to the literal \( l \) and the formula \( \Phi \).

  They replace every occurrence of \( l \) with "true" and every occurrence of \( \text{not} \) with "false" in the formula \( \Phi \), and simplify the resulting formula.

- \( \Phi \land l \) denotes the simplified result of substituting "true" for \( l \) in \( \Phi \).