

PART 1 DPLL Search:

Step 1: Decide x = 1

Partial assignment: x = 1.

Simplifying the formula under x = 1:

1) (\neg x \vee y \vee z) 🡪 (false \vee y \vee z) = (y \vee z)

2) (x \vee \neg y) 🡪 (true \vee \neg y) = true.

- This clause is satisfied and can be dropped.

3) (\neg x \vee \neg y \vee \neg z) 🡪 (false \vee \neg y \vee \neg z) = (\neg y \vee \neg z).

4) (\neg y \vee z) remains (\neg y \vee z).

So the formula remains:

(y \vee z) \wedge (\neg y \vee \neg z) \wedge (\neg y \vee z).

Since no unit clauses 🡪 no propagation

Step 2: Decide y = 1

Partial assignment: x = 1, y = 1.

Simplifying the remaining clauses:

1) (y \vee z) 🡪 (true \vee z) = true.

Satisfied, so it can be deleted.

2) (\neg y \vee \neg z) 🡪 (false \vee \neg z) = (\neg z).

3) (\neg y \vee z) 🡪 (false \vee z) = (z).

The formula remains:

(\neg z) \wedge (z).

This is a contradiction, therefore the assignment x=1, y=1 cannot be considered a solution

Step 3: Backtrack

Since y=1 led to a contradiction, DPLL backtracks and flips y: x=1 and y=0

Back to the simplification of the formula with the new condition:

1) (\neg x \vee y \vee z) 🡪 (false \vee 0 ∨\vee z) = (z).

- A unit clause forcing z=1.

2) (x \vee /neg y) 🡪 (true \vee true) = true 🡪 Satisfied

3) (\neg x \vee \neg y \vee \neg z) 🡪 (false \vee true \vee \neg z) = true.

4) (\neg y 7vee z) 🡪 (true \vee z) = true.

No conflict. Setting z=1 completes the assignment 🡪 x=1, y=0, z=1 satisfies all clauses

PART 2 WATCHING THE CLAUSE (\neg x \vee \neg y \vee \neg z)

Considering the clause C = (\neg x \vee \neg y \vee \neg z).

Considering the initially state \neg x and \neg y.

Step 1: Initialization

1) Watched literals: \neg x and \neg y

2) Unwatched literal: \neg z

Since both \neg x and \neg y are unassigned at this point, nothing propagates.

Step 2: After Decide x=1

1) \neg x becomes false (because x=1).

2) Trying to move its watch to a different literal in the clause.

3) \neg y remains unassigned, so it is still watched.

4) \neg z is unassigned, so moving the watch from \neg x to \neg z.

Hence the watched literals are now \neg y and \neg z.

No unit propagation occurs because there are still two unassigned literals being watched.

Step 3: After Decide y=1

1) \neg y becomes false (because y=1).

2) Now the clause has watched literals \neg y (false) and \neg z (unassigned).

3) We check if we can move the watch from \neg y back to \neg x,

but \neg x is already false (x=1).

4) Thus there is no other literal to watch, leaving only \neg z unassigned

in the clause.

This makes (\neg x \vee \neg y \vee \neg z) unit on \neg z, forcing z=0.

Conflict

1) Another clause (\neg y \vee z) under y=1 becomes (false \vee z) = (z),

forcing z=1.

2) Direct contradiction: one clause forces z=0, another z=1.

3) DPLL detects this conflict, backtracks on y=1, and discovers the

satisfying assignment x=1, y=0, z=1.