Define Version of DPLL

1 2 3 4 5 6 7 8

C0 = {a v r, b v c, ¬a v ¬p, ¬a v ¬q, ¬b v p, ¬b v ¬r, ¬c v q, ¬c v ¬r }

( 0 | C0 | \* )

Decide a

( a(d) | C0 | \* )

Propagate

( a[d] , not(p)[3]| C0 | \* )

Propagate

( a[d] , not(p)[3], not(q)[4] | C0 | \* )

Propagate

( a[d] , not(p)[3], not(q)[4], not(b)[5] | C0 | \* )

Propagate

( a[d] , not(p)[3], not(q)[4], not(b)[5], not(c)[7] | C0 | \* )

Conflict

( a[d] , not(p)[3], not(q)[4], not(b)[5], not(c)[7] | C0 | [2] )

Resolution between [7] and [2]

not(c) v q b v c

---------------------

q v b

Apply backjumping

( a[d] , not(p)[3], not(q)[4], not(b)[5] | C0 | q v b )

Conflict

Resolution between [5] and q v b

not(b) v p q v b

---------------------------

p v q

( a[d] , not(p)[3], not(q)[4] | C0 | p v q )

Conflict

Resolution between [4] and p v q

not(a) v not(q) p v q

----------------------------------

not(a) v p

( a[d] , not(p)[3] | C0 | not(a) v p )

Conflict

Resolution between [3] and not(a) v p

not(a) v not(p) not(a) v p

------------------------------

not(a)

Learned clause not(a) [9]

Back to search state

( not(a) | Co [9] | \* )

Propagate

( not(a), r[1] | Co [9] | \* )

Propagate

( not(a), r[1], not(b)[6] | Co [9] | \* )

Propagate

( not(a), r[1], not(b)[6], not(c)[8] | Co [9] | \* )

Conflict

( not(a), r[1], not(b)[6], not(c)[8] | Co [9] | b v c )

Since I go to conflict without any decided literal it is UNSAT