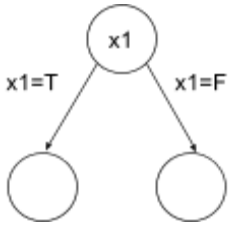
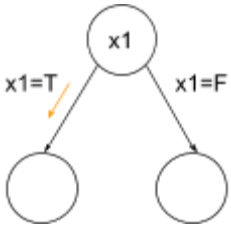
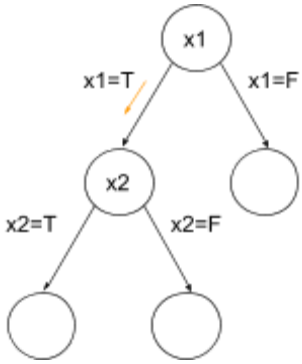
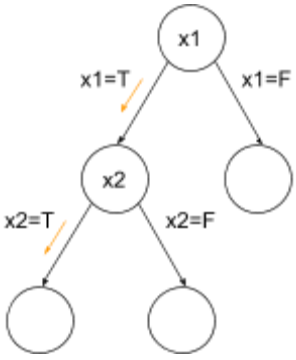
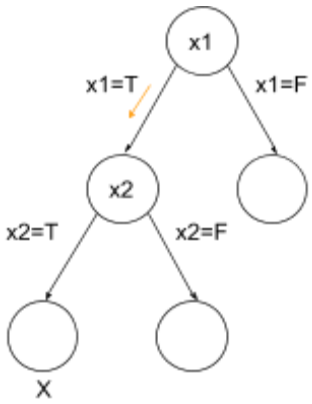
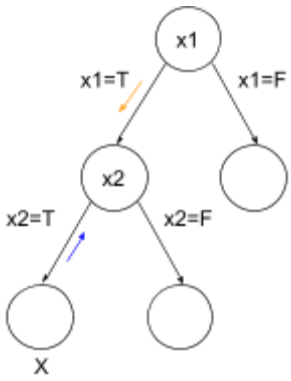
DPLL algorithm:

$(\neg x1 \vee \neg x2) \wedge (x1 \vee \neg x2) \wedge (\neg x1 \vee \neg x3)$

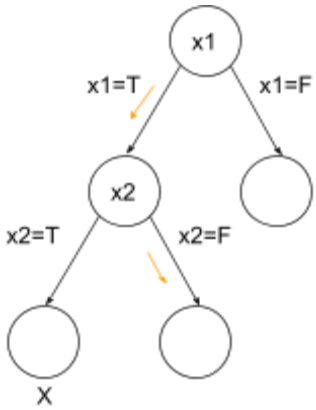
 two recursive calls with assignments {x1=true} and {x1=false}	 first recursive call is with assignment {x1=true} no clause of $(\neg x1 \vee \neg x2) \wedge (x1 \vee \neg x2) \wedge (\neg x1 \vee \neg x3)$ is falsified by {x1=true} no contradiction: choose an unassigned variable
 branching variable x2 (for example) do two recursive calls adding the two possible evaluations of x_2 to the original one partial interpretations in the recursive calls are then {x1=true, x2=true} and {x1=true, x2=false}	 first recursive call with assignment {x1=true, x2=true}: In $(\neg x1 \vee \neg x2) \wedge (x1 \vee \neg x2) \wedge (\neg x1 \vee \neg x3)$, the clause $(\neg x1 \vee \neg x2)$ is falsified

<table>
<tr>
<td> 

contradiction, close branch of the tree</td>
<td> 

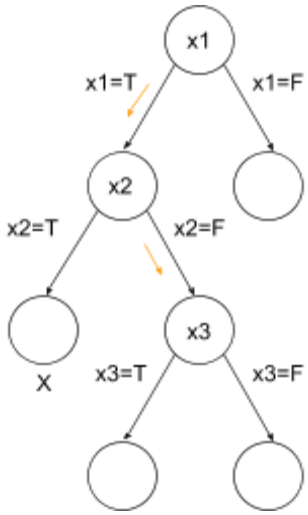
go back to node labeled x2

x2=true already tried

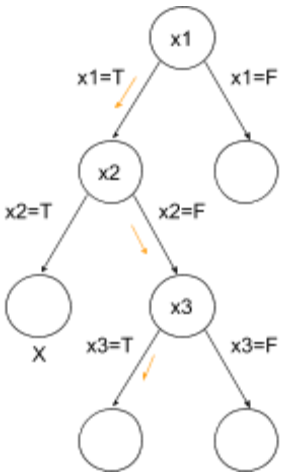
now try x2=false</td>
</tr>
<tr>
<td> 

assignment {x1=true, x2=false}

Formula (¬x1 v ¬x2) ^ (x1 v ¬x2) ^ (¬x1 v ¬x3), is not falsified

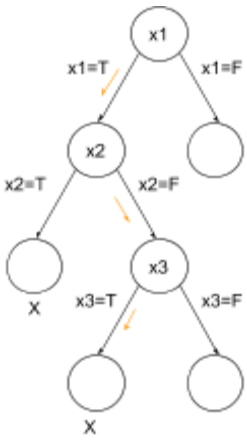
choose variable: only left unassigned is x3</td>
<td> 

two recursive calls: x3=true, x3=false</td>
</tr>
</table>



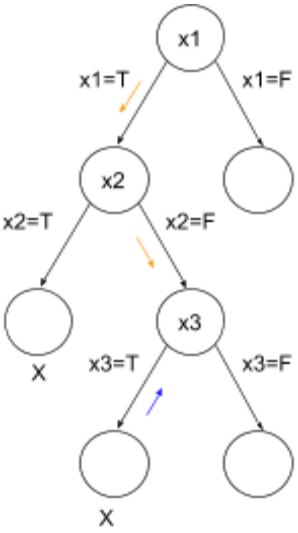
first recursive call has assignment {x1=true, x2=false, x3=true}

in formula (¬x1 v ¬x2) ^ (x1 v ¬x2) ^ (¬x1 v ¬x3), the clause (¬x1 v ¬x3) is falsified

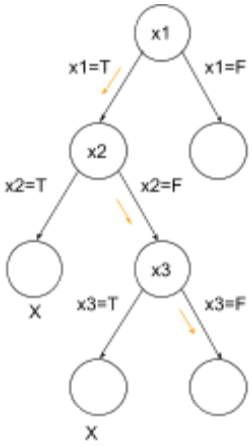


clause is falsified=formula is falsified

close branch



backtrack to node labeled x3

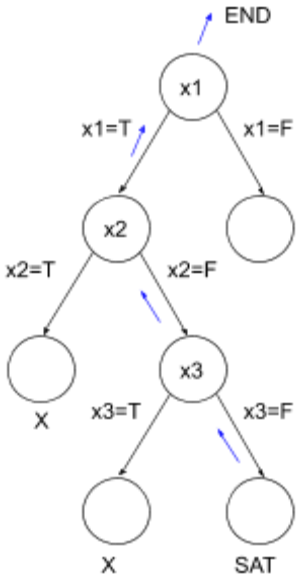


second recursive call for x3

value x3=false

assignment is {x1=true, x2=false, x3=false}

all clauses in (¬x1 v ¬x2) ^ (x1 v ¬x2)

	^ (¬x1 v ¬x3), are satisfied!
 no other recursive calls if a subcall returns true, the call returns true as well this means: in this case, we go back to the original call and return true model found, no need to go ahead formula is satisfiable	

Proof using resolution:

(1) If you go swimming you will get wet.

(2) If it is raining and you are outside then you will get wet.

(3) If it is warm and there is no rain then it is a pleasant day.

(4) You are not wet.

(5) You are outside.

(6) It is a warm day.

(1) swimming $\Rightarrow$ wet

(2) (rain $\wedge$ outside) $\Rightarrow$ wet

(3) (warm $\wedge$ $\sim$ rain) $\Rightarrow$ pleasant

(1) ($\sim$ swimming $\vee$ wet) $\wedge$

(2) ($\sim$ rain $\vee$ $\sim$ outside $\vee$ wet) $\wedge$

(3) ($\sim$ warm $\vee$ rain $\vee$ pleasant) $\wedge$

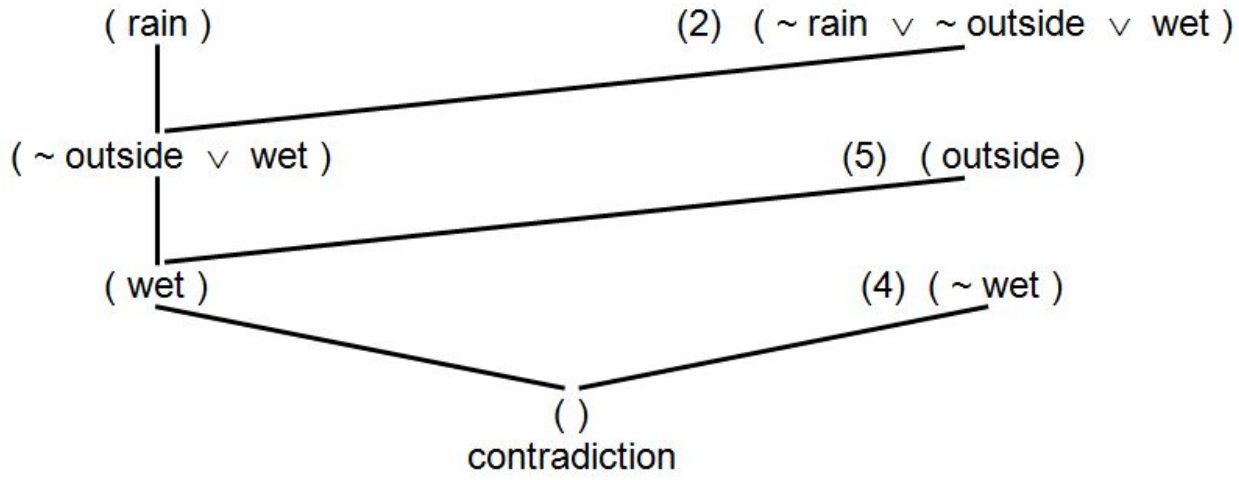
(4) ($\sim$ wet) $\wedge$

(5) (outside) $\wedge$

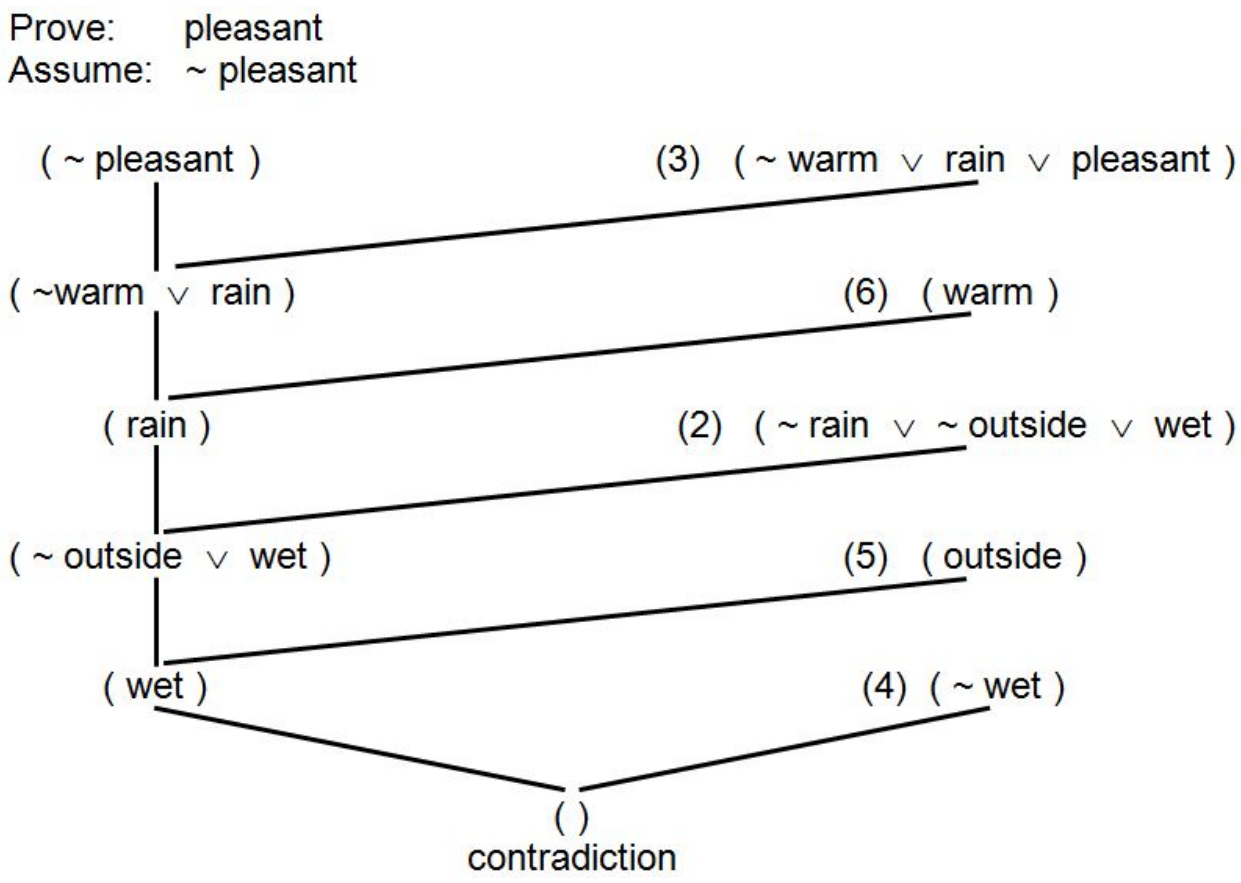
(6) (warm)

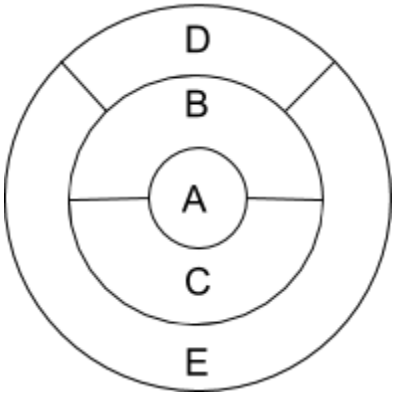
Prove: $\sim$ rain

Assume: rain



(1) (~ swimming ∨ wet) ∧
(2) (~ rain ∨ ~ outside ∨ wet) ∧
(3) (~ warm ∨ rain ∨ pleasant) ∧
(4) (~ wet) ∧
(5) (outside) ∧
(6) (warm)





$(A \neq B) \wedge (A \neq C) \wedge (B \neq C) \wedge (B \neq D) \wedge (B \neq E) \wedge (C \neq E) \wedge (D \neq E)$

For example:

$(A \neq B) \Leftrightarrow ((A=1 \wedge B=2) \vee (A=1 \wedge B=3) \vee (A=2 \wedge B=3) \vee (A=2 \wedge B=1) \vee (A=3 \wedge B=2) \vee (A=3 \wedge B=1))$

WalkSAT: (¬x1 v ¬x2) ^ (x1 v ¬x2) ^ (¬x1 v ¬x3)
Variables: x1, x2, x3
Clauses:
C1: (¬x1 v ¬x2)
C2: (x1 v ¬x2)
C3: (¬x1 v ¬x3)

Start by assigning a random value to each variable in the formula. x1=T, x2=T, x3=T Unsatisfied clauses = {C1, C3}	Pick a clause at random among unsatisfied clauses, e.g. C3: (¬x1 v ¬x3) Pick one of the variables in C3 at random, e.g. x3, and flip it (x3 = F) x1=T, x2=T, x3=F unsatisfied clauses = {C1}
Pick a clause at random among unsatisfied clauses, e.g. C1: (¬x1 v ¬x2) Pick one of the variables in C1 at random, e.g. x1, and flip it (x1 = F) x1=F, x2=T, x3=F unsatisfied clauses = {C2}	Pick a clause at random among unsatisfied clauses, e.g. C2: (x1 v ¬x2) Pick one of the variables in C2 at random, e.g. x2, and flip it (x2 = F) x1=F, x2=F, x3=F Unsatisfied clauses = {} All the clauses are satisfied.