By Reduction to SAT:

a. Systematic Search

(Backtrack Search)

b. Local Search

$\Delta \models \alpha$ iff $\Delta \wedge \neg \alpha$ is unsat

a. Complete Method: Systematic Search (dfs)

Always completes

b. Incomplete Method: Local search

Completes if sat.

If unsat, might loop infinitely.

Δ: $(A \vee B \vee \neg C) \wedge (\neg A \vee C) \wedge (A \vee C \vee \neg D)$ in CNF

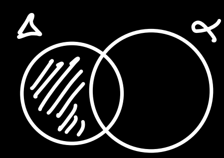
Treat SAT as a CSP

Truth Assignments

$$\omega \begin{cases} A \leftarrow T \\ B \leftarrow F \\ C \leftarrow T \\ D \leftarrow F \end{cases}$$ is returned.

→ Δ implies α

$\Delta \wedge \neg\alpha$ is unsat



→ Δ is equivalent to α

- Δ implies α
- α implies Δ

$\Delta \wedge \neg\alpha$ is unsat

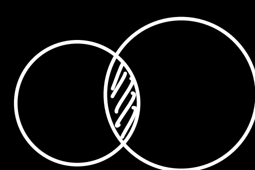
$\neg\Delta \wedge \alpha$ is unsat

→ Δ is valid

$\neg\Delta$ is unsat

→ Δ and α are mutually exclusive

$\Delta \wedge \alpha$ is unsat



COMPLETE METHODS:

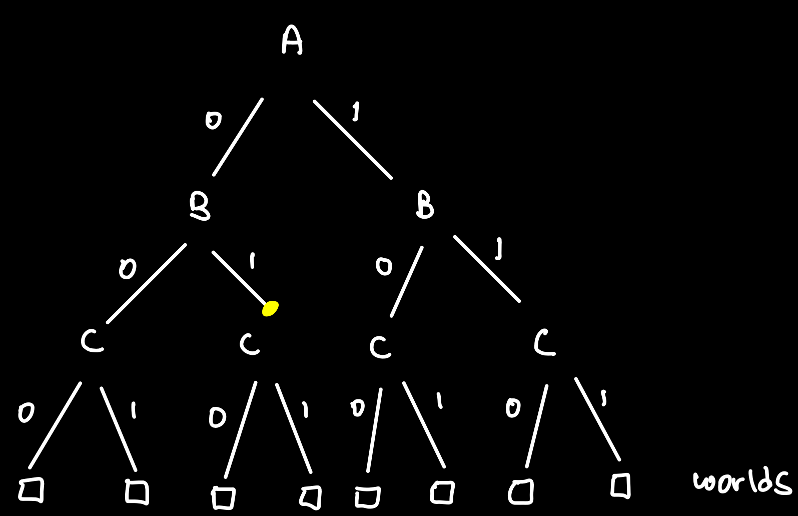
dfs, backtracking search

↳ value/variable ordering

↳ detecting failures early → unit resolution

DPLL ALGORITHM:

A, B, C



At • :

¬A   A ← F

B   B ← T

CNF :

$\alpha_1, \alpha_2 \dots \alpha_n$, ¬A, B

$\dots, \underbrace{A \vee \neg B \vee C}, \neg C, \neg A, B$

$$\underbrace{\underbrace{A \vee \neg B, \neg A}, B}$$

$$\underbrace{\neg B, B}$$

Contradiction.

Linear Time!

ORIGIN CNF Δ

$\rightarrow \Delta \wedge A$

$\rightarrow \Delta \wedge \neg A$ } $\rightarrow$ Branching on variable A.

LOCAL SEARCH (INCOMPLETE):

CNF $\alpha_1, \ldots \alpha_n$

Variables A, B, C, D

Assume

$w_1 = A \leftarrow T, B \leftarrow T, C \leftarrow F, D \leftarrow F$

$\hookrightarrow w \models \Delta$ Done. CNF is SAT. w is the solution

$\hookrightarrow w \not\models \Delta$

Try $w_2 = A \leftarrow F, B \leftarrow T, C \leftarrow F, D \leftarrow F.$

$\vdots$

* No memory. So some worlds could be missed and some repeated.

* Depends on initial assignment.

* Very low space complexity.

N - QUEENS:

Q		Q	
	Q		Q

State : Complete variable assignment

$x_1 \quad x_2 \quad \dots \quad x_n$ Binary variables

$\downarrow \quad \downarrow \quad \quad \downarrow$

$v_1 \quad v_2 \quad \quad v_n$

Neighbors : n

Neighbors:

A	B	C
0	1	1
1	1	1
0	0	1
0	1	0

3

Say we have 100 constraints

We violate 18 constraints

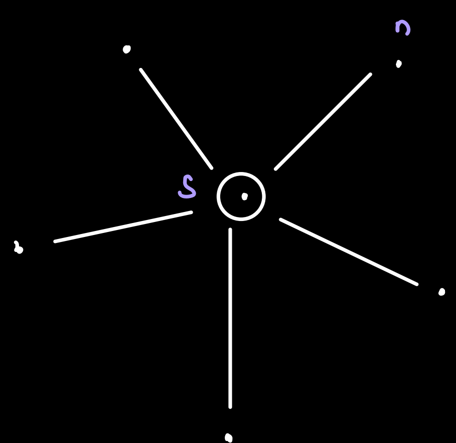
We want to find the neighbor which violates least constraints.

"Min-conflicts" or "hill climbing"

What if all neighbors are worse?

Local minima. We need to restart to a random place.

SIMULATED ANNEALING:



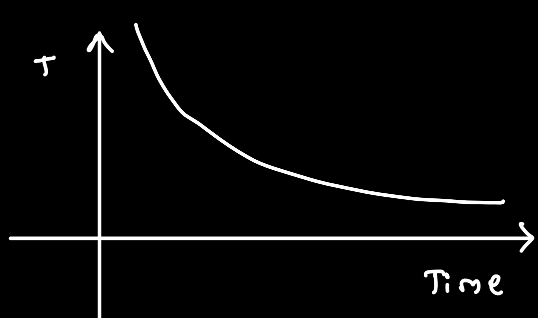
Pick random neighbor (n)

ΔE : violations (n) - violations (s)

↳ $\Delta E < 0$: go to n

↳ otherwise

go to n with $P \propto \frac{1}{e^{\Delta E / T}}$



T : Temperature

As time proceeds, chances to go to n decreases.

Randomization:

→ Avoids local minima

→ Reaches almost all nodes.

METHOD 4: TRACTABLE CIRCUITS / KNOWLEDGE COMPILATION:

Input Formula : CNF

↳ "Compile it" into a "tractable circuit".

It can solve beyond SAT, like

SAT / MODEL COUNTING:

$\Delta = (A \vee B) \wedge C$

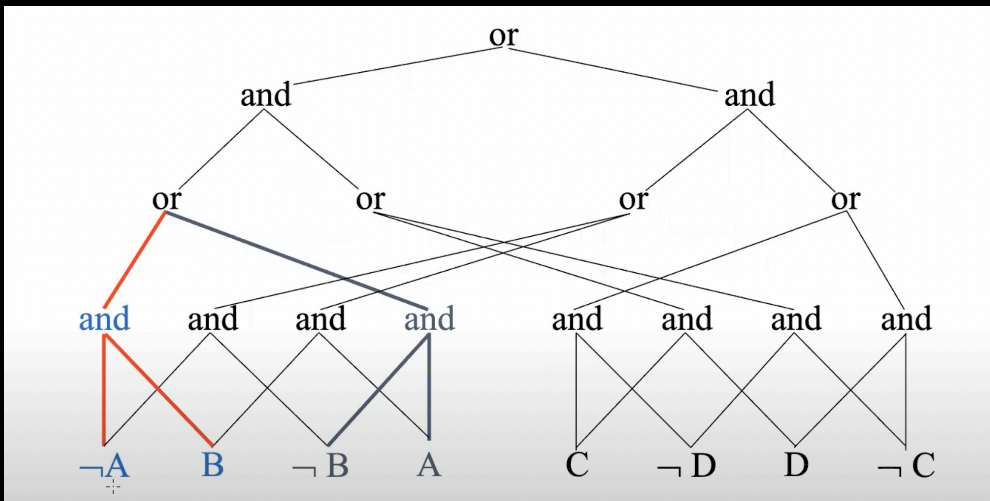
→ Total # of truth assignments 8

→ How many satisfy the CNF = 3

↓

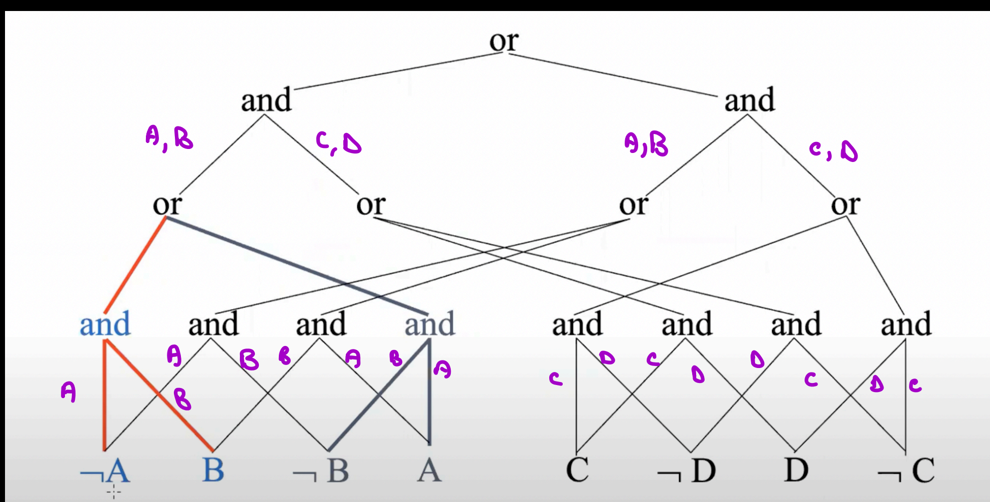
This is what # SAT returns.

NNF :



→ Decomposability :

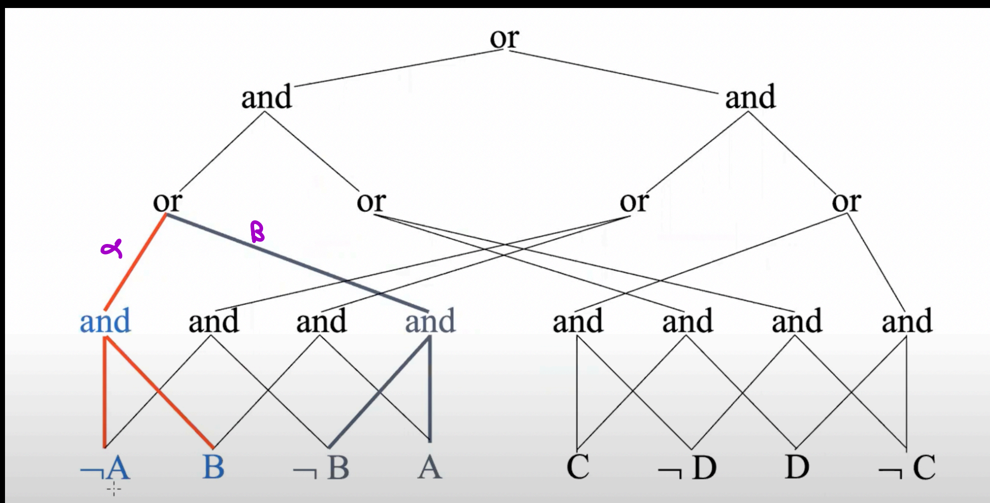
AND Gates : Children should not share variables



DNNF

→ DETERMINISM:

OR Gates. 2 Children → α, β then α, β are mutually exclusive. $\alpha \wedge \beta$ unsat.



α : $\neg A \cap B$

β = $A \wedge \neg B$

$\alpha \wedge \beta$ → unsat

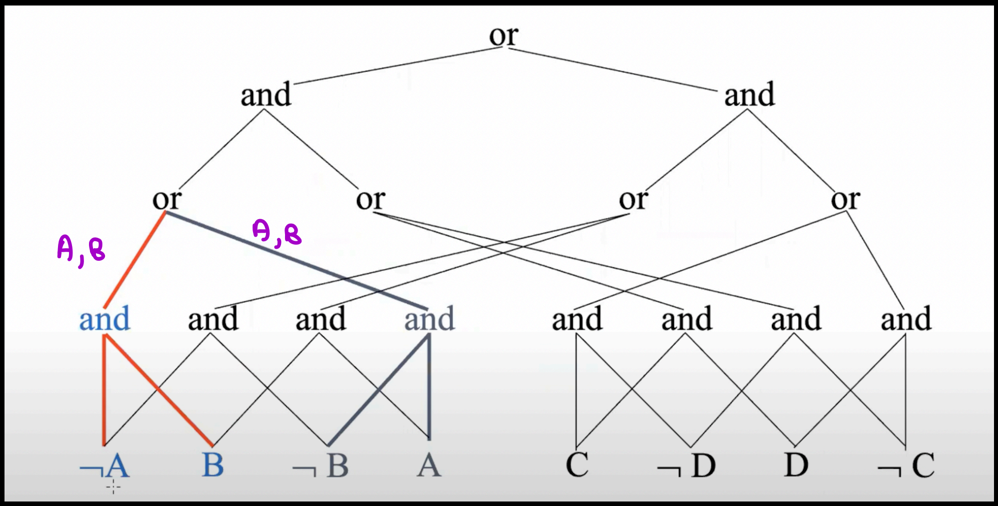
If 3 children α or β or γ

then $\alpha \wedge \beta$, $\beta \wedge \gamma$, $\alpha \wedge \gamma$ → unsat.

Decomposability + Determinism → d-DNNF

→ SMOOTHNESS:

OR Gates → Children same variables.



HOW TO SOLVE #SAT:

1. BASE CASE:

True gets 1

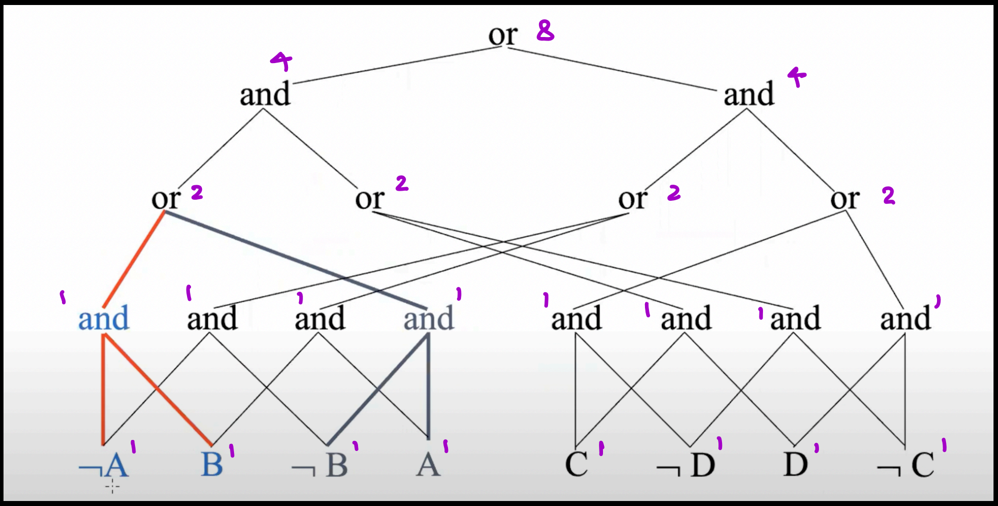
False gets 0

Every literal get 1.

(+ve / -ve)

2. AND Gates : Multiply

OR Gates : Add


or 8
and 4
and 4
or 2
or 2
or 2
or 2
1 and
1 and
1 and
and 1
1 and
1 and
1 and
and 1
¬A 1
B 1
¬ B 1
A 1
C 1
¬ D 1
D 1
¬ C 1