Satisfiability	NP-complete
Validity	coNP-complete
Model counting	#P-complete

Grammar of Boolean Forms

$$\varphi ::= p,\ q,\ r,\ \ldots \quad \text{(Atomic prop.)}$$
$$\mid\ \varphi_1 \text{ and } \varphi_2 \quad \text{(Conjunctions)}$$
$$\mid\ \varphi_1 \text{ or } \varphi_2 \quad \text{(Disjunctions)}$$
$$\mid\ \text{not } \varphi \quad \text{(Negations)}$$

Conjunctive Normal Form

Clauses

$\varphi =$ " (___ or ___ or ___) and
(___ or ___) and
___ and ...

Literals
p | not p

CNF $\varphi ::= \text{Clause}_1 \wedge \text{Clause}_2 \wedge \dots \wedge \text{Clause}_n$

$\text{Clause} ::= \text{Lit}_1 \vee \text{Lit}_2 \vee \dots \vee \text{Lit}_k$

$\text{Lit} ::= p \mid \neg p$

---

k-CNF $=$ All those formulas in CNF where each clause $\leq$ k literals

k-SAT $=$ Does a k-CNF formula admit a model?

1-SAT — trivial

2-SAT — linear time

$\underbrace{\text{3-SAT} \quad \dots}_{\text{NP-complete}}$

Disjunctive Normal Form

φ ::= (____ and ____ and ____) or

(____ and ____) or

Literals

(____ and ____) or . . .

Disjuncts | Terms

Ex: φ = (a and b) or

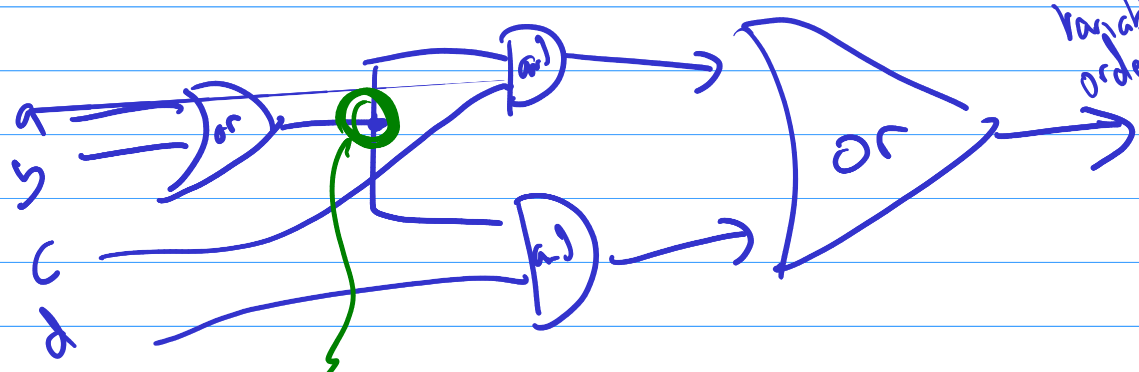
(not a and b) or

(a and not b)

DNF-SAT can be solved in linear time

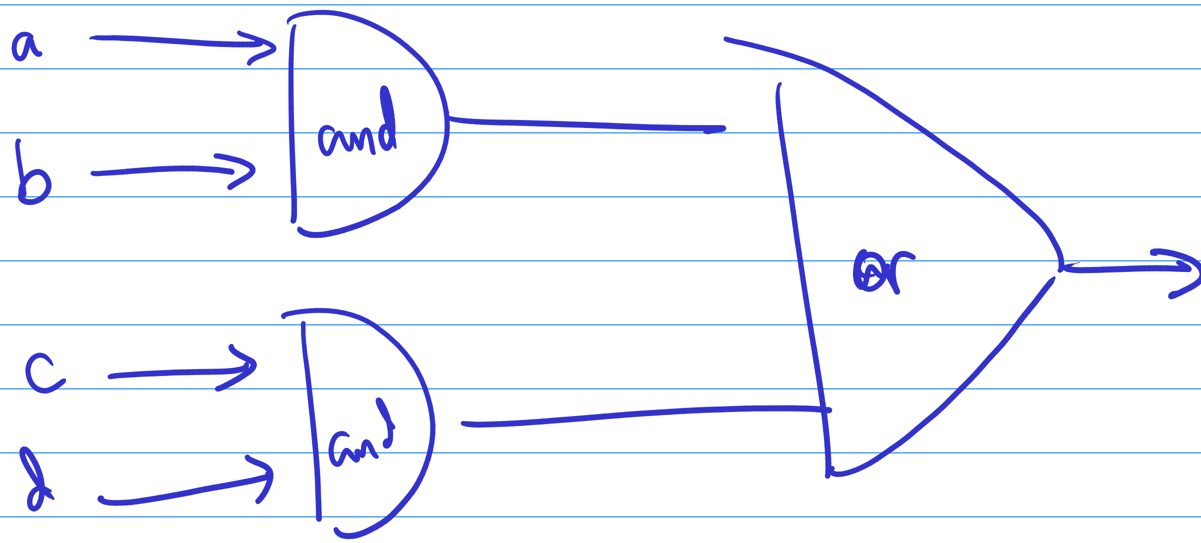
- Representation matters
- Not easy to switch representations

	General formulas	CNF	DNF	Circuits	Reduced order BDD
sat	NP-C	NP-C	Linear	NP-C	Trivial
valid	coNP-C	Linear	CONP-C	CONP-C	Trivial
Model counting	#P-C	#P-C	#P-C	#P-C	Simple



variable ordering | NP-h

(a and b) or (c and d)



(a and b) or ((a and b) or c)

let x = 5 + 3
let y = x + 8
let z = x + y

Traditionally no restriction on reuse

Rust / Linear types

(a and b) or

(a and $\bar{b}$) or $\quad\quad \bar{a}, \bar{b} \to$ False

($\bar{a}$ and b) or

($\bar{a}$ and $\bar{b}$)

<u>Distributivity</u> : a and (b or c)

$\equiv$ (a and b) or (a and c)

a or $(b \wedge c)$

$\equiv (a \vee b) \wedge (a \vee c)$

$$\overline{a \wedge b} \equiv \bar{a} \vee \bar{b}$$

$$\overline{a \vee b} \equiv \bar{a} \wedge \bar{b}$$

Equivalence vs. equisatisfiability

Problem 1: Given a circuit C

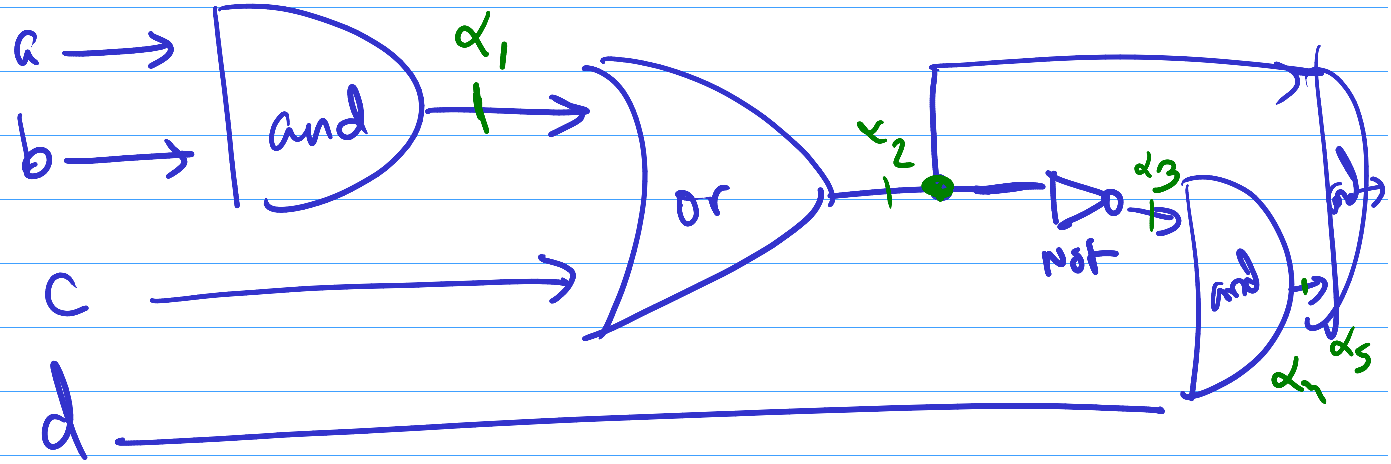
find a CNF φ

s.t. C is sat $\iff$ φ is sat

Problem 2: Given a CNF φ

find if φ is SAT.

Tseitin's Transform



$a \wedge b \Rightarrow \alpha_1$ $\quad \overline{a \wedge b} \vee \alpha_1$ $\quad \bar{a} \vee \bar{b} \vee \alpha_1$

$\alpha_1 \Rightarrow a \wedge b$ $\quad \bar{\alpha}_1 \vee (a \wedge b)$ $\quad \bar{\alpha}_1 \vee a$

$\bar{\alpha}_1 \vee b$

$\alpha_1 \vee c \Rightarrow \alpha_2$ $\quad \overline{\alpha_1 \vee c} \vee \alpha_2$ $\quad (\bar{\alpha}_1 \wedge \bar{c}) \vee \alpha_2$

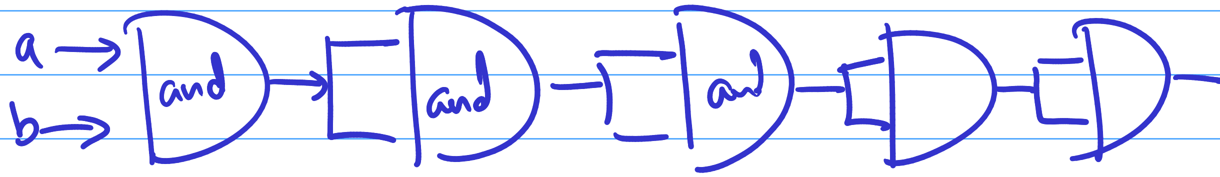
$\alpha_2 \Rightarrow \alpha_1 \vee c$ $\quad (\bar{\alpha}_2 \vee \alpha_1 \vee c)$ $\quad \bar{\alpha}_1 \vee \alpha_2$

$\bar{c} \vee \alpha_2$

$\alpha_2 \Rightarrow \bar{\alpha}_3$ $\quad \bar{\alpha}_2 \vee \alpha_3$

$\bar{\alpha}_3 \Rightarrow \alpha_2$ $\quad \alpha_3 \vee \alpha_2$

α_5



a
b
and
and
and

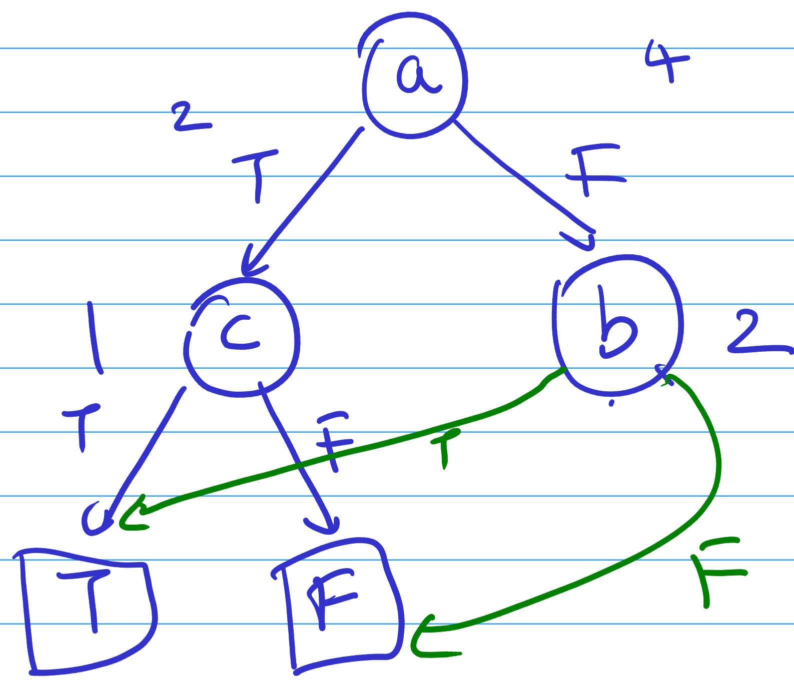
~~Reduced~~ Order Binary Decision Diagrams

Variables are ordered

Decision Trees?

$$\varphi = (a \text{ or } b) \wedge (\bar{a} \vee c)$$

$a < b < c$



Reduced: As much sharing as possible

Sharing subgraphs is mandatory

$(a \wedge b) \vee (a \wedge c)$

a and (b or c)

(b or c) and a

$D_1 \qquad D_2 \implies D_1 \wedge D_2$

poly time

$D_1 \qquad D_2 \implies D_1 \vee D_2$

$D_1 \implies \text{not } D_1$

$(x^2)^2 \ldots$

Algorithms for CNF-sat

① DPLL

$$(a \lor b) \land (\bar{a} \lor c) \land (\bar{b} \lor c)$$

② CDCL

↓

Conflict
Driven
Clause
Learning

$a \xrightarrow{F} b \xrightarrow{F}$