Lecture 4

Agenda: Binary Decision Diagrams (RO)

DPLL (1962)

Unit propagation (why CNF?)

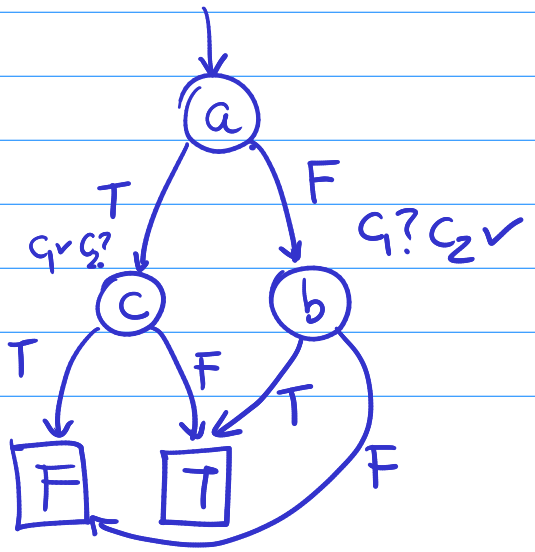
Horn-SAT
2-SAT
] sketch polytime algorithms

BDDs (Randall Bryant 1986

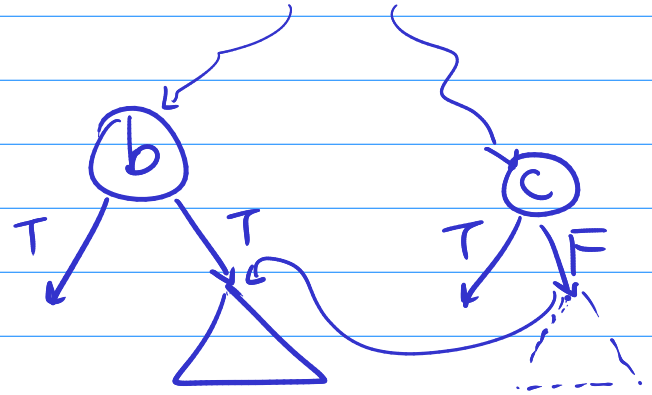
Graph Based Algorithms for Boolean Function Manipulation)

$\varphi = \underbrace{(a \vee b)}_{c_1} \wedge \underbrace{(\bar{a} \vee c)}_{c_2}$

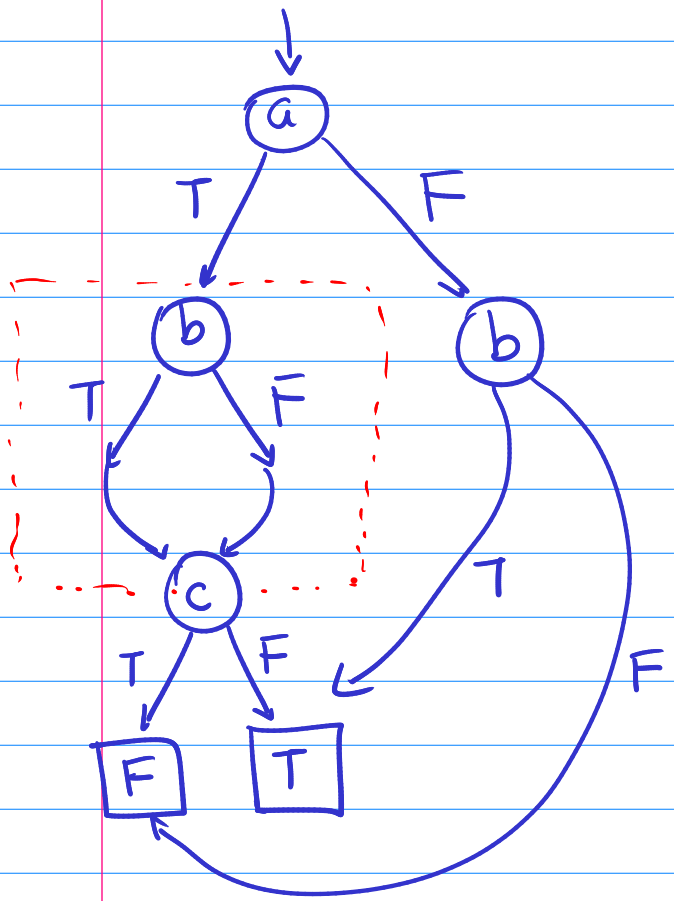
Decision DAG =

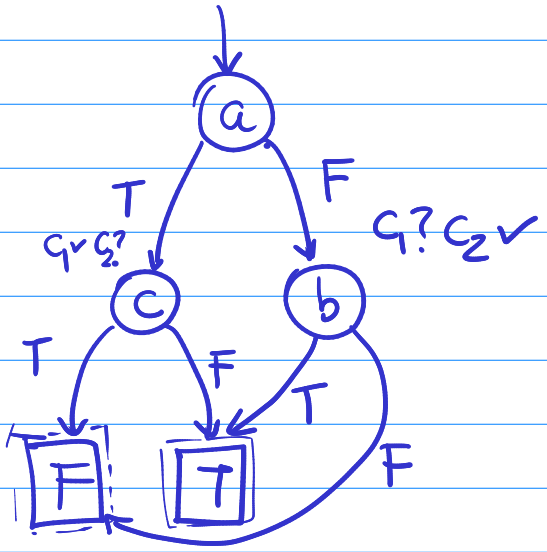


Reduced = "As much sharing as possible"

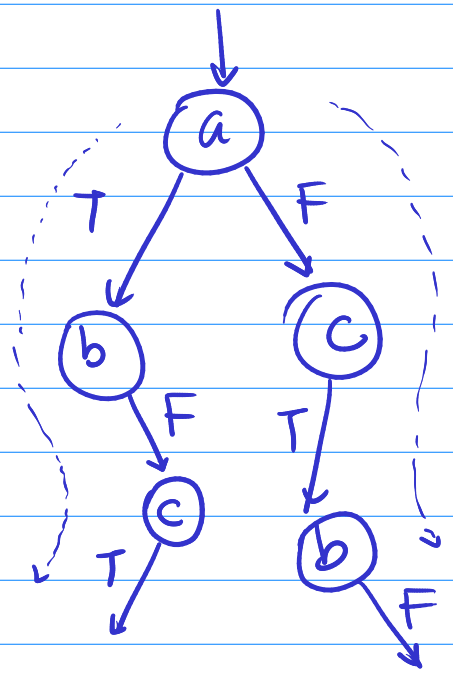


"No node with identical children"





Ordered = Along any path from root to leaf, nodes appear in the same order.



abc

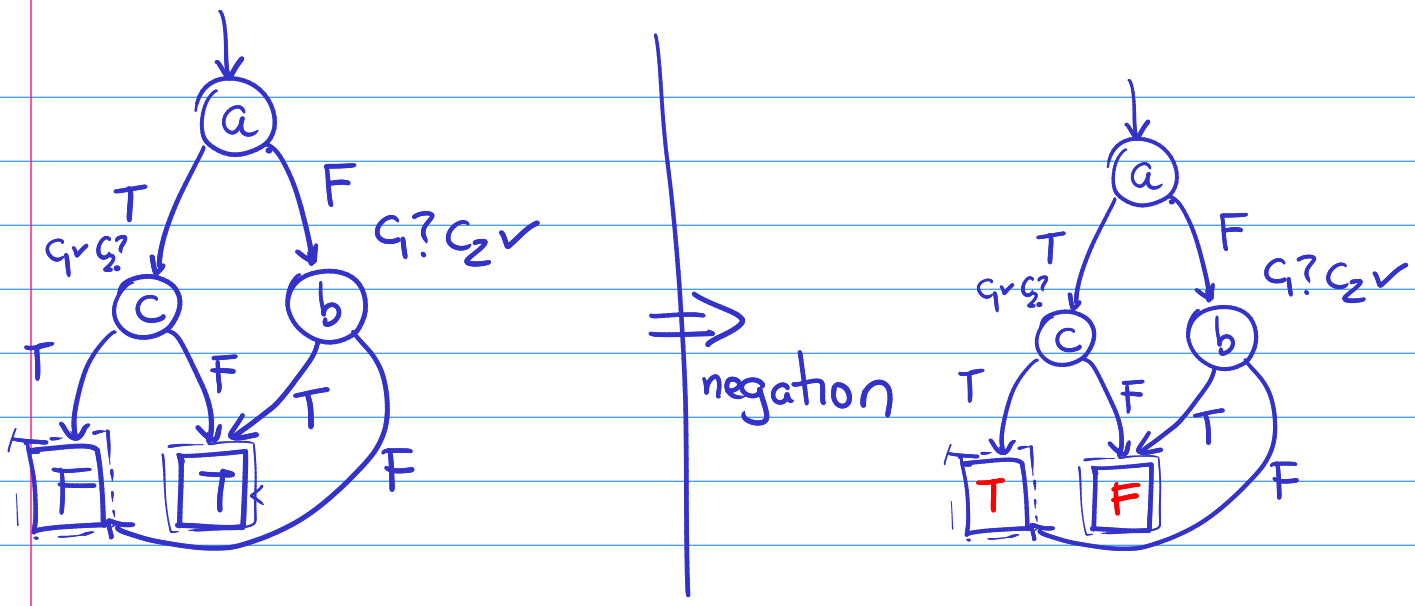
acb ← Not an ordered BDD

Claim: For a given variable ordering:

- ROBDDs are canonical
- For given ROBDDs φ_1 φ_2

BDDs for $(\varphi_1 \wedge \varphi_2)$ $(\varphi_1 \vee \varphi_2)$ $(\neg \varphi_1)$

can all be constructed in polynomial time



Question: How to check if a BDD is satisfiable?

Just check if [T] occurs!

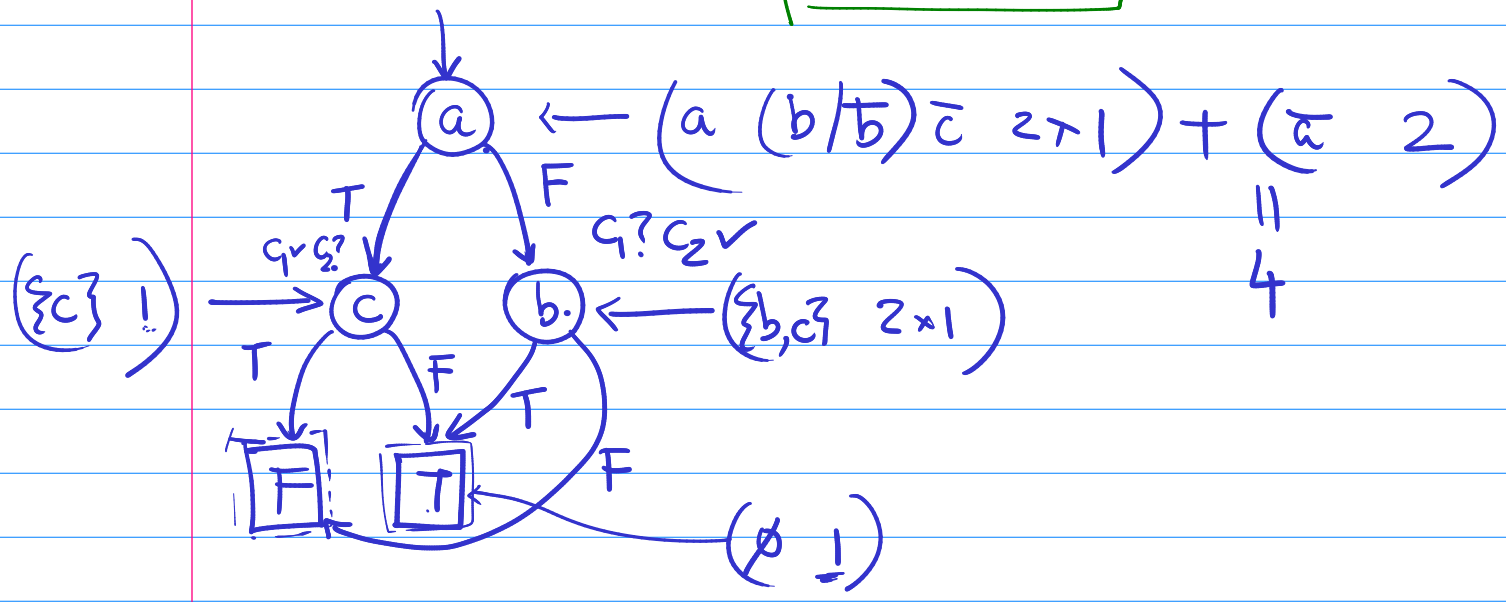
Linear time?

Question: How to check if a BDD is valid?

Just check if [F] occurs!

Question: How do you count the # of models?

$a < \underline{b} < \underline{c}$

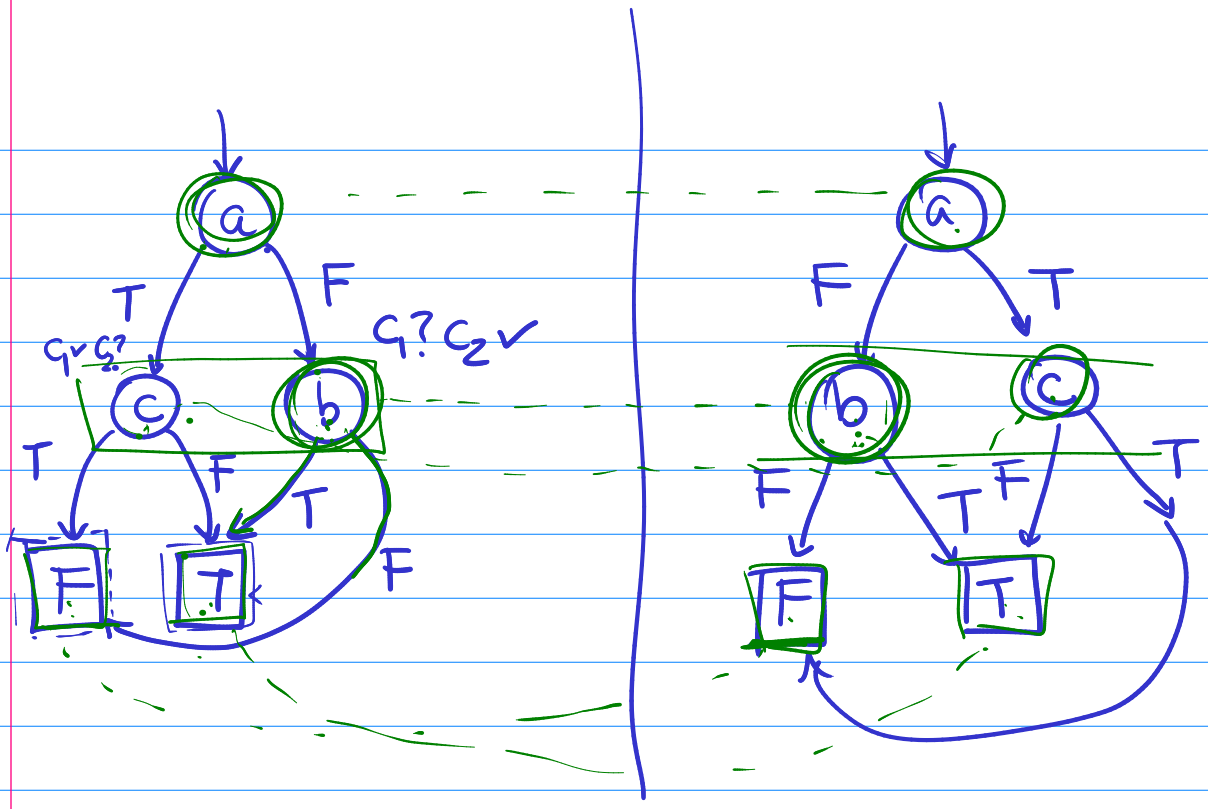


Question: How to determine if

$\varphi_1 = a \wedge (b \vee c)$ and $\varphi_2 = (a \wedge b) \vee (a \wedge c)$

are equivalent? (Same truth tables?)

- Just check if $(\varphi_1 \wedge \overline{\varphi_2}) \vee (\overline{\varphi_1} \wedge \varphi_2)$ is satisfiable.
- For BDDs, just check if φ_1 & φ_2 are the same graph!


a
T
F
c
b
T
F
T
F
F
T
c1 c2?
c1? c2 ✓
a
F
T
b
c
F
T
F
T
F
T

- DPLL (CNF formula φ)

φ = (— or — or — or x) and

(— or —) and

— and

(— or — or $\neg x$)

- If φ is empty, then return sat
- If an empty clause exists, return unsat
- Pick a variable x
- Check if DPLL($\varphi[x := \text{true}]$) is sat

  If yes, return sat.
- Check if DPLL($\varphi[x := \text{false}]$) is sat

  If yes, return sat
- Otherwise, return unsat.

$\varphi = (y \vee z \vee x)$ and $(\bar{z} \vee \bar{x})$ and $(y \vee \bar{z})$

$\varphi[x := true] =$ $\bar{z}$ and $(y \vee \bar{z})$

x = true

$\varphi[x := false] = (y \vee z)$ and $(y \vee \bar{z})$

Example 1: $\varphi = (x \vee y)$ and $(x \vee \bar{z})$

DPLL(φ)

DPLL($\varphi[x := true]$)

$\varphi[x := true]$ is empty

So return sat

Example 2 $\varphi = (\overline{x} \vee y) \wedge (\overline{x} \vee y \vee z) \wedge (\overline{x} \vee y \vee w)$

f f — Dead!

DPLL(φ)

→ $y \ldots$

DPLL($\varphi[y := \text{false}]$)

→ $\overline{y}\,\overline{x} \ldots$

DPLL($\varphi[y := \text{false}, x := \text{true}]$) $\leftrightsquigarrow \varphi'$

$\varphi' = \underbrace{\text{false}}_{C_1}$ and z and w

C_1 is an empty clause.

Return unsat

- Unit propagation

- Consider the formula

$\varphi = x$ and $(\bar{x} \vee y \vee z)$ and $(x \vee \bar{z})$

- A unit clause is one with a single literal

$\varphi = \bar{v}$ and ___ and ___

- Unit propagation rule

"If a formula has a unit clause, then immediately set the corresponding variable."

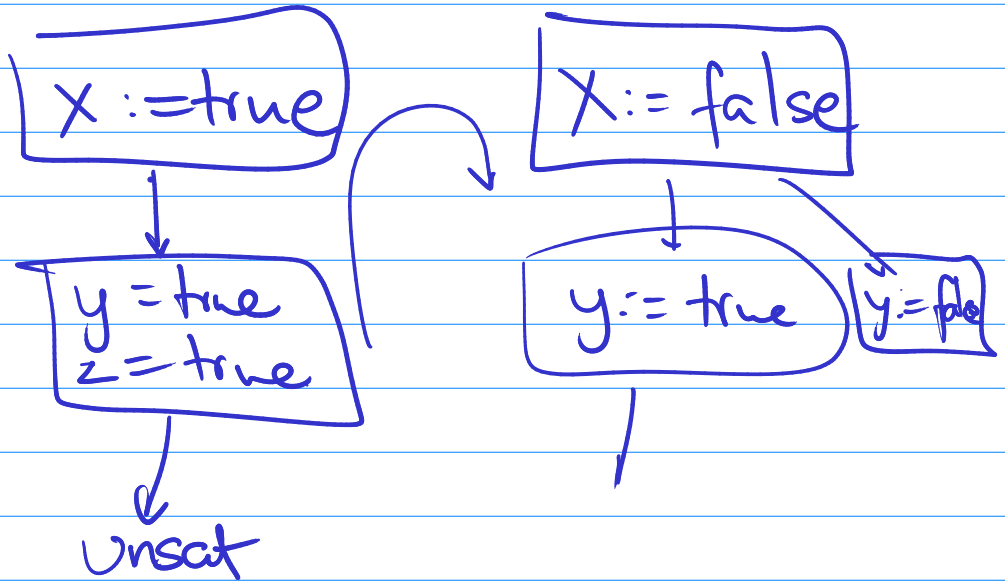
$UP(\varphi) =$ (yvz)

$DPLL_{up}(\varphi)$

- Let $\varphi' = up(\varphi)$

- If φ' is empty, then return sat
- If an empty clause exists in φ', return unsat
- Pick a variable x
- Check if $DPLL(\varphi'[x := true])$ is sat
  If yes, return sat.
- Check if $DPLL(\varphi[x := false])$ is sat
  If yes, return sat
- Otherwise, return unsat.

Example $\varphi = (\overbrace{\bar{x} \vee y}^{x \Rightarrow y}) \text{ and } (\overbrace{\bar{y} \vee z}^{y \Rightarrow z}) \text{ and } (\bar{x} \vee \bar{z})$



The formula can be satisfied with

$x := \text{false}, y := \text{false}, z := \text{true}$

Timeline of SAT solving

Quine — DP/DPLL — Bryant BDD —

1952 — 1960/62 — 1986 —

~10 vars — ~10 vars — ~100 vars

→ Trend away from benchmarks to industrial problem istances: planning/EDA/verification

GRASP — Chaff

Marques Silva
Karem Sakallah

1996

CDCL

~1K vars

Moskewicz et al.
2001

Watched lists
VSIDS
Restarts

~10K vars

- Horn SAT

Food for thought

$a \Rightarrow b \qquad \bar{a} \vee b$

A Horn clause is one with exactly 1 positive literal

$$(\bar{x} \vee \bar{y} \vee z) \Leftrightarrow (x \wedge y \Rightarrow z)$$

- If every clause is a Horn clause, how to determine satisfiability?

Food for thought 2

- 2 SAT

If every clause has at most 2 literals, then how do we check satisfiability?