COL:750

Foundations of Automatic Verification

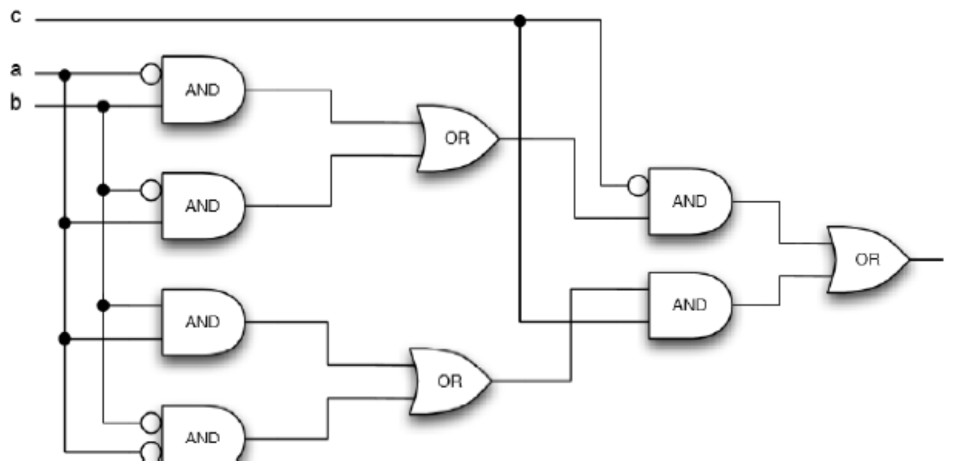
Instructor: Priyanka Golia

Course Webpage

https://priyanka-golia.github.io/teaching/COL-750/index.html

Formal Verification

```
PC1 (char[] SP, char[] UI) {
        for (int i=0; i<UI.length(); i++) {
                if(SP[i] != UI[i]) return No;
        }
        return Yes;
}
```



$\models$

System

Satisfies

Properties

$$S(I,O) \models P(I,O)$$

Is the always the case that S satisfies Property P?

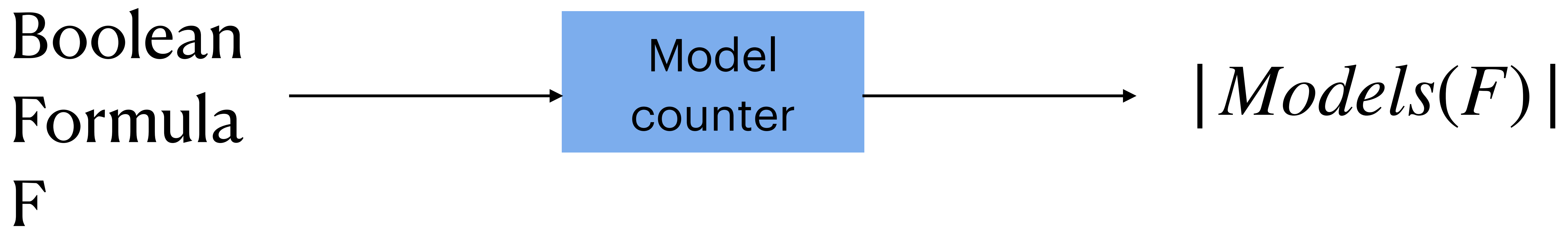
How often S satisfies P?

Why S doesn’t satisfy P?

How often System satisfies Property?

Model Counting!

Finding out how many solutions are there for a given set of constraints.



How often System satisfies Property?

Model Counting!

Finding out how many solutions are there for a given set of constraints.

$$
\begin{aligned}
&ModelCounter(F, count)\{\\
&\qquad Result, \sigma = CheckSAT(F)\\
&\qquad if\ (Result == SAT)\{\\
&\qquad\qquad count++\ \}\\
&\qquad else\ \ Return\ count\\
&\qquad ModelCounter(F \wedge \neg\sigma, count)\}
\end{aligned}
$$

Assuming access to a NP oracle !

How often System satisfies Property?

Model Counting!

Finding out how many solutions are there for a given set of constraints.

$ModelCounter(F)\{$

$If\ F\ is\ 0\ then\ Return\ 0$

$If\ F\ is\ 1\ then\ Return\ 1$

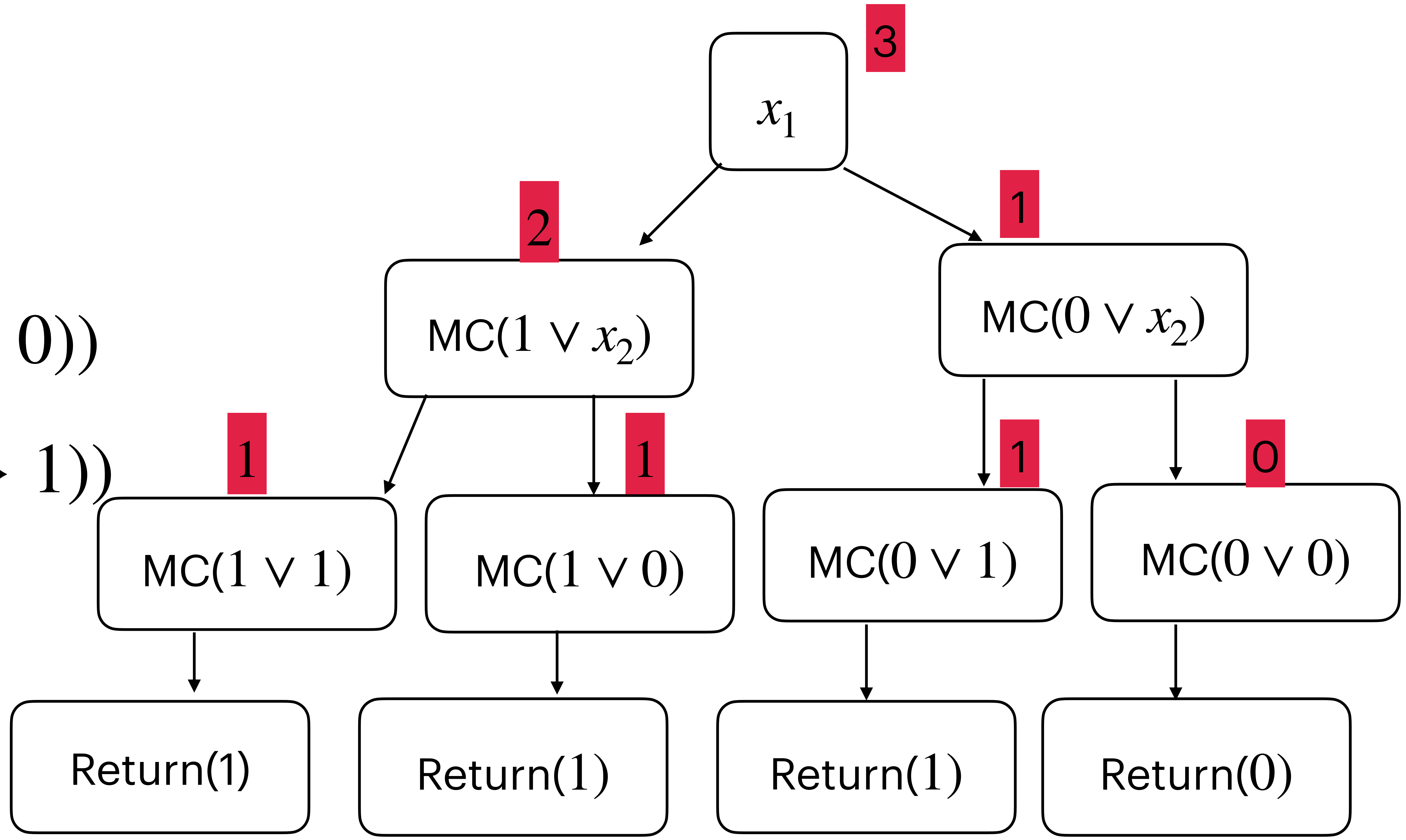
$pick\ \ x \leftarrow VARs(F)$

$C_o = ModelCounter(F(x \mapsto 0))$

$C_1 = ModelCounter(F(x \mapsto 1))$

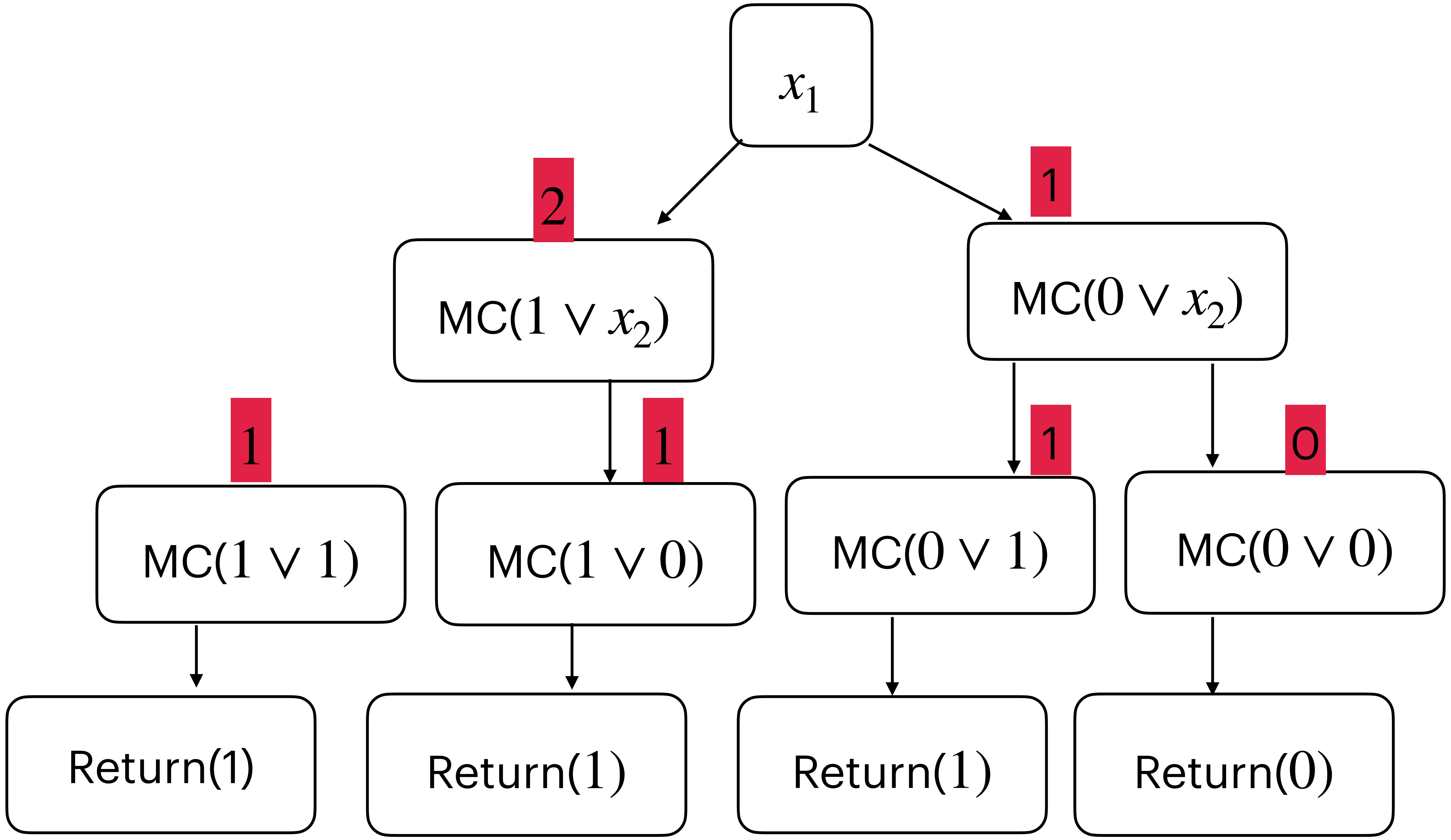
$return\ \ C_o + C_1\ \}$

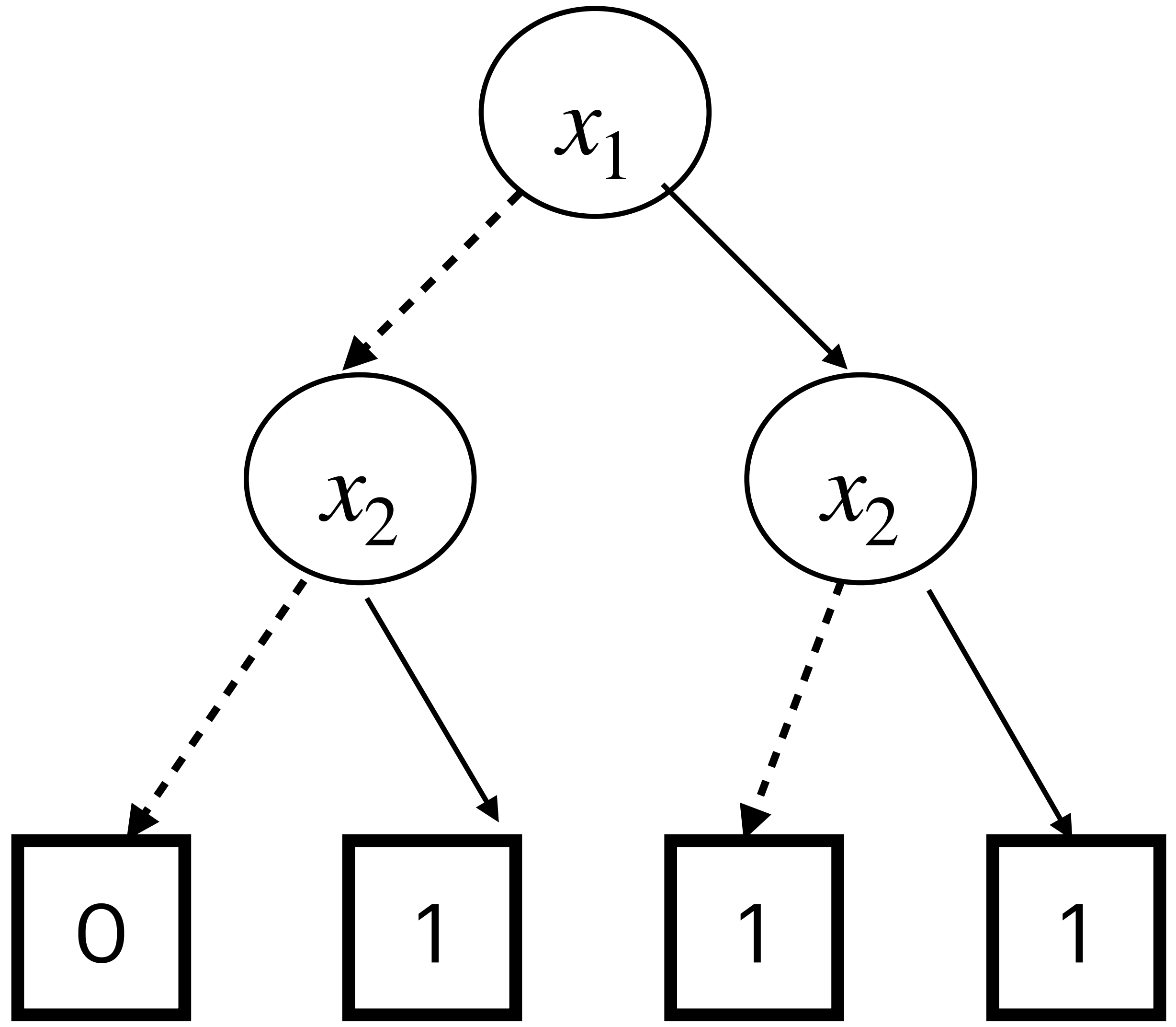
$F = x_1 \vee x_2$



How often System satisfies Property?

$$F = x_1 \lor x_2$$



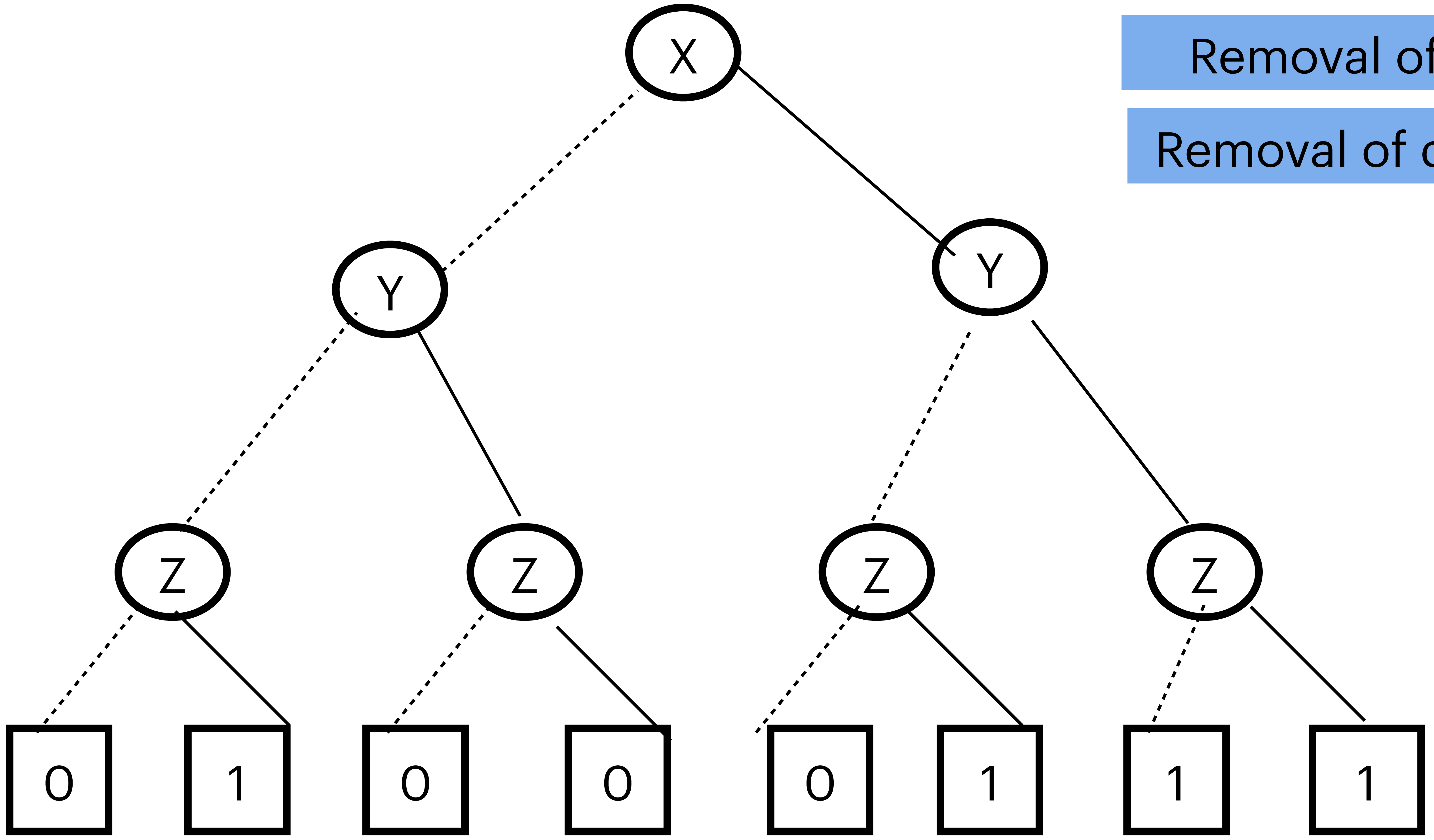


In OBDD, Model count is Sum of leaf nodes.

Model Counting

ROBDD — Reduced Ordered Binary Decision Diagrams

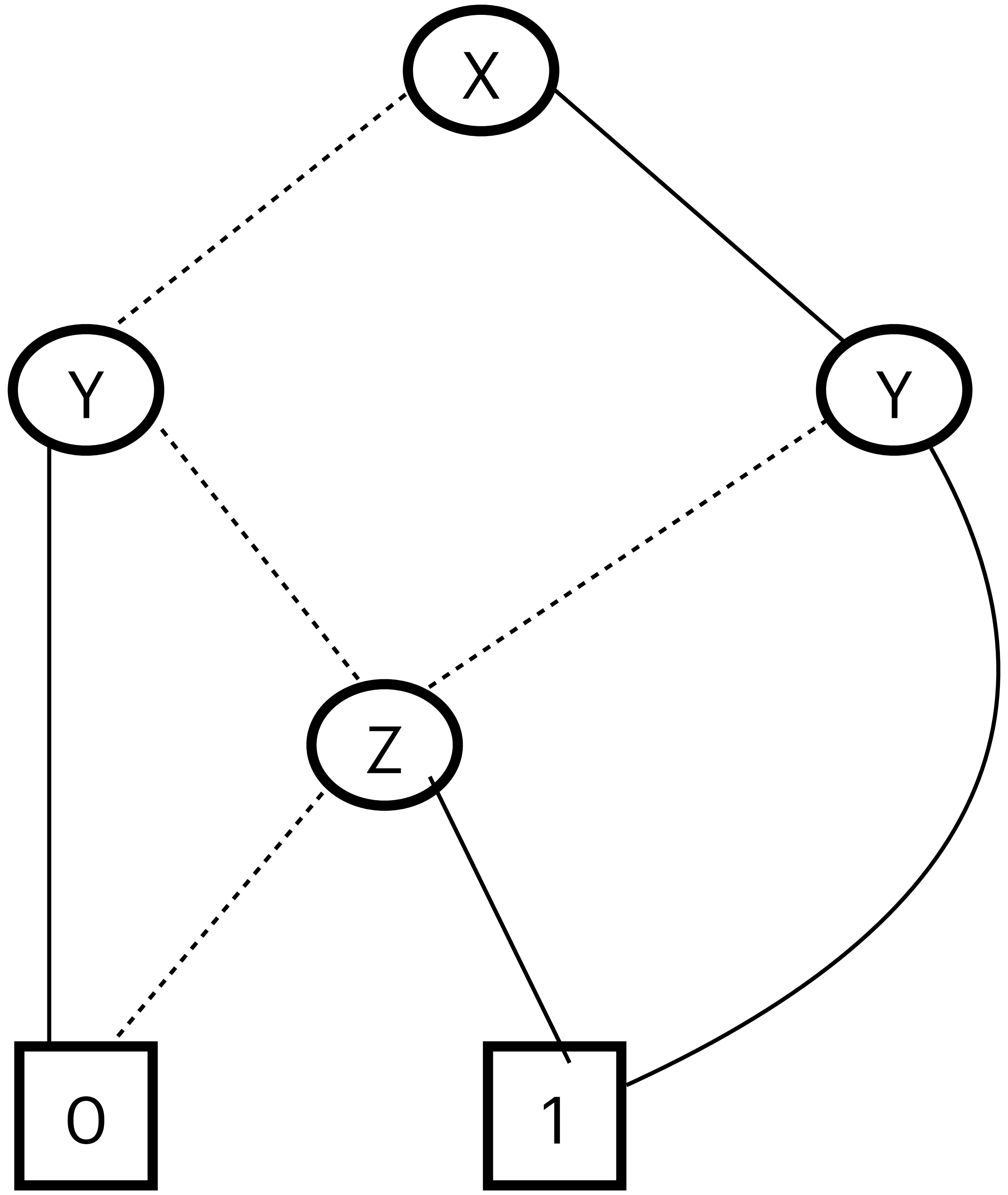
$$F = (x \wedge y) \vee (\neg y \wedge z)$$



Removal of duplicate leaves

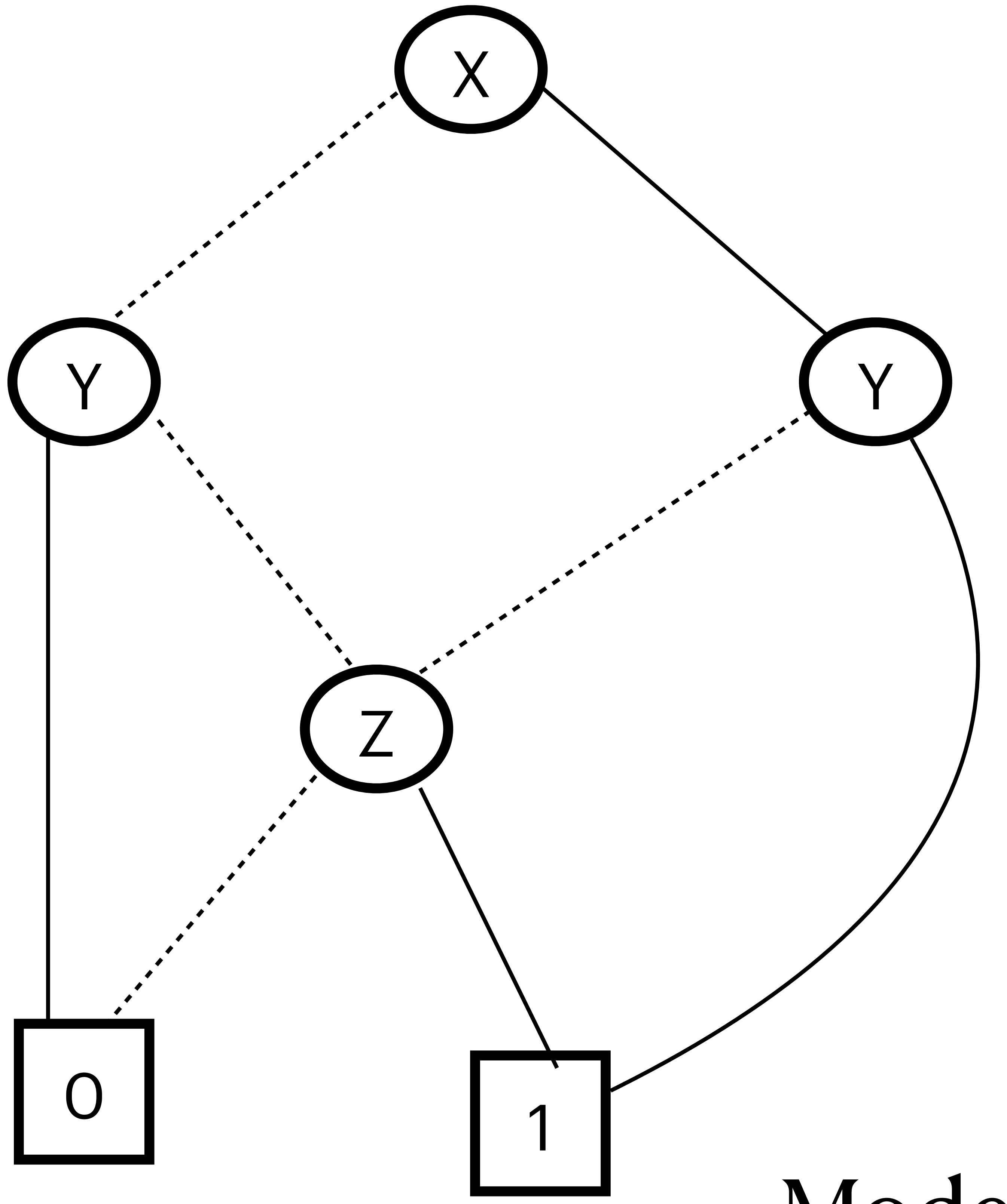
Removal of duplicate tests

Removal of duplicate sub-tree

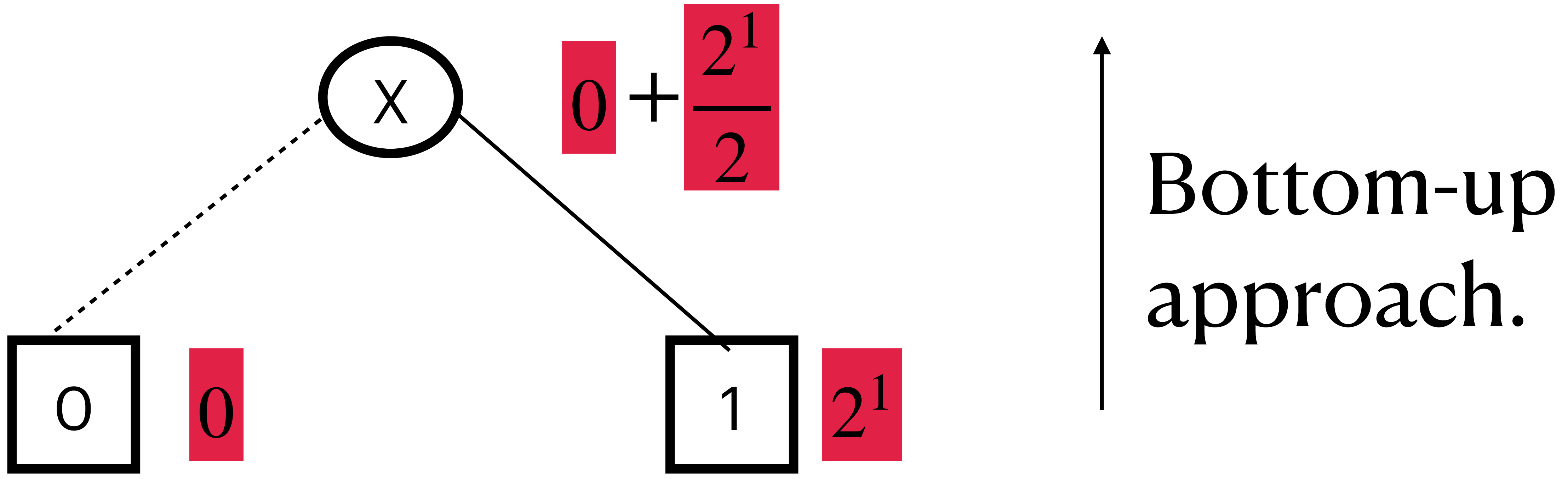


Model Counting

$$F = (x \land y) \lor (\neg y \land z)$$



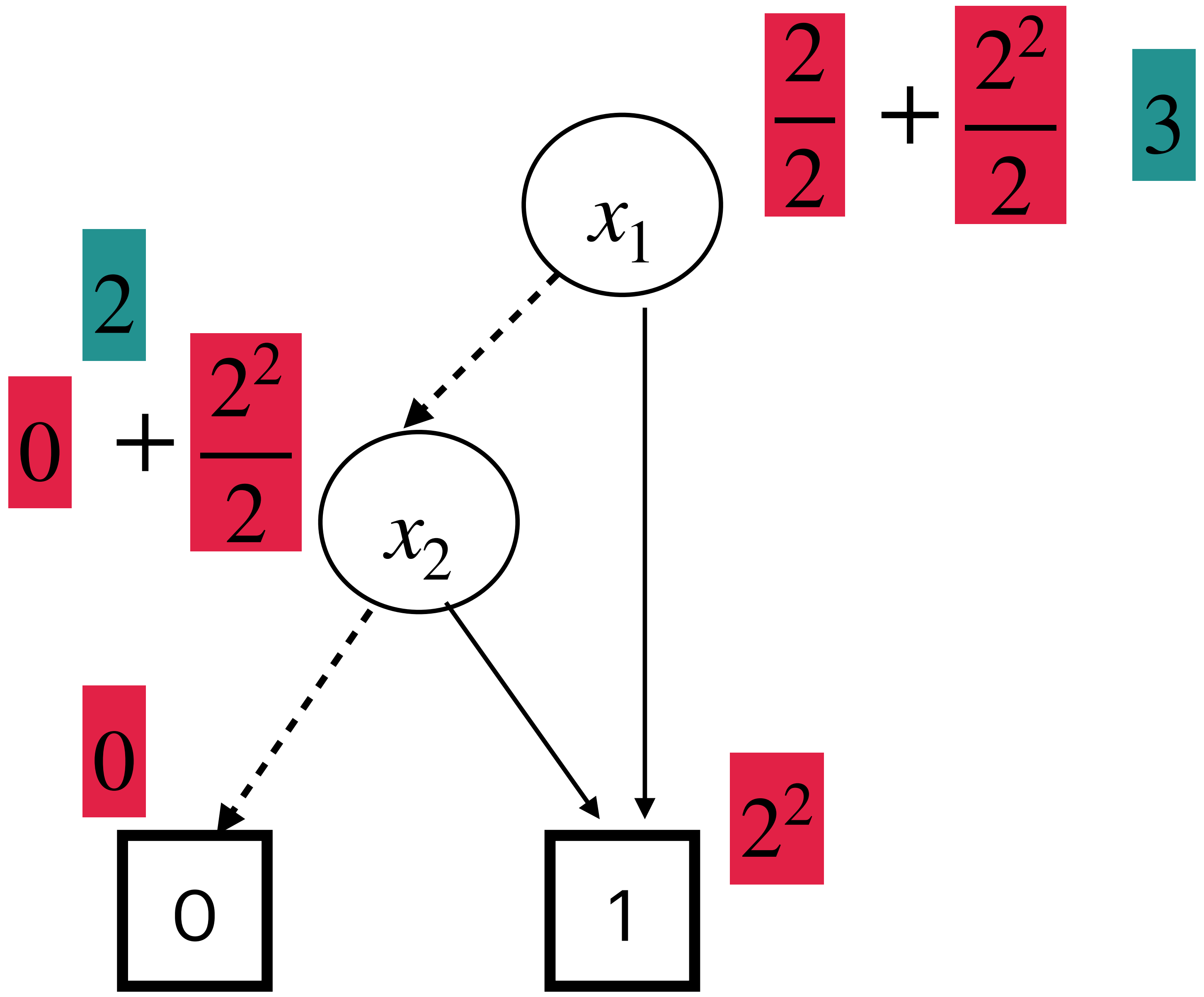
Key Observation: We are fixing a variable as we move from the child to the parent node.



Model Counting in ROBDD?

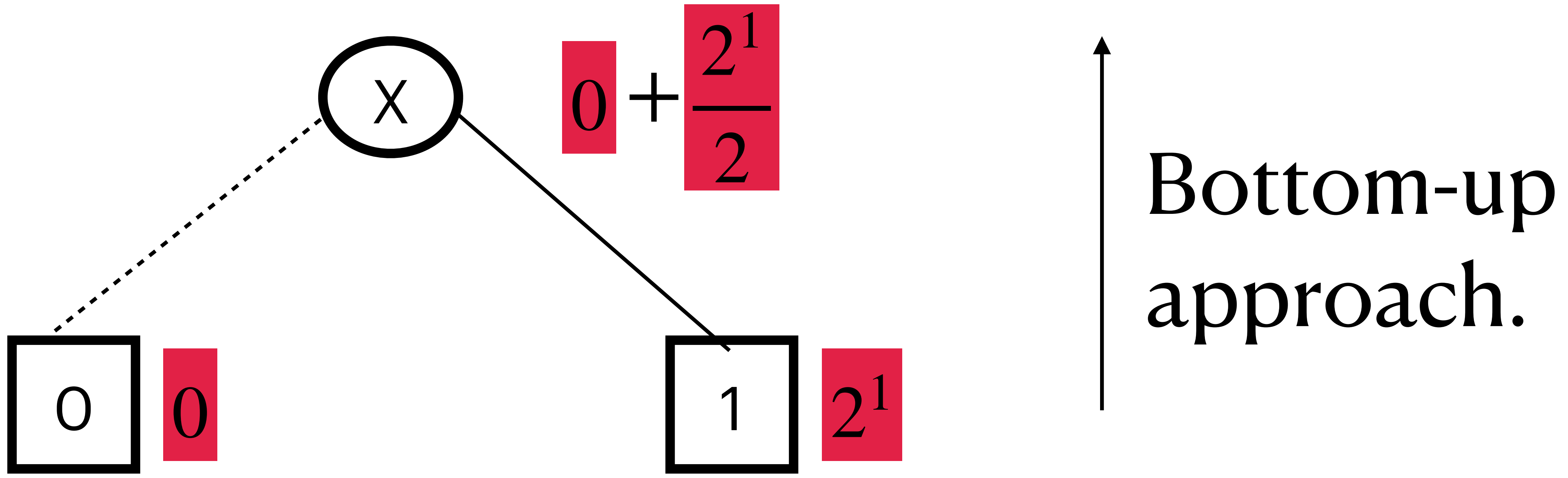
Model Counting

$F = x_1 \lor x_2$



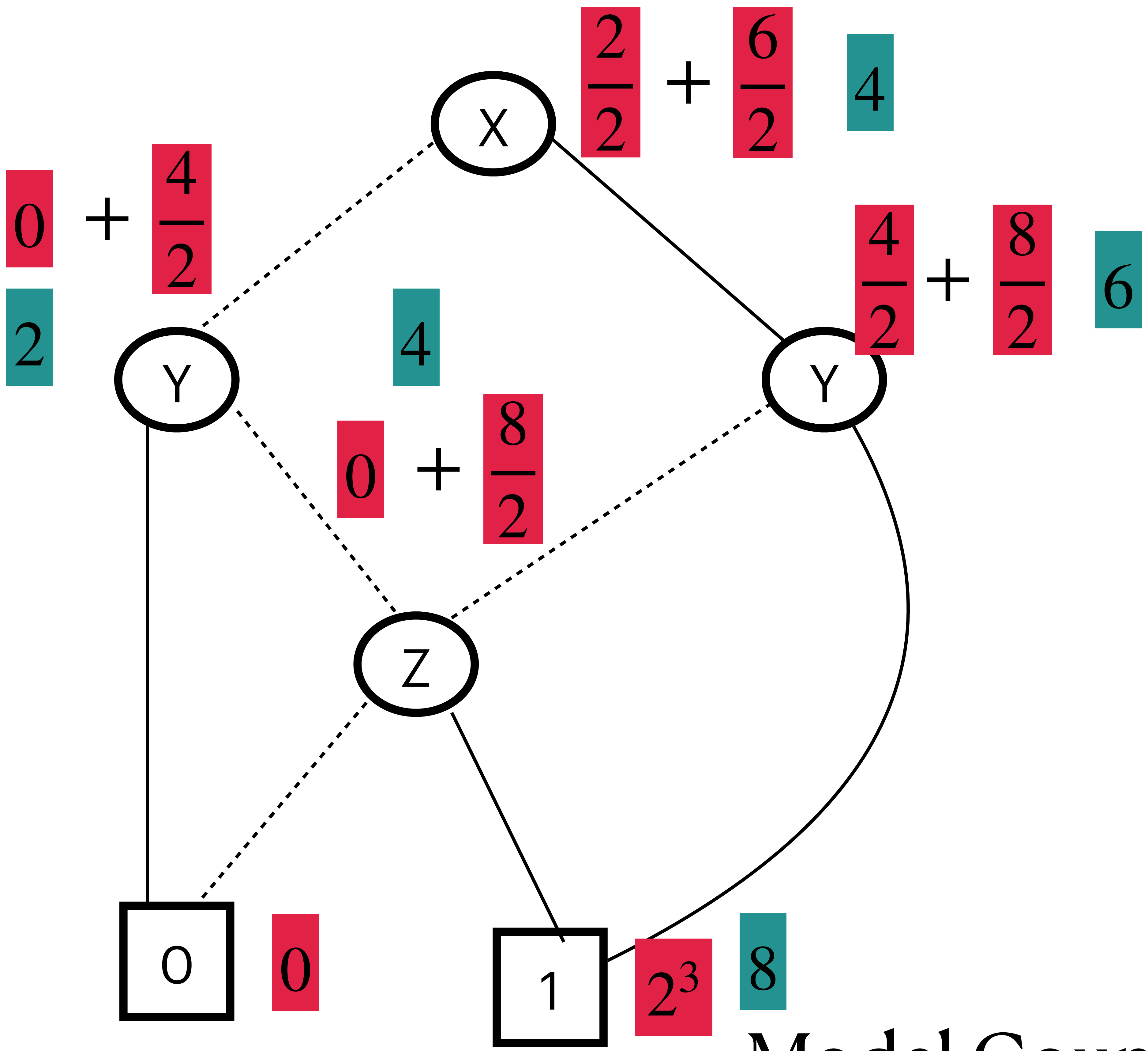
Model Counting in ROBDD?

Key Observation: We are fixing a variable as we move from the child to the parent node.



Model Counting

$$F = (x \wedge y) \vee (\neg y \wedge z)$$



$$|Models(F)| = 4$$

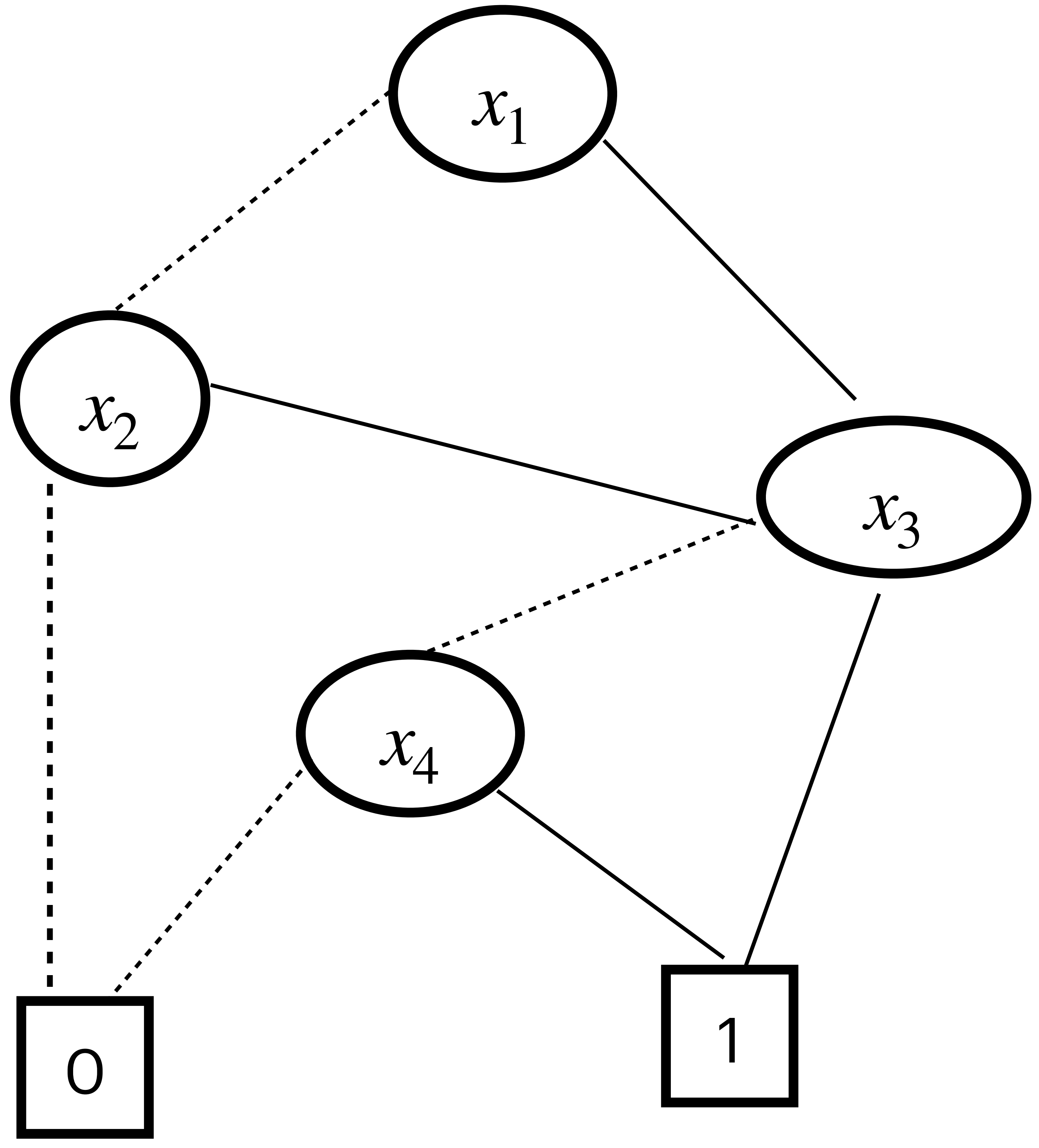
Model Counting in ROBDD?

Model Counting

$F = (x_1 \vee x_2) \wedge (x_3 \vee x_4)$ $x_1 > x_2 > x_3 > x_4$

Model Counting

$F = (x_1 \vee x_2) \wedge (x_3 \vee x_4)$ $\quad x_1 > x_2 > x_3 > x_4$



ROBDD vs CNF

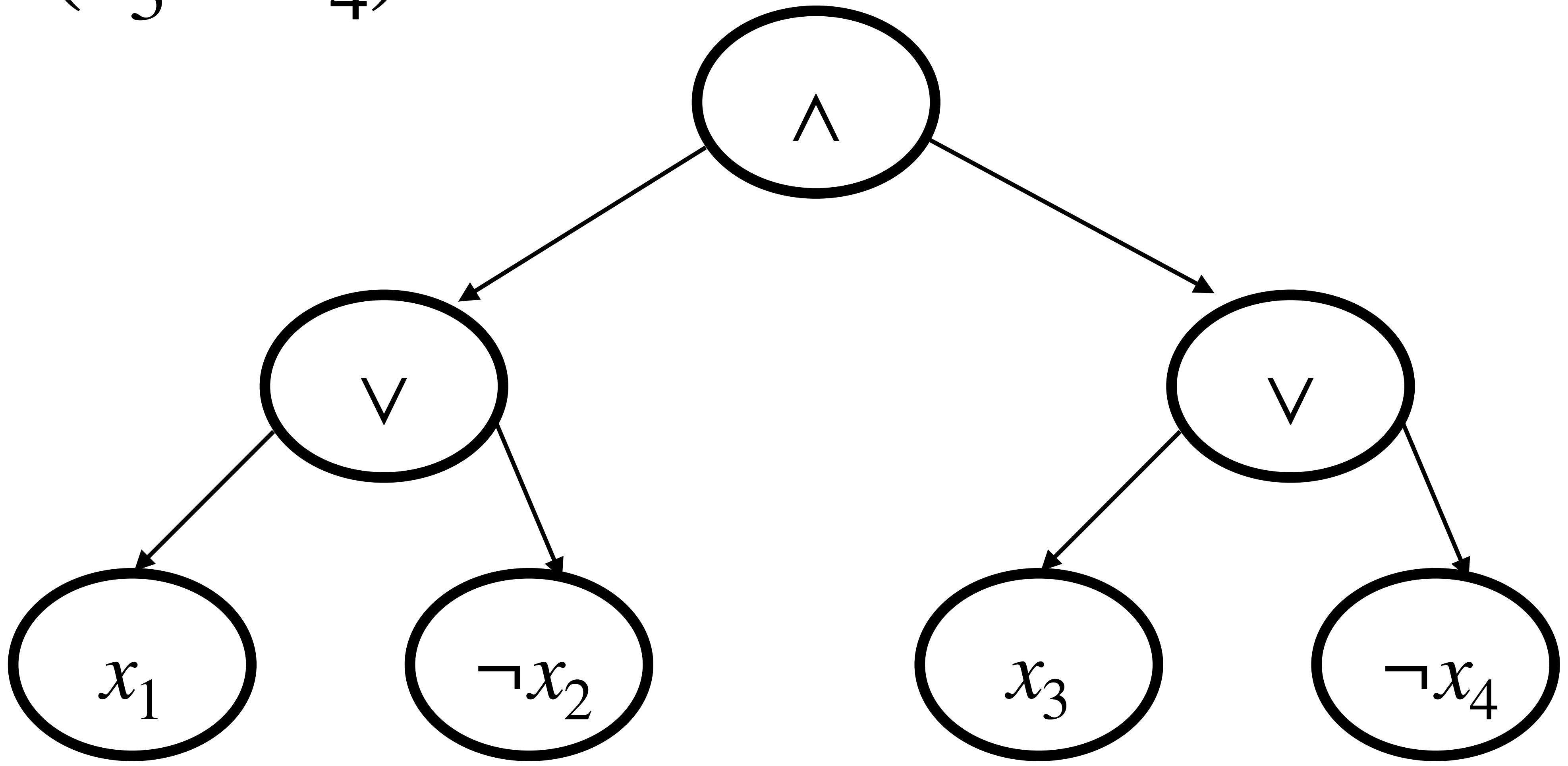
	CNF	ROBDD
SAT	NP-Hard	$O(\|F_{ROBDD}\|)$
Model Count	#P	$O(\|F_{ROBDD}\|)$
UNSAT	Co-NP	$O(1)$

Different Compilation Forms

NNF: Normal Negation Form

1. Each non-terminal node is either $\wedge$ or $\vee$
2. Each terminal node is either a literal or 0 or 1

$F = (x_1 \vee \neg x_2) \wedge (x_3 \vee x_4)$



Different Compilation Forms

d-NNF: Deterministic Normal Negation Form (*Darwiche* 1998)

A NNF is deterministic if for every $\vee$ (OR) node with children $\{c_1, c_2, \ldots, c_k\}$ following holds:

$$\forall i \neq j \; Models(c_i) \cap Models(c_j) = \varnothing$$

Any two children of $\vee$ (OR) node don't share models

Different Compilation Forms

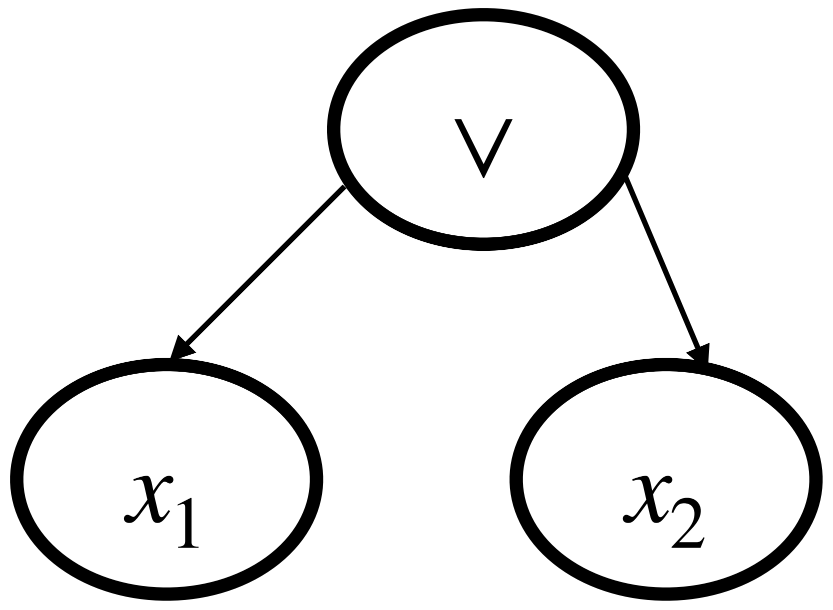
d-NNF: Deterministic Normal Negation Form (*Darwiche* 1998)

A NNF is deterministic if for every $\vee$ (OR) node with children $\{c_1, c_2, \dots, c_k\}$ following holds:

$$\forall i \neq j \; Models(c_i) \cap Models(c_j) = \varnothing$$

Any two children of $\vee$ (OR) node don't share models

$F = x_1 \vee x_2$



$Models(x_1) = \{(x_1, x_2), (x_1, \neg x_2)\}$

$Models(x_2) = \{(x_1, x_2), (\neg x_1, x_2)\}$

Different Compilation Forms

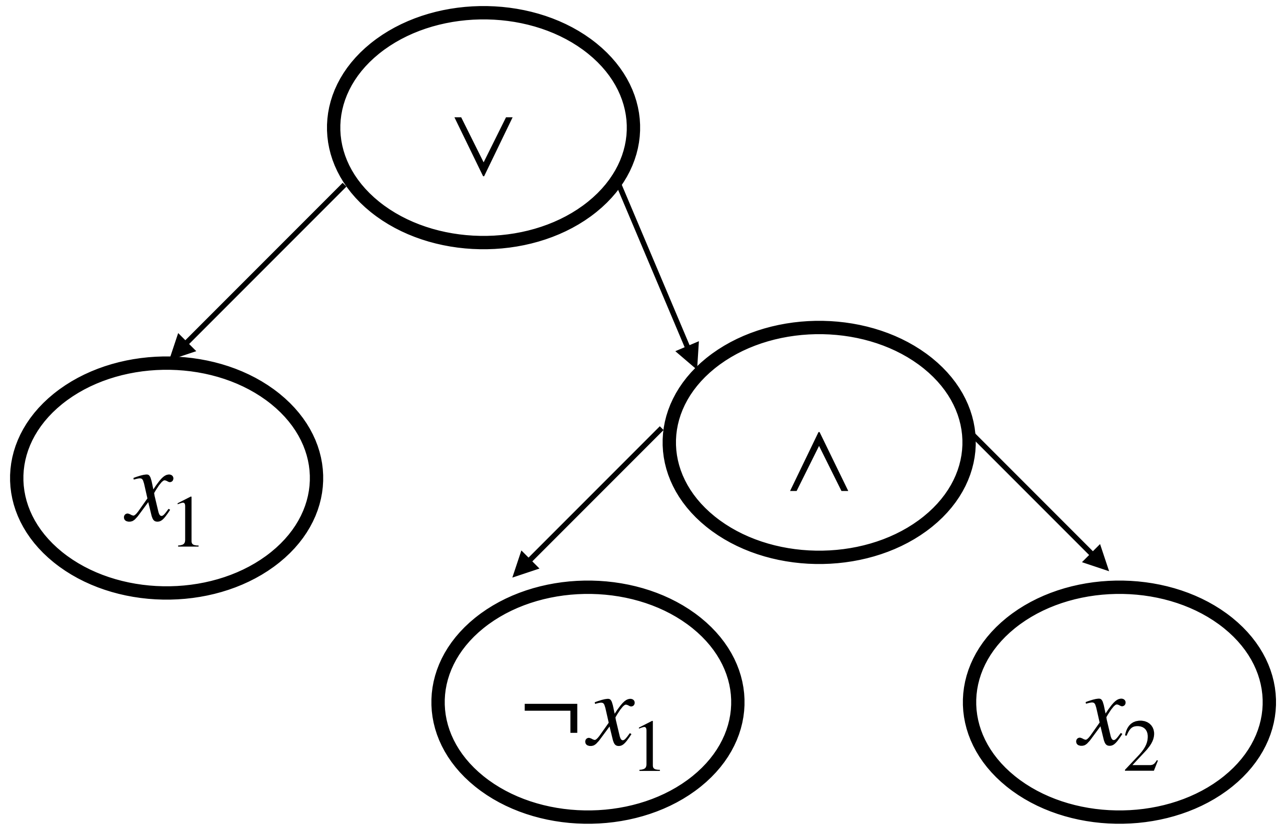
d-NNF: Deterministic Normal Negation Form (*Darwiche* 1998)

A NNF is deterministic if for every $\lor$ (OR) node with children $\{c_1, c_2, \ldots, c_k\}$ following holds:

$$\forall i \neq j \; Models(c_i) \cap Models(c_j) = \emptyset$$

Any two children of $\lor$ (OR) node don't share models

$F = x_1 \lor x_2$



$F \; in \; d - NNF$

Different Compilation Forms

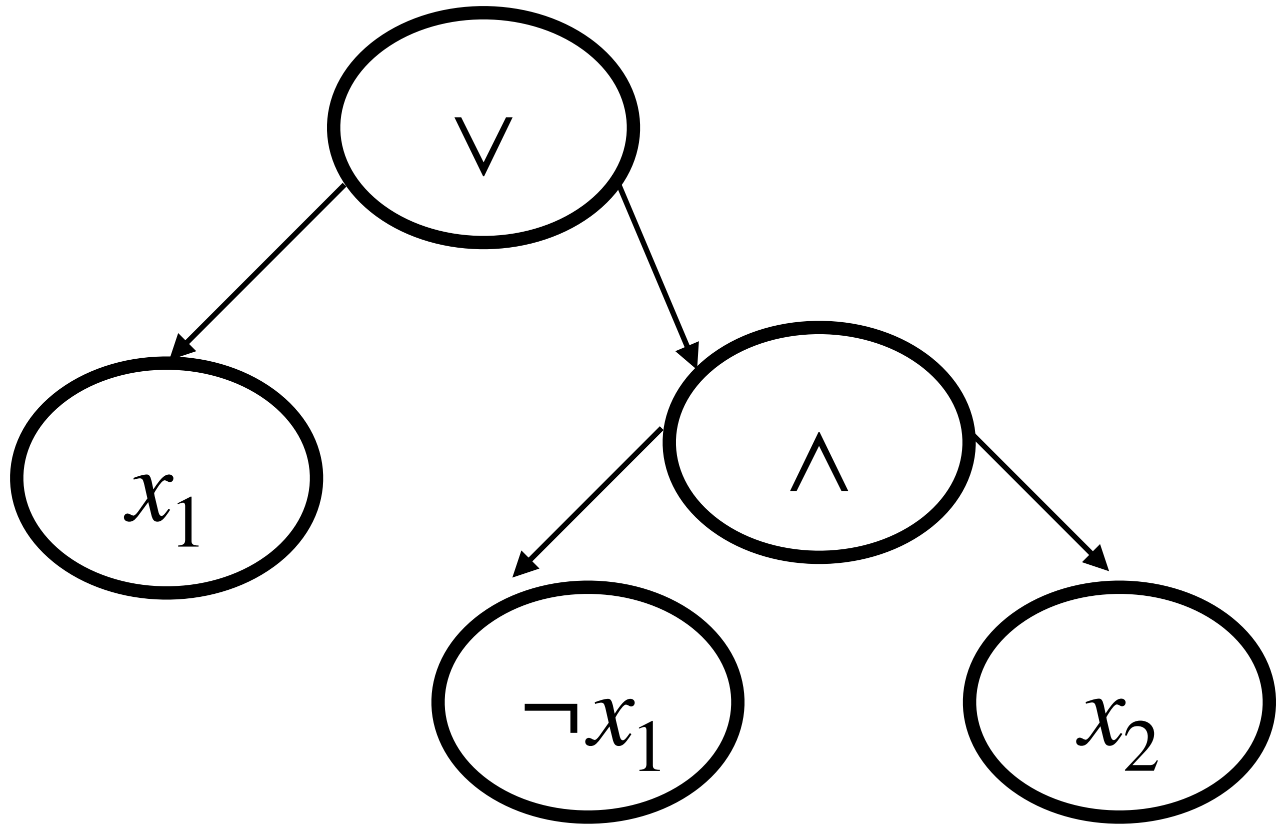
DNNF: Decomposable Normal Negation Form (*Darwiche* 2011)

A NNF is decomposable if for every $\wedge$ (AND) node with children $\{c_1, c_2, \ldots, c_k\}$ following holds:

$$\forall i \neq j \; Vars(c_i) \cap Vars(c_j) = \emptyset$$

Any two children of $\wedge$ (AND) node don't share variables/literals

$F = x_1 \vee x_2$



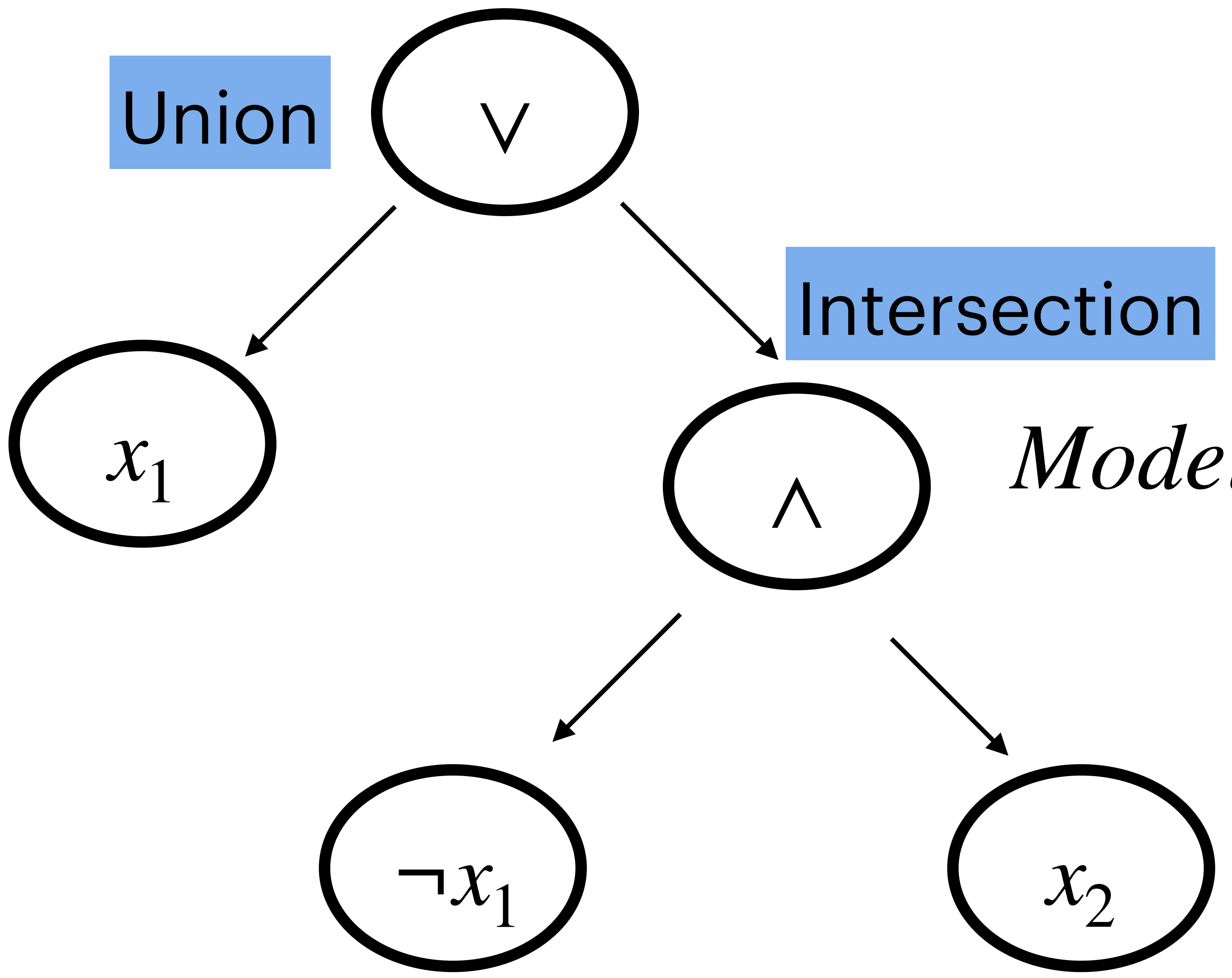
F in DNNF

Model Counting in d-DNNF

$F = x_1 \vee x_2$

$Models(\vee) = \{(x_1, x_2), (x_1, \neg x_2), (\neg x_1, x_2)\}$

$Models(x_1) = \{(x_1, x_2), (x_1, \neg x_2)\}$



$Models(\wedge) = \{(\neg x_1, x_2)\}$

$Models(\neg x_1) = \{(\neg x_1, x_2), (\neg x_1, \neg x_2)\}$

$Models(x_2) = \{(\neg x_1, x_2), (\neg x_1, x_2)\}$

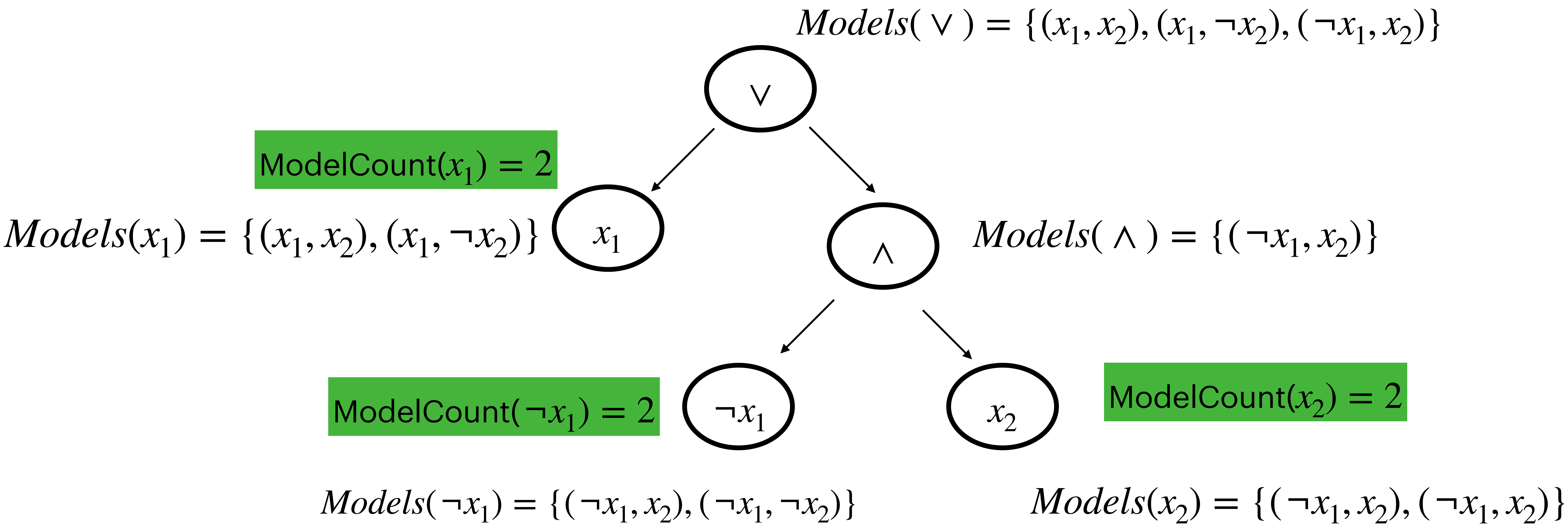
$F \ in \ d - DNNF$

We can’t store models at every node! We need to store count!

Model Counting in d-DNNF

Model count of a terminal node:

1. If node is 0, then Model count is 0
2. If node is 1, then Model count is $2^{|Vars(F)|}$
3. If node is a literal, Model count is $2^{|Vars(F)-1|}$



Model Counting in d-DNNF

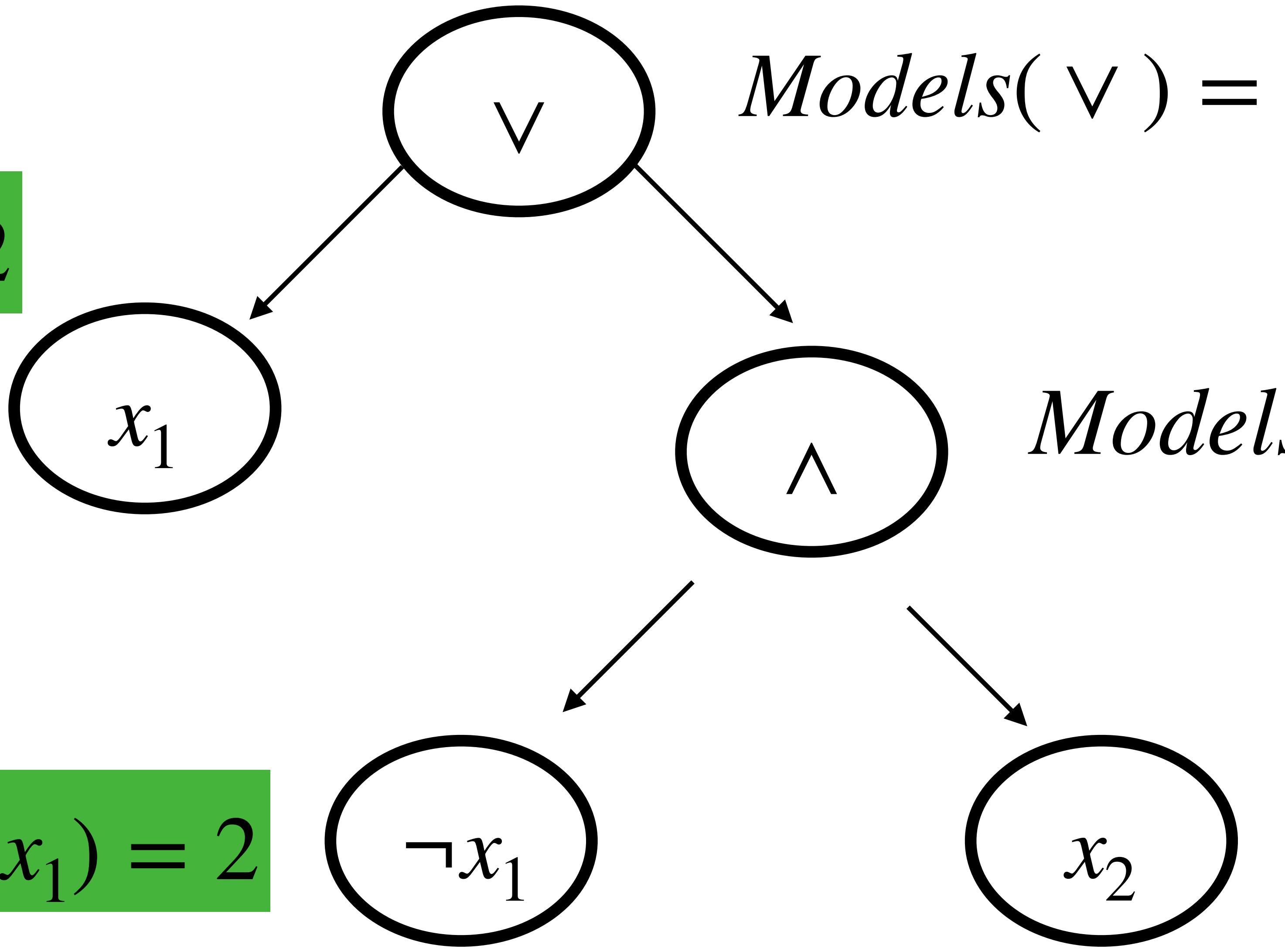
Model count of a AND node with children $\{c_1, c_2, \ldots, c_k\}$

$$\prod_{i \in [1,k]} \frac{ModelCount(c_i)}{2^{|Vars(F) - Vars(c_i)|}}$$

Children don't share a literal, each child may wrongly assign these missing literals and thus overcount.

$$\prod_{i \in [1,k]} \frac{ModelCount(c_i)}{2^{|Vars(F) - Vars(c_i)|}} \times 2^{|vars(F) - \bigcup_{i \in [1,k]} Vars(c_i)|}$$

we need to account for the fact that the variables that are not assigned in $\wedge$



$Models(\vee) = \{(x_1, x_2), (x_1, \neg x_2), (\neg x_1, x_2)\}$

ModelCount$(x_1) = 2$

ModelCount$(\wedge) = \frac{2}{2} \times \frac{2}{2} \times 1$

$Models(x_1) = \{(x_1, x_2), (x_1, \neg x_2)\}$

$Models(\wedge) = \{(\neg x_1, x_2)\}$

ModelCount$(\neg x_1) = 2$

ModelCount$(x_2) = 2$

$Models(\neg x_1) = \{(\neg x_1, x_2), (\neg x_1, \neg x_2)\}$

$Models(x_2) = \{(\neg x_1, x_2), (\neg x_1, x_2)\}$

Model Counting in d-DNNF

Model count of a OR node with children $\{c_1, c_2, \ldots, c_k\}$

$$\sum_{i \in [1,k]} ModelCount(c_i)$$

Children don’t share models

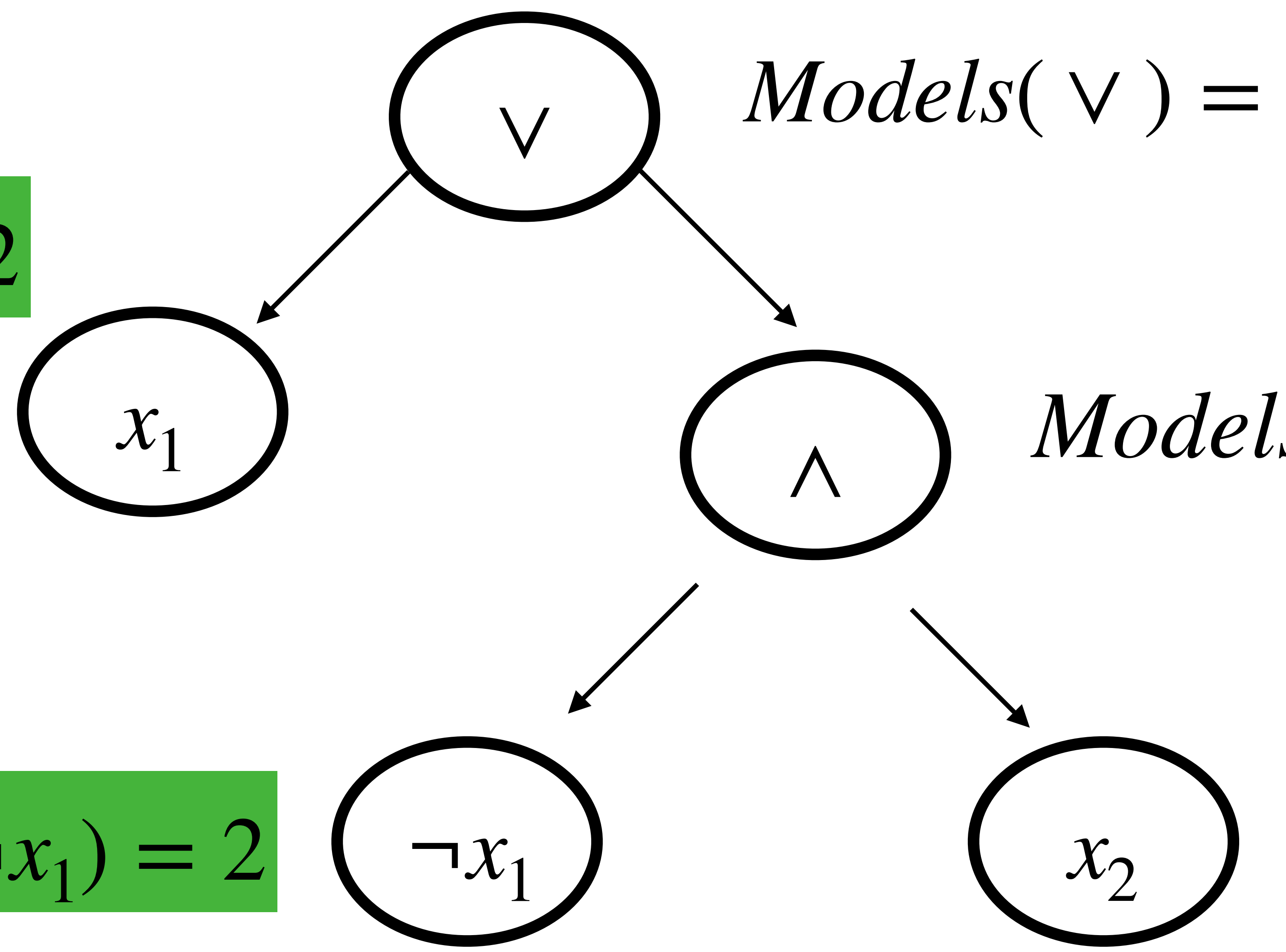
ModelCount($\vee$) $= 3$

$Models(\vee) = \{(x_1, x_2), (x_1, \neg x_2), (\neg x_1, x_2)\}$

ModelCount(x_1) $= 2$

ModelCount($\wedge$) $= \frac{2}{2} \times \frac{2}{2} \times 1$

$Models(x_1) = \{(x_1, x_2), (x_1, \neg x_2)\}$



$Models(\wedge) = \{(\neg x_1, x_2)\}$

ModelCount($\neg x_1$) $= 2$

ModelCount(x_2) $= 2$

$Models(\neg x_1) = \{(\neg x_1, x_2), (\neg x_1, \neg x_2)\}$

$Models(x_2) = \{(\neg x_1, x_2), (\neg x_1, x_2)\}$

Model Counting in d-DNNF

Model count of a terminal node:

1. If node is 0, then Model count is 0
2. If node is 1, then Model count is $2^{|Vars(F)|}$
3. If node is a literal, Model count is $2^{|Vars(F)-1|}$

Model count of a AND node with children $\{c_1, c_2, \ldots, c_k\}$

$$\prod_{i\in[1,k]} \frac{ModelCount(c_i)}{2^{|Vars(F)-Vars(c_i)|}} \times 2^{|vars(F)-\bigcup_{i\in[1,k]} Vars(c_i)|}$$

Model count of a OR node with children $\{c_1, c_2, \ldots, c_k\}$

$$\sum_{i\in[1,k]} ModelCount(c_i)$$

Model Counting in d-DNNF

$$F = (x_1 \vee x_2 \vee x_3)$$

In order to convert this to d-NNF,
Shannan Expansion:

$$F(x_1, x_2) = F(1, x_2) \vee F(0, x_2)$$

$$(x_1 \wedge (1 \vee x_2 \vee x_3)) \vee (\neg x_1 \wedge (0 \vee x_2 \vee x_3))$$

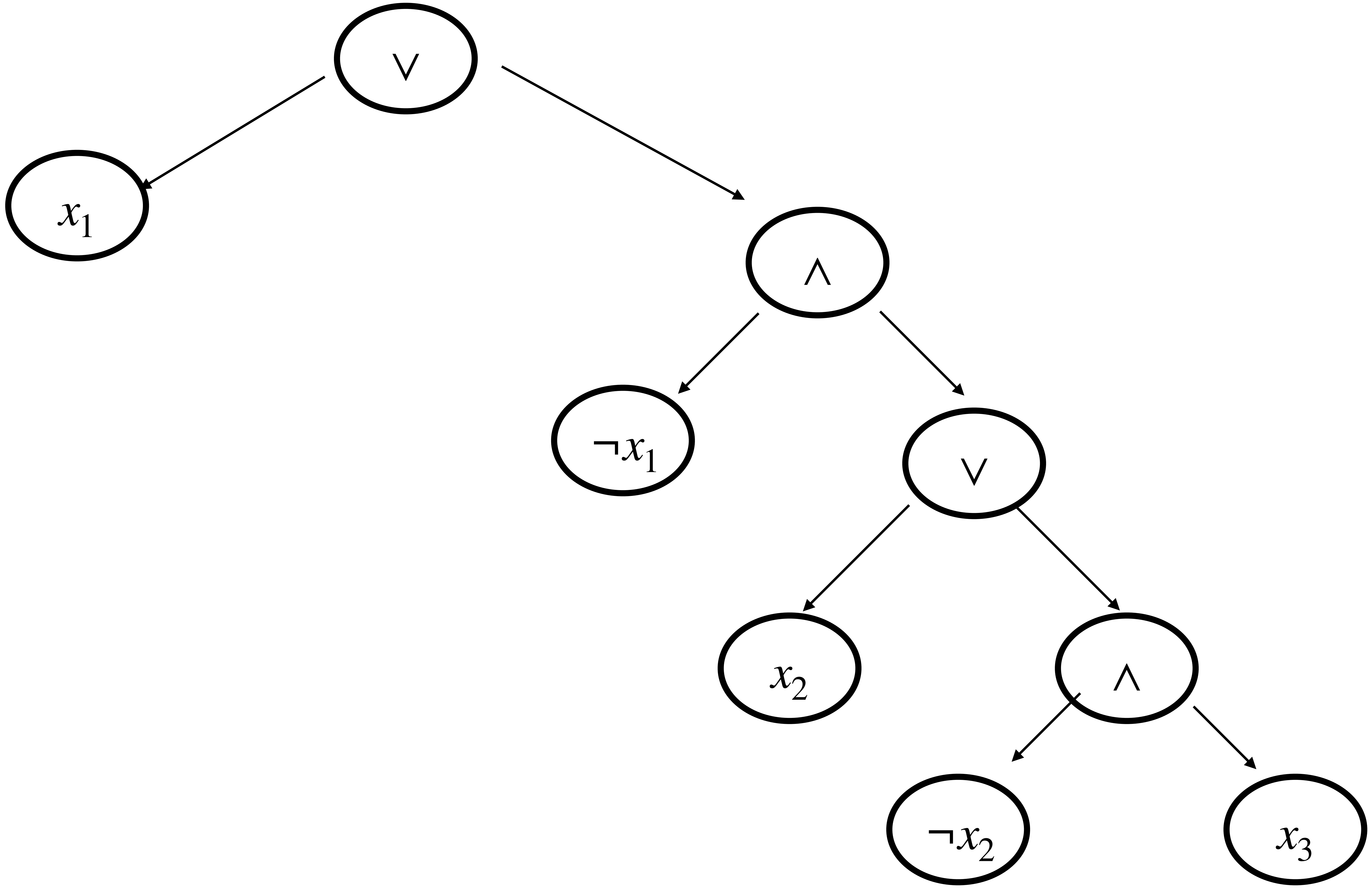
$$(x_1) \vee (\neg x_1 \wedge (x_2 \vee x_3))$$

$$(x_1) \vee (\neg x_1 \wedge ((x_2 \wedge (1 \vee x_3)) \vee (\neg x_2 \wedge (0 \vee x_3)))$$

$$(x_1) \vee (\neg x_1 \wedge (x_2 \vee (\neg x_2 \wedge x_3)))$$

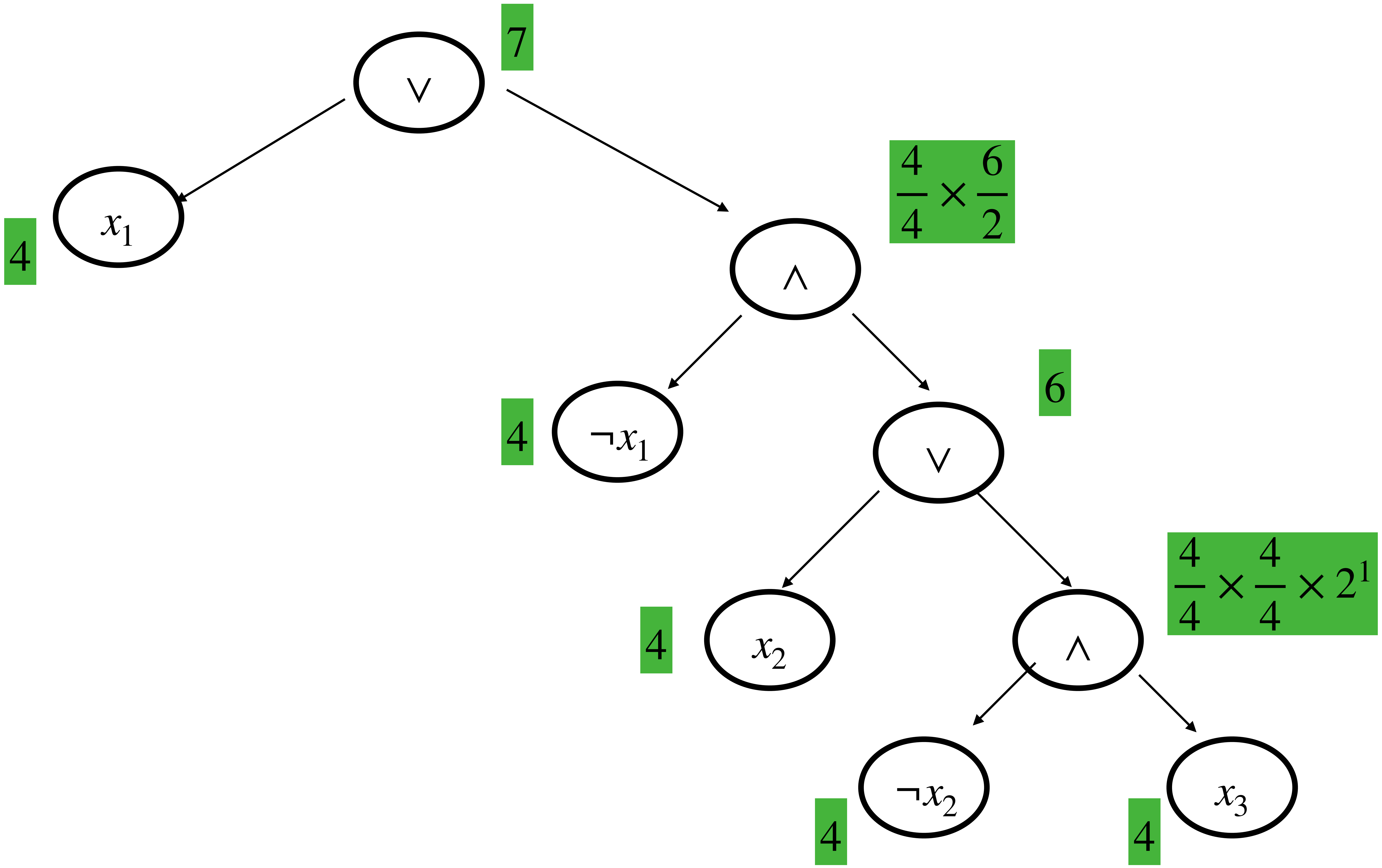
Model Counting in d-DNNF

$$F = (x_1 \vee x_2 \vee x_3) \equiv (x_1) \vee (\neg x_1 \wedge (x_2 \vee (\neg x_2 \wedge x_3)))$$



Model Counting in d-DNNF

$$F = (x_1 \vee x_2 \vee x_3) \equiv (x_1) \vee (\neg x_1 \wedge (x_2 \vee (\neg x_2 \wedge x_3)))$$

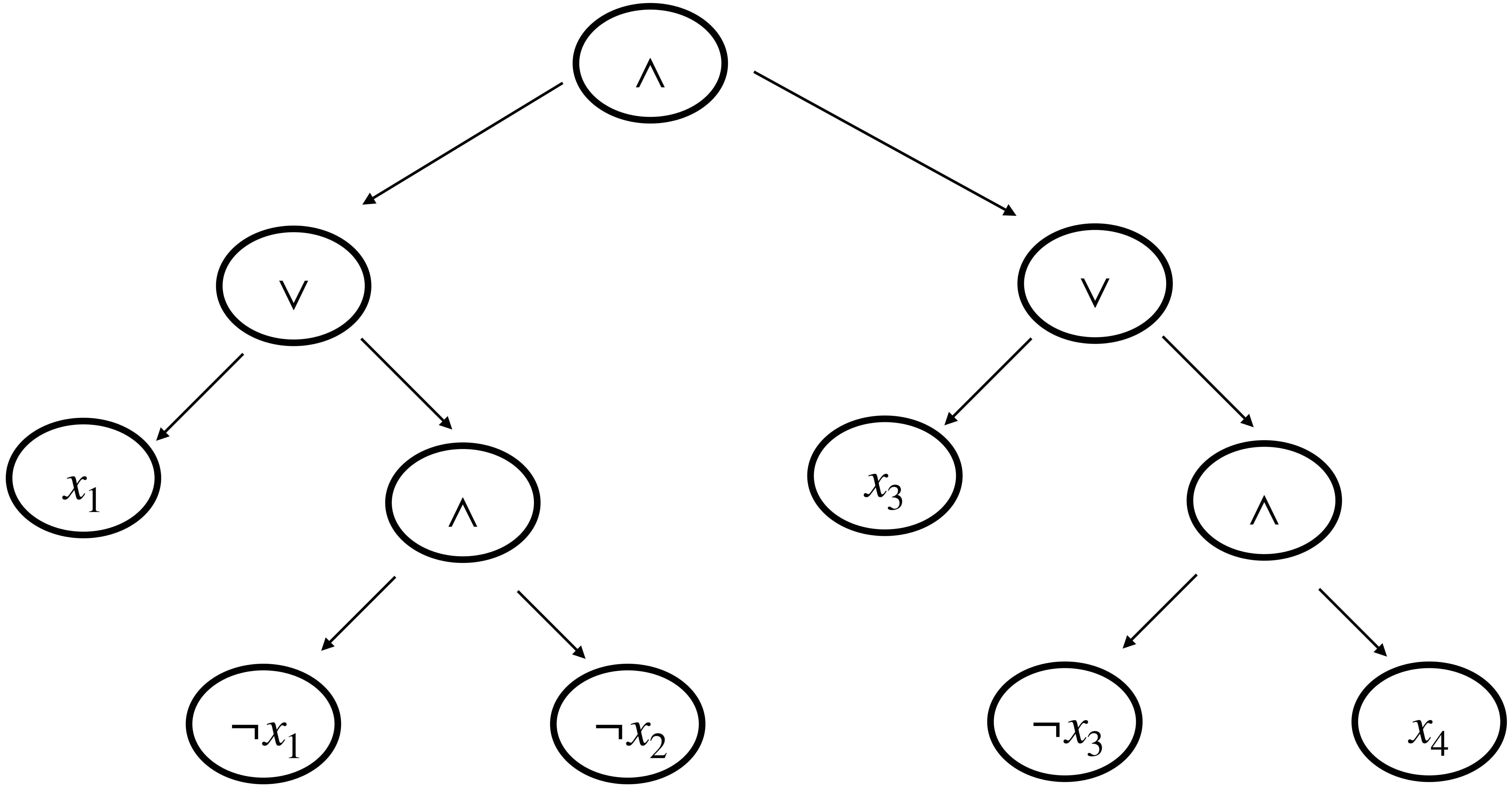


Model Counting in d-DNNF

$$F = (x_1 \vee \neg x_2) \wedge (x_3 \vee x_4)$$

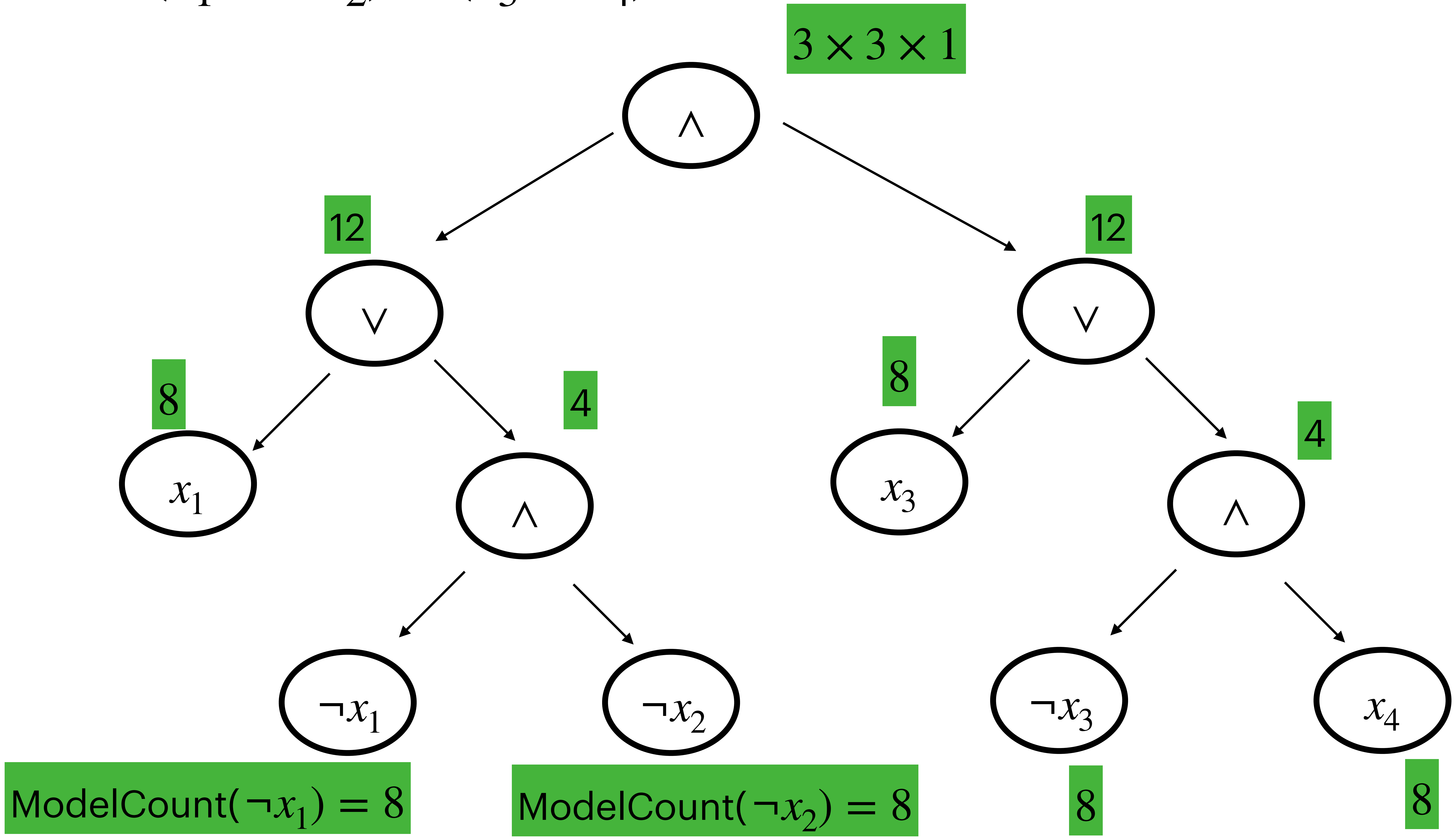
Model Counting in d-DNNF

$$F = (x_1 \lor \neg x_2) \land (x_3 \lor x_4)$$



Model Counting in d-DNNF

$$F = (x_1 \vee \neg x_2) \wedge (x_3 \vee x_4)$$



Model Counting in d-DNNF

$$F = (x_1 \vee x_2) \wedge (\neg x_1 \vee x_3)$$

Shannon Expansion on common variables.

$$F = x_1 \wedge ((1 \vee x_2) \wedge (\neg 1 \vee x_3)) \quad \vee (\neg x_1 \wedge ((0 \vee x_2) \wedge (\neg 0 \vee x_3)))$$

$$F = (x_1 \wedge x_3) \ \vee (\neg x_1 \wedge x_2)$$

Model Counting in d-DNNF

Tools like d4, c2d, Dsharp for conversion

Just like ROBDD, may result in exponential size formula, but model counting is linear in the size of the formula

Efficient model counter, GANAK

By Shubham Sharma, a dual-degree student from IITK as his MTP project