Proofs, Circuits, Communication

Robert Robere
School of Computer Science
McGill University

McGill

Reflections on Proofs in Algorithms and Complexity

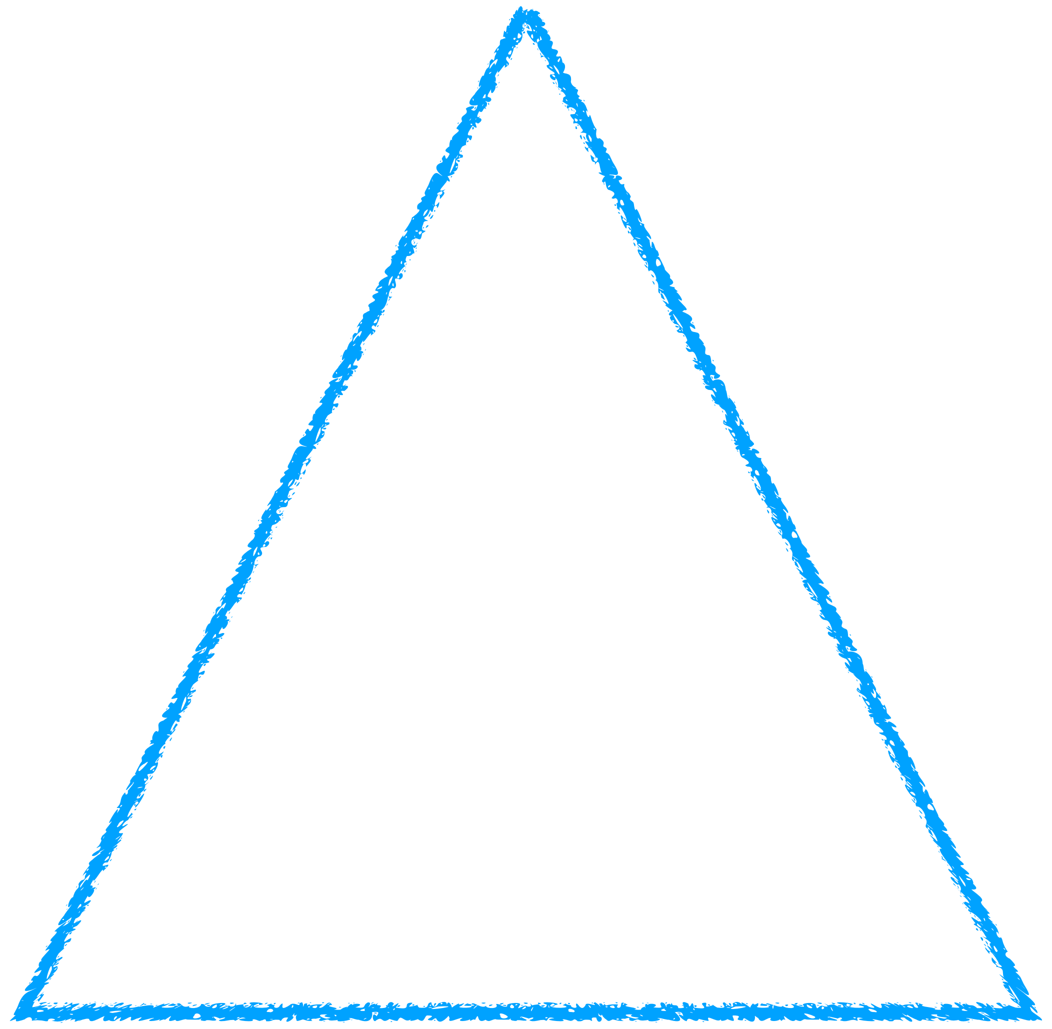
FOCS 2021

February 8, 2022

Starting Point: Lifting Theorems

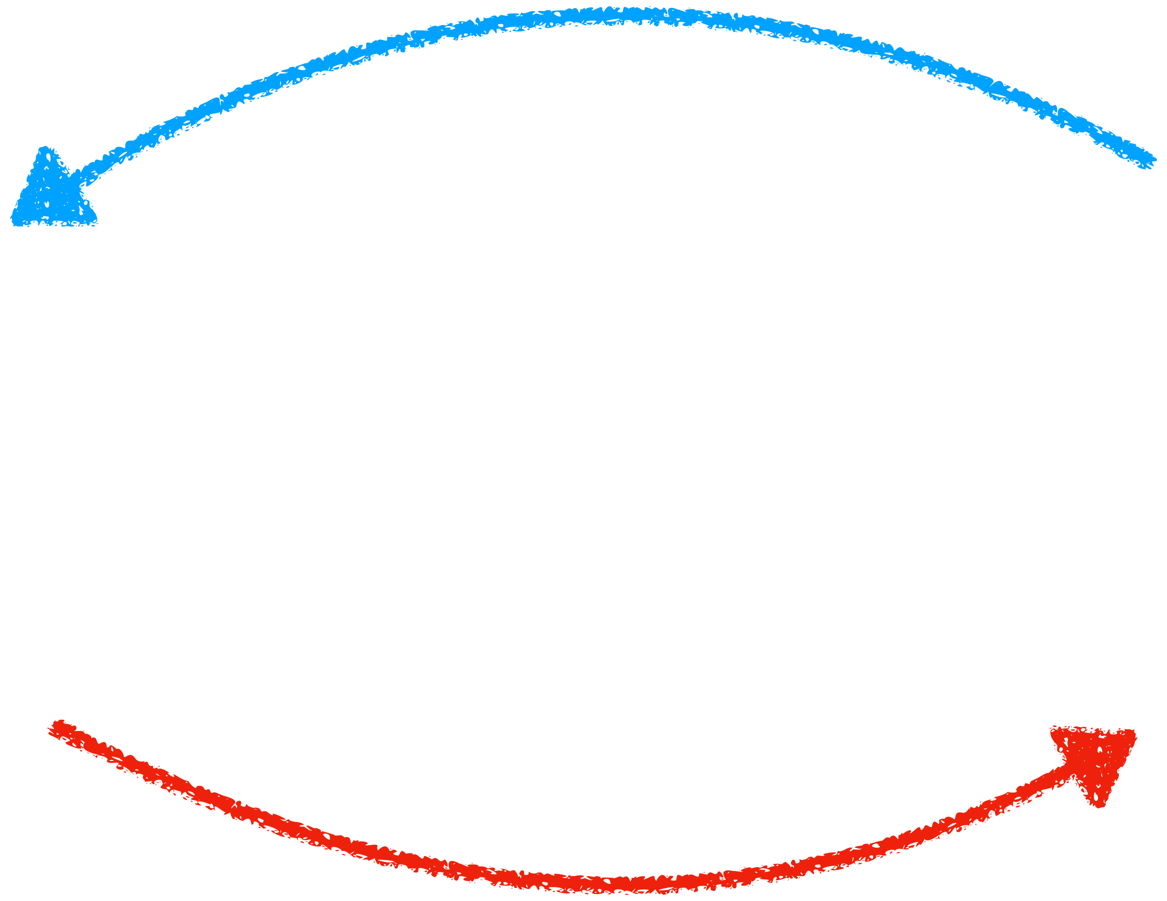
- Recent explosion of **lifting theorems** in query and communication complexity

Decision Tree



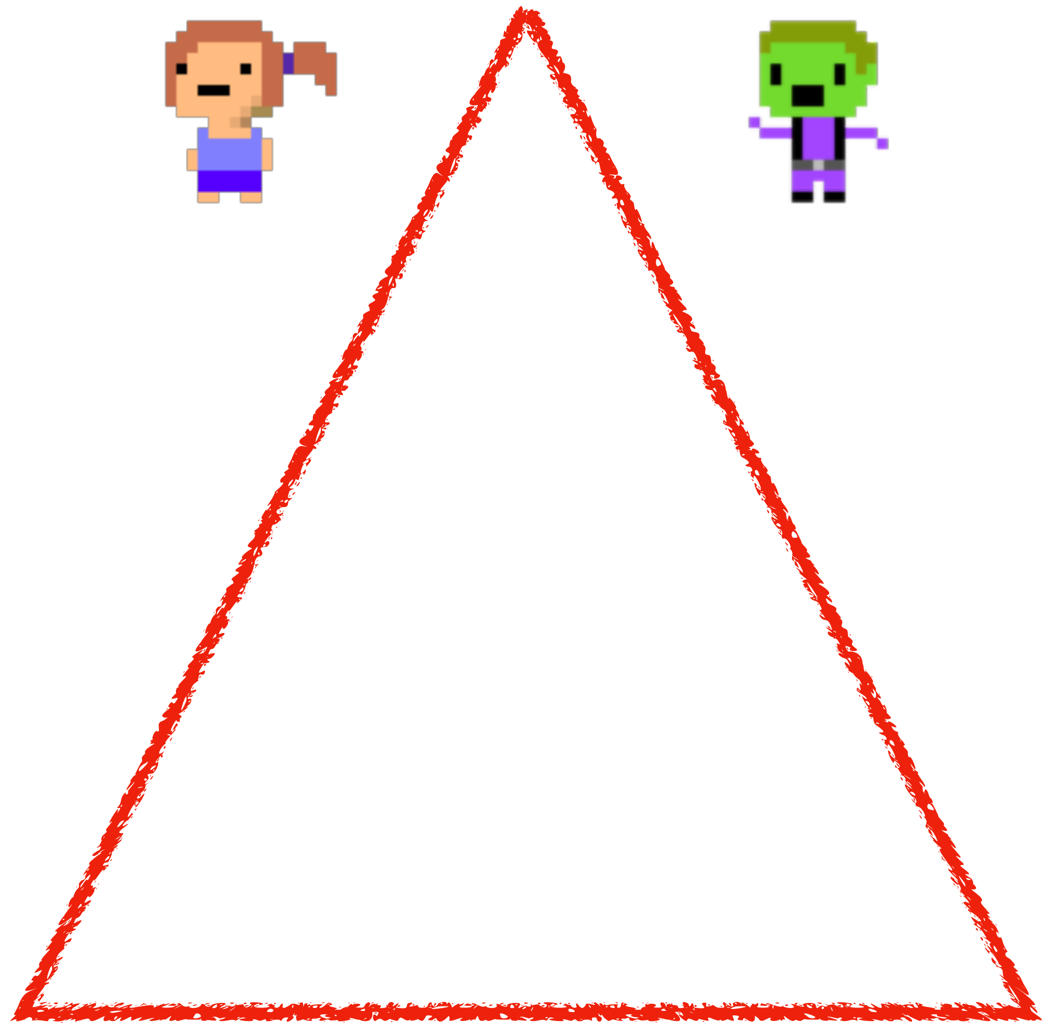
$f : \{0,1\}^n \to \{0,1\}$

Protocol simulates Tree



For “complex” g
this is **best** strategy!

Communication Protocol



$f \circ g^n : X^n \times Y^n \to \{0,1\}$

$g : X \times Y \to \{0,1\}$ is a “complex gadget”

Lifting Theorems and Proof Complexity

- Many new results in proof and circuit complexity using lifting theorems
  [GP12, GPW14, GLMWZ15, CLRS16, LRS16, **R**PRC16, P**R**17, KMR17, P**R**18, dRNV 16, GGKS18, GK**R**S18, dRMNP**R**18, dRMNP**R**V20, FGG**R**2022, LMMPZ22]

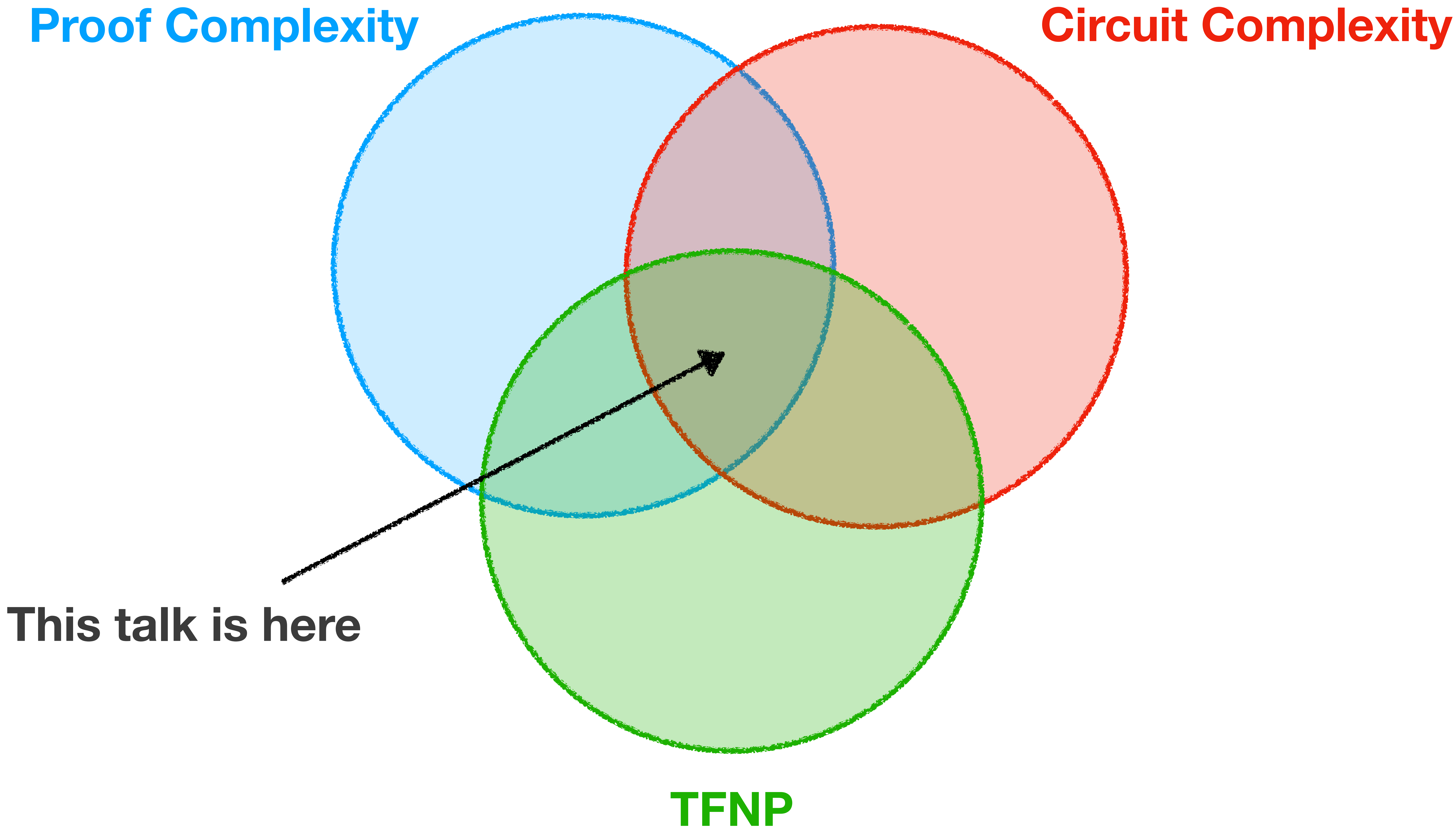
- The proofs of these results place **total search problems** at center stage!

  1. The **False Clause Search Problem** $\mathsf{Search}(F)$ for unsatisfiable CNF F

  2. The **Karchmer-Wigderson Game** $\mathsf{KW}(f)$ for boolean functions f

- To apply techniques, need **query models** that capture proof systems, **communication models** that capture circuit classes.

- Recent work ([GK**R**S18] building on [BCEIP98], closely related to [BK94, K94, …]) suggested using TNFP classes as a guide to find these models.


Proof Complexity
Circuit Complexity
This talk is here
TFNP

Goal for Today

- Tell two stories:
    - The False Clause Search Problem and Proof Complexity
    - The Karchmer-Wigderson Games and Circuit Complexity
- These are “two pieces” of a bigger theory of query/comm. TFNP
- Outline a research program using TFNP as a guide to capture proof systems and circuit classes.
    - (Close thematic links with Sam’s and Neil’s talks earlier.)

Part 1

Proofs and the False Clause Search Problem

False Clause Search Problem

- Focus on complexity of refuting **unsatisfiable CNF formulas**

$$F = C_1 \wedge C_2 \wedge \cdots \wedge C_m$$

- Each C_i is a **clause** (disjunction of boolean literals)
- F has an associated **total search problem:**

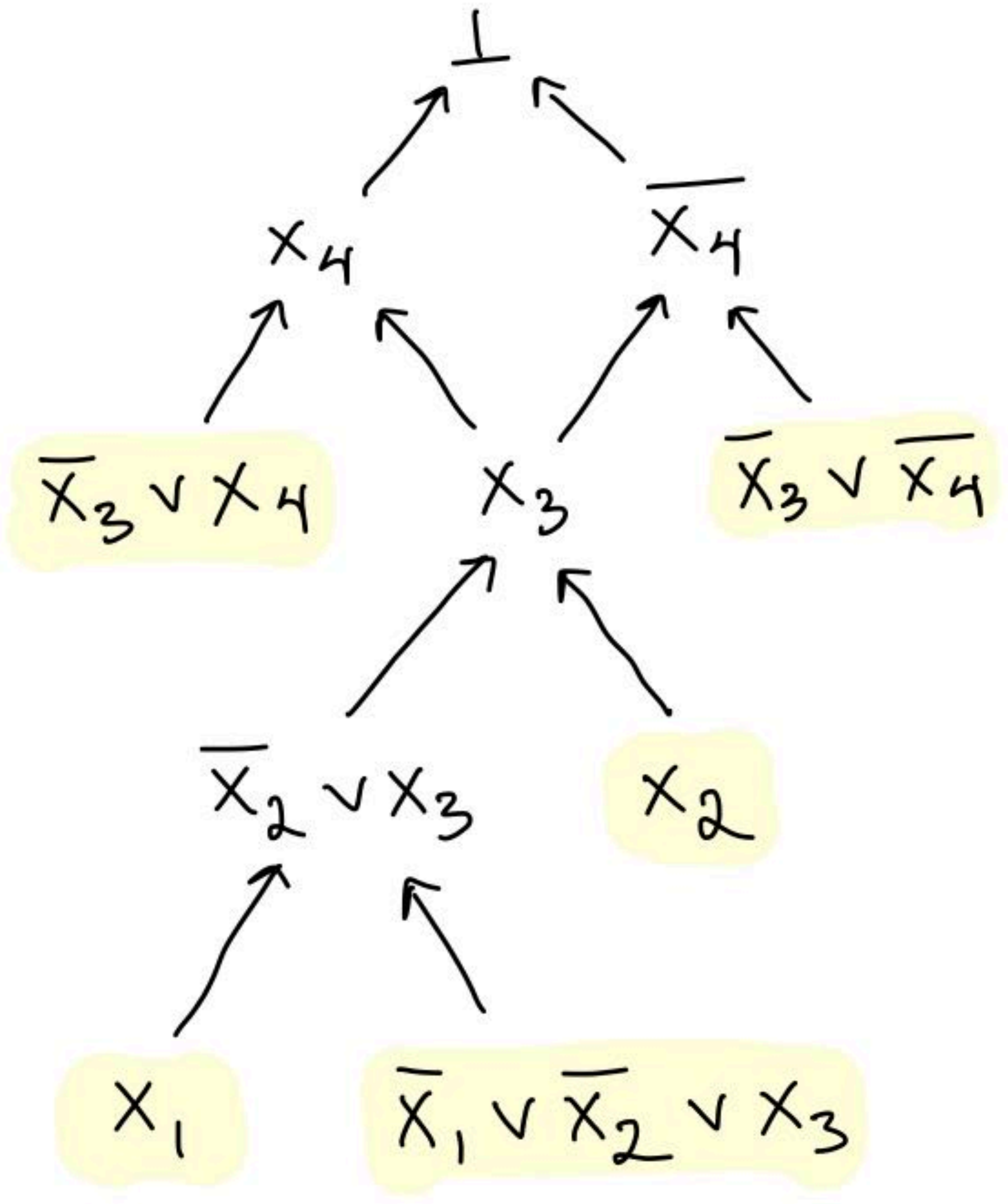
$$\text{Search}(F) \subseteq \{0,1\}^n \times [m]$$

Given $x \in \{0,1\}^n$, *find* $i \in [m]$ *such that* $C_i(x) = 0.$

- **Query Complexity** of $\text{Search}(F) \equiv$ **Proof Complexity** of F

Resolution Proofs

- Lines are **clauses**.
- New lines deduced using
  - **Resolution Rule**: $C \vee x, D \vee \bar{x} \vdash C \vee D$
- **Length**: Number of lines.
- **Depth**: Length of longest path.
- Proof is **tree-like** if each clause is used at most once.
  - Input clauses can be copied any number of times

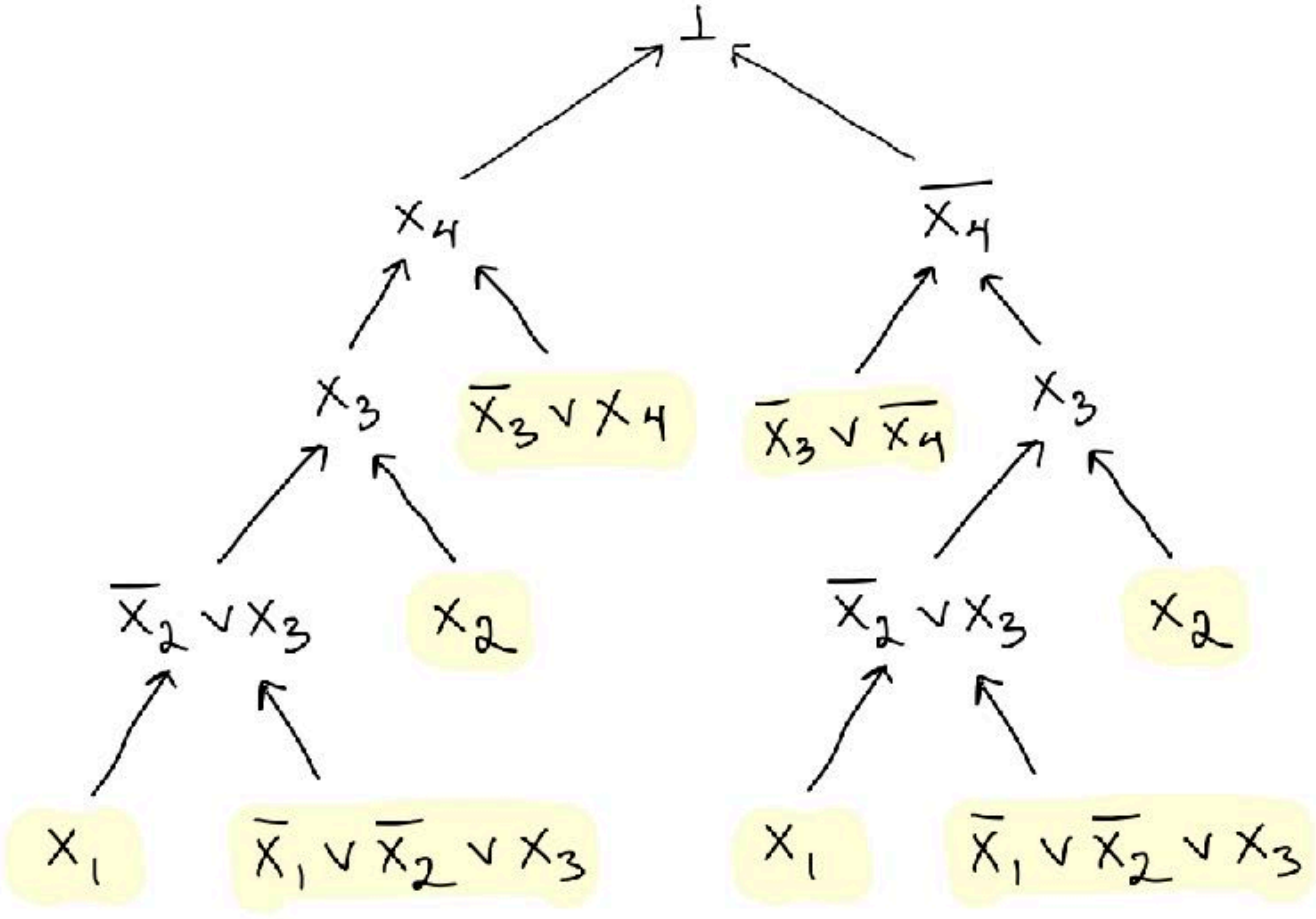


Example. $F = x_1 \wedge x_2 \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_3 \vee x_4) \wedge (\bar{x}_3 \vee \bar{x}_4)$

Length: 10, **Depth:** 4

Resolution Proofs

- Lines are **clauses**.
- New lines deduced using
  - **Resolution Rule**: $C \vee x, D \vee \bar{x} \vdash C \vee D$
- **Length**: Number of lines.
- **Depth**: Length of longest path.
- Proof is **tree-like** if each clause is used at most once.
  - Input clauses can be copied any number of times



Example. $F = x_1 \wedge x_2 \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_3 \vee x_4) \wedge (\bar{x}_3 \vee \bar{x}_4)$

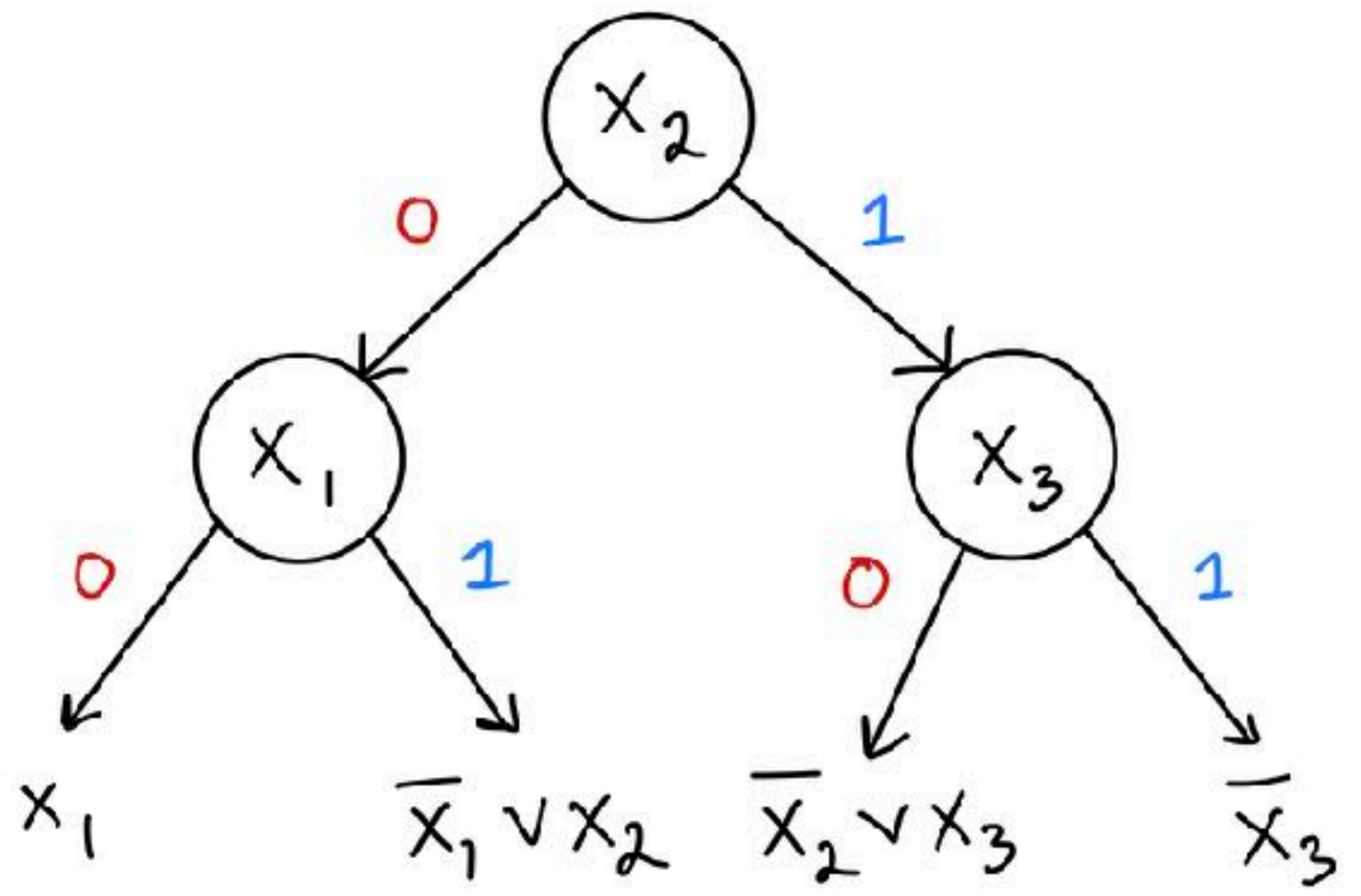
Length: 10, **Depth:** 4

Decision Trees for $\mathsf{Search}(F)$

- **Size**: Number of nodes
- **Depth**: Length of longest path
- Given boolean assignment, follow unique path consistent with that assignment, output violated clause.
    - Decision tree for $\mathsf{Search}(F)$ is related to the **DPLL method** for solving SAT.

$\mathsf{Search}(F) \subseteq \{0,1\}^n \times [m]$

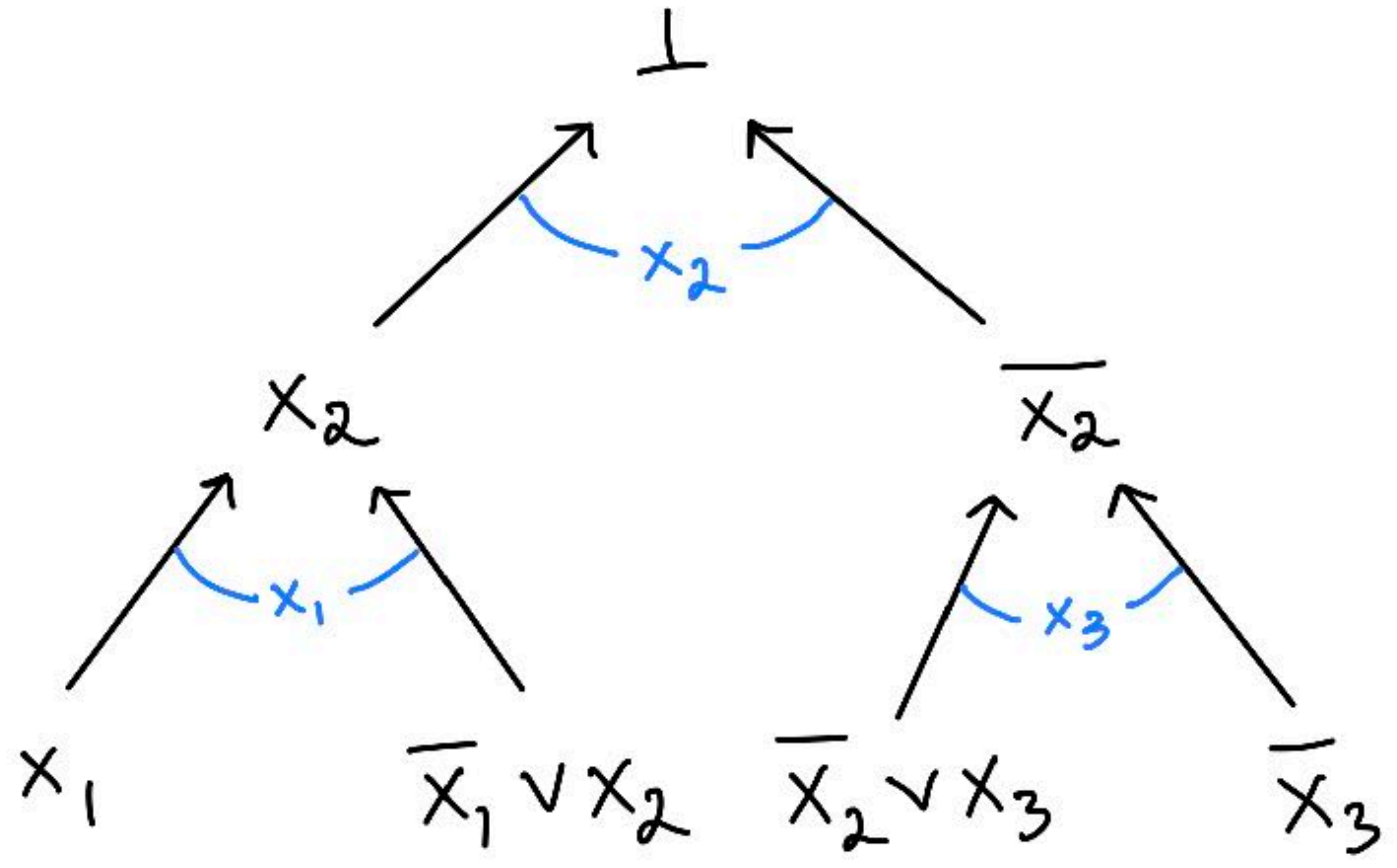
Given $x \in \{0,1\}^n$*, find* $i \in [m]$ *such that* $C_i(x) = 0$.



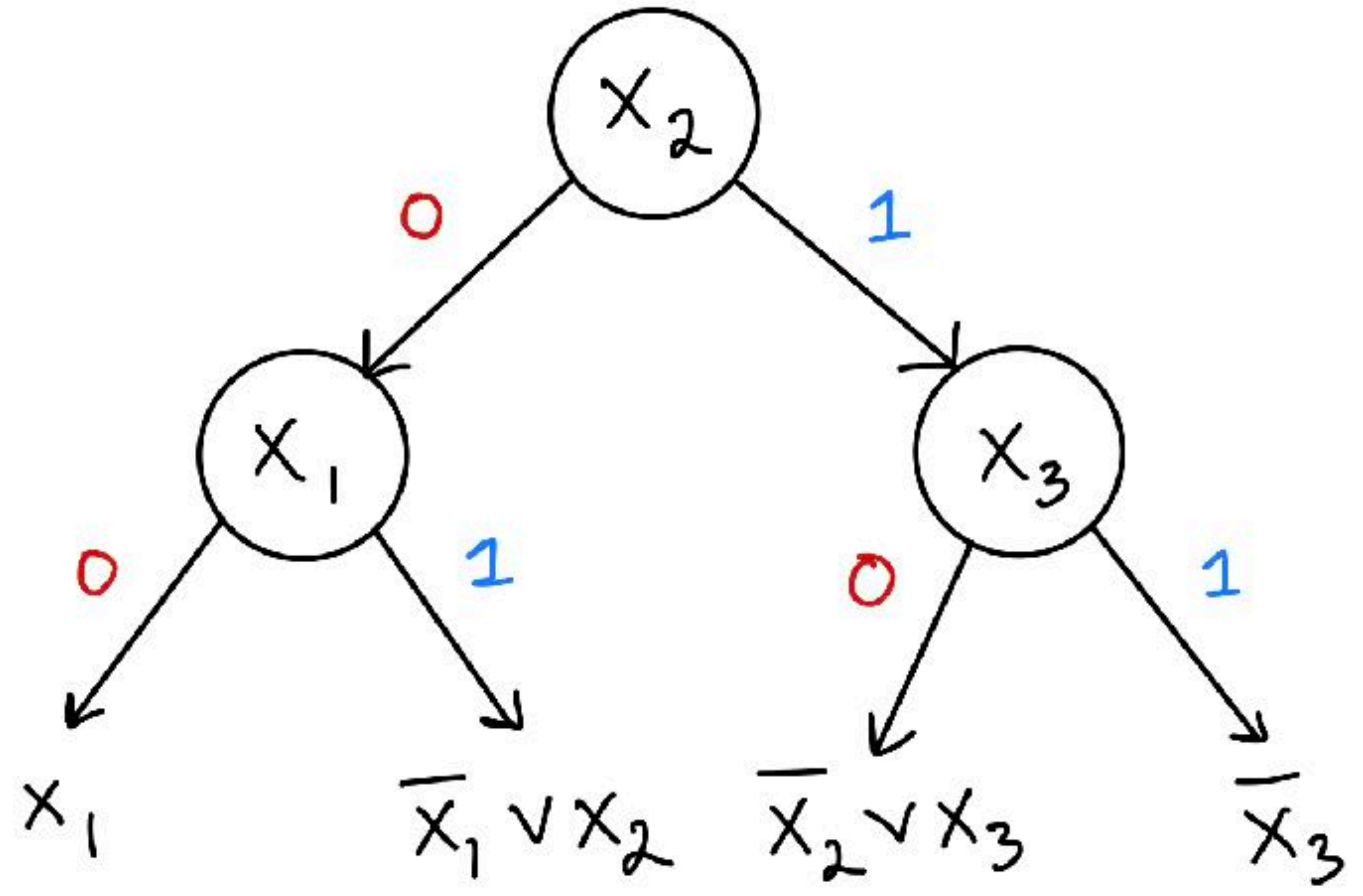
$F = x_1 \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_2 \vee x_3) \wedge \bar{x}_3$

Tree-Like Resolution $\equiv$ Decision Trees

Tree-Like Resolution of F



Decision Tree for Search(F)

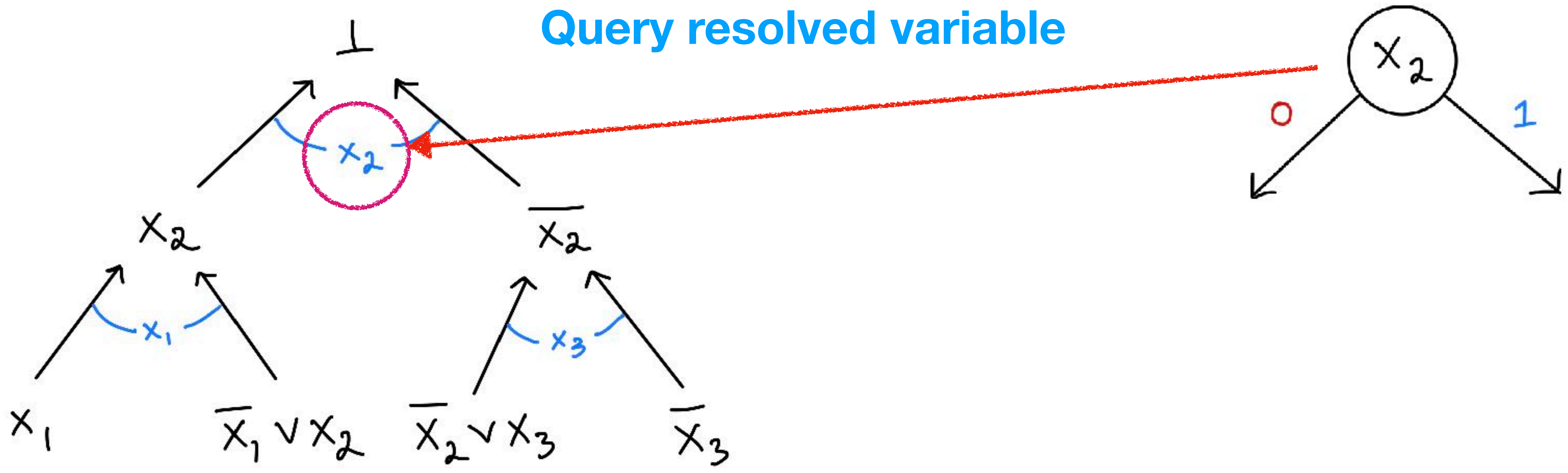


$$F = x_1 \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_2 \vee x_3) \wedge \bar{x}_3$$

Tree-Like Resolution $\equiv$ Decision Trees

Tree-Like Resolution of F

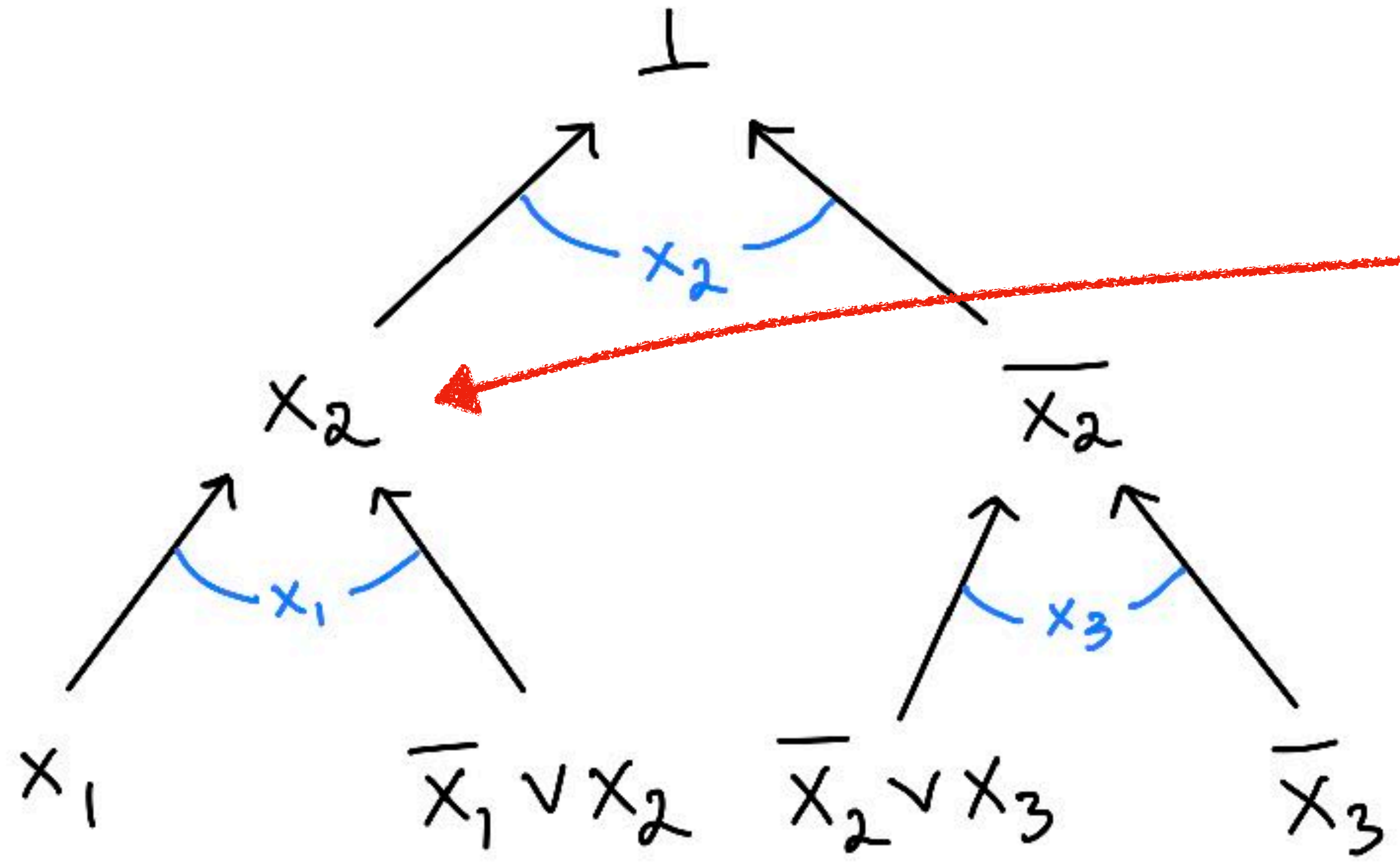
Decision Tree for Search(F)



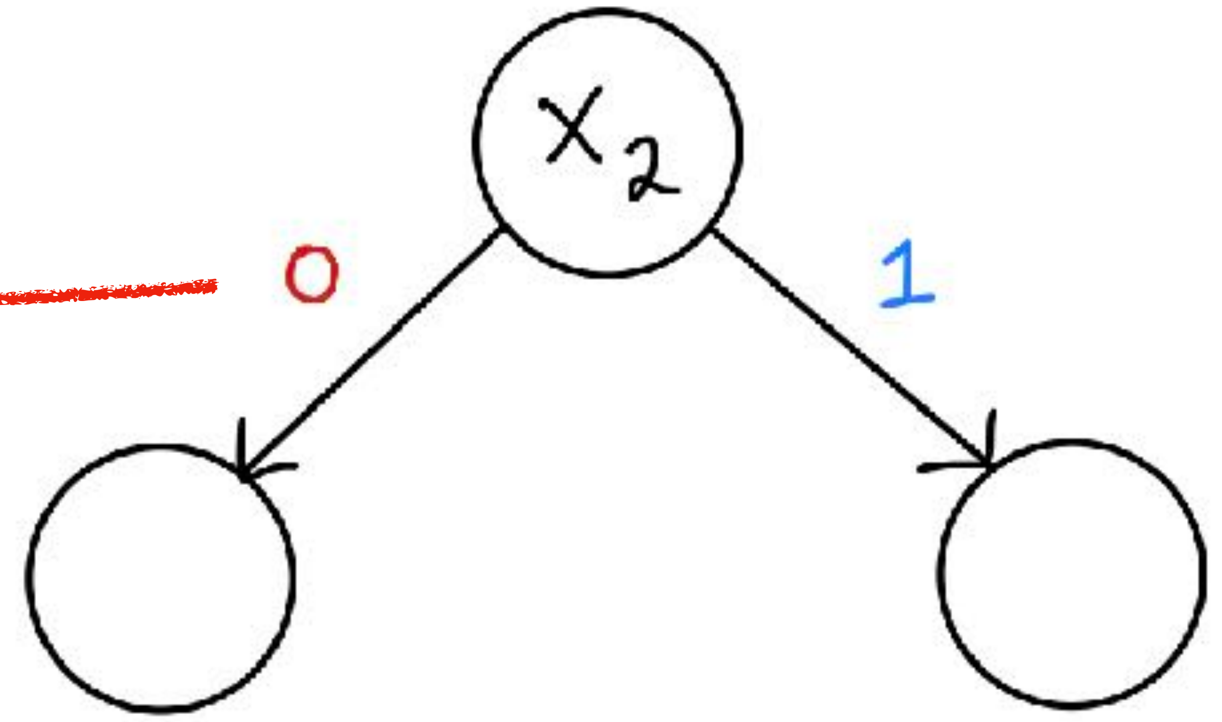
$$F = x_1 \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_2 \vee x_3) \wedge \bar{x}_3$$

Tree-Like Resolution $\equiv$ Decision Trees

Tree-Like Resolution of F



Decision Tree for Search(F)



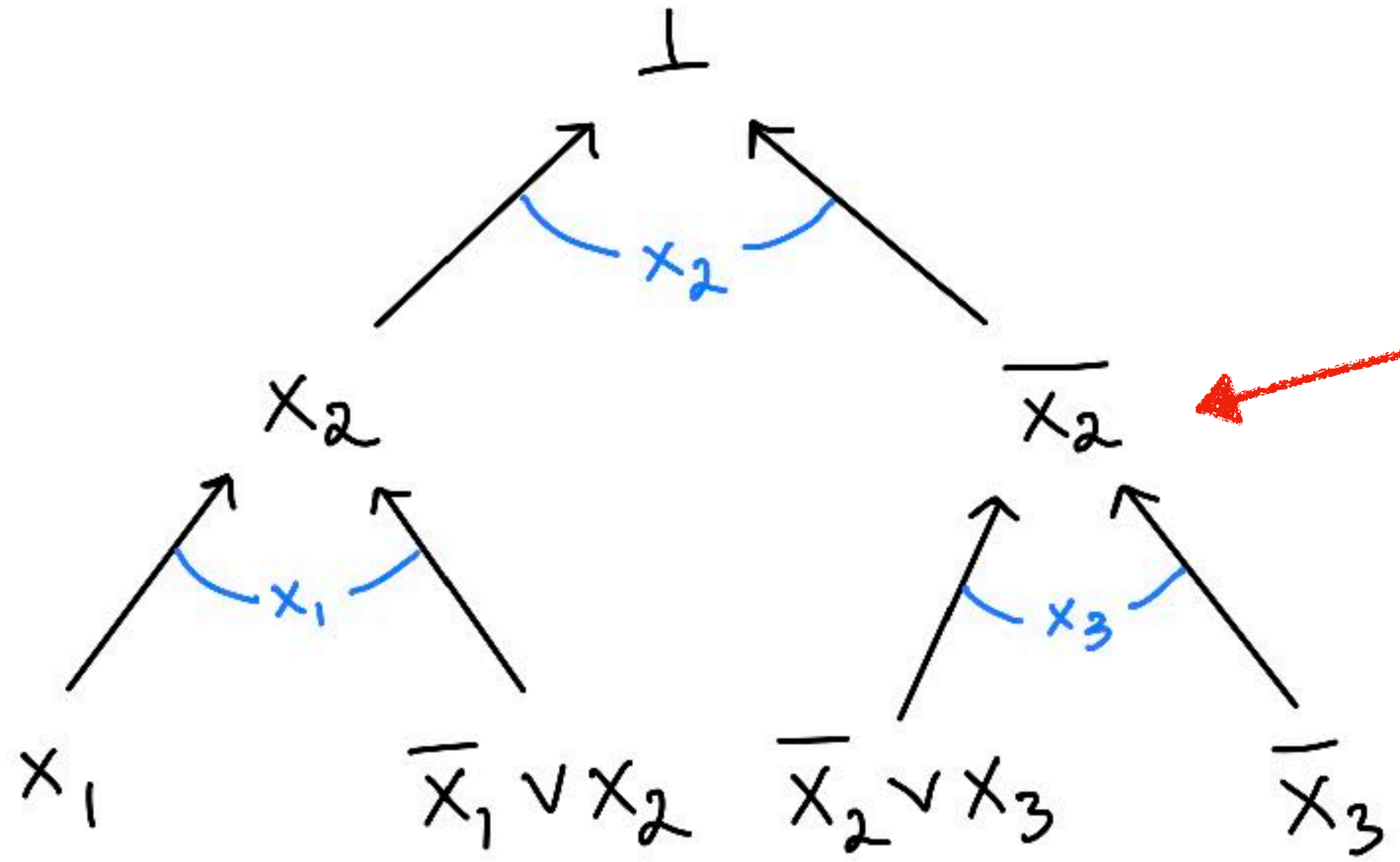
Invariant:

Assignment falsifies clause

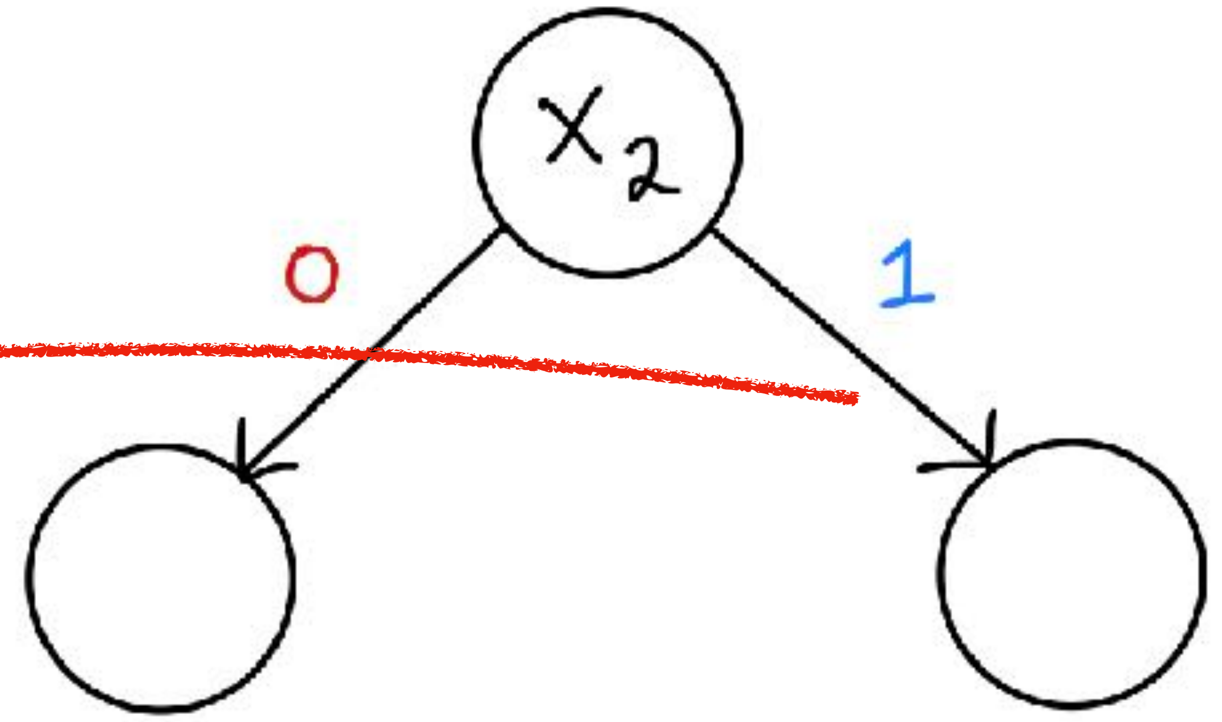
$$F = x_1 \wedge (\overline{x}_1 \vee x_2) \wedge (\overline{x}_2 \vee x_3) \wedge \overline{x}_3$$

Tree-Like Resolution ≡ Decision Trees

Tree-Like Resolution of F



Decision Tree for Search(F)



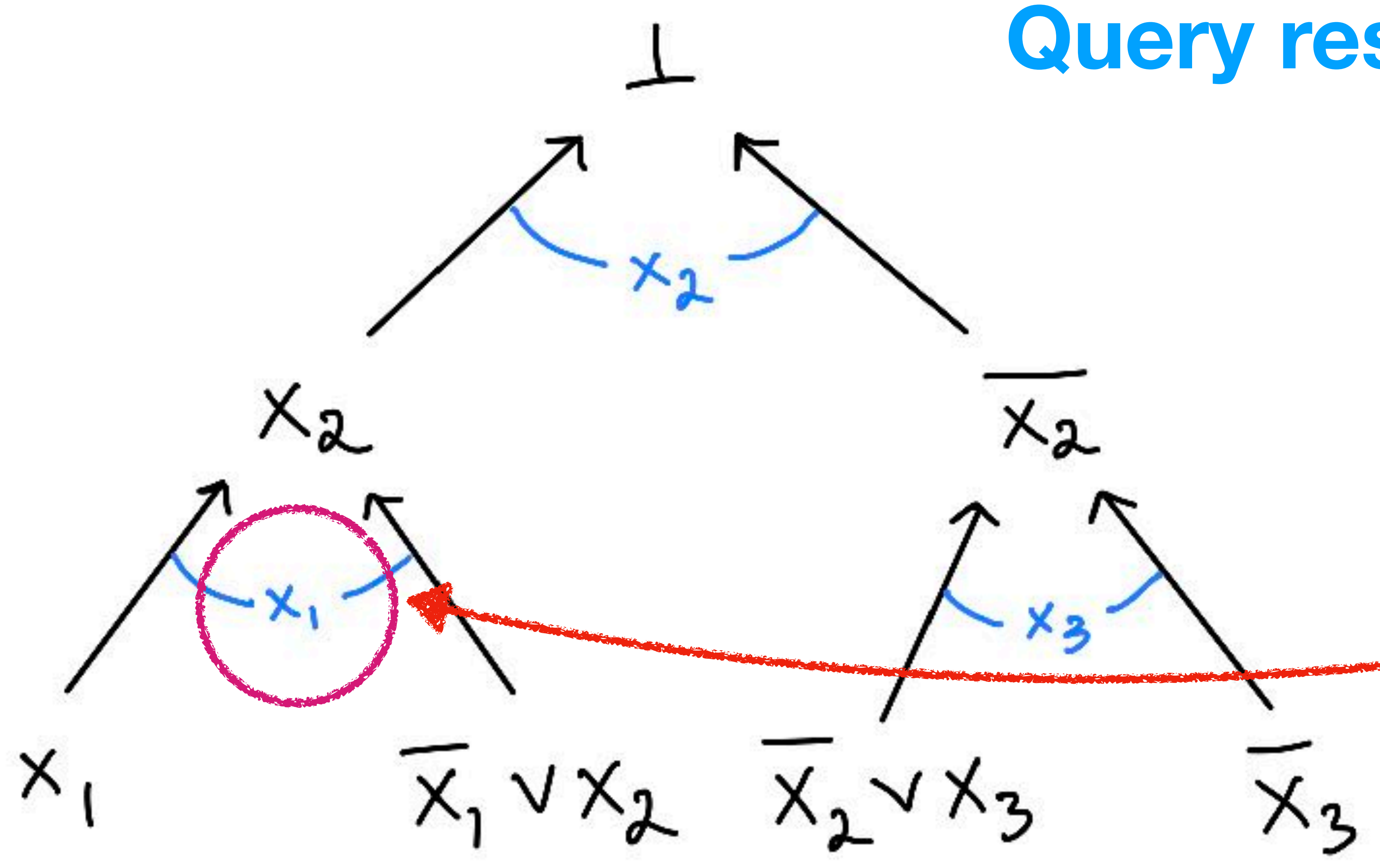
Invariant:

Assignment falsifies clause

$$F = x_1 \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_2 \vee x_3) \wedge \bar{x}_3$$

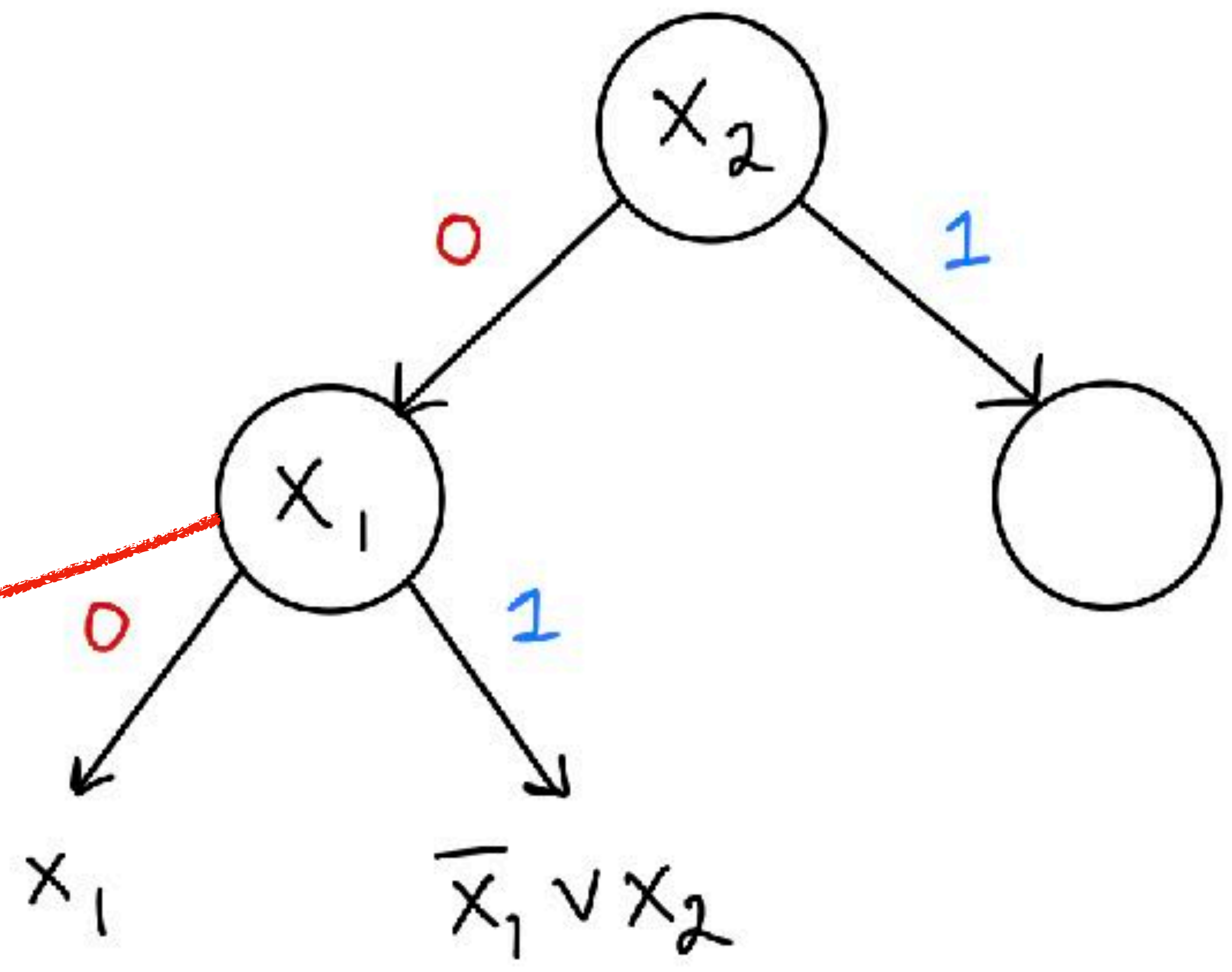
Tree-Like Resolution $\equiv$ Decision Trees

Tree-Like Resolution of F



Query resolved variable

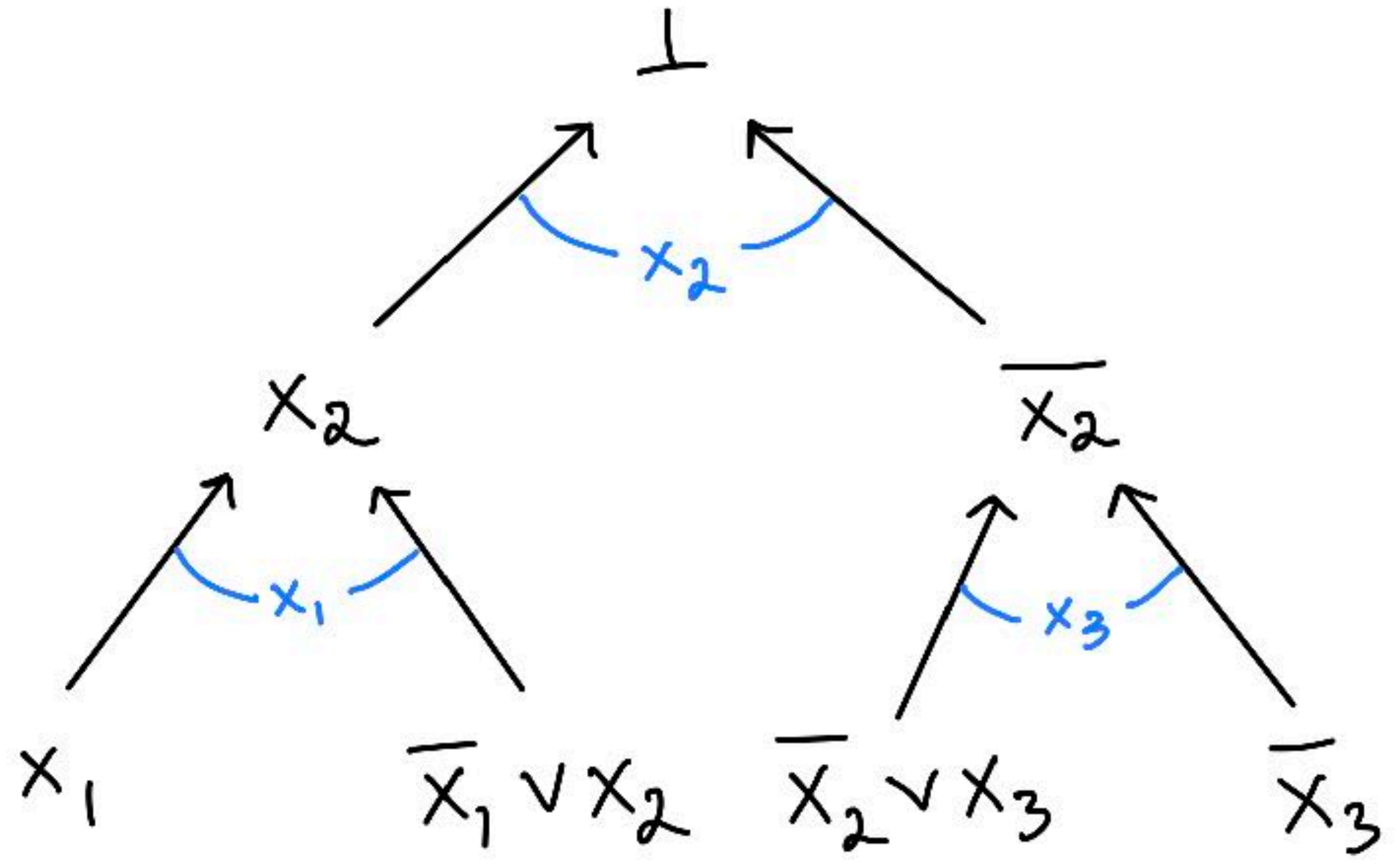
Decision Tree for Search(F)



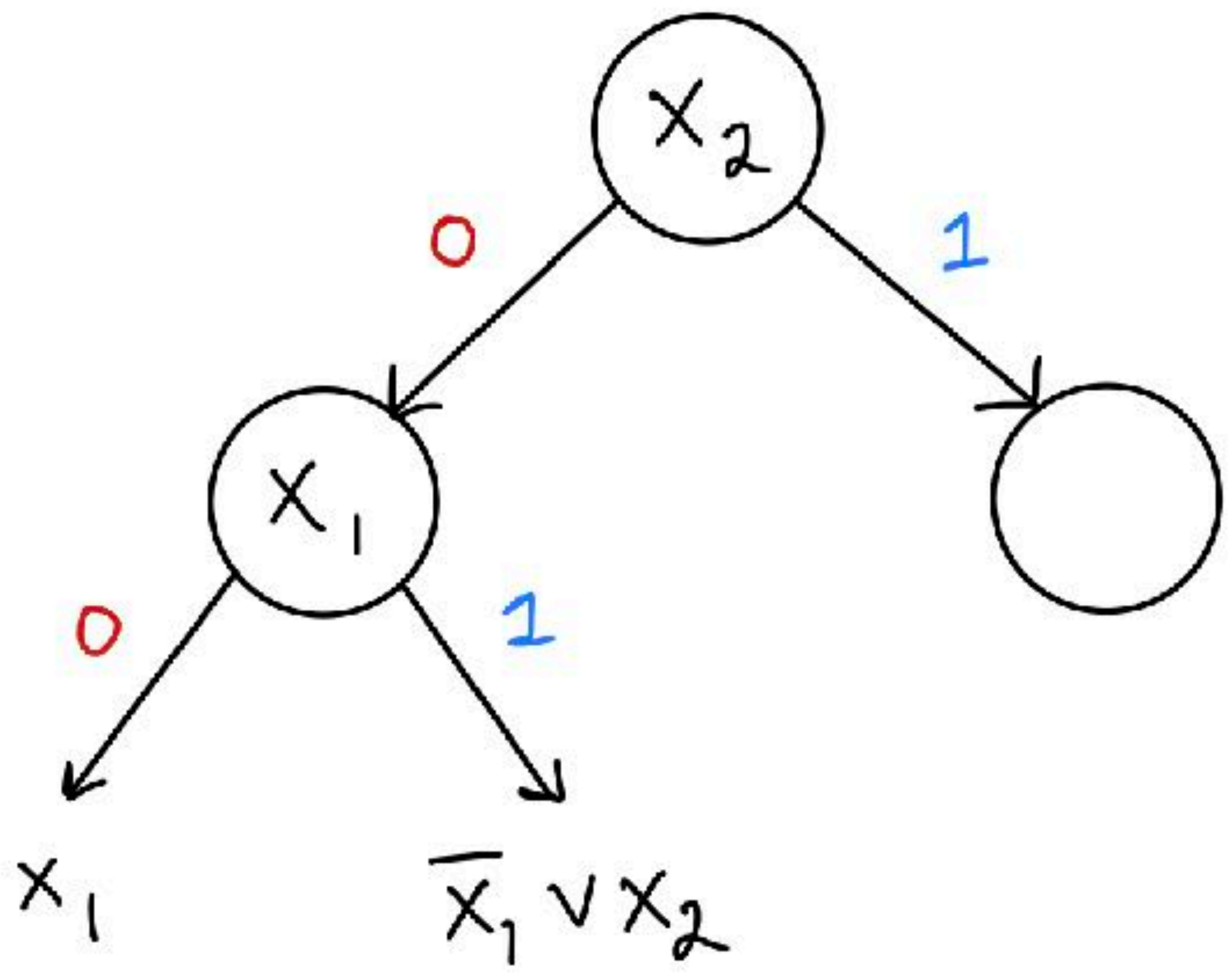
$$F = x_1 \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_2 \vee x_3) \wedge \bar{x}_3$$

Tree-Like Resolution $\equiv$ Decision Trees

Tree-Like Resolution of F



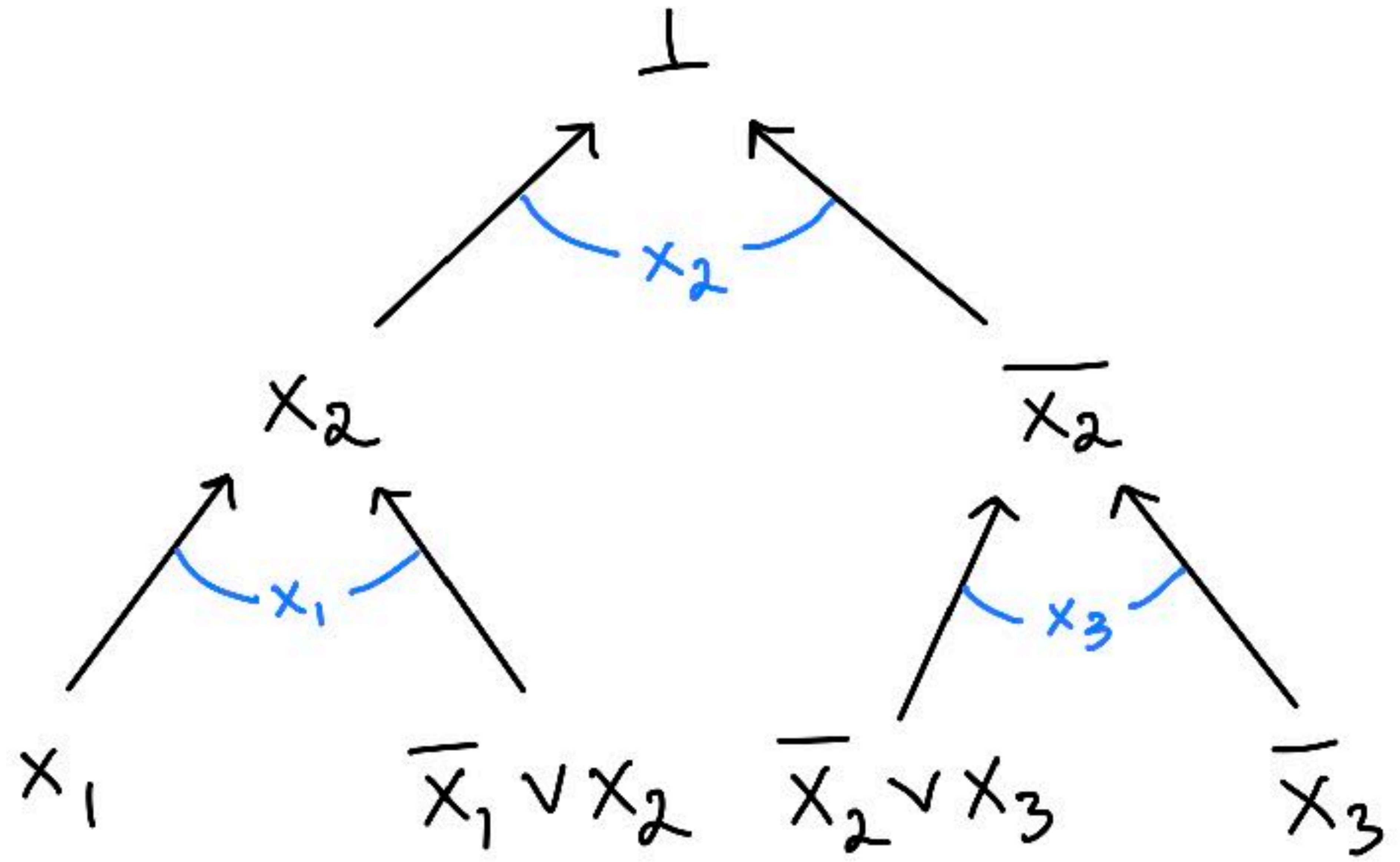
Decision Tree for Search(F)



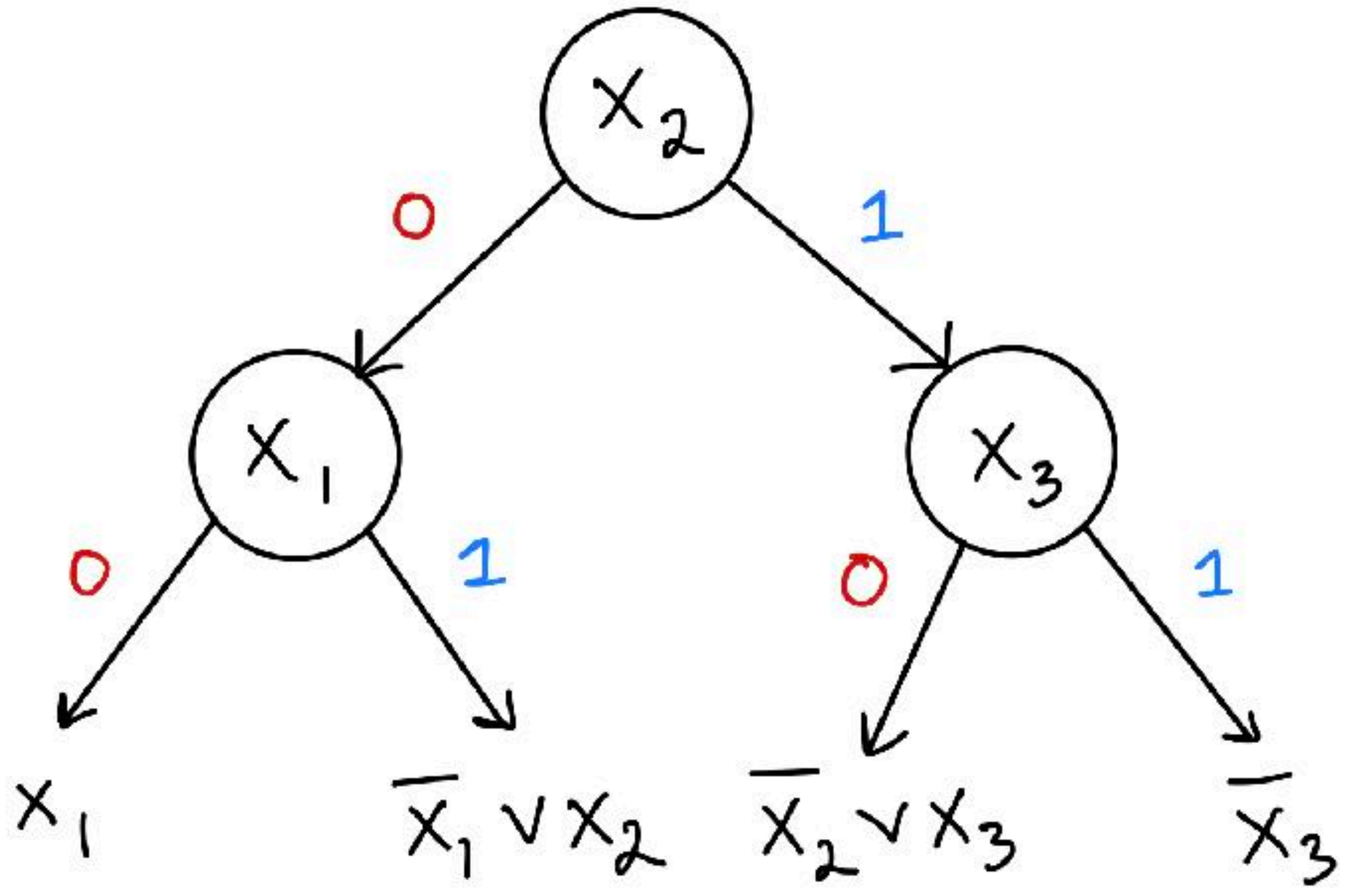
$$F = x_1 \wedge (\overline{x}_1 \vee x_2) \wedge (\overline{x}_2 \vee x_3) \wedge \overline{x}_3$$

Tree-Like Resolution ≡ Decision Trees

Tree-Like Resolution of F



Decision Tree for Search(F)

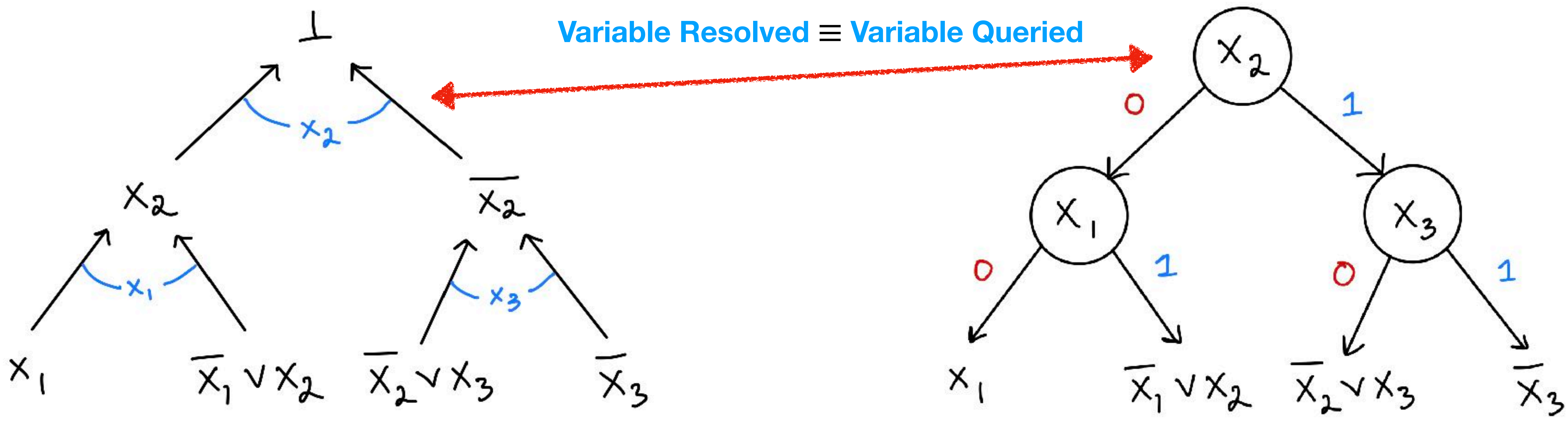


$$F = x_1 \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_2 \vee x_3) \wedge \bar{x}_3$$

Tree-Like Resolution $\equiv$ Decision Trees

Tree-Like Resolution of F

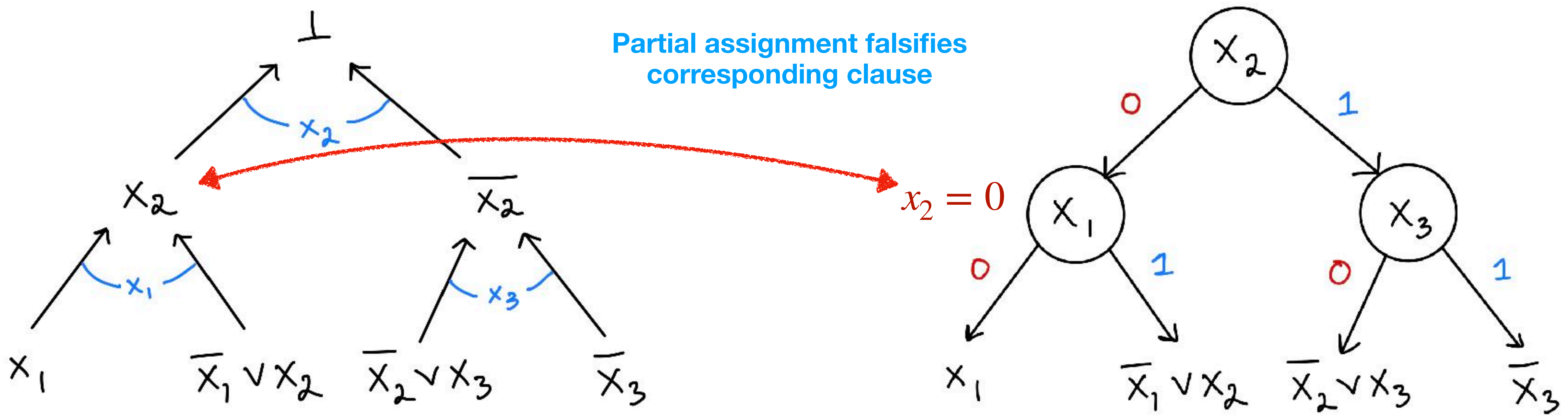
Decision Tree for Search(F)



$$F = x_1 \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_2 \vee x_3) \wedge \bar{x}_3$$

Tree-Like Resolution ≡ Decision Trees

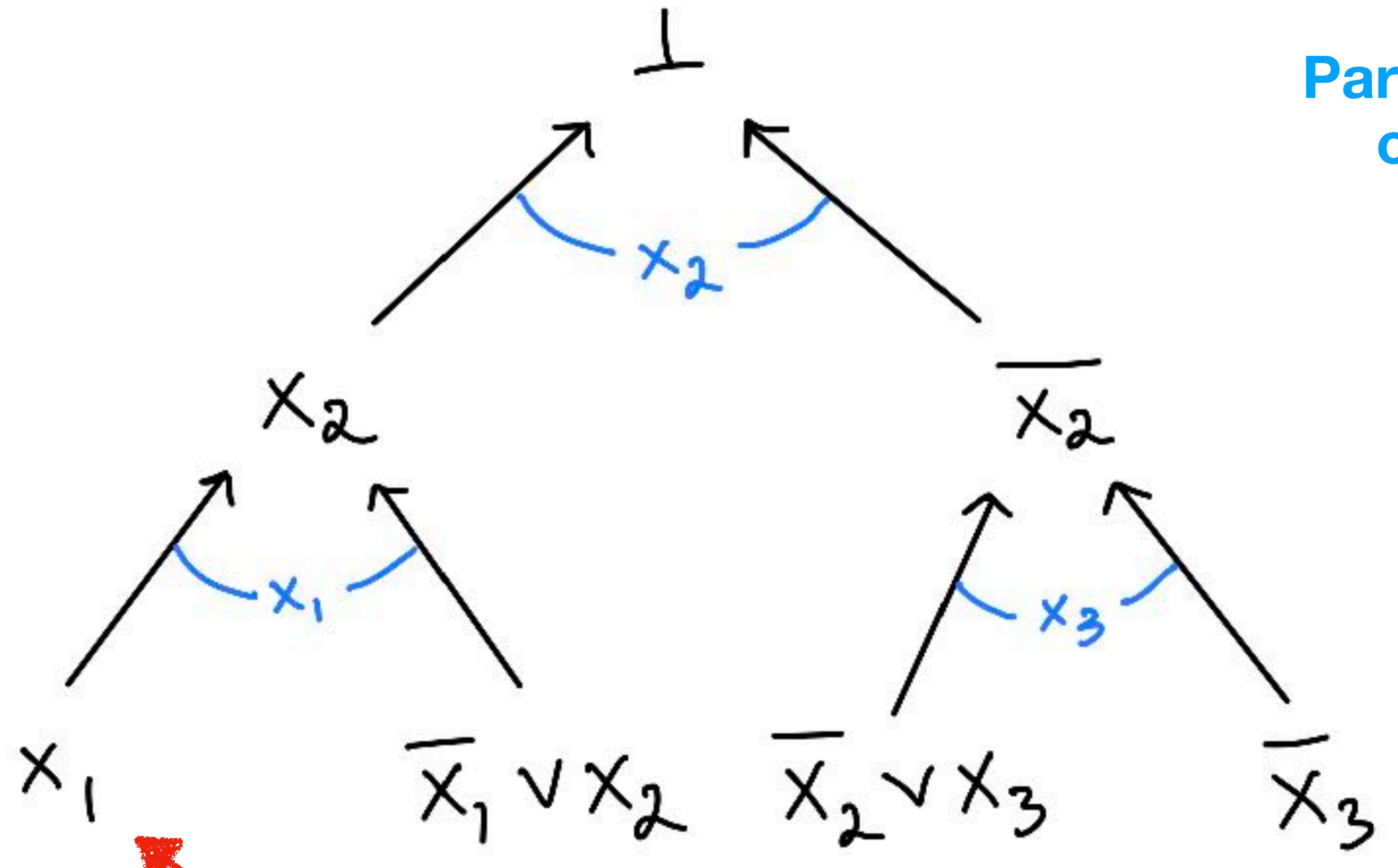
Tree-Like Resolution of F | Decision Tree for Search(F)



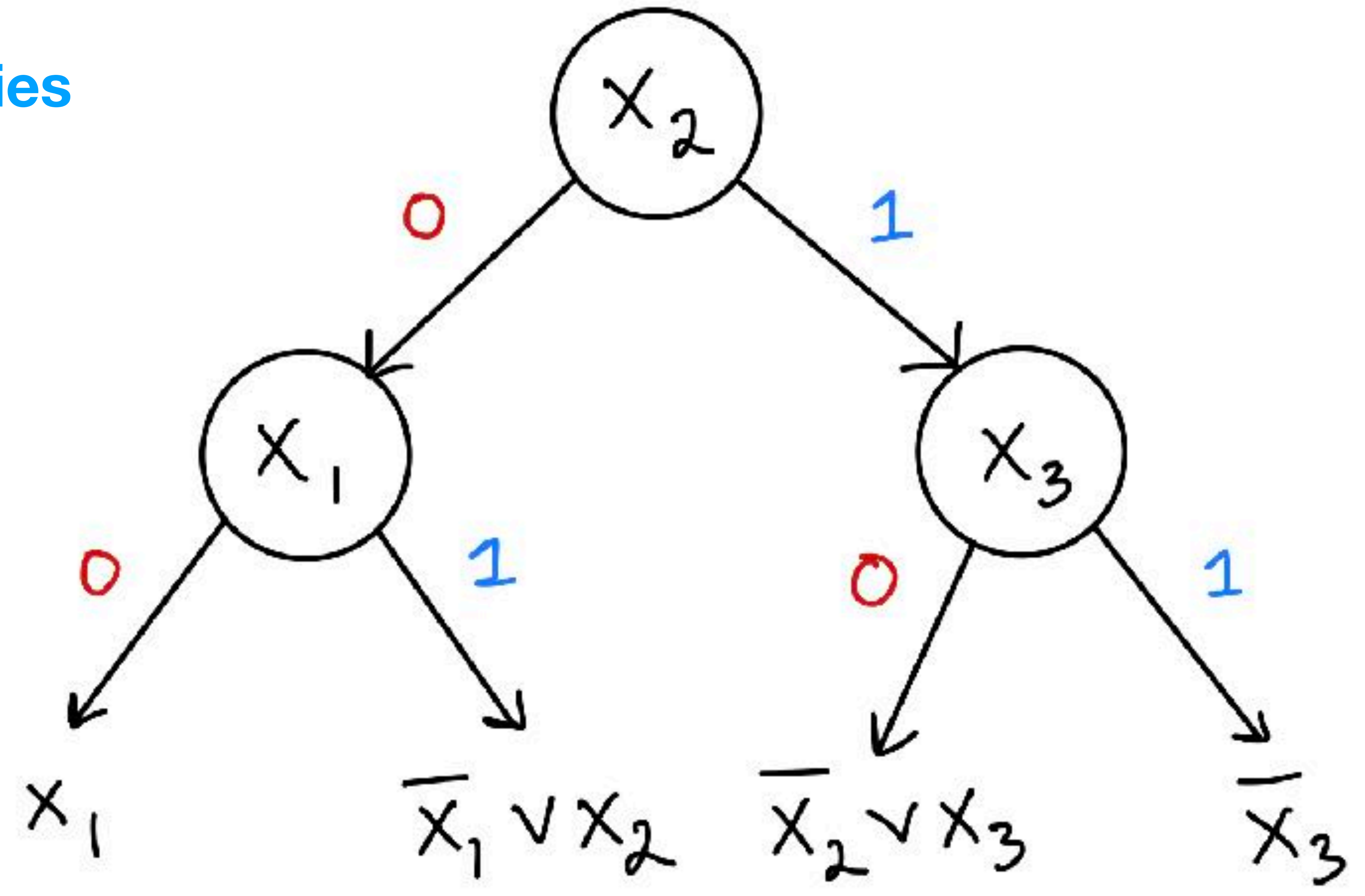
$$F = x_1 \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_2 \vee x_3) \wedge \bar{x}_3$$

Tree-Like Resolution $\equiv$ Decision Trees

Tree-Like Resolution of F



Decision Tree for Search(F)



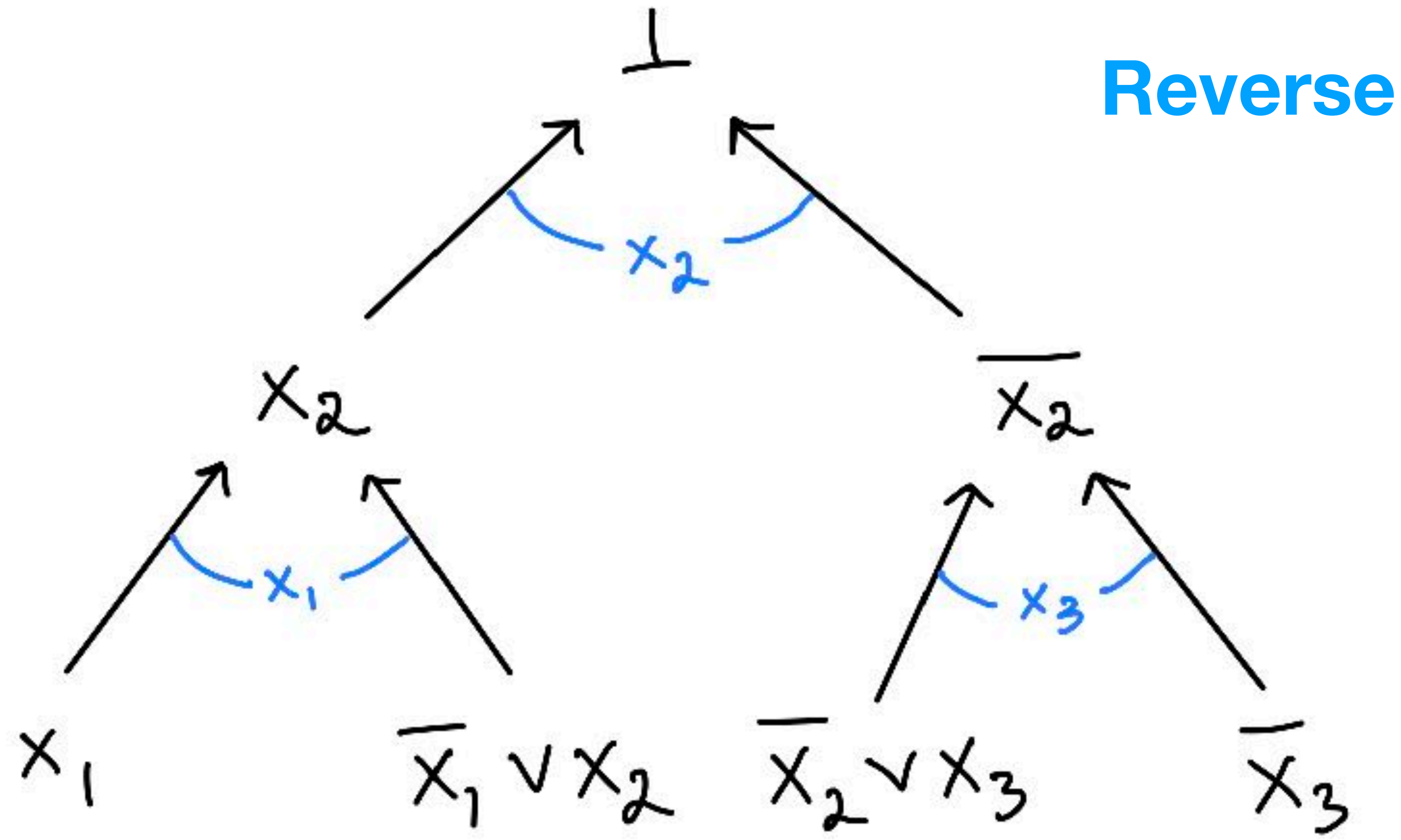
Partial assignment falsifies corresponding clause

$x_1 = 0,\ x_2 = 0$

$$F = x_1 \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_2 \vee x_3) \wedge \bar{x}_3$$

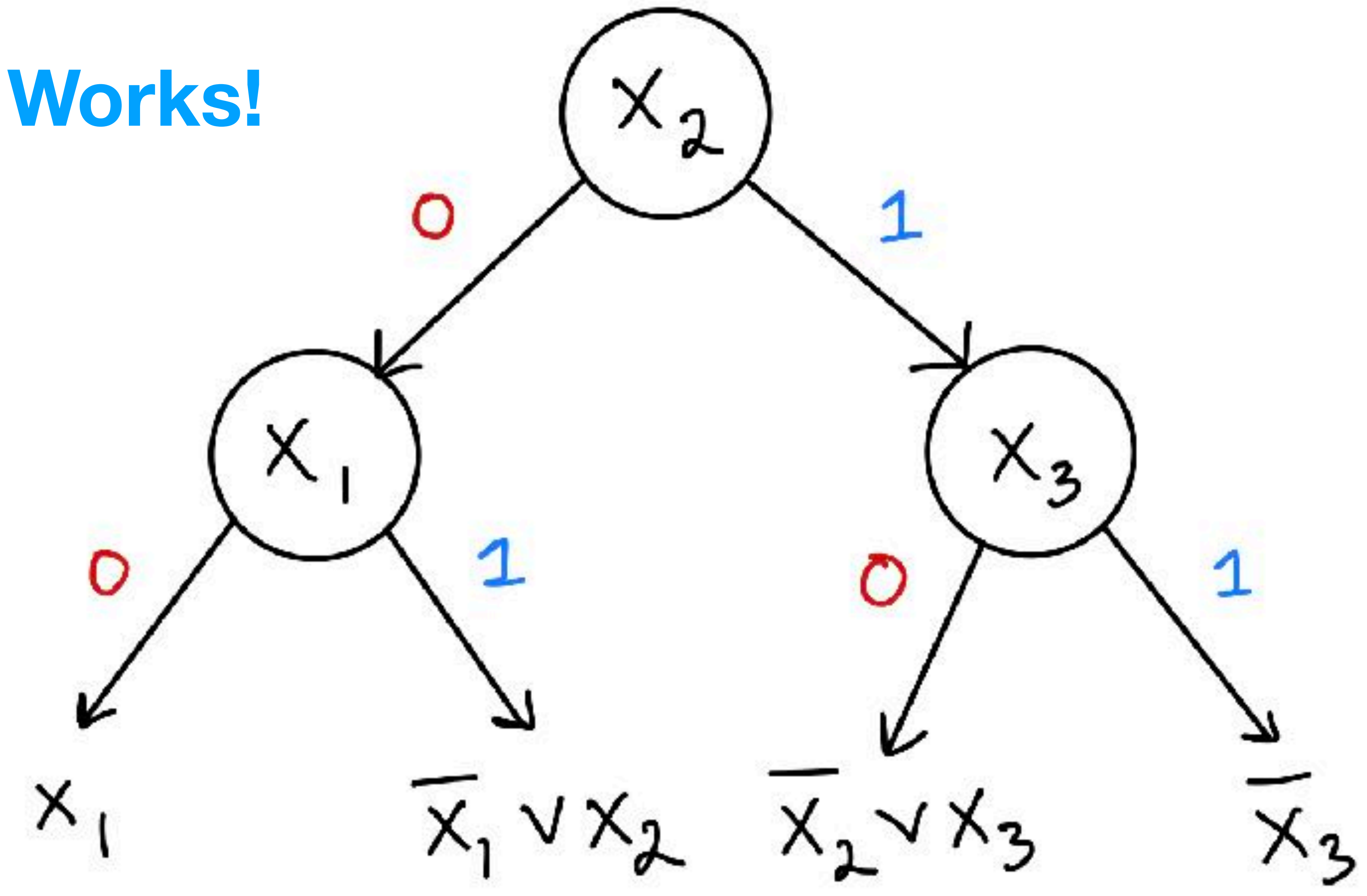
Tree-Like Resolution ≡ Decision Trees

Tree-Like Resolution of F



Reverse Direction Also Works!

Decision Tree for Search(F)



$$F = x_1 \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_2 \vee x_3) \wedge \bar{x}_3$$

Tree-Like Resolution $\equiv$ Decision Trees

Theorem. Let F be an unsatisfiable CNF formula. Then

Size $\leq s$, depth $\leq d$ Tree-like Res. refutation of F

if and only if

Size $\leq s$, depth $\leq d$ Decision Tree for $\mathsf{Search}(F)$

Correspondence is stronger: essentially the same object!

Total Search Problems in Query Complexity

- Let O be finite, $\mathcal{S} \subseteq \{0,1\}^n \times O$.
- $\mathcal{S}(x) := \{o \in O : (x, o) \in \mathcal{S}\}$ is set of **feasible solutions** for x.
- $\mathcal{S}$ is a **total search problem** if $\forall x : \mathcal{S}(x) \neq \emptyset$
- $\mathsf{FP}^{dt}(\mathcal{S}) :=$ **Query Complexity** of $\mathcal{S}$

  $:=$ Depth of shallowest decision tree solving $\mathcal{S}$
- $\mathsf{FP}^{dt} :=$ All total search problems $\mathcal{S}$ such that

$$\mathsf{FP}^{dt}(\mathcal{S}) = \log^{O(1)}(n)$$

Query TFNP

- A **certificate** of $\mathcal{S}$ is a partial restriction $\rho \in \{0,1,*\}^n$ s.t.

$$\exists o \in O, \forall x \in \{0,1\}^n \text{ consistent with } \rho\text{: } o \in \mathcal{S}(x)$$

- A **certificate cover** of $\mathcal{S}$ is a set of certificates R such that every $x \in \{0,1\}^n$ is consistent with some $\rho \in R$.

$$\text{TFNP}^{dt}(\mathcal{S}) := \min_{R \text{ cover}} \max_{\rho \in R} |\text{fixed}(\rho)|$$

- NP **Algorithm**: Given $x \in \{0,1\}^n$,
    - Non-deterministically guess $\rho \in R$,
    - Verify x is consistent by querying fixed coordinates in ρ

Query TFNP

- A **certificate** of $\mathcal{S}$ is a partial restriction $\rho \in \{0,1,*\}^n$ s.t.

$$\exists o \in O, \forall x \in \{0,1\}^n \text{ consistent with } \rho\text{: } o \in \mathcal{S}(x)$$

- A **certificate cover** of $\mathcal{S}$ is a set of certificates R such that every $x \in \{0,1\}^n$ is consistent with some $\rho \in R$.

$$\mathrm{TFNP}^{dt}(\mathcal{S}) := \min_{R \text{ cover}} \max_{\rho \in R} |\mathrm{fixed}(\rho)|$$

- TFNP^{dt} := all total search problems $\mathcal{S}$ with

$$\mathrm{TFNP}^{dt}(\mathcal{S}) = \log^{O(1)} n$$

What's So Special About $\mathsf{Search}(F)$?

- For unsat. $F = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, clauses **are** certificates:

$$\therefore \text{TFNP}^{dt}(\mathsf{Search}(F)) \leq \mathsf{width}(F)$$

- Any total $\mathcal{S}$ can be **reduced** to solving $\mathsf{Search}(F)$ for some F:
  - Given $\mathcal{S}$ and certificate cover R, define $F_R := \wedge_{\rho \in R} C_{\overline{\rho}}$ where $C_{\overline{\rho}}$ is the maximum-width clause falsified by ρ.
  - Solving $\mathsf{Search}(F_R)$ gives a certificate for $\mathcal{S}$ which then lets us solve $\mathcal{S}$. Converse also holds, under reasonable assumptions.

Summary

- Any unsatisfiable CNF F has an associated $\mathsf{Search}(F)$
- Decision trees for $\mathsf{Search}(F) \equiv$ Tree-Res. refutations of F
- $\mathsf{Search}(F)$ is complete* for TFNP^{dt}

Can we capture **other** proof systems?

Part 2

Circuits and the Karchmer-Wigderson Game

Karchmer-Wigderson Games

- Focus on complexity of boolean functions $f : \{0,1\}^n \to \{0,1\}$
  - f **monotone** if $x \leq y$ (coordinate-wise) implies $f(x) \leq f(y)$
  - f is **partial** if $f : \{0,1\}^n \to \{0,1,*\}$, $*$ means we "don't care".
- f has an associated **total search problem** [KW 90]

$$\mathrm{KW}(f) \subseteq f^{-1}(1) \times f^{-1}(0) \times [n]$$

Given $x \in f^{-1}(1), y \in f^{-1}(0)$*, find* $i \in [n]$ *such that* $x_i \neq y_i$

- **Circuit Complexity** of f $\equiv$ **Communication Complexity** of $\mathrm{KW}(f)$

Karchmer-Wigderson Games

- Focus on complexity of boolean functions $f: \{0,1\}^n \to \{0,1\}$
  - f **monotone** if $x \leq y$ (coordinate-wise) implies $f(x) \leq f(y)$
  - f is **partial** if $f: \{0,1\}^n \to \{0,1,*\}$, $*$ means we "don't care".
- f has an associated **total search problem** [KW 90]
- If f is **monotone**, then there is a more restricted game:

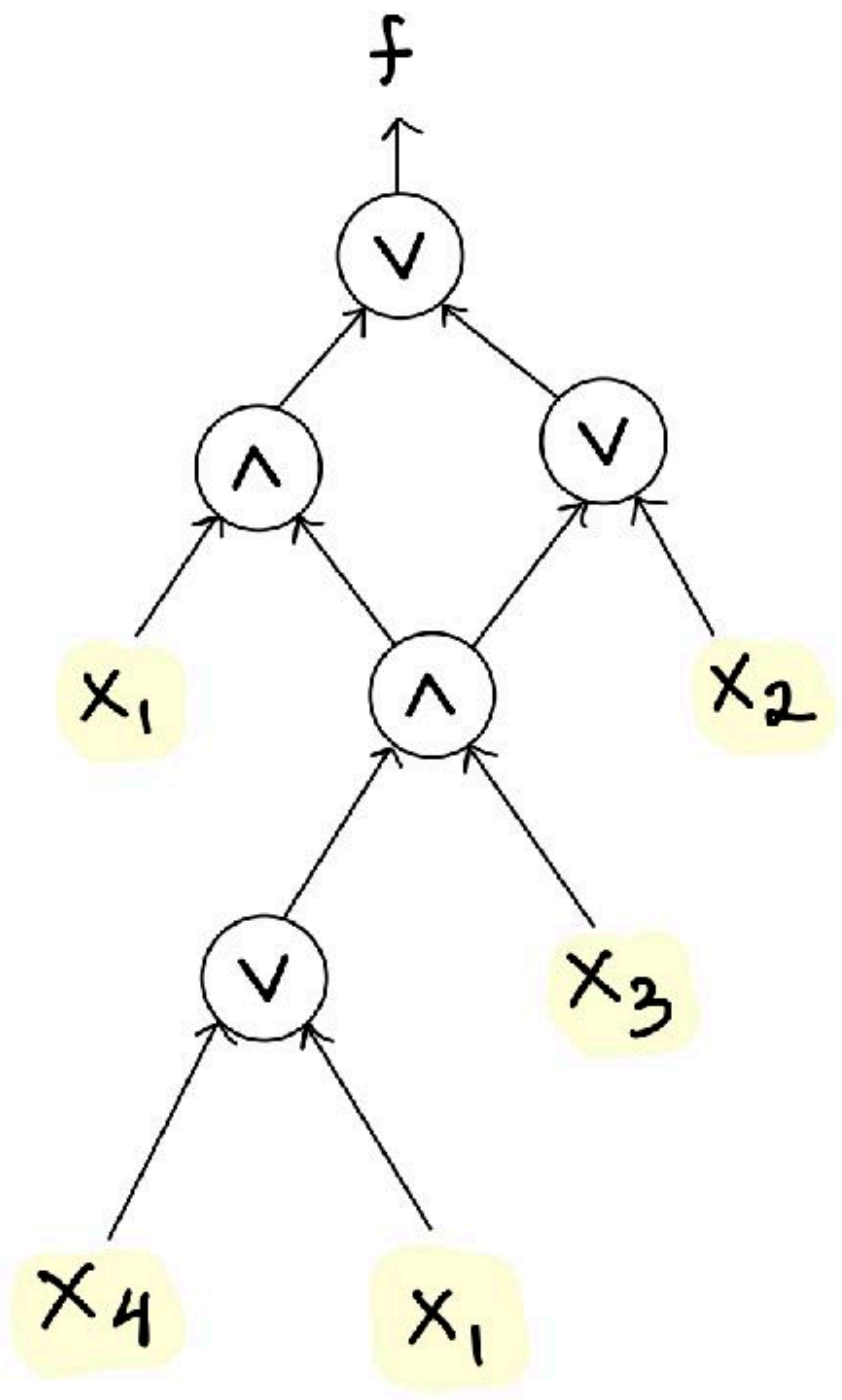
$$\text{mKW}(f) \subseteq f^{-1}(1) \times f^{-1}(0) \times [n]$$

Given $x \in f^{-1}(1), y \in f^{-1}(0)$, *find* $i \in [n]$ *such that* $x_i > y_i$

- **Circuit Complexity** of f $\equiv$ **Communication Complexity** of $\text{KW}(f)$

Boolean Circuits

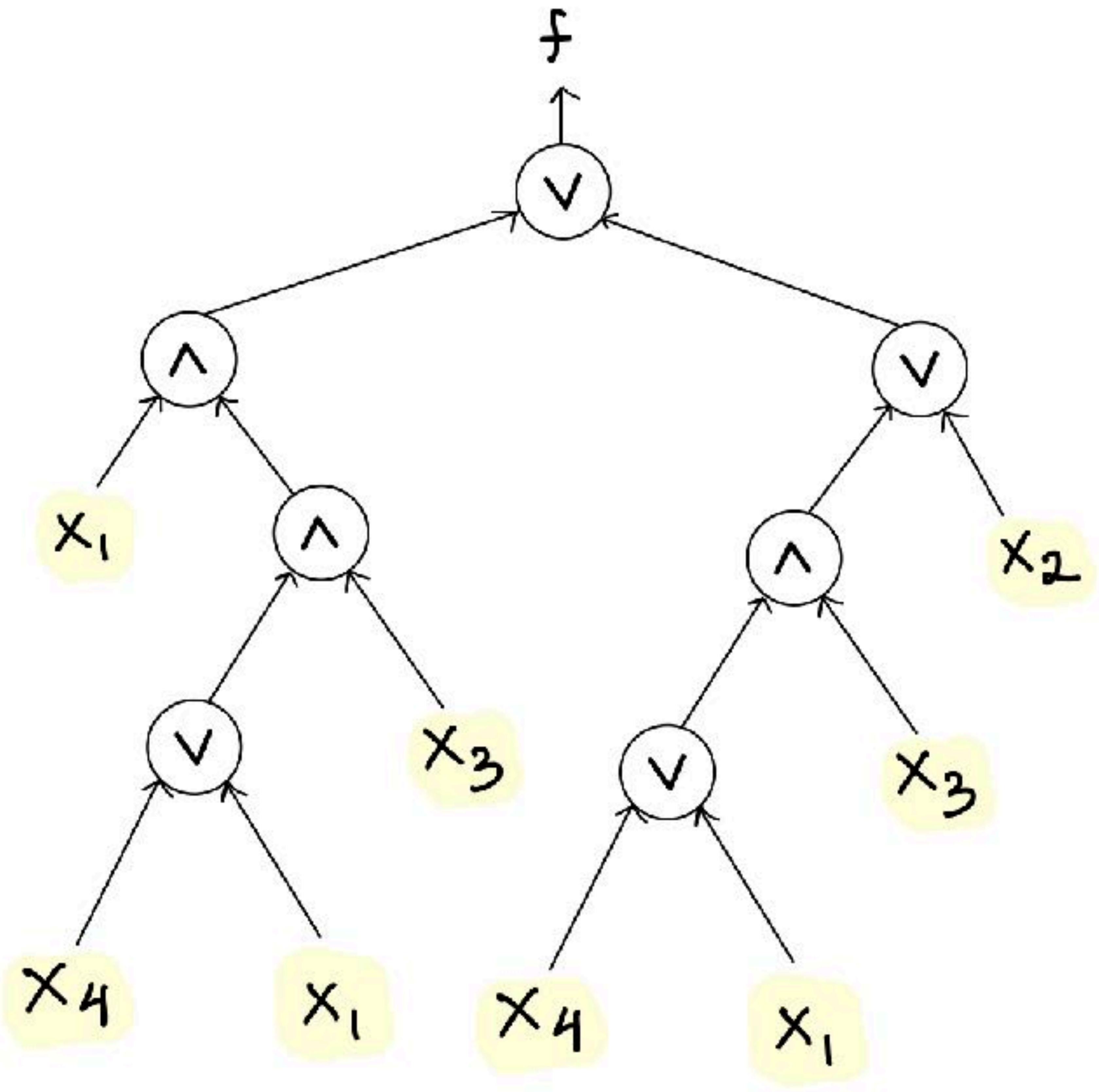
- Device to compute boolean functions
- Starting* from boolean literals, use $\wedge$ and $\vee$ **gates** to compute a target function.
- **Size** := Number of gates
- **Depth** := Length of longest root-leaf path
- Circuit is a **formula** if no gate re-used.



* These are technically **DeMorgan circuits**, but are polynomially equivalent to standard boolean circuits.

Boolean Circuits

- Device to compute boolean functions
- Starting* from boolean literals, use ∧ and ∨ **gates** to compute a target function.
- **Size** := Number of gates
- **Depth** := Length of longest root-leaf path
- Circuit is a **formula** if no gate re-used.



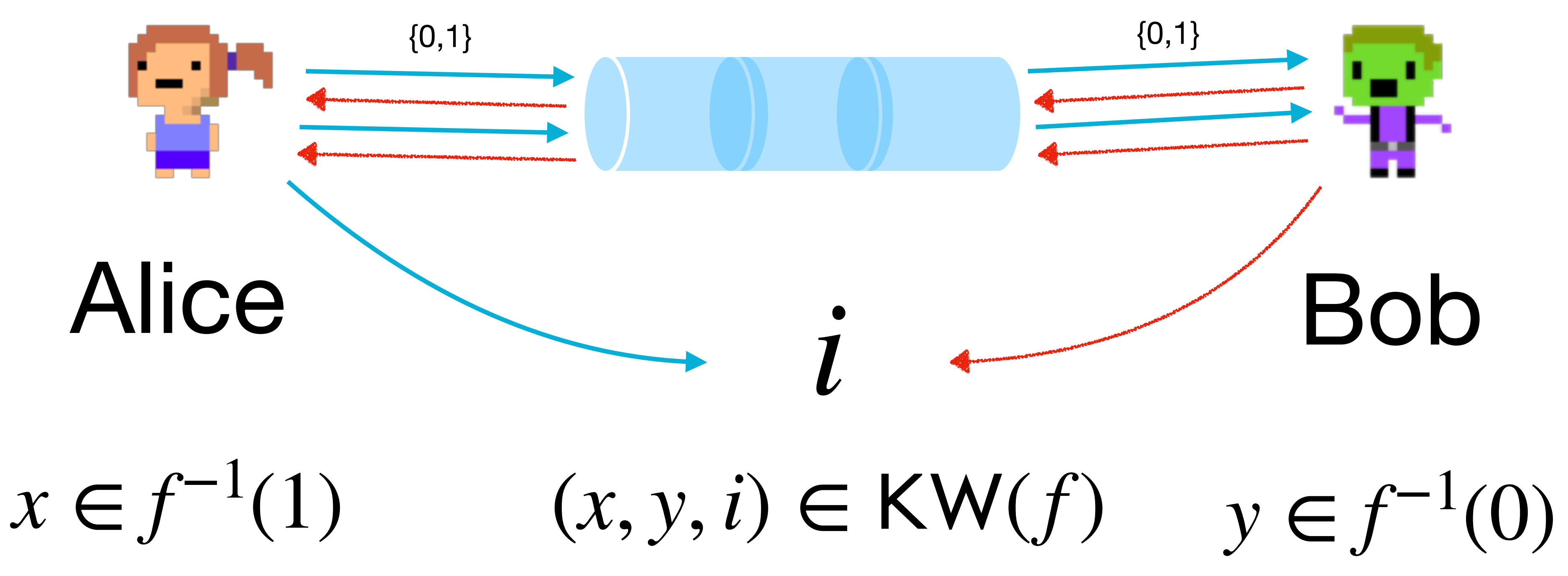
* These are technically **DeMorgan circuits**, but are polynomially equivalent to standard boolean circuits.

Communication Protocols for $\mathsf{KW}(f)$

- Two players, Alice and Bob
- A. $\leftarrow x \in f^{-1}(1)$, B. $\leftarrow y \in f^{-1}(0)$
- Communicate by sending bits over a channel, goal is to find $i : x_i \neq y_i$
- **Protocol:** Tree telling Alice and Bob who speaks at each point.

$$\mathsf{KW}(f) \subseteq f^{-1}(1) \times f^{-1}(0) \times [n]$$

Given $x \in f^{-1}(1), y \in f^{-1}(0)$, *find* $i \in [n]$ *such that* $x_i \neq y_i$

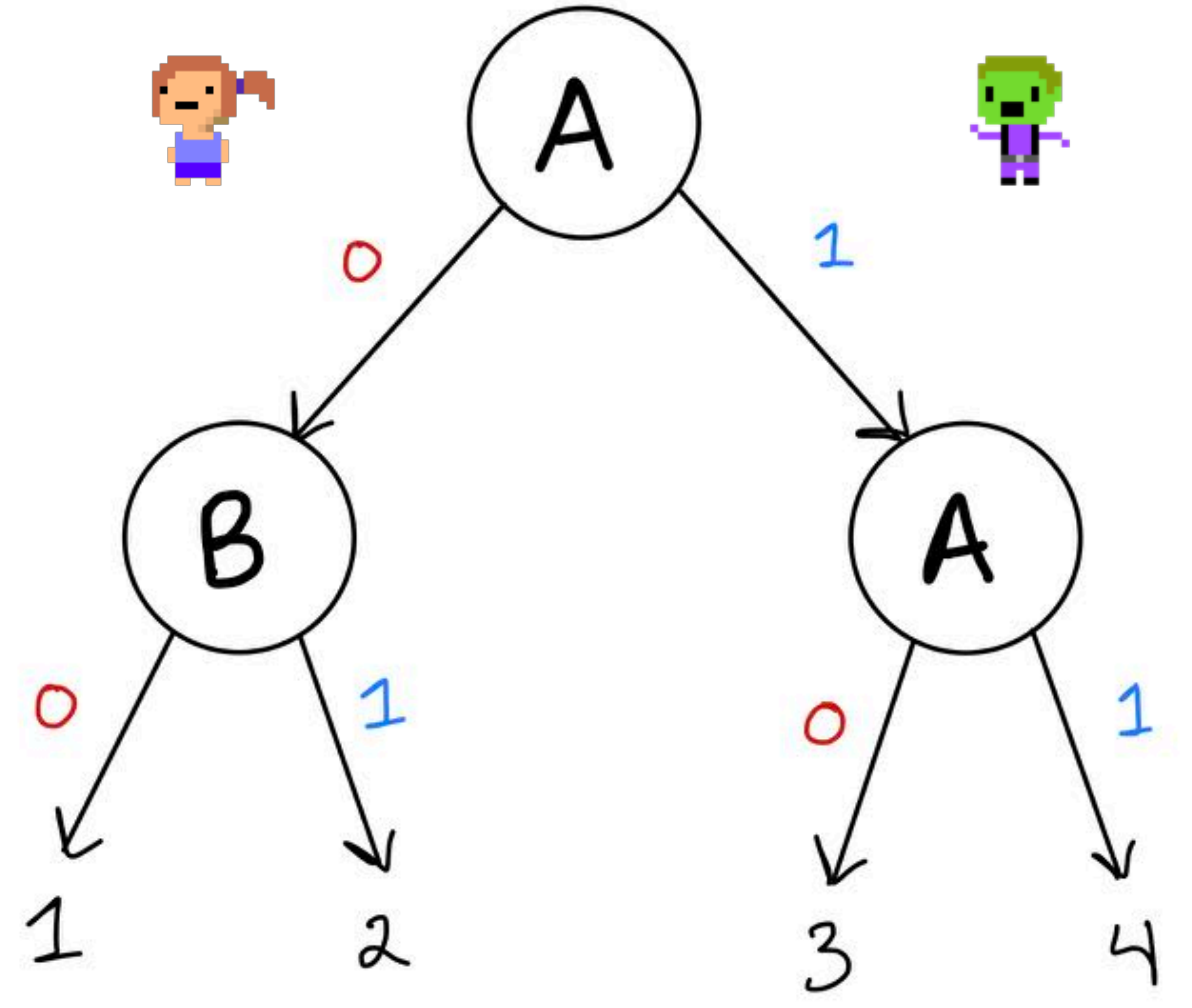


Communication Protocols for $\mathsf{KW}(f)$

- Two players, Alice and Bob
- A. $\leftarrow x \in f^{-1}(1)$, B. $\leftarrow y \in f^{-1}(0)$
- Communicate by sending bits over a channel, goal is to find $i : x_i \neq y_i$
- **Protocol:** Tree telling Alice and Bob who speaks at each point.
- **Depth** $:=$ Length of longest path
- **Size** $:=$ Number of nodes in tree

$\mathsf{KW}(f) \subseteq f^{-1}(1) \times f^{-1}(0) \times [n]$

Given $x \in f^{-1}(1), y \in f^{-1}(0)$, *find* $i \in [n]$ *such that* $x_i \neq y_i$



$f(x_1, x_2, x_3, x_4)$

Combinatorial Rectangles

- A **combinatorial rectangle** in $U \times V$ is a set of the form

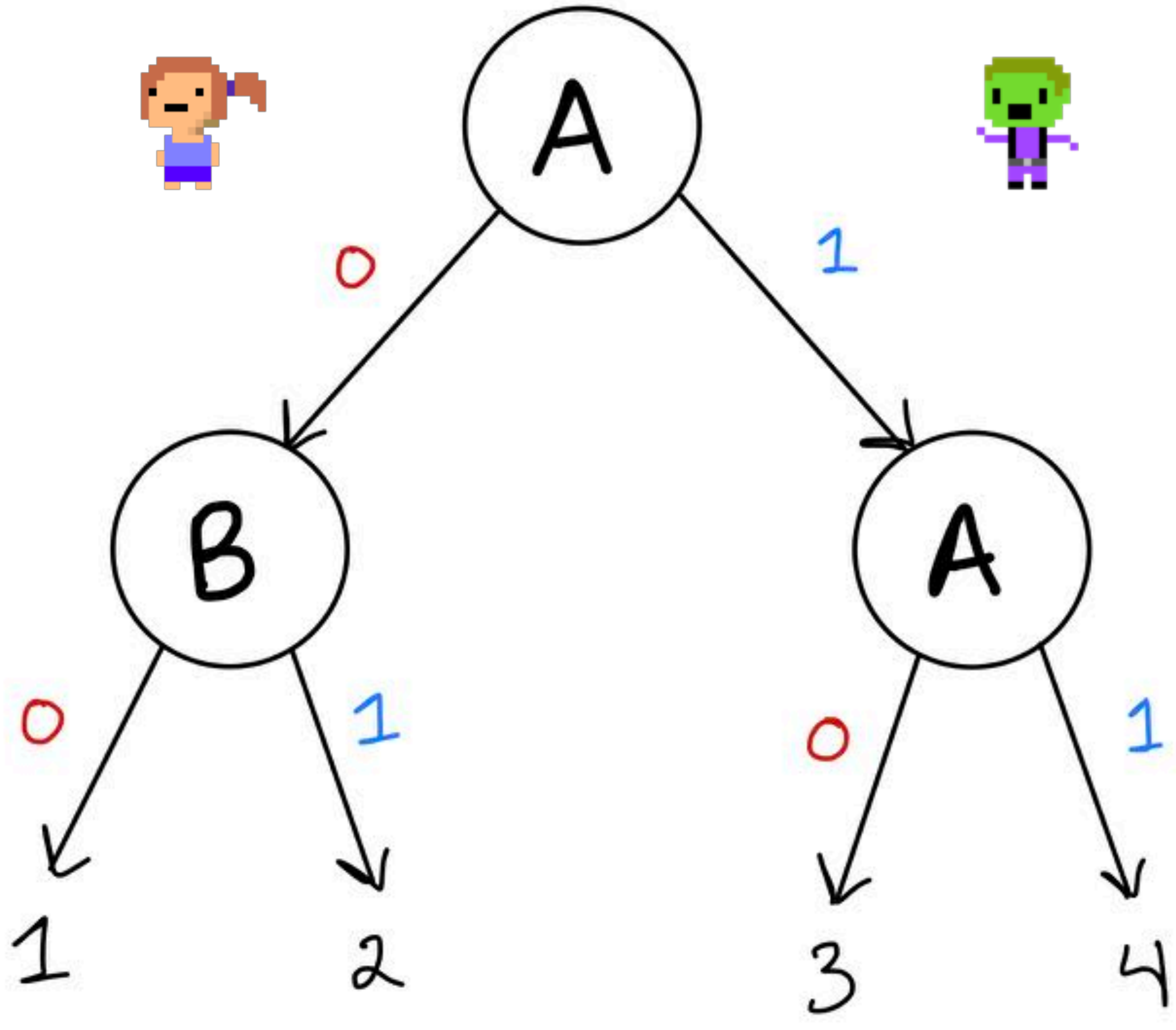
$$A \times B \subseteq U \times V$$

for $A \subseteq U$, $B \subseteq V$.

- The set of inputs reaching a node in a protocol is a combinatorial rectangle!

$\mathsf{KW}(f) \subseteq f^{-1}(1) \times f^{-1}(0) \times [n]$

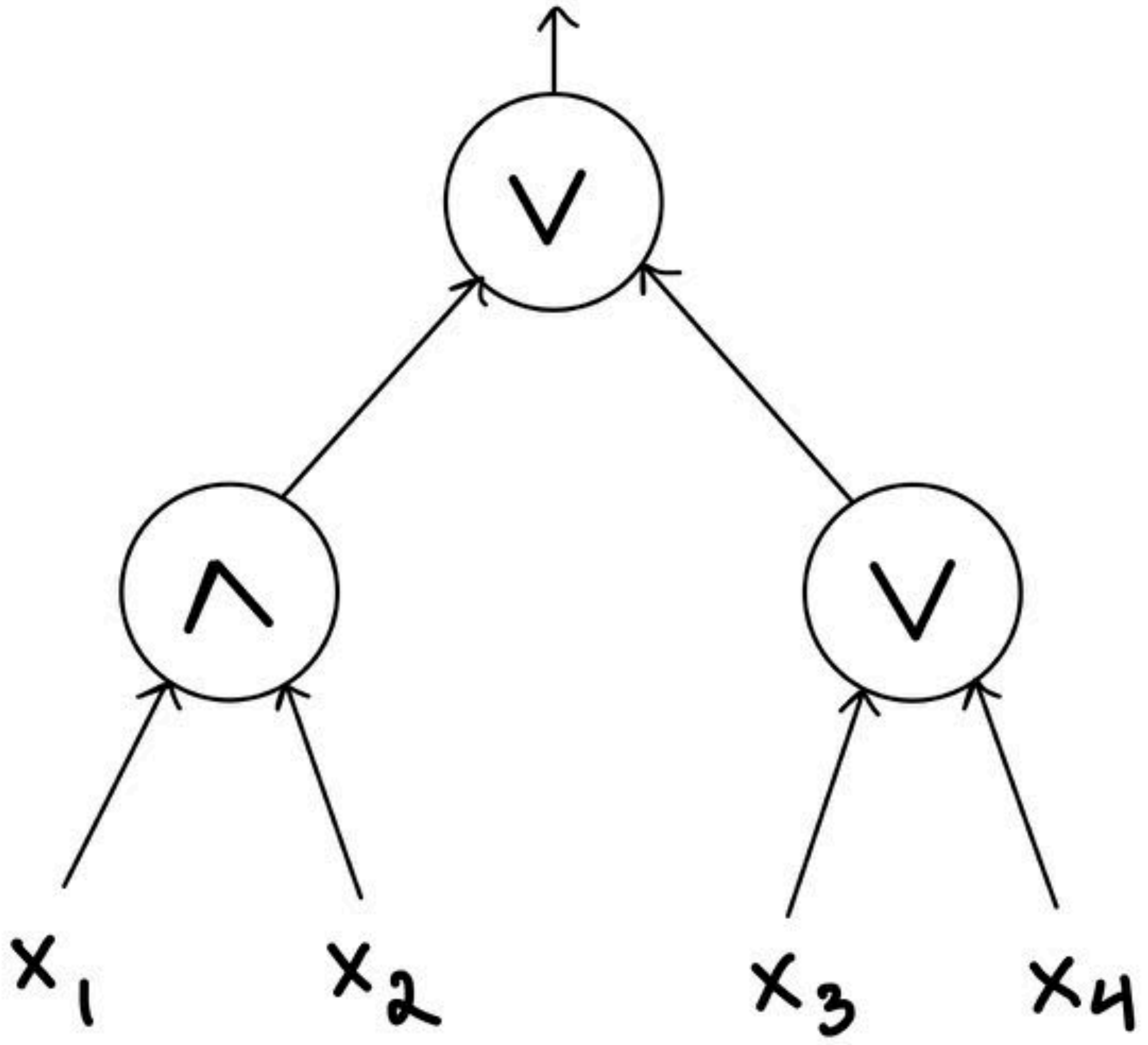
Given $x \in f^{-1}(1), y \in f^{-1}(0)$, *find* $i \in [n]$ *such that* $x_i \neq y_i$



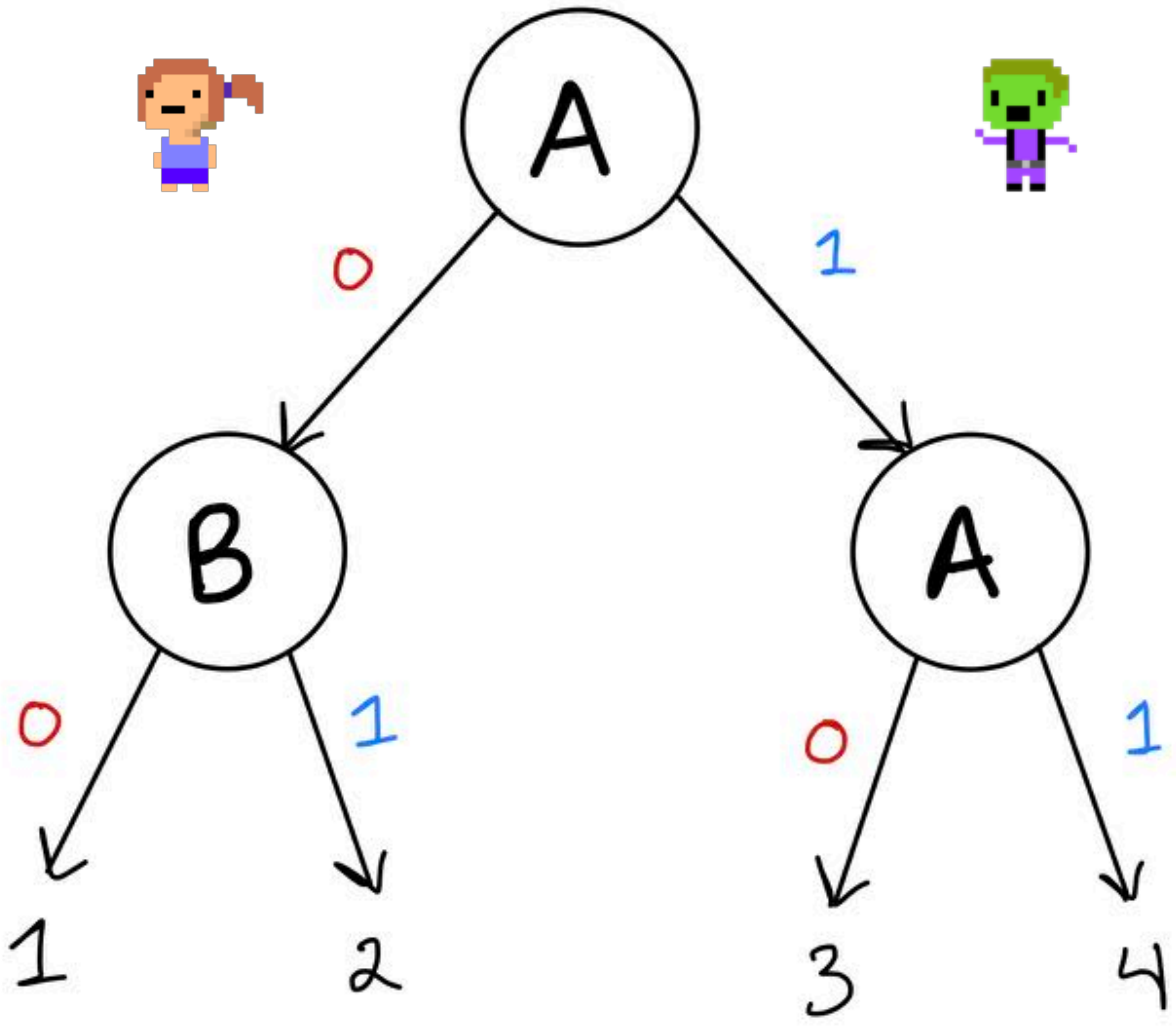
$f(x_1, x_2, x_3, x_4)$

Formulas $\equiv$ Communication

Boolean Formula for f



Protocol for $\mathrm{KW}(f)$



$$f = (x_1 \wedge x_2) \vee x_3 \vee x_4$$

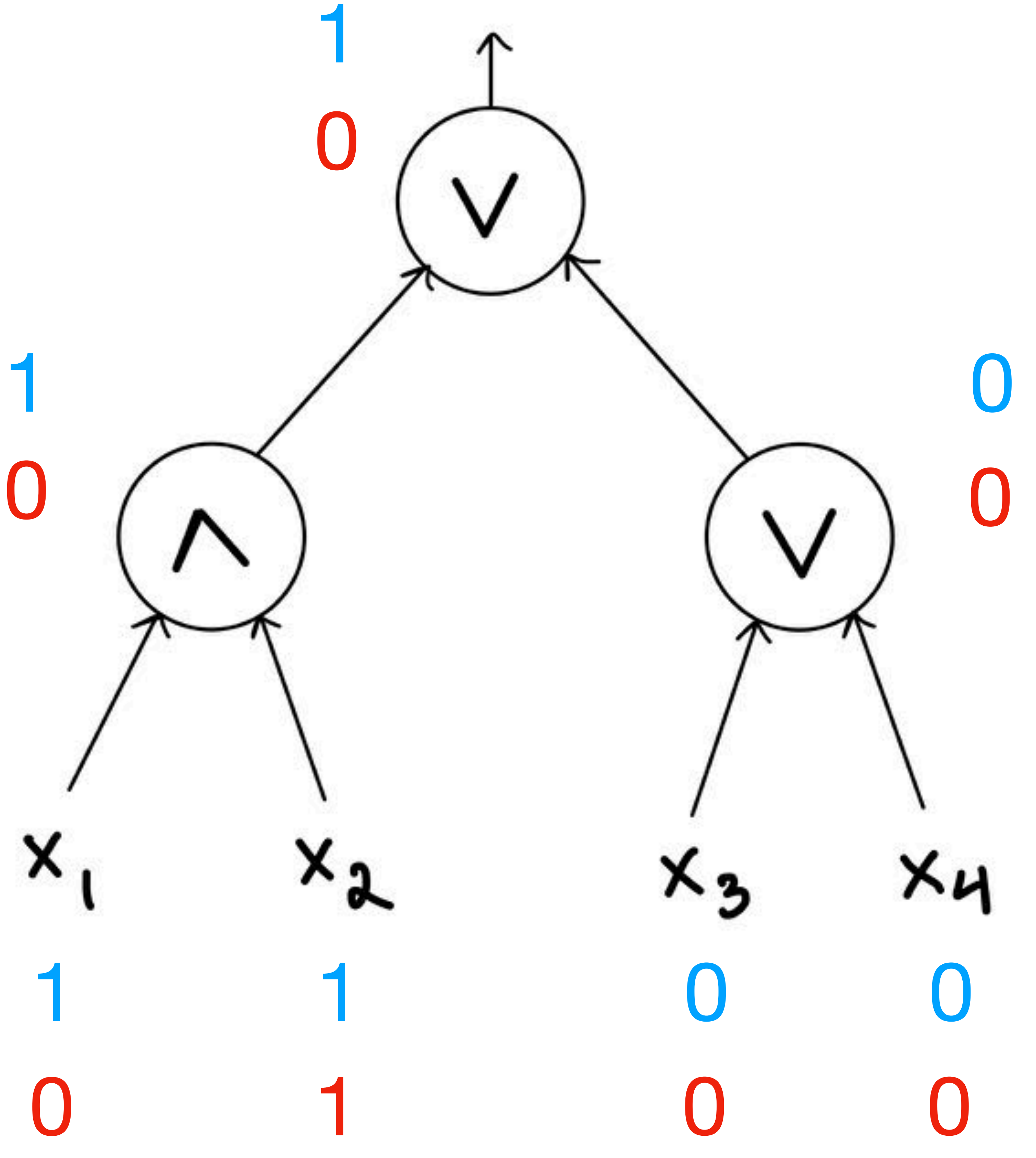
Formulas ≡ Communication


$x = (1,1,0,0)$


$y = (0,1,0,0)$

Boolean Formula for f

Protocol for $\mathrm{KW}(f)$



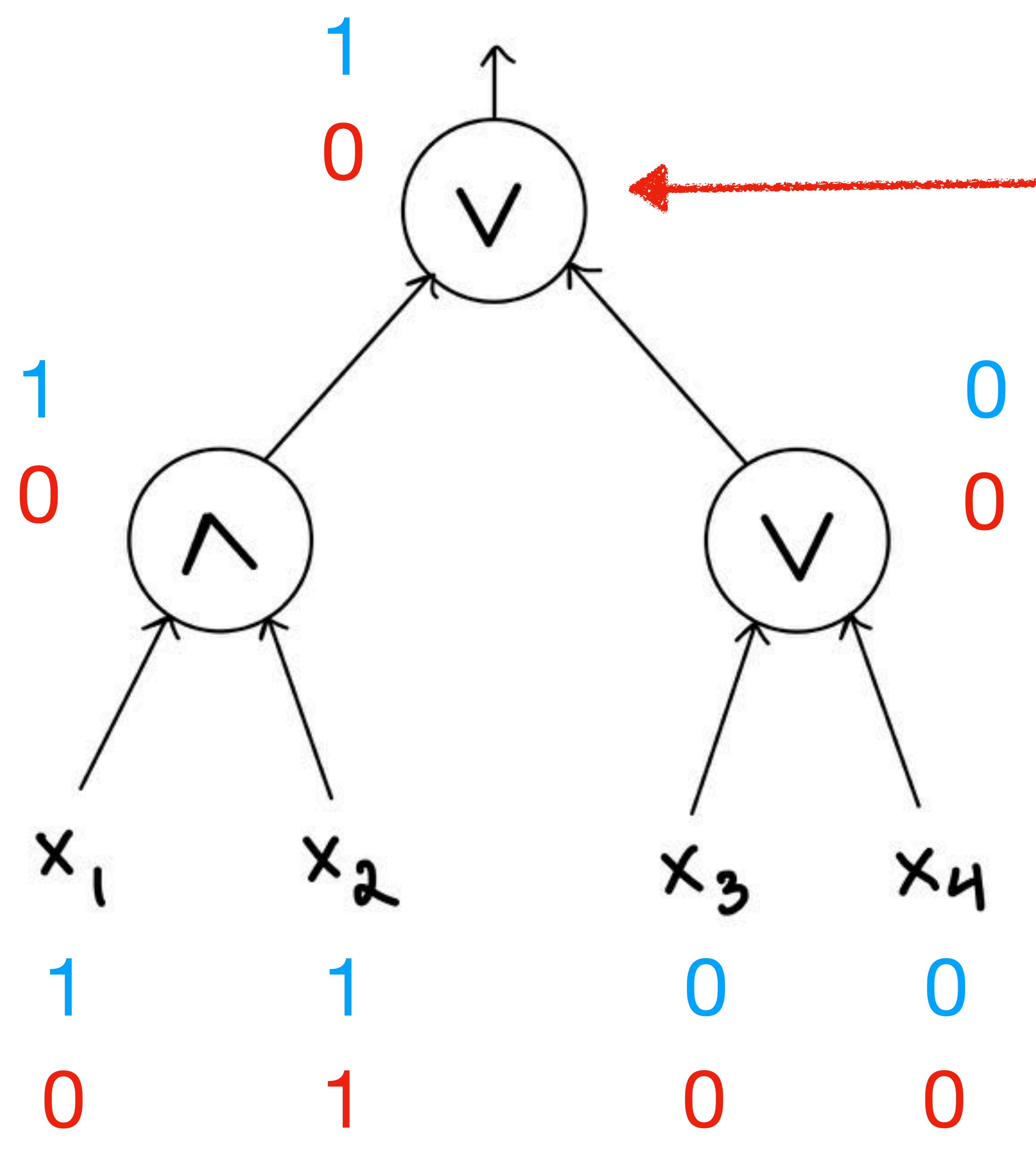
$$f = (x_1 \wedge x_2) \vee x_3 \vee x_4$$

Formulas ≡ Communication

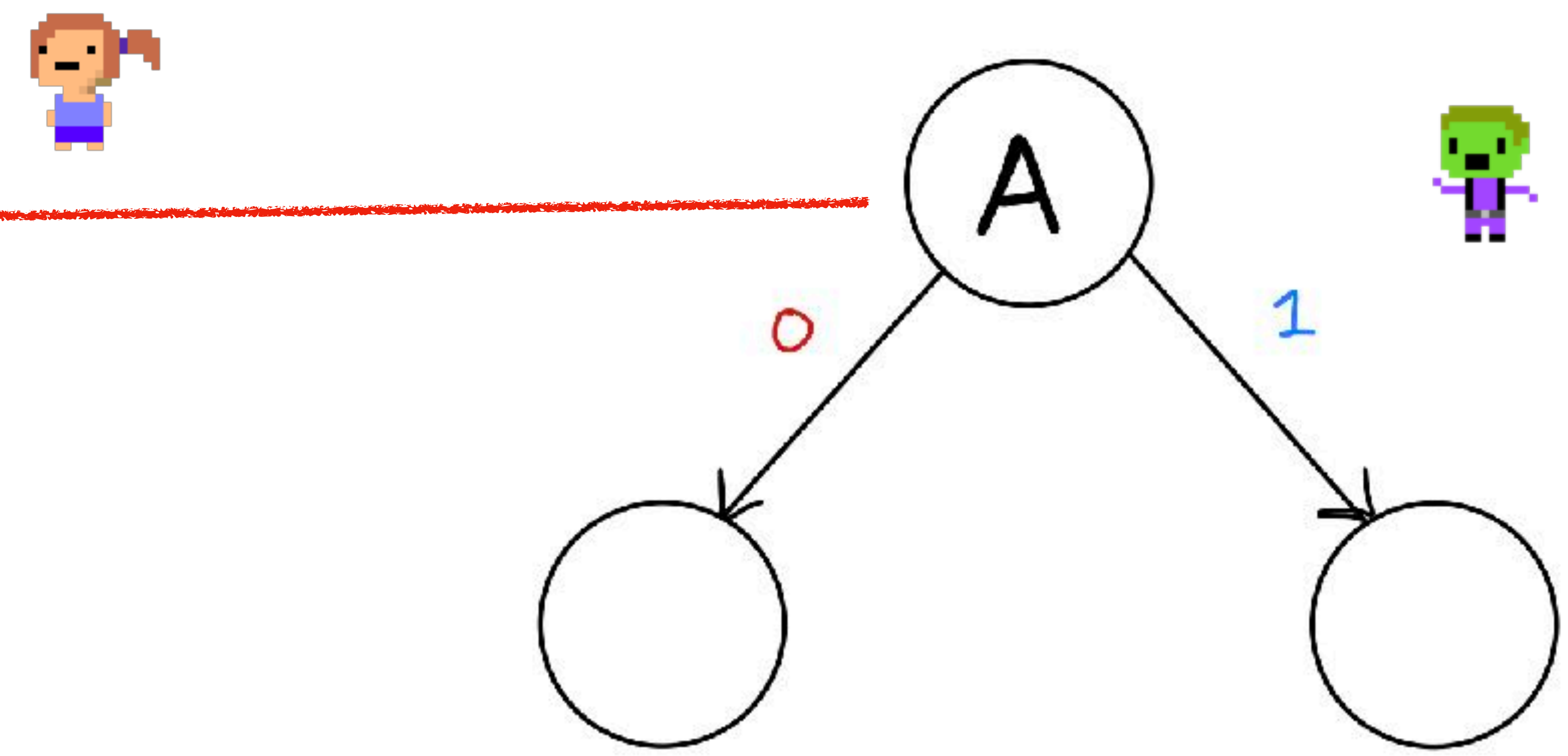
$x = (1,1,0,0)$

$y = (0,1,0,0)$

Boolean Formula for f



Protocol for $\mathrm{KW}(f)$



$$f = (x_1 \wedge x_2) \vee x_3 \vee x_4$$

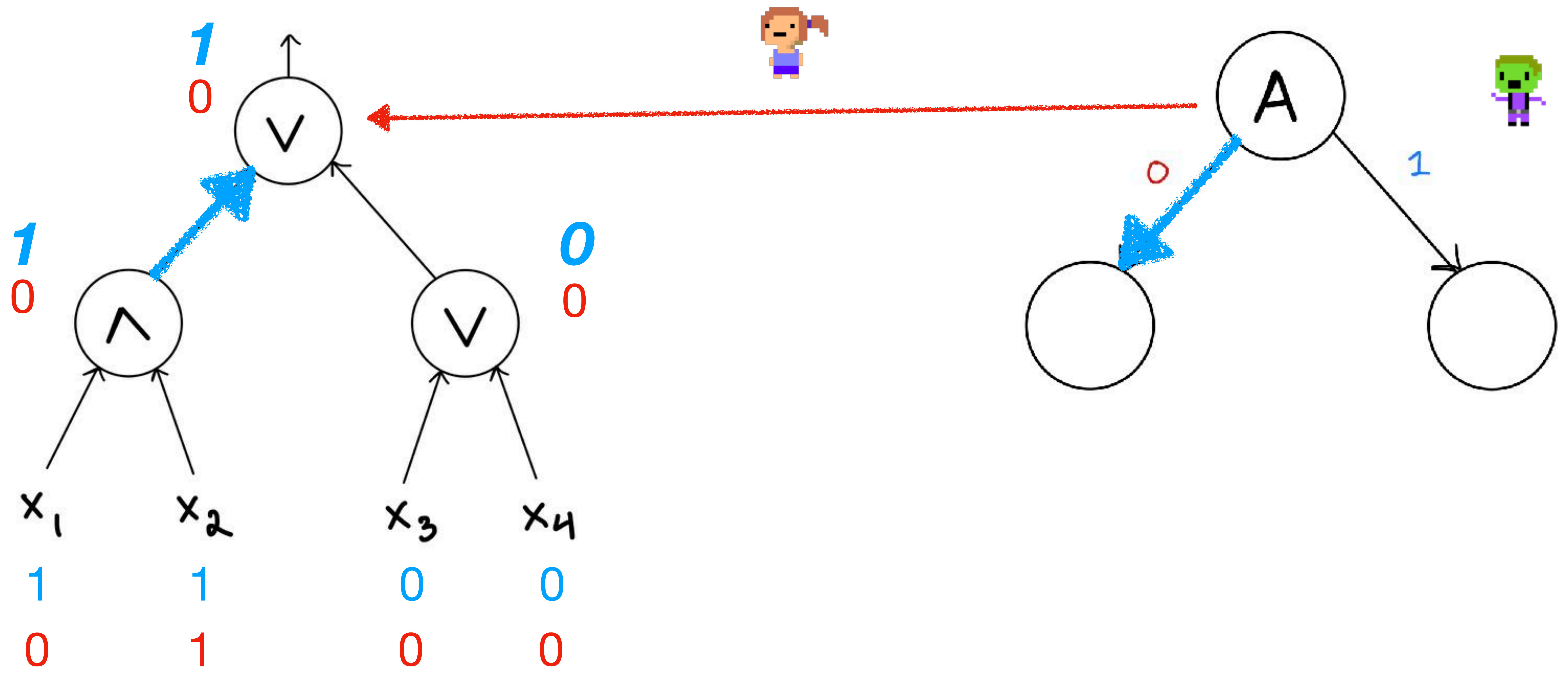
Formulas ≡ Communication

$x = (1,1,0,0)$

$y = (0,1,0,0)$

Boolean Formula for f

Protocol for $\mathrm{KW}(f)$



$$f = (x_1 \wedge x_2) \vee x_3 \vee x_4$$

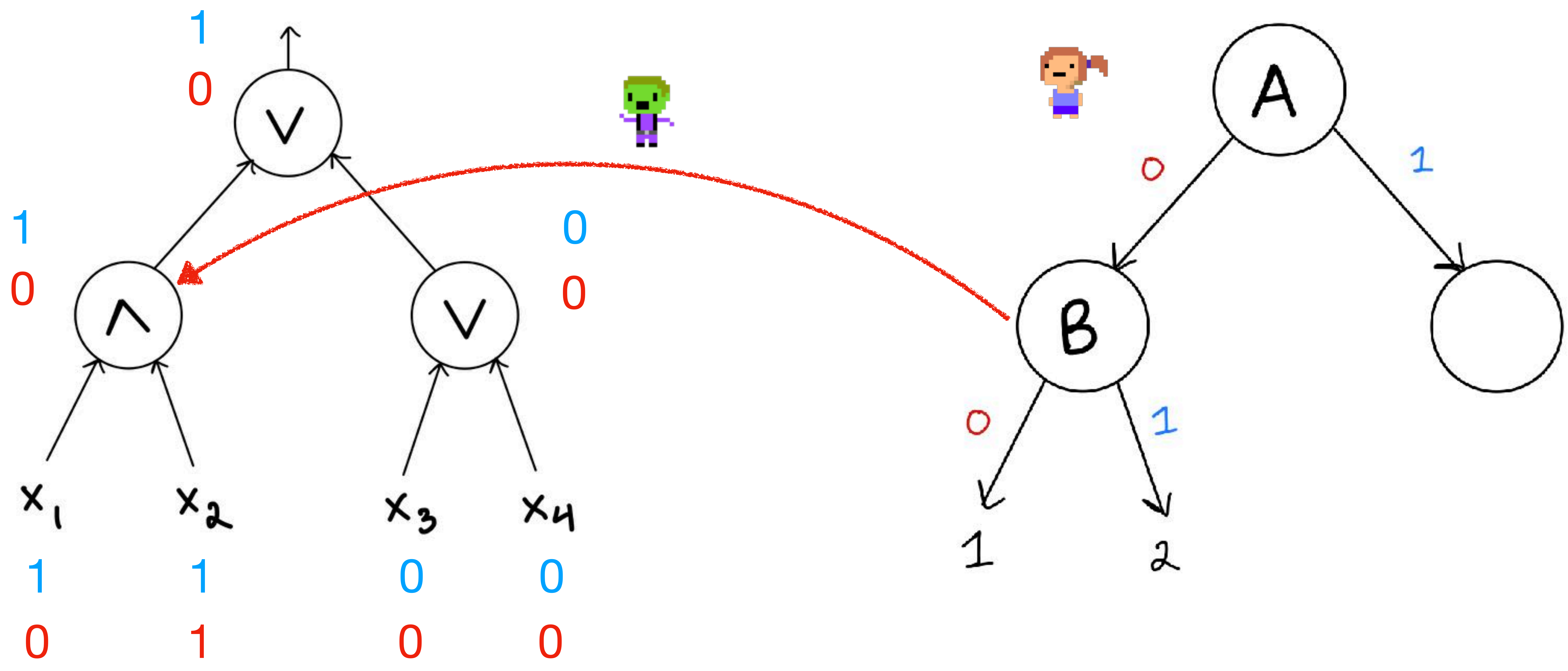
Formulas ≡ Communication

$x = (1,1,0,0)$

$y = (0,1,0,0)$

Boolean Formula for f

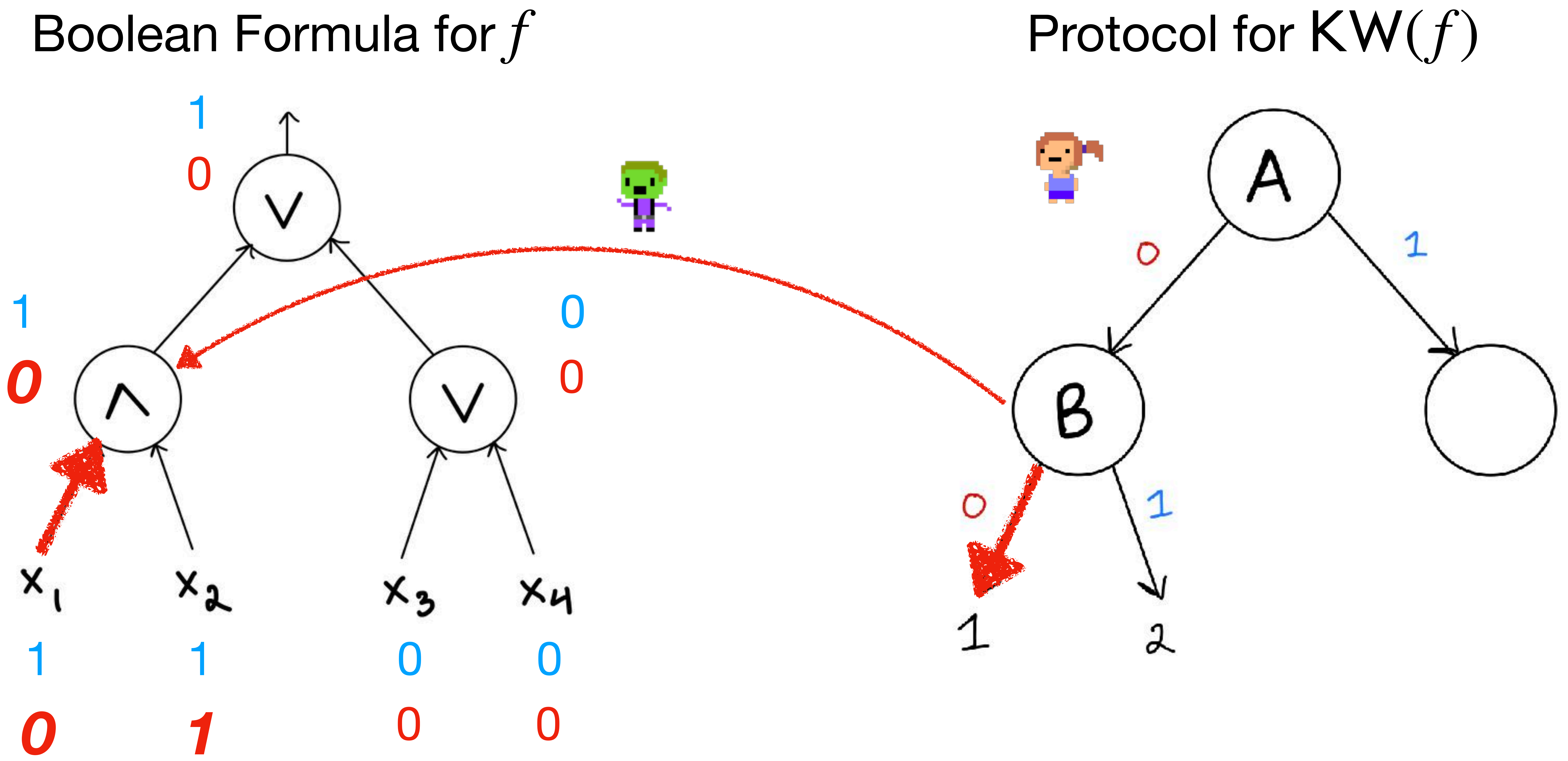
Protocol for $\mathrm{KW}(f)$



$$f = (x_1 \wedge x_2) \vee x_3 \vee x_4$$

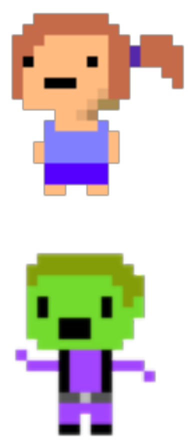
Formulas ≡ Communication

$x = (1,1,0,0)$
$y = (0,1,0,0)$



$$f = (x_1 \wedge x_2) \vee x_3 \vee x_4$$

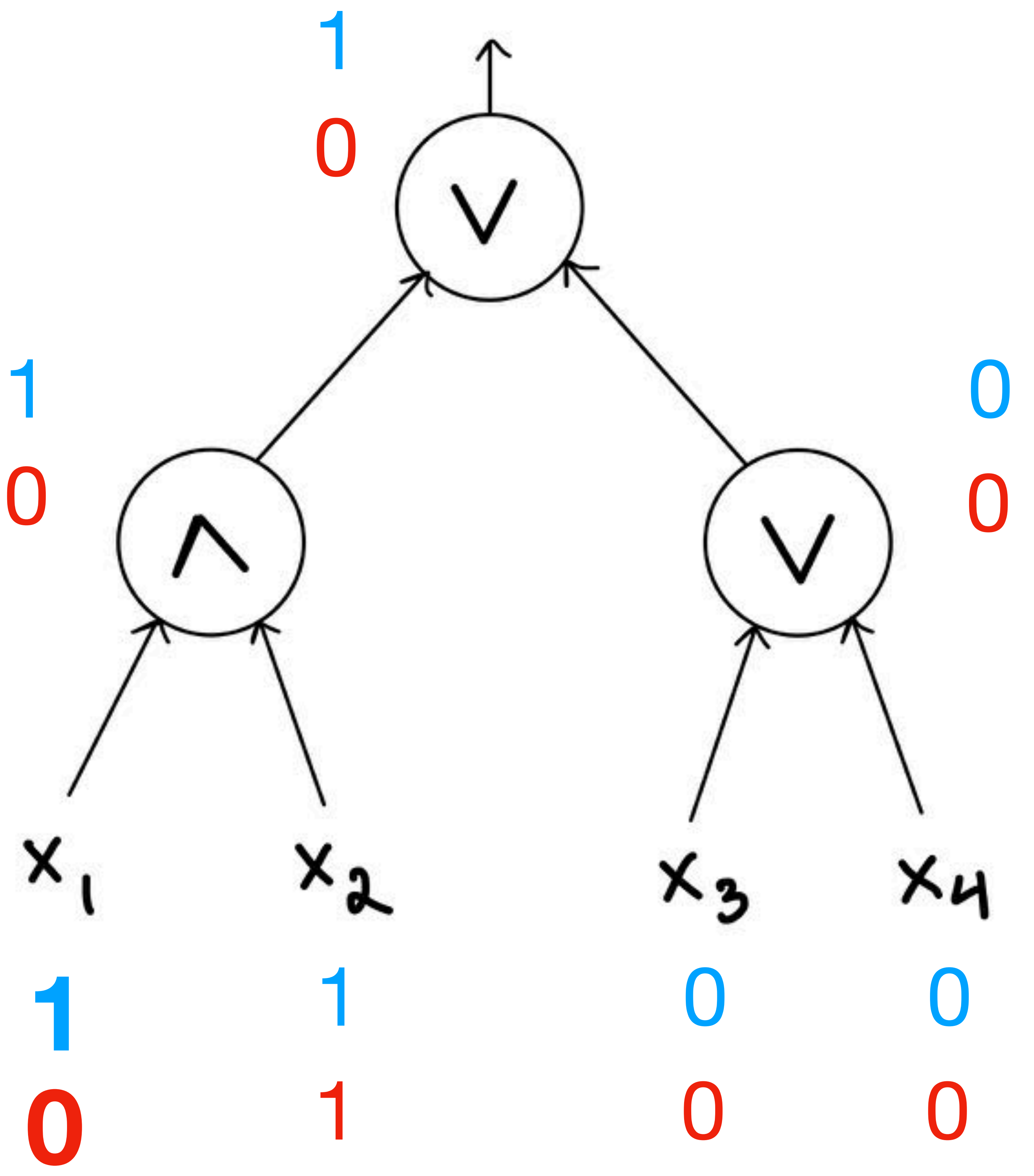
Formulas ≡ Communication



$x = (1,1,0,0)$

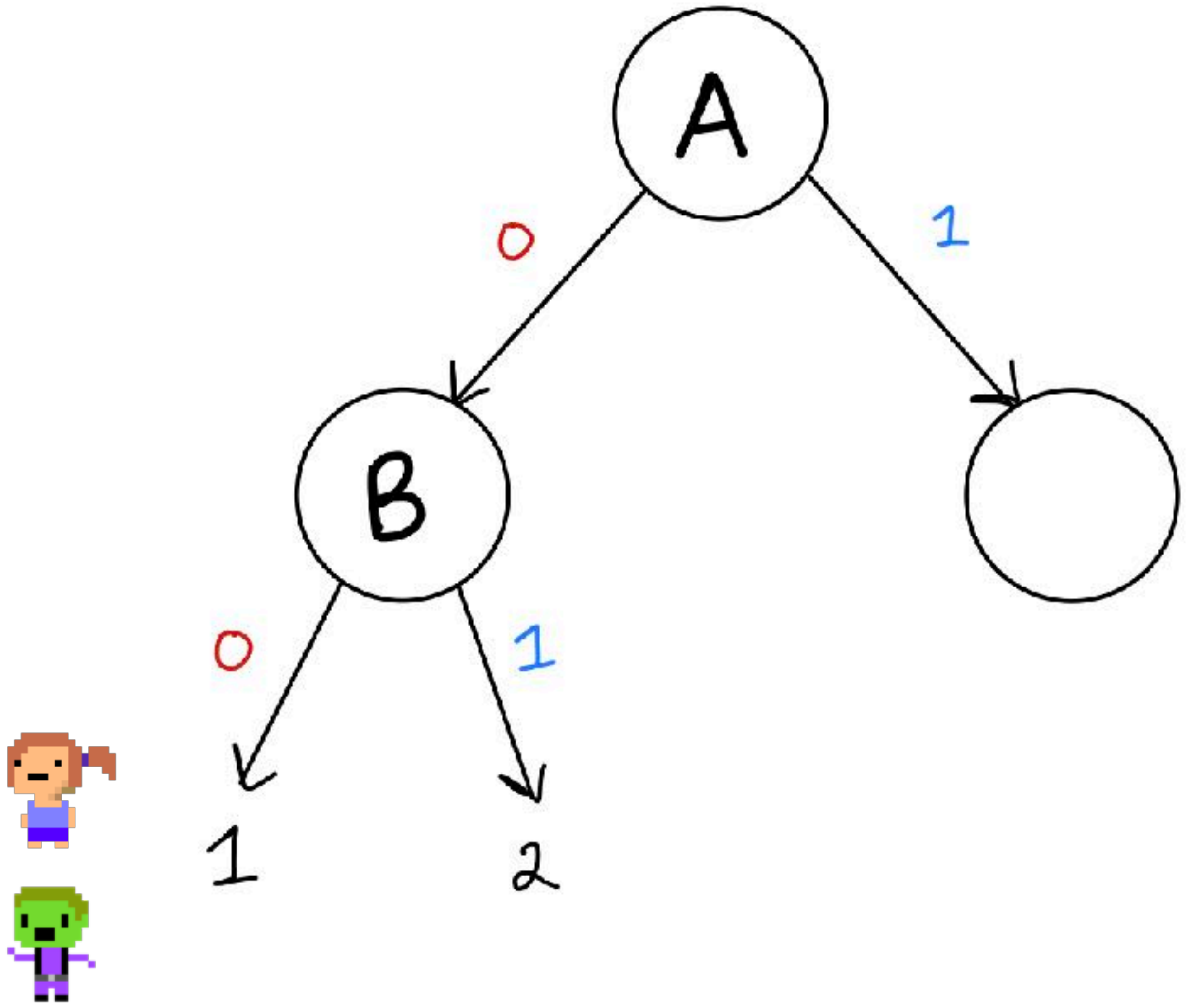
$y = (0,1,0,0)$

Boolean Formula for f



Finished!

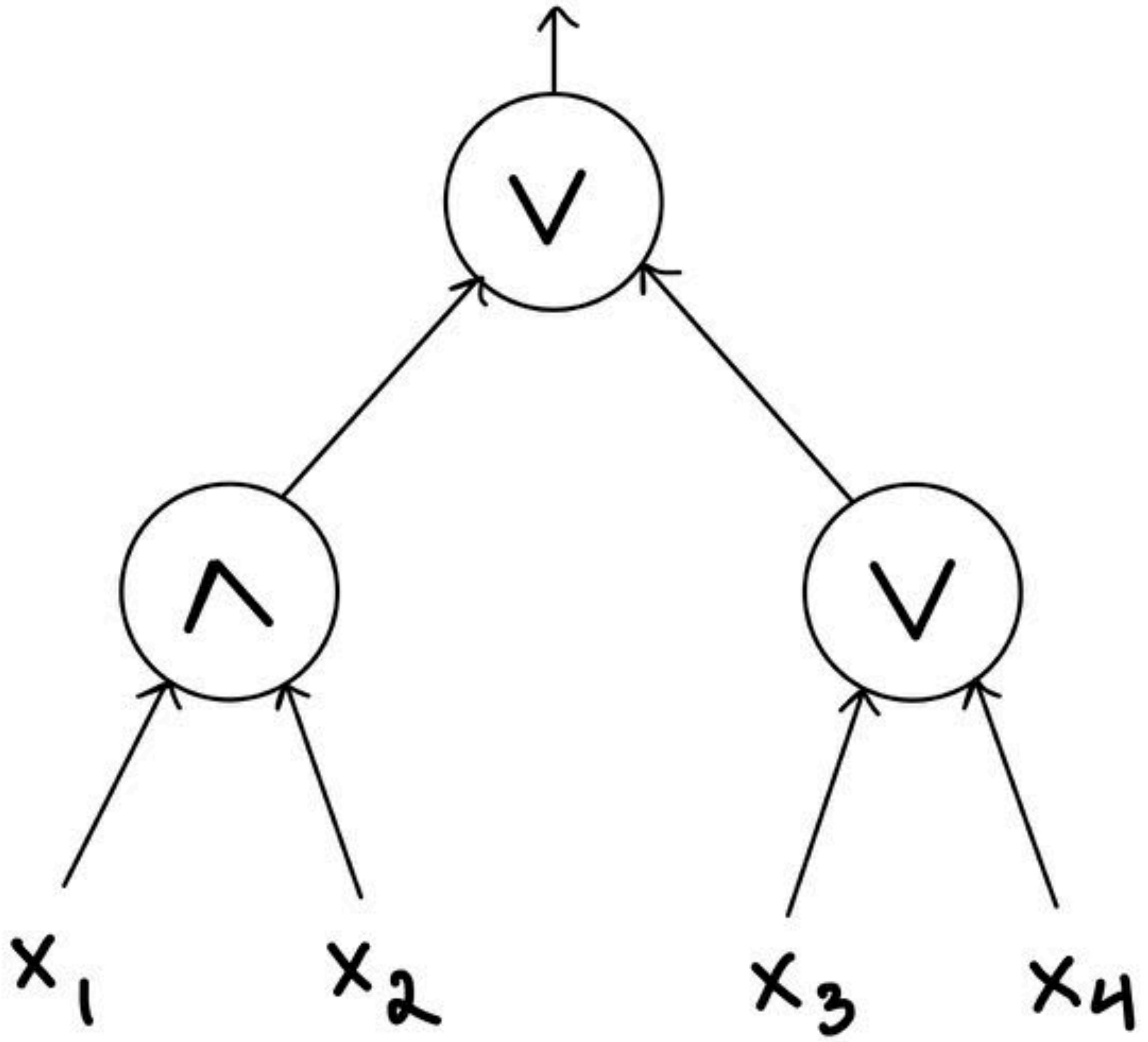
Protocol for $\mathrm{KW}(f)$



$$f = (x_1 \wedge x_2) \vee x_3 \vee x_4$$

Formulas ≡ Communication

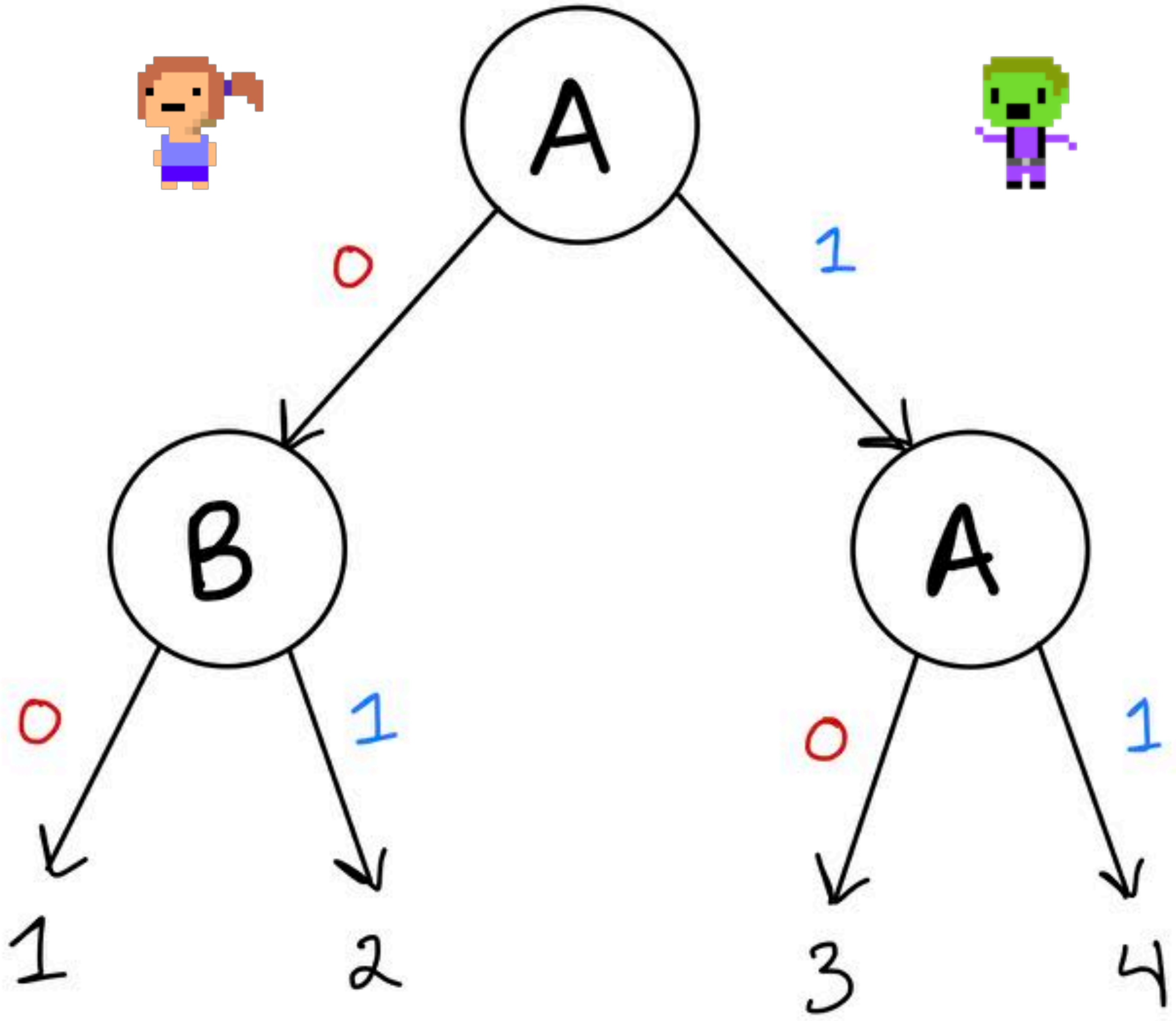
Boolean Formula for f



∨ gates

∧ gates

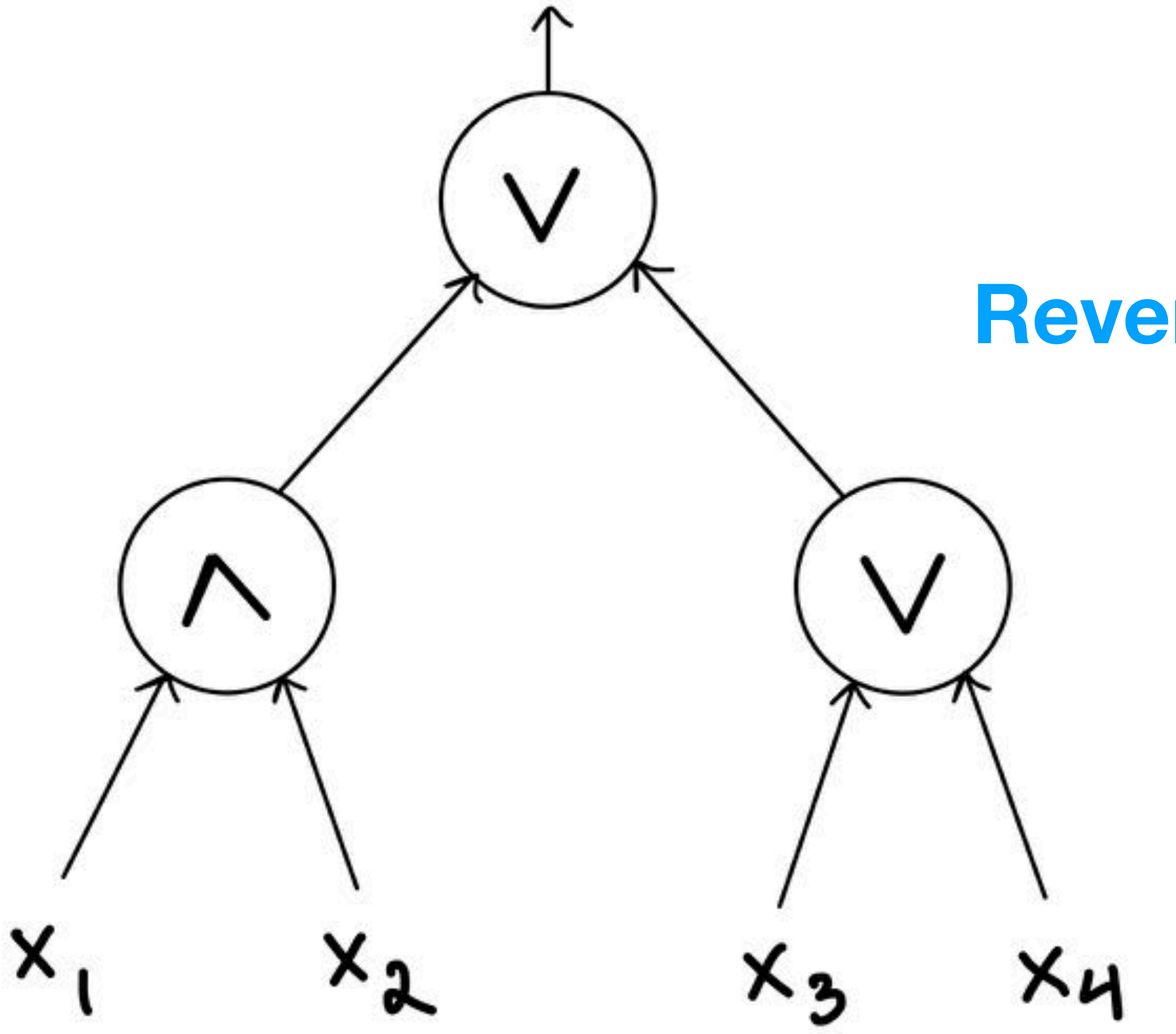
Protocol for $\mathrm{KW}(f)$



$$f = (x_1 \wedge x_2) \vee x_3 \vee x_4$$

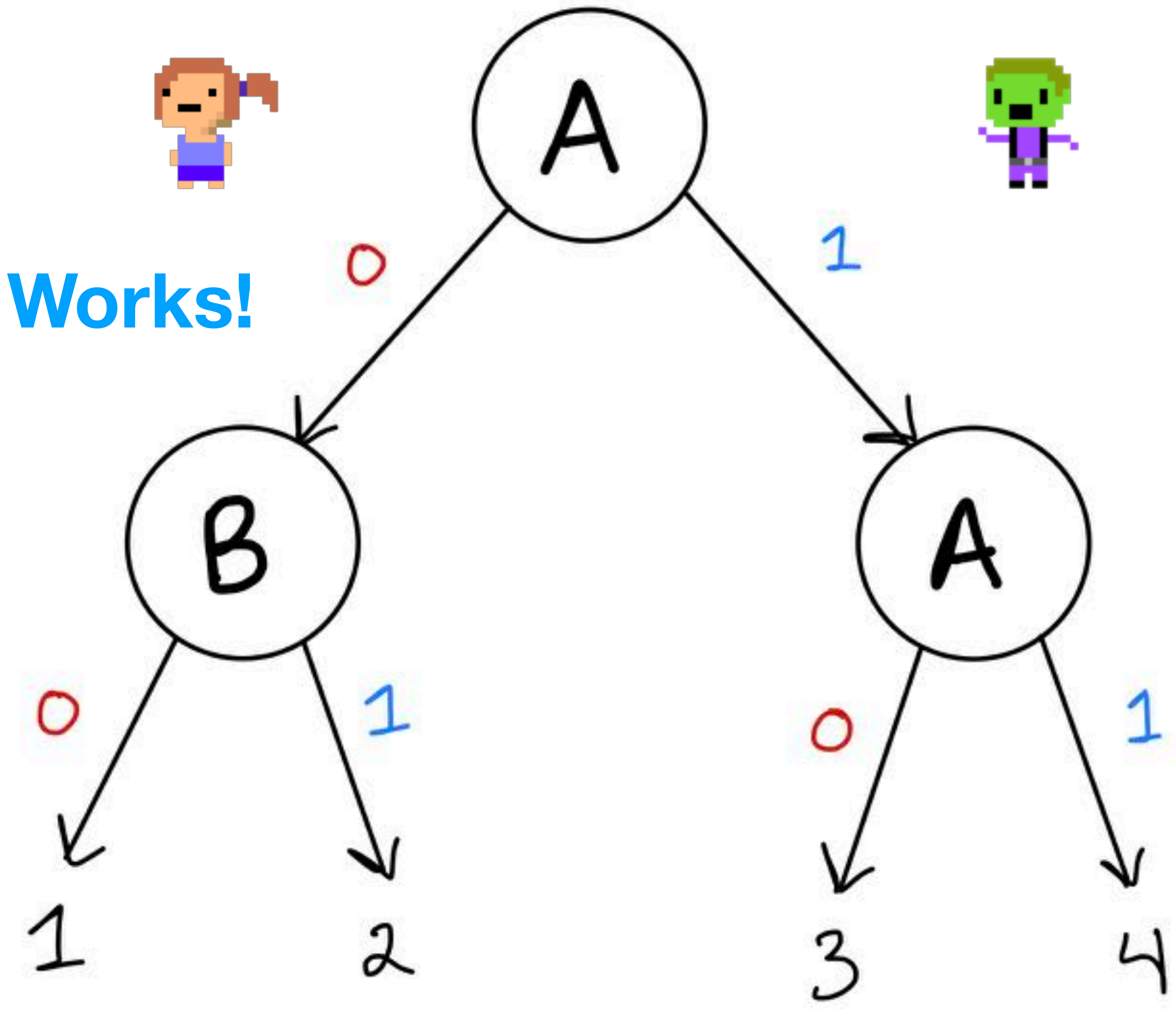
Formulas $\equiv$ Communication

Boolean Formula for f



Reverse Direction Also Works!

Protocol for $\mathrm{KW}(f)$



$$f = (x_1 \wedge x_2) \vee x_3 \vee x_4$$

Formulas $\equiv$ Communication

Theorem.

Let $f : \{0,1\}^n \to \{0,1,*\}$ be a partial boolean function. Then

Size $\leq s$, depth $\leq d$ Boolean formula for f

if and only if

Size $\leq s$, depth $\leq d$ communication protocol for $\mathrm{KW}(f)$

Correspondence is stronger: essentially the same object!

Formulas $\equiv$ Communication

Theorem.

Let $f : \{0,1\}^n \to \{0,1,*\}$ be a partial monotone boolean function. Then

Size $\leq s$, depth $\leq d$ monotone Boolean formula for f

if and only if

Size $\leq s$, depth $\leq d$ communication protocol for mKW(f)

Correspondence is stronger: essentially the same object!

Total Search Problems in Communication

- Let X, Y, O be finite, $\mathcal{S} \subseteq X \times Y \times O$.
- $\mathcal{S}(x, y) := \{o \in O : (x, y, o) \in \mathcal{S}\}$ are **feasible solutions** for (x, y).
- $\mathcal{S}$ is a **total search problem** if $\forall (x, y) : \mathcal{S}(x, y) \neq \emptyset$
- $\mathsf{FP}^{cc}(\mathcal{S}) :=$ **Communication Complexity** of $\mathcal{S}$

  $:=$ Depth of shallowest protocol solving $\mathcal{S}$
- $\mathsf{FP}^{cc} :=$ All total search problems $\mathcal{S}$ such that

$$\mathsf{FP}^{cc}(\mathcal{S}) = \log^{O(1)}(n)$$

Communication TFNP

- A **certificate cover** of $\mathscr{S}$ is a set of rectangles $\mathscr{R}$ such that every $(x, y) \in X \times Y$ is consistent with some $R \in \mathscr{R}$.

$$\text{TFNP}^{dt}(\mathscr{S}) := \min_{\mathscr{R} \text{ cover}} \log |\mathscr{R}|$$

- NP **Algorithm**: Given $(x, y) \in X \times Y$,
  - Non-deterministically guess $R \in \mathscr{R}$ ($\log |\mathscr{R}|$ bits)
  - Verify that $(x, y) \in R$ by exchanging 1 bit of communication

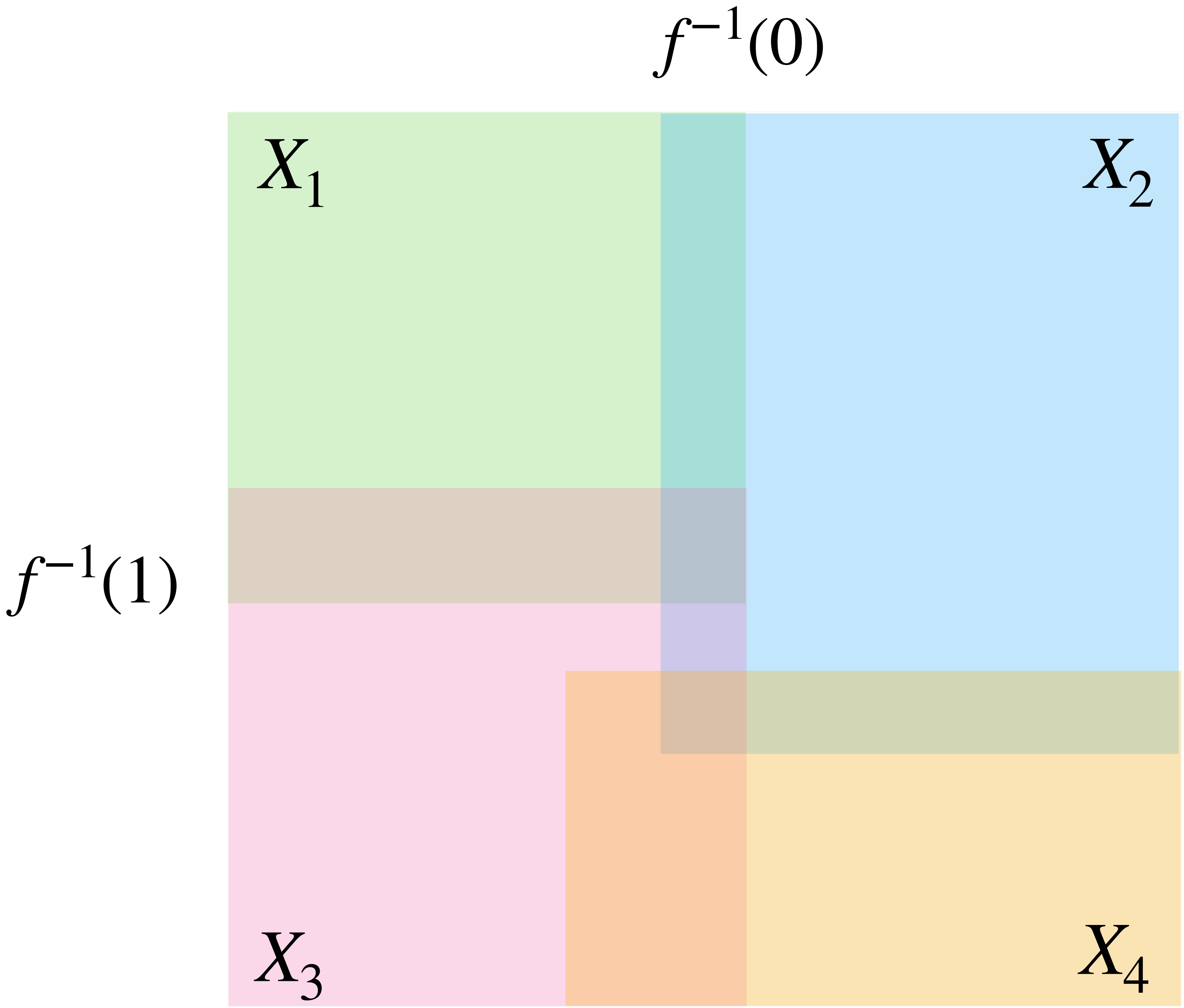
What's So Special About mKW(f)?

- If $f : \{0,1\}^n \to \{0,1,*\}$ and $i \in [n]$ let

$$X_i := \{x \in f^{-1}(1) : x_i = 1\} \times \{y \in f^{-1}(0) : y_i = 0\}$$

- The set $\{X_i : i \in [n]\}$ is a rectangle cover for mKW(f)

$$\therefore \text{TFNP}^{cc}(\text{mKW}(f)) \leq \log n$$



What's So Special About $\mathsf{mKW}(f)$?

Fact [R90, G01]. **Every** rectangle cover $\mathscr{R} = \{R_1, \ldots, R_n\}$ of a set $U \times V$ is equivalent to mKW_f for some partial monotone $f : \{0,1\}^n \to \{0,1,*\}$.

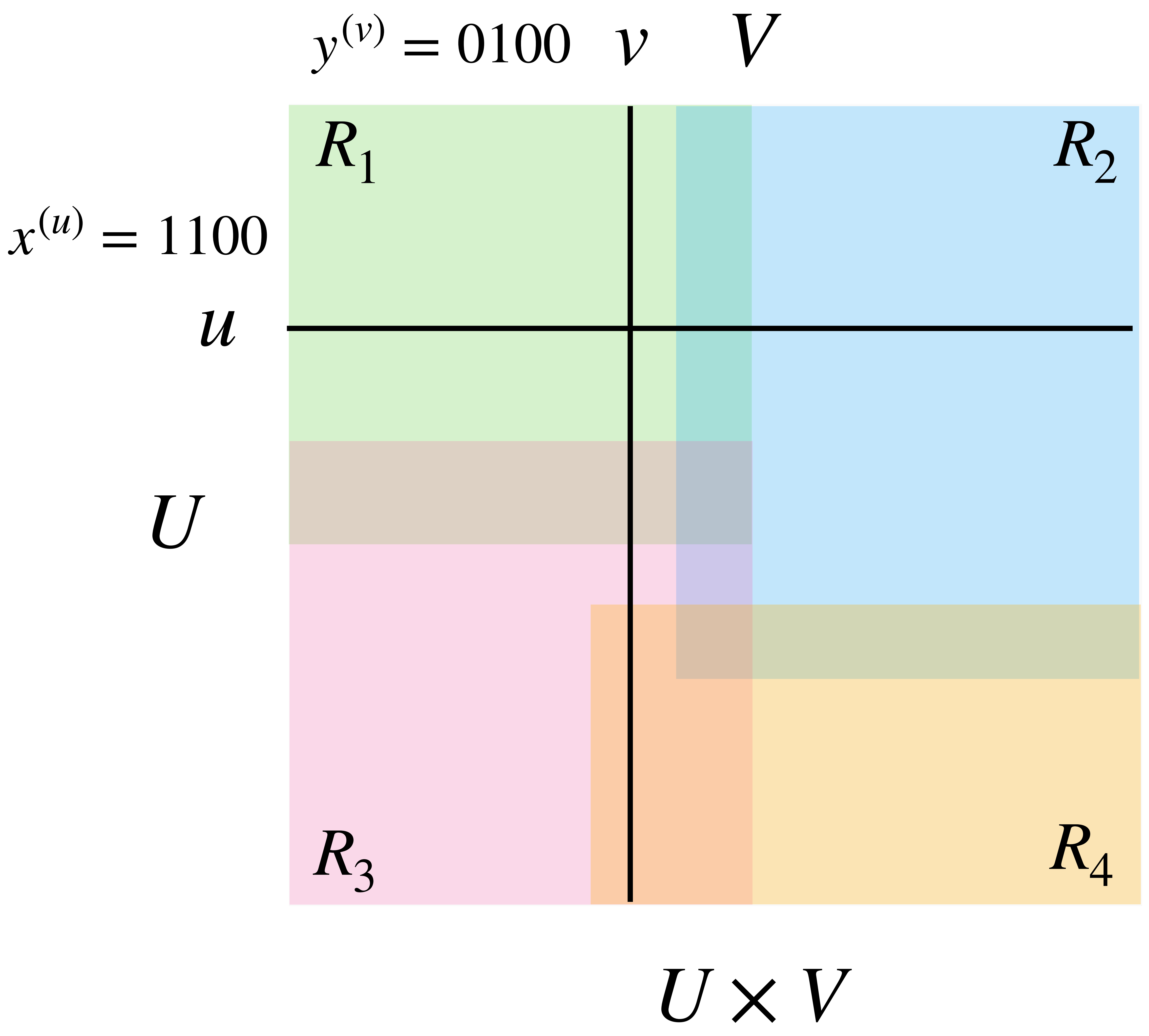
Proof. For each $u \in U$ let $x^{(u)} \in \{0,1\}^n$ be

$$x_i^{(u)} = 1 \Leftrightarrow u \in R_i$$

For each $v \in V$ let $y^{(v)} \in \{0,1\}^n$ be such that

$$y_i^{(v)} = 0 \Leftrightarrow v \in R_i$$

Define $f(x^{(u)}) = 1$ for all u, $f(y^{(v)}) = 0$ for all v.

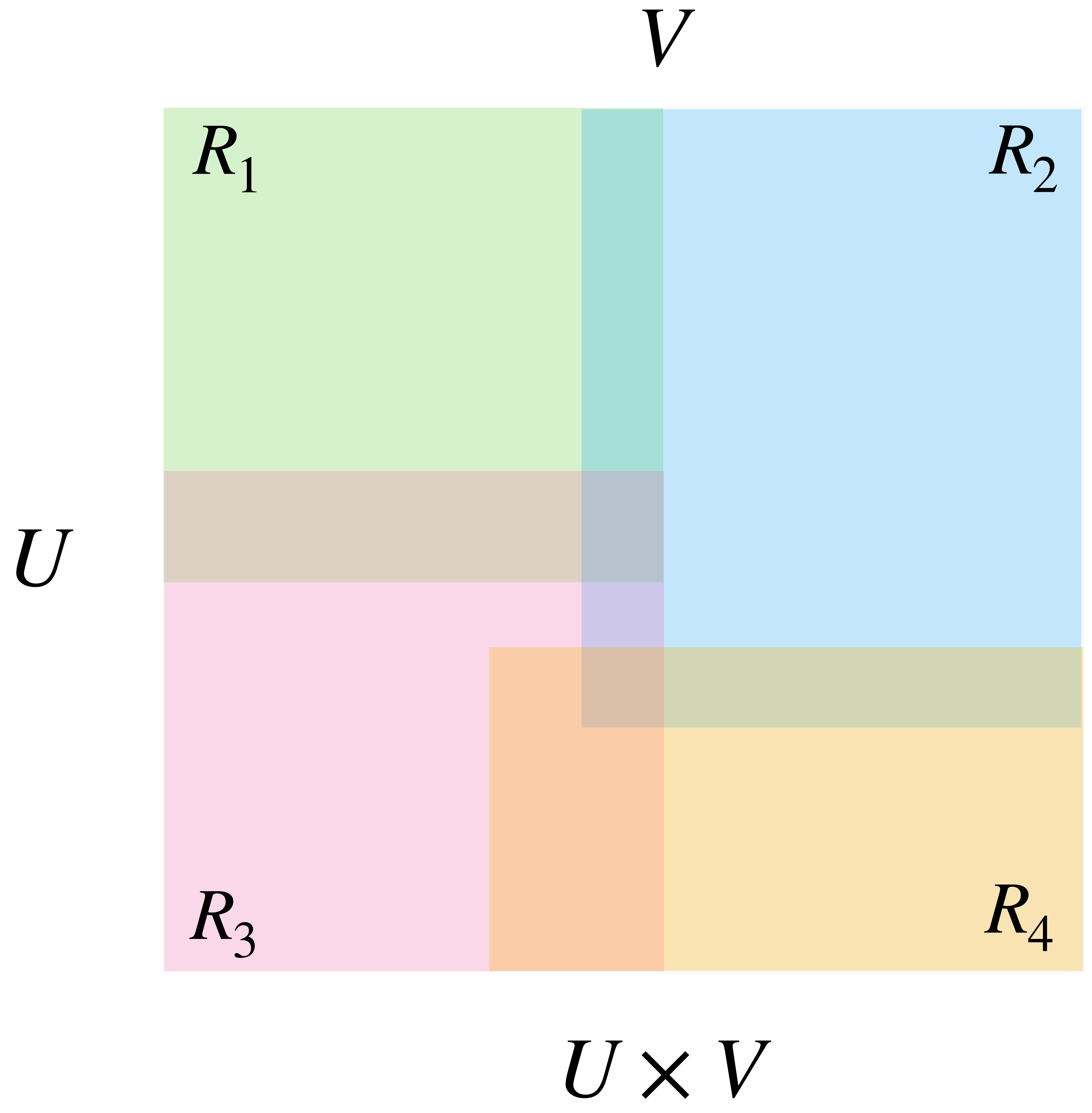


What's So Special About mKW(f)?

Lesson

Any total search problem $\mathcal{S} \in \mathsf{TFNP}^{cc}$ can be reduced to solving mKW(f) for some partial $f : \{0,1\}^n \to \{0,1,*\}$.

(Converse holds, under reasonable assumptions.)



Summary

- Any $f: \{0,1\}^n \to \{0,1,*\}$ has a Karchmer-Wigderson game
- Comm. Protocols for mKW(f) $\equiv$ Boolean formulas computing f
- mKW(f) is complete* for TFNP^{cc}

Can we capture **other** circuit models?

These Stories Are The Same

Bottom-up models (proofs, circuits)

are captured by

Top-down algorithms (decision trees, comm. protocols)

- Search(F) and mKW(f)
  - **Capture** the complexity of these processes
  - Are **canonical** examples of their respective TFNP classes
- We now outline a **general theory** capturing both of these cases.

Part 3

The TFNP Program in Proof and Circuit Complexity

Classical Theory of TFNP

- Introduced by Papadimitriou [Pap 94]
- TFNP := Class of NP problems for which a witness **always** exists.
- Subclasses are defined via **polynomial-time reductions** to particular problems.
  - Problems often represent theorems used to prove existence results; e.g. Handshaking Lemma, Fixed-Point Theorems, Sperner's Lemma, …
- Vibrant theory with many connections to other fields:
  - Game Theory, Cryptography, Combinatorics, **Bounded Arithmetic**, …

Example

Handshaking Lemma.

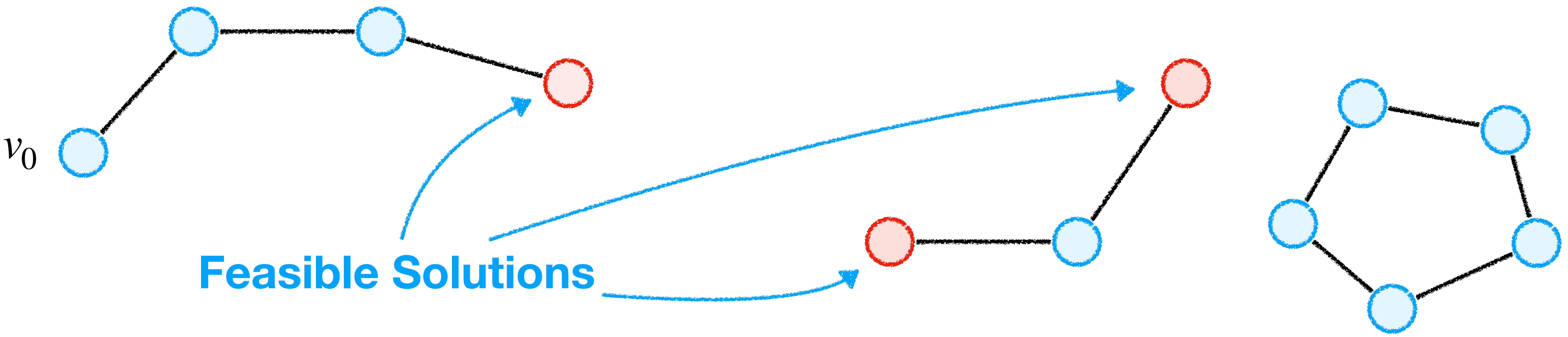
Every graph has an even number of odd-degree nodes.

PPA

Input: Set of nodes V, $v_0 \in V$, neighbourhood function $N(u) \subseteq V$ with $|N(u)| \leq 2$.

Feasible Solutions: Let $G = (V, E)$ be s.t. $uv \in E$ iff $u \in N(v), v \in N(u)$.

- v_0 if $\deg(v_0) \neq 1$, or
- $v \in V$ if $v \neq v_0$ and $\deg(v) = 1$



PPA

Input: Set of nodes V, $v_0 \in V$, neighbourhood function $N(u) \subseteq V$ with $|N(u)| \leq 2$.

Feasible Solutions: Let $G = (V, E)$ be s.t. $uv \in E$ iff $u \in N(v), v \in N(u)$.

- v_0 if $\deg(v_0) \neq 1$, or
- $v \in V$ if $v \neq v_0$ and $\deg(v) = 1$

- Complexity class PPA contains total search problems **reducible** to this problem
- Have poly-time Turing Machines $\mathbf{N}, \mathbf{S}$ such that on input $x \in \{0,1\}^*$
  - $\mathbf{N}(x, u)$:= Neighbourhood of node u on input x
  - $\mathbf{S}(x, u)$:= Solution labelling node u on input x
- Given x, get graph G_x, solve PPA problem on that graph, output solutions

Prominent Subclasses of TFNP

TFNP

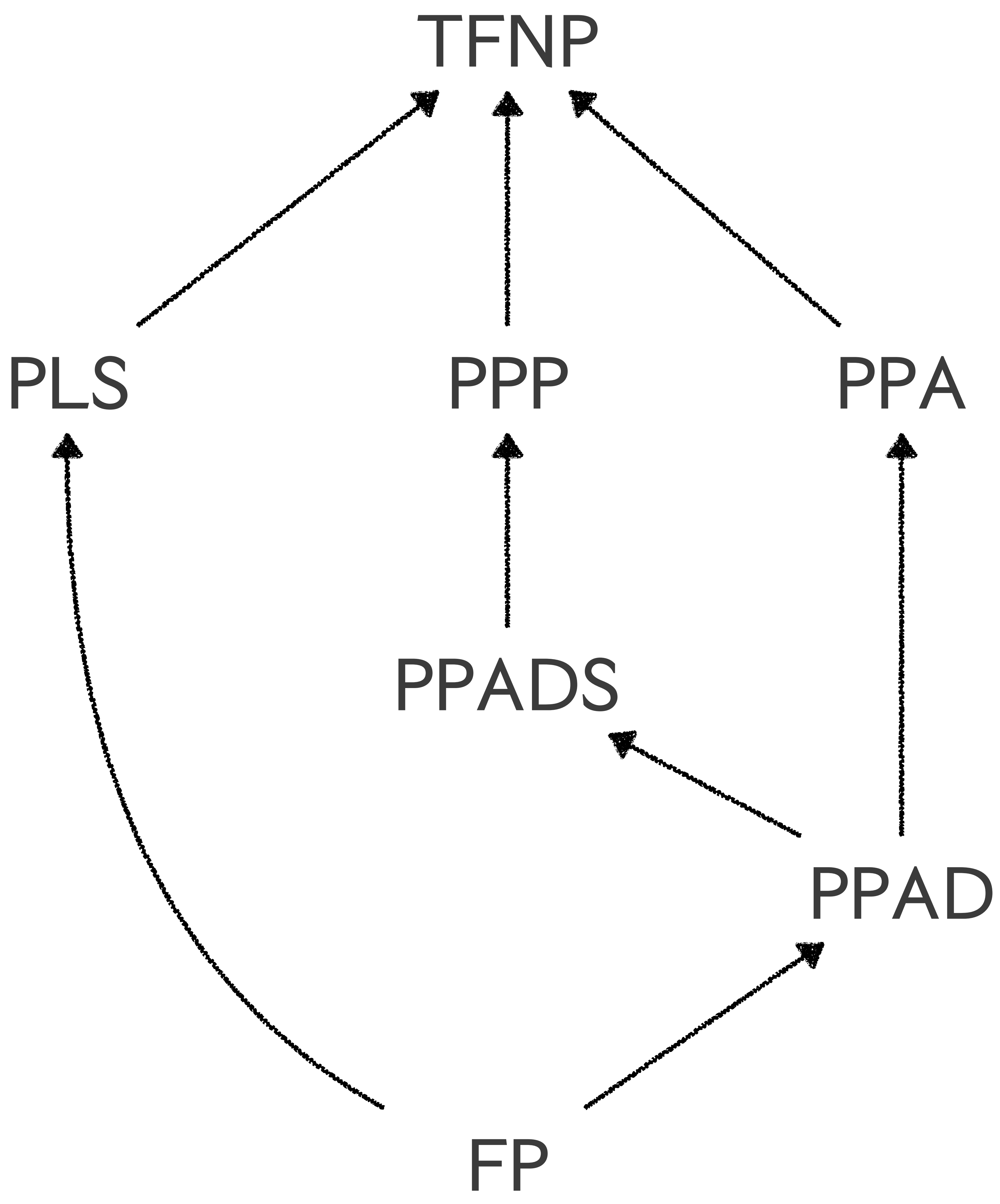
Prominent Subclasses of TFNP

TFNP

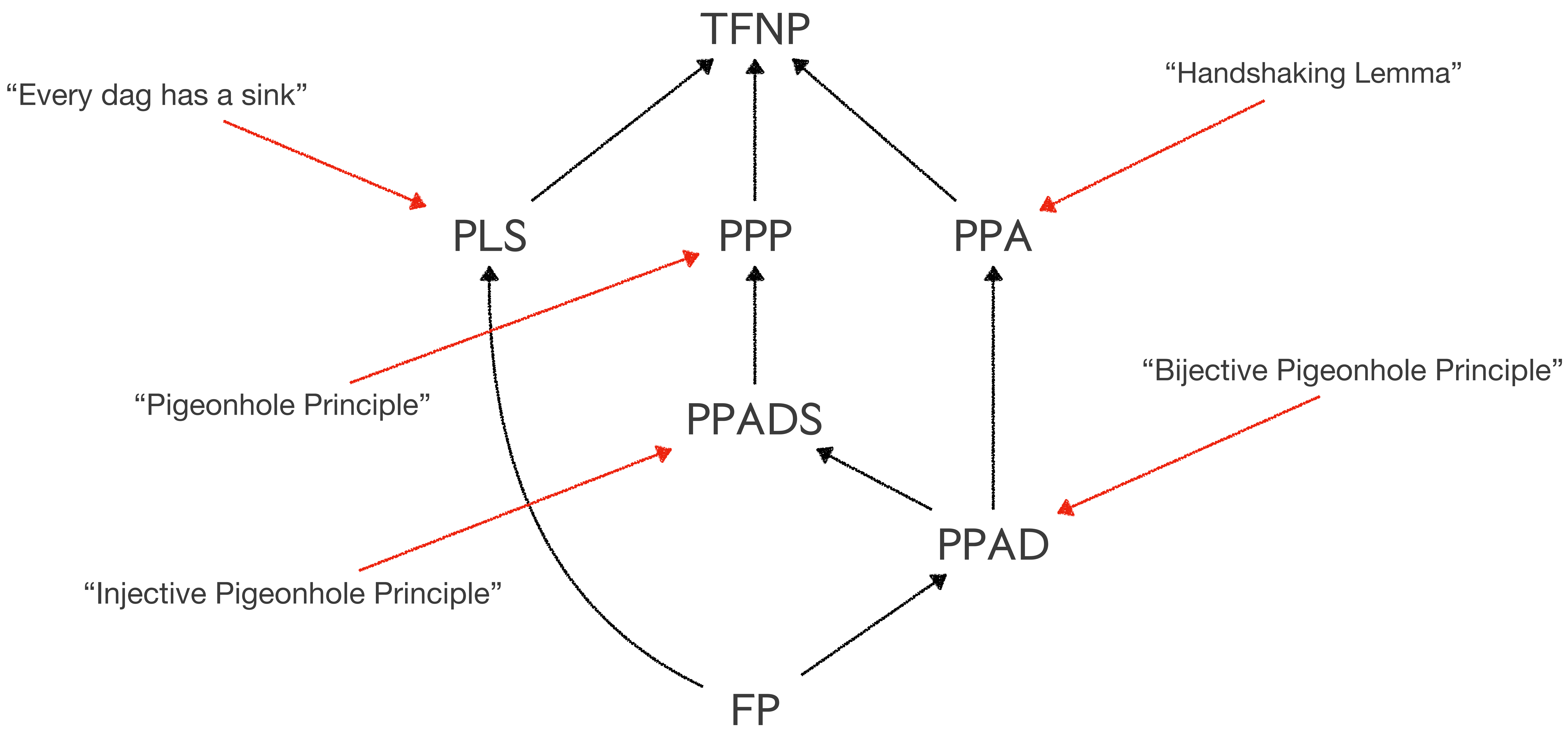
PPA

Guess solution, verify using neighbourhood functions

Prominent Subclasses of TFNP



Prominent Subclasses of TFNP



Communication and Query TFNP

- We have seen the classes:
  - FP^{dt} := total $\mathcal{S}$ with $\log^{O(1)} n$ - depth decision trees (**tree resolution**)
  - TFNP^{dt} := total $\mathcal{S}$ with $\log^{O(1)} n$ - width certificates (**narrow unsat. CNFs** F)
  - FP^{cc} := total $\mathcal{S}$ with $\log^{O(1)} n$ - depth comm. protocols (**boolean formulas**)
  - TFNP^{cc} := total $\mathcal{S}$ with $\log^{O(1)} n$ - size rectangle covers ($\mathrm{mKW}(f)$)
- Can define other classes by reductions.
  - Use either shallow **decision trees** or shallow **communication protocols**, rather than Turing Machines.
  - Can characterize other proof systems and circuit classes!

Query PPA

Input: Set of nodes V, $v_0 \in V$, neighbourhood function $N(u) \subseteq V$ with $|N(u)| \leq 2$.

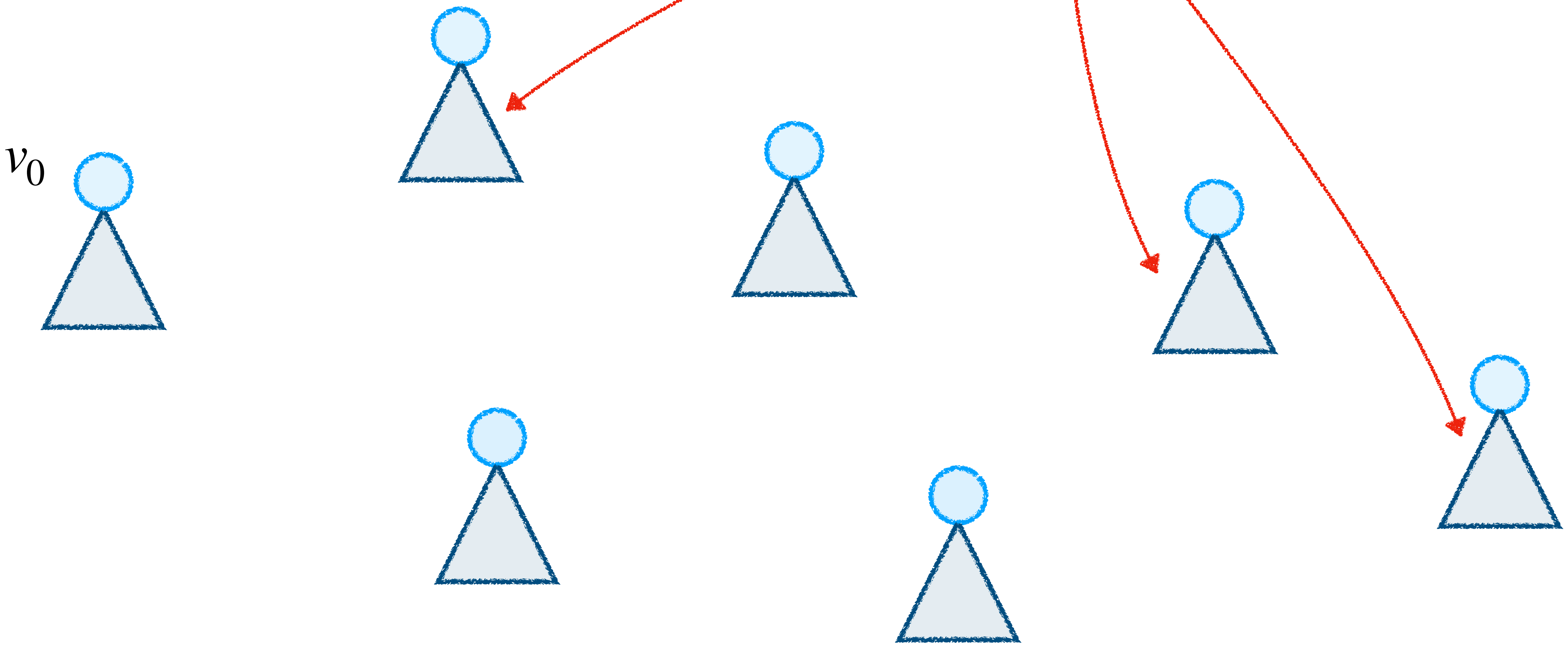
Feasible Solutions: Let $G = (V, E)$ be s.t. $uv \in E$ iff $u \in N(v), v \in N(u)$.

- v_0 if $\deg(v_0) \neq 1$, or
- $v \in V$ if $v \neq v_0$ and $\deg(v) = 1$

- To solve $\mathcal{S} \subseteq \{0,1\}^n \times O$, we reduce to above using decision trees:
  - $\mathbf{N}_u(x)$ **:=** Decision tree for u, outputs neighbourhood of node u on input x
  - $o_u \in O$ **:=** Solution of $\mathcal{S}$ labelling node u on input x
- Given x, run all decision trees in parallel to get graph G_x, solve PPA problem on that graph, output solutions labelling feasible solutions of G_x

Query PPA

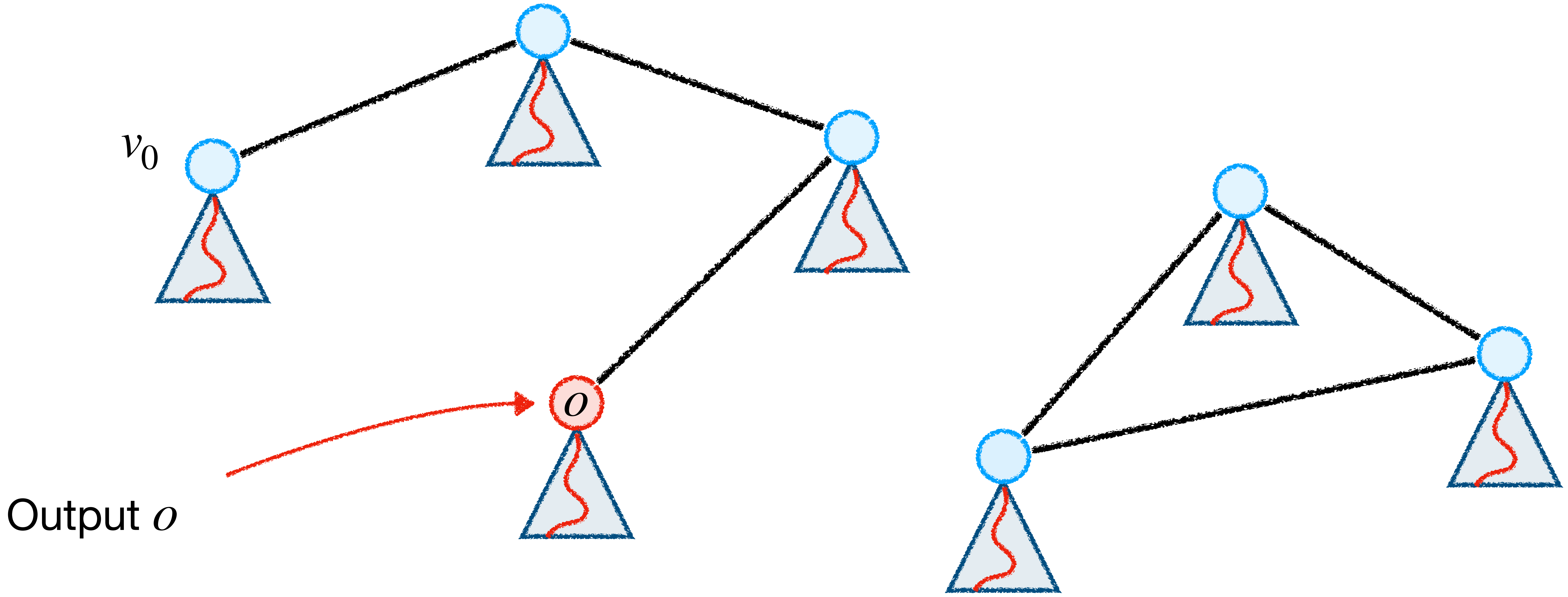
Decision trees querying $x \in \{0,1\}^n$



$$\mathcal{S} \subseteq \{0,1\}^n \times O$$

Query PPA

Evaluate all trees, output labels of feasible solutions

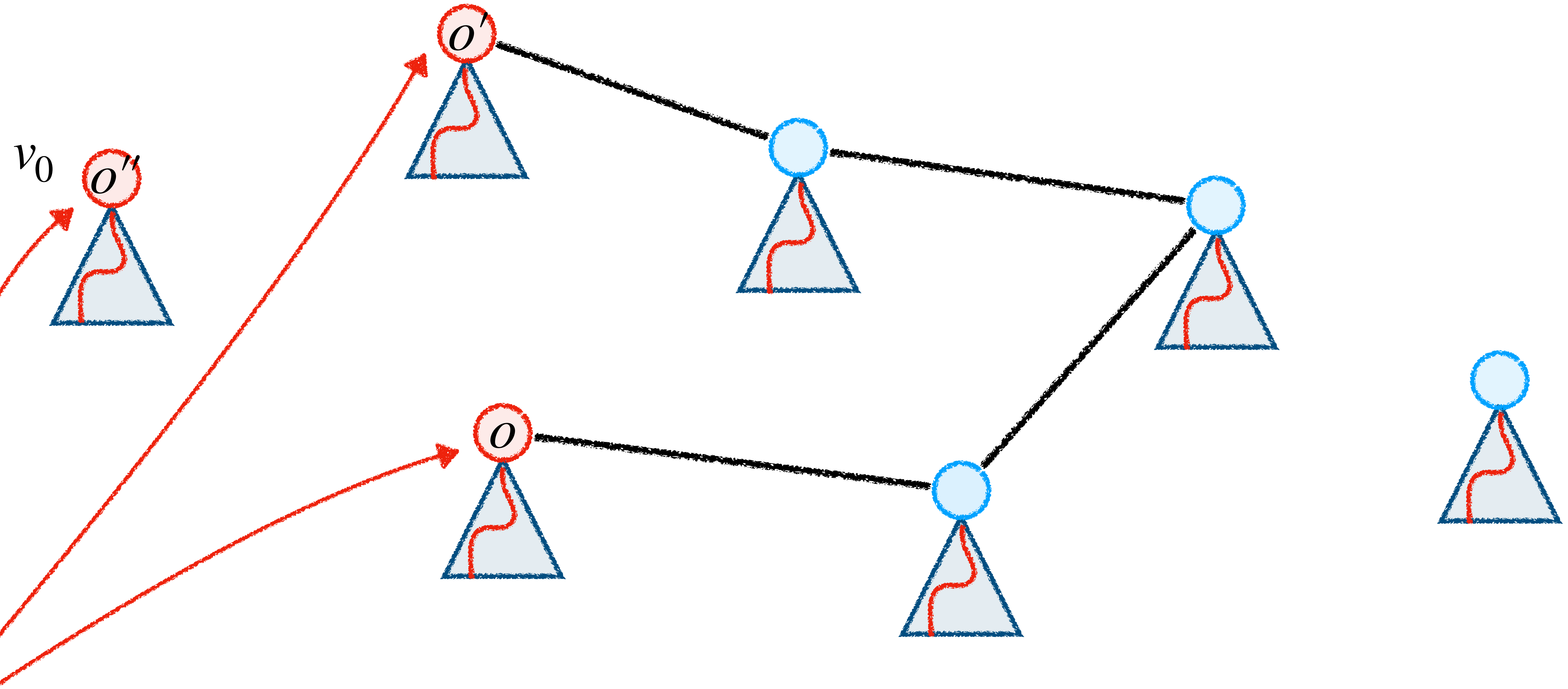


$$\mathcal{S} \subseteq \{0,1\}^n \times O$$

Query PPA

Evaluate all decision trees, output labels of feasible solutions

On a different input…



Output o, o', o''

$$\mathcal{S} \subseteq \{0,1\}^n \times O$$

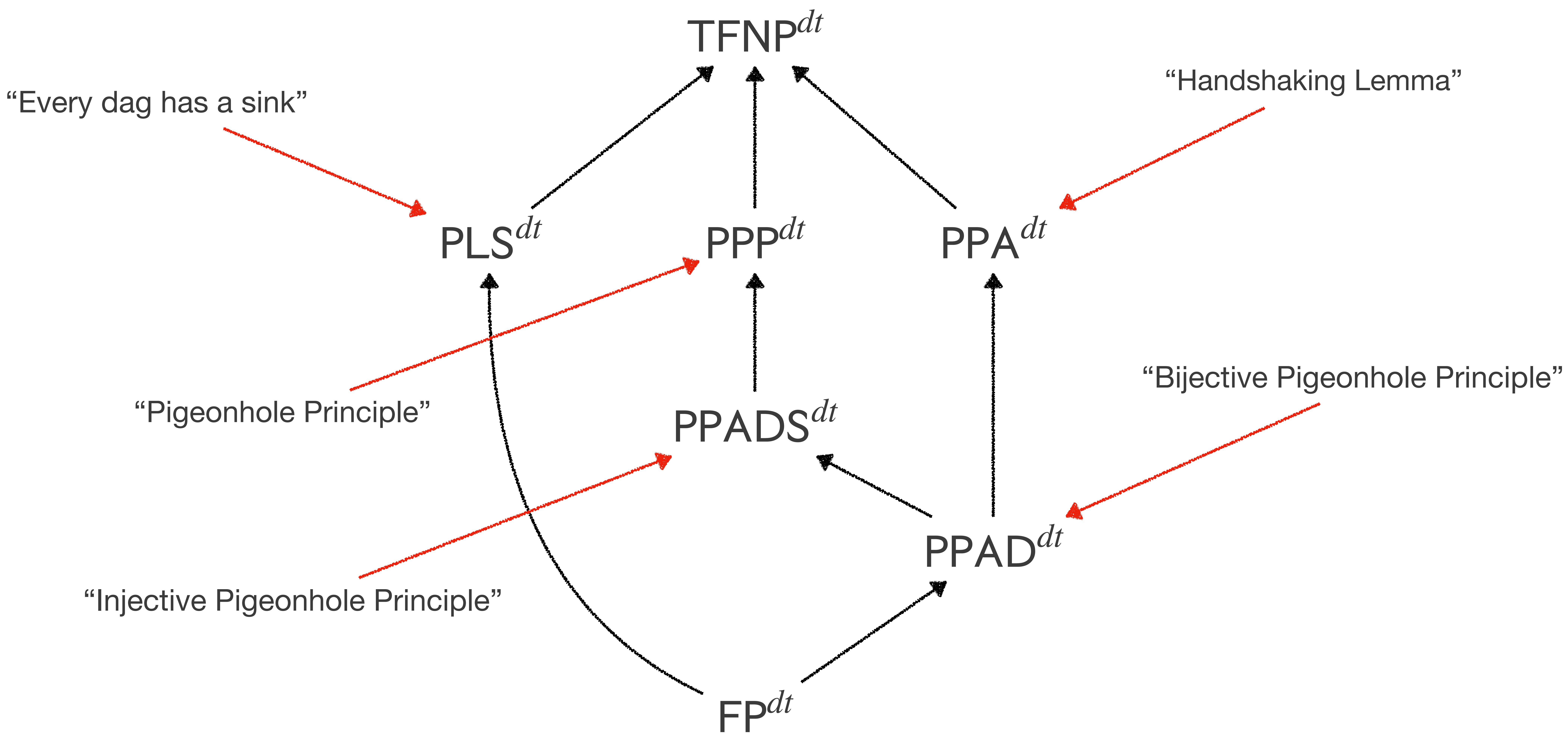
Query PPA

- [BCEIP 98] Observed that if $\mathsf{Search}(F) \in \mathsf{PPA}^{dt}$ then F has a low-degree **Nullstellensatz refutation** over $\mathbb{F}_2$

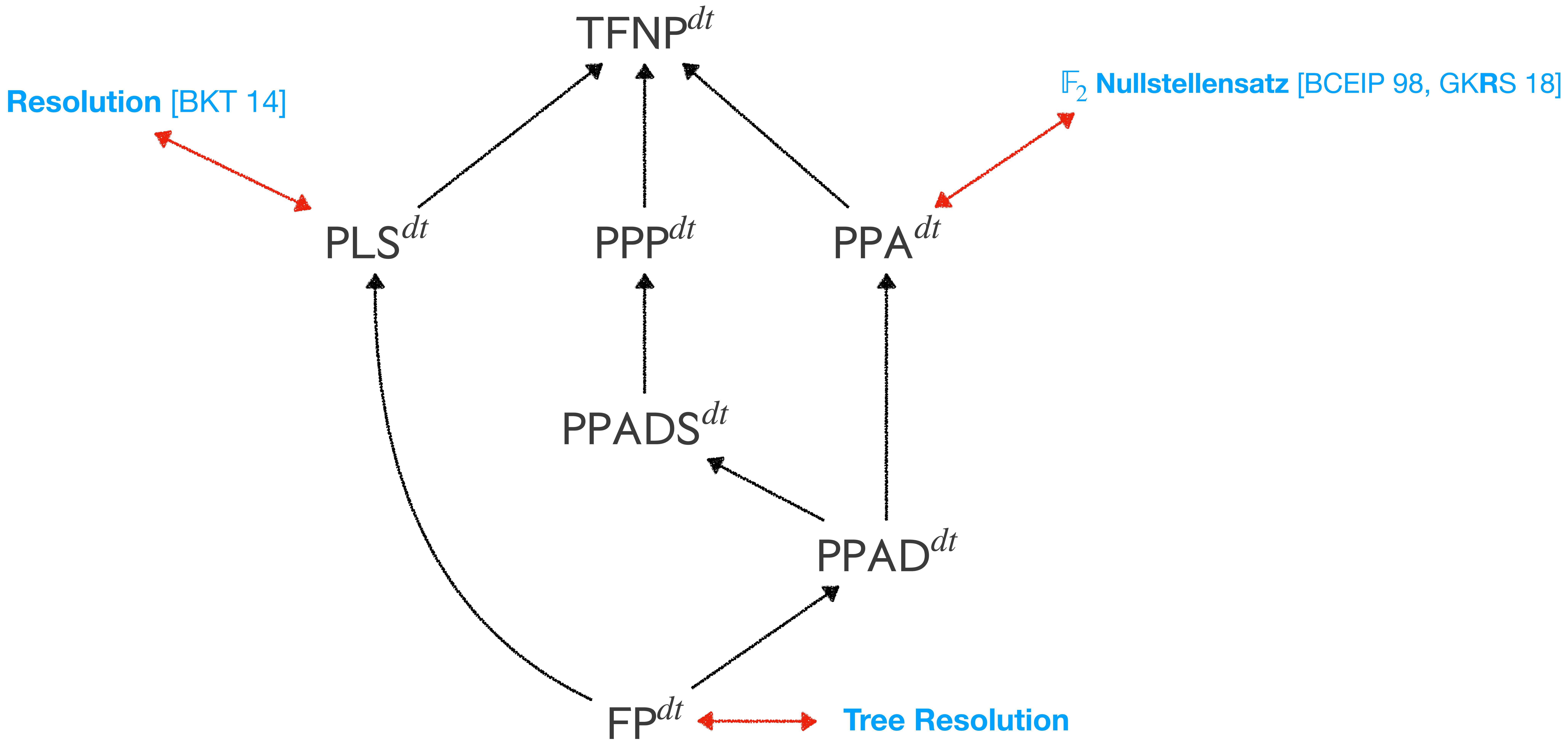
- [GKRS 18] Observed the converse: low-degree Nullstellensatz implies an efficient reduction to PPA

Theorem. [BCEIP 98, GKRS 18]
Let F be an unsatisfiable CNF. There is a size $\leq s$, degree $\leq d$ $\mathbb{F}_2$-Nullstellensatz refutation of F iff $\mathsf{Search}(F)$ can be depth $O(d)$ reduced to PPA on $s^{O(1)}$ vertices.

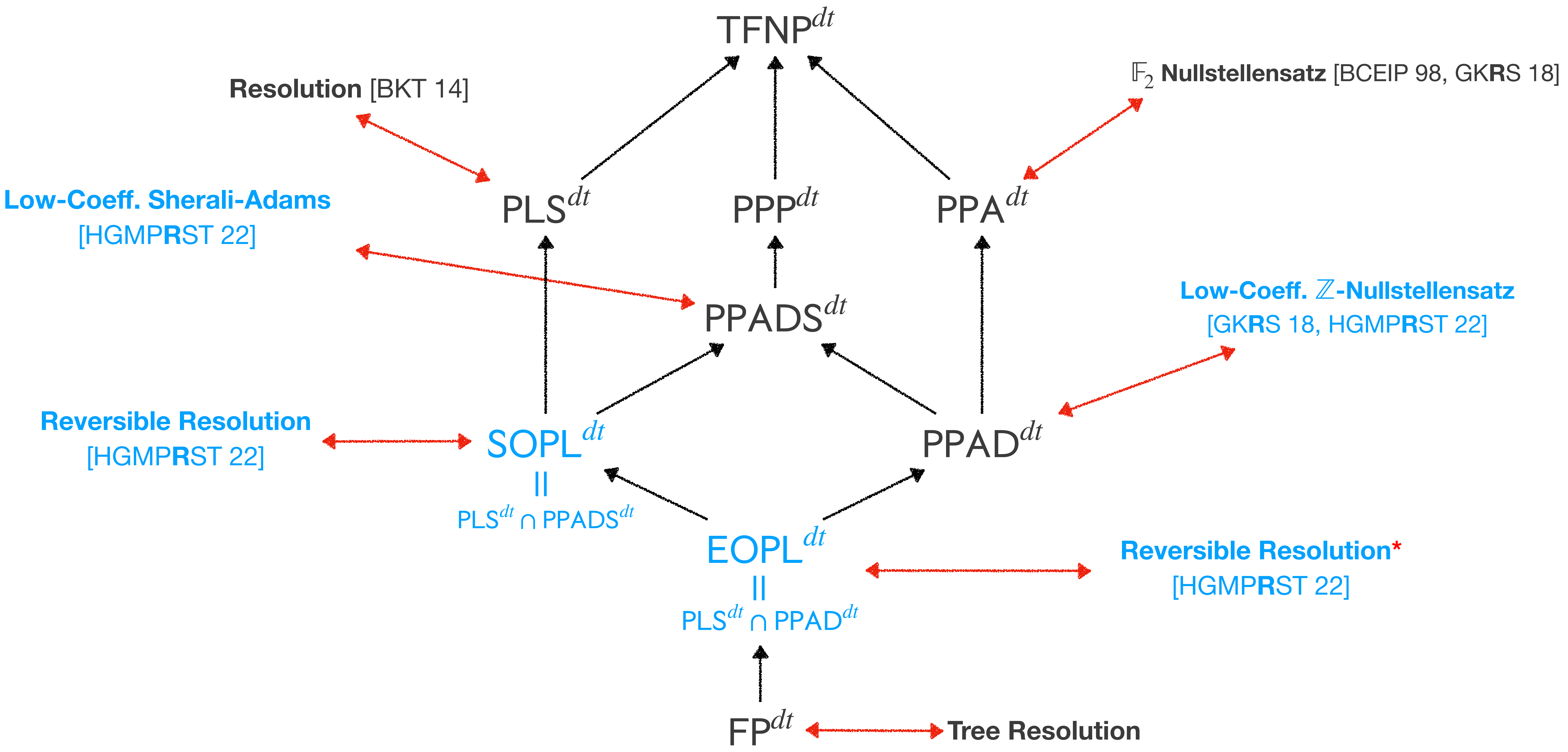
Query TFNP Classes



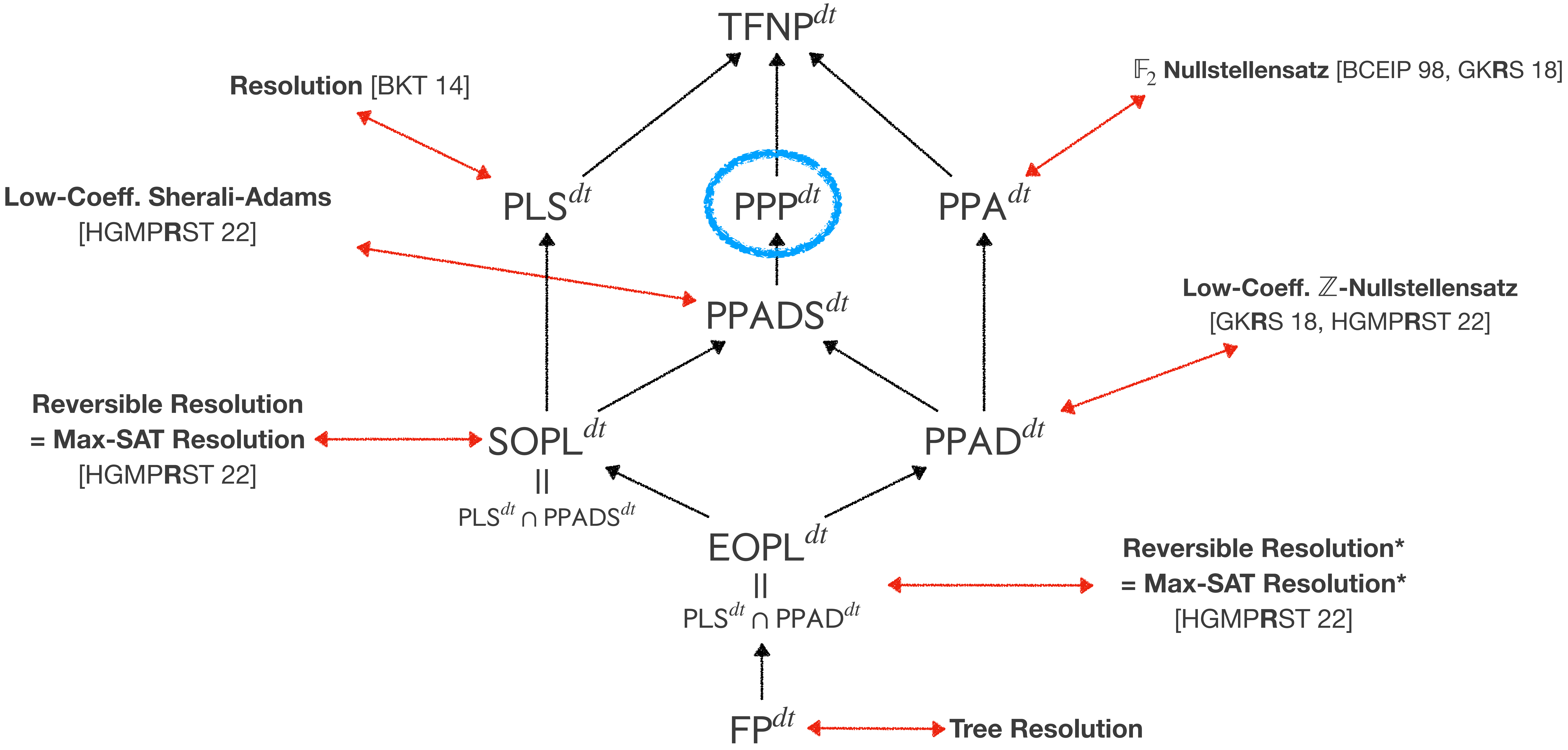
Query TFNP Classes



Query TFNP Classes (New Results)



Query TFNP Classes (New Results)



Reversible Resolution = Max-SAT Resolution

- Single "reversible" resolution rule:

$$\frac{C \vee \ell \quad C \vee \overline{\ell}}{C}$$

- From $C \vee \ell, C \vee \overline{\ell}$ deduce C or *vice-versa*, clauses are **consumed**!

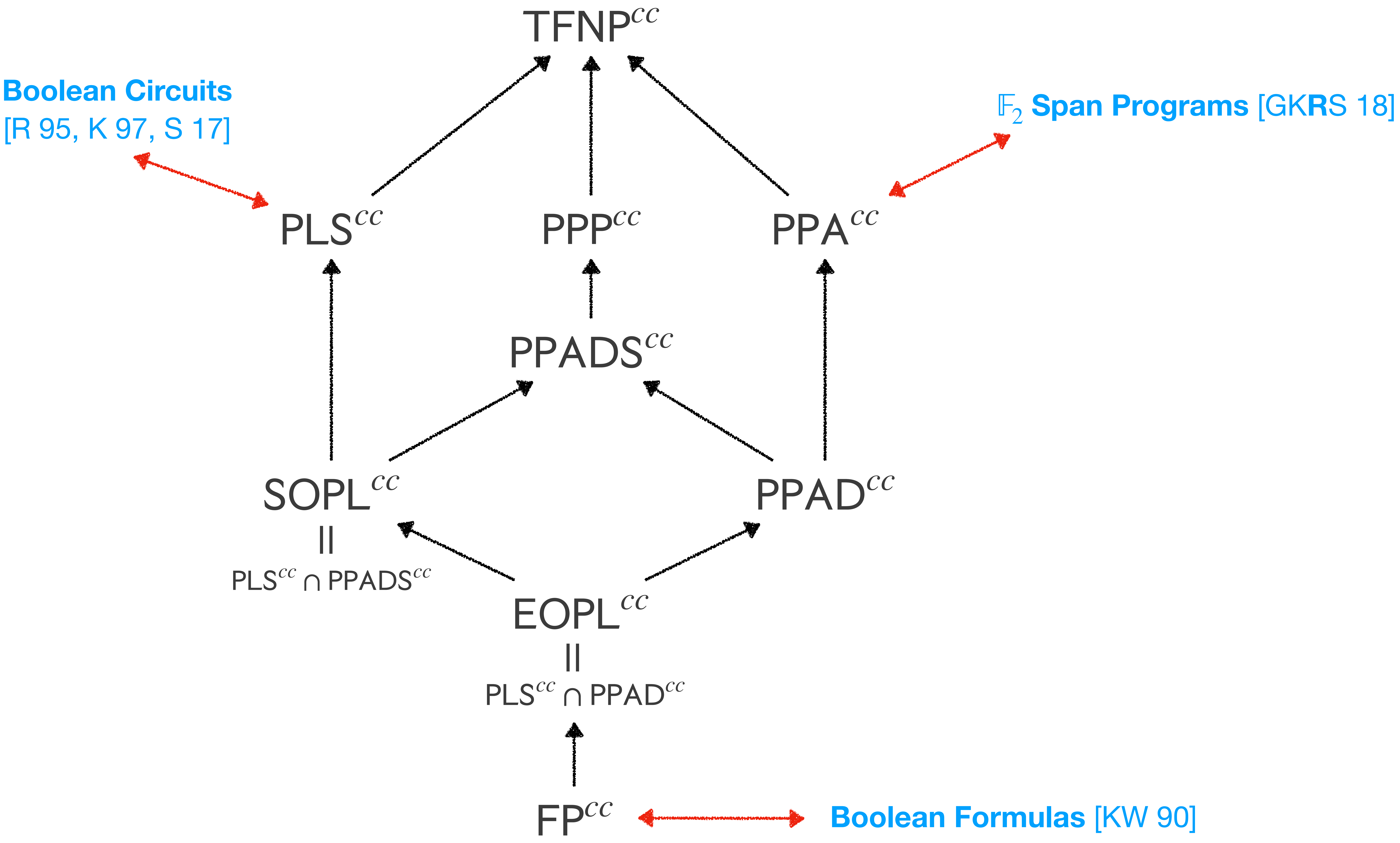
Theorem. [HGMP**R**ST 22]

There is a size-s, width-w Reversible Resolution proof of F

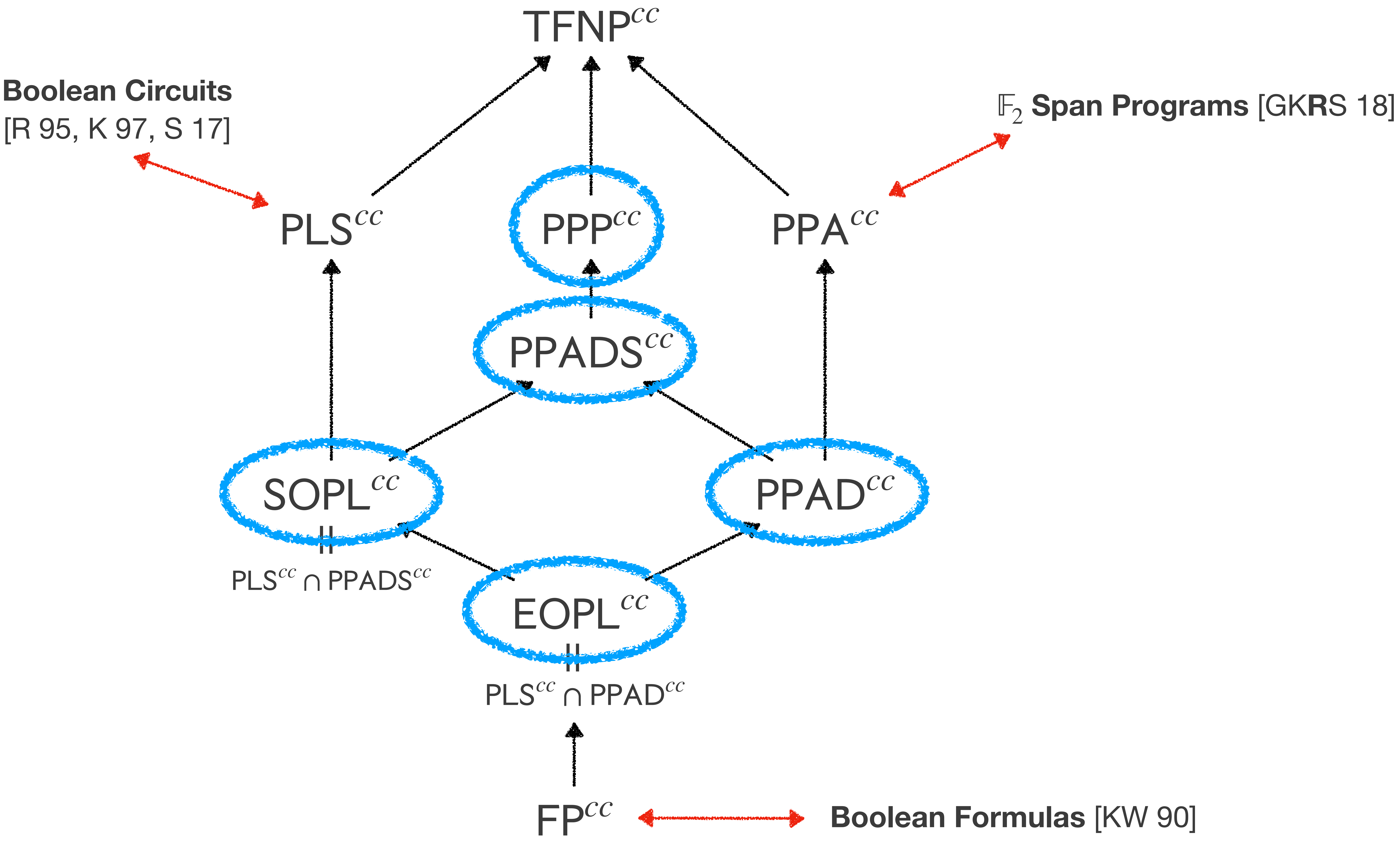
if and only if

there is a size $s^{O(1)}$, width-$O(w)$ Resolution proof **and** a size-$s^{O(1)}$, degree-$O(w)$ low-coefficient Sherali-Adams proof.

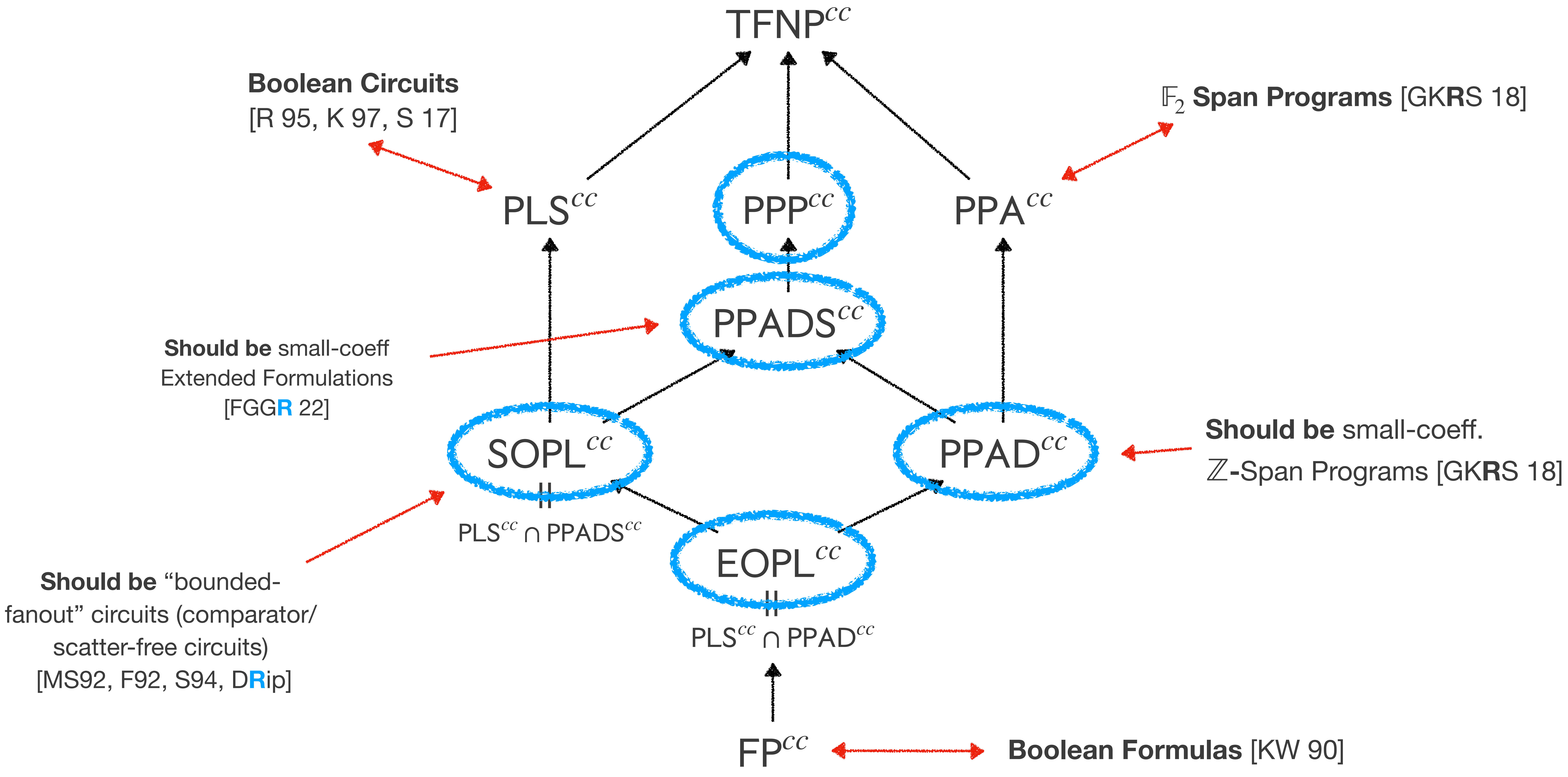
Communication TFNP

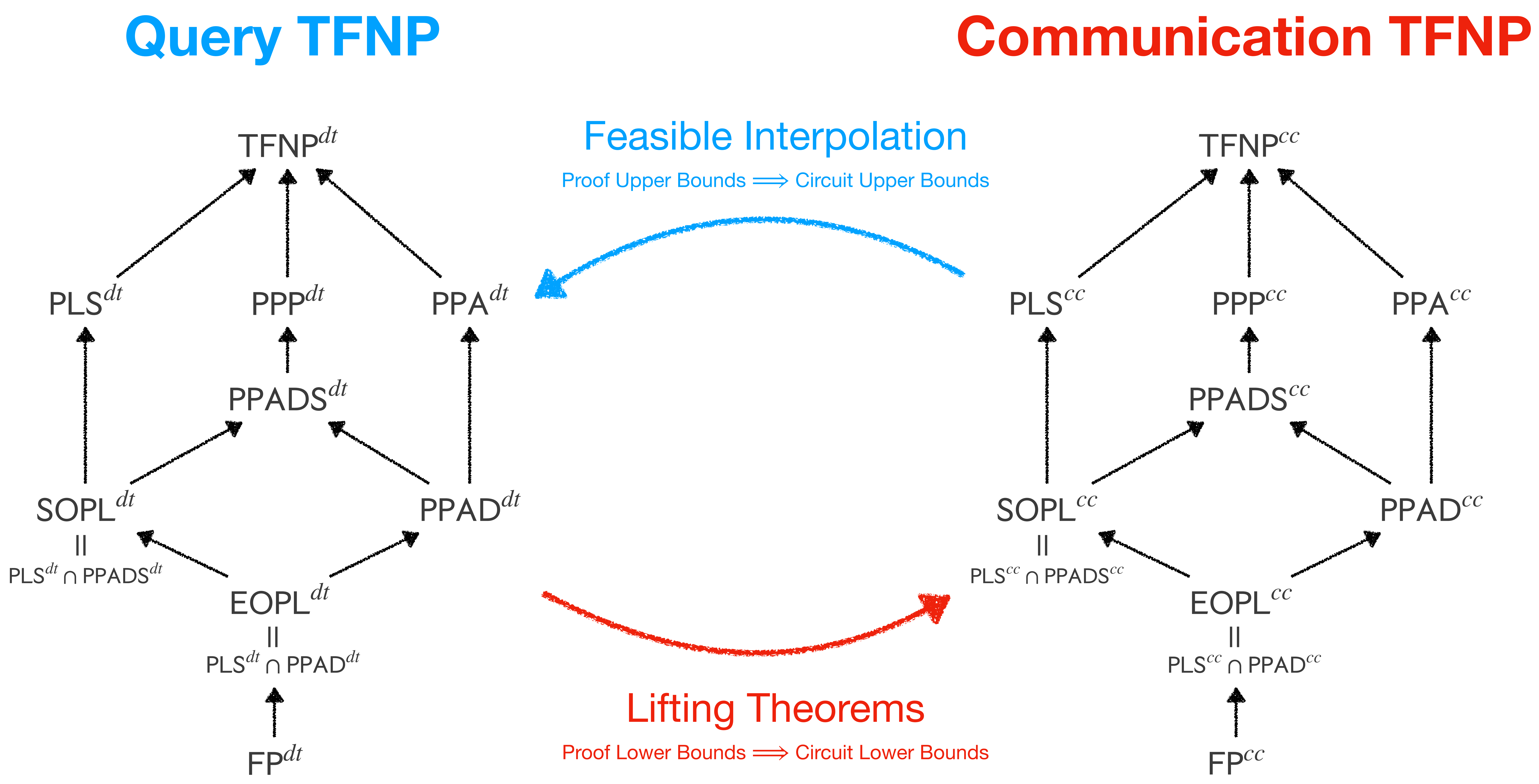


Communication TFNP



Communication TFNP





Query TFNP
Communication TFNP
Feasible Interpolation
Proof Upper Bounds ⟹ Circuit Upper Bounds
Lifting Theorems
Proof Lower Bounds ⟹ Circuit Lower Bounds
TFNP^{dt}
PLS^{dt}
PPP^{dt}
PPA^{dt}
PPADS^{dt}
$\mathrm{SOPL}^{dt} = \mathrm{PLS}^{dt} \cap \mathrm{PPADS}^{dt}$
PPAD^{dt}
$\mathrm{EOPL}^{dt} = \mathrm{PLS}^{dt} \cap \mathrm{PPAD}^{dt}$
FP^{dt}
TFNP^{cc}
PLS^{cc}
PPP^{cc}
PPA^{cc}
PPADS^{cc}
$\mathrm{SOPL}^{cc} = \mathrm{PLS}^{cc} \cap \mathrm{PPADS}^{cc}$
PPAD^{cc}
$\mathrm{EOPL}^{cc} = \mathrm{PLS}^{cc} \cap \mathrm{PPAD}^{cc}$
FP^{cc}

“Feasible Interpolation”

- Many interesting results from **relating** two worlds
- If $\mathcal{S} \subseteq \{0,1\}^n \times O$ is a **query** search problem, let $[n] = X \cup Y$ be variable partition
- Define $\mathcal{S}^{X,Y} \subseteq \{0,1\}^X \times \{0,1\}^Y \times O$ as a **communication** problem, so
  - Alice gets $x \in \{0,1\}^X$, Bob gets $y \in \{0,1\}^Y$, solutions are $\mathcal{S}^{X,Y}(x,y) = \mathcal{S}(xy)$
- Easy to see that for any TFNP class $\mathcal{C}$, $\mathcal{S} \in \mathcal{C}^{dt} \implies \mathcal{S}^{X,Y} \in \mathcal{C}^{cc}$
- Translates **circuit lower bounds** to **proof lower bounds**
  - Closely related to classical **feasible interpolation** results [K97, P97, BPR00,…]
  - Construction underlies Cutting Planes l.bs for random CNFs [FPP**R** 16, HP16]

Lifting Theorems

- **Query-to-communication lifting theorems** give the other direction
- $\mathcal{S} \subseteq \{0,1\}^n \times O$ is a query search problem, $g : X \times Y \to \{0,1\}^n$ is a gadget
- Define $\mathcal{S} \circ g \subseteq X^n \times Y^n \times O$ by $(\mathcal{S} \circ g)(x, y) = \mathcal{S}(g^n(x, y))$
  - Alice gets $x \in X^n$, Bob gets $y \in Y^n$, evaluate $z = g^n(x, y)$ and solve $\mathcal{S}(z)$
- If g "complex" then Alice and Bob's best strategy is to simulate the query strategy

Theorem. [RM 99, GPW 14]
Let $\mathcal{S} \subseteq \{0,1\}^n \times O$ be a search problem, let $\mathrm{Ind}_m : [m] \times \{0,1\}^m \to \{0,1\}$ by $\mathrm{Ind}_m(x, y) = y_x$. If $m = n^{O(1)}$ then

$$\mathrm{FP}^{cc}(\mathcal{S} \circ \mathrm{Ind}_m) = \Theta(\mathrm{FP}^{dt}(\mathcal{S}) \cdot \log m)$$

Lifting Theorems

Proof Complexity Size	Proof Complexity Degree	Circuit Complexity Measure	Gadget	Citation
Tree-Like Resolution Size	Resolution Depth	Monotone Formula Size	Index, Low-Discrepancy	[Folklore, RM99, GPW14, CKFMP19]
Resolution Size	Resolution Width	Monotone Circuit Size	Index	[GGKS17]
Nullstellensatz Monomial Size	Nullstellensatz Degree	Monotone Span Program Size	Any High Rank	[PR18, dRMNPR20]
Sherali-Adams Monomial Size	Sherali-Adams Degree	Linear Extension Complexity	Index, Inner Product*	[GLMW14, CLRS14, KMR17] (Incomplete)
Sums-of-Squares Monomial Size	SOS Degree	Semidefinite Extension Complexity	Index*	[LRS15] (Incomplete)

TFNP Program in Proof and Circuit Complexity

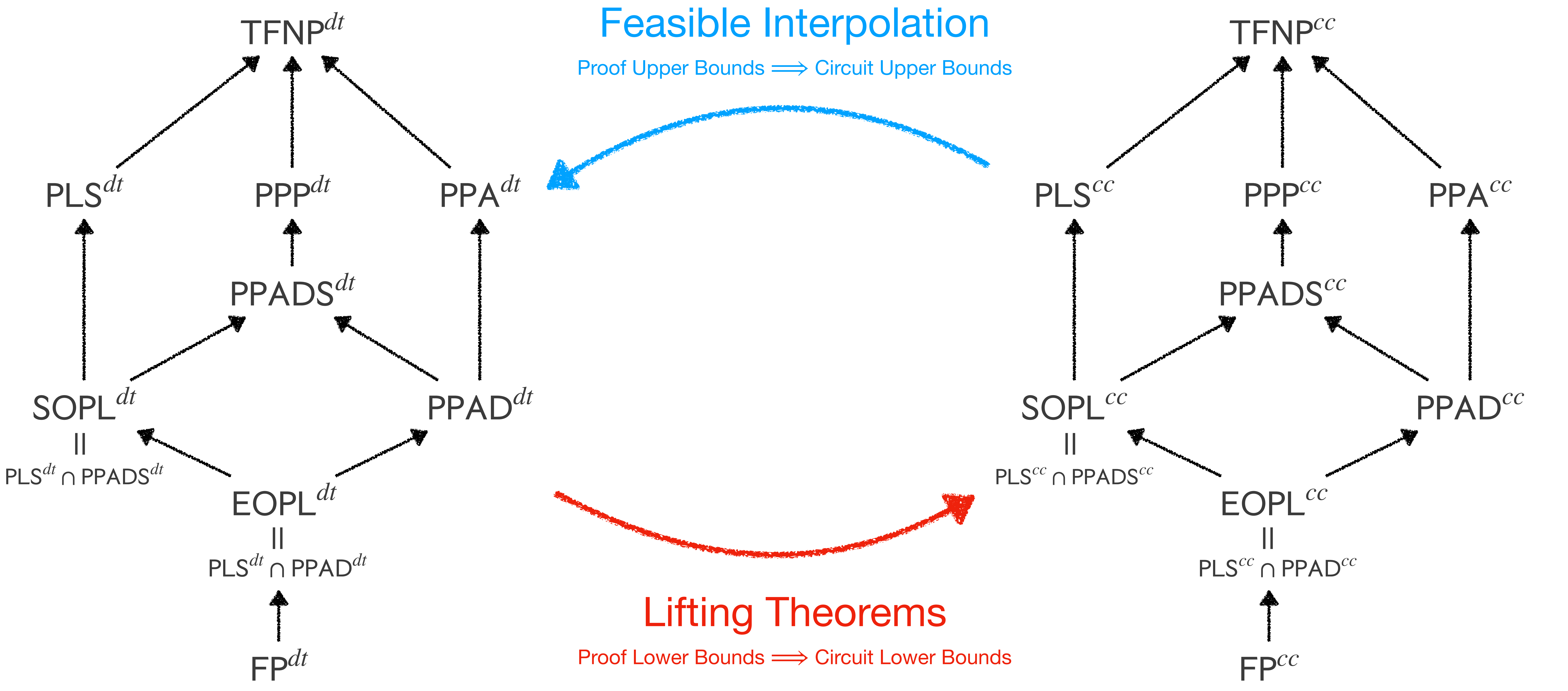
- All in all, this suggests a research program!
- Use TFNP classes to characterize circuit and proof classes.
- Relate these classes by **feasible interpolation** and **lifting theorems**
- Use intuition from one setting to prove results in the other setting.
  - **Many** TFNP classes are not characterized in either setting.
- Intersection theorems are particularly interesting!
  - Reversible Resolution = Resolution $\cap$ Sherali-Adams* [HGMPRST 22]

Open Problems

- What TFNP problem captures Sums-of-Squares?
- What about Cutting Planes, Lovasz-Shrijver? (These are somehow different.)
- Characterize more circuit and proof classes using TFNP classes.
- Can this approach (communication and query complexity) say anything novel about very powerful proof systems?
- What about non-monotone complexity? Can anything be said?

Query TFNP

Communication TFNP



Thanks for Listening!