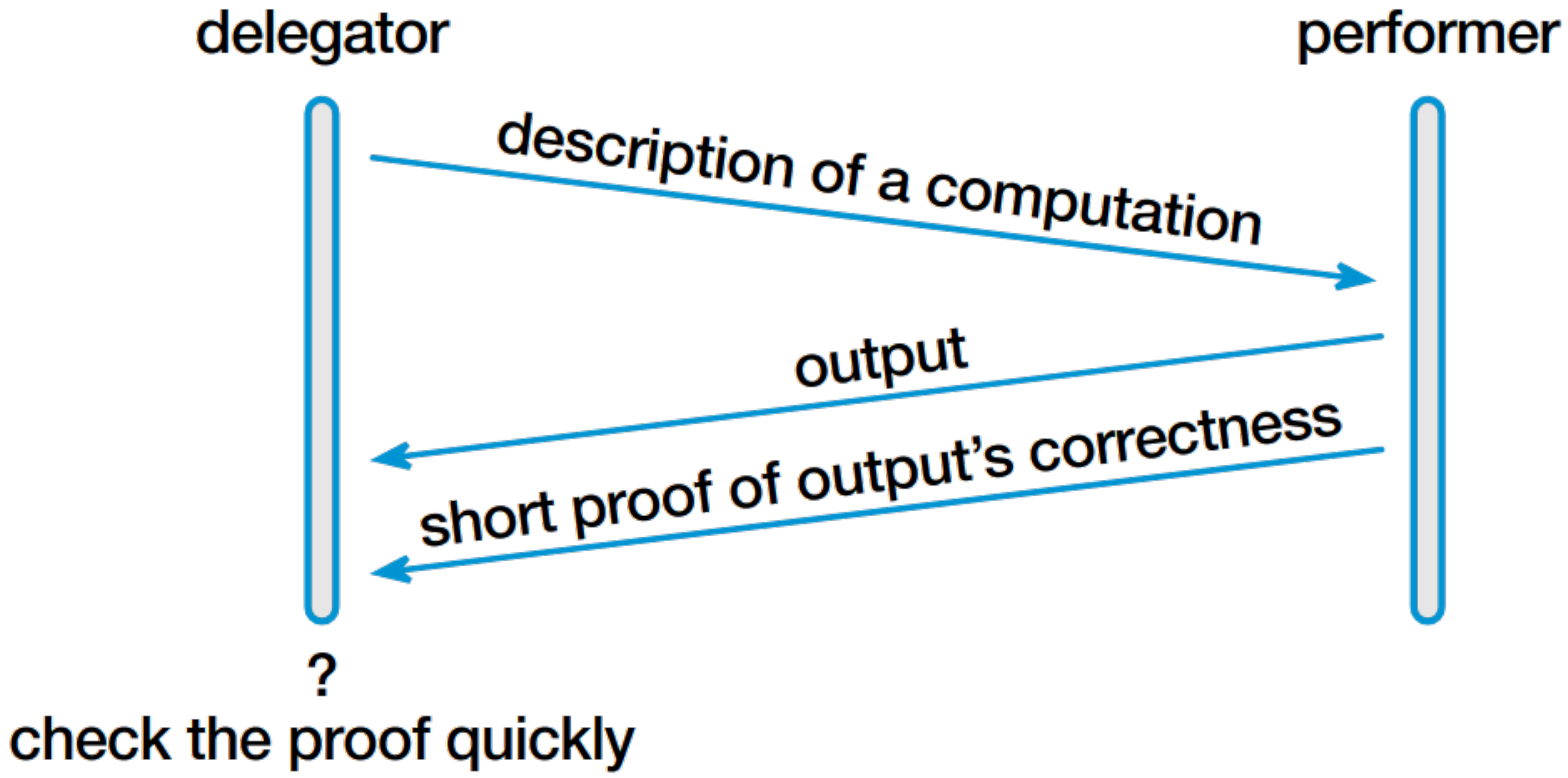
Interactive Proof

and more

Soowon Jeong, 9 Mar 2023

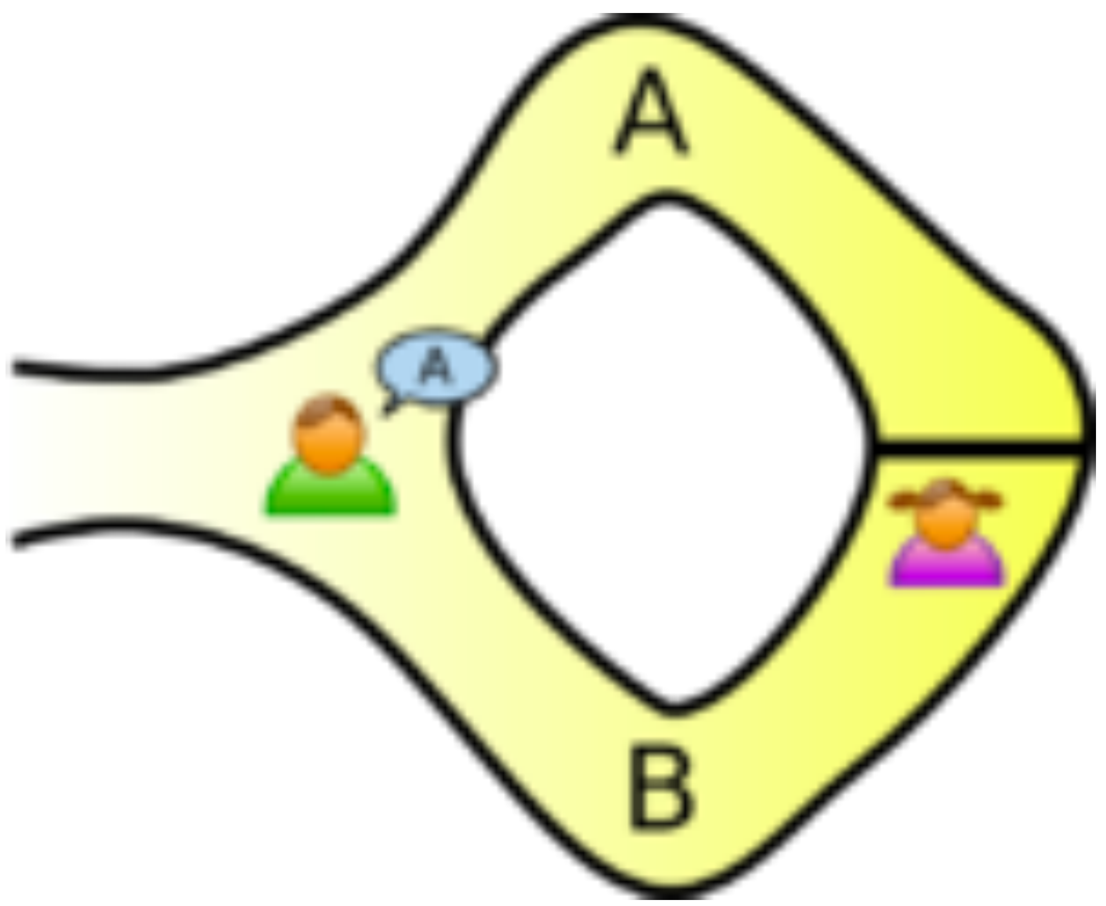
Proof

- 수학 증명 (Mathematical proof) ≠ 암호학 증명 (Cryptographic proof)
- 회로 (circuit) → 계산 과정
- 검증 가능한 계산 (Verifiable computing)



Interactive Proof

- 1980s
- IP가 NP보다 빠름이 증명됨
- IP = PSPACE [Sha92]
- P $\subseteq$ NP $\subseteq$ PSPACE ($\subsetneq$)



Interactive Proof

- 함수 f 가 주어졌을 때, $\{0,1\}^n \to \mathscr{R}$
- k-message interactive proof system
- 증명자 $\mathscr{P}$, 검증자 $\mathscr{V}$ 에게 같은 $x \in \{0,1\}^n$을 제공
- $\mathscr{P}$는 $f(x)$와 같다고 주장하는 y를 보냄
- 한 차례식 메시지 m_i를 주고받음 (둘 다 다음 메시지를 계산할 수 있다면)
- $(m_1, \ldots, m_k) \leftarrow$ transcript

Interactive Proof

- Decision problem
- $f : \{0,1\}^n \to \{0,1\}$
- $f(x)$ 를 질문으로 만든다면 “$f(x) = 1$?”
- $L \subseteq \{0,1\}^n$
- Completeness: For any $x \in L$, there is some prover strategy that will cause the verifier to accept with high probability.
- Soundness: For any $x \notin L$, then for *every* prover strategy, the verifier rejects with high probability.

Sum-check

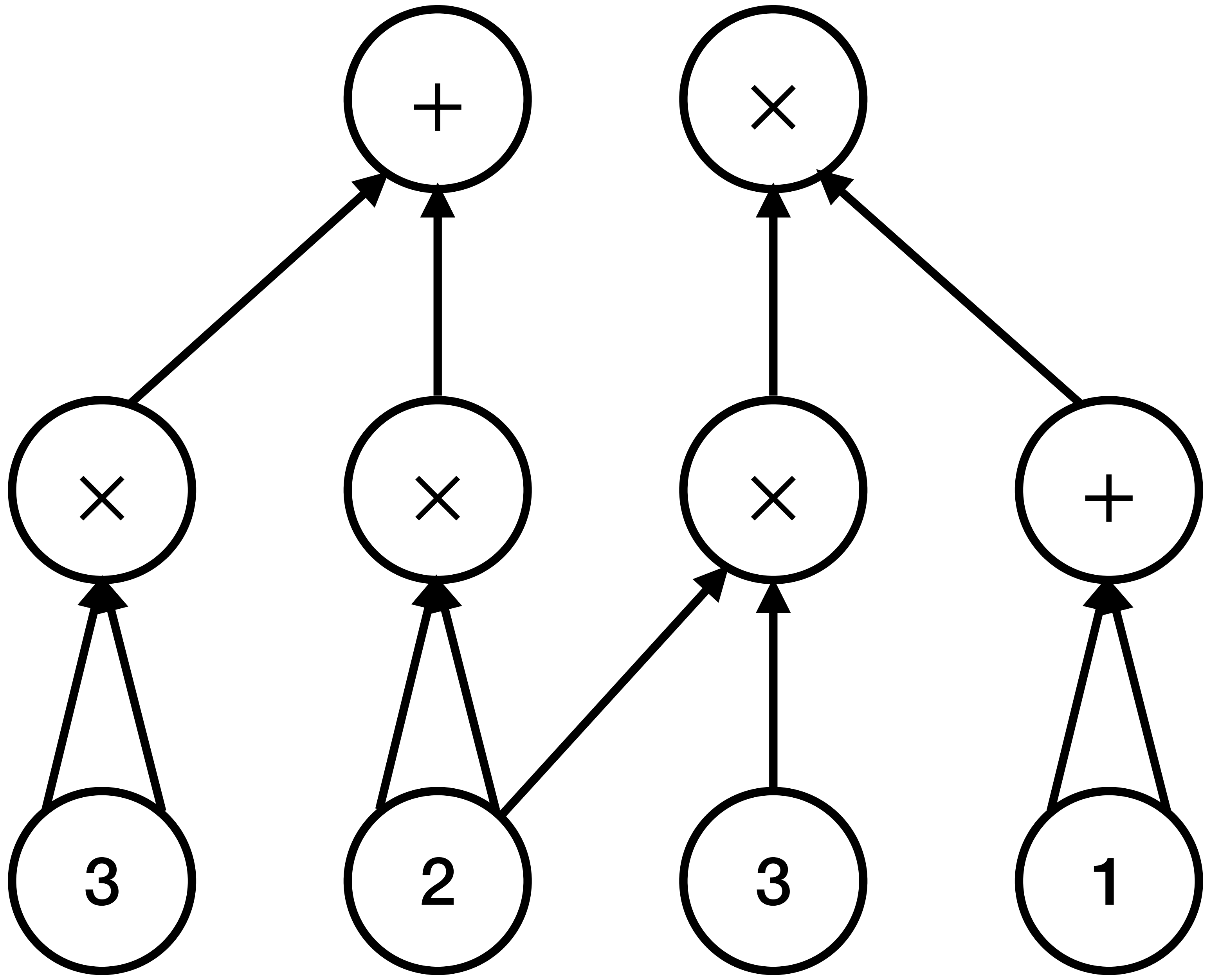
- For v-variate polynomial g, $H = \sum_{(b_1,\cdots,b_v)\in\{0,1\}^v} g(b_1, \cdots, b_v)$
- Prover $\mathscr{P}$ sends a value H and the univariate polynomial $g_1(X_1)$
  $g_1(X_1) = \sum_{(x_2,\cdots,x_v)\in\{0,1\}^v} g(X_1, x_2, \cdots, x_v)$.
  Verifier $\mathscr{V}$ checks that $H = g_1(0) + g_1(1)$
- For i in $1 < i < v$, $\mathscr{P}$ sends to $\mathscr{V}$ a univariate polynomial $g_i(X_i)$ claimed to equal
  $g_i(X_i) = \sum_{(x_{i+1},\cdots,x_v)\in\{0,1\}^{v-i}} g(r_1, \cdots, r_{i-1}, X_i, x_{i+1}, \cdots, x_v)$
  $\mathscr{V}$ checks that $g_{i-1}(r_{i-1}) = g_i(0) + g_i(1)$
  $\mathscr{V}$ chooses a random value $r_i \in \mathbb{F}$ and sends it to $\mathscr{P}$

Sum-check

- Final check
  - $\mathscr{V}$ chooses a random element r_v and evaluates $g(r_1, \ldots, r_v)$
  - checks that $g_v(r_v) = g(r_1, \ldots, r_v)$

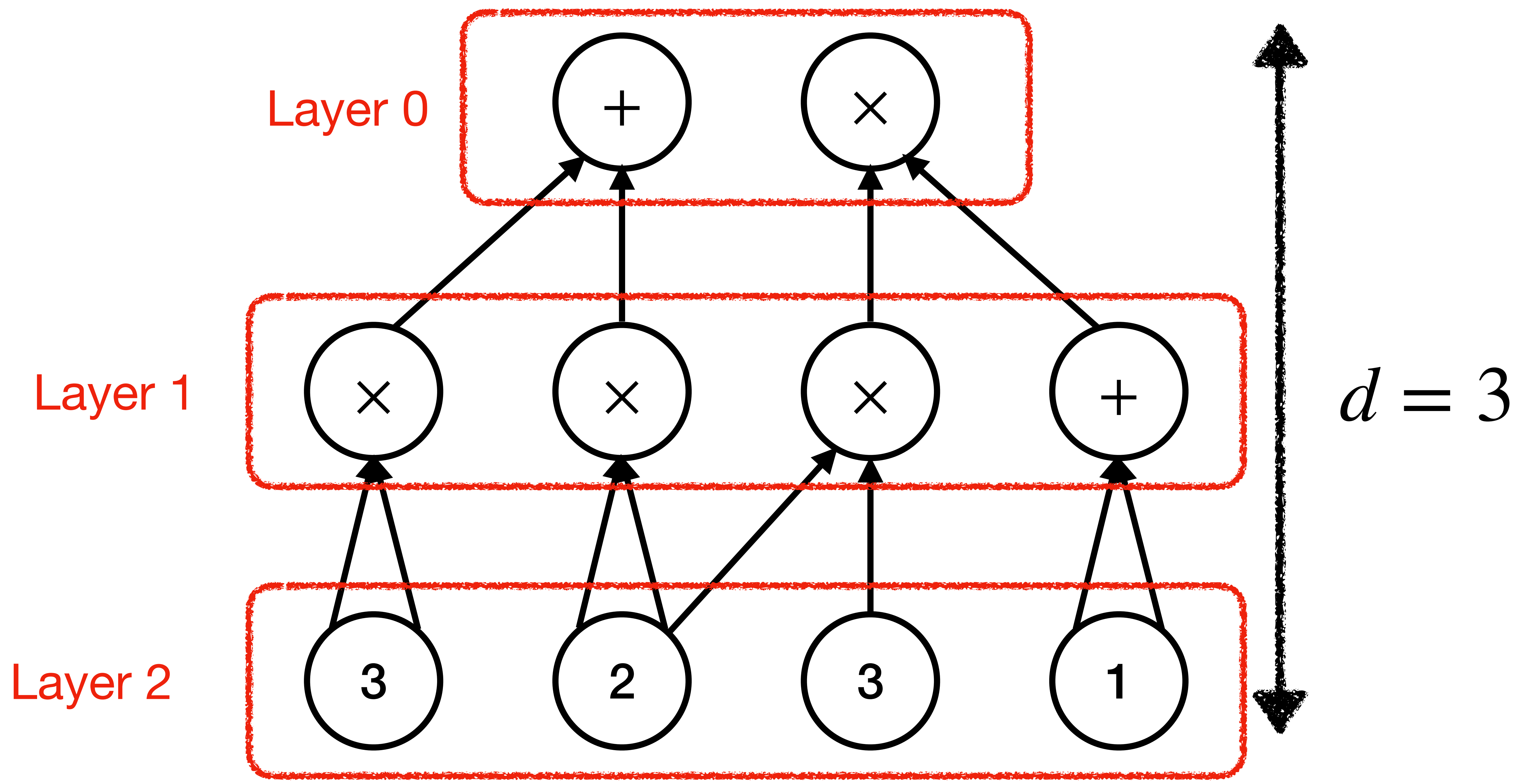
GKR

- Layered arithmetic circuit
- 제일 끝에 있는 gate의 값이 output



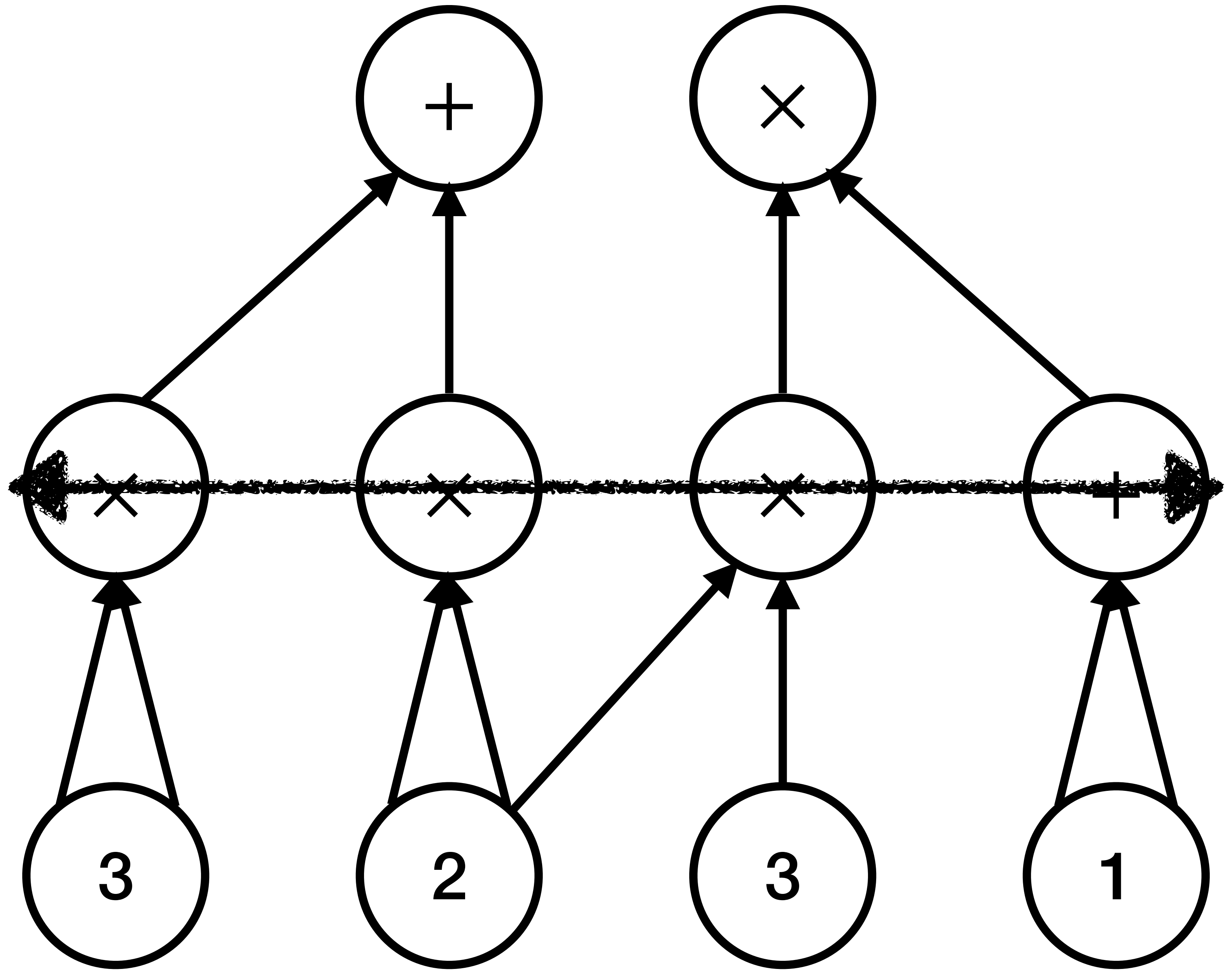
Notation in GKR

- d : GKR circuit의 깊이



Notation in GKR

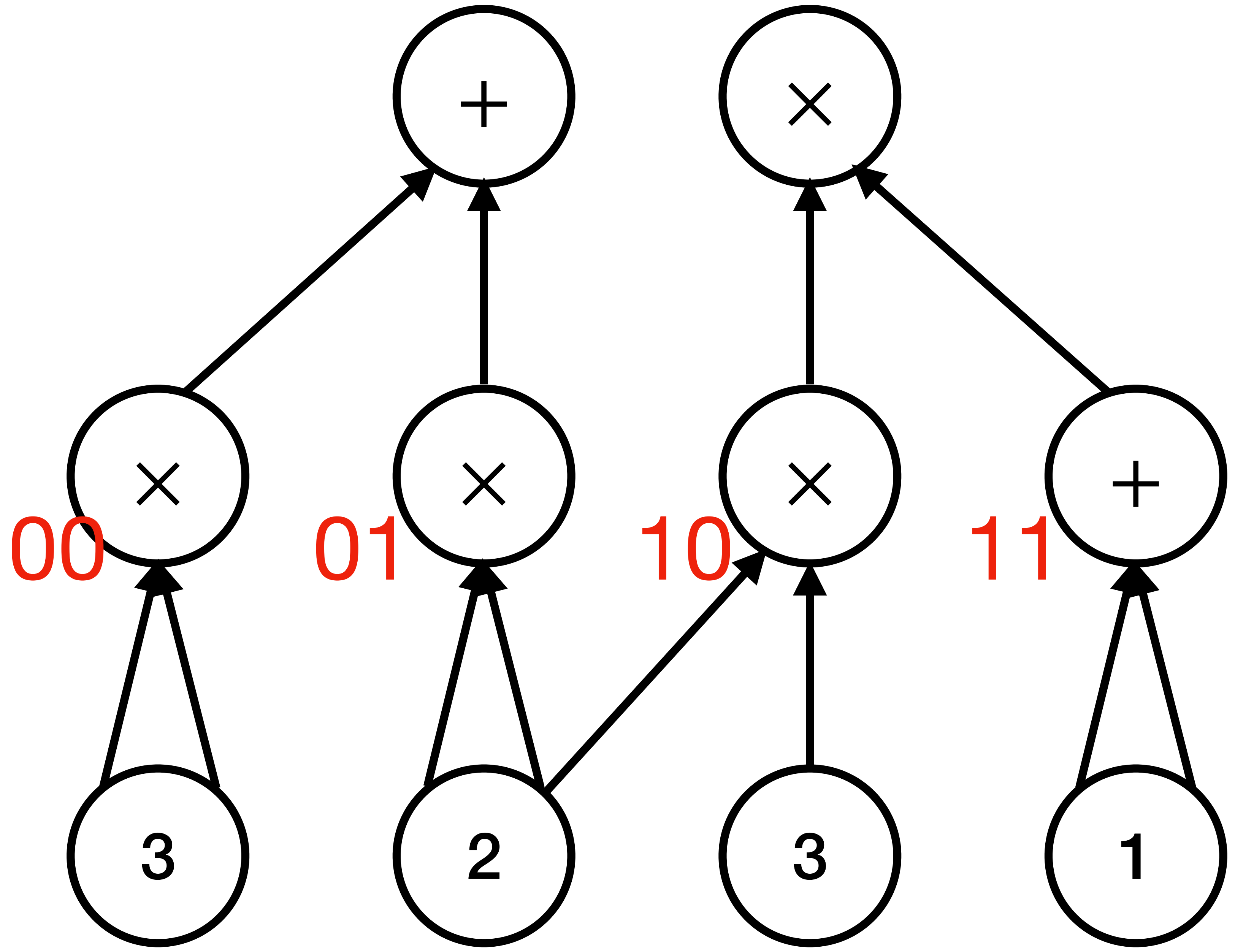
- S_i : # of gates at layer i. S_i is a power of 2 ($S_i = 2^{k_i}$).



$S_1 = 4,\ k_1 = 2$

Notation in GKR

- 왼쪽부터 번호를 붙임. let $W_i : \{0,1\}^{k_i} \rightarrow \mathbb{F}$. MLE of W_i is $\widetilde{W}_i$



$W_1(0,0) = 9,\ W_1(0,1) = 4,\ W_1(1,0) = 6,\ W_1(1,1) = 2$

Multilinear extension

$$f : \{0,1\}^2 \rightarrow \mathbb{F}$$

1	2
8	10

Multilinear extension

$$\tilde{f} : \mathbb{F}^2 \to \mathbb{F}$$

1	2	3	4	5	6
8	10	12	14	16	18
15	18	21	24	27	30
22	26	30	34	38	42
29	34	39	44	49	56

Multilinear extension

$$\tilde{f}(x_1, x_2) = (1 - x_1)(1 - x_2) + 2(1 - x_1)x_2 + 8x_1(1 - x_2) + 10x_1x_2$$

1	2	3	4	5	6
8	10	12	14	16	18
15	18	21	24	27	30
22	26	30	34	38	42
29	34	39	44	49	56

Multilinear extension

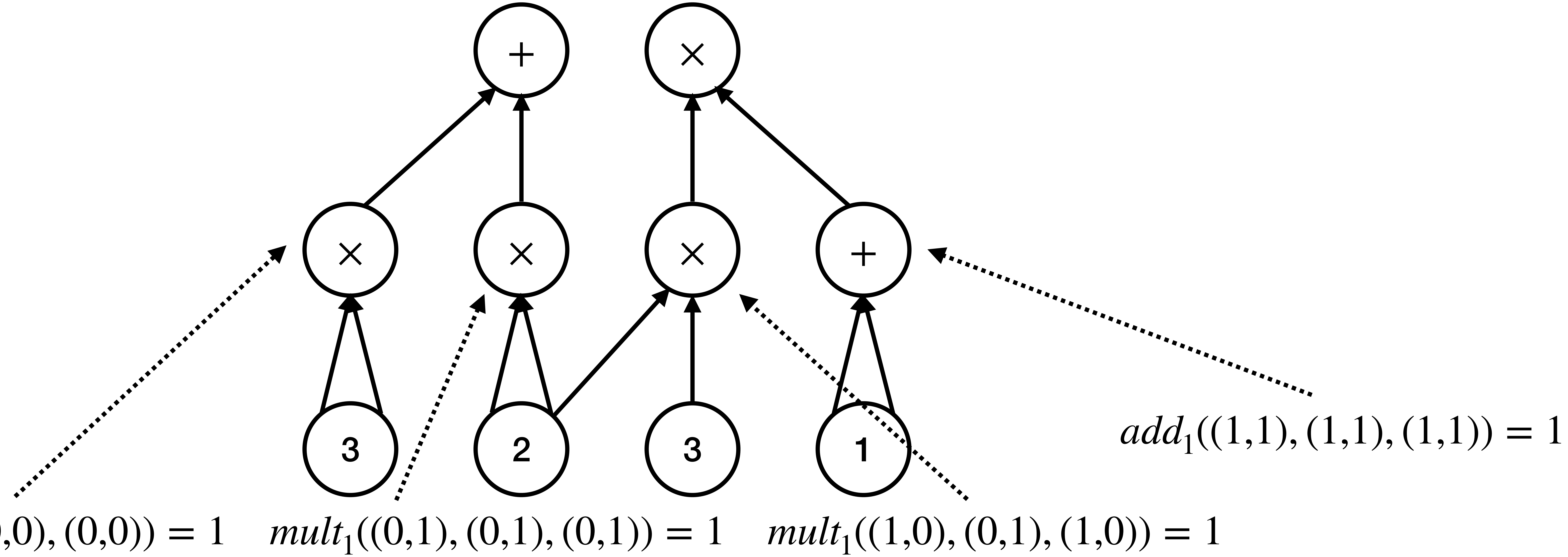
- Lagrange interpolation
- $$\tilde{\delta}_w(r) = \prod_{i=1}^{\ell} (r_i w_i + (1 - r_i)(1 - w_i))$$
    - multilinear Lagrange basis polynomial
- $$\tilde{f}(r) = \sum_{w \in \{0,1\}^{\ell}} f(w) \cdot \tilde{\delta}_w(r)$$

Notation in GKR

- $add_i, mult_i : \{0,1\}^{k_i + 2k_{i+1}} \rightarrow \{0,1\}$

  만약 a가 덧셈 gate라면, $add_i(a,b,c) = \begin{cases} 1 & \text{if } (b,c) = (\text{in}_1(a), \text{in}_2(a)) \\ 0 & \text{otherwise} \end{cases}$

  (곱셈도 동일)



Notation in GKR

- $f_{r_i}^{(i)}(b, c) = \widetilde{add}_i(r_i, b, c)(\widetilde{W}_{i+1}(b) + \widetilde{W}_{i+1}(c)) + \widetilde{mult}_i(r_i, b, c)(\widetilde{W}_{i+1}(b) \cdot \widetilde{W}_{i+1}(c))$
- $\widetilde{W}_i(r_i) = \sum_{b,c \in \{0,1\}^{k_{i+1}}} f_{r_i}^{(i)}(b, c) = m_i \ \ (\widetilde{W}_i : \mathbb{F}^{k_i} \to \mathbb{F})$

GKR

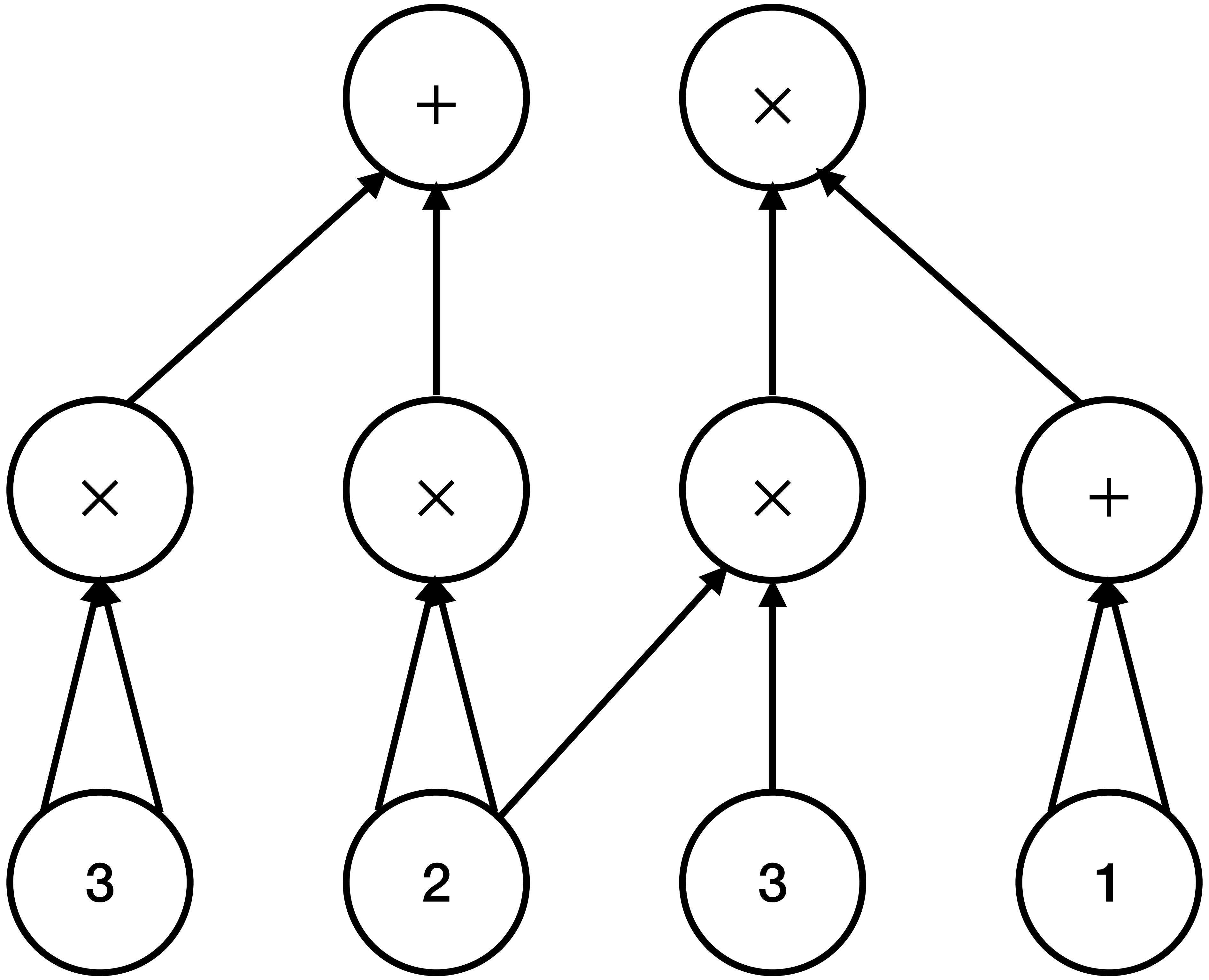
- Prover $\mathscr{P}$는 먼저 W_0와 동일하다고 주장하는 $D : \{0,1\}^{k_0} \to \mathbb{F}$라는 함수를 보낸다.

- Verifier $\mathscr{V}$는 무작위 field element $r_0 \in \mathbb{F}$를 고르고$\tilde{D}(r_0)$ 를 계산한 뒤 m_0에 넣는다. ($m_0 \leftarrow \tilde{D}(r_0)$) 남은 프로토콜은 $m_0 = \widetilde{W_0}(r_0)$인지 확인하는 과정이라고 볼 수 있다.

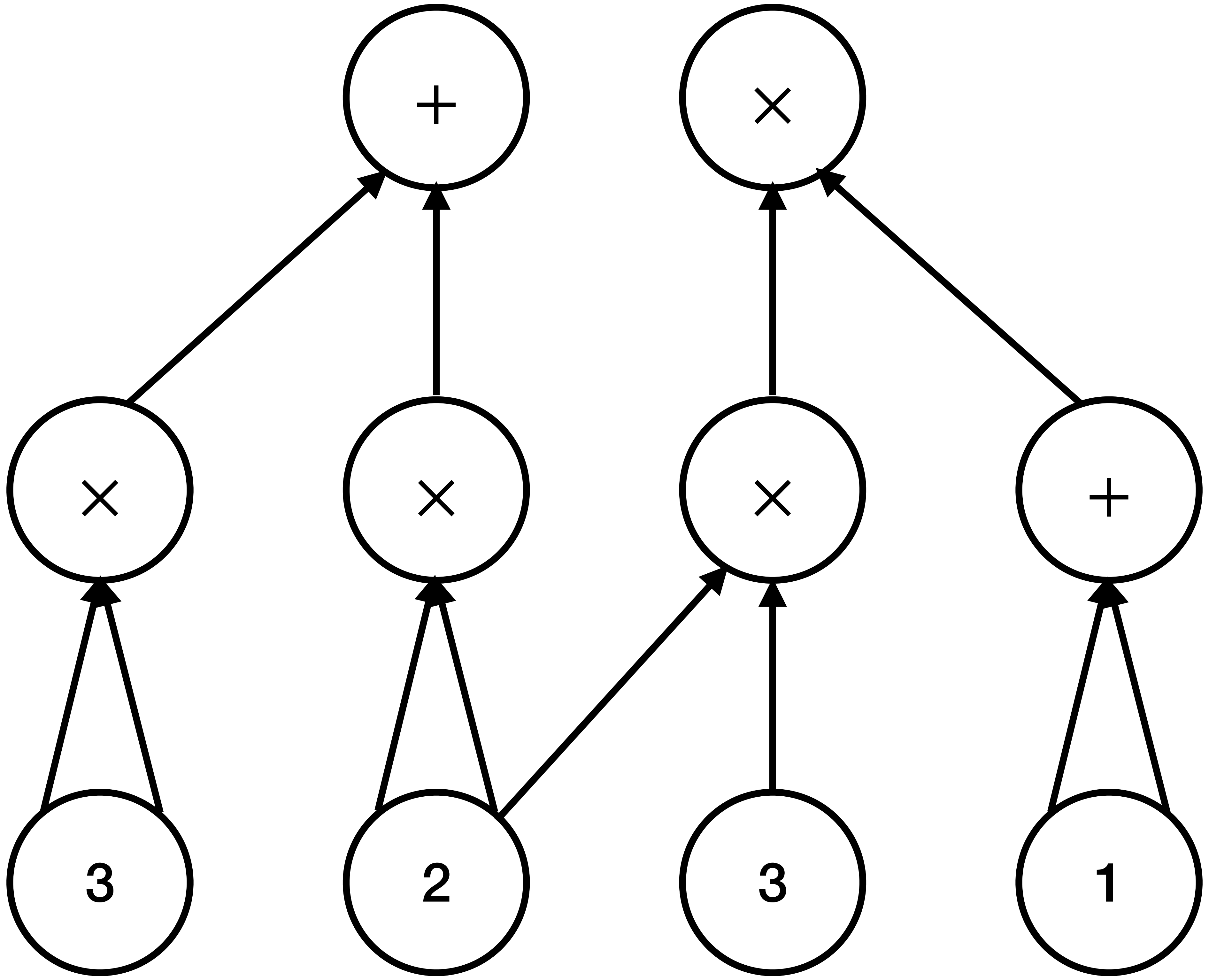
GKR

- circuit depth d에 대해 $i = 0,\ 1, \ldots, d-1$
- $\mathscr{P}$가 $f_{r_i}^{(i)}(b, c) = \tilde{add}_i(r_i, b, c)(\tilde{W_{i+1}}(b) + \tilde{W_{i+1}}(c)) + \tilde{mult}_i(r_i, b, c)(\tilde{W_{i+1}}(b) \cdot \tilde{W_{i+1}}(c))$를 정의한다.
   - $\displaystyle\sum_{b,c \in \{0,1\}^{k_{i+1}}} f_{r_i}^{(i)}(b, c) = m_i$임을 $\mathscr{V}$ 에게 보낸다.
   - $\mathscr{V}$ 는 sumcheck로 위 식을 확인하고 $(b^*, c^*) \in \mathbb{F}^{k_{i+1}} \times \mathbb{F}^{k_{i+1}}$인 무작위 값을 고른다.
   - $l(0) = b^*,\ l(1) = c^*$인 직선이 있다고 할 때, $\mathscr{P}$는 $\widetilde{W}_{i+1} \circ l$인 일변수 다항식 q를 보낸다.
   - $\mathscr{V}$ 는 q를 이용해 $q(0), q(1)$을 구할 수 있다. 이를 이용해$\widetilde{W}_{i+1}(b^*),\ \widetilde{W}_{i+1}(c^*)$를 식에서 대치할 수 있다.
   - $\mathscr{V}$ 는 새로운 무작위 값 $r^* \in \mathbb{F}$ 를 골라 $r_{i+1} = l(r^*)$로 두고 $m_{i+1} \leftarrow q(r^*)$를 넣는다.
- 위 과정이 끝나면 $\mathscr{V}$ 는 $m_d = \tilde{W}_d(r_d)$를 직접 확인할 수 있다.

GKR example

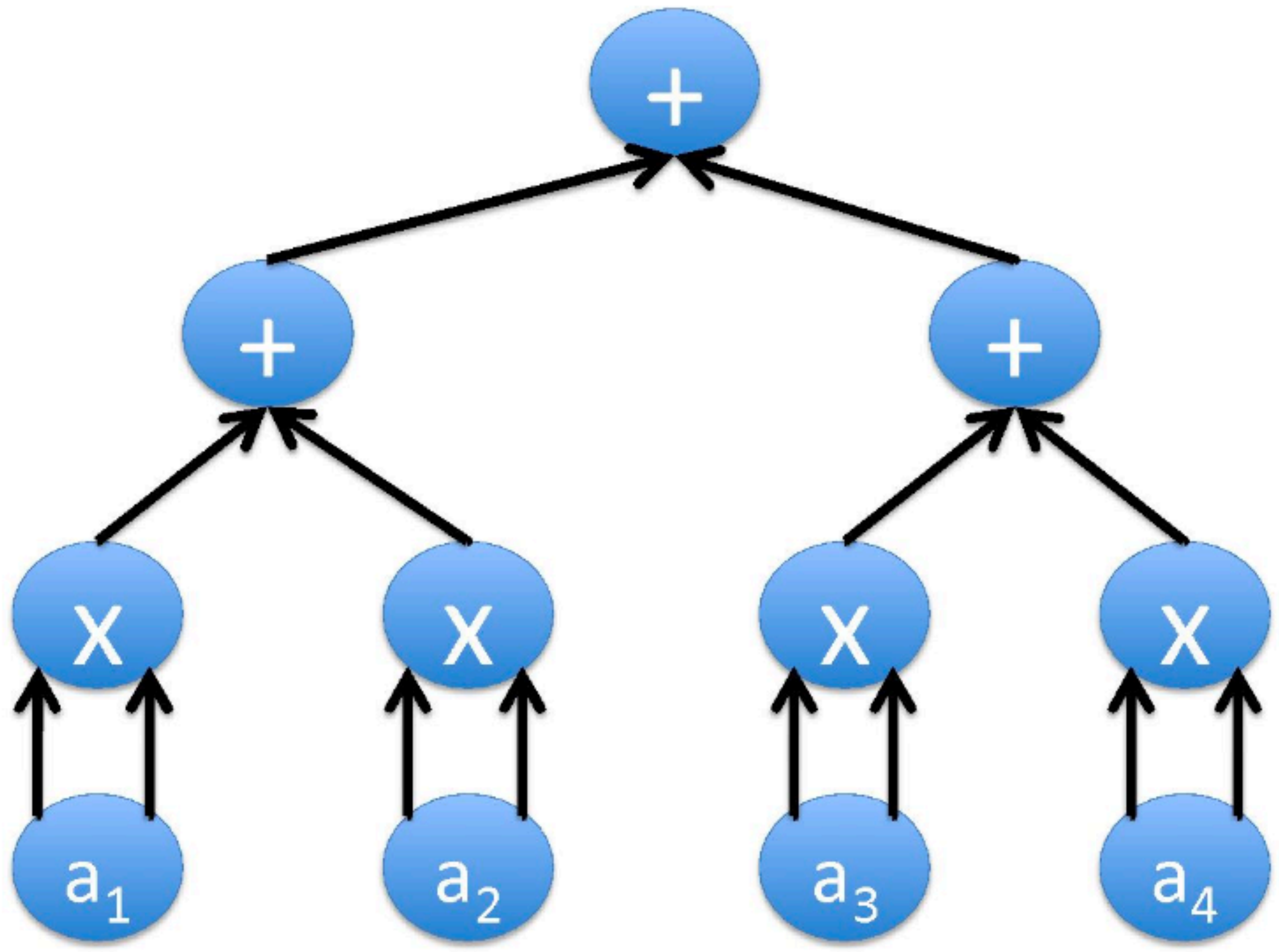


GKR example

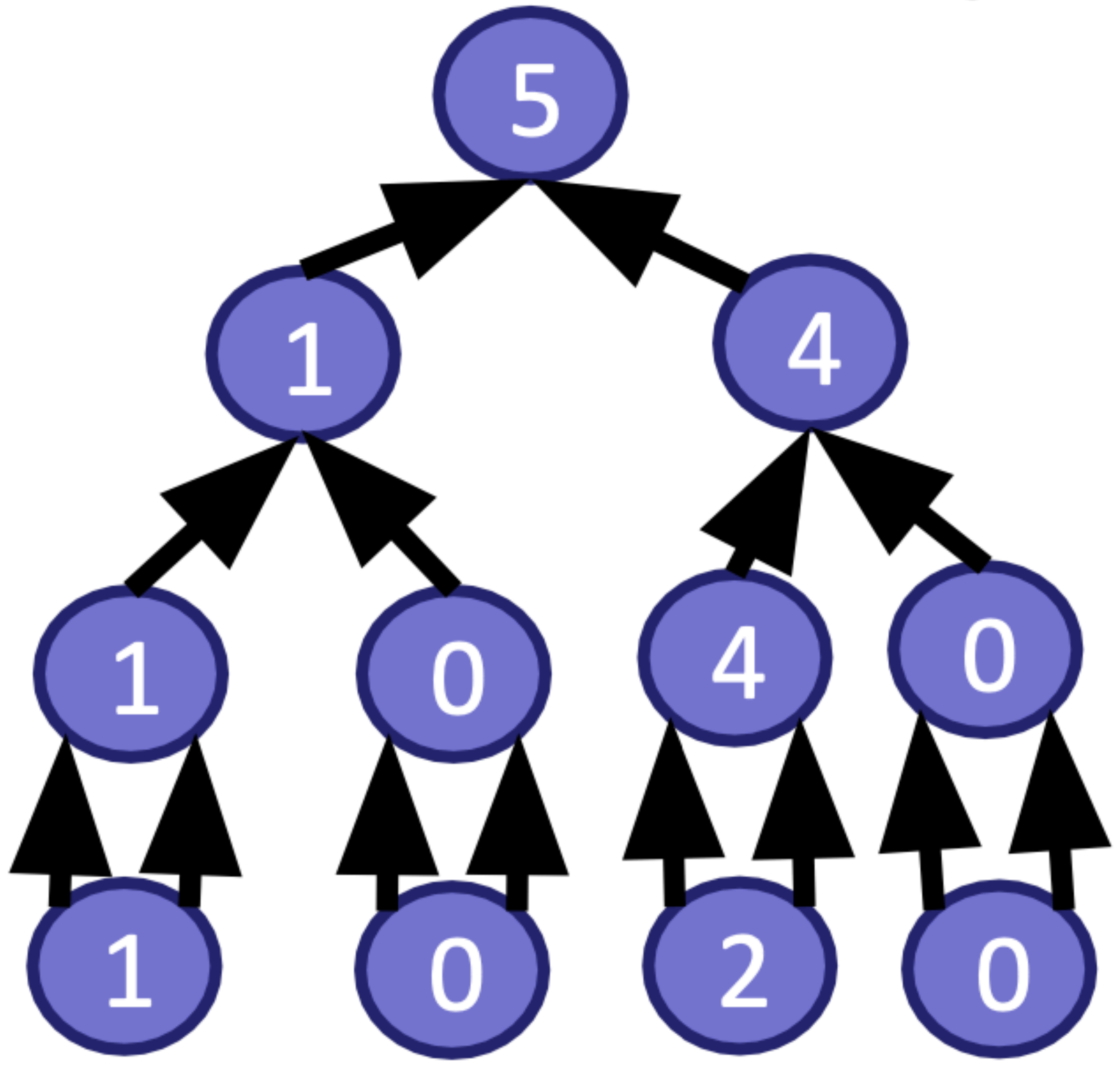


SNARKs for circuit-satisfiability

- Transcript T for C

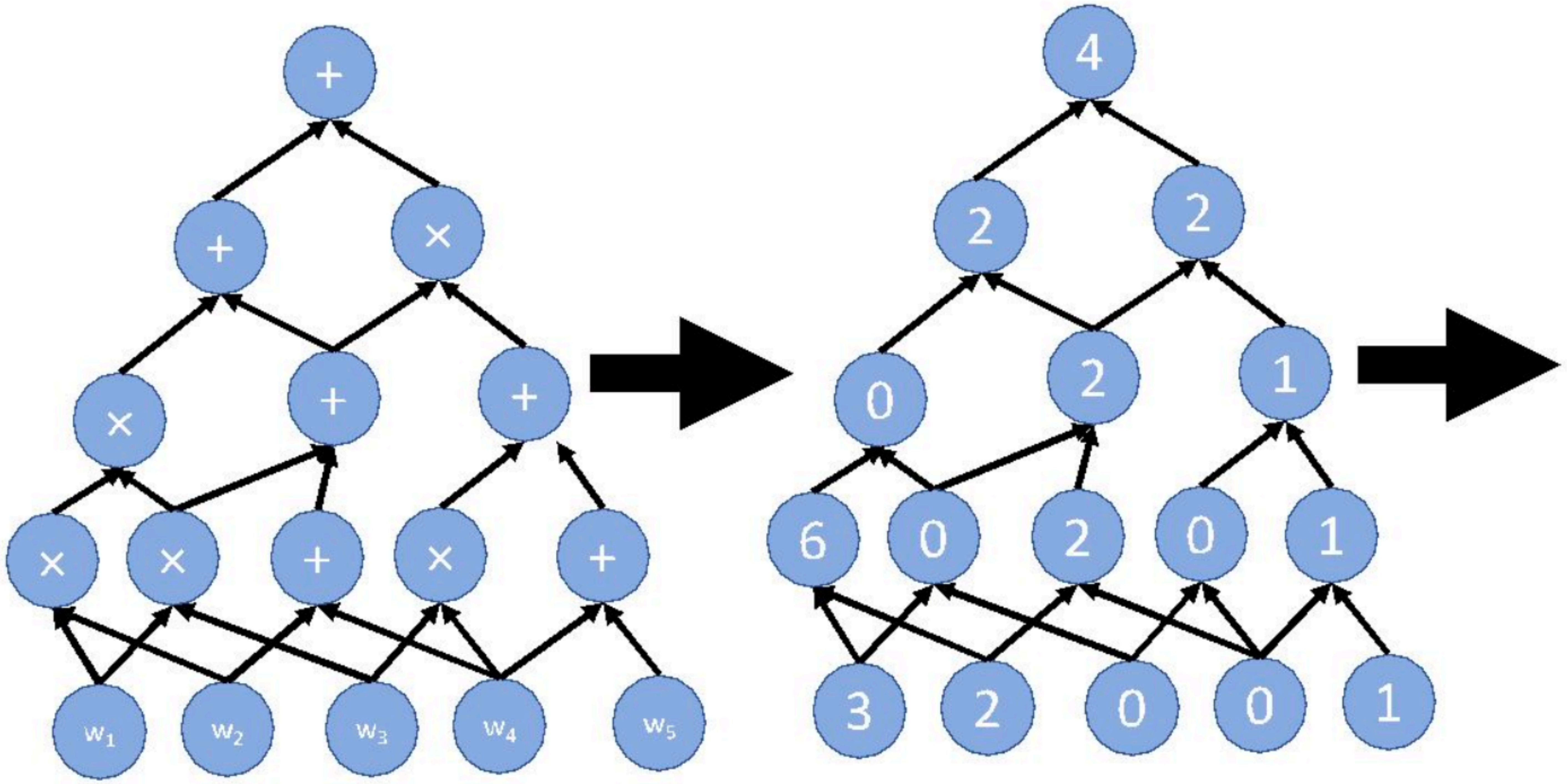


Circuit-SAT instance C



Viewing a transcript as a function

- 왼쪽 아래의 gate부터 테이블을 채움



Circuit-SAT instance C Correct transcript T for C

$T(0,0,0,0) = 3$
$T(0,0,0,1) = 2$
$T(0,0,1,0) = 0$
$T(0,0,1,1) = 0$
$T(0,1,0,0) = 1$
$T(0,1,0,1) = 6$
$T(0,1,1,0) = 0$
$T(0,1,1,1) = 2$
$T(1,0,0,0) = 0$
$T(1,0,0,1) = 1$
$T(1,0,1,0) = 0$
$T(1,0,1,1) = 2$
$T(1,1,0,0) = 1$
$T(1,1,0,1) = 2$
$T(1,1,1,0) = 2$
$T(1,1,1,1) = 4$

Polynomial Interactive Oracle Proof

- 왼쪽 아래의 gate부터 테이블을 채움
- $\mathscr{P}$는 $T(x)$를 확장시킨 $h(x)$를 claim
- $\mathscr{V}$ 는 evaluation 결과만 알 수 있음

Polynomial Interactive Oracle Proof

1. $(\log S)$-variate 다항식 h로 $(3\log S)$-variate 다항식 g_h를 만든다
   - h가 정확한 transcript T의 확장이다 $\Leftrightarrow g_h(a,b,c) = 0 \forall (a,b,c) \in \{0,1\}^{3\log S}$
2. $g_h(a,b,c) = 0 \forall (a,b,c) \in \{0,1\}^{3\log S}$ 인지 확인할 수 있는 Interactive proof
   - $\mathscr{V}$는 $g_h(r)$을 한번 계산하면 되도록

Polynomial Interactive Oracle Proof

- Define $g_h(a, b, c)$:

$$\widetilde{add}(a, b, c) \cdot (h(a) - (h(b) + h(c))) + \widetilde{mult}(a, b, c) \cdot (h(a) - h(b) \cdot h(c))$$

1. $g_h(a, b, c) = h(a) - (h(b) + h(c))$ (만약 a가 add gate라면)
2. $g_h(a, b, c) = h(a) - h(b) \cdot h(c)$ (만약 a가 mult gate라면)
3. 둘다 아닐경우 $g_h(a, b, c) = 0$

Polynomial Interactive Oracle Proof

- $g_h(a,b,c) = 0 \forall (a,b,c) \in \{0,1\}^{3\log S}$ 인지 확인할 수 있는 Interactive proof
  - 만약 univariate polynomial이었다면
    - $g_h(x) = 0 \forall x \in H \Leftrightarrow g_h$가 나눠지는 $Z_H(x) := \prod_{a \in H}(x-a)$
    - $Z_H(x)$는 vanishing polynomial
    - 그러나 여기서는 multivariate polynomial을 썼기 때문에 불가능

→ sumcheck 사용

Polynomial Interactive Oracle Proof

- $\sum_{a,b,c \in \{0,1\}^{\log S}} g_h(a, b, c)^2$ 에 대해서 sumcheck