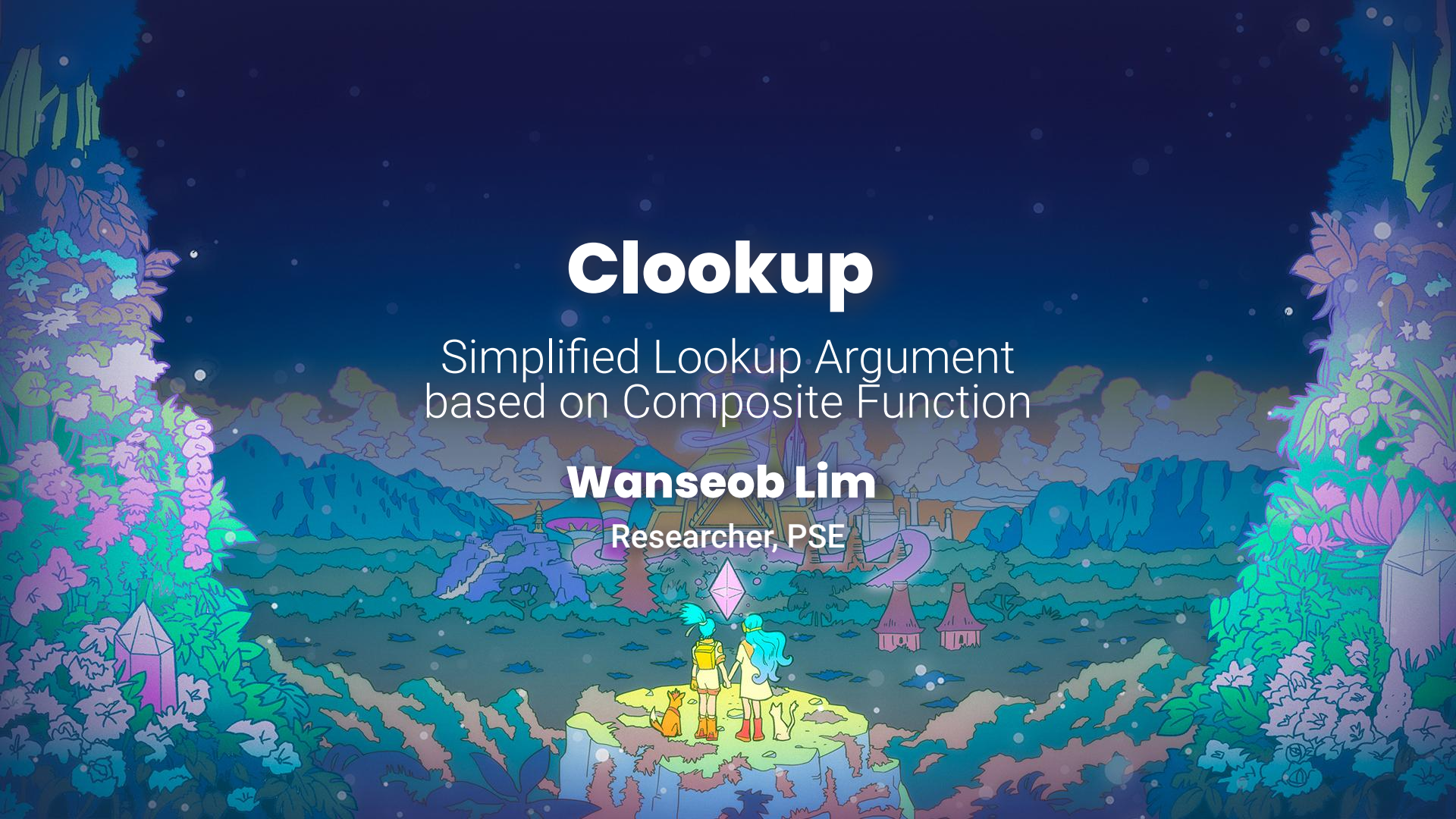
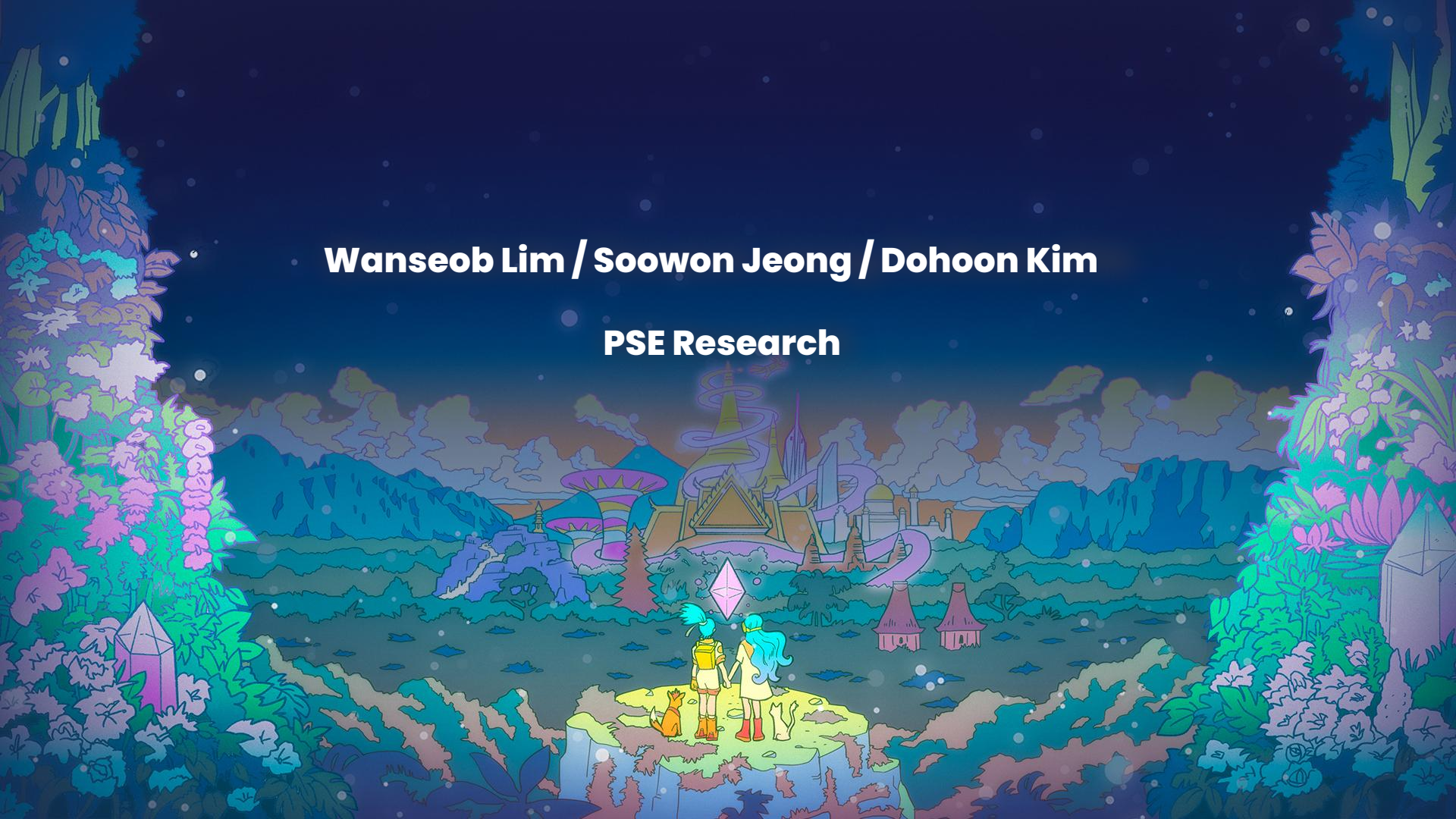

Clookup
Simplified Lookup Argument
based on Composite Function
Wanseob Lim
Researcher, PSE


Wanseob Lim / Soowon Jeong / Dohoon Kim
PSE Research

TL;DR

Given that

- n: Size of the table
- m: Size of the witness

TL;DR

Given that

- n: Size of the table
- m: Size of the witness

If $n \ll m$ (i.e., the table is small compared to the number of witnesses)

TL;DR

Given that

- n: Size of the table
- m: Size of the witness

If $n \ll m$ (i.e., the table is small compared to the number of witnesses)

then the time complexity with respect to the witness is approximately $\simeq O(\log^2 m)$

Motivations

- Inspired from GKR approach
-
-

Motivations

- Inspired from GKR approach
- Hyperplonk's permutation PIOP
-

Motivations

- Inspired from GKR approach
- Hyperplonk's permutation PIOP
- Using composite function for each layer

Lookup Argument

Lookup Argument

Given that

$$W := \{w_i\}_{i \in [m]} \quad T := \{t_i\}_{i \in [n]}$$

Lookup Argument

Given that

$$W := \{w_i\}_{i \in [m]} \quad T := \{t_i\}_{i \in [n]}$$

We prove that

$$\forall i \in [m], \quad w_i \in T$$

Lookup Argument

Given that

$$W := \{w_i\}_{i \in [m]} \quad T := \{t_i\}_{i \in [n]}$$

We prove that

$$\forall i \in [m], \quad w_i \in T$$

This implies that $W \subseteq T$.

Lookup Argument

What if:

- $w_i = (f_i, x_i, y_i)$ is a tuple of a function and its input and output.
- T is a table of predefined computations

Lookup Argument

What if:

- $w_i = (f_i, x_i, y_i)$ is a tuple of a function and its input and output.
- T is a table of predefined computations

Then, we can **formally verify** that:

- $\forall i \in [m], f_i(x_i) = y_i$ holds true
- Provided that $W \subseteq T$

Lookup Argument

What if:

- $w_i = (f_i, x_i, y_i)$ is a tuple of a function and its input and output.
- T is a table of predefined computations

Then, we can **formally verify** that:

- $\forall i \in [m], f_i(x_i) = y_i$ holds true
- Provided that $W \subseteq T$

➡ Formally verifiable computations

Composite Function based Lookup Argument

The relation $\mathcal{R}_{\text{clookup}}$ *is the set of all tuples* $(\mathbb{x}; \mathbb{w})$, *where:*

Composite Function based Lookup Argument

The relation $\mathcal{R}_{\text{clookup}}$ is the set of all tuples $(\mathbb{x}; \mathbb{w})$, where:

- **The public instance** $\mathbb{x} = ([[f]], [[t]], [[\sigma]])$
  is a tuple of oracles of functions f, t, σ.

Composite Function based Lookup Argument

The relation $\mathcal{R}_{\text{clookup}}$ is the set of all tuples $(\mathbb{x}; \mathbb{w})$, where:

- **The public instance** $\mathbb{x} = ([[f]], [[t]], [[\sigma]])$
  is a tuple of oracles of functions f, t, σ.
- **The witness** $\mathbb{w} = (f, t, \sigma)$
  is a tuple of functions f, t, σ

Composite Function based Lookup Argument

(...continued) *where:*

- $f : \mathcal{X}_f \to \mathcal{Y}_f$ maps witnesses
- $t : \mathcal{X}_t \to \mathcal{Y}_t$ maps table elements

Composite Function based Lookup Argument

(...continued) *where:*

- $f : \mathcal{X}_f \to \mathcal{Y}_f$ maps witnesses
- $t : \mathcal{X}_t \to \mathcal{Y}_t$ maps table elements

If there exists a domain transformation function $\sigma : \mathcal{X}_f \to \mathcal{X}_t$ *which satisfies the following:*

$$\forall x \in \mathcal{X}_f, \quad f(x) = t(\sigma(x))$$

Composite Function based Lookup Argument

(...continued) *where:*

- $f : \mathcal{X}_f \to \mathcal{Y}_f$ maps witnesses
- $t : \mathcal{X}_t \to \mathcal{Y}_t$ maps table elements

If there exists a domain transformation function $\sigma : \mathcal{X}_f \to \mathcal{X}_t$ *which satisfies the following:*

$$\forall x \in \mathcal{X}_f, \quad f(x) = t(\sigma(x))$$

Then, $(\mathbb{x}; \mathbb{w}) \in \mathcal{R}_{\text{clookup}}$ *stating that* $\mathcal{Y}_f \subseteq \mathcal{Y}_t$

Prove $((f, t, \sigma); ([[f]], [[t]], [[\sigma]])) \in \mathcal{R}_{clookup}$

Prove $((f, t, \sigma); ([[f]], [[t]], [[\sigma]])) \in \mathcal{R}_{clookup}$

Naive Approach:

- Let f, t, and σ be polynomials in $\mathbb{F}[X]$
- $\mathcal{X}_f := \{\omega^0, \ldots, \omega^{m-1}\}$ where $\omega^m = 1$

Prove $((f, t, \sigma); ([[f]], [[t]], [[\sigma]])) \in \mathcal{R}_{clookup}$

Naive Approach:

- Let f, t, and σ be polynomials in $\mathbb{F}[X]$
- $\mathcal{X}_f := \{\omega^0, \ldots, \omega^{m-1}\}$ where $\omega^m = 1$

$\forall x \in \mathcal{X}_f,\ f(x) \equiv t(\sigma(x))$ holds iff

Prove $((f, t, \sigma); ([[f]], [[t]], [[\sigma]])) \in \mathcal{R}_{clookup}$

Naive Approach:

- Let f, t, and σ be polynomials in $\mathbb{F}[X]$
- $\mathcal{X}_f := \{\omega^0, \ldots, \omega^{m-1}\}$ where $\omega^m = 1$

$\forall x \in \mathcal{X}_f,\ f(x) \equiv t(\sigma(x))$ holds iff

- $\exists Q \in \mathbb{F}[X]$ s.t.
- $f(X) - t\big(\sigma(X)\big) = Z(X) \cdot Q(X)$
- where $Z(X) = X^m - 1$

Prove $((f, t, \sigma); ([[f]], [[t]], [[\sigma]])) \in \mathcal{R}_{clookup}$

Naive Approach:

- Let f, t, and σ be polynomials in $\mathbb{F}[X]$
- $\mathcal{X}_f := \{\omega^0, \ldots, \omega^{m-1}\}$ where $\omega^m = 1$

$\forall x \in \mathcal{X}_f,\ f(x) \equiv t(\sigma(x))$ holds iff

- $\exists Q \in \mathbb{F}[X]$ s.t.
- $f(X) - t\big(\sigma(X)\big) = Z(X) \cdot Q(X)$
- where $Z(X) = X^m - 1$

Caveat:

- $\deg(t \circ \sigma) = \deg(t) \times \deg(\sigma)$

Prove $((f, t, \sigma); ([[f]], [[t]], [[\sigma]])) \in \mathcal{R}_{clookup}$

GKR-ish multivariate approach:

Prove $((f, t, \sigma); ([[f]], [[t]], [[\sigma]])) \in \mathcal{R}_{clookup}$

GKR-ish multivariate approach:

Let $\mathcal{X}_f := \{0,1\}^{\log m}$ and $\mathcal{X}_t := \{0,1\}^{\log n}$

Let $f \in \mathbb{F}[X_1, \ldots, X_{\log m}]$
Let $t \in \mathbb{F}[X_1, \ldots, X_{\log n}]$

Prove $((f, t, \sigma); ([[f]], [[t]], [[\sigma]])) \in \mathcal{R}_{clookup}$

GKR-ish multivariate approach:

Let $\mathcal{X}_f := \{0,1\}^{\log m}$ and $\mathcal{X}_t := \{0,1\}^{\log n}$

Let $f \in \mathbb{F}[X_1, \ldots, X_{\log m}]$
Let $t \in \mathbb{F}[X_1, \ldots, X_{\log n}]$

Let $\tilde{\sigma} := (\sigma_1, \ldots, \sigma_{\log n})$
where $\sigma_i \in \mathbb{F}[X_1, \ldots, X_{\log m}]$

$\forall x \in \mathcal{X}_f,\ f(x) \equiv t(\tilde{\sigma}(x))$ holds iff
(continue...)

Prove $((f, t, \sigma); ([[f]], [[t]], [[\sigma]])) \in \mathcal{R}_{clookup}$

(continued...)

$\exists h \in \mathbb{F}[X_1, \ldots, X_{\log m}]^{(\leq d)}$ s.t.

$$\sum_{\mathbf{x} \in \{0,1\}^{\log m}} h(\mathbf{x}) \equiv 0$$

and,

$$h(\mathbf{r}) \equiv \left(\begin{array}{l} f(\mathbf{r}) - t(\sigma(\mathbf{r})) \\ + \sum_{i \in [\log n]} \gamma_i \cdot \sigma_i(\mathbf{r}) \cdot (\sigma_i(\mathbf{r}) - 1) \end{array} \right) \cdot eq(\mathbf{r}, \mathbf{r}')$$

Prove $((f, t, \sigma); ([[f]], [[t]], [[\sigma]])) \in \mathcal{R}_{clookup}$

(continued...)
where:

- $\mathbf{r}, \mathbf{r}', \boldsymbol{\gamma}$ are random vectors uniformly chosen by the verifier.
- $h(\mathbf{X}) := \left(\underbrace{f(\mathbf{X}) - t(\tilde{\sigma}(\mathbf{X}))}_{\textbf{Mapping}} + \sum_{i \in [\log n]} \gamma_i \cdot \underbrace{\sigma_i(\mathbf{X}) \cdot (\sigma_i(\mathbf{X}) - 1)}_{\textbf{Domain Transformation}} \right) \cdot eq(\mathbf{X}, \mathbf{r}')$
- $eq(\mathbf{X}, \mathbf{Y}) := \prod_{i=1}^{\log m} (X_i Y_i + (1 - X_i)(1 - Y_i))$

Prove $((f, t, \sigma); ([[f]], [[t]], [[\sigma]])) \in \mathcal{R}_{clookup}$

(continued...)

Result:

- Time complexity

$$O(\underbrace{\log m}_{\text{sumcheck rounds}} \cdot (\underbrace{\log m}_{\text{Eval in } \mathbb{F}^{\log m}} + \underbrace{\log n}_{\text{Eval in } \mathbb{F}^{\log n}}))$$

Protocol

Assume a witness vector

$$\vec{f} = (2, 3, 5, 7)$$

is to be verified against a table vector

$$\vec{t} = (11, 5, 3, 17, 2, 13, 7, 19).$$

Protocol - Step 1. Preparation

Map the table into a multilinear polynomial

$$
\begin{aligned}
t(0,0,0) &= 11, & t(0,0,1) &= 5,\\
t(0,1,0) &= 3, & t(0,1,1) &= 17,\\
t(1,0,0) &= 2, & t(1,0,1) &= 13,\\
t(1,1,0) &= 7, & t(1,1,1) &= 19.
\end{aligned}
$$

In the preparation phase, we prepare the table polynomial in a coefficient form:

$$
\begin{aligned}
t(X_1, X_2, X_3) = &-19X_1X_2X_3\\
&+13X_1X_2 + 17X_1X_3 + 20X_2X_3\\
&-9X_1 - 8X_2 - 6X_3 + 11.
\end{aligned}
$$

Protocol - Step 2.1 Interpolate Witness

Interpolation of witnesses

$$f(0,0) = 2, \quad f(0,1) = 3, \quad f(1,0) = 5, \quad f(1,1) = 7.$$

We compute other polynomials in the evaluation form, but if we express this in a coefficient form for easier understanding:

$$f(X_1, X_2) = X_1X_2 + 3X_1 + X_2 + 2.$$

Protocol - Step 2.2 Interpolate $\tilde{\sigma}$

Interpolation of domain transformation

$$\begin{aligned}
\tilde{\sigma}(0,0) &= (\sigma_1(0,0), \sigma_2(0,0), \sigma_3(0,0)) = (1,0,0),\\
\tilde{\sigma}(0,1) &= (\sigma_1(0,1), \sigma_2(0,1), \sigma_3(0,1)) = (0,1,0),\\
\tilde{\sigma}(1,0) &= (\sigma_1(1,0), \sigma_2(1,0), \sigma_3(1,0)) = (0,0,1),\\
\tilde{\sigma}(1,1) &= (\sigma_1(1,1), \sigma_2(1,1), \sigma_3(1,1)) = (1,1,0).
\end{aligned}$$

Which means that

$$\begin{aligned}
&\sigma_1(0,0) = 1, \quad \sigma_1(0,1) = 0, \quad \sigma_1(1,0) = 0, \quad \sigma_1(1,1) = 1,\\
&\sigma_2(0,0) = 0, \quad \sigma_2(0,1) = 1, \quad \sigma_2(1,0) = 0, \quad \sigma_2(1,1) = 1,\\
&\sigma_3(0,0) = 0, \quad \sigma_3(0,1) = 0, \quad \sigma_3(1,0) = 1, \quad \sigma_3(1,1) = 0.
\end{aligned}$$

In coefficient form for easier understanding:

$$\begin{aligned}
\sigma_1(X_1, X_2) &= 2X_1X_2 - X_1 - X_2 + 1,\\
\sigma_2(X_1, X_2) &= X_2,\\
\sigma_3(X_1, X_2) &= -X_1X_2 + X_1.
\end{aligned}$$

Protocol - Step 3. Public Coin

The verifier initiates the process by selecting two random vectors:

1. For ZeroCheck: $\mathbf{r}, \mathbf{r}' = (r_1, \ldots, r_{\log m}) \in \mathbb{F}^{\log m}$
2. For random linear combination for batch: $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_{\log n}) \in \mathbb{F}^{\log n}$

Protocol - Step 3. Public Coin

The verifier initiates the process by selecting two random vectors:

1. For ZeroCheck: $\mathbf{r}, \mathbf{r}' = (r_1, \ldots, r_{\log m}) \in \mathbb{F}^{\log m}$
2. For random linear combination for batch: $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_{\log n}) \in \mathbb{F}^{\log n}$

Let's say we choose
$\mathbf{r} = (31, 37)$
$\mathbf{r}' = (5, 7)$
$\boldsymbol{\gamma} = (9, 11, 13)$

Protocol - Step 4. Sumcheck

$$
\begin{aligned}
h_i(X_i) = & \sum_{(\mathbf{x_{i+1}},\ldots,\mathbf{x_m})\in\{0,1\}^{m-i}} h(r_1,\ldots,r_{i-1},X_i,x_{i+1},\ldots,x_m) \\
& = h(r_1,\ldots,r_{i-1},x_i,0,\ldots,0) \\
& + h(r_1,\ldots,r_{i-1},x_i,0,\ldots,1) \\
& \vdots \\
& + h(r_1,\ldots,r_{i-1},x_i,1,\ldots,1)
\end{aligned}
$$

Protocol - Step 4. Sumcheck

1st round:

$$
\begin{aligned}
h_1(0) &= h(0,0) + h(0,1) = 0, \\
h_1(1) &= h(1,0) + h(1,1) = 0, \\
h_1(2) &= h(2,0) + h(2,1) = -204240, \\
h_1(3) &= h(3,0) + h(3,1) = -1018980, \\
h_1(4) &= h(4,0) + h(4,1) = -2850480,
\end{aligned}
$$

resulting in a polynomial representation of
$h_1(X_1) = -67710X_1^3 + 101010X_1^2 - 33300X_1$.

With a randomly selected $r_1' = 5$,
the computation yields

$h_1(r_1' = 5) = -6105000$.

Protocol - Step 4. Sumcheck

2nd round:

$$
\begin{aligned}
h_1(0) &= h(0,0) + h(0,1) = 0, \\
h_1(1) &= h(1,0) + h(1,1) = 0, \\
h_1(2) &= h(2,0) + h(2,1) = -204240, \\
h_1(3) &= h(3,0) + h(3,1) = -1018980, \\
h_1(4) &= h(4,0) + h(4,1) = -2850480,
\end{aligned}
$$

resulting in a polynomial representation of
$h_2(X_2) = -17399250X_2^4 + 70726425X_2^3 - 87850950X_2^2 + 44291775X_2 - 7936500$
.

To ensure consistency, the evaluations are summed and compared against the previously calculated value of h_1 at r_1':
$h_2(0) = h(r_1', 0) = -7936500$
$h_2(1) = h(r_1', 1) = 1831500$

The verifier checks if the sum of $h_2(0)$ and $h_2(1)$ matches the computed $h_1(r_1')$:
$h_2(0) + h_2(1) \stackrel{?}{=} h_1(r_1') = -6105000$

Protocol - Step 5. Final Evaluation

From the last round(2nd round), we could compute
$h_2(r_2' = 7) = -21519026100$

The verifier then validates that the final evaluation $h(5, 7)$ equals -21519026100

The process begins with the verifier requesting evaluations for the domain transformation $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_{\log n}) \in \mathbb{F}^{\log n}$, with each ν_i defined as $\nu_i := \tilde{\sigma}_i(\mathbf{r}')$.

Following this, the verifier queries both $f(\mathbf{r}')$ and $t(\boldsymbol{\nu})$, thereby preparing for a comprehensive comparison:

$$h(\mathbf{r}') \equiv \left(f(\mathbf{r}') - t(\boldsymbol{\nu}) + \sum_{i \in [\log n]} \gamma_i \nu_i (\nu_i - 1) \right) \cdot eq(\mathbf{r}, \mathbf{r}').$$

Protocol - Step 5. Final Evaluation

Consider the example, where the prover is tasked with providing oracle proofs for the queried values:

$$
\begin{aligned}
f(5,7) &= 59, \\
\sigma_1(5,7) &= 59, \\
\sigma_2(5,7) &= 7, \\
\sigma_3(5,7) &= -30, \\
t(59,7,-30) &= 206093.
\end{aligned}
$$

Subsequently, the verifier calculates $h(5,7)$ utilizing the aforementioned values:

$$
h(5,7) = \begin{pmatrix}
59 - 206093 \\
+ 9 \cdot (59 \cdot (59 - 1)) \\
+ 11 \cdot (7 \cdot (7 - 1)) \\
+ 13 \cdot (-30 \cdot (-30 - 1))
\end{pmatrix} \cdot 132275 = -21519026100.
$$

Table Decomposition

X_1	X_2	X_3	$t(X_1, X_2, X_3)$
0	0	0	11
0	0	1	5
0	1	0	3
0	1	1	17
1	0	0	2
1	0	1	13
1	1	0	7
1	1	1	19

Table Decomposition

X_1	X_2	X_3	$t(X_1, X_2, X_3)$	$t_1(X_2, X_3)$	$t_2(X_2, X_3)$
0	0	0	11	11	-
0	0	1	5	5	-
0	1	0	3	3	-
0	1	1	17	17	-
1	0	0	2	-	2
1	0	1	13	-	13
1	1	0	7	-	7
1	1	1	19	-	19

Table Decomposition

X_1	X_2	X_3	$t(X_1, X_2, X_3)$	$t_1(X_2, X_3)$	$t_2(X_2, X_3)$	$t_3(X_3)$	$t_4(X_3)$	$t_5(X_3)$	$t_6(X_3)$
0	0	0	11	11	-	11	-	-	-
0	0	1	5	5	-	5	-	-	-
0	1	0	3	3	-	-	3	-	-
0	1	1	17	17	-	-	17	-	-
1	0	0	2	-	2	-	-	2	-
1	0	1	13	-	13	-	-	13	-
1	1	0	7	-	7	-	-	-	7
1	1	1	19	-	19	-	-	-	19

Proving Intersection Size - Generalized Version

Let $b : \mathbb{Z}_q \to \{0,1\}^N$ be the bit decomposition function, which converts an element of $\mathbb{Z}_q$ into its N-bit binary representation.

Let $T \in \mathbb{F}_p[X_1, \ldots, X_N]$ be a multilinear polynomial that satisfies:

$$\begin{cases} T(\mathbf{x}) = 0 & \forall \mathbf{x} \in \{0,1\}^N - \{b(t_i)\}_{i \in [m]} \\ T(\mathbf{x}) = 1 & \forall \mathbf{x} \in \{b(t_i)\}_{i \in [m]} \end{cases}$$

Proving Intersection Size - Generalized Version

Let $\mathbf{W} \in (\mathbb{F}_p[X_1, \ldots, X_{\log n}])^N$ be defined as:

$$\mathbf{W} := (W_1, \ldots, W_N)$$

where $W_i(\mathbf{x})$ gives the i-th bit of a witness for $\mathbf{x}$.

Define the polynomial:

$$H := T(W_1(\mathbf{X}), \ldots, W_N(\mathbf{X}))$$

Then, we can compute $s := \sum H$ which means s is the size of the intersection.

Cost Analysis

$$T \times S = T_{\min} \times S_{\max} = C$$

$$\mathsf{T}(\mathsf{sumcheck}(h)) \propto \underbrace{\log m}_{\text{number of rounds}} \cdot \underbrace{\max(\mathsf{T}(f), \mathsf{T}(t \circ \tilde{\sigma}), \mathsf{T}(\gamma_i \sigma_i (\sigma_i - 1)), \mathsf{T}(eq))}_{\text{minimum time to evaluate } h}$$

$$\begin{aligned}
&= \log m \cdot \max(\log m, \log mn, \log m + 2, \log m) \\
&\simeq \log m \cdot (\log m + \log n) \\
&\simeq \log^2 m \\
&\text{(for small tables)}
\end{aligned}$$

Cost Analysis

$$T \times S = T_{\min} \times S_{\max} = C$$

$$\mathsf{T}(\mathsf{sumcheck}(h)) \propto \underbrace{\log m}_{\text{number of rounds}} \cdot \underbrace{\max(\mathsf{T}(f), \mathsf{T}(t \circ \tilde{\sigma}), \mathsf{T}(\gamma_i \sigma_i (\sigma_i - 1)), \mathsf{T}(eq))}_{\text{minimum time to evaluate } h}$$

$$= \log m \cdot \max(\log m, \log mn, \log m + 2, \log m)$$

$$\simeq \log m \ (\log m + \log n)$$

$$\simeq \log^2 m$$

(for small tables)

Cost Analysis

$$T \times S = T_{\min} \times S_{\max} = C$$

$$\mathsf{T}(\mathsf{sumcheck}(h)) \propto \underbrace{\log m}_{\text{number of rounds}} \cdot \underbrace{\max(\mathsf{T}(f), \mathsf{T}(t \circ \tilde{\sigma}), \mathsf{T}(\gamma_i \sigma_i (\sigma_i - 1)), \mathsf{T}(eq))}_{\text{minimum time to evaluate } h}$$

$$= \log m \cdot \max(\log m, \log mn, \log m + 2, \log m)$$

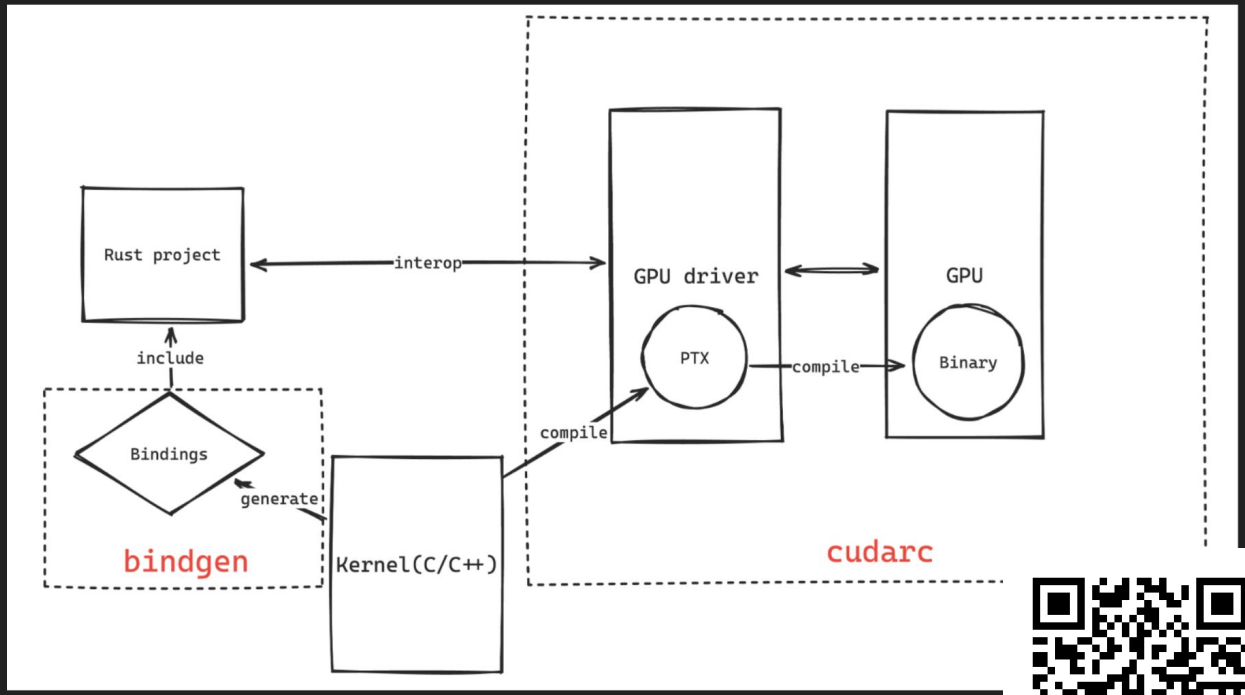
$$\simeq \log m \cdot (\log m + \log n)$$

$$\simeq \log^2 m$$

(for small tables)

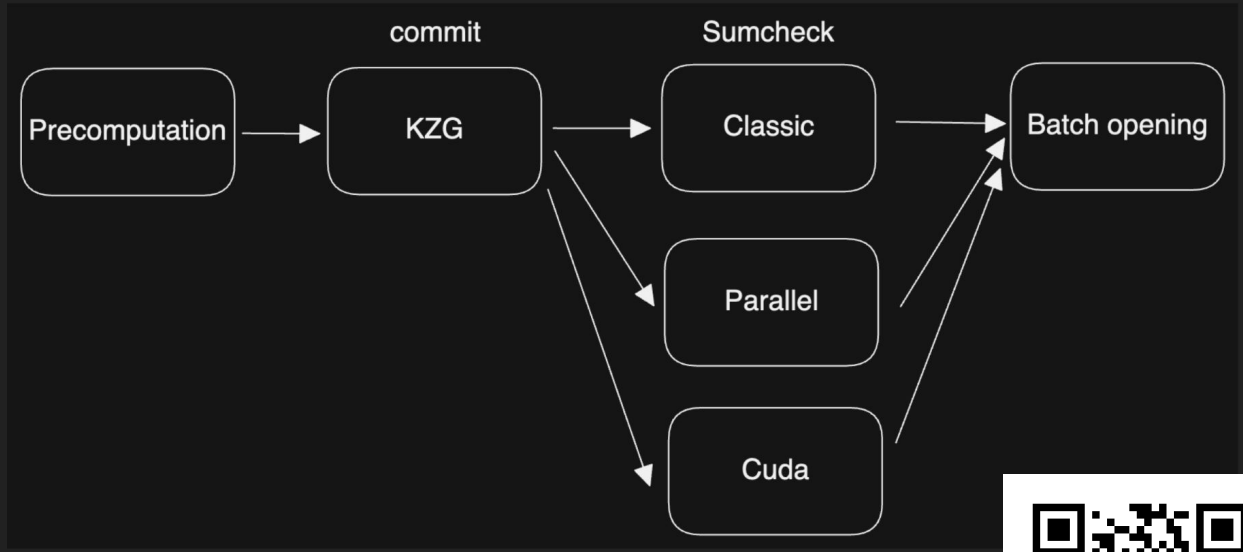
Composite function is a very generalized form of GKR

Cuda-sumcheck



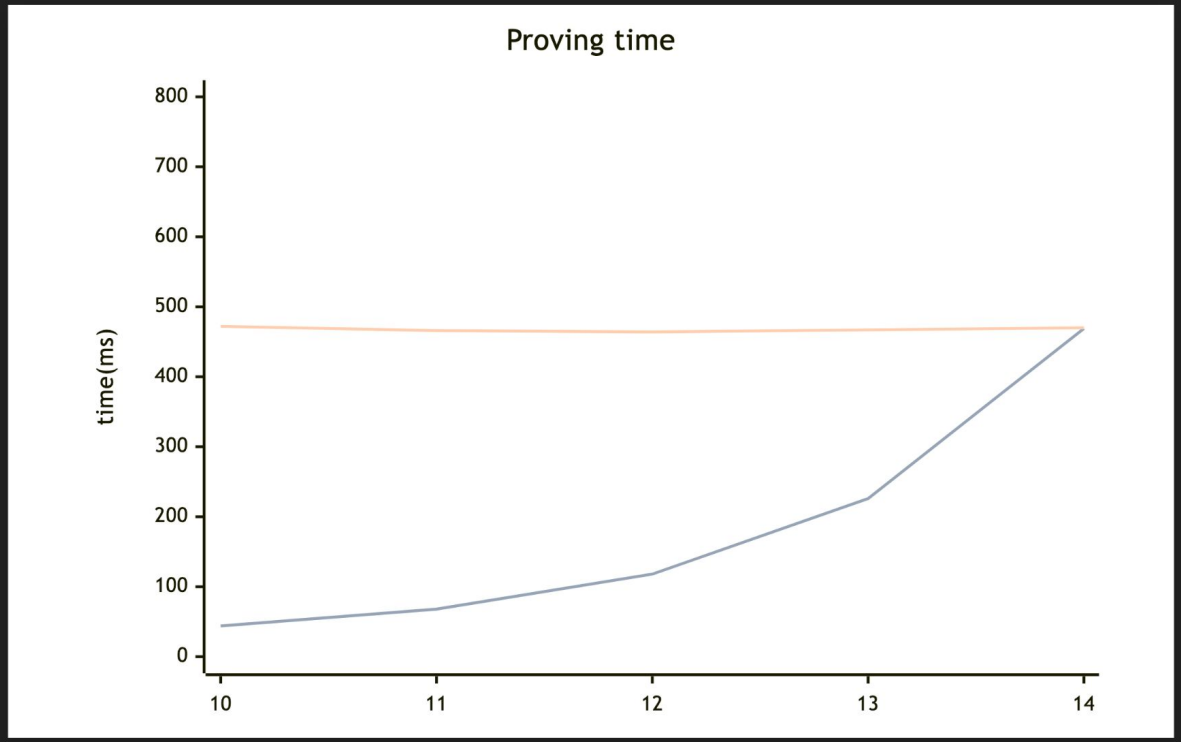
github.com/pseXperiments/cuda-sumcheck

Implementation



github.com/pseXperiments/clookup

Result


Proving time
time(ms)
800
700
600
500
400
300
200
100
0
10
11
12
13
14

Conclusion

- **Simple** scheme for lookup argument
- Support **table decomposition**
- Highly parallelizable and achieves **sublinear proving** time

Thank you!

Wanseob

Researcher, PSE