Bypassing the characteristic bound in logUp

Liam Eagen, Ulrich Haböck
liameagen@protonmail.com, uhaboeck@polygon.technology

December 20, 2024

Abstract

In this informal note, we describe how to bypass the characteristic bound in logUp [Hab22], by abstracting the notion of (pole) multiplicity. The method applies as well to the GKR-variant from [PH23], and moreover unlocks fractional decomposition lookups over binary fields.

Contents

1 Introduction **1**

2 The protocol **3**

3 Other bases **7**

3.1 Multilinear units 7

3.2 Repetition of units 8

1 Introduction

LogUp [Hab22] and its improved variant LogUp-GKR [PH23] are lookup arguments which use fractional decompositions for proving a sequence of values $x_1, x_2, \ldots, x_n$ from a finite field $\mathbb{F}_q$ being throughout contained in a given set of values $T = \{t_1, \ldots, t_N\}$, the “table”. Unlike in the classical approach, most prominently the permutation argument of Plonk [GWC19] and Plookup [GW20], the sequence is encoded into poles instead of roots, which allows a simpler treatment of multiplicity.[1] Membership across the sequence $x_1, x_2, \ldots$ is argued by providing for each table value $t_k \in T$ its multiplicity m_k in the sequence, satisfying

$$\frac{1}{X - x_1} + \frac{1}{X - x_2} + \ldots = \sum_{k=1}^{N} \frac{m_k}{X - t_k}, \tag{1}$$

[1] While [Eag22] is the first publication using the technique, the concurrent works [GK22] and [Hab22] rediscovered the same approach independently. For more details on the history of the technique, see the most recent version of [Hab22].

which is then proven by a sumcheck protocol for fractions. In the setting where the length of the sequence is significantly larger than the table (which is frequently met in wide-trace zk-VMs such as [Polb, Sta, BG, Suc, Pola]) the pole based approach drastically reduces the commitment pressure of the prover, down to a single table-sized column of "small" entries (the multiplicities) in logUp-GKR.

However, the soundness of the argument relies on the assumption that the number of elements to be looked up is *smaller* than the characteristic p of the finite field, $n < p$, so that poles on the left-hand side of Equation (1) cannot cancel out. While this does not seem an issue over prime fields, at least at first glance, this makes the approach seemingly impossible for STARKs over binary fields [BBHR18] which recently gained momentum due to "packed" polynomial commitment schemes [DP23, DP24, BCF$^+$24]. However, even over smaller primes such as the BabyBear prime [BG], the Mersenne prime M31 [HLN23, HLP24] or the recent KoalaBear prime [Gru] the characteristic bound may be exceeded in an extensive bus (or shared memory) consistency check of a multi-chip VM architecture.

In this writeup we discuss "Liam's trick"[2] for bypassing the characteristic bound, which in particular unlocks the fractional decomposition technique for binary fields. In a nutshell, the trick goes as follows. In order that poles cannot cancel out, regardless of the characteristic of the field, one takes any algebraic basis $u_1(Y), u_2(Y), \ldots, u_n(Y)$ (or more generally a basis in several variables) as "units" for each of the poles added,

$$\frac{u_1(Y)}{X - x_1} + \frac{u_2(Y)}{X - x_2} + \ldots,$$

which are thereby reduced to scalar "units" by evaluation at a random α,

$$\frac{u_1(\alpha)}{X - x_1} + \frac{u_2(\alpha)}{X - x_2} + \ldots.$$

Once these scalar units $u_1(\alpha), u_2(\alpha), \ldots$ are set, the protocol follows the usual logUp mechanics: The prover commits the abstracted table multiplicities $m_k(\alpha) = \sum_{j \in \{i \,:\, x_i = t_k\}} u_j(\alpha)$, and continues with a fractional sumcheck for

$$\frac{u_1(\alpha)}{X - x_1} + \frac{u_2(\alpha)}{X - x_2} + \ldots = \sum_{k=1}^{N} \frac{m_k(\alpha)}{X - t_k}.$$

This sumcheck may use univariate techniques, or more efficiently the GKR circuit described in [PH23].

Intuitively, the reduction step from polynomial units to scalar units should be sound since α is not known ahead of time, and it seems impossible to prepare

[2] The first author did not name the trick after himself

a sequence $x_1, x_2, \dots, x_n$ so that elements which are not contained in the table will cancel out. This intuition is correct, and in fact a simple application of the Schwartz-Zippel Lemma to the multiplicity polynomial of any malicious x' not contained in the table: We show that

1. for any sequence of polynomials $u_i(Y)$ of degree less than n and linearly independent in the vector space $\mathbb{F}_q^{<n}[Y]$ over $\mathbb{F}_q$, the soundness error is *linear* in n, and analogously,
2. for any sequence of multi-linear polynomials $u_i(Y_1, \dots, Y_d)$ in $d = \lceil \log n \rceil$ variables, again linearly independent over $\mathbb{F}_q$, the soundness error by evaluating at $\vec{\alpha} = (\alpha_1, \dots, \alpha_d)$ is logarithmic in n.

These two cases cover the most natural choices as units, which is the univariate monomial basis $u_i(Y) = Y^i$ and its multilinear counterpart $u_i(X_1, \dots, X_m) = X_1^{i_1} \cdot \ldots X_m^{i_m}$, with $i = \sum i_k \cdot 2^{k-1}$. Both choices are easily integrated in the gates of the fractional sumcheck circuit of logUp-GKR, or in the running sum constraint of a purely univariate proof, without increasing the degree.

This note is arranged as follows. Section 2 gives a high-level description of the modified logUp protocol, while taking the most natural choice of algebraic units, the (univariate) monomial one. We then discuss two other choices of units in Section 3. Section 3.1 covers the case of multilinear units, and in Section 3.2 we consider a weakening of the linear independence assumption, which is useful in certain use cases.

2 The protocol

We assume that the reader is familiar with the fractional sumcheck techniques used in [Hab22] and [PH23]. As therein, our modified protocol is described as interactive oracle proof, and for the sake of generality we do not specify the concrete oracle model, and how witness data is mapped/encoded into oracles. The idea applies to polynomial oracles (univariate or multivariate) in essentially the same manner as for ordinary oracles as used in STARKs, and we shall confine ourselves to one of these only when necessary.

Let $\mathbb{F}_q$ be a finite field of arbitrary characteristic $p \geq 2$. In a *batch-column lookup*, we want to prove set membership across $M \geq 1$ columns $x_0, \dots, x_{M-1} \in \mathbb{F}_q^N$ of length $N \geq 1$, with respect to a given table sequence $t = (t_i)_{i=0}^{N-1} \in \mathbb{F}_q^N$, i.e.

$$\{x_{i,j}\} \subseteq \{t_i\}$$

as sets, with multiplicities removed. We describe the protocol using the most elementary choice of polynomial units, the monomials

$$u_{i,j}(X) = X^{j \cdot M + i},$$

where $(i, j) \in [0, M-1] \times [0, N-1]$. (Here and in the sequel, $[a, b]$ denotes the integer interval including the boundaries a and b.) As pointed out before, this choice is just one of several possibilities, and in some applications taking just a few monomials, or the multilinear basis might be preferable, see Section 3.

Protocol 1 (logUp beyond the characteristic bound)**.** *Let $\mathbb{F}_q$ be a finite field of arbitrary characteristic $p \geq 2$, and F be a finite extension of $\mathbb{F}_q$. Given a table column $t \in \mathbb{F}_q^N$, the prover wishes to prove set membership for $M \geq 1$ witness columns $x_0, \ldots, x_{M-1} \in \mathbb{F}_q^N$, where $M \cdot N$ is allowed to be larger than p. (The columns $x_0, \ldots, x_{M-1}$ and t are committed, and the verifier has access to their oracles.)*

1. *The verifier samples $\alpha \leftarrow\!\!\$\ F$ and sends it to the prover.*
2. *The prover sets up the oracle for $m \in F^N$, the column of "randomized table multiplicities" defined as*

$$m_k = \sum_{\{(i,j)\,:\,x_{i,j}=t_k\}} u_{i,j}(\alpha), \tag{2}$$

   with $u_{i,j}(X)$ as defined above, and gives the verifier access to it.

3. *The verifier draws $\beta \leftarrow\!\!\$\ F$, and both prover and verifier engage in a fractional sumcheck protocol for*

$$\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \frac{u_{i,j}(\alpha)}{\beta - x_{i,j}} = \sum_{k=0}^{N-1} \frac{m_k}{\beta - t_k}. \tag{3}$$

If the sumcheck passes, the verifier accepts. (Otherwise, it rejects.)

Note that unlike in logUp or logUp-GKR, the terms on the left-hand side of the fractional sumcheck (3) now have non-trivial numerators $u_{i,j}(\alpha)$. However, these numerators are well-defined by the protocol and the verifier randomness α, and there is no need for an explicit commitment.

In both a univariate or multivariate strategy for the sumcheck, the $u_{i,j}(\alpha)$ can be integrated into the constraints without an increase of their degree. For example, in a purely univariate instantiation which uses additional columns

$$h, \quad \text{and} \quad h_0, \ldots, h_{M-1}$$

for the inverses $(1/(\beta - t_i))_{i=0}^{N-1}$ and $(1/(\beta - x_{j,i}))_{i=0}^{N-1}$, the constraint for the running sum $(U_i)_{i=0}^{N-1}$ of the fractional sumcheck is

$$\tilde{m}_i \cdot h_i - \sum_{j=0}^{M-1} \alpha^i \cdot h_{j,i} = U_i - \alpha^M \cdot U_{i+1 \bmod N},$$

for all $i \in [0, N-1]$, where

$$\tilde{m}_i = \frac{m_i}{\alpha^{M \cdot i}}$$

are the corrected table multiplicities, which are provided by the prover instead of m. (Note that the incremental constraint is in reverse direction, in order to accommodate our choice of powers.)

Likewise, the GKR circuit in [PH23] can be adapted to the *weighted* fractional sumcheck over the Boolean hypercube $H_n = \{\pm 1\}^n$,

$$\sum_{\vec{x} \in H_n} \alpha^{\iota(\vec{x})} \cdot \frac{p(\vec{x})}{q(\vec{x})} = 0,$$

where $\iota(\vec{x}) = \sum_{i=0}^{n-1} \frac{1-x_i}{2} \cdot 2^i$. (In this part, we assume that the reader is familiar with [PH23].) The modified circuit again computes the fractional sum along a binary tree with n layers, and in each of its layers $k = 0, \ldots, n-1$, we throughout take the gate which multiplies the right child by α^{2^k} before it adds it to the left child, i.e.

$$\begin{aligned} p_k(\vec{x}) &= p_{k+1}(\vec{x}, +1) \cdot q_{k+1}(\vec{x}, -1) + \alpha^{2^k} \cdot p_{k+1}(\vec{x}, -1) \cdot q_{k+1}(\vec{x}, +1), \\ q_k(\vec{x}) &= q_{k+1}(\vec{x}, +1) \cdot q_{k+1}(\vec{x}, -1). \end{aligned}$$

This additional multiplication does not change the degree of the GKR constraints (they are still cubic), although their evaluation is slightly costlier. Under the simplifying assumption that $M = 2^k - 1$, the modified circuit may also be used for the combined multilinear polynomial over the hypercube $H_{n+k} = H_k \times H_n$, which also includes the table column,

$$\begin{aligned} p(\vec{X}, \vec{Y}) &= L_k(\vec{Y}, \vec{1}) \cdot m(\vec{X}) - \sum_{\vec{y} \in H_k \setminus \{\vec{1}\}} L_k(\vec{Y}, \vec{y}) \cdot 1, \\ q(\vec{X}, \vec{Y}) &= L_k(\vec{Y}, \vec{1}) \cdot (\beta - t(\vec{X})) + \sum_{\vec{y} \in H_k \setminus \{\vec{1}\}} L_k(\vec{Y}, \vec{y}) \cdot (\beta - x_{\iota(\vec{y})}(\vec{X})), \end{aligned}$$

with $L_k(\vec{x}, \vec{y}) = \frac{1}{2^k} \cdot \prod_{i=1}^{k} (1 + x_i \cdot y_i)$ as the k-dimensional multilinear Lagrange kernel, and $\vec{1} = (1, \ldots, 1) \in \mathbb{F}_q^k$. For that, the prover commits to the corrected multiplicities

$$\tilde{m}_i = \frac{m_i}{\alpha^{i \cdot M}},$$

instead, so that the powers of α in the GKR circuit cancel out for the table entries.

For simplicity, and without explicitly defining the security notions, we state the soundness error of the protocol as a *polynomial interactive oracle proof (PIOP)*. An adaption of the analysis to the ordinary oracle model is along the common steps of turning an algebraic interactive oracle proof into an IOP of proximity.

Theorem 1. *The soundness error of Protocol 1 as an PIOP is at most*

$$\varepsilon < \frac{M \cdot N}{|F|} + \frac{(M+1) \cdot N}{|F|} + \varepsilon_{sumcheck},$$

where $\varepsilon_{sumcheck}$ is the soundness error of the fractional sumcheck in Step (3) of the protocol.

The proof of Theorem 1 is in the usual round-by-round manner. The first term in the error bound is due to the reduction from monomial units to scalar units, and the second term for reducing the rational identity to a scalar-valued sumcheck. We note that the overall soundness error is slightly reduced when using a multilinear basis as units, see Theorem 3.

Proof of Theorem 1. First of all, we observe that with the sequence of values $x_{i,j}$, $0 \leq i < N$, $0 \leq j < M$, the prover implicitly committed to the partition of the index set into subsets of equal value, and with it the multiplicity polynomials $m_x(X) = \sum_{\{(i,j)\,:\,x_{i,j}=x\}} u_{i,j}(X)$. Since the $u_{i,j}(X)$ are linearly independent and of degree $< M \cdot N$, all multiplicity polynomials are non-zero and of degree $< M \cdot N$.

Assume that the set $S = \{x_{i,j}\}$ is not contained in the set of table values $T = \{t_i\}$. We decompose the bad event E, that the prover is able to pass the verifier, into a series of bad events E_1, E_2, E_3 corresponding to each of the probabilistic reductions steps made during the protocol.

- Step 1 (Sampling α). By the degree of the multiplicity polynomials, the number of α for which $m_{x'}(\alpha) = 0$ for all $x' \in S \setminus T$ is less than $M \cdot N$. Thus, except for a set E_1 of probability of $M \cdot N/|F|$ in α, there is a value $x' \in S \setminus T$ such that its multiplicity polynomial evaluates to a non-zero value at α. Consequently the rational function $\sum_{x \in S} \frac{m_x(\alpha)}{X-x}$ has pole at $x' \notin T$, whereas the function $\sum_{i=0}^{N-1} \frac{m_i}{X-t_i}$ has only poles over a subset of T. By the uniqueness of fractional decomposition, the identity

  $$\sum_{x \in S} \frac{m_x(\alpha)}{X - x} = \sum_{i=0}^{N-1} \frac{m_i}{X - t_i}$$

  cannot hold.

- Step 2 (Sampling β). By multiplying with each of the denominators, the fractional identity is equivalent to a polynomial identity of degree $N \cdot (M+1)$, and therefore, except for set E_2 of probability $N \cdot (M+1)/|F|$ in β, the identity also does not hold at β.

Finally, the soundness error for the fractional sumcheck tells us that the set E_3 on which the sumcheck passes although the identity does not hold at β, has probability at most $\varepsilon_{sumcheck}$. Taking the union bound of these three bad events yields the claim of the Proposition. □

Remark 2. It is immediate from the proof that the same soundness error applies to an arbitrary basis of linearly independent polynomials $u_{i,j}(Y)$ of degree $< n$. An alternative choice is the Lagrange basis over the index set, the latter embedded onto a disjoint coset union in a sufficiently large extension of $\mathbb{F}_q$, in order to serve compact representations of the Lagrangians. Contrary to the monomial case, the GKR circuit remains untouched. However, although the Lagrangian is succinctly evaluable, the concrete prover costs are higher.

3 Other bases

In this section we quickly discuss other choices for the algebraic units $u_{i,j}$ in Protocol 1.

3.1 Multilinear units

In the context of multivariate proofs, multilinear monomials are the more natural choice of units. We again assume for simplicity that $M = 2^k - 1$, and that all columns, including the table column, are of length $N = 2^n$ and hosted by the hypercube $H_{k+n} = H_k \times H_n$. The units are

$$u_{\vec{i},\vec{j}}(X_1, \ldots, X_{k+n}) = X_1^{i_1} \cdot \ldots \cdot X_k^{i_k} \cdot X_{k+1}^{j_1} \cdot \ldots X_{k+n}^{j_n},$$

with $\vec{i} = (i_1, \ldots, i_k) \in H_k$ selecting the column and $\vec{j} = (j_1, \ldots, j_n) \in H_n$ their entries.

The modifications to Protocol 1 and the GKR fractional sumcheck are minimal: In Step 1 the verifier now samples a random vector $\vec{\alpha} = (\alpha_1, \ldots, \alpha_{k+n})$ uniformly from F^{k+n}, which is used to evaluate the unit monomials $u_{\vec{i},\vec{j}}$. The circuit for the $\vec{\alpha}$-weighted sumcheck over $H_{k+n} = \{\pm 1\}^{k+n}$,

$$\sum_{\vec{x} \in H_{k+n}} \alpha_1^{\frac{1-x_1}{2}} \cdot \ldots \cdot \alpha_n^{\frac{1-x_{k+n}}{2}} \cdot \frac{p(\vec{x})}{q(\vec{x})} = 0,$$

uses in each of its layers $k = 0, \ldots, n+k-1$ the gate which multiplies the right child by α_{k+1} before it adds it to the left child, i.e.

$$p_k(\vec{x}) = p_{k+1}(\vec{x}, +1) \cdot q_{k+1}(\vec{x}, -1) + \alpha_{k+1} \cdot p_{k+1}(\vec{x}, -1) \cdot q_{k+1}(\vec{x}, +1),$$
$$q_k(\vec{x}) = q_{k+1}(\vec{x}, +1) \cdot q_{k+1}(\vec{x}, -1),$$

where we the input polynomials are

$$p(\vec{X},\vec{Y}) = L_k(\vec{Y},\vec{1}) \cdot \tilde{m}(\vec{X}) - \sum_{\vec{y}\in H_k\setminus\{\vec{1}\}} L_k(\vec{Y},\vec{y}) \cdot 1,$$

$$q(\vec{X},\vec{Y}) = L_k(\vec{Y},\vec{1}) \cdot (\beta - t(\vec{X})) + \sum_{\vec{y}\in H_k\setminus\{\vec{1}\}} L_k(\vec{Y},\vec{y}) \cdot (\beta - x_{\iota(\vec{y})}(\vec{X})),$$

with $\tilde{m}(\vec{X})$ as the corrected table multiplicities

$$\tilde{m}_{j_1,\ldots,j_n} = \frac{m_{j_1,\ldots,j_n}}{\alpha_{k+1}^{j_1} \cdot \alpha_{k+n}^{j_n}}.$$

An inspection of the proof of Theorem 1 shows that the soundness error in the multilinear regime is even slightly less, due to the Schwartz-Zippel lemma for multilinears. We state the result without proof.

Theorem 3 (Multilinear units)**.** *Under the above described modification of Protocol 1, which uses the multilinear basis as units, the soundness error the IOP of the batch-lookup of $M = 2^m - 1$ columns of length $N = 2^n$ into a table of length N is*

$$\varepsilon < \frac{n+m}{|F|} + \frac{(M+1)\cdot N}{|F|} + \varepsilon_{sumcheck},$$

where $\varepsilon_{sumcheck}$ is the soundness error of the fractional sumcheck in Step (3) of the protocol.

3.2 Repetition of units

Another useful variant of Protocol 1 utilizes a weakening of the linear independence assumption of the unit polynomials. For the proof of Theorem 1 it is sufficient that the any subset sum of $(u_{i,j}(Y))_{i,j}$ cannot cancel out. That is, for every subset S of the index set,

$$\sum_{(i,j)\in S} u_{i,j}(Y) \neq 0$$

in $\mathbb{F}_q^{<n}[Y]$. This allows to expand a basis by repeating each basis vector up to $p-1$ times, where p is the characteristic of $\mathbb{F}_q$. For example, in the case

$$N < p$$

it suffices to take a single monomial per witness column, or more generally per group of columns the size of which does not exceed the characteristic. Likewise, if the entire payload to be looked up is less than

$$N \cdot M \leq [F : \mathbb{F}_q] \cdot (p-1),$$

where $[F : \mathbb{F}_q]$ is the degree of the extension F, it is sufficient to take a basis of $F/\mathbb{F}_q$ as units, dropping the need of sampling α in Protocol 1.

Acknowledgements

The second author would like to thank the first one for approaching him with this elegant solution. We would also like thank Angus Gruen to point out the constant units case in Section 3.2.

References

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. In *IACR ePrint Archive 2018/046*, 2018. https://eprint.iacr.org/2018/046.

[BCF$^+$24] Martijn Brehm, Binyi Chen, Ben Fisch, Nicolas Resch, Ron D. Rothblum, and Hadas Zeilberger. Blaze: Fast SNARKs from interleaved RAA codes. In *IACR preprint archive 2024/1609*, 2024. https://eprint.iacr.org/2024/1609.

[BG] Jeremy Bruestle and Paul Gafni. RISC Zero zkVM: scalable, transparent arguments of RISC-V integrity. https://www.risczero.com/proof-system-in-detail.pdf.

[DP23] Benjamin E. Diamond and Jim Posen. Succinct arguments over towers of binary fields. In *IACR ePrint Archive 2023/1784*, 2023. https://eprint.iacr.org/2023/1784.

[DP24] Benjamin E. Diamond and Jim Posen. Polylogarithmic proofs for multilinears over binary towers. In *IACR preprint archive 2024/504*, 2024. https://eprint.iacr.org/2024/504.

[Eag22] Liam Eagen. Bulletproofs++: Next generation confidential transactions via reciprocal set membership arguments. In *IACR ePrint Archive 2022/510*, 2022. https://eprint.iacr.org/2022/510.

[GK22] Ariel Gabizon and Dmitry Khovratovich. flookup: Fractional decomposition-based lookups in quasi-linear time independent of the table size. In *IACR ePrint Archive 2022/1447*, 2022. https://eprint.iacr.org/2022/1447.

[Gru] Angus Gruen. Small prime fields in Plonky3. https://hackmd.io/51_gB_3ZSBSZ4KafGGvkUw#.

[GW20] Ariel Gabizon and Zachary J. Williamson. Plookup: A simplified polynomial protocol for lookup tables. In *IACR ePrint Archive 2020/315*, 2020. https://eprint.iacr.org/2020/315.

[GWC19] Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. PLONK: Permutations over Lagrange-bases for oecumenical noninteractive arguments of knowledge. In *IACR ePrint Archive 2019/953*, 2019. `https://eprint.iacr.org/2019/953`.

[Hab22] Ulrich Haböck. Multivariate lookups based on logarithmic derivatives. In *IACR ePrint Archive 2022/1530*, 2022. `https://eprint.iacr.org/2022/1530`.

[HLN23] Ulrich Haböck, Daniel Lubarov, and Jacqueline Nabaglo. Reed-Solomon Codes over the Circle Group. In *IACR preprint archive*, 2023. `https://eprint.iacr.org/2023/824`.

[HLP24] Ulrich Haböck, David Levit, and Shahar Papini. Circle STARKs. In *IACR preprint archive*, 2024. `https://eprint.iacr.org/2024/278`.

[PH23] Shahar Papini and Ulrich Haböck. Improving logarithmic derivative lookups using GKR. In *IACR ePrint Archive 2023/1284*, 2023. `https://eprint.iacr.org/2023/1284`.

[Pola] Polygon Miden. Polygon Miden: A STARK-based virtual machine. `https://github.com/maticnetwork/miden`.

[Polb] Polygon Zero. Polygon Zero Type 1 zkEVM. `https://github.com/0xPolygonZero/zk_evm`.

[Sta] Starkware. Stwo Prover. `https://github.com/starkware-libs/stwo`.

[Suc] Succinct Labs. SP1. `https://github.com/succinctlabs/sp1`.