TIP 0001: Contiguity Argument for Memory Consistency

TIP	0001
authors:	Alan Szepieniec and Ferdinand Sauer
title:	Contiguity Argument for Memory Consistency
status:	draft
created:	2022-08-15
issue tracker:	https://github.com/TritonVM/triton-vm/pull/35
pdf:	tip-0001.pdf

Abstract. In the current specification, the memory-like tables `RamTable`, `JumpStackTable`, and `OpStackTable` do not satisfy memory-consistency. Specifically, they are vulnerable to Yuncong's attack, which exploits the unverified and thus possibly-incorrect sorting in these tables. This TIP addresses one part of the issue by introducing a new table argument, the *contiguity argument*. It establishes the contiguity of the regions of fixed memory pointer. This note is a companion to TIP-0003, which establishes that in memory-like tables, jumps in the clock cycle can only be directed forward, never backward. Together, TIP-0001 and TIP-0003 fix the memory consistency issue.

Introduction

The memory-like tables `RamTable`, `JumpStackTable`, and `OpStackTable` should be sorted by memory pointer first, and by clock cycle second. When this correct sorting is not enforced, it gives rise to attack undermining memory-consistency.

Part V of the BrainSTARK tutorial shows that memory-consistency follows if, in the memory table, every sublist of rows with the same memory pointer forms a contiguous region. The sorting rule is just one way to guarantee this contiguity. The sorting by clock cycle within each contiguous region is still necessary.

This TIP proposes a subprotocol for establishing the contiguity of all regions with a given memory pointer. This *contiguity argument* is a collection of one base column, two extension columns, two polynomial commitments, two consistency constraints, two randomized transition constraints, and one randomized terminal constraint. The base column and consistency constraints enable conditioning on a changed memory pointer. The first extension column is a running product similar to that of a conditioned permutation argument. The first randomized transition constraint verifies the correct accumulation of factors for updating this column The second extension column is the formal derivative of the second. The second randomized transition constraint verifies the correct application of the product rule of differentiation to update this column. The two commited-to polynomials are Bézout coefficients. The terminal constraint takes the weighted

sum of the running product and the formal derivative, where the weights are the Bézout coefficients, and equates it to one. This equation asserts the Bézout relation. It can only be satisfied if the greatest common divisor of the running product and its formal derivative is one – implying that no change in the memory pointer resets it to a value used earlier.

Contiguity Argument

The contiguity argument is only needed for the Ram Table because the memory pointer there (`ramp`) can take any value. In contrast, the two other memory-like tables, OpStackTable and JumpStackTable, are stacks. After sorting by the memory pointer, checking the contiguity of the access pattern is easy: across two consecutive rows, the memory pointer either remains unchanged or increments by one.

Since the contiguity argument may have applications elsewhere, it is presented here in a generic language.

Let `ramp` be the column whose contiguity we wish to establish. We add one base column and two extension columns: the difference inverse `di`, the running product `rp`, and the formal derivative `fd`. Additionally, the prover commits to two polynomials $a(X)$ and $b(X)$, but these polynomials do not necessarily correspond to columns in the table.

The values contained in the extension columns are undetermined until the verifier's challenge α is known; before that happens it is worthwhile to present the polynomial expressions in X, anticipating the substitution $X \mapsto \alpha$.

The difference inverse `di` takes the inverse of the difference between the current and next `ramp` values if that difference is non-zero, and zero else. This constraint corresponds to two consistency constraint polynomials:

- $(\mathsf{ramp}^\star - \mathsf{ramp}) \cdot ((\mathsf{ramp}^\star - \mathsf{ramp}) \cdot \mathsf{di} - 1)$, and
- $\mathsf{di} \cdot ((\mathsf{ramp}^\star - \mathsf{ramp}) \cdot \mathsf{di} - 1)$.

The running product `rp` starts with 1 initially and accumulates a factor $X - \mathsf{ramp}$ in every pair of rows where $\mathsf{ramp} \neq \mathsf{ramp}^\star$. This evolution corresponds to one transition constraint: $(\mathsf{ramp}^\star - \mathsf{ramp}) \cdot (\mathsf{rp}^\star - \mathsf{rp} \cdot (X - \mathsf{ramp})) + (1 - (\mathsf{ramp}^\star - \mathsf{ramp}) \cdot \mathsf{di}) \cdot (\mathsf{rp}^\star - \mathsf{rp})$.

Denote by $f_{\mathsf{rp}}(X)$ the polynomial that accumulates all factors $X - \mathsf{ramp}$ in every pair of rows where $\mathsf{ramp} \neq \mathsf{ramp}^\star$. If the last row is a padding row then the value in the rp column is exactly $f_{\mathsf{rp}}(X)$. If the last row is not a padding row then the column has not accumulated the factor $X - \mathsf{ramp}$ yet, and so the relevant relation is $f_{\mathsf{rp}}(X) = \mathsf{rp} \cdot (X - \mathsf{ramp})$.

The column `fd` contains the "formal derivative" of the running product with respect to X. The formal derivative is initially 0. The transition constraint applies the product rule of differentiation conditioned upon the difference in `ramp`

being nonzero; in other words, if $\mathsf{ramp} = \mathsf{ramp}^\star$ then the same value persists; but if $\mathsf{ramp} \neq \mathsf{ramp}^\star$ then fd is mapped as

$$\mathsf{fd} \mapsto \mathsf{fd}^\star = (X - \mathsf{ramp}) \cdot \mathsf{fd} + \mathsf{rp}.$$

This update rule is called the *product rule of differentiation* because, assuming $\mathsf{ramp}^\star \neq \mathsf{ramp}$, then

$$\begin{aligned}\frac{\mathsf{drp}^\star}{\mathsf{d}X} &= \frac{\mathsf{d}(X - \mathsf{ramp}) \cdot \mathsf{rp}}{\mathsf{d}X} \\ &= (X - \mathsf{ramp}) \cdot \frac{\mathsf{drp}}{\mathsf{d}X} + \frac{\mathsf{d}(X - \mathsf{ramp})}{\mathsf{d}X} \cdot \mathsf{rp} \\ &= (X - \mathsf{ramp}) \cdot \mathsf{fd} + \mathsf{rp} \ .\end{aligned}$$

The transition constraint for fd is $(\mathsf{ramp}^\star - \mathsf{ramp}) \cdot (\mathsf{fd}^\star - \mathsf{rp} - (X - \mathsf{ramp}) \cdot \mathsf{fd}) + (1 - (\mathsf{ramp}^\star - \mathsf{ramp}) \cdot \mathsf{di}) \cdot (\mathsf{fd}^\star - \mathsf{fd})$.

Let $f_{\mathsf{fd}}(X)$ denote the formal derivative of $f_{\mathsf{rp}}(X)$. If the last row is a padding row, then fd is exactly $f_{\mathsf{fd}}(X)$. However, if the last row is not a padding row then the product rule must be applied one last time. In this case the relevant relation is $f_{\mathsf{fd}}(X) = \mathsf{rp} + \mathsf{fd} \cdot (X - \mathsf{ramp})$.

The polynomials $a(X - \tau_a)$ and $b(X - \tau_b)$ are *randomized* Bézout coefficients of the relation

$$a(X - \tau_a) \cdot f_{\mathsf{rp}}(X) + b(X - \tau_b) \cdot f_{\mathsf{fd}}(X) = \gcd(f_{\mathsf{rp}}(X), f_{\mathsf{fd}}(X)) \ .$$

The prover finds $a(X)$ and $b(X)$ as $a(X) = a^*(X + \tau_a) + k \cdot f_{\mathsf{fd}}(X + \tau_a)$ and $b(X) = b^*(X + \tau_b) - k \cdot f_{\mathsf{rp}}(X + \tau_b)$, where $a^*(X)$ and $b^*(X)$ are the minimal-degree Bézout coefficients as returned by the extended Euclidean algorithm, and where $k \in \mathbb{F}[X]$ is a uniformly random polynomial of degree at most $8t$ called the *Bézout randomizer*. For the sake of the soundness analysis, set the degree bound on $a(X)$ and $b(X)$ to $T + 8t$, where T is the height of the table.

The terms τ_a and τ_b are offsets chosen uniformly at random by the prover. They are transmitted simultaneously with the commitment to the algebraic execution trace.

When the verifier supplies α, the prover responds with (among other things) the randomized Bézout coefficients $A = a(\alpha - \tau_a)$ and $B = b(\alpha - \tau_b)$. The verifier verifies their correct calculation using the DEEP technique: in addition to adding $a(X)$ and $b(X)$, the prover adds $q_a(X) = \frac{a(X) - A}{X - \alpha + \tau_a}$ and $q_b(X) = \frac{b(X) - B}{X - \alpha + \tau_b}$ to the nonlinear combination, and the verifier verifies that it was added.

The verifier additionally verifies the randomized AIR terminal constraint $A \cdot f_{\mathsf{rp}}(\alpha) + B \cdot f_{\mathsf{fd}}(\alpha) = 1$. Recall that if the last row is a padding row then $f_{\mathsf{rp}}(\alpha) = \mathsf{rp}$

and $f_{\mathsf{fd}}(\alpha) = \mathsf{fd}$; whereas if it is not a padding row then $f_{\mathsf{rp}}(\alpha) = \mathsf{rp} \cdot (\alpha - \mathsf{ramp})$ and $f_{\mathsf{fd}}(\alpha) = \mathsf{rp} + \mathsf{fd} \cdot (\alpha - \mathsf{ramp})$.

Completeness. For honest provers, the gcd is guaranteed to be one. As a result, the protocol has perfect completeness. □

Soundness. If the table has at least one non-contiguous region, then $f_{\mathsf{rp}}(X)$ and $f_{\mathsf{fd}}(X)$ share at least one factor. As a result, no Bézout coefficients $a(X)$ and $b(X)$ can exist such that $a(X) \cdot f_{\mathsf{rp}}(X) + b(X) \cdot f_{\mathsf{fd}}(X) = 1$. The verifier therefore probes unequal polynomials of degree at most [todo]. According to the Schwartz-Zippel lemma, the false positive probability is at most $2T + 8t/|\mathbb{F}|$. □

Zero-Knowledge. The standard workflow of interpolating columns with randomizers guarantees that the columns do not leak information. However, the same argument does not apply for the polynomial commitments to $a(X)$ and $b(X)$. The transcripts contains the evaluations of these polynomials in $4t + 1$ points, where t is the number of colinearity checks in FRI and where the $+1$ comes from the evaluation $a(\alpha)$ and $b(\alpha)$.

Let D be the set of $4t$ points where the evaluations of $a(X)$ and $b(X)$ are released as part of the FRI colinearity checks. Let $a(D)$ be the set of evaluations of $a(X)$ on D, and let $b(D)$ be the set of evaluations of $b(X)$ on D. There is an invertible affine relation between the $8t+1$ points $a(D) \cup b(D) \cup \{a(\alpha - \tau)\}$ and the $8t+1$ coefficients of $k(X)$. As individual equations, this relation is given by

- $a(x) = a^\star(x - \tau_a) + f_{\mathsf{fd}}(x - \tau_a) \cdot k(x - \tau_a)$ for all $x \in D$;
- $a(\alpha + \tau) = a^\star(\alpha) + f_{\mathsf{fd}}(\alpha) \cdot k(\alpha)$; and
- $b(x) = b^\star(x - \tau_b) + f_{\mathsf{rp}}(x - \tau_b) \cdot k(x - \tau_b)$ for all $x \in D$.

As a single matrix equation it is:

$$\begin{pmatrix} \vdots \\ a(x) \\ \vdots \\ \hline a(\alpha+\tau) \\ \hline \vdots \\ b(x) \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ a^\star(x+\tau_a) \\ \vdots \\ \hline a^\star(\alpha) \\ \hline \vdots \\ b^\star(x+\tau_b) \\ \vdots \end{pmatrix} + \left(\begin{array}{ccc|c|cc} \ddots & & & & & \\ & f_{\mathsf{fd}}(x+\tau_a) & & & & \\ & & \ddots & & & \\ \hline & & & f_{\mathsf{fd}}(\alpha) & & \\ \hline & & & & \ddots & \\ & & & & & -f_{\mathsf{rp}}(x+\tau_b) \\ & & & & & & \ddots \end{array}\right) \begin{pmatrix} \vdots & \\ (x+\tau_a)^0 & \cdots \\ \vdots & \\ \hline \alpha^0 & \cdots \\ \hline \vdots & \\ (x+\tau_b)^0 & \cdots \\ \vdots & \end{pmatrix}$$

The rightmost matrix in this expression is Vandermonde except for 1 choice for τ_a and $4t + 1$ choices for τ_b. By the union bound, it is Vandermonde unless with probability $(4t + 2)/|\mathbb{F}|$.

The Vandermonde matrix is multiplied on the left by a diagonal matrix. The entries of this diagonal matrix are zero only if τ_a or τ_b coincides with one of the

roots of $f_{\mathsf{fd}}(X)$ or $f_{\mathsf{rp}}(X)$, respectively. The probability of this event is bounded by the Schwarz-Zippel lemma at $T/|\mathbb{F}|$.

By the union bound, the probability that the coefficient matrix is not invertible is at most $(12t + 2)/|\mathbb{F}|$. With all but this probability, the $8t + 1$ observed evaluations of $a(X)$ and $b(X)$ can be explained as the appropriate choice of $k(X)$.

The exception is $b(\alpha - \tau)$. However, this value cannot vary freely because it is constrained by the Bézout relation. Therefore it does not leak any new information. □