TIP 0001: Contiguity Argument for Memory Consistency

TIP	0001
authors:	Alan Szepieniec, Alexander Lemmens, and Ferdinand Sauer
title:	Contiguity Argument for Memory Consistency
status:	integrated
created:	2022-08-15
issue tracker:	https://github.com/TritonVM/triton-vm/issues/77
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 three base columns, four extension columns, four deterministic initial constraints, one randomized initial constraint, four deterministic transition constraints, four randomized transition constraints, and one randomized terminal constraint. The first base column and two deterministic transition constraints enable conditioning on a changed memory pointer. The second and third base columns and the other two deterministic transition constraints contain and constrain the symbolic Bézout coefficient polynomials' coefficients. 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 first. The second randomized transition constraint verifies the correct application of the product rule of differentiation to update this column. The third and fourth extension columns build up the Bézout coefficient polynomials based on the corresponding base columns. The remaining two randomized transition constraints enforce the correct build-up of the Bézout coefficient polynomials. The terminal constraint takes the weighted sum of the running product and the formal derivative, where the weights are the Bézout coefficient polynomials, 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 three base column and four extension columns:

- base column "difference inverse" `di`,
- base columns "Bézout coefficient polynomials' coefficients" `bcpc0` and `bcpc1`,
- extension column "running product" `rp`,
- extension column "formal derivative" `fd`, and
- extension columns "Bézout coefficients" `bc0` and `bc1`.

The following table illustrates the idea. Columns not involved in the proposal are not displayed.

`ramp`	`di`	`bcpc0`	`bcpc1`	`rp`	`fd`	`bc0`	`bc1`
a	0	0	ℓ	$(X-a)$	1	0	ℓ
a	$(b-a)^{-1}$	0	ℓ	$(X-a)$	1	0	ℓ
b	0	j	m	$(X-a)(X-b)$	$p(X)$	j	$\ell X+m$
b	0	j	m	$(X-a)(X-b)$	$p(X)$	j	$\ell X+m$
b	$(c-b)^{-1}$	j	m	$(X-a)(X-b)$	$p(X)$	j	$\ell X+m$
c	0	k	n	$(X-a)(X-b)(X-c)$	$q(X)$	$jX+k$	ℓX^2+mX+n
c	-	k	n	$(X-a)(X-b)(X-c)$	$q(X)$	$jX+k$	ℓX^2+mX+n

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 transition 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 $X - \mathsf{ramp}$ initially, which is enforced by an initial constraint. It accumulates a factor $X - \mathsf{ramp}^\star$ in every pair of rows where $\mathsf{ramp} \neq \mathsf{ramp}^\star$. This evolution corresponds to one transition constraint:

$$(\mathsf{ramp}^\star - \mathsf{ramp}) \cdot (\mathsf{rp}^\star - \mathsf{rp} \cdot (X - \mathsf{ramp}^\star)) + (1 - (\mathsf{ramp}^\star - \mathsf{ramp}) \cdot \mathsf{di}) \cdot (\mathsf{rp}^\star - \mathsf{rp}) \enspace .$$

Denote by $f_{\mathsf{rp}}(X)$ the polynomial that accumulates all factors $X - \mathsf{ramp}^\star$ in every pair of rows where $\mathsf{ramp} \neq \mathsf{ramp}^\star$.

The column `fd` contains the "formal derivative" of the running product with respect to X. The formal derivative is initially 1, which is enforced by an initial constraint. 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

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

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

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

The transition constraint for fd is

$$(\mathsf{ramp}^\star - \mathsf{ramp}) \cdot (\mathsf{fd}^\star - \mathsf{rp} - (X - \mathsf{ramp}^\star) \cdot \mathsf{fd}) + (1 - (\mathsf{ramp}^\star - \mathsf{ramp}) \cdot \mathsf{di}) \cdot (\mathsf{fd}^\star - \mathsf{fd}) \enspace .$$

Denote by $f_{\mathsf{fd}}(X)$ the formal derivative of $f_{\mathsf{rp}}(X)$.

The polynomials $f_{\mathsf{bc0}}(X)$ and $f_{\mathsf{bc1}}(X)$ are the Bézout coefficient polynomials satisfying the relation

$$f_{\mathsf{bc0}}(X) \cdot f_{\mathsf{rp}}(X) + f_{\mathsf{bc1}}(X) \cdot f_{\mathsf{fd}}(X) = \gcd(f_{\mathsf{rp}}(X), f_{\mathsf{fd}}(X)) \overset{!}{=} 1 \enspace .$$

The prover finds $f_{\mathsf{bc0}}(X)$ and $f_{\mathsf{bc1}}(X)$ as the minimal-degree Bézout coefficients as returned by the extended Euclidean algorithm. Concretely, the degree of

$f_{\mathsf{bc0}}(X)$ is smaller than the degree of $f_{\mathsf{fd}}(X)$, and the degree of $f_{\mathsf{bc1}}(X)$ is smaller than the degree of $f_{\mathsf{rp}}(X)$.

The (scalar) coefficients of the Bézout coefficient polynomials are recorded in base columns `bcpc0` and `bcpc1`, respectively. The transition constraints for these columns enforce that the value in one such column can only change if the memory pointer `ramp` changes. However, unlike the conditional update rule enforced by the transition constraints of `rp` and `fd`, the new value is unconstrained. Concretely, the two transition constraints are:

- $(1 - (\mathsf{ramp}^\star - \mathsf{ramp}) \cdot \mathsf{di}) \cdot (\mathsf{bcpc0}^\star - \mathsf{bcpc0})$
- $(1 - (\mathsf{ramp}^\star - \mathsf{ramp}) \cdot \mathsf{di}) \cdot (\mathsf{bcpc1}^\star - \mathsf{bcpc1})$

Additionally, `bcpc0` must initially be zero, which is enforced by an initial constraint. This upper-bounds the degrees of the Bézout coefficient polynomials, which are built from base columns `bcpc0` and `bcpc1`. Two transition constraints enforce the correct build-up of the Bézout coefficient polynomials:

- $(1-(\mathsf{ramp}^\star-\mathsf{ramp})\cdot\mathsf{di})\cdot(\mathsf{bc0}^\star-\mathsf{bc0})+(\mathsf{ramp}^\star-\mathsf{ramp})\cdot(\mathsf{bc0}^\star-X\cdot\mathsf{bc0}-\mathsf{bcpc0}^\star)$
- $(1-(\mathsf{ramp}^\star-\mathsf{ramp})\cdot\mathsf{di})\cdot(\mathsf{bc1}^\star-\mathsf{bc1})+(\mathsf{ramp}^\star-\mathsf{ramp})\cdot(\mathsf{bc1}^\star-X\cdot\mathsf{bc1}-\mathsf{bcpc1}^\star)$

Additionally, `bc0` must initially be zero, and `bc1` must initially equal `bcpc1`. This is enforced by initial constraints.

Lastly, the verifier verifies the randomized AIR terminal constraint

$$\mathsf{bc0} \cdot \mathsf{rp} + \mathsf{bc1} \cdot \mathsf{fd} = 1 \ \ .$$

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 $f_{\mathsf{bc0}}(X)$ and $f_{\mathsf{bc1}}(X)$ can exist such that $f_{\mathsf{bc0}}(X)\cdot f_{\mathsf{rp}}(X)+f_{\mathsf{bc1}}(X)\cdot f_{\mathsf{fd}}(X) = 1$. The verifier therefore probes unequal polynomials of degree at most $2T-2$. According to the Schwartz-Zippel lemma, the false positive probability is at most $(2T-2)/|\mathbb{F}|$. □

Zero-Knowledge. Since the contiguity argument is constructed using only AET and AIR, zero-knowledge follows from Triton VM's zk-STNARK engine. □

Summary of constraints

We present a summary of all constraints.

Initial

- $\mathsf{bcpc0}$
- $\mathsf{bc0}$
- $\mathsf{bc1} - \mathsf{bcpc1}$

- $\mathsf{fd} - 1$
- $\mathsf{rp} - (X - \mathsf{mp})$

Consistency

None.

Transition

- $(\mathsf{ramp}^\star - \mathsf{ramp}) \cdot ((\mathsf{ramp}^\star - \mathsf{ramp}) \cdot \mathsf{di} - 1)$
- $\mathsf{di} \cdot ((\mathsf{ramp}^\star - \mathsf{ramp}) \cdot \mathsf{di} - 1)$
- $(1 - (\mathsf{ramp}^\star - \mathsf{ramp}) \cdot \mathsf{di}) \cdot (\mathsf{bcpc0}^\star - \mathsf{bcpc0})$
- $(1 - (\mathsf{ramp}^\star - \mathsf{ramp}) \cdot \mathsf{di}) \cdot (\mathsf{bcpc1}^\star - \mathsf{bcpc1})$
- $(\mathsf{ramp}^\star - \mathsf{ramp}) \cdot (\mathsf{rp}^\star - \mathsf{rp} \cdot (X - \mathsf{ramp}^\star)) + (1 - (\mathsf{ramp}^\star - \mathsf{ramp}) \cdot \mathsf{di}) \cdot (\mathsf{rp}^\star - \mathsf{rp})$
- $(\mathsf{ramp}^\star - \mathsf{ramp}) \cdot (\mathsf{fd}^\star - \mathsf{rp} - (X - \mathsf{ramp}^\star) \cdot \mathsf{fd}) + (1 - (\mathsf{ramp}^\star - \mathsf{ramp}) \cdot \mathsf{di}) \cdot (\mathsf{fd}^\star - \mathsf{fd})$
- $(1 - (\mathsf{ramp}^\star - \mathsf{ramp}) \cdot \mathsf{di}) \cdot (\mathsf{bc0}^\star - \mathsf{bc0}) + (\mathsf{ramp}^\star - \mathsf{ramp}) \cdot (\mathsf{bc0}^\star - X \cdot \mathsf{bc0} - \mathsf{bcpc0}^\star)$
- $(1 - (\mathsf{ramp}^\star - \mathsf{ramp}) \cdot \mathsf{di}) \cdot (\mathsf{bc1}^\star - \mathsf{bc1}) + (\mathsf{ramp}^\star - \mathsf{ramp}) \cdot (\mathsf{bc1}^\star - X \cdot \mathsf{bc1} - \mathsf{bcpc1}^\star)$

Terminal

- $\mathsf{bc0} \cdot \mathsf{rp} + \mathsf{bc1} \cdot \mathsf{fd} - 1$