

Research Article

Exact Current-State Compilation of a History-Derived Parity in an A_5 -Seeded State-Transition System

Masaru Numagaki¹

1. Medcure Co., Ltd, Japan

This paper asks whether the cumulative value of a transition-derived observable can be represented exactly by the current state, without maintaining a dedicated event-log cache. We first define a minimal transition core consisting only of a state space, a move set, a legality predicate, and a next-state map. For a commutative target group M , a current-state map Φ and a transition label σ form an exact register when

$$\Phi(\text{Next}(s, m)) = \Phi(s) + \sigma(s, m)$$

for every legal transition. The resulting path sum telescopes to an endpoint difference; this elementary identity provides the framework for the explicit construction developed below.

We then specialize to $M = \mathbb{Z}_2$ in an A_5 -seeded D1b state-transition system. The general exact-register result does not select or require A_5 ; the D1b system is used here as a finite, source-auditable, nontrivial test case. From current cell-presence and face-incidence data we construct an explicit potential Φ , and prove that its one-step change equals an independently defined LOW/HIGH relative-side transition parity. Under a registered source certificate, a reachable HIGH-HIGH branch exhibits

$$\Phi_0 \longrightarrow \Phi_0 + 1 \longrightarrow \Phi_0,$$

which establishes certificate-conditional reachable non-vacuity. We further show that the register is recomputable from the current D1b residue and that a coherent dedicated cache contains no information independent of that residue for this observable.

The result is an exact current-state compilation theorem for one explicit history-derived parity. It does not imply reconstruction of the full event history or preservation of future behavior; additional scope boundaries are summarized in Appendix A.

Corresponding author: Masaru Numagaki, m.numagaki@medcure.co.jp

1. Introduction

1.1. Problem setting: is an event log necessary for a history-derived observable?

The fact that a quantity is cumulatively formed along a transition sequence is not the same as the fact that the transition sequence itself must be held as an independent memory object to obtain its current value.

Schematically, consider a legal transition sequence $s_0 \longrightarrow s_1 \longrightarrow \dots \longrightarrow s_n$. When each transition contributes to a certain observable, one way to maintain its cumulative value is to save and update each event in a separate log. On the other hand, it is possible that the information necessary for that observable is already incorporated into the current state s_n .

This paper studies that second possibility.

Here, *history-derived* means that the observable is defined cumulatively from transition labels along a legal history; it does not mean that the observable retains path identity once represented on the endpoint state.

Can we represent a single cumulative transition-derived observable exactly on the current state, and eliminate the need for a dedicated event-log cache for that observable?

The qualifier "for that observable" is essential. This paper does not claim that the current state stores the full history losslessly. Nor does it claim that the entire event sequence can be reconstructed from the current register.

The question is therefore one of observable-specific sufficiency: whether the current state alone is sufficient to evaluate a particular cumulative quantity.

1.2. History register and history-effective residue

We distinguish two notions:

History register:

A quantity that can be reconstructed from the current state, and whose value was formed by past transitions.

History-effective residue:

A quantity whose current-state differences change future legal language or reachable continuation.

Therefore,

$$\boxed{\text{history register} \Rightarrow \neg \text{history-effective residue}}$$

This paper establishes the former. Future-effective reachability is reserved for a companion study in preparation.

1.3. Why the A_5 -seeded D1b realization is used

The exact-register framework developed in Sections 2–4 is group-independent. It neither derives nor requires A_5 . The A_5 -seeded D1b system enters only as the concrete finite realization studied in the second half of the paper.

This choice is useful for three specific reasons. First, the D1b realization supplies a finite geometric move catalogue with an independently defined LOW/HIGH transition label. Second, its current residue contains explicit cell-presence and face-incidence data from which a candidate state potential can be constructed. Third, the frozen raw catalogue provides a source-auditable reachable branch on which non-vacuity can be tested without redesigning the dynamics after inspecting the register.

Accordingly, the paper does not argue that A_5 is physically necessary, unique, or empirically preferred. A different finite transition system satisfying the same abstract interface could support an analogous exact-register analysis. The role of A_5 here is therefore that of a concrete, nontrivial testbed rather than a premise of the general theorem.

1.4. Contributions of this paper

This paper makes four contributions.

First, it isolates a minimal transition interface and formulates an exact-register framework whose target is an arbitrary commutative group rather than $\mathbb{Z}_2$ from the outset.

Second, in the A_5 -seeded D1b realization, it constructs an explicit $\mathbb{Z}_2$ -valued current-state potential from cell-presence and face-incidence data and proves

$$\Phi(U(r, i)) = \Phi(r) + \sigma(i),$$

where $\sigma(i)$ is defined independently from the LOW/HIGH relative-side geometry.

Third, under a registered source certificate, it identifies a reachable HIGH–HIGH branch

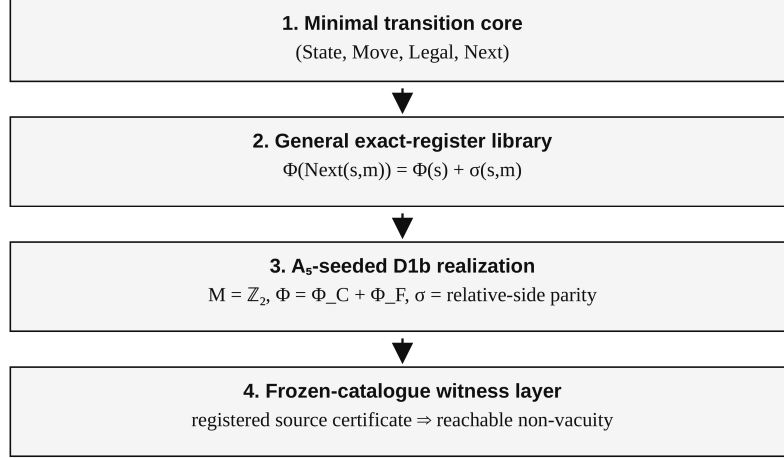
$$\Phi_0 \longrightarrow \Phi_0 + 1 \longrightarrow \Phi_0,$$

thereby establishing certificate-conditional reachable non-vacuity.

Fourth, it proves that the register can be recomputed from the current residue and that a coherent dedicated cached copy carries no information independent of that residue for this observable.

Architecture of the exact current-state compilation

General structure → concrete A_5 realization → certified witness



A_5 and the frozen catalogue enter only in the realization / witness layers.

Figure 1. Architecture of the exact current-state compilation. Layered structure of the construction. The minimal transition core consists only of state, move, legality, and next-state data. The general exact-register layer introduces a current-state potential Φ and transition increment σ satisfying $\Phi(\text{Next}(s, m)) = \Phi(s) + \sigma(s, m)$ on legal transitions. The A_5 -seeded D1b realization specializes the target to $\mathbb{Z}_2$ and constructs Φ from cell and face contributions. The frozen-catalogue layer is used only as a certificate-backed witness environment for reachable non-vacuity; it is not part of the general transition-core assumptions.

1.5. Related work and novelty positioning

At the abstract level, the exact-register identity is standard: when a transition increment is the coboundary of a state potential, its path sum depends only on the endpoints. This is familiar from elementary cochain language in algebraic topology [1]. Potential differences on transitions also appear in other contexts, such as potential-based reward transformations [2]. The paper therefore does *not* claim novelty for the generic telescoping lemma.

A second neighboring literature is state aggregation and lumpability. Classical lumpability and its later refinements ask when a quotient of a stochastic state space preserves specified transition or stationary behavior [3][4][5]. Related model-reduction work for Markov decision processes studies stronger state equivalences designed to preserve decision-relevant future behavior [6]. Computational mechanics goes further in a different direction by constructing causal states as minimal representations sufficient for prediction [7]. The present problem is deliberately weaker and different: Φ need only preserve the value of one cumulative observable. No quotient dynamics are constructed, and equality of Φ is not claimed to preserve future transition behavior. This distinction separates *observable-specific state sufficiency* from both *future-behavior-preserving aggregation* and *future-predictive sufficiency*.

The contribution of the present paper is the concrete integration of four ingredients in one source-auditable realization:

1. an independently defined LOW/HIGH transition label on the D1b move system;
2. a current-state potential built explicitly from cell presence and face incidence;
3. a reachable, source-grounded branch on which the register is non-constant; and
4. current-state recomputation together with coherent-cache equivalence, formalized in Lean.

The broader A_5 -substrate program provides context for why this finite realization was available [8], but that broader program is not a premise of the exact-register theorem.

2. Minimal transition core

2.1. Primitive data

We define the transition core by

$$\mathcal{H}_{\text{core}} = (\text{State}, \text{Move}, \text{Legal}, \text{Next}).$$

$\text{Legal}(s, m)$ means that move m is legal in state s , and $\text{Next} : \text{State} \times \text{Move} \rightarrow \text{State}$ returns the next state for a state-move pair.

A legal transition is written $s \xrightarrow{m} s'$ and means $\text{Legal}(s, m)$ together with $s' = \text{Next}(s, m)$.

The transition core does not include a readout, a history register, or a distinguished observable as primitive data.

2.2. Finite legal paths

Let $s_0 \xrightarrow{m_1} s_1 \xrightarrow{m_2} \dots \xrightarrow{m_n} s_n$ be a legal path of length n .

The move list can be used to describe a path mathematically, but it is not assumed to be stored as an event log within the current state s_n .

Therefore, we separate

how a state was reached

and

what can be reconstructed from the reached state

2.3. Autonomy as an interface choice

In this paper, we define $s_k = \text{Next}(s_{k-1}, m_k)$ and do not use the full previous event sequence as a primitive argument of Next.

This is not an assertion that all physical systems are Markovian. The current state itself may have a complex structure formed by past transitions.

Furthermore, global determinism is not required. There may be multiple legal moves for a given s .

2.4. What the transition core does not assume

The transition core does not assume

$$A_5, 2I, \mathbb{Z}_2,$$

the existence of a current-state readout, finite local readability, a finite/global/closed catalogue, insertion-only growth, Avail, strict depletion, three-dimensionality, energy, Genesis, or eternal/terminal history.

Thus,

$$\boxed{\mathcal{H}_{\text{core}} \Rightarrow \neg \text{global finiteness}} \quad \text{and} \quad \boxed{\mathcal{H}_{\text{core}} \Rightarrow \neg \text{terminality}}.$$

2.5. Core primitives and theorem library

$$\boxed{\text{Core primitives} \neq \text{Core theorem library}}$$

There are only four primitive components. The register target, readout, transition label, exactness relation, and cache equivalence are constructed later.

3. Exact Current-State Register

3.1. Registers on a general commutative group

Let M be a commutative group, and consider $\Phi : \text{State} \rightarrow M, \sigma : \text{State} \times \text{Move} \rightarrow M$.

Φ is the current-state register, and σ is the transition increment.

3.2. Definition of Exact Register

For every legal transition, if

$$\boxed{\Phi(\text{Next}(s, m)) = \Phi(s) + \sigma(s, m)},$$

then (Φ, σ) is called an exact current-state register.

Equivalently, $\Delta_m \Phi(s) = \sigma(s, m)$.

3.3. General Exact-Register Lemma

For a legal path

$$s_0 \xrightarrow{m_1} s_1 \rightarrow \cdots \xrightarrow{m_n} s_n,$$

we have

$$\boxed{\sum_{k=1}^n \sigma(s_{k-1}, m_k) = \Phi(s_n) - \Phi(s_0)}.$$

Proof. We induct on n . For $n = 0$, the left side is the empty sum 0, while the right side is $\Phi(s_0) - \Phi(s_0) = 0$. Assume the statement holds for length n . For the final move,

$$s_{n+1} = \text{Next}(s_n, m_{n+1}), \text{Legal}(s_n, m_{n+1}),$$

so exactness gives

$$\sigma(s_n, m_{n+1}) = \Phi(s_{n+1}) - \Phi(s_n).$$

Since M is commutative,

$$\begin{aligned}\sum_{k=1}^{n+1} \sigma(s_{k-1}, m_k) &= (\Phi(s_n) - \Phi(s_0)) + (\Phi(s_{n+1}) - \Phi(s_n)) \\ &= \Phi(s_{n+1}) - \Phi(s_0).\end{aligned}$$

□

This is a telescoping identity and is not claimed as the main novelty of the paper.

3.4. What Exactness Means

Exactness does not mean full path reconstruction.

Furthermore,

$$\boxed{\text{exact register} \neq \text{conserved quantity}}$$

and

$$\boxed{\text{exact current-state register} \Rightarrow \neg \text{history-effective residue}}.$$

3.5. Cochain language

If an appropriate transition complex is fixed separately, the exact-register relation $\Phi(\text{Next}(s, m)) = \Phi(s) + \sigma(s, m)$ can be expressed in cochain terminology as $\sigma = \delta\Phi$.

In this paper, the transition complex itself is not defined as a primitive, nor is the calculation of the cohomology class part of the argument. This cochain representation is not equated with the central-extension degree-2 structure in Paper I.

4. Cached Register Redundancy

4.1. Coherent cached state

For $\Phi : \text{State} \rightarrow M$, define

$$\text{CachedState}_\Phi = \{(s, b) \mid b = \Phi(s)\}.$$

Projection $\pi(s, b) = s$ and reconstruction $\iota(s) = (s, \Phi(s))$ are inverses of each other.

Therefore,

$$\boxed{\text{CachedState}_\Phi \simeq \text{State}}$$

4.2. Meaning

A coherent cached copy b does not add any independent information to the underlying state s . This equivalence is specific to the chosen register.

4.3. Cached update

For an exact register, define

$$\text{updB}(b; s, m) = b + \sigma(s, m).$$

Then

$$\boxed{\text{updB}(\Phi(s); s, m) = \Phi(\text{Next}(s, m))}.$$

5. Analysis Discipline

The concrete analysis follows a freeze-before-diagnosis rule. The D1b catalogue, legality predicate, and update map are fixed before Φ is evaluated, and they are not modified in response to the observed register values. Thus, the logical direction is

$$\text{fixed transition system} \longrightarrow \text{diagnostic register},$$

not the reverse. This is a methodological guard against target leakage, not an axiom of the abstract transition core. Build and provenance statuses are reported separately in the Formal Provenance section.

6. A_5 -Seeded D1b Realization

6.1. Concrete realization

We now specialize the minimal transition core to an A_5 -seeded D1b state-transition system. The A_5 substrate identifies the finite geometric catalogue used by this realization; it is not part of the definition of the general exact-register framework. No empirical claim is made that this catalogue is a model of a particular detector or microscopic physical process.

6.2. Notation and local transition picture

Only a small part of the D1b machinery is needed in this paper. Table 1 summarizes the notation.

Symbol	Meaning
r	current D1b residue, containing the present-cell set and current face-incidence classes
c_i	candidate cell associated with move slot i
f_i	designated frontier face through which c_i is attached
$U(r, i)$	residue obtained by the legal insertion i
LOW/HIGH	relative-side class of the selected cell–face pair
$\sigma(i)$	$\mathbb{Z}_2$ -valued transition parity, with LOW = 0 and HIGH = 1
$\Phi(r)$	current-state exact register constructed in Section 7

Table 1. Notation used in the D1b realization.

The local transition picture is therefore

$$\begin{aligned} \text{current residue } r &\longrightarrow \text{frontier face } f_i \longrightarrow \text{legal insertion } c_i \\ &\longrightarrow \text{LOW/HIGH label} \longrightarrow \sigma(i). \end{aligned}$$

6.3. D1b cell and face data

A D1b cell c has a base address and three selected directions. Its codimension-one faces inherit corresponding face data. For a face f , the current incidence class in residue r is denoted by

$$\text{inc}_r(f).$$

6.4. Current residue

For the purposes of this paper, the current residue r contains at least:

- the set of currently present cells; and
- the current incidence class of each relevant face.

The complete insertion history is not a primitive field of r .

6.5. Legal insertion

A move i inserts the candidate cell c_i , producing

$$r \xrightarrow{i} U(r, i).$$

Within the realization studied here, legality requires an exact one-face attachment through the designated frontier face f_i . This is a property of the D1b realization, not an assumption of the general transition core.

6.6. Relative-side transition label

Each legal insertion carries an independently defined label

$$\sigma(i) \in \mathbb{Z}_2, \text{LOW} \mapsto 0, \text{HIGH} \mapsto 1.$$

The label is determined by the relative-side geometry of the selected cell–face pair; it is not defined from the potential Φ .

6.7. Role of $\mathbb{Z}_2$

Section 3 allows an arbitrary commutative target group M . The choice

$$M = \mathbb{Z}_2$$

belongs specifically to the present D1b realization.

6.8. Role of the frozen catalogue

The frozen catalogue provides a finite witness environment for the concrete non-vacuity result. It is not assumed by the general exact-register theorem, and neither one-step availability nor strict catalogue depletion is required for the explicit construction of Φ .

7. Explicit $\mathbb{Z}_2$ Potential

7.1. Fixed parity data

The parity data are fixed directly from the six integer base coordinates, independently of the transition label. For a base address $b : \text{Fin } 6 \rightarrow \mathbb{Z}$, define

$$p_{\text{base}}(b) := \sum_{j: \text{Fin } 6} (b(j) : \mathbb{Z}_2).$$

For a cell c and a face f , set

$$p_C(c) := p_{\text{base}}(c.\text{base}), p_F(f) := p_{\text{base}}(f.\text{base}).$$

These are the source-faithful definitions used in the Lean realization. In particular, p_C and p_F are determined by base geometry before λ or σ is evaluated; they are not chosen to fit the transition labels.

7.2. Cell contribution

Let $\mathcal{C}(r)$ be the set of present cells and define

$$\Phi_C(r) = \sum_{c \in \mathcal{C}(r)} p_C(c).$$

7.3. Face contribution

In the Lean realization, face incidence is stored in discrete classes, and we select the `twoOrMore` class.

For exposition, define

$$\mathcal{F}_{2+}(r) = \{f \mid \text{inc}_r(f) = \text{twoOrMore}\}, \Phi_F(r) = \sum_{f \in \mathcal{F}_{2+}(r)} p_F(f).$$

This corresponds to the shorthand “incidence ≥ 2 ” used in the prose, but the source-faithful condition is equality with the `twoOrMore` class.

7.4. Explicit potential

$$\Phi(r) = \Phi_C(r) + \Phi_F(r).$$

Equivalently,

$$\Phi(r) = \sum_{\text{present } c} p_C(c) + \sum_{\text{inc}_r(f) = \text{twoOrMore}} p_F(f).$$

This Φ is computed from the current residue alone. In the formal source, the cell term is implemented as a finite sum conditioned by `cellPresent = true`, and the face term as a finite sum conditioned by `faceIncidence = twoOrMore`; the displayed formula therefore matches the Lean definition.

7.5. One-step change

The change in Φ due to legal insertion i is divided into cell terms and face terms.

7.5.1. Cell term

By legality, the inserted cell c_i is absent before insertion and present after the update. The presence of other catalogue cells does not change with this single action. Therefore,

$$\Delta_i \Phi_C = p_C(c_i).$$

7.5.2. Face term

Let f_i be the frontier face. In the formal legality projection, before insertion, $\text{inc}_r(f_i) = \text{one}$, and furthermore, there are no occupied faces for the other faces of the new cell c_i . Therefore, the incidence class of those other faces is zero before insertion.

Under the residue update, the frontier face changes as $f_i : \text{one} \rightarrow \text{twoOrMore}$, while the other faces of the new cell change as $\text{zero} \rightarrow \text{one}$. The incidence class of faces that do not belong to the new cell remains unchanged. Moreover, legality places the frontier face in class one and every other face of c_i in class zero before insertion, so no face of the inserted cell is already in `twoOrMore`. Thus no term already counted in Φ_F leaves the sum during this update.

Φ_F sums parity only over faces in the `twoOrMore` class. Hence the only face newly entering Φ_F in this step is the frontier face f_i , and

$$\Delta_i \Phi_F = p_F(f_i).$$

Therefore,

$$\Delta_i \Phi = p_C(c_i) + p_F(f_i).$$

7.5.3. Independent geometric decomposition of relative-side labels

The D1b relative-side move label $\sigma(i)$ is not a quantity defined from Φ . It is defined independently as a move label based on the relative-side signature of upstream dynamics, taking 0 if the selected face is LOW / base-aligned, and 1 if it is HIGH / non-base-aligned.

For a cell c and face f , define the geometry label

$$\lambda(c, f) = \begin{cases} 0, & f.\text{base} = c.\text{base}, \\ 1, & \text{otherwise}, \end{cases} \in \mathbb{Z}_2.$$

In the formal D1b realization, whenever f is a genuine face of c , the parity-gap theorem gives

$$\boxed{\lambda(c, f) = p_C(c) + p_F(f)}.$$

For a LOW face the two base parities agree, giving 0; for a HIGH face the face base is shifted by one unit, giving 1.

The relative-side move label of catalogue move i agrees with the geometry label of its selected pair (c_i, f_i) . Hence

$$\boxed{\sigma(i) = p_C(c_i) + p_F(f_i)}.$$

Here, the general exact-register interface allows σ as the state-move quantity $\sigma(s, m)$, but the move label of this concrete D1b realization is determined from the catalogue K and move slot i , and does not depend on the current residue r . Therefore, in this section, it will be abbreviated as $\sigma(i)$.

7.6. Explicit Potential Integration Theorem

For a D1b legal insertion i , the following holds:

$$\boxed{\Delta_i \Phi = \sigma(i)}$$

Indeed, Sections 7.5.1–7.5.2 give

$$\Delta_i \Phi = p_C(c_i) + p_F(f_i),$$

while Section 7.5.3 gives

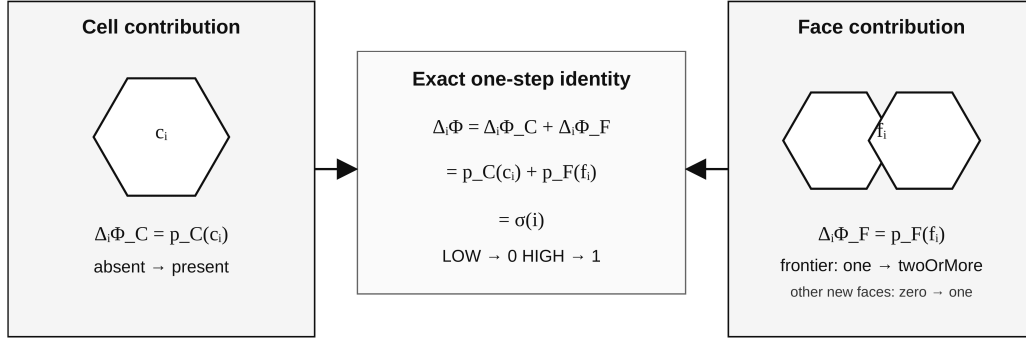
$$\sigma(i) = p_C(c_i) + p_F(f_i).$$

Thus the two quantities coincide. This is the explicit exact-register theorem used below.

The corresponding Lean source implements the same cell term, face term, parity-gap identity, one-step integration theorem, and legal-word endpoint formula. Formal source identity and build status are summarized later rather than used as substitutes for the argument above.

One-step D1b potential decomposition

Schematic: the figure shows the theorem structure, not a literal embedding of the D1b geometry.



Independent geometric label $\sigma(i)$ is matched by the potential difference.

Figure 2. One-step decomposition of the D1b current-state potential. Schematic representation of the explicit one-step integration theorem. For a legal insertion i , the change in the cell contribution is $\Delta_i\Phi_C = p_C(c_i)$, while the change in the face contribution is $\Delta_i\Phi_F = p_F(f_i)$. The designated frontier face changes from incidence class one to twoOrMore, whereas the other newly introduced faces change from zero to one. Hence $\Delta_i\Phi = p_C(c_i) + p_F(f_i) = \sigma(i)$. The drawing is schematic and represents the theorem structure rather than a literal geometric embedding of the D1b cells.

7.7. Exactness and non-vacuity

This theorem alone does not imply that Φ is non-constant on the reachable state space.

exactness $\neq$ non-vacuity

8. Endpoint Reconstruction

8.1. Path accumulation

For a legal insertion sequence

$$r_0 \xrightarrow{i_1} r_1 \rightarrow \cdots \xrightarrow{i_n} r_n,$$

define

$$\Sigma_{\sigma}(w) = \sum_{k=1}^n \sigma(i_k).$$

8.2. Endpoint formula

By the one-step theorem of Section 7,

$$\boxed{\Sigma_{\sigma}(w) = \Phi(r_n) - \Phi(r_0)}.$$

Therefore, given the start/end state, the cumulative parity can be calculated without replaying the intermediate event sequence.

8.3. Path independence

For legal paths w_1, w_2 with the same start/end,

$$\boxed{\Sigma_{\sigma}(w_1) = \Sigma_{\sigma}(w_2)}$$

Proof. Applying the endpoint formula in Section 8.2 to both paths gives

$$\Sigma_{\sigma}(w_1) = \Phi(r_{\text{end}}) - \Phi(r_{\text{start}}) = \Sigma_{\sigma}(w_2).$$

□

However, $w_1 = w_2$ does not follow from this equality; path identity is not reconstructed.

8.4. Closed loop

In a legally closed loop,

$$\boxed{\Sigma_{\sigma} = 0}$$

Indeed, for a closed loop the endpoint equals the start point, so Section 8.2 yields

$$\Sigma_{\sigma} = \Phi(r_0) - \Phi(r_0) = 0.$$

Thus this is a direct consequence of the exact-register condition; no cochain terminology is required.

9. Certificate-Conditional Reachable Non-Vacuity

Exactness alone does not show that Φ changes on any reachable branch. The concrete non-vacuity result therefore uses a registered source certificate. The certificate contributes source identity and finite source-geometry facts; it does not alter the transition law.

9.1. Unified source facts

For the authoritative frozen catalogue, the unified certificate packages five facts:

1. (C1) row 165 identifies the initial witness cell c_{D0} ;
2. (C2) row 0 identifies the first inserted cell c_1 ;
3. (C3) row 394 identifies the second inserted cell c_2 ;
4. (C4) the row-0 cell is unique in the catalogue; and
5. (C5) in the corrected row0-dir2-HIGH incident family, no candidate other than the catalogue cell itself occurs in the indexed catalogue.

The first three facts identify the two-step witness; the last two supply the source-level uniqueness and finite incidence exclusion needed by the source-closure construction.

9.2. Worked branch from the frozen catalogue

The three registered catalogue cells are

$$\begin{aligned} \text{row 165: } &((-2, -1, -1, -1, 0, 1), \{0, 2, 4\}), \\ \text{row 0: } &((-2, -2, -1, -1, 1, 1), \{0, 1, 2\}), \\ \text{row 394: } &((-1, -2, -1, -1, 1, 0), \{1, 2, 5\}). \end{aligned}$$

Under the registered certificate, the source-closure theorem identifies the legal two-step branch

$$r_0 \xrightarrow{i_1} r_1 \xrightarrow{i_2} r_2$$

with relative-side labels

$$\sigma(i_1) = 1, \sigma(i_2) = 1.$$

Applying the one-step integration theorem twice gives

$$\Phi(r_1) = \Phi(r_0) + 1, \Phi(r_2) = \Phi(r_1) + 1 = \Phi(r_0) + 2$$

in $\mathbb{Z}_2$. Hence

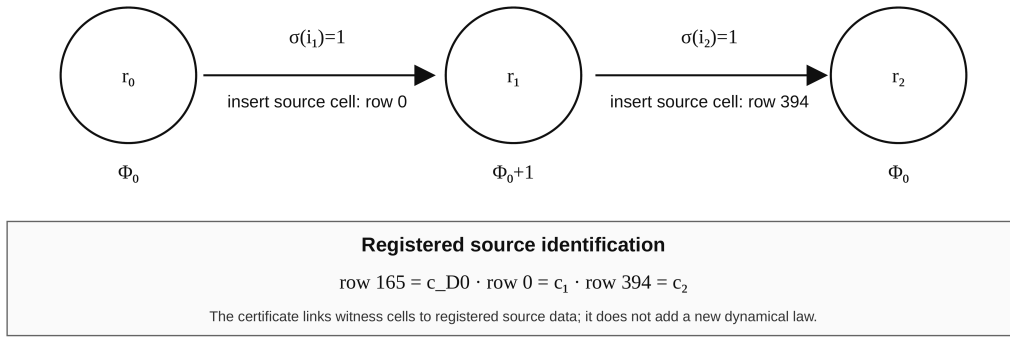
$$\boxed{\Phi_0 \longrightarrow \Phi_0 + 1 \longrightarrow \Phi_0}.$$

In particular, $\Phi(r_1) \neq \Phi(r_0)$, so the register is non-constant on a reachable branch. This is the concrete non-vacuity statement used in the paper.

The recurrence $\Phi(r_2) = \Phi(r_0)$ does *not* imply $r_2 = r_0$, path equality, or equality of future behavior. It is recurrence of one register value only.

Certified two-step HIGH–HIGH branch

Catalogue row identities are source-cell identifiers; r_0, r_1, r_2 are states.



Reachable non-vacuity: Φ changes after the first legal HIGH step.

Figure 3. Certificate-conditional two-step HIGH–HIGH branch. A certified reachable branch

$r_0 \xrightarrow{i_1} r_1 \xrightarrow{i_2} r_2$ with $\sigma(i_1) = \sigma(i_2) = 1$ gives $\Phi(r_0) \rightarrow \Phi(r_0) + 1 \rightarrow \Phi(r_0)$. Thus the exact register is non-constant on the certified reachable branch, although its value returns after two HIGH transitions. The catalogue row numbers identify the registered source cells used by the witness certificate and are distinct from the dynamical states r_0, r_1, r_2 . The certificate provides source provenance for the witness cells rather than an additional dynamical law.

9.3. Certificate parameter

The non-vacuity theorem takes the registered source certificate as an explicit parameter:

$$\boxed{\text{registered source certificate} \implies \text{reachable non-vacuity}}$$

The raw-source provenance of that certificate is recorded separately in Section 14 and Supplementary Provenance Record S1.

10. Current-State Readout and Cache Redundancy

Define the readout

$$B(r) := \Phi(r).$$

Because Φ is computed from the present-cell set, present face-incidence classes, and fixed parity data, $B(r)$ is a function of the current residue alone.

For a hypothetical dedicated cache value b , define

$$\text{updB}(b, i) = b + \sigma(i).$$

Then the one-step integration theorem gives

$$B(U(r, i)) = \Phi(U(r, i)) = \Phi(r) + \sigma(i) = \text{updB}(B(r), i).$$

Thus recomputing the register after an update agrees exactly with sequentially updating a dedicated cache.

This redundancy can be stated without appealing to a formal identifier. Let

$$\text{Cached}(K) = \{(r, b) : b = B(r)\}.$$

Projection $(r, b) \mapsto r$ and reconstruction $r \mapsto (r, B(r))$ are inverse maps. Therefore

$$\boxed{\text{Cached}(K) \simeq \text{Residue}(K)}.$$

A coherent cache dedicated to this parity adds no information independent of the current residue.

Current-state readout and coherent-cache redundancy

The commuting square expresses recompute-after-update = cached-register update.

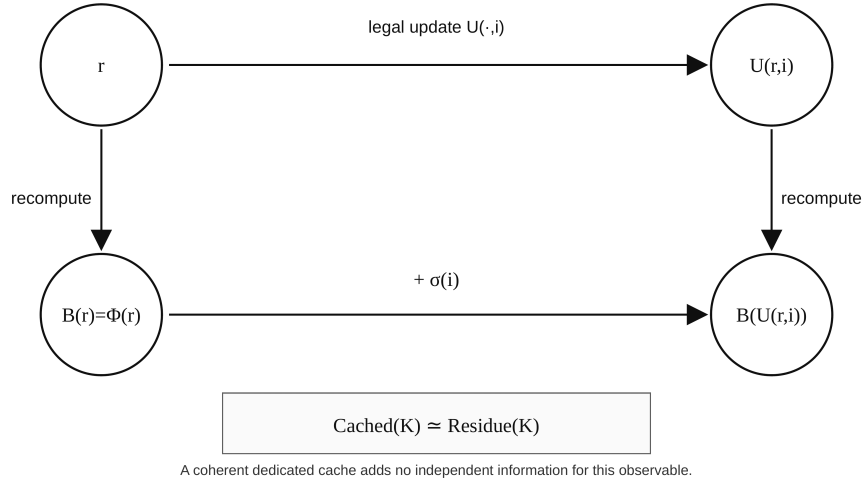


Figure 4. Current-state readout and coherent-cache redundancy. Commuting diagram for the autonomous readout $B(r) = \Phi(r)$. Recomputing the register after a legal state update agrees with updating a coherent cached register by the transition increment: $B(U(r, i)) = \text{updB}(B(r), i) = B(r) + \sigma(i)$. Consequently, a coherent dedicated cache carries no information independent of the underlying current residue for this observable. This equivalence concerns the specified parity register only and does not imply reconstruction or storage of the full event history.

11. Observable-Specific State Sufficiency and Adjacent Notions

11.1. State summary versus full history

Let a full finite history be $h = (i_1, \dots, i_n)$. The history may encode move identities, order, path length, and intermediate states, whereas $\Phi(r_n) \in \mathbb{Z}_2$ is a single binary observable. Hence

$$h \not\equiv \Phi(r_n).$$

What the exact-register theorem establishes is only observable-specific sufficiency:

$$\sum_{k=1}^n \sigma(i_k) = \Phi(r_n) - \Phi(r_0).$$

Thus the current endpoint is sufficient to recover this cumulative parity once the initial register is fixed. Here, “history compilation” means that a past-derived observable becomes rereadable from the current

state.

11.2. Relation to state aggregation and lumpability

State aggregation asks a stronger question: whether multiple states can be quotiented while preserving specified transition or probabilistic behavior. Classical lumpability, later proportional variants, and state-equivalence/model-minimization methods provide preservation results for such reductions in Markovian or decision-process settings [3][4][5][6]. Predictive-state constructions such as causal states impose an even stronger sufficiency target: retaining exactly the information needed for future prediction [7]. The present register defines neither kind of reduction. In particular,

$$\Phi(r) = \Phi(r') \Rightarrow \mathcal{L}(r) = \mathcal{L}(r'),$$

where $\mathcal{L}(r)$ denotes the set of legal finite continuations from r . Therefore the present result should be read as a theorem about one observable, not as a future-behavior-preserving state reduction. Determining a coarsest future-sufficient quotient is a separate problem.

11.3. Relation to the central-extension residue

The register Φ and the central-extension residue $\varepsilon_{\text{hist}}$ studied in the companion manuscript [9] are different constructions. The present exact register is a state potential satisfying a one-step coboundary relation. The companion construction instead uses the nontrivial central extension

$$1 \rightarrow C_2 \rightarrow 2I \rightarrow A_5 \rightarrow 1.$$

No theorem in this paper identifies the two. In particular,

$$\boxed{\Phi \neq \varepsilon_{\text{hist}}}.$$

No bridge between the two constructions is asserted here.

12. Formal Provenance and Verification Status

12.1. Status summary

The formal development uses Lean 4 [10] and Mathlib [11].

The formal record distinguishes three notions. **Author-reported BUILD VERIFIED** means that the supplied module or package was reported by the author to build successfully in the stated Lean/Mathlib

environment. **STATIC AUDITED** records source identity, hashes, imports, and the absence or presence of placeholders such as `sorry`, `axiom`, and `admit`. **HYPOTHESIS-PARAMETERIZED** means that the theorem remains conditional on an explicit certificate argument.

Table 2 summarizes the four formal assets directly supporting the present paper. The mathematical steps used in the main claims are written explicitly in the manuscript; module names are reported here for traceability rather than used in place of proofs.

Asset	Mathematical role	Current status
Exact potential	explicit current-state potential and one-step integration	author-reported BUILD VERIFIED; static audit finds no <code>sorry</code> , <code>axiom</code> , <code>admit</code> , or <code>native_decide</code>
Reachable witness	certificate-conditional reachable non-vacuity	author-reported BUILD VERIFIED; HYPOTHESIS-PARAMETERIZED; no <code>native_decide</code> in the module
Readout/cache	current-state readout and coherent-cache equivalence	author-reported BUILD VERIFIED; no <code>native_decide</code> in the module
Actual-ref2 source closure	unified source certificate and projection $C \rightarrow R$	author-reported BUILD VERIFIED; $C \rightarrow R$ formalized; raw payload $\rightarrow W \rightarrow C$ remains an open ingestion bridge

Table 2. Formal assets directly supporting the paper. Detailed hashes and line counts are given in Supplementary Provenance Record S1.

12.2. Raw-source audit and the remaining ingestion bridge

The frozen source package used for this manuscript is `A5_History_lc(20260919-002908).zip`, reported by the author as BUILD VERIFIED on 2026-09-19. A static audit of the package records no `sorry`, `axiom`, or `admit`. The package contains 42 uses of `native_decide` in older finite-certificate modules, but none of the four modules directly supporting this paper in Table 2 uses `native_decide`.

For the authoritative raw catalogue `t7_ref2_tiles.json`, included with the supplementary material and matching the frozen digests recorded in Supplementary Provenance Record S1, a separate exhaustive raw-data audit verifies the factual fields used by the unified certificate: 691 rows are distinct; rows 0, 165, and 394 match the required cells; row 0 is unique; and the seven non-catalogue members of the corrected

row0-dir2-HIGH incident family are absent. The audit logic is frozen in the accompanying `verify_ref2_unified_certificate.py`, and a machine-readable summary is supplied as `REF2_RAW_UNIFIED_CERTIFICATE_AUDIT.json`. These raw-data checks are distinct from the still-open Lean ingestion bridge described below.

The remaining formal gate is not a mathematical assumption about the dynamics but an ingestion/provenance bridge:

$$\text{authoritative raw payload} \longrightarrow W \longrightarrow C.$$

Here W denotes a `DifferenceBarrierD1bRef2ExactImportWitnessV1` instance and C denotes a `W20ActualRef2UnifiedSourceCertificateV1` instance over that witness. The projection $C \rightarrow R$, where R has type `W20ActualRef2Legal2SourceRowsV1`, is already formalized. Until the raw-payload-to- W/C bridge is itself frozen and build-verified, the reachable non-vacuity claim remains certificate-conditional.

12.3. Known reproducibility limitations

The frozen source ZIP is a source package rather than a self-contained Lake snapshot: the complete Mathlib commit, `lean-toolchain`, `lake-manifest.json`, and exact archived build log are not bundled in the ZIP. These limitations affect archival reproducibility, not the algebraic derivation of the exact-register theorem. Detailed module hashes, raw-data digests, and environment records are collected in Supplementary Provenance Record S1.

Lean checks derivations inside the defined model; physical realization is a separate question.

13. Discussion

13.1. What the theorem adds beyond the telescoping identity

The abstract telescoping lemma is elementary. The nontrivial content of the case study is that an independently defined geometric transition label can be integrated by an explicit potential on the current D1b residue, and that the resulting register is demonstrably non-constant on a reachable source-grounded branch. This is what turns the general identity into a concrete current-state compilation result rather than a restatement of a coboundary formula.

13.2. Why the finite D1b example is informative

The D1b realization serves as a finite stress test with three useful properties: the transition label is fixed independently of the potential, the potential is computed from explicit present-state geometry, and the raw catalogue supplies a finite branch whose source provenance can be audited. The example therefore exposes where the argument could fail: the parity-gap identity could fail, the potential could be constant on all reachable states, or the readout could require information absent from the current residue. None of those failures occurs in the certified branch considered here.

13.3. Limitations and generalization target

The present result is restricted to one $\mathbb{Z}_2$ -valued observable and one finite realization. A natural generalization problem is to characterize when a transition-labelled system admits a current-state potential at all, and when a collection of such potentials is sufficient for a broader family of observables. A different, stronger direction is to ask for quotients that preserve future behavior; that direction connects more directly to lumpability and bisimulation and is not solved here.

14. Conclusion

This paper isolates a narrow question: can the current value of one cumulative transition-derived parity be recovered exactly from the current state without maintaining a dedicated event-log cache for that observable?

For a general transition system, the exact-register relation

$$\Phi(\text{Next}(s, m)) = \Phi(s) + \sigma(s, m)$$

implies endpoint reconstruction of the cumulative transition label. In the concrete A_5 -seeded D1b realization, an explicit $\mathbb{Z}_2$ -valued potential constructed from current cell-presence and face-incidence data integrates an independently defined LOW/HIGH transition parity. A registered source certificate supplies a reachable HIGH–HIGH branch on which the register changes and then returns, establishing non-vacuity. The same register is recomputable from the current residue, and a coherent dedicated cache is therefore redundant for this observable.

Thus one explicit, non-constant history-derived parity admits exact current-state compilation in this finite realization. The claim boundaries are summarized in Appendix A.

Appendix A. Claim and Scope Firewalls

Established or used	Not established or not identified
A current-state exact register Φ for one explicit D1b parity observable	reconstruction of the full event history
Endpoint reconstruction of the cumulative transition parity	recovery of path identity from the register
Current-state readability and coherent-cache redundancy	preservation of future transition behavior
Certificate-conditional reachable non-vacuity	unconditional raw-payload-to-Lean certificate instantiation
Φ as a present-state ledger/readout coordinate	identification with the companion central-extension residue $\varepsilon_{\text{hist}}$
Finite frozen catalogue as a concrete witness environment	a finite catalogue requirement for the general transition core
Externally audited raw-source facts	a fully self-contained archival Lake snapshot

Table 3. Compact claim firewall for the present paper.

Statements and Declarations

Funding

This research received no external funding.

Potential Competing Interests

The author declares no competing interests.

Ethics

Not applicable. This work is theoretical and formal and does not involve human participants, animals, clinical data, or identifiable personal data.

Data and Code Availability

No experimental or participant data were generated. The supplementary material accompanying this submission includes the frozen Lean source package used for the reported formal status, the authoritative raw catalogue `t7_ref2_tiles.json`, the raw-catalogue audit verifier, the machine-readable audit certificate, a saved verifier output, and Supplementary Provenance Record S1. The included raw catalogue is identified in S1 by its frozen filename and cryptographic digests. A fully self-contained archival Lake snapshot that additionally bundles the complete toolchain metadata and exact build log has not yet been frozen; this limitation is stated explicitly in the Formal Provenance section.

Author Contributions

M. Numagaki is the sole author and is responsible for the conception of the study, the mathematical analysis, the D1b exact-register construction, the formalization strategy, provenance decisions, interpretation of the results, and the writing and revision of the manuscript.

Use of Generative AI

OpenAI ChatGPT was used during manuscript preparation for English translation and copyediting, LaTeX formatting, organization and readability improvements, and assistance in drafting or revising formal-code text. The mathematical claims, source identities, formal-proof acceptance decisions, build-status assertions, scientific interpretations, and conclusions were reviewed and approved by the author. The author takes full responsibility for the manuscript and the accompanying formal source package.

Acknowledgments

The author thanks the Lean 4 and Mathlib developer communities for the open-source formal tools used in this work.

References

1. [△]Hatcher A (2002). *Algebraic Topology*. Cambridge: Cambridge University Press.
2. [△]Ng A, Harada D, Russell S (1999). "Policy Invariance under Reward Transformations: Theory and Application to Reward Shaping." Morgan Kaufmann.

3. ^a ^bKemeny J, Snell J (1976). *Finite Markov Chains: With a New Appendix "Generalization of a Fundamental Matrix"*. Springer-Verlag.
4. ^a ^bMarin A, Rossi S (2017). "On the Relations Between Markov Chain Lumpability and Reversibility." *Acta Inform.* **54**(5):447–485. doi:[10.1007/s00236-016-0266-1](https://doi.org/10.1007/s00236-016-0266-1).
5. ^a ^bMarin A, Piazza C, Rossi S (2022). "Proportional Lumpability and Proportional Bisimilarity." *Acta Inform.* **59**:211–244. doi:[10.1007/s00236-021-00404-y](https://doi.org/10.1007/s00236-021-00404-y).
6. ^a ^bGivan R, Dean T, Greig M (2003). "Equivalence Notions and Model Minimization in Markov Decision Processes." *Artif Intell.* **147**(1–2):163–223. doi:[10.1016/S0004-3702\(02\)00376-4](https://doi.org/10.1016/S0004-3702(02)00376-4).
7. ^a ^bShalizi C, Crutchfield J (2001). "Computational Mechanics: Pattern and Prediction, Structure and Simplicity." *J Stat Phys.* **104**(3–4):817–879. doi:[10.1023/A:1010388907793](https://doi.org/10.1023/A:1010388907793).
8. ^aNumagaki M (2026). "Non-Solvability as a Physical Principle: $\mathcal{A}_5$ and the Algebraic Origin of Information Barriers." 2026 Preprint.
9. ^aNumagaki M (2026). "Admissible History Deformations and Central-Extension Residues: An $\mathcal{A}_5/2$ Case Study." 2026 Companion manuscript in preparation.
10. ^ade Moura L, Ullrich S (2021). "The Lean 4 Theorem Prover and Programming Language." Springer. doi:[10.1007/978-3-030-79876-5_37](https://doi.org/10.1007/978-3-030-79876-5_37).
11. ^aThe mathlib Community (2020). "The Lean Mathematical Library." Association for Computing Machinery. doi:[10.1145/3372885.3373824](https://doi.org/10.1145/3372885.3373824).

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.