From Distinction to Loop Order: A Concept Guide

Distinction

In plain words. Noticing that two things are not the same. That is all: no counting, no adding, no repeating.

Precisely. For a carrier K, the statement $\mathrm{Dist}(K)$ says that there exist $x, y \in K$ with $x \neq y$. A proof of it is a distinction witness.

Example. The two sides of a coin are distinct. Knowing this alone gives you no way to count coins.

One-step extension $r \mapsto Sr$

In plain words. The one move used to build the first carrier: take the record you have and add one more mark.

Precisely. 0 is the empty record. If r is a record, then Sr is a record. The symbol δ is the name of this one-step extension.

Example. Cut one more notch into a stick. The stick is the record; the new cut is one more application of S.

δ-orbit (the generated records)

In plain words. Everything reached from the empty record by repeating the one-step extension: no mark, one mark, two marks, and so on. The important point is that this collection is generated. It is not selected from a larger set of numbers which was already assumed.

Precisely. $\mathrm{Orb}(0, S)$ is generated by the two rules "0 is a record" and "if r is a record, then Sr is a record". With addition and multiplication attached, the arithmetic structure is written $\mathbb{N}_\delta$.

$$0 \xrightarrow{S} S0 \xrightarrow{S} SS0 \xrightarrow{S} SSS0 \xrightarrow{S} SSSS0 \qquad \cdots$$

Figure 1. One rule starts the orbit and one rule continues it.

Example. Three notches is the record $SSS0$. It is not "the number 3 picked out of $\mathbb{N}$". It is the result of doing the act three times.

Successor laws and induction

In plain words. Three facts come with the generated orbit. A record with one more mark is never the blank record. If two records become equal after one more mark, they were already equal. If a property holds at 0 and stays true after one more step, then it holds for every generated record.

Precisely. $Sr \neq 0$, $Sr = Sq \Rightarrow r = q$, together with structural induction. In the first paper these are proved from the recursion and induction principles of the inductively generated orbit.

δ-algebra

In plain words. Any system where you can point to a start and you have a "do one more" button. It need not behave like counting.

Precisely. A triple $A = (|A|, 0_A, S_A)$ with $0_A \in |A|$ and $S_A : |A| \to |A|$.

Example. A light switch is a δ-algebra: start = off, button = flip. Pressing twice returns to off. This is allowed.

Initiality

In plain words. For every "start plus one-more button" system, there is exactly one structure-preserving way to send every generated record into that system.

Precisely. For every δ-algebra A there is a unique homomorphism

$$\mathrm{rec}_A : \mathbb{N}_\delta \longrightarrow A.$$

Example. For the light switch, $0, SS0, SSSS0, \ldots$ go to off, while $S0, SSS0, \ldots$ go to on. The map is unique, but it is far from injective.

Rigidity

In plain words. If the button never returns to the start and never merges two different states, the canonical map loses no information. If the system also has induction for all predicates, then it is the generated orbit, up to a unique relabelling. So the Peano conditions do not build a number system. They describe the one obtained by repeating a single act.

Precisely. If S_A is injective and $0_A \notin \mathrm{im}(S_A)$, then rec_A is injective. If A also satisfies induction for all predicates, rec_A is the unique isomorphism $\mathbb{N}_\delta \cong A$.

Careful. This is a theorem inside a metatheory which already supports inductive generation. It is a uniqueness statement, not a claim that arithmetic exists without background assumptions.

Monoid

In plain words. A way of combining things with a "do nothing" element. Changing the brackets does not change the result.

Precisely. A set with an associative binary operation and a neutral element.

Example. Notched sticks with addition and the blank stick form $(\mathbb{N}_\delta, +, 0)$. Strings with concatenation and the empty string are another example.

Commutative and noncommutative

In plain words. Commutative means that changing the order of two pieces does not change the result. Noncommutative means that sometimes it does.

Example. Three notches followed by two gives the same total as two followed by three. Putting on socks and then shoes is not the same process as doing it in the reverse order.

Recognizer

In plain words. A summary device. It reads an object and reports a value, and it respects the way objects are combined.

Precisely. A monoid homomorphism $r : M \to N$. The target need not be finite, which is more general than the usual finite-automaton convention.

Example. "Even number of notches or odd number of notches" is a recognizer into a two-element monoid.

Kernel congruence and recognition quotient

In plain words. Two objects are indistinguishable to a recognizer when it gives them the same value. The quotient is the world seen by that recognizer.

Precisely.

$$x \sim_r y \quad \iff \quad r(x) = r(y),$$

and $M/{\sim_r}$ is isomorphic to the image of r.

Example. For parity, all even records are in one class and all odd records are in the other.

Index-period congruence $\equiv_{i,p}$ and $M(i, p)$

In plain words. An odometer which counts correctly for a while and then enters a loop. It remembers an initial tail of length i and then only the position in a cycle of length p.

Precisely. $x \equiv_{i,p} y$ if $x = y$, or if both counts are at least i and are congruent modulo p. The quotient $M(i, p)$ has $i + p$ elements and is generated by one element.

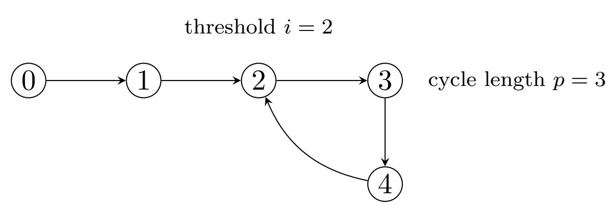

Figure 2. The lasso shape of $M(2, 3)$.

Example. A clock face is $M(0, 12)$. The summary "zero, one, or many" is $M(2, 1)$.

Classification of recognition quotients

In plain words. Counting leaves almost no room for interesting summaries. A recognizer either loses nothing at all, or it has the single lasso shape above: honest counting up to a threshold, then position in a cycle. There is no third possibility, and the reason is simple. A record is nothing but a pile of identical steps, so there is no order information for a summary to pick up.

Precisely. Under excluded middle, every congruence on $(\mathbb{N}_\delta, +, 0)$ is either equality or $\equiv_{i,p}$ for a unique $i \geq 0$ and $p \geq 1$. Equivalently, every recognizer is either injective or has image $M(i, p)$.

Constructive (intuitionistic) reasoning

In plain words. A style of proof where "there exists" should give a way to find or construct the object. In general, it is not enough to say that non-existence would be impossible.

Example. Classically, "either every object in an infinite list fails a test, or some object passes" is automatic. Constructively, the second alternative should come with a witness.

Decidable

In plain words. There is a test which always stops with yes or no.

Example. Two finite records have the same number of notches: compare them step by step.

Excluded middle (EM)

In plain words. The permission to say "either P holds or P does not hold" for any proposition P, even when there is no decision procedure for P.

Markov's principle (MP)

In plain words. For a decidable test, if it is impossible that the test never succeeds, MP allows us to conclude that it succeeds somewhere.

Example. If a decidable congruence is known not to be equality, MP is what extracts an actual distinct related pair when only double-negated existence is available.

Limited principle of omniscience (LPO)

In plain words. For a decidable sequence of tests: either one succeeds somewhere, or none ever succeeds. It sits between EM and MP. In the first paper it is tracked in the ledger but has no separate role in the final classification price.

Logical price, and what "exact" means

In plain words. How much nonconstructive logic is needed for a theorem. The interesting half is not the upper bound but the matching lower bound: the paper shows that from the classification one can *derive the permission back out*. So the price is not an artifact of the proof that happened to be chosen. It is the real cost.

Precisely. For the recognition-quotient classification there are three levels: an explicit distinct related pair needs no extra nonconstructive principle; a decidable nontrivial congruence has exact price MP; an arbitrary congruence has exact price EM.

Example. If someone hands you a collapsed pair, a bounded search finds the threshold and period. If you know only that a decidable device is not faithful, no pair is in your hand, and MP is exactly what produces one. If the congruence is arbitrary, even the split "equality or not" requires EM.

The ledger

In plain words. A receipt attached to a formal derivation. It records which nonconstructive principles were used inside the δ-calculus.

Precisely. The ledger is a subset of $\{\mathrm{EM}, \mathrm{LPO}, \mathrm{MP}, \mathrm{QInd}\}$. A derivation is forced when EM, LPO and MP are absent. QInd is recorded but does not affect forcedness.

Careful. The ledger and the logical price are different. The ledger is syntactic and belongs to derivations inside the calculus. The logical price belongs to proofs in the surrounding metatheory. Any summary which merges the two is reporting something the papers do not say.

Soundness of the forced fragment

In plain words. If a closed theorem has a forced derivation, then it is true for the ordinary natural numbers.

Precisely. Every closed formula derivable in the forced fragment from the empty context holds in $\mathfrak{N} = (\mathbb{N}, 0, S, +, \cdot)$.

The number tower $\delta \rightsquigarrow \mathbb{N}_\delta \hookrightarrow \mathbb{Z}_\delta \hookrightarrow \mathbb{Q}_\delta$

In plain words. Negatives are made from pairs "credit, debit". Fractions are made from numerator-denominator pairs. Both steps are explicit recipes.

Precisely.

$$\mathbb{Z}_\delta = (\mathbb{N}_\delta \times \mathbb{N}_\delta)/{\sim}, \qquad (a, b) \sim (c, d) \iff a + d = c + b.$$

The rational stage uses signed-orbit numerators and nonzero denominators with cross-equality. Both stages are choice-free, and $\mathbb{Z}_\delta \cong \mathbb{Z}$, $\mathbb{Q}_\delta \cong \mathbb{Q}$.

Example. $(2, 5)$ and $(4, 7)$ describe the same integer because $2 + 7 = 4 + 5$. Both describe -3.

δ-encodable

In plain words. A set is δ-encodable if its elements can be given different labels from the generated natural-number carrier.

Precisely. There is an injection $X \hookrightarrow \mathbb{N}_\delta$. The first paper constructs such injections for $\mathbb{N}$, $\mathbb{Z}$ and $\mathbb{Q}$.

Careful. Encodable means that labels fit. It does not say that the arithmetic structures are identical.

Choice-free

In plain words. No step of the construction asks us to make an arbitrary choice from infinitely many possible representatives.

Example. For rational numbers, normal forms can be obtained algorithmically; no arbitrary representative is selected.

Why the tower stops at the rationals

In plain words. The real-number stage is a new constructive problem. The usual Cauchy and Dedekind constructions need not agree under weak constructive assumptions. For this reason, the first paper stops at $\mathbb{Q}_\delta$.

Bridge to Part II. The first carrier is generated by one repeated act. It records how many times the act is repeated, but it has no independent order information, and that is exactly why its summaries collapse to the single lasso shape. The second paper keeps the same recognition idea and changes only the carrier. It uses closed walks with several independent loops. Now order can matter, and the question becomes which readings can see it.

1 Part II: from one act to loop order

1.1 The board you are playing on

Graph and 1-skeleton

In plain words. Dots joined by lines: like a road map. The 1-skeleton of a cube keeps only the corners and edges. The solid inside is not used.

Hypercube Q_n and the three-cube Q_3

In plain words. Q_n has one vertex for each binary string of length n. Two vertices are joined when they differ in exactly one coordinate.

Example. Q_2 is a square. Q_3 is the ordinary cube with eight vertices, twelve edges and three coordinate directions.

Based closed walk

In plain words. A round trip. Start at the fixed home corner, move along edges, and come back to the same corner. An edge may be used more than once, and you may turn back immediately.

Precisely. A walk is written $w = (o; a_1 \cdots a_n)$ with $a_k \in \{0, 1, 2\}$ on Q_3, where a_k is the coordinate flipped at step k. It is closed when the final vertex equals the basepoint o. Its length is $|w| = n$.

Example. $(000; 0101)$ goes around one square face and returns to 000.

When is a walk closed?

In plain words. Exactly when every coordinate direction is used an even number of times. Every use flips that coordinate, so an even number of flips is needed to return it.

The carrier $\mathcal{W}$

In plain words. All based closed walks, with "do this trip and then that trip" as multiplication. The empty trip is the do-nothing element. This monoid replaces the notched sticks of Part I.

First-return loop

In plain words. A round trip which touches home only at the start and the end. Every closed walk splits uniquely into such pieces by cutting at every return home.

Precisely. $\mathcal{W}$ is the free monoid on the set L of first-return loops. A monoid recognizer on $\mathcal{W}$ is therefore determined by arbitrary values on L: the exact analogue of the one-generator case, with L in place of the single generator.

Example. $(000; 00101101)$ cuts as $(00)(101101)$: one short there-and-back and one longer first-return loop.

Spanning tree and cotree edges

In plain words. Choose enough roads to reach every vertex, but do not make a loop. This gives a spanning tree. In Q_3 it has seven edges. The other five edges are cotree edges, and each one closes a loop, so they give five independent loop generators.

Precisely. The paper fixes the Gray-code tree

$$000, 001, 011, 010, 110, 111, 101, 100.$$

The five cotree edges are labelled $x_1, \ldots, x_5$.

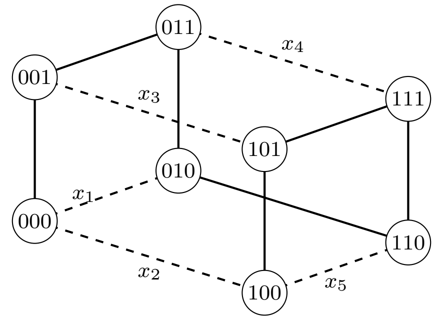

Figure 3. Solid edges form the chosen tree. Dashed edges are the five cotree edges.

Reduced loop word ρ

In plain words. Follow the trip and write a letter only when a cotree edge is crossed. One direction gives x_i, and the opposite direction gives x_i^{-1}. Tree edges write nothing. Then cancel neighbouring inverse pairs. The word which remains keeps the order of the loop crossings, not merely how many there were.

Precisely. $\rho : \mathcal{W} \to F_5$ is a monoid homomorphism into the free group of rank five: the word of a concatenation is the concatenation of the words.

Example. This is bracket matching, exactly. Read x_i as an opening bracket of type i and x_i^{-1} as the matching closing bracket; cancellation is the matching of an adjacent pair. This is why the memory questions further down have the answers they do.

Free group F_5

In plain words. Words in five letters and their inverses, with no relations except cancellation of a neighbouring letter and its inverse. "Free" means that nothing else is declared equal.

Precisely. The fundamental group of the 1-skeleton of Q_3 is F_5 because the cycle rank is $12-8+1 = 5$. For Q_2 the rank is 1, so its fundamental group is $\mathbb{Z}$, which is commutative. Thus Q_3 is the first hypercube where noncommutative loop order can appear.

Abelianization and signed cycle counts a

In plain words. Forget the order and keep only the net tally for each of the five generators: crossings one way minus crossings the other way.

Precisely.

$$a = \mathrm{ab} \circ \rho : \mathcal{W} \longrightarrow \mathbb{Z}^5.$$

The condition $a(w) = 0$ means zero net oriented flow.

Example. The commutator word

$$x_1 x_2 x_1^{-1} x_2^{-1}$$

is not empty, but its signed tally is $(0, 0, 0, 0, 0)$. Order survives in ρ and dies in a.

Directed transition counts τ

In plain words. For every directed edge, count how many times the trip used it. This gives the full road tally, but it forgets the order of the trip.

Precisely.

$$\tau_w(p,q) = \#\{k : (v_{k-1}, v_k) = (p,q)\}, \qquad \tau_{uv} = \tau_u + \tau_v.$$

Example. A shop receipt tells you what you bought and how many of each item, but not the order in which you put the items in the basket.

Additive one-step reading

In plain words. Put a weight on each directed road and add the weights along the trip. Since addition is commutative, such a reading cannot recover the order in which roads were used. This is the general form of an additive reading which accumulates values step by step.

Precisely. For a weight g into an abelian group,

$$R_g(w) = \sum_k g(v_{k-1}, v_k).$$

Every such reading factors through τ_w. So $\tau_u = \tau_v$ forces $R_g(u) = R_g(v)$ for all weights at once.

Careful. The theorem is about additive readings. It does not say that no function of a trip can see order. Some readings can, and one appears two entries below.

Potential reading

In plain words. Give every vertex a height and add the height changes along the trip. On a closed trip all changes cancel.

Precisely. $R_\phi(w) = \phi(v_n) - \phi(v_0)$, so $R_\phi(w) = 0$ on every closed walk.

1.2 Order is real: the AB versus BA pair

The square faces A, B and the pair AB, BA

In plain words. Take two based loops around two adjacent square faces. Do A then B, or B then A. The same directed roads are used the same number of times, but the reduced loop words are different.

Precisely. Let

$$A = (000; 0101), \qquad B = (000; 1212).$$

Then $\tau_{AB} = \tau_{BA}$, while

$$\rho(AB) = x_5^{-1} x_2^{-1}, \qquad \rho(BA) = x_1 x_5^{-1} x_2^{-1} x_1^{-1}.$$

Example. The totals can agree while the history does not. This is the first simple witness that additive accumulation is blind to loop order.

The hierarchy $\Delta_{\mathcal{W}} \subsetneq \sim_\rho \subsetneq \sim_a$

In plain words. Three levels of blurring. Equality forgets nothing. The reduced loop word forgets immediate tree backtracking. Abelianization forgets still more: it also forgets order. So ρ sits strictly between the two, discarding something but keeping strictly more than any tally.

Precisely.

$$\Delta_{\mathcal{W}} \subsetneq \sim_\rho \subsetneq \sim_a .$$

Example. The trip $(000;00)$ steps out along a tree edge and straight back. It is not the do-nothing trip, yet its reduced word is empty. That single example is what makes the left inclusion strict.

ρ and τ are incomparable

In plain words. Neither the loop word nor the road tally determines the other. The AB, BA pair has one tally and two words; the trip $(000;00)$ has one word with the do-nothing trip but a different tally.

1.3 Memory

Sequential reading (clock)

In plain words. A machine with an internal state. It starts in one state, every directed edge updates the state, and at the end an output is read. Counters, flags and automata are all of this kind.

Precisely. A state set S, initial state s_0, updates $T_e : S \to S$, and output map q. On edges $e_1, \ldots, e_n$ the value is

$$q(T_{e_n} \circ \cdots \circ T_{e_1}(s_0)).$$

Careful. A clock need not be a monoid recognizer: its value on a concatenation need not be built from the two separate values. The term "recognition quotient" is reserved for genuine monoid homomorphisms, and a clock is never quietly promoted to one.

k-state fold

In plain words. A sequential reading with at most k states. This means a fixed finite amount of memory.

One flag is enough for AB versus BA

In plain words. Finite memory is not useless. One bit can remember which of the two face loops was seen last, using one road private to each face. So it separates AB from BA, although every additive reading fails on this pair.

Why every fixed memory bound fails: the $k!$ trick

In plain words. Feed the same block repeatedly to a machine with only k states. Eventually a state repeats, because there is nowhere else to go, and from then on the machine cycles with period at most k. Every such period divides $k!$. Therefore $k!$ and $2k!$ repetitions leave it in the same state, although their reduced loop words differ. Enlarge the memory and the same argument runs again with a larger factorial.

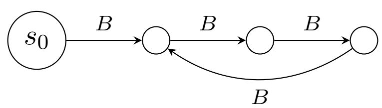

Figure 4. Repeating one block forces every finite-state machine into a cycle.

Equal-count obstruction

In plain words. The stronger pair has the same length *and* the same complete directed transition counts. So every additive one-step reading also gives the same value, and handing the machine the exact tally for free does not help it.

Precisely. With the based loops

$$A = (000; 0101), \qquad C = (000; 01020102),$$

whose reduced words are x_1^{-1} and x_2^{-1}, put

$$U_k = A^{k!}C^{k!}A^{2k!}, \qquad V_k = A^{2k!}C^{k!}A^{k!}.$$

Both have length $20k!$ and the same transition counts $3k!\tau_A + k!\tau_C$, and every sequential reading with at most k states identifies them, from every initial state. Their reduced words

$$x_1^{-k!}x_2^{-k!}x_1^{-2k!} \qquad \text{and} \qquad x_1^{-2k!}x_2^{-k!}x_1^{-k!}$$

already differ at the $(k!+1)$-st letter.

Conjugacy and free homotopy

In plain words. For loops without a fixed starting point, cyclically shifting the starting point should not change the loop. In a free group this is represented by conjugacy.

Precisely. Conjugate reduced words represent the same free-homotopy class. The pair U_k, V_k is conjugate, so a sceptic can object that they are one loop read from two starting points. The paper answers that objection with the next pair.

The nonconjugate pair

In plain words. Two histories with the same length and the same counts, which every k-state reading identifies, and whose reduced words are not even conjugate. So the obstruction is not an artifact of insisting on a basepoint: it survives after the basepoint is forgotten.

Precisely.

$$P_k = A^{k!}CA^{3k!}C, \qquad Q_k = A^{2k!}CA^{2k!}C,$$

both of length $16k! + 16$, with the same transition counts and nonconjugate reduced words.

The trace trick

In plain words. Send the reduced words to 2×2 matrices and add the two diagonal entries. Conjugate matrices have the same trace, so different traces prove that no cyclic shift can match the words.

Precisely. Send $x_1^{-1} \mapsto X$, $x_2^{-1} \mapsto Y$ and the remaining generators to the identity, where

$$X = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \qquad Y = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \qquad \operatorname{tr}(X^r Y X^s Y) = 2 + 2(r+s) + rs.$$

For P_k and Q_k this gives

$$2 + 8k! + 3(k!)^2 \qquad \text{and} \qquad 2 + 8k! + 4(k!)^2.$$

The stack recovers everything

In plain words. Give the machine a store which may grow, arranged as a pile. Keep the current reduced word on it: push each new letter, except pop when the new letter cancels the top one. At the end the pile *is* the loop word. One procedure, every history, exact answer.

Precisely. An unbounded stack computes $\rho(w)$ exactly for every $w \in \mathcal{W}$. Together with the finite-memory results this locates the real boundary. It is not finite versus infinite information; it is bounded state versus a store you are allowed to grow.

Example. The same idea checks arbitrarily deep matching brackets. No fixed-size checklist handles every depth; one pile of paper handles all of them.

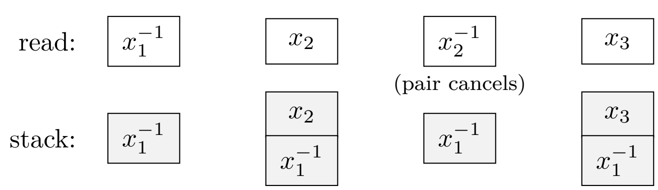

Figure 5. Free reduction run as a stack: push, or pop when the new letter meets its own inverse on top.

Reading	AB vs BA	U_k vs V_k	P_k vs Q_k
additive one-step reading	no	no	no
full directed tally τ	no	no	no
some fixed finite-state machine	yes	no	no
unbounded stack ρ	yes	yes	yes

Table 1. Which reading sees which pair. "Yes" means the reading separates that pair.

1.4 How short can a hidden loop be?

Commutator and lower central series

In plain words. A commutator measures failure to commute: do g, then h, then undo g, then undo h. If order never mattered you would be back home. The lower central series is a ladder of deeper levels of noncommutativity.

Precisely.

$$[g,h] = ghg^{-1}h^{-1}, \qquad \gamma_1(F_5) = F_5, \qquad \gamma_{n+1}(F_5) = [\gamma_n(F_5), F_5].$$

For a walk, $\rho(w) \in \gamma_2$ exactly when $a(w) = 0$, and $\rho(w) \notin \gamma_3$ says that the first order-sensitive measurement is nonzero.

The degree-two coordinate μ_{ij}

In plain words. The simplest measurement of order, and one which can stay nonzero after all the ordinary signed counts vanish. Walk the trip keeping a running count of the i-crossings; each time you cross j, add the current i-count, and symmetrically with a minus sign. Two trips with the same tally but different interleaving get different answers. Geometrically it behaves like a signed area.

Precisely. If $s^{(k)}$ is the signed cotree step at time k and $a^{(k-1)}$ is the running exponent-sum vector, then

$$\mu_{ij}(w) = \sum_k \Big(a_i^{(k-1)} s_j^{(k)} - a_j^{(k-1)} s_i^{(k)}\Big).$$

It is invariant under free reduction, so it depends only on $\rho(w)$. When $a(w) = 0$, $\mu(w)/2$ gives the class in γ_2/γ_3.

The fourteen-step witness

In plain words. A closed walk with zero abelianization but with a nontrivial reduced loop word. Its signed cycle tally is zero, and every antisymmetric additive reading gives zero, but the reduced loop word is still nontrivial: the trip cannot be shrunk to nothing inside the road network. It has fourteen steps, and no shorter example exists on any hypercube of dimension at least three.

Precisely.

$$W = (000; 01012020101202),$$

with

$$a(W) = 0, \qquad \rho(W) = x_1^{-1}x_2x_3^{-1}x_1x_3x_2^{-1} \neq 1, \qquad \mu_{12}(W) = -2.$$

It arises as the commutator of two adjacent square faces, after one immediate there-and-back is deleted.

step	1	2	3	4	5	6	7	8	9	10	11	12	13	14
direction	0	1	0	1	2	0	2	0	1	0	1	2	0	2
letter				x_1^{-1}	x_2		x_3^{-1}		x_1			x_3		x_2^{-1}

Figure 6. Fourteen steps, but only six cotree letters survive in the word.

Why fourteen?

In plain words. Three steps. First, a nontrivial edge-neutral loop cannot fit in a network with only one independent cycle, since there the tally already determines the loop; so its roads carry at least two. Second, six roads will not do: two cycles on six roads need five vertices, and the only such shape joins two vertices by three separate paths, which no cube contains, since two corners have at most two common neighbours. Hence at least seven roads. Third, zero net flow forces every used road to be driven equally often both ways, so at least twice. Seven roads, twice each: fourteen, attained by the witness above.

Precisely. The forbidden shape is $K_{2,3}$. For every $n \geq 3$,

$$\min\{|w| : a_n(w) = 0,\ \rho_n(w) \neq 1\} = \min\{|w| : \rho_n(w) \in \gamma_2 \setminus \gamma_3\} = 14.$$

The argument uses only that the graph is bipartite and contains no $K_{2,3}$, and both hold for every hypercube. The two finite-memory obstructions also carry over to every Q_n with $n \geq 3$.

Relation to abelian girth

In plain words. The shortest nontrivial edge-neutral closed walk is an instance of the graph invariant called abelian girth. The paper proves the hypercube value directly and also shows that a shortest witness already has nonzero degree-two commutator information.

Does the answer depend on the spanning tree?

In plain words. No. Changing the spanning tree changes the coordinates used to write the free group, but it does not change whether a loop is trivial or its position in the lower central series. The number fourteen belongs to the graph, not to the chosen labels. Only the individual value $\mu_{12}(W) = -2$ refers to the particular tree fixed in the paper.

Why the three-cube?

In plain words. Two reasons, and it is worth being exact about them. Q_3 is not the smallest graph with noncommuting loop words: a four-vertex graph with five edges already has two independent cycles. But Q_3 is the first *hypercube* with that property, which is what continues the one-generator story of the first paper, and its edges split into three coordinate directions, which is what the declared clock below uses.

Careful. "Three coordinate directions" here is a feature of the cube used as a combinatorial board. It is not a statement about the dimension of physical space, and neither paper makes one.

1.5 A second, different way to see order

The declared quarter-turn clock $c = (\tau, h)$

In plain words. Attach one noncommuting quaternion factor to each coordinate direction and multiply them in the order in which the directions are used. Combine that result with the directed transition counts. Since the quaternion factors do not commute, the product can see order.

Precisely. With $q_0 = 1 + i$, $q_1 = 1 + j$, $q_2 = 1 + k$,

$$h(w) = q_{a_1} \cdots q_{a_n}, \qquad c(w) = (\tau_w, h(w)).$$

This gives a monoid homomorphism and

$$\Delta_{\mathcal{W}} \subsetneq \sim_c \subsetneq \sim_\tau .$$

Careful. The word *declared* is important. This clock is extra structure chosen and announced by the model. It is not forced by the reduced loop word, not derived from the walks, and not determined by anything earlier in the paper. It also does not factor through loops at all: an immediate there-and-back contributes $(1 + i)^2 = 2i \neq 1$, so it is sensitive to backtracking, while ρ is not.

Incomparable readings

In plain words. Neither the quaternion clock nor the reduced loop word contains all the information in the other. Each separates a pair the other merges. So being order-sensitive is not a single ladder with a best rung.

Precisely. $\sim_c$ and $\sim_\rho$ are incomparable. The quaternion clock separates

$$(000; 001122) \qquad \text{and} \qquad (000; 002211),$$

which ρ cannot see because both are built from pure backtracks, while the reduced word separates AB from BA, which the declared clock merges.

The 24-element group $2T$

In plain words. After the common size factor is removed, the quaternion part takes only 24 values on closed walks. Those values form the binary tetrahedral group, so this part of the reading is finite.

Precisely. The normalized quaternion component has 24 possible values on closed walks; the sequential computation can be implemented with 48 prefix states.

Careful. The numbers 24 and 48, and the three coordinate directions, are features of this declared combinatorial clock. Nothing here identifies physical space, and nothing here should be quoted as a dimension count.

1.6 Constraints that couple a state to a history

Declared states $\mathcal{A}$

In plain words. Alongside a history we allow a separately declared quantity: one integer attached to each vertex of the cube. This is the "state" half of the pairs studied below.

Precisely. $\mathcal{A} = \mathbb{Z}^V$, the integer-valued functions on the vertex set. A constraint is a set of pairs $(\alpha, w) \in \mathcal{A} \times \mathcal{W}$.

Rectangular (projection-separable) constraint

In plain words. A rule on a pair (state, history) is rectangular when it can be checked by two independent tests: one form to fill in about the state, one about the history, with no cross-references. If it is not rectangular, the two must be compared against each other.

Precisely. On $\mathcal{A} \times \mathcal{W}$, a constraint is rectangular when

$$P = P_{\mathcal{A}} \times P_{\mathcal{W}}.$$

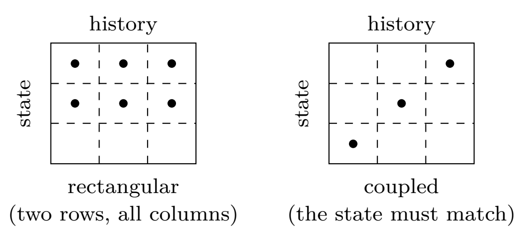

Figure 7. A product-shaped rule versus a diagonal matching rule. On the right, allowing one row and one column would force the off-diagonal corner, which the rule forbids.

Occupation vector occ and occupation binding

In plain words. For a trip, count how many times it occupies each vertex. The binding rule says that the declared state must equal that count vector. This couples state and history directly, and is the standard example of a rule which does not split.

Precisely. With $\mathrm{occ}(w) \in \mathcal{A}$ the occupation vector, the binding constraint $\mathcal{O} \subseteq \mathcal{A} \times \mathcal{W}$ is

$$(\alpha, w) \in \mathcal{O} \quad \Longleftrightarrow \quad \alpha = \mathrm{occ}(w).$$

The exact rectangularity criterion

In plain words. On a fixed class of histories, every occupation-based rule splits into separate state and history conditions exactly when all histories in the class have the same occupation vector. One condition, and it works in both directions.

Precisely. For a class $\mathcal{S} \subseteq \mathcal{W}$, the following are equivalent: the occupation vector is constant on $\mathcal{S}$; every occupation-based constraint is rectangular on $\mathcal{A} \times \mathcal{S}$.

Example. If every allowed history visits the same eight cube vertices with the same multiplicities, occupation carries no further information inside that class and occupation-based constraints factor.

2 Notation

Symbol	Reads as
δ	the act: extend a record by one step
$\mathrm{Dist}(K)$	two elements of K are unequal; a distinction witness
$0, Sr$	empty record; the record r extended once
$\mathbb{N}_\delta, \mathbb{Z}_\delta, \mathbb{Q}_\delta$	the generated naturals, integers and rationals
$\mathfrak{N}$	the standard model $(\mathbb{N}, 0, S, +, \cdot)$
$M(i, p)$	the odometer with threshold i and period p
$\equiv_{i,p}$	indistinguishable to that index-period quotient
EM, LPO, MP	three logical principles tracked by the work, strongest first
Q_n, Q_3	the n-cube graph; the three-cube
$\mathcal{W}$	all based closed walks from the fixed corner
L	the first-return loops; $\mathcal{W}$ is free on L
ρ	reduced loop word
F_5	free group on the five cotree letters
$a = \mathrm{ab} \circ \rho$	signed cycle tally; order deleted
τ	full directed transition tally
μ_{ij}	degree-two order coordinate
γ_2, γ_3	lower-central-series levels
$c = (\tau, h)$	the declared quaternion clock
$2T$	the 24-element normalized quaternion group
$\mathcal{A} = \mathbb{Z}^V$	declared states: one integer per vertex
occ	occupation vector
$\mathcal{O}$	the occupation-binding constraint
$\Delta_{\mathcal{W}}$	equality of histories
$\sim_r$	"the reading r cannot tell these apart"

Index