ACF 89 — Lean 4 Proof Dossier

The Chrysene Formalism as Candidate Mathematics and Candidate Biogenesis

A Quantitative and Categorical Survey of Volume 1, Number 2

Charles D. Schaper, Ph.D., Schaper Systems Press

Annals of the Chrysene Formalism, Vol. 1, Issue 4, November 2026, first page 1808

Proof file: `acf-89-proof.lean` Verification harness: `verify_A89.py`

Kernel Certification

Item	Value
Proof file	`acf-89-proof.lean`
Lean version	`4.12.0` (core only, no Mathlib)
Namespace	`ChryseneFormalism.Article89`
Sections	S1–S15
Definitions	42
Theorems	85
Proved by `decide`	85
Depending on no axioms	**85**
Depending on `propext`	0
Depending on `Quot.sound` or `funext`	0
`sorry`	0
`native_decide`	0
Elaboration time	about 1 min

What This Certifies

The Lean 4.12.0 kernel independently verifies 85 claims from Article 89. Each is stated over `Nat`, `Bool` and `List Nat` and discharged by `decide`. The goal is a decidable proposition; the kernel evaluates a verified Boolean decision procedure, and accepts the proof only when that evaluation yields `true`. There is no external oracle, no `native_decide`, no `sorry` and no Mathlib import.

The file assumes nothing at all. An `#print axioms` audit over every theorem reports *"does not depend on any axioms"* in all 85 cases: not `propext`, not `Quot.sound`, not `funext`, not `Classical.choice`. The customary three-axiom footprint quoted in Lean dossiers would overstate what is assumed here, so it is not quoted.

And what a clean audit does NOT mean here

Article 89 is an editorial survey, and most of what it asserts is not the kind of thing a kernel can reach.
The kernel reaches the article's formal content at the instances it uses: Theorem 1 and each step of its Burnside proof for $N = 1, \ldots, 5$; Corollary 1 for $N \leq 60$; the premises of Proposition 2; Corollary 2; and Proposition 3. Those are results, and there a certified theorem

is a result. Theorem 1 for general N rests on the article's proof, which the file checks instance by instance and does not replace.
It reaches the rest of the article only as arithmetic. The category, basis, technique, cluster, tier and stratum of each paper are a single coder's judgments, entered in this file as data. When S12 certifies that Table 2's No. 2 column is $12, 6, 0, 0, 0, 0, 1, 2$, it certifies that the recorded classification yields those counts, not that the classification is right. Proposition 4 is true by construction, and Proposition 8 is model-theoretic. Neither is encoded.
A reader who reads "85 of 85 axiom-free" as though it settled the survey's judgments has read the wrong thing into it.

Encoding

Marker assignments are natural numbers. A stack of $2N$ sites carries one binary marker per site, so an assignment is a natural number $x < 2^{2N}$ whose bit i is the marker at site i. Complementation is $\varphi(x) = 2^{2N} - 1 - x$. Reversal ρ is computed by `revAux`, which shifts the $2N$ bits out of x and into an accumulator. A class is represented by its least element, so $|\Omega_N/\mathrm{Fr}|$ is a count of representatives and no set type is needed. Every quantifier is bounded and written as a structurally recursive Boolean (`allBelow`, `countBelow`), which keeps `decide` cheap and the recursion shallow.

Codes are lists. The standard and vertebrate mitochondrial tables are lists of 64 amino-acid indices in TCAG order, transcribed from NCBI tables 1 and 2. A fibre is the length of a filtered list.

The identification is a parameter. Under the consecutive-pair convention, a codon's three letters occupy sites $(4, 5)$, $(2, 3)$ and $(0, 1)$. A *labeling* sends each letter to a marker pair. S11 ranges over all 24 labelings instead of choosing one, because the article's source paper does not fix it here. Whether complementation is Watson–Crick under a labeling is tested by $3 - v$, the complement of a two-bit value.

The corpus is data. The 59 rows of the Volume 1 ledger are entered as a `List Row`, each row carrying its pages, category, basis and technique indices. Every printed decimal is certified by `roundsTo`, the integer bound $2|ns - rd| < d$ written without subtraction.

One encoding choice the audit forced. A first version tested Watson–Crick complementation with `Nat.xor`. The audit found that the four theorems through it depended on `propext`, inherited from the library definition of the bitwise operation. Writing the complement of a two-bit value as $3 - v$ states the same thing on the values that occur, and removed the dependence. All four theorems are now axiom-free.

What is not encoded.

- **The step from premises to Proposition 2.** S7 certifies that every class has even cardinality and that φ has no fixed point. That every union of classes then has even cardinality is certified by exhaustion over all 2^{20} unions in the harness (T7-01), not here.
- **Theorem 1 beyond $N = 5$.** The harness reaches $N = 8$.

- **The transcriptions** from the Number 2 register.
- **Propositions 4 and 8.**

Theorem Index

Section	Name	Statement (informal)
S1	`devices_reject_wrong_rounding`	the rounding bound accepts 32.7 and rejects 32.6, 32.8
S2	`phi_involution_3`	$\varphi^2 = 1$ on Ω_3
	`rho_involution_3`	$\rho^2 = 1$ on Ω_3
	`phi_rho_commute_3`	$\varphi\rho = \rho\varphi$ on Ω_3 (Def. 1)
	`phi_ne_rho_3`	$\varphi \neq \rho$, witnessed at $x = 0$
	`phi_rho_not_identity_3`	$\varphi\rho \neq 1$
	`frame_group_is_klein_1_to_5`	all three facts for $N = 1, \ldots, 5$
S3	`twenty_classes`	**Thm. 1**: $\lvert\Omega_3/\mathrm{Fr}\rvert = 20$
	`seventy_two_classes`	$\lvert\Omega_4/\mathrm{Fr}\rvert = 72$
	`class_count_formula_1_to_5`	$2^{2N-2} + 2^{N-1}$ for $N = 1, \ldots, 5$
	`orbits_have_size_2_or_4_3`	Thm. 1(2) at $N = 3$
	`no_orbit_of_size_1_1_to_5`	Thm. 1(2) for $N = 1, \ldots, 5$
	`eight_classes_of_size_2`	eight classes of size 2
	`twelve_classes_of_size_4`	twelve classes of size 4
	`size_split_formula_1_to_5`	Thm. 1(3) for $N = 1, \ldots, 5$
	`pairs_number_64`	$8 \cdot 2 + 12 \cdot 4 = 64$ (Sec. 2)
S4	`fix_phi_empty_1_to_5`	$\mathrm{Fix}(\varphi) = \varnothing$
	`fix_rho_palindromes_1_to_5`	$\lvert\mathrm{Fix}(\rho)\rvert = 2^N$
	`fix_phi_rho_antipalindromes_1_to_5`	$\lvert\mathrm{Fix}(\varphi\rho)\rvert = 2^N$
	`burnside_average_1_to_5`	Burnside's average equals the formula
	`fixed_sets_disjoint_3`	$\mathrm{Fix}(\rho) \cap \mathrm{Fix}(\varphi\rho) = \varnothing$
S5	`capacity_ratio_1_to_60`	**Cor. 1**: ratio $= \frac{1}{4} + 2^{-(N+1)}$, $N \leq 60$
	`capacity_below_one_1_to_60`	ratio < 1, $N \leq 60$
	`capacity_strictly_decreasing_1_to_59`	ratio strictly decreasing
	`multiset_coincidence_only_at_three`	classes = multisets only at $N = 3$
	`at_four_72_against_35`	72 against 35 at $N = 4$
S6	`phi_orbits_32`	$\langle\varphi\rangle$ has 32 orbits
	`rho_orbits_36`	$\langle\rho\rangle$ has 36 orbits
	`phi_rho_orbits_36`	$\langle\varphi\rho\rangle$ has 36 orbits
S7	`every_orbit_even_1_to_5`	**Prop. 2** premise: every class even
	`phi_fixed_point_free_3`	and φ has no fixed point
S8	`tables_have_64_codons`	both tables have 64 entries
	`std_met_trp_singletons`	**Cor. 2(1)**: Met, Trp singletons
	`std_odd_fibres_exactly`	odd fibres: Met, Trp, Ile (3), stop (3)
	`mito_all_fibres_even`	**Cor. 2(2)**: all fibres even
	`mito_fibre_profile`	the full fibre profile
	`mito_met_trp_pairs`	AUA $\mapsto$ Met, UGA $\mapsto$ Trp
S9	`codes_differ_at_four_codons`	UGA, AUA, AGA, AGG
	`mito_normal_form_everywhere`	mitochondrial: all 16 blocks normal
	`std_leaves_normal_form_at_UG_and_AU`	standard: exactly UG and AU are not
	`AG_purine_half_uniform_in_both`	AG purine half uniform in both
S10	`transversality_bit_level`	**Prop. 3**, consecutive-pair convention
S11	`twenty_four_labelings`	24 letter labelings
	`eight_watson_crick_labelings`	eight make complementation Watson–Crick
	`nineteen_of_twenty_under_watson_crick`	under those, both codes split 19 of 20
	`twenty_of_twenty_otherwise`	under the other 16, all 20
	`unsplit_class_is_AGU_UCA`	the unsplit class is $\{\mathrm{AGU}, \mathrm{UCA}\}$
	`unsplit_class_painted_Ser`	painted Ser in both

Section	Name	Statement (informal)
	`std_splits_under_every_labeling`	Cor. 2(1) directly, all 24
S12	`fifty_nine_articles`	$59 = 38 + 21$
	`pages_contiguous`	pp. 1–1162 without gap or overlap
	`table2_number1`	**Table 2**, No. 1 column
	`table2_number2`	**Table 2**, No. 2 column
	`concentration_over_half`	**Rem. 5**: $12 > 21/2$
	`category_shares`	$34.2 \to 57.1\%$, $15.8 \to 28.6\%$
	`basis_number2`	$11, 8, 2$; 52.4, 38.1, 9.5%
	`basis_number1`	$18, 20, 0$
	`table3_number2`	**Table 3**, No. 2 column
	`table3_number1_undaggered`	Table 3, undaggered No. 1 entries
	`sixteen_of_twenty_eight`	16 of 28 techniques occur
	`thermo_and_topology_absent`	$11 \to 0$ and $8 \to 0$
	`twos_and_ones`	the techniques at 2 and at 1
S13	`new_techniques_unassessed_on_number1`	**Prop. 5**: $T'\|_I = T$
	`article39_table1_reproduced`	Art. 39 Table 1, unchanged
	`article39_table2_reproduced`	Art. 39 Table 2, unchanged
	`article39_shares`	Art. 39 shares, unchanged
S14	`page_totals`	687 and 475 pages
	`means`	32.7; 12.5 exactly
	`ratio_of_means`	ratio 2.6
	`median_number2_is_29`	median 29, by counting
	`median_number1_is_11`	median 11, by counting
	`ranges`	9–80 and 5–23
	`three_longest`	Art. 40 (72), 49 (64), 54 (80)
	`tier_pages`	214 and 473
S15	`clusters_partition_J`	**Def. 9**: a partition, sizes $2, 2, 2, 8, 4, 1, 2$
	`noncollapse_witnesses`	**Prop. 6**: 47 Math, 57 Mol
	`noncollapse_without_48`	robust to Art. 48
	`cb_split_7_5_and_12_1`	7/5 against 12/1
	`allocation_reconstructs_tiers`	**Def. 6**: (a)∪(b)∪(c) = tier
	`clause_c_excludes_50_and_55`	Art. 50, 55 not selected
	`lineage_acyclic`	**Fig. 1**: every edge forward, so acyclic
	`lineage_double_bordered`	double borders = multi-pass
	`strata_sizes`	**Def. 10**: strata $5, 3, 5$
	`precedence_needed`	two rows need the precedence
	`model_families`	$15, 2, 3, 1$; 71.4%

Proof Listing: `acf-89-proof.lean`

`acf-89-proof.lean` — complete source

```
/-
  acf-89-proof.lean

  Annals of the Chrysene Formalism, Vol. 1, No. 4, Article 89, first page 1808
  "The Chrysene Formalism as Candidate Mathematics and Candidate Biogenesis:
   A Quantitative and Categorical Survey of Volume 1, Number 2"

  Lean 4.12.0, core only (no Mathlib).  Every theorem is a decidable
  proposition over Nat, Bool and List Nat, discharged by `decide`.
  No `sorry`, no `native_decide`, no axioms declared.

  Companion harness: verify_A89.py.  The harness derives; this file certifies.

  Encodings
    * A marker assignment of a 2n-site stack is a natural number x < 2^(2n);
      bit i is the marker at site i.
    * phi n x  : complementation of all 2n markers  (2^(2n) - 1 - x).
    * rho n x  : reversal of the 2n markers.
    * A class (Fr-orbit) is represented by its least element.
    * Codon index in TCAG order: 16*first + 4*second + third, T=0 C=1 A=2 G=3.
    * Amino acids: A0 C1 D2 E3 F4 G5 H6 I7 K8 L9 M10 N11 P12 Q13 R14 S15
      T16 V17 W18 Y19, stop 20.
    * Corpus rows: art, first page, last page, category, basis, techniques.
      Category PM0 CB1 AT2 QA3 CR4 MD5 FN6 ED7.  Basis Mol0 Math1 Meth2.
-/

set_option maxRecDepth 100000
set_option exponentiation.threshold 600

namespace ChryseneFormalism.Article89

/-! ## S1. Devices -/

/-- `n/d` rounds to `r/s`: `2|ns - rd| < d`, written without subtraction. -/
def roundsTo (n d r s : Nat) : Bool :=
  decide (2 * (n * s) < 2 * (r * d) + d) && decide (2 * (r * d) < 2 * (n * s) +
      d)

/-- `∀ x < k, p x`, as a Boolean, by structural recursion. -/
def allBelow (p : Nat → Bool) : Nat → Bool
  | 0 => true
  | k + 1 => allBelow p k && p k

/-- `#{x < k | p x}`. -/
def countBelow (p : Nat → Bool) : Nat → Nat
  | 0 => 0
  | k + 1 => countBelow p k + (if p k then 1 else 0)

theorem devices_reject_wrong_rounding :
    roundsTo 687 21 327 10 = true ∧ roundsTo 687 21 328 10 = false ∧
    roundsTo 687 21 326 10 = false := by decide

/-! ## S2. The frame group (Definition 1) -/
```

```
def phi (n x : Nat) : Nat := 2 ^ (2 * n) - 1 - x

def revAux : Nat → Nat → Nat → Nat
  | 0, _, acc => acc
  | k + 1, x, acc => revAux k (x / 2) (acc * 2 + x % 2)

def rho (n x : Nat) : Nat := revAux (2 * n) x 0

def sizeOmega (n : Nat) : Nat := 2 ^ (2 * n)

theorem phi_involution_3 : allBelow (fun x => phi 3 (phi 3 x) == x) 64 = true :=
    by decide
theorem rho_involution_3 : allBelow (fun x => rho 3 (rho 3 x) == x) 64 = true :=
    by decide
theorem phi_rho_commute_3 :
    allBelow (fun x => phi 3 (rho 3 x) == rho 3 (phi 3 x)) 64 = true := by
      decide
theorem phi_ne_rho_3 : phi 3 0 = 63 ∧ rho 3 0 = 0 := by decide
theorem phi_rho_not_identity_3 : phi 3 (rho 3 0) = 63 := by decide

theorem frame_group_is_klein_1_to_5 :
    (List.range 5).all (fun m =>
      allBelow (fun x =>
        phi (m+1) (phi (m+1) x) == x && rho (m+1) (rho (m+1) x) == x &&
        phi (m+1) (rho (m+1) x) == rho (m+1) (phi (m+1) x)) (sizeOmega (m+1))) =
          true := by
  decide

/-! ## S3. Theorem 1: the registry-class count -/

/-- Number of distinct elements among `x, phi x, rho x, phi rho x`. -/
def orbSize (n x : Nat) : Nat :=
  let b := phi n x
  let c := rho n x
  let d := phi n (rho n x)
  1 + (if b != x then 1 else 0) + (if c != x && c != b then 1 else 0)
    + (if d != x && d != b && d != c then 1 else 0)

/-- `x` is the least element of its class. -/
def isRep (n x : Nat) : Bool :=
  decide (x ≤ phi n x) && decide (x ≤ rho n x) && decide (x ≤ phi n (rho n x))

def classCount (n : Nat) : Nat := countBelow (isRep n) (sizeOmega n)

def classFormula (n : Nat) : Nat := 2 ^ (2 * n - 2) + 2 ^ (n - 1)

theorem twenty_classes : classCount 3 = 20 := by decide
theorem seventy_two_classes : classCount 4 = 72 := by decide
theorem class_count_formula_1_to_5 :
    (List.range 5).all (fun m => classCount (m+1) == classFormula (m+1)) = true
      := by decide

theorem orbits_have_size_2_or_4_3 :
    allBelow (fun x => orbSize 3 x == 2 || orbSize 3 x == 4) 64 = true := by
      decide
theorem no_orbit_of_size_1_1_to_5 :
    (List.range 5).all (fun m =>
      allBelow (fun x => orbSize (m+1) x == 2 || orbSize (m+1) x == 4)
        (sizeOmega (m+1))) = true := by decide
```

```
theorem eight_classes_of_size_2 :
    countBelow (fun x => isRep 3 x && orbSize 3 x == 2) 64 = 8 := by decide
theorem twelve_classes_of_size_4 :
    countBelow (fun x => isRep 3 x && orbSize 3 x == 4) 64 = 12 := by decide
theorem size_split_formula_1_to_5 :
    (List.range 5).all (fun m =>
      countBelow (fun x => isRep (m+1) x && orbSize (m+1) x == 2) (sizeOmega (m
          +1))
        == 2 ^ (m+1) &&
      countBelow (fun x => isRep (m+1) x && orbSize (m+1) x == 4) (sizeOmega (m
          +1))
        == 2 ^ (2*(m+1) - 2) - 2 ^ m) = true := by decide

theorem pairs_number_64 :
    countBelow (fun x => isRep 3 x && orbSize 3 x == 2) 64 * 2 +
    countBelow (fun x => isRep 3 x && orbSize 3 x == 4) 64 * 4 = 64 := by decide

/-! ## S4. Burnside fixed points -/

theorem fix_phi_empty_1_to_5 :
    (List.range 5).all (fun m =>
      countBelow (fun x => phi (m+1) x == x) (sizeOmega (m+1)) == 0) = true :=
          by decide
theorem fix_rho_palindromes_1_to_5 :
    (List.range 5).all (fun m =>
      countBelow (fun x => rho (m+1) x == x) (sizeOmega (m+1)) == 2 ^ (m+1)) =
          true := by
  decide
theorem fix_phi_rho_antipalindromes_1_to_5 :
    (List.range 5).all (fun m =>
      countBelow (fun x => phi (m+1) (rho (m+1) x) == x) (sizeOmega (m+1))
        == 2 ^ (m+1)) = true := by decide
theorem burnside_average_1_to_5 :
    (List.range 5).all (fun m =>
      (2 ^ (2*(m+1)) + 0 + 2 ^ (m+1) + 2 ^ (m+1)) == 4 * classFormula (m+1)) =
          true := by
  decide
theorem fixed_sets_disjoint_3 :
    allBelow (fun x => !(rho 3 x == x && phi 3 (rho 3 x) == x)) 64 = true := by
        decide

/-! ## S5. Corollary 1: the capacity ratio -/

theorem capacity_ratio_1_to_60 :
    (List.range 60).all (fun m =>
      classFormula (m+1) * (4 * 2 ^ (m+2)) == 4 ^ (m+1) * (2 ^ (m+2) + 4)) =
          true := by
  decide
theorem capacity_below_one_1_to_60 :
    (List.range 60).all (fun m => decide (classFormula (m+1) < 4 ^ (m+1))) =
        true := by
  decide
theorem capacity_strictly_decreasing_1_to_59 :
    (List.range 59).all (fun m =>
      decide (classFormula (m+2) * 4 ^ (m+1) < classFormula (m+1) * 4 ^ (m+2)))
          = true := by
  decide
theorem multiset_coincidence_only_at_three :
```

```
    (List.range 60).all (fun m =>
      (classFormula (m+1) * 6 == (m+4) * (m+3) * (m+2)) == (m+1 == 3)) = true :=
        by decide
theorem at_four_72_against_35 : classFormula 4 = 72 ∧ 7 * 6 * 5 = 35 * 6 := by
    decide

/-! ## S6. The subgroups of Fr at N = 3 -/

theorem phi_orbits_32 : countBelow (fun x => decide (x ≤ phi 3 x)) 64 = 32 := by
    decide
theorem rho_orbits_36 : countBelow (fun x => decide (x ≤ rho 3 x)) 64 = 36 := by
    decide
theorem phi_rho_orbits_36 :
    countBelow (fun x => decide (x ≤ phi 3 (rho 3 x))) 64 = 36 := by decide

/-! ## S7. Proposition 2: the premises of the parity argument -/

theorem every_orbit_even_1_to_5 :
    (List.range 5).all (fun m =>
      allBelow (fun x => orbSize (m+1) x % 2 == 0) (sizeOmega (m+1))) = true :=
        by decide
theorem phi_fixed_point_free_3 : allBelow (fun x => phi 3 x != x) 64 = true :=
    by decide

/-! ## S8. Corollary 2: the two codes -/

def std : List Nat := [4, 4, 9, 9, 15, 15, 15, 15, 19, 19, 20, 20, 1, 1, 20, 18,
    9, 9, 9, 9, 12, 12, 12, 12, 6, 6, 13, 13, 14, 14, 14, 14, 7, 7, 7, 10, 16,
    16, 16, 16, 11, 11, 8, 8, 15, 15, 14, 14, 17, 17, 17, 17, 0, 0, 0, 0, 2, 2,
    3, 3, 5, 5, 5, 5]
def mito : List Nat := [4, 4, 9, 9, 15, 15, 15, 15, 19, 19, 20, 20, 1, 1, 18,
    18, 9, 9, 9, 9, 12, 12, 12, 12, 6, 6, 13, 13, 14, 14, 14, 14, 7, 7, 10, 10,
    16, 16, 16, 16, 11, 11, 8, 8, 15, 15, 20, 20, 17, 17, 17, 17, 0, 0, 0, 0, 2,
    2, 3, 3, 5, 5, 5, 5]

def fibre (t : List Nat) (a : Nat) : Nat := (t.filter (· == a)).length

theorem tables_have_64_codons : std.length = 64 ∧ mito.length = 64 := by decide
theorem std_met_trp_singletons : fibre std 10 = 1 ∧ fibre std 18 = 1 := by
    decide
theorem std_odd_fibres_exactly :
    (List.range 21).filter (fun a => fibre std a % 2 == 1) = [7, 10, 18, 20] ∧
    fibre std 7 = 3 ∧ fibre std 20 = 3 := by decide
theorem mito_all_fibres_even :
    (List.range 21).all (fun a => fibre mito a % 2 == 0) = true := by decide
theorem mito_fibre_profile :
    (List.range 21).map (fibre mito) =
      [4, 2, 2, 2, 2, 4, 2, 2, 2, 6, 2, 2, 4, 2, 4, 6, 4, 4, 2, 2, 4] := by
        decide
theorem mito_met_trp_pairs :
    mito.getD 34 99 = 10 ∧ mito.getD 14 99 = 18 ∧ fibre mito 10 = 2 ∧ fibre mito
        18 = 2 := by
  decide

/-! ## S9. Normal form; where the codes differ -/

def halfUniform (t : List Nat) (blk : Nat) (purine : Bool) : Bool :=
  if purine then t.getD (4*blk+2) 99 == t.getD (4*blk+3) 99
  else t.getD (4*blk) 99 == t.getD (4*blk+1) 99
```

```
def inNormalForm (t : List Nat) (blk : Nat) : Bool :=
  halfUniform t blk false && halfUniform t blk true

theorem codes_differ_at_four_codons :
    (List.range 64).filter (fun i => std.getD i 99 != mito.getD i 99) = [14, 34,
        46, 47] := by
  decide
theorem mito_normal_form_everywhere :
    (List.range 16).all (inNormalForm mito) = true := by decide
theorem std_leaves_normal_form_at_UG_and_AU :
    (List.range 16).filter (fun b => !inNormalForm std b) = [3, 8] := by decide
theorem AG_purine_half_uniform_in_both :
    halfUniform std 11 true = true ∧ halfUniform mito 11 true = true := by
       decide

/-! ## S10. Proposition 3: transversality (consecutive-pair convention) -/

/-- A class meets a codon block (the first four markers) in at most one word:
    each non-identity element either fixes `x` or moves it out of its block. -/
theorem transversality_bit_level :
    allBelow (fun x =>
      phi 3 x / 4 != x / 4 &&
      (rho 3 x == x || rho 3 x / 4 != x / 4) &&
      (phi 3 (rho 3 x) == x || phi 3 (rho 3 x) / 4 != x / 4)) 64 = true := by
         decide

/-! ## S11. Nineteen of twenty split classes -/

def idx : List Nat → Nat → Nat
  | [], _ => 0
  | a :: l, v => if a == v then 0 else idx l v + 1

/-- Codon index of assignment `x`; `lab` sends a letter (T,C,A,G) to its marker
    pair. -/
def codonOf (lab : List Nat) (x : Nat) : Nat :=
  idx lab (x / 16) * 16 + idx lab ((x / 4) % 4) * 4 + idx lab (x % 4)

def val (t lab : List Nat) (x : Nat) : Nat := t.getD (codonOf lab x) 99

def splitAt (t lab : List Nat) (x : Nat) : Bool :=
  !(val t lab x == val t lab (phi 3 x) && val t lab x == val t lab (rho 3 x) &&
    val t lab x == val t lab (phi 3 (rho 3 x)))

def splitCount (t lab : List Nat) : Nat :=
  countBelow (fun x => isRep 3 x && splitAt t lab x) 64

/-- Complementing the markers realises T<->A and C<->G.  On a marker pair
    `v < 4`, complementation is `3 - v`. -/
def watsonCrick (lab : List Nat) : Bool :=
  3 - lab.getD 0 9 == lab.getD 2 9 && 3 - lab.getD 1 9 == lab.getD 3 9

def labelings : List (List Nat) := [[0, 1, 2, 3], [0, 1, 3, 2], [0, 2, 1, 3],
    [0, 2, 3, 1], [0, 3, 1, 2], [0, 3, 2, 1], [1, 0, 2, 3], [1, 0, 3, 2], [1, 2,
     0, 3], [1, 2, 3, 0], [1, 3, 0, 2], [1, 3, 2, 0], [2, 0, 1, 3], [2, 0, 3,
    1], [2, 1, 0, 3], [2, 1, 3, 0], [2, 3, 0, 1], [2, 3, 1, 0], [3, 0, 1, 2],
    [3, 0, 2, 1], [3, 1, 0, 2], [3, 1, 2, 0], [3, 2, 0, 1], [3, 2, 1, 0]]

def unsplitCodonsAre (t lab : List Nat) (c1 c2 : Nat) : Bool :=
```

```
  allBelow (fun x => splitAt t lab x || codonOf lab x == c1 || codonOf lab x ==
      c2) 64 &&
  countBelow (fun x => !splitAt t lab x) 64 == 2

theorem twenty_four_labelings : labelings.length = 24 := by decide
theorem eight_watson_crick_labelings : (labelings.filter watsonCrick).length = 8
    := by decide
theorem nineteen_of_twenty_under_watson_crick :
    (labelings.filter watsonCrick).all (fun lab =>
      splitCount std lab == 19 && splitCount mito lab == 19) = true := by decide
theorem twenty_of_twenty_otherwise :
    (labelings.filter (fun l => !watsonCrick l)).all (fun lab =>
      splitCount std lab == 20 && splitCount mito lab == 20) = true := by decide
theorem unsplit_class_is_AGU_UCA :
    (labelings.filter watsonCrick).all (fun lab =>
      unsplitCodonsAre std lab 44 6 && unsplitCodonsAre mito lab 44 6) = true :=
          by decide
theorem unsplit_class_painted_Ser :
    std.getD 44 99 = 15 ∧ std.getD 6 99 = 15 ∧ mito.getD 44 99 = 15 ∧ mito.getD
        6 99 = 15 := by
  decide
theorem std_splits_under_every_labeling :
    labelings.all (fun lab => decide (0 < splitCount std lab)) = true := by
        decide

/-! ## S12. The corpus: Tables 2 and 3, Remark 5, Section 5 -/

structure Row where
  art : Nat
  p0 : Nat
  p1 : Nat
  cat : Nat
  basis : Nat
  tech : List Nat

def rows : List Row := [
  ⟨1, 1, 5, 0, 1, [0, 1, 2]⟩,
  ⟨2, 6, 14, 0, 1, [0, 3, 4, 5]⟩,
  ⟨3, 15, 26, 1, 1, [6, 2, 7]⟩,
  ⟨4, 27, 40, 1, 0, [8, 9]⟩,
  ⟨5, 41, 60, 1, 0, [8, 10, 9]⟩,
  ⟨6, 61, 77, 5, 1, [5, 11, 0]⟩,
  ⟨7, 78, 87, 5, 1, [5, 12, 2]⟩,
  ⟨8, 88, 97, 5, 1, [5, 13, 14]⟩,
  ⟨9, 98, 112, 3, 0, [15, 9]⟩,
  ⟨10, 113, 123, 4, 1, [8, 11, 0]⟩,
  ⟨11, 124, 134, 4, 1, [4, 3]⟩,
  ⟨12, 135, 148, 0, 1, [8, 0, 16]⟩,
  ⟨13, 149, 156, 0, 1, [16, 8, 1]⟩,
  ⟨14, 157, 166, 5, 1, [0, 3, 17]⟩,
  ⟨15, 167, 176, 3, 0, [15, 2, 1]⟩,
  ⟨16, 177, 186, 2, 0, [13, 15]⟩,
  ⟨17, 187, 205, 1, 0, [8, 6]⟩,
  ⟨18, 206, 219, 1, 0, [11, 9]⟩,
  ⟨19, 220, 232, 1, 0, [9, 13]⟩,
  ⟨20, 233, 247, 1, 0, [8, 9]⟩,
  ⟨21, 248, 258, 1, 0, [7, 2]⟩,
  ⟨22, 259, 272, 1, 0, [3, 9]⟩,
  ⟨23, 273, 282, 1, 0, [18, 1]⟩,
```

```
  ⟨24, 283, 291, 1, 0, [18, 7]⟩,
  ⟨25, 292, 304, 1, 0, [13, 2]⟩,
  ⟨26, 305, 321, 1, 0, [8, 1]⟩,
  ⟨27, 322, 331, 6, 1, [5, 9, 1]⟩,
  ⟨28, 332, 341, 5, 0, [14, 12, 9]⟩,
  ⟨29, 342, 351, 6, 1, [1, 13]⟩,
  ⟨30, 352, 363, 4, 1, [3, 0, 2]⟩,
  ⟨31, 364, 371, 6, 1, [9, 1, 5]⟩,
  ⟨32, 372, 385, 5, 1, [5, 3, 14]⟩,
  ⟨33, 386, 394, 0, 1, [19, 9, 4]⟩,
  ⟨34, 395, 402, 0, 1, [8, 16]⟩,
  ⟨35, 403, 413, 2, 0, [5, 11, 7, 12, 17]⟩,
  ⟨36, 414, 436, 2, 0, [15, 2]⟩,
  ⟨37, 437, 458, 2, 0, [15, 13]⟩,
  ⟨38, 459, 475, 2, 0, [15]⟩,
  ⟨39, 476, 488, 7, 2, [20, 0]⟩,
  ⟨40, 489, 560, 0, 1, [21, 0, 3, 2, 13]⟩,
  ⟨41, 561, 581, 7, 2, [20]⟩,
  ⟨42, 582, 602, 0, 1, [21, 13]⟩,
  ⟨43, 603, 645, 0, 1, [5, 22, 23, 0]⟩,
  ⟨44, 646, 675, 0, 1, [0, 3, 4, 2]⟩,
  ⟨45, 676, 704, 0, 1, [2, 24, 25]⟩,
  ⟨46, 705, 729, 0, 1, [5, 22, 23]⟩,
  ⟨47, 730, 767, 1, 1, [24, 2, 25]⟩,
  ⟨48, 768, 793, 1, 0, [2, 24]⟩,
  ⟨49, 794, 857, 3, 0, [15, 0, 23, 21]⟩,
  ⟨50, 858, 882, 1, 1, [24, 26, 2]⟩,
  ⟨51, 883, 913, 1, 1, [26, 24]⟩,
  ⟨52, 914, 925, 1, 0, [2, 24]⟩,
  ⟨53, 926, 934, 1, 1, [24, 2]⟩,
  ⟨54, 935, 1014, 1, 0, [27, 8, 2]⟩,
  ⟨55, 1015, 1047, 1, 1, [24]⟩,
  ⟨56, 1048, 1075, 1, 0, [27, 25]⟩,
  ⟨57, 1076, 1106, 1, 0, [13, 24, 27]⟩,
  ⟨58, 1107, 1136, 1, 0, [2, 13, 27]⟩,
  ⟨59, 1137, 1162, 1, 0, [2, 27, 24]⟩
]

def issue1 : List Row := rows.filter (fun r => decide (r.art ≤ 38))
def issue2 : List Row := rows.filter (fun r => decide (39 ≤ r.art))
def catCount (rs : List Row) (c : Nat) : Nat := (rs.filter (fun r => r.cat == c)
    ).length
def basisCount (rs : List Row) (b : Nat) : Nat := (rs.filter (fun r => r.basis
    == b)).length
def techCount (rs : List Row) (t : Nat) : Nat := (rs.filter (fun r => r.tech.
    contains t)).length
def len (r : Row) : Nat := r.p1 + 1 - r.p0
def pages (rs : List Row) : Nat := (rs.map len).foldl (· + ·) 0

theorem fifty_nine_articles :
    rows.length = 59 ∧ issue1.length = 38 ∧ issue2.length = 21 := by decide
theorem pages_contiguous :
    (rows.zip rows.tail).all (fun p => p.2.p0 == p.1.p1 + 1) = true := by decide
theorem table2_number1 :
    [1, 0, 5, 2, 4, 6, 3, 7].map (catCount issue1) = [13, 6, 6, 5, 3, 3, 2, 0]
        := by decide
theorem table2_number2 :
    [1, 0, 5, 2, 4, 6, 3, 7].map (catCount issue2) = [12, 6, 0, 0, 0, 0, 1, 2]
        := by decide
```

```
theorem concentration_over_half : 2 * catCount issue2 1 > issue2.length := by
    decide
theorem category_shares :
    roundsTo 1300 38 342 10 = true ∧ roundsTo 1200 21 571 10 = true ∧
    roundsTo 600 38 158 10 = true ∧ roundsTo 600 21 286 10 = true := by decide
theorem basis_number2 :
    [1, 0, 2].map (basisCount issue2) = [11, 8, 2] ∧
    roundsTo 1100 21 524 10 = true ∧ roundsTo 800 21 381 10 = true ∧
    roundsTo 200 21 95 10 = true := by decide
theorem basis_number1 : [1, 0, 2].map (basisCount issue1) = [18, 20, 0] := by
    decide
theorem table3_number2 :
    [2, 24, 0, 27, 13, 21, 23, 25].map (techCount issue2) = [11, 10, 5, 5, 4, 3,
        3, 3] := by
  decide
theorem table3_number1_undaggered : [2, 0, 13].map (techCount issue1) = [8, 7,
    6] := by decide
theorem sixteen_of_twenty_eight :
    ((List.range 28).filter (fun t => decide (0 < techCount issue2 t))).length =
        16 := by
  decide
theorem thermo_and_topology_absent :
    techCount issue1 9 = 11 ∧ techCount issue2 9 = 0 ∧
    techCount issue1 1 = 8 ∧ techCount issue2 1 = 0 := by decide
theorem twos_and_ones :
    (List.range 28).filter (fun t => techCount issue2 t == 2) = [3, 5, 20, 22,
        26] ∧
    (List.range 28).filter (fun t => techCount issue2 t == 1) = [4, 8, 15] := by
        decide

/-! ## S13. Proposition 5: conservativity on Number 1 -/

theorem new_techniques_unassessed_on_number1 :
    issue1.all (fun r => r.tech.all (fun t => decide (t < 20))) = true := by
        decide
theorem article39_table1_reproduced :
    [1, 0, 5, 2, 4, 6, 3].map (catCount issue1) = [13, 6, 6, 5, 3, 3, 2] := by
        decide
theorem article39_table2_reproduced :
    [9, 8, 1, 5, 2, 0, 3, 13, 15].map (techCount issue1) = [11, 9, 8, 8, 8, 7,
        6, 6, 6] := by
  decide
theorem article39_shares :
    roundsTo 1100 38 289 10 = true ∧ roundsTo 900 38 237 10 = true ∧
    roundsTo 800 38 211 10 = true ∧ roundsTo 700 38 184 10 = true ∧
    roundsTo 2000 38 526 10 = true ∧ roundsTo 1800 38 474 10 = true := by decide

/-! ## S14. Section 5: page figures -/

def lens2 : List Nat := issue2.map len
def lens1 : List Nat := issue1.map len
def multiPass : List Nat := [40, 44, 45, 48, 57, 59]

theorem page_totals : pages issue2 = 687 ∧ pages issue1 = 475 := by decide
theorem means : roundsTo 687 21 327 10 = true ∧ 475 * 2 = 38 * 25 := by decide
theorem ratio_of_means : roundsTo (687 * 38) (21 * 475) 26 10 = true := by
    decide
theorem median_number2_is_29 :
    lens2.contains 29 = true ∧ (lens2.filter (· < 29)).length ≤ 10 ∧
```

```
    (lens2.filter (· > 29)).length ≤ 10 := by decide
theorem median_number1_is_11 :
    (lens1.filter (· < 11)).length ≤ 18 ∧ (lens1.filter (· > 11)).length ≤ 18 :=
      by decide
theorem ranges :
    lens2.all (fun l => decide (9 ≤ l ∧ l ≤ 80)) = true ∧ lens2.contains 9 =
      true ∧
    lens2.contains 80 = true ∧ lens1.all (fun l => decide (5 ≤ l ∧ l ≤ 23)) =
      true ∧
    lens1.contains 5 = true ∧ lens1.contains 23 = true := by decide
theorem three_longest :
    (issue2.filter (fun r => decide (64 ≤ len r))).map (fun r => (r.art, len r))
      =
      [(40, 72), (49, 64), (54, 80)] := by decide
theorem tier_pages :
    pages (issue2.filter (fun r => multiPass.contains r.art)) = 214 ∧ 687 - 214
      = 473 := by
  decide

/-! ## S15. Clusters, Proposition 6, Definition 6, Figure 1, Definition 10 -/

def clusters : List (List Nat) :=
  [[40, 42], [43, 46], [44, 45], [47, 48, 50, 51, 52, 53, 55, 57], [54, 56, 58,
    59],
   [49], [39, 41]]

def basisOf (a : Nat) : Nat := ((rows.filter (fun r => r.art == a)).map (·.basis
  )).getD 0 9

theorem clusters_partition_J :
    (List.range 21).all (fun i =>
      (clusters.filter (fun c => c.contains (39 + i))).length == 1) = true ∧
    (clusters.map List.length) = [2, 2, 2, 8, 4, 1, 2] := by decide
theorem noncollapse_witnesses : basisOf 47 = 1 ∧ basisOf 57 = 0 := by decide
theorem noncollapse_without_48 :
    ([47, 50, 51, 52, 53, 55, 57].any (fun a => basisOf a == 0) &&
     [47, 50, 51, 52, 53, 55, 57].any (fun a => basisOf a == 1)) = true := by
      decide
theorem cb_split_7_5_and_12_1 :
    (issue2.filter (fun r => r.cat == 1 && r.basis == 0)).length = 7 ∧
    (issue2.filter (fun r => r.cat == 1 && r.basis == 1)).length = 5 ∧
    (issue1.filter (fun r => r.cat == 1 && r.basis == 0)).length = 12 ∧
    (issue1.filter (fun r => r.cat == 1 && r.basis == 1)).length = 1 := by
      decide
theorem allocation_reconstructs_tiers :
    (List.range 60).all (fun a =>
      ([40, 44, 45].contains a || [45, 48, 59].contains a || [57].contains a)
        == multiPass.contains a) = true := by decide
theorem clause_c_excludes_50_and_55 :
    ([50, 55].all (fun a => !multiPass.contains a)) = true := by decide

def edges : List (Nat × Nat) :=
  [(44, 45), (45, 47), (45, 48), (48, 50), (48, 57), (50, 51), (51, 52), (52,
    53),
   (50, 55), (55, 56), (54, 56), (57, 58), (57, 59), (58, 59)]

/-- Every edge points forward in publication order, so Figure 1 has no cycle. -/
theorem lineage_acyclic : edges.all (fun e => decide (e.1 < e.2)) = true := by
    decide
```

```
theorem lineage_double_bordered :
    [44, 45, 47, 48, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59].filter multiPass.
       contains
      = [44, 45, 48, 57, 59] := by decide

/-- Table 4 rows: (chemical assertion, table as hypothesis or input). -/
def strataRows : List (Bool × Bool) :=
  [(false, false), (false, false), (false, false), (false, false), (false, false
      ),
   (false, true), (false, true), (false, true),
   (true, false), (true, true), (true, true), (true, false), (true, false)]

def sigma (r : Bool × Bool) : Nat := if r.1 then 3 else if r.2 then 2 else 1

theorem strata_sizes :
    [1, 2, 3].map (fun k => (strataRows.filter (fun r => sigma r == k)).length)
       = [5, 3, 5] := by
  decide
theorem precedence_needed :
    (strataRows.filter (fun r => r.1 && r.2)).length = 2 := by decide

/-- Review families for Number 2: Claude 0, ChatGPT 1, Grok 2, mixed 3. -/
def family2 : List Nat :=
  [0, 3, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 2, 2, 2, 0, 0, 0, 0, 0, 0]

theorem model_families :
    [0, 1, 2, 3].map (fun f => (family2.filter (· == f)).length) = [15, 2, 3, 1]
        ∧
    roundsTo 1500 21 714 10 = true := by decide

end ChryseneFormalism.Article89
```

Kernel Axiom Audit

Axiom Inventory

Running `#print axioms` on each of the 85 theorems in `acf-89-proof.lean` yields:

Result	Meaning	Count
no axioms	The proof term reduces to primitive inference steps alone. Nothing is assumed beyond the type theory.	**85**
`propext`	Not used.	0
`Quot.sound`	Not used.	0
`funext`	Not used.	0
`Classical.choice`	Not used.	0
Extra axioms introduced by this proof:		**none**

Verification Note

The proof file makes no use of `native_decide`, which bypasses the kernel by running compiled code. Nor does it use `sorry`, `Classical.choice`, `Classical.em`, or any Mathlib import. The `decide` tactic in Lean 4 core reduces a goal to a kernel-evaluated Boolean expression, and the kernel itself performs the evaluation.

Two `set_option` lines appear at the head of the file, and neither weakens anything.

- `maxRecDepth` is raised because the exhaustive searches over Ω_5 (1024 assignments) recurse past the default.
- `exponentiation.threshold` raises the point at which the *elaborator* declines to pre-evaluate a power. Past it Lean emits a warning, not an error, and the kernel evaluates the power either way. Corollary 1 is certified up to 4^{60}.

Neither option admits an axiom, and the audit above is the evidence.

To reproduce, install Lean 4.12.0 and run `lean acf-89-proof.lean`. A silent run with exit code 0 is the certificate; it takes about a minute. There are no dependencies beyond the Lean core, so no `lakefile` and no package fetch are required.

Scope: three limits, stated plainly

A kernel-checked theorem certifies that a conclusion follows from the definitions in the file. Three limits belong with that.

First, the classifications are data. S12–S15 certify arithmetic on the recorded category, basis, technique, cluster, tier and stratum of each paper. They do not certify those records.

Second, S10–S11 hold under a convention. The identification of codons with marker assignments is the consecutive-pair convention, and S11 ranges over all 24 labelings rather than assume one. That Art. 50 uses this convention is not shown here. What is shown is that the article's "nineteen of twenty" holds exactly for the Watson–Crick labelings and fails, in favour of twenty, for all others.

Third, the file is a regression test on the article. The allocation theorem fails if Definition 6 stops reproducing the recorded tiers, and S12 fails if a classification changes. That is deliberate: a published verification artifact should stop passing when the claims it supports stop holding. It also means the file cannot test a revised classification without being edited.