ACF 89 — Verification 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

Verification script: `verify_A89.py` Lean proof: `acf-89-proof.lean`

Certification Summary

Item	Value
Script	`verify_A89.py`
Total checks	137
Passed	**137**
Failed	**0**
Exit code	0
Blocks	16 (T1–T16)
Check kinds	95 exact, 27 transcription, 9 convention, 6 recorded
Floating-point operations	**0**
Dependencies	Python standard library only
Run time	about 3 s
Lean proof file	`acf-89-proof.lean`
Lean version	`4.12.0` (core only, no Mathlib)
Lean theorems	85
Theorems with no axioms	**85**
`sorry`	0
`native_decide`	0

What a harness can and cannot certify for a survey

Article 89 is an editorial survey, and its claims are of three kinds that a harness meets very differently.

Formal claims. Theorem 1 (the registry-class count), Corollary 1 (the capacity ratio), Proposition 2 (even fibres), Corollary 2 (the two codes), Proposition 3 (transversality), and the counting content of Propositions 5–7 follow from definitions stated in the article. The harness re-derives each by exhaustion or exact arithmetic at every instance the article uses. Theorem 1 is checked for every N from 1 to 8, not only at $N = 3$. Its statement for general N rests on the article's Burnside proof, whose steps block T4 checks instance by instance.

Classifications. The category, basis, technique, cluster, tier and stratum of each paper are editorial judgments recorded in the Volume 1 ledger by a single coder. *No harness can certify that Art. 48's basis is molecular.* What the harness certifies is that every count, share, median and cross-tabulation the article prints is the correct result of the recorded classification. A reader who disagrees with a classification can change one row of the embedded data, and the harness will show which printed figures move.

Transcriptions and model theory. Figures quoted from the papers of Number 2 are checked against excerpts of the register abstracts, which re-measures nothing. Proposition 4 is true by construction and Proposition 8 is model-theoretic; neither is arithmetic, and both are marked RECORDED.

Purpose and Scope

This dossier accompanies Article 89, "The Chrysene Formalism as Candidate Mathematics and Candidate Biogenesis: A Quantitative and Categorical Survey of Volume 1, Number 2," the editorial introduction to Volume 1, Number 3. The article surveys the twenty-one papers of Number 2. It proves the registry-class count and the even-fibre obstruction, formalizes the two orthogonal review tiers, extends the classification vocabulary conservatively, stratifies the genetic-code results, and sets the protocell program of Number 3.

The verification script executes 137 checks in sixteen blocks (T1–T16):

- the embedded ledger data and its provenance;
- the frame group, Theorem 1, the Burnside computation and the stabilizer argument;
- the capacity ratio, and the orbit counts of every subgroup of Fr;
- the parity obstruction and the fibres of the two codes; normal form;
- transversality and the split-class count, both under a stated convention;
- every distribution, share, median and page figure of Section 5;
- the conservativity of the extension on Number 1;
- the clusters, the allocation rule, the lineage figure and the stratification;
- the transcriptions, and the scope.

All 137 checks passed with exit code 0.

The companion Lean 4 proof file `acf-89-proof.lean` provides kernel-checked certificates for 85 of these claims.

Exact throughout

No quantity in this harness is a float. The article's numbers are of two kinds, and both are exact by nature. Counts of marker assignments, classes, codons, papers and pages are integers. The printed decimals — shares, means, ratios — are ratios of such integers, which `fractions.Fraction` carries without loss.

Every printed decimal is certified by one integer device:

Device	Certifies	By the integer relation
`rounds_to(n,d,r,s)`	n/d rounds to r/s	$2\,\lvert ns - rd\rvert < d$

The thirty-five decimals certified this way are collected as they are checked. Check T16-19a then verifies that the device is not vacuous: every one of the thirty-five certificates *fails* when its printed figure is moved by one unit in the last place.

Medians are exact Fractions. The capacity ratio of Corollary 1 is compared as an identity between Fractions for every N from 1 to 60, so no power of two is ever approximated.

Block Index

Block	Topic	Checks
T1	The data and its provenance (TRANSCRIPTION)	8
T2	Def. 1: the frame group is a Klein four-group, $N = 1, \ldots, 8$	5
T3	Thm. 1: the registry-class count, by exhaustion, $N = 1, \ldots, 8$	8
T4	Thm. 1: Burnside's fixed points and the stabilizer argument	6
T5	Cor. 1: the capacity ratio; the multiset coincidence	5
T6	Orbit counts of the five subgroups of Fr at $N = 3$	4
T7	Prop. 2: all 2^{20} unions of classes have even cardinality	3
T8	Cor. 2: fibres of the standard and mitochondrial codes	10
T9	Normal form, and where the two codes differ	6
T10	Prop. 3: transversality (CONVENTION)	4
T11	Nineteen of twenty split classes (CONVENTION)	8
T12	Tables 2–3, Rem. 5, Sec. 5: distributions	16
T13	Prop. 5: every Number 1 figure of Art. 39 unchanged	9
T14	Sec. 5: page figures, means, medians, ranges	9
T15	Clusters, Prop. 6, Def. 6, Fig. 1, Def. 10	16
T16	Transcriptions from Number 2; rounding vacuity; scope	20
	Total	**137**

What the Harness Found

Four results go beyond restating the article, and they bear on how three of its sentences should be read.

1. The split count holds exactly for the Watson–Crick identifications (T10, T11). The article states, after Corollary 2, that under the identification of Art. 50 the mitochondrial code splits nineteen of the twenty classes. The harness cannot know Art. 50's identification. It therefore fixes the natural convention — letter k of a codon occupies sites $2k$ and $2k{+}1$ — and runs over all 24 ways of assigning the four letters to marker pairs. Under that convention:

- transversality (Proposition 3) holds for all 24;
- both codes split exactly 19 of 20 classes for precisely the 8 assignments in which complementing the markers is Watson–Crick complementation (T↔A, C↔G);
- for the other 16, both codes split all 20;
- in every one of the 8, the unsplit class is $\{\mathrm{AGU}, \mathrm{UCA}\}$, painted Ser in both codes, exactly as Art. 53 reports.

The count 8 of 24 also agrees with Art. 45's "eight of the twenty-four bijections". The article's sentence is therefore correct exactly when Art. 50's identification makes complementation Watson–

Crick, which is consistent with Art. 52's use of it (AAA $\mapsto$ UUU). These checks are marked CONVENTION, and the article records the qualification in its Remark 10 (Appendix B).

2. The parity obstruction is over-determined (T8-03). The article cites the singletons Met and Trp. The standard code has four odd fibres: Met (1), Trp (1), Ile (3) and stop (3). Corollary 2(1) is unaffected; it needs only one.

3. The counts 32 and 36 are subgroup orbit counts (T6). Of the five subgroups of Fr, exactly one, $\langle\varphi\rangle$, has 32 orbits on Ω_3. This is the count the article cites from Art. 48 for the case in which the frame data are physical. The residual count 36 of Art. 52 is the orbit count of $\langle\rho\rangle$. The first is an interpretation of Art. 48 and is recorded as such (T6-04); the second agrees with Art. 52's register abstract.

4. The page ratio does not depend on the skew (T14-09). The ratio of means rounds to 2.6, and the ratio of medians, 29/11, also rounds to 2.6.

Scope of Verification

What is verified.

- That Fr $= \langle\varphi, \rho\rangle$ is a Klein four-group on Ω_N for $N = 1, \ldots, 8$, the commutation being checked pointwise.
- Theorem 1 in full for $N = 1, \ldots, 8$: the count, the absence of orbits of size 1, and the split into 2^N classes of size 2 and $2^{2N-2} - 2^{N-1}$ of size 4; $20 = 8 + 12$ at $N = 3$ and 72 at $N = 4$; and each step of the Burnside proof.
- Corollary 1 as an exact identity for $N = 1, \ldots, 60$, and that the class and multiset counts coincide only at $N = 3$ in that range.
- Proposition 2 by exhaustion: all 1 048 576 unions of the twenty classes have even cardinality.
- Corollary 2 against NCBI tables 1 and 2: the singletons, the odd fibres, the complete mitochondrial fibre profile, and the ten amino acids the article lists.
- That the two codes differ at exactly four codons in three purine half-blocks, that the mitochondrial code is in normal form at all sixteen blocks, and that the standard code leaves it at exactly two.
- Every count, share, mean, median, range and page figure of Section 5 and Tables 2–3; and that every Number 1 figure of Art. 39 is reproduced unchanged from the extended ledger.
- That the clusters partition J, that Proposition 6 survives either classification of Art. 48, that the allocation rule reproduces the recorded tiers, that Figure 1 is acyclic, and that Definition 10's precedence is needed for exactly two rows.

What is not verified.

- **The classifications.** Category, basis, technique, cluster, tier and stratum are single-coder judgments. The harness checks their consequences, not their correctness (T16-19).

- **Art. 50's identification.** Blocks T10–T11 hold under the consecutive-pair convention; that Art. 50 uses it is assumed (T10-04).
- **Propositions 4 and 8, and Remark 9.** The first is true by construction; the others are model-theoretic. No arithmetic settles them (T16-17, T16-18).
- **The allocation rule as a judgment.** The harness checks that Definition 6 reproduces the recorded tiers. That it is the right rule is an editorial judgment, and its test is Number 3 (T15-09).
- **The papers of Number 2.** Figures quoted from them are compared with the register abstracts and with nothing else. Two are recomputed independently and agree: the 72 classes at $N = 4$ and the residual count 36.

Regression, not independence. The harness encodes the article's conclusions; T15-07 fails if the allocation rule stops reproducing the tiers. A published verification artifact should stop passing when the claims it supports stop holding. That makes it a regression test on the conclusions rather than an independent test of them.

Script Listing: `verify_A89.py`

verify_A89.py — complete source

```
#!/usr/bin/env python3
"""
verify_A89.py -- verification harness for Article 89 of the
Annals of the Chrysene Formalism (Vol. 1, No. 4, November 2026,
first page 1808):

    "The Chrysene Formalism as Candidate Mathematics and Candidate
     Biogenesis: A Quantitative and Categorical Survey of Volume 1,
     Number 2"

Usage
    python3 verify_A89.py        # block summaries and the total
    python3 verify_A89.py -v     # every check, one line each

Exit code 0 if and only if every check passes.

Dependencies: Python 3.8+ standard library only.  No floating point is
used anywhere: every quantity is an int or a fractions.Fraction, and every
printed decimal is certified by an integer rounding bound.

Check kinds
    EXACT          an exact computation or exhaustion
    TRANSCRIPTION  agreement between a figure in the article and the same
                   figure in an embedded source (ledger, register, NCBI)
    CONVENTION     exact, but under a stated convention that the article's
                   source paper is assumed, not shown, to use
    RECORDED       an obligation or limit carried with the numbers; it
                   asserts nothing and is counted as passed by design
"""

import sys
from fractions import Fraction
from itertools import permutations
from math import comb

VERBOSE = "-v" in sys.argv[1:]

# ---------------------------------------------------------------------
#  Check machinery
# ---------------------------------------------------------------------
RESULTS = []           # (block, ident, kind, description, passed)
BLOCK_TITLES = {}


def block(tag, title):
    BLOCK_TITLES[tag] = title


def check(ident, desc, cond, kind="EXACT"):
    tag = ident.split("-")[0]
    RESULTS.append((tag, ident, kind, desc, bool(cond)))


ROUND_LOG = []         # every (n, d, r, s) certified, for the vacuity test
```

```

def _bound(n, d, r, s):
    return 2 * abs(n * s - r * d) < d


def rounds_to(n, d, r, s):
    """n/d rounds to r/s (half-unit bound), decided in integers:
    2|ns - rd| < d."""
    ROUND_LOG.append((n, d, r, s))
    return _bound(n, d, r, s)


def pct_rounds(k, n, tenths):
    """k/n, as a percentage, rounds to tenths/10 per cent."""
    return rounds_to(100 * k, n, tenths, 10)


# ====================================================================
#  EMBEDDED DATA
#  Source: the Volume 1 ledger (acf_vol1_ledger.xlsx), tabs
#  'Vol 1 No 1', 'Vol 1 No 2', 'Molecular vs Mathematical',
#  'Techniques by Article', 'AI Review Process'.
#  ARTICLES rows: (art, first page, last page, category, basis, techniques)
#  Category: PM CB AT QA CR MD FN ED.  Basis: Mol Math Meth.
# ====================================================================
ARTICLES = [
    ( 1,    1,    5, 'PM' , 'Math' , (0, 1, 2)),
    ( 2,    6,   14, 'PM' , 'Math' , (0, 3, 4, 5)),
    ( 3,   15,   26, 'CB' , 'Math' , (6, 2, 7)),
    ( 4,   27,   40, 'CB' , 'Mol'  , (8, 9)),
    ( 5,   41,   60, 'CB' , 'Mol'  , (8, 10, 9)),
    ( 6,   61,   77, 'MD' , 'Math' , (5, 11, 0)),
    ( 7,   78,   87, 'MD' , 'Math' , (5, 12, 2)),
    ( 8,   88,   97, 'MD' , 'Math' , (5, 13, 14)),
    ( 9,   98,  112, 'QA' , 'Mol'  , (15, 9)),
    (10,  113,  123, 'CR' , 'Math' , (8, 11, 0)),
    (11,  124,  134, 'CR' , 'Math' , (4, 3)),
    (12,  135,  148, 'PM' , 'Math' , (8, 0, 16)),
    (13,  149,  156, 'PM' , 'Math' , (16, 8, 1)),
    (14,  157,  166, 'MD' , 'Math' , (0, 3, 17)),
    (15,  167,  176, 'QA' , 'Mol'  , (15, 2, 1)),
    (16,  177,  186, 'AT' , 'Mol'  , (13, 15)),
    (17,  187,  205, 'CB' , 'Mol'  , (8, 6)),
    (18,  206,  219, 'CB' , 'Mol'  , (11, 9)),
    (19,  220,  232, 'CB' , 'Mol'  , (9, 13)),
    (20,  233,  247, 'CB' , 'Mol'  , (8, 9)),
    (21,  248,  258, 'CB' , 'Mol'  , (7, 2)),
    (22,  259,  272, 'CB' , 'Mol'  , (3, 9)),
    (23,  273,  282, 'CB' , 'Mol'  , (18, 1)),
    (24,  283,  291, 'CB' , 'Mol'  , (18, 7)),
    (25,  292,  304, 'CB' , 'Mol'  , (13, 2)),
    (26,  305,  321, 'CB' , 'Mol'  , (8, 1)),
    (27,  322,  331, 'FN' , 'Math' , (5, 9, 1)),
    (28,  332,  341, 'MD' , 'Mol'  , (14, 12, 9)),
    (29,  342,  351, 'FN' , 'Math' , (1, 13)),
    (30,  352,  363, 'CR' , 'Math' , (3, 0, 2)),
    (31,  364,  371, 'FN' , 'Math' , (9, 1, 5)),
    (32,  372,  385, 'MD' , 'Math' , (5, 3, 14)),
```

```
    (33,  386,  394, 'PM' , 'Math' , (19, 9, 4)),
    (34,  395,  402, 'PM' , 'Math' , (8, 16)),
    (35,  403,  413, 'AT' , 'Mol'  , (5, 11, 7, 12, 17)),
    (36,  414,  436, 'AT' , 'Mol'  , (15, 2)),
    (37,  437,  458, 'AT' , 'Mol'  , (15, 13)),
    (38,  459,  475, 'AT' , 'Mol'  , (15,)),
    (39,  476,  488, 'ED' , 'Meth' , (20, 0)),
    (40,  489,  560, 'PM' , 'Math' , (21, 0, 3, 2, 13)),
    (41,  561,  581, 'ED' , 'Meth' , (20,)),
    (42,  582,  602, 'PM' , 'Math' , (21, 13)),
    (43,  603,  645, 'PM' , 'Math' , (5, 22, 23, 0)),
    (44,  646,  675, 'PM' , 'Math' , (0, 3, 4, 2)),
    (45,  676,  704, 'PM' , 'Math' , (2, 24, 25)),
    (46,  705,  729, 'PM' , 'Math' , (5, 22, 23)),
    (47,  730,  767, 'CB' , 'Math' , (24, 2, 25)),
    (48,  768,  793, 'CB' , 'Mol'  , (2, 24)),
    (49,  794,  857, 'QA' , 'Mol'  , (15, 0, 23, 21)),
    (50,  858,  882, 'CB' , 'Math' , (24, 26, 2)),
    (51,  883,  913, 'CB' , 'Math' , (26, 24)),
    (52,  914,  925, 'CB' , 'Mol'  , (2, 24)),
    (53,  926,  934, 'CB' , 'Math' , (24, 2)),
    (54,  935, 1014, 'CB' , 'Mol'  , (27, 8, 2)),
    (55, 1015, 1047, 'CB' , 'Math' , (24,)),
    (56, 1048, 1075, 'CB' , 'Mol'  , (27, 25)),
    (57, 1076, 1106, 'CB' , 'Mol'  , (13, 24, 27)),
    (58, 1107, 1136, 'CB' , 'Mol'  , (2, 13, 27)),
    (59, 1137, 1162, 'CB' , 'Mol'  , (2, 27, 24)),
]
TECHNIQUES = [
    'Non-Commutative Geometry',  # 0
    'Algebraic Topology / Homology',  # 1
    'Group Theory / Symmetry',  # 2
    'Operator Algebras / C*-Algebras',  # 3
    'Finite Fields / Coding Theory',  # 4
    'H-infinity Robust Control',  # 5
    'General Relativity',  # 6
    'Monte Carlo / MCMC Simulation',  # 7
    'Category Theory / Topos Theory',  # 8
    'Thermodynamics / Statistical Mechanics',  # 9
    'KAM Theory',  # 10
    'Optimal Transport (Wasserstein)',  # 11
    'Hamilton-Jacobi-Isaacs / Differential Games',  # 12
    'Graph Theory / Spectral Graph Theory',  # 13
    'Stochastic Differential Equations',  # 14
    'Quantum Chemistry / DFT',  # 15
    'Representation Theory / Geometric Langlands',  # 16
    'Generative AI / Machine Learning',  # 17
    'Discrete Morse Theory',  # 18
    'Automata Theory',  # 19
    'Corpus & Case-Study Coding',  # 20
    'Mathematical Logic / Model Theory',  # 21
    'Spectral / Scattering Theory',  # 22
    'Linear Systems / State-Space Theory',  # 23
    'Combinatorics / Matching Theory',  # 24
    'Variational / Max-Min Optimization',  # 25
    'Statistical Inference / Hypothesis Testing',  # 26
    'Chemical Graph Rewriting / Reaction Theory',  # 27
]
REVIEW = {  # art: (author-side family, editor-side family, structure)
```

```
 1: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
 2: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
 3: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
 4: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
 5: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
 6: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
 7: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
 8: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
 9: ('Gemini 3.1 Pro (Extended) then Grok 4.6', 'Grok 4.6', 'Single-round'
    ),
10: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
11: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
12: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
13: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
14: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
15: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
16: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
17: ('Gemini 3.1 Pro (Extended) [Prelim]; Grok 4.6 [Final]', 'Gemini 3.1
    Pro (Extended) [Prelim]; Grok 4.6 [Final]', 'Two-round'),
18: ('Gemini 3.1 Pro (Extended) [Prelim]; Grok 4.6 [Final]', 'Gemini 3.1
    Pro (Extended) [Prelim]; Grok 4.6 [Final]', 'Two-round'),
19: ('Gemini 3.1 Pro (Extended) [Prelim]; Grok 4.6 [Final]', 'Gemini 3.1
    Pro (Extended) [Prelim]; Grok 4.6 [Final]', 'Two-round'),
20: ('Grok 4.6', 'Grok 4.6', 'Single-round'),
21: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
22: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
23: ('Grok 4.6 and Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)'
    , 'Single-round'),
24: ('Grok 4.6 then Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)
    ', 'Single-round'),
25: ('Grok 4.6', 'Grok 4.6', 'Single-round'),
26: ('Grok 4.6 then Grok 4.6 and Gemini 3.1 Pro (Extended)', 'Gemini 3.1
    Pro (Extended)', 'Single-round'),
27: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
28: ('Gemini 3.1 Pro (Extended) [Prelim]; Grok 4.6 [Final]', 'Gemini 3.1
    Pro (Extended) [Prelim]; Grok 4.6 [Final]', 'Two-round'),
29: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
30: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
    round'),
31: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
```

```
        round'),
    32: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
        round'),
    33: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
        round'),
    34: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
        round'),
    35: ('Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (Extended)', 'Single-
        round'),
    36: ('ChatGPT 4.6 then Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (
        Extended)', 'Single-round'),
    37: ('ChatGPT 4.6 then Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (
        Extended)', 'Single-round'),
    38: ('ChatGPT 4.6 then Gemini 3.1 Pro (Extended)', 'Gemini 3.1 Pro (
        Extended)', 'Single-round'),
    39: ('Claude Opus 5.5 / Opus 5.0 / Sonnet 3.7', 'Claude Opus 5.5 / Opus
        5.0 / Sonnet 3.7', 'Orthogonal, single-round'),
    40: ('ChatGPT 4.6', 'ChatGPT 4.6; Gemini 3.1 Pro (Extended); Grok 4.6', '
        Orthogonal, multi-pass'),
    41: ('Claude Opus 5.5 / Opus 5.0 / Sonnet 3.7', 'Claude Opus 5.5 / Opus
        5.0 / Sonnet 3.7', 'Orthogonal, single-round'),
    42: ('Claude Opus 5.5 / Opus 5.0 / Sonnet 3.7', 'Claude Opus 5.5 / Opus
        5.0 / Sonnet 3.7', 'Orthogonal, single-round'),
    43: ('ChatGPT 4.6', 'ChatGPT 4.6', 'Orthogonal, single-round'),
    44: ('Claude Opus 5.5 / Opus 5.0 / Sonnet 3.7', 'Claude Opus 5.5 / Opus
        5.0 / Sonnet 3.7', 'Orthogonal, multi-pass'),
    45: ('Claude Opus 5.5 / Opus 5.0 / Sonnet 3.7', 'Claude Opus 5.5 / Opus
        5.0 / Sonnet 3.7', 'Orthogonal, multi-pass'),
    46: ('ChatGPT 4.6', 'ChatGPT 4.6', 'Orthogonal, single-round'),
    47: ('Claude Opus 5.5 / Opus 5.0 / Sonnet 3.7', 'Claude Opus 5.5 / Opus
        5.0 / Sonnet 3.7', 'Orthogonal, single-round'),
    48: ('Claude Opus 5.5 / Opus 5.0 / Sonnet 3.7', 'Claude Opus 5.5 / Opus
        5.0 / Sonnet 3.7', 'Orthogonal, multi-pass'),
    49: ('Claude Opus 5.5 / Opus 5.0 / Sonnet 3.7', 'Claude Opus 5.5 / Opus
        5.0 / Sonnet 3.7', 'Orthogonal, single-round'),
    50: ('Claude Opus 5.5 / Opus 5.0 / Sonnet 3.7', 'Claude Opus 5.5 / Opus
        5.0 / Sonnet 3.7', 'Orthogonal, single-round'),
    51: ('Grok 4.6', 'Grok 4.6', 'Orthogonal, single-round'),
    52: ('Grok 4.6', 'Grok 4.6', 'Orthogonal, single-round'),
    53: ('Grok 4.6', 'Grok 4.6', 'Orthogonal, single-round'),
    54: ('Claude Opus 5.5 / Opus 5.0 / Sonnet 3.7', 'Claude Opus 5.5 / Opus
        5.0 / Sonnet 3.7', 'Orthogonal, single-round'),
    55: ('Claude Opus 5.5 / Opus 5.0 / Sonnet 3.7', 'Claude Opus 5.5 / Opus
        5.0 / Sonnet 3.7', 'Orthogonal, single-round'),
    56: ('Claude Opus 5.5 / Opus 5.0 / Sonnet 3.7', 'Claude Opus 5.5 / Opus
        5.0 / Sonnet 3.7', 'Orthogonal, single-round'),
    57: ('Claude Opus 5.5 / Opus 5.0 / Sonnet 3.7', 'Claude Opus 5.5 / Opus
        5.0 / Sonnet 3.7', 'Orthogonal, multi-pass'),
    58: ('Claude Opus 5.5 / Opus 5.0 / Sonnet 3.7', 'Claude Opus 5.5 / Opus
        5.0 / Sonnet 3.7', 'Orthogonal, single-round'),
    59: ('Claude Opus 5.5 / Opus 5.0 / Sonnet 3.7', 'Claude Opus 5.5 / Opus
        5.0 / Sonnet 3.7', 'Orthogonal, multi-pass'),
}


# NCBI translation tables 1 (standard) and 2 (vertebrate mitochondrial),
# 64 codons in TCAG order, '*' = stop.
STD = "FFLLSSSSYY**CC*WLLLLPPPPHHQQRRRRIIIMTTTTNNKKSSRRVVVVAAAADDEEGGGG"
MITO = "FFLLSSSSYY**CCWWLLLLPPPPHHQQRRRRIIMMTTTTNNKKSS**VVVVAAAADDEEGGGG"
```

```
BASES = "TCAG"

# Excerpts of the Number 2 abstracts (journal register), for T16.
REGISTER = {
    43: "Fourteen non-implications, twelve new deferred targets, a "
        "concordance table recording five corrections to the prior corpus, "
        "and a self-identified defect ledger are stated.",
    46: "Twelve non-implications, four new targets of which two are "
        "load-bearing, three corrections to the prior paper including one "
        "to a hypothesis its invariance theorem rests on, a "
        "seventeen-item defect ledger, and an explicit witness solved in "
        "closed form are stated.",
    45: "The twenty classes agree with the twenty three-letter multisets "
        "only at N=3; at N=4 the counts are seventy-two and thirty-five.",
    47: "The admissible bijections number one hundred and twenty-eight; "
        "... Five physical filters select one bijection uniquely, by a "
        "margin of 0.17% ... Existence of a coarsening is not rare, "
        "holding for seventy per cent of codes carrying the standard "
        "degeneracy profile.",
    50: "Seventeen seeds suffice, with deficiency zero and three classes to "
        "spare; exactly eight hundred and forty seed maps achieve it and "
        "none is distinguished. ... The construction's own output is one "
        "of the four of twenty-seven published translation tables that lie "
        "inside the range ... it excludes the standard code and does not "
        "exclude the mitochondrial one, which has no odd fibre yet splits "
        "nineteen of twenty classes.",
    53: "Both tables split nineteen of twenty frame classes and leave the "
        "same unsplit orbit {AGU, UCA}, painted Ser.",
    55: "All twenty-seven published translation tables lie inside the "
        "range against four of twenty-seven inside its predecessor's.",
    57: "The structure space is three-parameter ... at nineteen of twenty "
        "... the map to the underlying subgraph takes eleven values on the "
        "twenty structures ... And block degeneracy follows a rule at "
        "sixteen of sixteen blocks.",
    58: "a bridge would require a site within 1.02 Å of that centre",
    41: "Sixty-four review/process actions were identified, of which 63 "
        "were manuscript-directed.",
    52: "The residual frame group after the pattern is observed is <rev>, "
        "and the residual count is thirty-six.",
}

# ====================================================================
#  Frame group machinery (Definition 1)
# ====================================================================


def phi(n, x):
    """Complementation of all 2n markers."""
    return (1 << (2 * n)) - 1 - x


def rho(n, x):
    """Reversal of the 2n markers."""
    y = 0
    for i in range(2 * n):
        if (x >> i) & 1:
            y |= 1 << (2 * n - 1 - i)
    return y
```

```
def orbit(n, x):
    return frozenset({x, phi(n, x), rho(n, x), phi(n, rho(n, x))})


def orbits(n):
    return sorted({orbit(n, x) for x in range(1 << (2 * n))}, key=min)


def orbit_count_formula(n):
    return 2 ** (2 * n - 2) + 2 ** (n - 1)


# ---------------------------------------------------------------------
#  Derived corpus views
# ---------------------------------------------------------------------
ART = {a: dict(p0=p0, p1=p1, cat=c, basis=b, tech=t)
       for (a, p0, p1, c, b, t) in ARTICLES}
I1 = [a for a in ART if a <= 38]
I2 = [a for a in ART if 39 <= a <= 59]


def length(a):
    return ART[a]["p1"] - ART[a]["p0"] + 1


def count(arts, key, value):
    return sum(1 for a in arts if ART[a][key] == value)


def tech_count(arts, name):
    i = TECHNIQUES.index(name)
    return sum(1 for a in arts if i in ART[a]["tech"])


def family(s):
    if ";" in s:
        return "Mix"
    return s.split()[0]


def median2(values):
    """Median as an exact Fraction."""
    v = sorted(values)
    m = len(v)
    return Fraction(v[m // 2]) if m % 2 else Fraction(v[m // 2 - 1] + v[m //
        2], 2)


# =====================================================================
#  T1  The data and its provenance  (TRANSCRIPTION)
# =====================================================================
block("T1", "The data and its provenance")
check("T1-01", "59 articles: 38 in Number 1, 21 in Number 2",
      (len(ART), len(I1), len(I2)) == (59, 38, 21), "TRANSCRIPTION")
check("T1-02", "pages are contiguous: No. 1 spans 1--475, No. 2 spans
    476--1162",
      all(ART[a + 1]["p0"] == ART[a]["p1"] + 1 for a in range(1, 59))
      and ART[1]["p0"] == 1 and ART[38]["p1"] == 475
```

```
      and ART[39]["p0"] == 476 and ART[59]["p1"] == 1162, "TRANSCRIPTION")
check("T1-03", "every category label is one of the eight of Def. 8",
      all(ART[a]["cat"] in {"PM", "CB", "AT", "QA", "CR", "MD", "FN", "ED"}
          for a in ART), "TRANSCRIPTION")
check("T1-04", "every basis label is one of Mol, Math, Meth; Meth only in No.
    2",
      all(ART[a]["basis"] in {"Mol", "Math", "Meth"} for a in ART)
      and all(ART[a]["basis"] != "Meth" for a in I1), "TRANSCRIPTION")
check("T1-05", "28 techniques in T' = 20 original + 8 new (Def. 8)",
      len(TECHNIQUES) == 28 and len(set(TECHNIQUES)) == 28, "TRANSCRIPTION")
check("T1-06", "no No. 1 article uses a technique of T_new (T'|_I = T)",
      all(max(ART[a]["tech"]) < 20 for a in I1), "TRANSCRIPTION")
check("T1-07", "every article has at least one technique",
      all(len(ART[a]["tech"]) >= 1 for a in ART), "TRANSCRIPTION")
check("T1-08", "No. 2 review structures: 6 multi-pass, 15 single-round",
      sum(REVIEW[a][2] == "Orthogonal, multi-pass" for a in I2) == 6
      and sum(REVIEW[a][2] == "Orthogonal, single-round" for a in I2) == 15,
      "TRANSCRIPTION")

# ====================================================================
#  T2  The frame group (Definition 1)
# ====================================================================
block("T2", "Def. 1: the frame group is a Klein four-group")
NS = range(1, 9)
check("T2-01", "phi is an involution on Omega_N, N = 1..8",
      all(phi(n, phi(n, x)) == x for n in NS for x in range(1 << (2 * n))))
check("T2-02", "rho is an involution on Omega_N, N = 1..8",
      all(rho(n, rho(n, x)) == x for n in NS for x in range(1 << (2 * n))))
check("T2-03", "phi and rho commute (both give 1 - x_{2N+1-i}), N = 1..8",
      all(phi(n, rho(n, x)) == rho(n, phi(n, x))
          for n in NS for x in range(1 << (2 * n))))
check("T2-04", "phi != rho, and phi*rho is not the identity, N = 1..8",
      all(any(phi(n, x) != rho(n, x) for x in range(1 << (2 * n)))
          and any(phi(n, rho(n, x)) != x for x in range(1 << (2 * n)))
          for n in NS))
check("T2-05", "phi(x)_i = 1 - x_i: the bit-level reading of Eq. (1)",
      all(((phi(3, x) >> i) & 1) == 1 - ((x >> i) & 1)
          for x in range(64) for i in range(6)))

# ====================================================================
#  T3  Theorem 1: the registry-class count
# ====================================================================
block("T3", "Thm. 1: the registry-class count")
check("T3-01", "|Omega_N / Fr| = 2^(2N-2) + 2^(N-1) for N = 1..8",
      all(len(orbits(n)) == orbit_count_formula(n) for n in NS))
check("T3-02", "every orbit has cardinality 2 or 4, none 1, N = 1..8",
      all(len(o) in (2, 4) for n in NS for o in orbits(n)))
check("T3-03", "exactly 2^N orbits of cardinality 2, N = 1..8",
      all(sum(len(o) == 2 for o in orbits(n)) == 2 ** n for n in NS))
check("T3-04", "exactly 2^(2N-2) - 2^(N-1) orbits of cardinality 4, N = 1..8",
      all(sum(len(o) == 4 for o in orbits(n)) == 2 ** (2 * n - 2) - 2 ** (n -
          1)
          for n in NS))
O3 = orbits(3)
check("T3-05", "N = 3: twenty classes, eight of size 2 and twelve of size 4",
      (len(O3), sum(len(o) == 2 for o in O3), sum(len(o) == 4 for o in O3))
      == (20, 8, 12))
check("T3-06", "N = 4: seventy-two classes",
      len(orbits(4)) == 72)
```

```
check("T3-07", "orbit-stabilizer: the (class, element) pairs number 64 at N =
    3",
      sum(len(o) for o in O3) == 64)
check("T3-08", "the classes partition Omega_3 (disjoint and covering)",
      len(set().union(*O3)) == 64 and sum(len(o) for o in O3) == 64)

# ================================================================
#  T4  Burnside's lemma and the stabilizer argument
# ================================================================
block("T4", "Thm. 1: Burnside and the stabilizers")
fix = lambda n, g: sum(1 for x in range(1 << (2 * n)) if g(x) == x)
check("T4-01", "|Fix(phi)| = 0 for N = 1..8",
      all(fix(n, lambda x, n=n: phi(n, x)) == 0 for n in NS))
check("T4-02", "|Fix(rho)| = 2^N (palindromes) for N = 1..8",
      all(fix(n, lambda x, n=n: rho(n, x)) == 2 ** n for n in NS))
check("T4-03", "|Fix(phi rho)| = 2^N (anti-palindromes) for N = 1..8",
      all(fix(n, lambda x, n=n: phi(n, rho(n, x))) == 2 ** n for n in NS))
check("T4-04", "Burnside: (2^(2N) + 0 + 2^N + 2^N)/4 = 2^(2N-2) + 2^(N-1)",
      all(Fraction(2 ** (2 * n) + 2 * 2 ** n, 4) == orbit_count_formula(n)
          for n in NS))
check("T4-05", "Fix(rho) and Fix(phi rho) are disjoint (their meet is in Fix(
    phi))",
      all(not (rho(n, x) == x and phi(n, rho(n, x)) == x)
          for n in NS for x in range(1 << (2 * n))))
check("T4-06", "every stabilizer has order 1 or 2",
      all(1 + (rho(n, x) == x) + (phi(n, rho(n, x)) == x) + (phi(n, x) == x)
          in (1, 2)
          for n in NS for x in range(1 << (2 * n))))

# ================================================================
#  T5  Corollary 1: the capacity ratio
# ================================================================
block("T5", "Cor. 1: the capacity ratio")
check("T5-01", "|X_N| / 4^N = 1/4 + 2^-(N+1) exactly, N = 1..60",
      all(Fraction(orbit_count_formula(n), 4 ** n)
          == Fraction(1, 4) + Fraction(1, 2 ** (n + 1)) for n in range(1, 61))
          )
check("T5-02", "the ratio is < 1 for every N = 1..60",
      all(orbit_count_formula(n) < 4 ** n for n in range(1, 61)))
check("T5-03", "the ratio is strictly decreasing in N, so no N is singled out"
    ,
      all(Fraction(orbit_count_formula(n), 4 ** n)
          > Fraction(orbit_count_formula(n + 1), 4 ** (n + 1)) for n in range
              (1, 60)))
check("T5-04", "class counts equal the multiset counts C(N+3,3) only at N = 3
    (N = 1..60)",
      [n for n in range(1, 61) if orbit_count_formula(n) == comb(n + 3, 3)] ==
          [3])
check("T5-05", "at N = 4: 72 classes against 35 multisets (Art. 45's
    refutation)",
      (orbit_count_formula(4), comb(7, 3)) == (72, 35))

# ================================================================
#  T6  Orbit counts of the subgroups of Fr (consistency)
# ================================================================
block("T6", "Orbit counts of the subgroups of Fr")


def sub_orbits(gens):
```

```
    seen, out = set(), 0
    for x in range(64):
        if x in seen:
            continue
        o = {x}
        frontier = [x]
        while frontier:
            y = frontier.pop()
            for g in gens:
                z = g(y)
                if z not in o:
                    o.add(z)
                    frontier.append(z)
        seen |= o
        out += 1
    return out


P3 = lambda x: phi(3, x)
R3 = lambda x: rho(3, x)
PR3 = lambda x: phi(3, rho(3, x))
SUB = {"1": sub_orbits([]), "<phi>": sub_orbits([P3]), "<rho>": sub_orbits([R3
    ]),
       "<phi rho>": sub_orbits([PR3]), "Fr": sub_orbits([P3, R3])}
check("T6-01", "orbit counts: 1 -> 64, <phi> -> 32, <rho> -> 36, <phi rho> ->
    36, Fr -> 20",
      SUB == {"1": 64, "<phi>": 32, "<rho>": 36, "<phi rho>": 36, "Fr": 20})
check("T6-02", "the count 32 cited from Art. 48 is attained by exactly one
    subgroup, <phi>",
      [k for k, v in SUB.items() if v == 32] == ["<phi>"])
check("T6-03", "the residual count 36 of Art. 52 is the orbit count of <rho>",
      SUB["<rho>"] == 36 and "thirty-six" in REGISTER[52], "TRANSCRIPTION")
check("T6-04", "reading: the 32 of the article is the <phi> count (an
    interpretation, not Art. 48's proof)",
      True, "RECORDED")

# ======================================================================
#  T7  Proposition 2: even fibres
# ======================================================================
block("T7", "Prop. 2: even fibres")
S2 = sum(1 << k for k, o in enumerate(O3) if len(o) == 2)
S4 = sum(1 << k for k, o in enumerate(O3) if len(o) == 4)
check("T7-01", "all 2^20 = 1,048,576 unions of classes at N = 3 have even
    cardinality",
      all((2 * bin(m & S2).count("1") + 4 * bin(m & S4).count("1")) % 2 == 0
          for m in range(1 << 20)))
check("T7-02", "every orbit has even cardinality for N = 1..8 (the general
    case)",
      all(len(o) % 2 == 0 for n in NS for o in orbits(n)))
check("T7-03", "phi maps every class to itself and has no fixed point (the
    involution argument)",
      all(phi(3, x) in o and phi(3, x) != x for o in O3 for x in o))

# ======================================================================
#  T8  Corollary 2: the two codes
# ======================================================================
block("T8", "Cor. 2: the standard and mitochondrial codes")
```

```
def fibres(table):
    return {c: table.count(c) for c in sorted(set(table))}


FS, FM = fibres(STD), fibres(MITO)


def cod(i):
    return BASES[i // 16] + BASES[(i // 4) % 4] + BASES[i % 4]


check("T8-01", "both tables have 64 codons over 20 amino acids and a stop symbol",
      len(STD) == len(MITO) == 64 and len(FS) == len(FM) == 21, "TRANSCRIPTION")
check("T8-02", "standard: Met = {AUG} and Trp = {UGG} are singletons",
      [cod(i) for i in range(64) if STD[i] == "M"] == ["ATG"]
      and [cod(i) for i in range(64) if STD[i] == "W"] == ["TGG"])
check("T8-03", "standard: the odd fibres are exactly Met (1), Trp (1), Ile (3), stop (3)",
      {c: k for c, k in FS.items() if k % 2} == {"M": 1, "W": 1, "I": 3, "*": 3})
check("T8-04", "hence c_std is class-functional for no beta (Cor. 2(1))",
      any(k % 2 for k in FS.values()))
check("T8-05", "mitochondrial: every fibre has even cardinality (Cor. 2(2))",
      all(k % 2 == 0 for k in FM.values()))
check("T8-06", "mitochondrial: AUA -> Met and UGA -> Trp, so both fibres have size 2",
      MITO[BASES.index("A") * 16 + 0 * 4 + 2] == "M"
      and MITO[0 * 16 + 3 * 4 + 2] == "W" and FM["M"] == 2 and FM["W"] == 2)
check("T8-07", "mitochondrial: Leu and Ser have 6; Ala Gly Pro Thr Val Arg and stop have 4",
      FM["L"] == FM["S"] == 6
      and all(FM[c] == 4 for c in "AGPTVR*"))
check("T8-08", "mitochondrial: the ten listed amino acids have fibres of size 2",
      all(FM[c] == 2 for c in "FYHQNKDECI"))
check("T8-09", "mitochondrial fibre sizes account for all 64 codons",
      2 * 6 + 7 * 4 + 12 * 2 == 64 == sum(FM.values()))
check("T8-10", "stop in the mitochondrial table is {UAA, UAG, AGA, AGG}",
      sorted(cod(i) for i in range(64) if MITO[i] == "*") == ["AGA", "AGG", "TAA", "TAG"])

# ================================================================
#  T9  Normal form and the four differing codons
# ================================================================
block("T9", "Normal form; where the two codes differ")


def half_uniform(table, blk, purine):
    a, b = (2, 3) if purine else (0, 1)   # TCAG: T,C pyrimidine; A,G purine
    return table[4 * blk + a] == table[4 * blk + b]


def normal_blocks(table):
    return [blk for blk in range(16)
            if half_uniform(table, blk, False) and half_uniform(table, blk, True)]
```

```
DIFF = [i for i in range(64) if STD[i] != MITO[i]]
check("T9-01", "the codes differ at exactly four codons: AGA, AGG, AUA, UGA",
      sorted(cod(i) for i in DIFF) == ["AGA", "AGG", "ATA", "TGA"])
check("T9-02", "the four codons lie in three purine half-blocks (AG, AU, UG)",
      sorted({cod(i)[:2] for i in DIFF}) == ["AG", "AT", "TG"]
      and all(cod(i)[2] in "AG" for i in DIFF))
check("T9-03", "the mitochondrial code is in normal form at all 16 blocks",
      len(normal_blocks(MITO)) == 16)
check("T9-04", "the standard code leaves normal form at exactly two blocks, AU and UG",
      sorted(cod(4 * b)[:2] for b in range(16) if b not in normal_blocks(STD))
          == ["AT", "TG"])
check("T9-05", "at the AG purine half both codes are uniform (stop/stop; Arg/Arg)",
      half_uniform(STD, 2 * 4 + 3, True) and half_uniform(MITO, 2 * 4 + 3, True))
check("T9-06", "standard = mitochondrial with two further divisions (at AUA and UGA)",
      sorted(cod(i) for i in DIFF if not half_uniform(STD, i // 4, True)) == ["ATA", "TGA"])

# ================================================================
#  T10  Proposition 3: transversality  (CONVENTION)
# ================================================================
block("T10", "Prop. 3: transversality (convention)")
LABELINGS = list(permutations(range(4)))   # letter index (TCAG) -> marker pair value


def word_of(lab, x):
    """Codon index of the 6-marker assignment x when letter k occupies
    sites (2k, 2k+1) and carries the marker pair lab[letter]."""
    inv = {v: k for k, v in enumerate(lab)}
    return inv[x >> 4] * 16 + inv[(x >> 2) & 3] * 4 + inv[x & 3]


def transverse(lab):
    return all(len({word_of(lab, y) // 4 for y in o}) == len(o) for o in O3)


check("T10-01", "class and block meet in at most one word, for all 24 letter labelings",
      all(transverse(lab) for lab in LABELINGS), "CONVENTION")
check("T10-02", "reason: rho or phi*rho keeps x's block only when it fixes x",
      all(not (g(x) != x and g(x) >> 2 == x >> 2) for x in range(64)
          for g in (P3, R3, PR3)), "CONVENTION")
check("T10-03", "the 20 classes and 16 blocks give 64 = sum of class sizes = 16 x 4",
      sum(len(o) for o in O3) == 16 * 4 == 64)
check("T10-04", "that Art. 50 uses the consecutive-pair convention is assumed, not shown",
      True, "RECORDED")

# ================================================================
#  T11  The mitochondrial code splits nineteen of twenty classes
# ================================================================
block("T11", "Nineteen of twenty split classes (convention)")
```

```
def split_count(table, lab):
    return sum(len({table[word_of(lab, y)] for y in o}) > 1 for o in O3)


def unsplit(table, lab):
    return [sorted(cod(word_of(lab, y)).replace("T", "U") for y in o)
            for o in O3 if len({table[word_of(lab, y)] for y in o}) == 1]


def watson_crick(lab):
    """Complementing the markers realises T<->A and C<->G."""
    T, C, A, G = lab
    return T ^ 3 == A and C ^ 3 == G


SPLIT = {lab: (split_count(STD, lab), split_count(MITO, lab)) for lab in
    LABELINGS}
WC = [lab for lab in LABELINGS if watson_crick(lab)]
check("T11-01", "exactly 8 of the 24 labelings make complementation Watson--
    Crick",
      len(WC) == 8, "CONVENTION")
check("T11-02", "under each of those 8, both codes split exactly 19 of the 20
    classes",
      all(SPLIT[lab] == (19, 19) for lab in WC), "CONVENTION")
check("T11-03", "under each of the other 16, both codes split all 20 classes",
      all(SPLIT[lab] == (20, 20) for lab in LABELINGS if lab not in WC), "
          CONVENTION")
check("T11-04", "the unsplit class is {AGU, UCA}, painted Ser in both codes",
      all(unsplit(STD, lab) == unsplit(MITO, lab) == [["AGU", "UCA"]] for lab
          in WC)
      and STD[2 * 16 + 3 * 4 + 0] == MITO[2 * 16 + 3 * 4 + 0] == "S"
      and STD[0 * 16 + 1 * 4 + 2] == MITO[0 * 16 + 1 * 4 + 2] == "S", "
          CONVENTION")
check("T11-05", "agreement with Art. 53's register abstract (19 of 20, {AGU,
    UCA}, Ser)",
      "nineteen of twenty" in REGISTER[53] and "{AGU, UCA}" in REGISTER[53]
      and "Ser" in REGISTER[53], "TRANSCRIPTION")
check("T11-06", "so c_mt is not class-functional under those identifications (
    text after Cor. 2)",
      all(SPLIT[lab][1] > 0 for lab in WC), "CONVENTION")
check("T11-08", "Cor. 2(1) directly: c_std splits at least one class under all
     24 labelings",
      all(SPLIT[lab][0] > 0 for lab in LABELINGS), "CONVENTION")
check("T11-07", "the count 8 of 24 agrees with Art. 45's 'eight of the twenty-
    four bijections'",
      len(WC) == 8, "CONVENTION")

# ====================================================================
#  T12  Distributions (Tables 2, 3; Remark 5; Section 5)
# ====================================================================
block("T12", "Tables 2-3, Rem. 5: distributions")
CATS = ["CB", "PM", "MD", "AT", "CR", "FN", "QA", "ED"]
T2N1 = [13, 6, 6, 5, 3, 3, 2, 0]
T2N2 = [12, 6, 0, 0, 0, 0, 1, 2]
check("T12-01", "Table 2, No. 1 column: 13, 6, 6, 5, 3, 3, 2, 0",
      [count(I1, "cat", c) for c in CATS] == T2N1)
check("T12-02", "Table 2, No. 2 column: 12, 6, 0, 0, 0, 0, 1, 2",
      [count(I2, "cat", c) for c in CATS] == T2N2)
```

```
check("T12-03", "Table 2, Vol. 1 column and totals 38, 21, 59",
      [count(list(ART), "cat", c) for c in CATS] == [25, 12, 6, 5, 3, 3, 3, 2]
      and (sum(T2N1), sum(T2N2)) == (38, 21))
check("T12-04", "Rem. 5: 12 > 21/2, and shares 34.2 -> 57.1 %, 15.8 -> 28.6 %"
    ,
      2 * 12 > 21 and pct_rounds(13, 38, 342) and pct_rounds(12, 21, 571)
      and pct_rounds(6, 38, 158) and pct_rounds(6, 21, 286))
check("T12-05", "Rem. 5: four of the seven No. 1 categories have no No. 2
    paper",
      sum(1 for c in CATS[:7] if count(I2, "cat", c) == 0) == 4)
check("T12-06", "basis, No. 2: Math 11 (52.4 %), Mol 8 (38.1 %), Meth 2 (9.5
    %)",
      (count(I2, "basis", "Math"), count(I2, "basis", "Mol"), count(I2, "basis
          ", "Meth"))
      == (11, 8, 2) and pct_rounds(11, 21, 524) and pct_rounds(8, 21, 381)
      and pct_rounds(2, 21, 95))
check("T12-07", "basis, No. 1: Math 18, Mol 20, Meth 0",
      (count(I1, "basis", "Math"), count(I1, "basis", "Mol"), count(I1, "basis
          ", "Meth"))
      == (18, 20, 0))
check("T12-08", "tiers: 6 multi-pass (28.6 %), 15 single-round (71.4 %)",
      pct_rounds(6, 21, 286) and pct_rounds(15, 21, 714))
FAM = {a: (family(REVIEW[a][0]), family(REVIEW[a][1])) for a in I2}
check("T12-09", "models: Claude both sides 15 (71.4 %), Grok 3, ChatGPT 2,
    mixed 1",
      sum(f == ("Claude", "Claude") for f in FAM.values()) == 15
      and sum(f == ("Grok", "Grok") for f in FAM.values()) == 3
      and sum(f == ("ChatGPT", "ChatGPT") for f in FAM.values()) == 2
      and sum("Mix" in f for f in FAM.values()) == 1 and pct_rounds(15, 21,
          714))
check("T12-10", "20 of 21 papers used one model family on both sides (Sec.
    9.5)",
      sum(f[0] == f[1] for f in FAM.values()) == 20)
T3ROWS = [("Group Theory / Symmetry", 11, 8),
          ("Combinatorics / Matching Theory", 10, None),
          ("Non-Commutative Geometry", 5, 7),
          ("Chemical Graph Rewriting / Reaction Theory", 5, None),
          ("Graph Theory / Spectral Graph Theory", 4, 6),
          ("Mathematical Logic / Model Theory", 3, None),
          ("Linear Systems / State-Space Theory", 3, None),
          ("Variational / Max-Min Optimization", 3, None)]
check("T12-11", "Table 3: the eight rows, No. 2 counts and No. 1 counts",
      all(tech_count(I2, t) == n2 and (n1 is None or tech_count(I1, t) == n1)
          for t, n2, n1 in T3ROWS))
check("T12-12", "Table 3: the daggered rows are exactly those in T_new",
      all((n1 is None) == (TECHNIQUES.index(t) >= 20) for t, _, n1 in T3ROWS))
check("T12-13", "sixteen of the twenty-eight techniques occur in No. 2",
      sum(1 for t in TECHNIQUES if tech_count(I2, t) > 0) == 16)
check("T12-14", "thermodynamics (11 in No. 1) and algebraic topology (8) do
    not occur in No. 2",
      (tech_count(I1, "Thermodynamics / Statistical Mechanics"),
       tech_count(I2, "Thermodynamics / Statistical Mechanics"),
       tech_count(I1, "Algebraic Topology / Homology"),
       tech_count(I2, "Algebraic Topology / Homology")) == (11, 0, 8, 0))
check("T12-15", "techniques with 2 occurrences in No. 2: H-inf, op. alg.,
    spectral, stat., corpus",
      sorted(t for t in TECHNIQUES if tech_count(I2, t) == 2) == sorted(
          ["H-infinity Robust Control", "Operator Algebras / C*-Algebras",
           "Spectral / Scattering Theory", "Statistical Inference / Hypothesis
```

```
                Testing",
711             "Corpus & Case-Study Coding"]))
check("T12-16", "techniques with 1 occurrence in No. 2: finite fields,
    category theory, QC",
      sorted(t for t in TECHNIQUES if tech_count(I2, t) == 1) == sorted(
          ["Finite Fields / Coding Theory", "Category Theory / Topos Theory",
           "Quantum Chemistry / DFT"]))
716
# ================================================================
#  T13  Proposition 5: conservativity on Number 1
# ================================================================
block("T13", "Prop. 5: conservativity on Number 1")
721 check("T13-01", "Art. 39 Table 1 (categories) reproduced from the extended
    ledger",
      [count(I1, "cat", c) for c in ["CB", "PM", "MD", "AT", "CR", "FN", "QA"
          ]]
      == [13, 6, 6, 5, 3, 3, 2], "TRANSCRIPTION")
check("T13-02", "Art. 39 Table 1 shares: 34.2, 15.8, 15.8, 13.2, 7.9, 7.9, 5.3
     %",
      all(pct_rounds(k, 38, s) for k, s in
726           [(13, 342), (6, 158), (6, 158), (5, 132), (3, 79), (3, 79), (2, 53)
              ]))
A39T2 = [("Thermodynamics / Statistical Mechanics", 11, 289),
         ("Category Theory / Topos Theory", 9, 237),
         ("Algebraic Topology / Homology", 8, 211),
         ("H-infinity Robust Control", 8, 211),
731      ("Group Theory / Symmetry", 8, 211),
         ("Non-Commutative Geometry", 7, 184),
         ("Operator Algebras / C*-Algebras", 6, 158),
         ("Graph Theory / Spectral Graph Theory", 6, 158),
         ("Quantum Chemistry / DFT", 6, 158)]
736 check("T13-03", "Art. 39 Table 2 (nine most frequent techniques) reproduced",
      all(tech_count(I1, t) == k for t, k, _ in A39T2), "TRANSCRIPTION")
check("T13-04", "Art. 39 Table 2 shares certified by rounding bound",
      all(pct_rounds(k, 38, s) for _, k, s in A39T2))
check("T13-05", "Art. 39's omitted eleven: six at 3--4, two at 2, two at 1 (
    and Discrete Morse at 2)",
741   all(tech_count(I1, t) in (3, 4) for t in
          ["Optimal Transport (Wasserstein)", "Monte Carlo / MCMC Simulation",
           "Finite Fields / Coding Theory", "Hamilton-Jacobi-Isaacs /
               Differential Games",
           "Stochastic Differential Equations",
           "Representation Theory / Geometric Langlands"])
746   and tech_count(I1, "General Relativity") == 2
      and tech_count(I1, "Generative AI / Machine Learning") == 2
      and tech_count(I1, "KAM Theory") == 1 and tech_count(I1, "Automata
          Theory") == 1)
check("T13-06", "Art. 39 basis split 20 Mol / 18 Math (52.6 / 47.4 %)",
      count(I1, "basis", "Mol") == 20 and pct_rounds(20, 38, 526)
751   and pct_rounds(18, 38, 474))
check("T13-07", "Art. 39 review figures: 34 single (89.5 %), 4 two-round (10.5
     %)",
      sum(REVIEW[a][2] == "Single-round" for a in I1) == 34
      and sum(REVIEW[a][2] == "Two-round" for a in I1) == 4
      and pct_rounds(34, 38, 895) and pct_rounds(4, 38, 105))
756 FAM1 = {a: (REVIEW[a][0], REVIEW[a][1]) for a in I1}
G1 = "Gemini 3.1 Pro (Extended)"
check("T13-08", "Art. 39 model figures: 25 Gemini-only, 2 Grok-only, 11 hand-
    off",
```

```
      sum(f == (G1, G1) for f in FAM1.values()) == 25
      and sum(f == ("Grok 4.6", "Grok 4.6") for f in FAM1.values()) == 2
      and 38 - 25 - 2 == 11 and pct_rounds(25, 38, 658) and pct_rounds(11, 38,
          289))
check("T13-09", "T_new is unassessed on No. 1: its No. 1 counts are 0 by
    construction",
      all(tech_count(I1, t) == 0 for t in TECHNIQUES[20:]))

# ================================================================
#  T14  Pages (Section 5)
# ================================================================
block("T14", "Section 5: page figures")
L1 = [length(a) for a in I1]
L2 = [length(a) for a in I2]
OM = [a for a in I2 if REVIEW[a][2] == "Orthogonal, multi-pass"]
check("T14-01", "No. 2 spans 687 pages; No. 1 spans 475",
      (sum(L2), sum(L1)) == (687, 475))
check("T14-02", "mean 687/21 rounds to 32.7; 475/38 = 12.5 exactly",
      rounds_to(687, 21, 327, 10) and Fraction(475, 38) == Fraction(25, 2))
check("T14-03", "ratio of means rounds to 2.6",
      rounds_to(687 * 38, 21 * 475, 26, 10))
check("T14-04", "medians: 29 pages (No. 2) and 11 pages (No. 1)",
      (median2(L2), median2(L1)) == (29, 11))
check("T14-05", "ranges: 9--80 (No. 2) and 5--23 (No. 1)",
      (min(L2), max(L2), min(L1), max(L1)) == (9, 80, 5, 23))
check("T14-06", "the three longest: Art. 54 (80), Art. 40 (72), Art. 49 (64)",
      sorted(I2, key=length, reverse=True)[:3] == [54, 40, 49]
      and [length(a) for a in (54, 40, 49)] == [80, 72, 64])
check("T14-07", "tier pages: multi-pass 214, single-round 473",
      sum(length(a) for a in OM) == 214 and 687 - 214 == 473)
check("T14-08", "the multi-pass set is {40, 44, 45, 48, 57, 59}",
      OM == [40, 44, 45, 48, 57, 59])
check("T14-09", "medians ratio 29/11 also rounds to 2.6 (the skew does not
    drive the ratio)",
      rounds_to(29, 11, 26, 10))

# ================================================================
#  T15  Clusters, Proposition 6, allocation, lineage, strata
# ================================================================
block("T15", "Clusters, Prop. 6, Def. 6, Fig. 1, Def. 10")
CLUSTERS = {1: [40, 42], 2: [43, 46], 3: [44, 45],
            4: [47, 48, 50, 51, 52, 53, 55, 57], 5: [54, 56, 58, 59],
            6: [49], 7: [39, 41]}
flat = sorted(a for v in CLUSTERS.values() for a in v)
check("T15-01", "the seven clusters partition J = {39..59} (Def. 9)",
      flat == list(range(39, 60)))
check("T15-02", "cluster sizes 2, 2, 2, 8, 4, 1, 2",
      [len(CLUSTERS[k]) for k in range(1, 8)] == [2, 2, 2, 8, 4, 1, 2])
check("T15-03", "Gamma_4 and Gamma_5 lie in Cosmological & Biological
    Emergence",
      all(ART[a]["cat"] == "CB" for a in CLUSTERS[4] + CLUSTERS[5]))
check("T15-04", "Prop. 6 witnesses: Basis(47) = Math, Basis(57) = Mol, both in
     Gamma_4",
      ART[47]["basis"] == "Math" and ART[57]["basis"] == "Mol")
check("T15-05", "Prop. 6 survives either classification of Art. 48",
      all(len({("Mol" if a == 48 and alt else ART[a]["basis"]) for a in
          CLUSTERS[4]}) == 2
          for alt in (False, True)))
check("T15-06", "C&B basis split: 7 Mol / 5 Math in No. 2; 12 / 1 in No. 1",
```

```
      sum(1 for a in I2 if ART[a]["cat"] == "CB" and ART[a]["basis"] == "Mol")
          == 7
      and sum(1 for a in I2 if ART[a]["cat"] == "CB" and ART[a]["basis"] == "
          Math") == 5
      and sum(1 for a in I1 if ART[a]["cat"] == "CB" and ART[a]["basis"] == "
          Mol") == 12
      and sum(1 for a in I1 if ART[a]["cat"] == "CB" and ART[a]["basis"] == "
          Math") == 1)
CLAUSE_A, CLAUSE_B, CLAUSE_C = {40, 44, 45}, {45, 48, 59}, {57}
check("T15-07", "Def. 6: clauses (a) U (b) U (c) = the recorded multi-pass set
    ",
      sorted(CLAUSE_A | CLAUSE_B | CLAUSE_C) == OM)
check("T15-08", "clause (c) does not select Art. 50 or 55",
      not ({50, 55} & (CLAUSE_A | CLAUSE_B | CLAUSE_C)))
check("T15-09", "the rule is an ex post reconstruction; its test is No. 3",
      True, "RECORDED")
EDGES = [(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)]
NODES = sorted({a for e in EDGES for a in e})


def acyclic(nodes, edges):
    indeg = {v: 0 for v in nodes}
    for _, v in edges:
        indeg[v] += 1
    queue, seen = [v for v in nodes if indeg[v] == 0], 0
    while queue:
        u = queue.pop()
        seen += 1
        for (a, b) in edges:
            if a == u:
                indeg[b] -= 1
                if indeg[b] == 0:
                    queue.append(b)
    return seen == len(nodes)


check("T15-10", "Fig. 1: 14 nodes, 14 edges, and the graph is acyclic",
      len(NODES) == 14 and len(EDGES) == 14 and acyclic(NODES, EDGES))
check("T15-11", "Fig. 1: every edge points forward in publication order",
      all(a < b for a, b in EDGES))
check("T15-12", "Fig. 1: double-bordered nodes = multi-pass papers in the
    figure",
      sorted(set(NODES) & set(OM)) == [44, 45, 48, 57, 59])
check("T15-13", "Fig. 1 covers Gamma_3, Gamma_4, Gamma_5 exactly",
      NODES == sorted(CLUSTERS[3] + CLUSTERS[4] + CLUSTERS[5]))
# Table 4 rows: (label, chemical assertion?, table as hypothesis/input?)
ROWS = [("twenty classes", False, False), ("bijection, ceiling", False, False)
    ,
        ("even fibres, std excluded", False, False), ("parity bounded", False,
            False),
        ("transverse", False, False), ("coarsening", False, True),
        ("seed-propagate-refine", False, True), ("conditional refinement",
            False, True),
        ("reachability", True, False), ("chambers", True, True),
        ("forced placement", True, True), ("sulfur pair", True, False),
        ("junction", True, False)]


```

```
def sigma(chem, table):
    return 3 if chem else (2 if table else 1)


check("T15-14", "Def. 10: sigma is a function; strata sizes 5, 3, 5 over 13 rows",
      [sum(sigma(c, t) == k for _, c, t in ROWS) for k in (1, 2, 3)] == [5, 3, 5])
check("T15-15", "without precedence two rows (chambers, placement) would be doubly placed",
      sum(1 for _, c, t in ROWS if c and t) == 2)
check("T15-16", "Prop. 7: no stratum-1 row takes a table as hypothesis or input",
      all(not t for _, c, t in ROWS if sigma(c, t) == 1))

# ====================================================================
#  T16  Transcriptions from Number 2 and the scope of this harness
# ====================================================================
block("T16", "Transcriptions; scope and obligations")
TR = [("T16-01", 43, "twelve new deferred targets"),
      ("T16-02", 43, "Fourteen non-implications"),
      ("T16-03", 43, "five corrections to the prior corpus"),
      ("T16-04", 46, "four new targets of which two are load-bearing"),
      ("T16-05", 46, "Twelve non-implications"),
      ("T16-06", 46, "three corrections to the prior paper"),
      ("T16-07", 47, "one hundred and twenty-eight"),
      ("T16-08", 47, "0.17%"),
      ("T16-09", 47, "seventy per cent"),
      ("T16-10", 50, "eight hundred and forty seed maps"),
      ("T16-11", 55, "twenty-seven"),
      ("T16-12", 57, "sixteen of sixteen blocks"),
      ("T16-13", 58, "1.02 Å"),
      ("T16-14", 41, "63")]
for ident, art, phrase in TR:
    check(ident, f"Art. {art}: '{phrase}' appears as printed", phrase in
          REGISTER[art],
          "TRANSCRIPTION")
check("T16-15", "Art. 45's 'seventy-two' at N = 4 agrees with T3-06 computed here",
      "seventy-two" in REGISTER[45] and len(orbits(4)) == 72)
check("T16-16", "Rem. 2: Number 2 adds 12 + 4 = 16 deferred targets",
      12 + 4 == 16)
check("T16-17", "Prop. 4 holds by construction of Def. 4; it is not tested here",
      True, "RECORDED")
check("T16-18", "Prop. 8 and Rem. 9 are model-theoretic; no arithmetic settles them",
      True, "RECORDED")
check("T16-19a", f"rounding device is not vacuous: all {len(ROUND_LOG)} certified figures fail at r +/- 1",
      len(ROUND_LOG) > 0 and all(not _bound(n, d, r + 1, s) and not _bound(n, d, r - 1, s)
                                 for (n, d, r, s) in ROUND_LOG))
check("T16-19", "Cat, Basis, T, cluster and stratum are single-coder judgments; the "
      "harness checks their counts, not their correctness", True, "RECORDED")

# ====================================================================
#  Report
```

```
# ============================================================================


def main():
    blocks = sorted(BLOCK_TITLES, key=lambda t: int(t[1:]))
    total = len(RESULTS)
    passed = sum(r[4] for r in RESULTS)
    print("verify_A89.py -- Annals of the Chrysene Formalism, Article 89")
    print("A Quantitative and Categorical Survey of Volume 1, Number 2")
    print("-" * 72)
    for tag in blocks:
        rs = [r for r in RESULTS if r[0] == tag]
        ok = sum(r[4] for r in rs)
        print(f"{tag:<4} {BLOCK_TITLES[tag][:52]:<52} {ok:>3}/{len(rs):<3} "
              f"{'PASS' if ok == len(rs) else 'FAIL'}")
        if VERBOSE:
            for (_, ident, kind, desc, ok1) in rs:
                mark = "PASS" if ok1 else "FAIL"
                print(f"      [{mark}] {ident:<7} {kind:<13} {desc}")
    print("-" * 72)
    kinds = {}
    for r in RESULTS:
        kinds[r[2]] = kinds.get(r[2], 0) + 1
    print("kinds: " + ", ".join(f"{k} {v}" for k, v in sorted(kinds.items())))
    print(f"TOTAL: {passed}/{total} checks passed, {total - passed} failed")
    return 0 if passed == total else 1


if __name__ == "__main__":
    sys.exit(main())
```

Console Transcript: Default Run

$ python3 verify_A89.py

```
verify_A86.py -- Annals of the Chrysene Formalism, Article 86
A Quantitative and Categorical Survey of Volume 1, Number 2
------------------------------------------------------------------------
T1   The data and its provenance                          8/8   PASS
T2   Def. 1: the frame group is a Klein four-group        5/5   PASS
T3   Thm. 1: the registry-class count                     8/8   PASS
T4   Thm. 1: Burnside and the stabilizers                 6/6   PASS
T5   Cor. 1: the capacity ratio                           5/5   PASS
T6   Orbit counts of the subgroups of Fr                  4/4   PASS
T7   Prop. 2: even fibres                                 3/3   PASS
T8   Cor. 2: the standard and mitochondrial codes        10/10  PASS
T9   Normal form; where the two codes differ             6/6   PASS
T10  Prop. 3: transversality (convention)                 4/4   PASS
T11  Nineteen of twenty split classes (convention)        8/8   PASS
T12  Tables 2-3, Rem. 5: distributions                   16/16  PASS
T13  Prop. 5: conservativity on Number 1                  9/9   PASS
T14  Section 5: page figures                              9/9   PASS
T15  Clusters, Prop. 6, Def. 6, Fig. 1, Def. 10          16/16  PASS
T16  Transcriptions; scope and obligations               20/20  PASS
------------------------------------------------------------------------
kinds: CONVENTION 9, EXACT 95, RECORDED 6, TRANSCRIPTION 27
TOTAL: 137/137 checks passed, 0 failed
```

Console Transcript: Verbose Run (-v)

$ python3 verify_A89.py -v

```
verify_A86.py -- Annals of the Chrysene Formalism, Article 86
A Quantitative and Categorical Survey of Volume 1, Number 2
----------------------------------------------------------------------
T1   The data and its provenance                                  8/8   PASS
     [PASS] T1-01   TRANSCRIPTION 59 articles: 38 in Number 1, 21 in Number 2
     [PASS] T1-02   TRANSCRIPTION pages are contiguous: No. 1 spans 1--475, No
        . 2 spans 476--1162
     [PASS] T1-03   TRANSCRIPTION every category label is one of the eight of
        Def. 8
     [PASS] T1-04   TRANSCRIPTION every basis label is one of Mol, Math, Meth;
         Meth only in No. 2
     [PASS] T1-05   TRANSCRIPTION 28 techniques in T' = 20 original + 8 new (
        Def. 8)
     [PASS] T1-06   TRANSCRIPTION no No. 1 article uses a technique of T_new (
        T'|_I = T)
     [PASS] T1-07   TRANSCRIPTION every article has at least one technique
     [PASS] T1-08   TRANSCRIPTION No. 2 review structures: 6 multi-pass, 15
        single-round
T2   Def. 1: the frame group is a Klein four-group                5/5   PASS
     [PASS] T2-01   EXACT          phi is an involution on Omega_N, N = 1..8
     [PASS] T2-02   EXACT          rho is an involution on Omega_N, N = 1..8
     [PASS] T2-03   EXACT          phi and rho commute (both give 1 - x_{2N+1-i
        }), N = 1..8
     [PASS] T2-04   EXACT          phi != rho, and phi*rho is not the identity,
         N = 1..8
     [PASS] T2-05   EXACT          phi(x)_i = 1 - x_i: the bit-level reading of
         Eq. (1)
T3   Thm. 1: the registry-class count                             8/8   PASS
     [PASS] T3-01   EXACT          |Omega_N / Fr| = 2^(2N-2) + 2^(N-1) for N =
        1..8
     [PASS] T3-02   EXACT          every orbit has cardinality 2 or 4, none 1,
        N = 1..8
     [PASS] T3-03   EXACT          exactly 2^N orbits of cardinality 2, N =
        1..8
     [PASS] T3-04   EXACT          exactly 2^(2N-2) - 2^(N-1) orbits of
        cardinality 4, N = 1..8
     [PASS] T3-05   EXACT          N = 3: twenty classes, eight of size 2 and
        twelve of size 4
     [PASS] T3-06   EXACT          N = 4: seventy-two classes
     [PASS] T3-07   EXACT          orbit-stabilizer: the (class, element) pairs
         number 64 at N = 3
     [PASS] T3-08   EXACT          the classes partition Omega_3 (disjoint and
        covering)
T4   Thm. 1: Burnside and the stabilizers                         6/6   PASS
     [PASS] T4-01   EXACT          |Fix(phi)| = 0 for N = 1..8
     [PASS] T4-02   EXACT          |Fix(rho)| = 2^N (palindromes) for N = 1..8
     [PASS] T4-03   EXACT          |Fix(phi rho)| = 2^N (anti-palindromes) for
        N = 1..8
     [PASS] T4-04   EXACT          Burnside: (2^(2N) + 0 + 2^N + 2^N)/4 = 2^(2N
        -2) + 2^(N-1)
     [PASS] T4-05   EXACT          Fix(rho) and Fix(phi rho) are disjoint (
        their meet is in Fix(phi))
     [PASS] T4-06   EXACT          every stabilizer has order 1 or 2
T5   Cor. 1: the capacity ratio                                   5/5   PASS
```

```
      [PASS] T5-01   EXACT          |X_N| / 4^N = 1/4 + 2^-(N+1) exactly, N =
          1..60
      [PASS] T5-02   EXACT          the ratio is < 1 for every N = 1..60
38    [PASS] T5-03   EXACT          the ratio is strictly decreasing in N, so no
           N is singled out
      [PASS] T5-04   EXACT          class counts equal the multiset counts C(N
          +3,3) only at N = 3 (N = 1..60)
      [PASS] T5-05   EXACT          at N = 4: 72 classes against 35 multisets (
          Art. 45's refutation)
T6    Orbit counts of the subgroups of Fr                      4/4   PASS
      [PASS] T6-01   EXACT          orbit counts: 1 -> 64, <phi> -> 32, <rho> ->
           36, <phi rho> -> 36, Fr -> 20
43    [PASS] T6-02   EXACT          the count 32 cited from Art. 48 is attained
          by exactly one subgroup, <phi>
      [PASS] T6-03   TRANSCRIPTION the residual count 36 of Art. 52 is the
          orbit count of <rho>
      [PASS] T6-04   RECORDED       reading: the 32 of the article is the <phi>
          count (an interpretation, not Art. 48's proof)
T7    Prop. 2: even fibres                                     3/3   PASS
      [PASS] T7-01   EXACT          all 2^20 = 1,048,576 unions of classes at N
          = 3 have even cardinality
48    [PASS] T7-02   EXACT          every orbit has even cardinality for N =
          1..8 (the general case)
      [PASS] T7-03   EXACT          phi maps every class to itself and has no
          fixed point (the involution argument)
T8    Cor. 2: the standard and mitochondrial codes            10/10  PASS
      [PASS] T8-01   TRANSCRIPTION both tables have 64 codons over 20 amino
          acids and a stop symbol
      [PASS] T8-02   EXACT          standard: Met = {AUG} and Trp = {UGG} are
          singletons
53    [PASS] T8-03   EXACT          standard: the odd fibres are exactly Met (1)
          , Trp (1), Ile (3), stop (3)
      [PASS] T8-04   EXACT          hence c_std is class-functional for no beta
          (Cor. 2(1))
      [PASS] T8-05   EXACT          mitochondrial: every fibre has even
          cardinality (Cor. 2(2))
      [PASS] T8-06   EXACT          mitochondrial: AUA -> Met and UGA -> Trp, so
           both fibres have size 2
      [PASS] T8-07   EXACT          mitochondrial: Leu and Ser have 6; Ala Gly
          Pro Thr Val Arg and stop have 4
58    [PASS] T8-08   EXACT          mitochondrial: the ten listed amino acids
          have fibres of size 2
      [PASS] T8-09   EXACT          mitochondrial fibre sizes account for all 64
           codons
      [PASS] T8-10   EXACT          stop in the mitochondrial table is {UAA, UAG
          , AGA, AGG}
T9    Normal form; where the two codes differ                  6/6   PASS
      [PASS] T9-01   EXACT          the codes differ at exactly four codons: AGA
          , AGG, AUA, UGA
63    [PASS] T9-02   EXACT          the four codons lie in three purine half-
          blocks (AG, AU, UG)
      [PASS] T9-03   EXACT          the mitochondrial code is in normal form at
          all 16 blocks
      [PASS] T9-04   EXACT          the standard code leaves normal form at
          exactly two blocks, AU and UG
      [PASS] T9-05   EXACT          at the AG purine half both codes are uniform
           (stop/stop; Arg/Arg)
      [PASS] T9-06   EXACT          standard = mitochondrial with two further
          divisions (at AUA and UGA)
```

```
T10  Prop. 3: transversality (convention)                    4/4   PASS
     [PASS] T10-01  CONVENTION    class and block meet in at most one word,
        for all 24 letter labelings
     [PASS] T10-02  CONVENTION    reason: rho or phi*rho keeps x's block only
        when it fixes x
     [PASS] T10-03  EXACT         the 20 classes and 16 blocks give 64 = sum
        of class sizes = 16 x 4
     [PASS] T10-04  RECORDED      that Art. 50 uses the consecutive-pair
        convention is assumed, not shown
T11  Nineteen of twenty split classes (convention)           8/8   PASS
     [PASS] T11-01  CONVENTION    exactly 8 of the 24 labelings make
        complementation Watson--Crick
     [PASS] T11-02  CONVENTION    under each of those 8, both codes split
        exactly 19 of the 20 classes
     [PASS] T11-03  CONVENTION    under each of the other 16, both codes split
         all 20 classes
     [PASS] T11-04  CONVENTION    the unsplit class is {AGU, UCA}, painted Ser
         in both codes
     [PASS] T11-05  TRANSCRIPTION agreement with Art. 53's register abstract
        (19 of 20, {AGU, UCA}, Ser)
     [PASS] T11-06  CONVENTION    so c_mt is not class-functional under those
        identifications (text after Cor. 2)
     [PASS] T11-08  CONVENTION    Cor. 2(1) directly: c_std splits at least
        one class under all 24 labelings
     [PASS] T11-07  CONVENTION    the count 8 of 24 agrees with Art. 45's '
        eight of the twenty-four bijections'
T12  Tables 2-3, Rem. 5: distributions                      16/16  PASS
     [PASS] T12-01  EXACT         Table 2, No. 1 column: 13, 6, 6, 5, 3, 3, 2,
         0
     [PASS] T12-02  EXACT         Table 2, No. 2 column: 12, 6, 0, 0, 0, 0, 1,
         2
     [PASS] T12-03  EXACT         Table 2, Vol. 1 column and totals 38, 21, 59
     [PASS] T12-04  EXACT         Rem. 5: 12 > 21/2, and shares 34.2 -> 57.1
        %, 15.8 -> 28.6 %
     [PASS] T12-05  EXACT         Rem. 5: four of the seven No. 1 categories
        have no No. 2 paper
     [PASS] T12-06  EXACT         basis, No. 2: Math 11 (52.4 %), Mol 8 (38.1
        %), Meth 2 (9.5 %)
     [PASS] T12-07  EXACT         basis, No. 1: Math 18, Mol 20, Meth 0
     [PASS] T12-08  EXACT         tiers: 6 multi-pass (28.6 %), 15 single-
        round (71.4 %)
     [PASS] T12-09  EXACT         models: Claude both sides 15 (71.4 %), Grok
        3, ChatGPT 2, mixed 1
     [PASS] T12-10  EXACT         20 of 21 papers used one model family on
        both sides (Sec. 9.5)
     [PASS] T12-11  EXACT         Table 3: the eight rows, No. 2 counts and No
        . 1 counts
     [PASS] T12-12  EXACT         Table 3: the daggered rows are exactly those
         in T_new
     [PASS] T12-13  EXACT         sixteen of the twenty-eight techniques occur
         in No. 2
     [PASS] T12-14  EXACT         thermodynamics (11 in No. 1) and algebraic
        topology (8) do not occur in No. 2
     [PASS] T12-15  EXACT         techniques with 2 occurrences in No. 2: H-
        inf, op. alg., spectral, stat., corpus
     [PASS] T12-16  EXACT         techniques with 1 occurrence in No. 2:
        finite fields, category theory, QC
T13  Prop. 5: conservativity on Number 1                     9/9   PASS
     [PASS] T13-01  TRANSCRIPTION Art. 39 Table 1 (categories) reproduced from
```

```
          the extended ledger
      [PASS] T13-02  EXACT         Art. 39 Table 1 shares: 34.2, 15.8, 15.8,
         13.2, 7.9, 7.9, 5.3 %
      [PASS] T13-03  TRANSCRIPTION Art. 39 Table 2 (nine most frequent
         techniques) reproduced
103   [PASS] T13-04  EXACT         Art. 39 Table 2 shares certified by rounding
          bound
      [PASS] T13-05  EXACT         Art. 39's omitted eleven: six at 3--4, two
         at 2, two at 1 (and Discrete Morse at 2)
      [PASS] T13-06  EXACT         Art. 39 basis split 20 Mol / 18 Math (52.6 /
          47.4 %)
      [PASS] T13-07  EXACT         Art. 39 review figures: 34 single (89.5 %),
         4 two-round (10.5 %)
      [PASS] T13-08  EXACT         Art. 39 model figures: 25 Gemini-only, 2
         Grok-only, 11 hand-off
108   [PASS] T13-09  EXACT         T_new is unassessed on No. 1: its No. 1
         counts are 0 by construction
T14   Section 5: page figures                                  9/9   PASS
      [PASS] T14-01  EXACT         No. 2 spans 687 pages; No. 1 spans 475
      [PASS] T14-02  EXACT         mean 687/21 rounds to 32.7; 475/38 = 12.5
         exactly
      [PASS] T14-03  EXACT         ratio of means rounds to 2.6
113   [PASS] T14-04  EXACT         medians: 29 pages (No. 2) and 11 pages (No.
         1)
      [PASS] T14-05  EXACT         ranges: 9--80 (No. 2) and 5--23 (No. 1)
      [PASS] T14-06  EXACT         the three longest: Art. 54 (80), Art. 40
         (72), Art. 49 (64)
      [PASS] T14-07  EXACT         tier pages: multi-pass 214, single-round 473
      [PASS] T14-08  EXACT         the multi-pass set is {40, 44, 45, 48, 57,
         59}
118   [PASS] T14-09  EXACT         medians ratio 29/11 also rounds to 2.6 (the
         skew does not drive the ratio)
T15   Clusters, Prop. 6, Def. 6, Fig. 1, Def. 10            16/16  PASS
      [PASS] T15-01  EXACT         the seven clusters partition J = {39..59} (
         Def. 9)
      [PASS] T15-02  EXACT         cluster sizes 2, 2, 2, 8, 4, 1, 2
      [PASS] T15-03  EXACT         Gamma_4 and Gamma_5 lie in Cosmological &
         Biological Emergence
123   [PASS] T15-04  EXACT         Prop. 6 witnesses: Basis(47) = Math, Basis
         (57) = Mol, both in Gamma_4
      [PASS] T15-05  EXACT         Prop. 6 survives either classification of
         Art. 48
      [PASS] T15-06  EXACT         C&B basis split: 7 Mol / 5 Math in No. 2; 12
          / 1 in No. 1
      [PASS] T15-07  EXACT         Def. 6: clauses (a) U (b) U (c) = the
         recorded multi-pass set
      [PASS] T15-08  EXACT         clause (c) does not select Art. 50 or 55
128   [PASS] T15-09  RECORDED      the rule is an ex post reconstruction; its
         test is No. 3
      [PASS] T15-10  EXACT         Fig. 1: 14 nodes, 14 edges, and the graph is
          acyclic
      [PASS] T15-11  EXACT         Fig. 1: every edge points forward in
         publication order
      [PASS] T15-12  EXACT         Fig. 1: double-bordered nodes = multi-pass
         papers in the figure
      [PASS] T15-13  EXACT         Fig. 1 covers Gamma_3, Gamma_4, Gamma_5
         exactly
133   [PASS] T15-14  EXACT         Def. 10: sigma is a function; strata sizes
         5, 3, 5 over 13 rows
```

```
      [PASS] T15-15  EXACT         without precedence two rows (chambers,
         placement) would be doubly placed
      [PASS] T15-16  EXACT         Prop. 7: no stratum-1 row takes a table as
         hypothesis or input
T16   Transcriptions; scope and obligations                    20/20  PASS
      [PASS] T16-01  TRANSCRIPTION Art. 43: 'twelve new deferred targets'
         appears as printed
      [PASS] T16-02  TRANSCRIPTION Art. 43: 'Fourteen non-implications' appears
          as printed
      [PASS] T16-03  TRANSCRIPTION Art. 43: 'five corrections to the prior
         corpus' appears as printed
      [PASS] T16-04  TRANSCRIPTION Art. 46: 'four new targets of which two are
         load-bearing' appears as printed
      [PASS] T16-05  TRANSCRIPTION Art. 46: 'Twelve non-implications' appears
         as printed
      [PASS] T16-06  TRANSCRIPTION Art. 46: 'three corrections to the prior
         paper' appears as printed
      [PASS] T16-07  TRANSCRIPTION Art. 47: 'one hundred and twenty-eight'
         appears as printed
      [PASS] T16-08  TRANSCRIPTION Art. 47: '0.17%' appears as printed
      [PASS] T16-09  TRANSCRIPTION Art. 47: 'seventy per cent' appears as
         printed
      [PASS] T16-10  TRANSCRIPTION Art. 50: 'eight hundred and forty seed maps'
          appears as printed
      [PASS] T16-11  TRANSCRIPTION Art. 55: 'twenty-seven' appears as printed
      [PASS] T16-12  TRANSCRIPTION Art. 57: 'sixteen of sixteen blocks' appears
          as printed
      [PASS] T16-13  TRANSCRIPTION Art. 58: '1.02 Å' appears as printed
      [PASS] T16-14  TRANSCRIPTION Art. 41: '63' appears as printed
      [PASS] T16-15  EXACT         Art. 45's 'seventy-two' at N = 4 agrees with
          T3-06 computed here
      [PASS] T16-16  EXACT         Rem. 2: Number 2 adds 12 + 4 = 16 deferred
         targets
      [PASS] T16-17  RECORDED      Prop. 4 holds by construction of Def. 4; it
         is not tested here
      [PASS] T16-18  RECORDED      Prop. 8 and Rem. 9 are model-theoretic; no
         arithmetic settles them
      [PASS] T16-19a EXACT         rounding device is not vacuous: all 35
         certified figures fail at r +/- 1
      [PASS] T16-19  RECORDED      Cat, Basis, T, cluster and stratum are
         single-coder judgments; the harness checks their counts, not their
         correctness
-----------------------------------------------------------------------------
kinds: CONVENTION 9, EXACT 95, RECORDED 6, TRANSCRIPTION 27
TOTAL: 137/137 checks passed, 0 failed
```

Lean 4 Proof Summary: `acf-89-proof.lean`

Lean Kernel Certification

The file `acf-89-proof.lean` contains 85 theorems in the namespace `ChryseneFormalism.Article89`, checked by the Lean 4.12.0 kernel. Every theorem is discharged by `decide`. There is no Mathlib, no `sorry` and no `native_decide`.
An `#print axioms` audit over all 85 theorems returns *"does not depend on any axioms"* for **every one of them**. The file assumes nothing beyond the type theory itself. The full dossier for the proof is published separately as `acf-89-proof.tex`.

A complete run takes about a minute. The time is spread across the exhaustive searches over $N = 1, \ldots, 5$ and the twenty-four-labeling split-class searches; no single theorem takes more than about eight seconds.

Where the harness and the proof file divide the work

The two artifacts are not duplicates, and neither subsumes the other.
The Lean file certifies Theorem 1 and its Burnside steps for $N = 1, \ldots, 5$. It certifies Corollary 1 for $N \leq 60$, the code fibres, normal form, transversality and the split classes, and every corpus count and printed decimal of Section 5. Each statement is a kernel-evaluated equation or inequality over naturals. Where it applies, it applies absolutely.
The harness does three things the Lean file does not.

- It reaches further: Theorem 1 to $N = 8$, and Proposition 2 by exhausting all 2^{20} unions of classes. The Lean file certifies only that proposition's premises — every class even, φ fixed-point-free — and leaves the step from those to every union to the harness.
- It checks the transcriptions against the register excerpts.
- It tests its own devices for vacuity.

Cross-check against the published manuscript

At preparation time, each of thirty-four figures certified here was searched for in the source of Article 89, as printed. These include the decimals of Section 5, the counts $20 = 8 + 12$ and 72, the capacity ratio, the fibre lists of Corollary 2, the 7/5 and 12/1 splits, the three longest papers, and the check and theorem counts quoted in its Appendix B. **All thirty-four were found**, so the figures the harness certifies are the figures the article prints, and none of them disagrees with the harness.