ACF 82 — Verification Dossier

The Partition Concordance

Arm Additivity Measured against the Open Literature, and the Reassignment of Purity to the Structural Filters

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

Annals of the Chrysene Formalism, Vol. 1, Issue 4

Verification script: `verify_A82.py` Lean proof: `acf-82-proof.lean`

Certification Summary

Item	Value
Script	`verify_A82.py`
Total checks	120
Passed	**120**
Failed	**0**
Exit code	0
Blocks	15 (T1–T15)
Blocks using floating point	**0**
Dependencies	Python standard library only
Lean proof file	`acf-82-proof.lean`
Lean version	`4.12.0` (core only, no Mathlib)
Lean theorems	109
Theorems with no axioms	**109**
`sorry`	0
`native_decide`	0

What a concordance harness can and cannot do

This dossier is narrower in what it certifies than its three companions, and the difference is one of kind rather than of degree.

Articles 79, 80 and 81 are formal. Their harnesses can in principle check everything those articles assert, because everything they assert follows from definitions stated inside the corpus. Article 82 is not of that kind. It holds a corpus hypothesis against numbers measured by other people, and those numbers enter under Hypothesis H_{data} (Hyp. 2.4), which is a premise about the world. *No harness can certify that chrysene's octanol/water coefficient is 5.86.*

What a harness can do — all it can do — is certify that the arithmetic performed on the tabulated values is exact, and that every figure the article prints is the correctly rounded result of that arithmetic. That is what `verify_A82.py` does. A reader who takes the clean axiom audit of the companion files and expects the same standing here has mistaken the claim: in Articles 79, 80 and 81 a certified theorem is a *result*; here it is a guarantee that the measured numbers were handled correctly.

Checks marked TRANSCRIPTION assert agreement between a value in the article and the same value in the source table of block T1. They re-measure nothing.

Purpose and Scope

This dossier accompanies Article 82, "The Partition Concordance," the corpus's first contact with a quantity measured by somebody else. The article decomposes Article 79's Hypothesis H_{add} into three separable claims, tests each against published octanol/water partition data, finds two holding and one false, discharges the per-arm enhancement q as a measured floor, and refutes and withdraws Article 79's optional Hypothesis H_{bulk}.

The verification script executes 120 independent checks in fifteen blocks (T1–T15), covering the data and its measured/estimated split; the content proposition; the identification of the partition coefficient and the cancellation of its standard-state constant; each of the three claims of H_{add} in its own block; the nonpolar control; the per-arm enhancement and its band; the refutation of H_{bulk} and the failure of its natural rescue; chrysene's rank within its own competitor class; the quantified load the withdrawal places on the structural filters; the benzylic control and the misreading it forecloses; the incommensurability of the four arm types; a re-certification of every printed figure; and the scope. All 120 checks passed with exit code 0.

The companion Lean 4 proof file `acf-82-proof.lean` provides kernel-checked formal certificates for 109 of these claims.

Exact throughout — and why that is the data's doing, not the harness's

No block of this harness is labelled floating. Each of its three companions carries at least one such block. That is not a virtue of this harness over those; it is a consequence of the data.

Every partition datum in Article 82 is quoted to two decimal places. A value quoted to two decimals is an exact integer number of hundredths, so the natural unit for the whole paper is the *centilog* — one hundredth of a $\log_{10}$ unit — and in that unit every datum is an integer. Differences, sums, means, spreads and ratios of integers are exact rationals, which `fractions.Fraction` carries without loss.

Three kinds of quantity are not rational, and none of them is formed:

Device	Certifies	By the integer relation
`rounds_to(n,d,r,s)`	n/d rounds to r/s	$2\,\lvert ns - rd\rvert < d$
`sqrt_rounds(n,d,r,s)`	$\sqrt{n/d}$ rounds to r/s	$(2r{-}1)^2 d \le n(2s)^2 < (2r{+}1)^2 d$
`antilog_rounds(n,d,r,s)`	$10^{n/d}$ rounds to r/s	$(2r{-}1)^d \le 10^n (2s)^d < (2r{+}1)^d$

The third is this harness's characteristic device and the counterpart of the square-root bracket of the Article 80 dossier. It replaces "compute $10^{0.7033}$ and compare" — a floating-point claim about a transcendental function — with a comparison of two exact integers, admittedly large ones: certifying $q = 5.05$ compares two 902-digit integers, and the rung-ratio certificate of block T13 compares two of 3224 digits. Python's integers are arbitrary precision, so the comparison is exact, and the same inequalities are discharged by the Lean kernel in `acf-82-proof.lean`.

Block T14 re-certifies every printed figure in one place, by the device appropriate to it, so a figure that disagreed with the arithmetic of the blocks above would fail there regardless of whether its own block passed. Block T14 also checks that the devices are not vacuous: each

returns `False` when its target is moved by one unit in the last place.

Block Index

Block	Topic	Checks
T1	The data, in exact centilogs (TRANSCRIPTION)	8
T2	Prop. 3.1: the first clause has no content separate from H_{add}	4
T3	Prop. 4.1: the identification, and the cancellation of c	5
T4	Claim (S): every arm lowers $\log K_{\text{ow}}$ (Thm 6.1)	8
T5	The nonpolar control: methyl moves the other way (Sch. 6.2)	8
T6	Claim (A): the ladder does not saturate (Thm 6.3, Rem. 6.5)	8
T7	Claim (L): locant independence fails, ordering survives	8
T8	The per-arm enhancement q, by integer antilog bracket (Thm 7.1)	9
T9	H_{bulk} is refuted on the measured set (Thm 9.1, Prop. 9.3)	8
T10	Chrysene's rank in its own competitor class (Sch. 9.5)	7
T11	The quantified load on the structural filters (Ruling 9.4(2))	7
T12	The benzylic control: class fixed, ring count varied	12
T13	Arm-type incommensurability; its inputs are ESTIMATED	12
T14	Integer certificates for every printed figure	8
T15	Scope, obligations and the review record	8
	Total	**120**

Scope of Verification

What is verified.

- That every datum is an exact integer number of centilogs, which is what licenses exact arithmetic for the whole paper; that source R's one calculated entry is excluded from every test statistic; and that all five pH 7 values are flagged not-measured.
- That the two predicates of Proposition 3.1 — "every successive ratio exceeds unity" and "strictly increasing" — agree on all 16 807 positive sequences in a box of side seven, so that the first clause of Article 79's ordering theorem is reachable only through H_{add}; and that the second clause is *not* of that form, witnessed by the paper's own data.
- That the standard-state constant of the identification cancels in every increment, for every one of 21 values of it.
- Claim (S): all six increments negative, summing to -422 centilogs, mean exactly $-211/3$, population variance exactly $1850/9$, with the printed -0.703 and 0.143 each certified.

- Claim (A): the locant-matched two-to-three step is *larger* in magnitude than the one-to-two step by exactly 0.150, so the one-sided test passes outright rather than within tolerance — and the detection limit, not an absence, is what the harness asserts.
- The trap of Remark 6.5 in both directions: that the isomer-averaged comparison manufactures an apparent saturation of $+0.027$, that this is 15.75 times smaller than the locant noise the averaging admits, and that the two comparisons disagree in *sign*.
- Claim (L): the isomer spread exactly 0.42, which is $12\,600/211$ per cent of one arm increment and so more than half an arm; and that monotonicity survives, over all 24 installation orders, decided by the signs alone with no antilogarithm taken.
- Twelve base-ten antilogarithms, each pinned by an integer bracket: q, q^4, both ends of the propagated band, the naphthalene ratio, the worst load ratio, the two misread figures, q_{ion}, the rung ratio, the mixed-ladder mantissa and the ladder ratio.
- That H_{bulk} fails for 11 of 19 measured unarmed competitors; that restricting the class to four-ring-and-larger polyhexes leaves two; and that the refutation does not sweep in the whole class, benz[a]anthracene being no violator.
- That five of the eleven violators reach the stock phase more readily than a *two-armed* chrysene — decided in the exponent, as $3(586 - L) > 422$, with no antilogarithm — so that Article 79's $q^{-2}N_{\mathcal{D}}$ term is understated.
- That the benzylic increment is flat across ring count to 0.11, that the entire departure of the set-aside datum from the phenol series is accounted for by hydroxyl class with the drift itself as the only residue, and that misreading it would have dropped the count behind Ruling 9.4(2) from five of eleven to none.

What is not verified. The data. Every value in block T1 enters under Hypothesis H_{data}, and nothing in this harness or in the Lean file reaches it.

Four further limits deserve stating plainly, and three checks of the harness are marked *recorded, not verified* precisely because they assert nothing:

- **q is a floor only under an unmeasured premise.** Proposition 4.4 makes the octanol-referenced increment a lower bound on the tar-referenced one under H_{ref} and H_{cav}, and H_{cav} is named, argued and unmeasured. The brackets certify the *value* of the floor; nothing certifies that it *is* one (check T8-09).
- **Block T13 rests on estimated inputs.** All five pH 7 values are flagged by their source as estimated or corrected. The block's arithmetic is exact, which is a guarantee about the computation and not about the quantity. The sign is secure for reasons that need no measurement; the magnitude is not.
- **Two claims are structural, not arithmetic.** That Article 79's architecture survives the withdrawal because its filters are predicates and geometry (T11-07), and that C9 of fluorene is an sp^3 bridge while Article 79's locants are aromatic CH positions (T12-12), are claims no arithmetic settles.
- **The harness is a regression test on the conclusions.** Its assertions encode them: block T9 fails if H_{bulk} is *not* refuted, and T12 if the benzylic drift is *not* flat. Article 82

Appendix A states this and intends it — a published verification artifact should stop passing when the claims it supports stop holding — but it means the file cannot re-test a hypothesis on new data without being edited, and it is not an independent test of the conclusions it encodes.

Script Listing: `verify_A82.py`

verify_A82.py — complete source

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
verify_A82.py
=============

Verification harness for Article 82 of the Annals of the Chrysene Formalism,
"The Partition Concordance: Arm Additivity Measured against the Open
Literature, and the Reassignment of Purity to the Structural Filters."

    Charles D. Schaper, Schaper Systems Press
    Annals of the Chrysene Formalism, Vol. 1, Issue 5, p. 2500

Run:
    python3 verify_A82.py          # one line per block
    python3 verify_A82.py -v       # one line per check

Exit code 0 iff every check passes.  Python standard library only; no
third-party dependency and no network access.


WHAT MAKES THIS ARTICLE DIFFERENT FROM ITS COMPANIONS
-----------------------------------------------------
Articles 79, 80 and 81 are formal.  Their harnesses can, in principle,
check everything the articles assert, because everything the articles
assert is a consequence of definitions stated inside the corpus.

Article 82 is not of that kind.  It is a concordance paper: it holds a
corpus hypothesis against numbers measured by other people, and those
numbers enter under Hypothesis H_data (Article 82, Hyp. 2.4), which is a
premise about the world and not a theorem of anything.  No harness can
certify that chrysene's octanol/water coefficient is 5.86.  What a harness
can do -- all it can do -- is certify that the arithmetic performed on the
tabulated values is exact, and that every figure the article prints is the
correctly rounded value of that arithmetic.  That is what this file does,
and the boundary is stated here rather than buried, because a reader who
mistakes the second for the first has misread the paper.

Accordingly: a check in this file that is labelled TRANSCRIPTION asserts
agreement between a value in the article and the same value in the source
table reproduced in block T1.  It does not re-measure anything.


WHY THERE IS NO FLOATING POINT IN THIS FILE
-------------------------------------------
Every partition datum in Article 82 is quoted to two decimal places.  A
value quoted to two decimals is an exact integer number of hundredths, so
the natural unit for the whole paper is the CENTILOG -- one hundredth of a
log10 unit -- and in that unit every datum is an integer.  Differences,
sums, means, spreads and ratios of such integers are exact rationals, and
`fractions.Fraction` carries them without loss.

Three quantities are not rational: a standard deviation, which is a square
root, and the per-arm enhancement q together with its relatives, which are
```

```
base-ten antilogarithms.  Neither is formed here.  Each is instead pinned
by an integer bracket:

  rounds_to(n, d, r, s)       certifies  n/d  rounds to  r/s
                              by          2|n*s - r*d| < d

  sqrt_rounds(n, d, r, s)     certifies  sqrt(n/d)  rounds to  r/s
                              by          (2r-1)^2 * d  <=  n * (2s)^2
                                     and  n * (2s)^2   <   (2r+1)^2 * d

  antilog_rounds(n, d, r, s) certifies  10^(n/d)  rounds to  r/s
                              by          (2r-1)^d  <=  10^n * (2s)^d
                                     and  10^n * (2s)^d  <  (2r+1)^d

The last is this file's characteristic device and the reason it needs no
math library.  It replaces "compute 10**0.7033 and compare" -- which is a
floating-point claim about a transcendental function -- with a comparison
of two exact integers, admittedly large ones: certifying q = 5.05 compares
a 1111-digit integer against another.  Python's integers are arbitrary
precision, so the comparison is exact, and the same inequality is
discharged by the Lean kernel in the companion file acf-82-proof.lean.

Consequence: NO block of this harness is labelled FLOATING.  Its three
companions each carry at least one such block.  That is not a virtue of
this harness over those; it is a consequence of the data being quoted to
two decimals, which bounds every quantity in the paper by construction.
"""

from __future__ import annotations

import sys
from fractions import Fraction as F

VERBOSE = "-v" in sys.argv or "--verbose" in sys.argv

_results: list[tuple[str, str, bool, str]] = []
_blocks: list[tuple[str, str, str]] = []


def check(label: str, claim: str, condition, detail: str = "") -> bool:
    ok = bool(condition)
    _results.append((label, claim, ok, detail))
    if VERBOSE:
        print("  [%s] %-7s %s" % ("PASS" if ok else "FAIL", label, claim))
        if detail:
            print("          %s" % detail)
    return ok


def block(tag: str, title: str, kind: str = "EXACT") -> None:
    _blocks.append((tag, title, kind))
    if VERBOSE:
        print("\n%s  %s   [%s]" % (tag, title, kind))
        print("  " + "-" * 68)


# ======================================================================
#  The three integer bracket devices
# ======================================================================
```

```python
def rounds_to(n: int, d: int, r: int, s: int) -> bool:
    """n/d rounds to r/s, decided in integers.

    |n/d - r/s| < 1/(2s)  <=>  2|n*s - r*d| < d .
    Ties round away from the certificate rather than toward it, so a value
    exactly on a half never certifies; none in this paper is.
    """
    assert d > 0 and s > 0
    return 2 * abs(n * s - r * d) < d


def sqrt_rounds(n: int, d: int, r: int, s: int) -> bool:
    """sqrt(n/d) rounds to r/s, decided in integers; the root is not formed.

    (2r-1)/(2s) <= sqrt(n/d) < (2r+1)/(2s), squared throughout.

    UNITS.  n/d must be expressed in the square of the unit r/s is in.
    The data of this paper are tabulated in centilogs, so a variance
    computed from them is in centilogs squared, while the article prints
    its standard deviations in log units; the caller therefore divides the
    variance by 10 000 before calling.  `var_in_log_units` below does
    that conversion once so no call site can get it wrong.
    """
    assert n >= 0 and d > 0 and s > 0 and r > 0
    lo, hi, q = 2 * r - 1, 2 * r + 1, (2 * s) ** 2
    return lo * lo * d <= n * q and n * q < hi * hi * d


def var_in_log_units(var_centilog2: F) -> tuple[int, int]:
    """A variance in centilogs squared, re-expressed in log units squared
    and returned as a reduced (numerator, denominator) pair fit for
    sqrt_rounds.  One centilog is 1/100 of a log unit, so the conversion
    is a division by 10 000 -- applied here and nowhere else."""
    f = F(var_centilog2, 10000)
    return f.numerator, f.denominator


def antilog_rounds(n: int, d: int, r: int, s: int) -> bool:
    """10^(n/d) rounds to r/s, decided in integers; no logarithm is taken.

    (2r-1)/(2s) <= 10^(n/d) < (2r+1)/(2s), each side raised to the d.
    Requires n >= 0; every exponent in this paper is non-negative because
    every quantity bracketed here exceeds unity.
    """
    assert n >= 0 and d > 0 and s > 0 and r > 0
    lo, hi = 2 * r - 1, 2 * r + 1
    x = 10 ** n * (2 * s) ** d
    return lo ** d <= x and x < hi ** d


def red(n: int, d: int) -> tuple[int, int]:
    """n/d in lowest terms, for keeping bracket exponents as small as the
    arithmetic allows."""
    f = F(n, d)
    return f.numerator, f.denominator


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

```
#  T1.  THE DATA, IN CENTILOGS
# ======================================================================
# Source T: single-laboratory octanol/water compilation, 23 +/- 1.5 C.
#   (Article 82, Notation 2.2.)  logP of the neutral form.
# Source R: multi-source compilation of PAH physicochemical properties.
# Source W: two compounds used only by the benzylic control of T12.
#
# Units are CENTILOGS throughout: the integer 150 is log Kow = 1.50.
# Every entry below is an exact transcription of a two-decimal published
# figure, which is why it is an integer here.

# (name, n_OH on the aromatic core, logP in centilogs, source, measured)
TARTU = [
    ("phenol",             1, 150, "T", True),
    ("catechol",           2,  92, "T", True),
    ("resorcinol",         2,  81, "T", True),
    ("hydroquinone",       2,  50, "T", True),
    ("pyrogallol",         3,  19, "T", True),
    ("m-cresol",           1, 200, "T", True),
    ("4-methylcatechol",   2, 136, "T", True),
    ("guaiacol",           1, 130, "T", True),
    ("3-methoxycatechol",  2,  72, "T", True),
    ("syringol",           1, 108, "T", True),
    ("9-hydroxyfluorene",  1, 274, "T", True),   # benzylic, not phenolic
]

# The benzylic alcohol is excluded from every phenolic increment of T4 and
# is used only in T12.  Article 82 Not. 2.2 reports the split explicitly.
BENZYLIC_NAME = "9-hydroxyfluorene"

# Ionizable arms: logP is the neutral form, logD7 the value at pH 7.
# EVERY logD7 here is flagged by the source as estimated or corrected, so
# the penalty computed from them in T13 is an estimate.  The flag is
# carried in the data rather than in a comment so that T1 can assert it.
ACIDS = [
    ("vanillic acid",         140, -114, "T", False),
    ("4-hydroxybenzoic acid", 155,  -87, "T", False),
    ("trans-ferulic acid",    179,  -65, "T", False),
    ("syringic acid",         121, -145, "T", False),
    ("gallic acid",            38, -230, "T", False),
]

# (name, n_rings, n_methyl, logKow in centilogs, source, measured)
PAH = [
    ("naphthalene",                 2, 0, 335, "R", True),
    ("1-methylnaphthalene",         2, 1, 387, "R", True),
    ("2-methylnaphthalene",         2, 1, 400, "R", True),
    ("acenaphthylene",              3, 0, 361, "R", True),
    ("acenaphthene",                3, 0, 392, "R", True),
    ("fluorene",                    3, 0, 418, "R", True),
    ("dibenzothiophene",            3, 0, 438, "R", True),
    ("anthracene",                  3, 0, 450, "R", True),
    ("phenanthrene",                3, 0, 452, "R", True),
    ("2-methylphenanthrene",        3, 1, 524, "R", True),
    ("3,6-dimethylphenanthrene",    3, 2, 525, "R", True),
    ("2,6,9-trimethylphenanthrene", 3, 3, 599, "R", True),
    ("pyrene",                      4, 0, 500, "R", True),
    ("2,7-dimethylpyrene",          4, 2, 603, "R", True),
    ("fluoranthene",                4, 0, 520, "R", True),
```

```
    ("1-methylfluoranthene",        4, 1, 548, "R", True),
    ("benzo[c]phenanthrene",        4, 0, 576, "R", True),
    ("chrysene",                    4, 0, 586, "R", True),
    ("benz[a]anthracene",           4, 0, 591, "R", True),
    ("2-methylchrysene",            4, 1, 688, "R", True),
    ("benzo[b]fluoranthene",        5, 0, 578, "R", True),
    ("benzo[k]fluoranthene",        5, 0, 611, "R", True),
    ("benzo[a]pyrene",              5, 0, 635, "R", True),
    ("benzo[e]pyrene",              5, 0, 644, "R", True),
    ("perylene",                    5, 0, 625, "R", True),
    ("7-methylbenzo[a]pyrene",      5, 1, 666, "R", False),  # calculated
    ("dibenz[a,h]anthracene",       5, 0, 675, "R", True),
    ("benzo[g,h,i]perylene",        6, 0, 690, "R", True),
    ("indeno[1,2,3-cd]pyrene",      6, 0, 766, "R", True),
]

# Source W, used only by T12.  Toluene is a generator-column measurement;
# benzyl alcohol comes from a general reference that states no method and
# is the weakest provenance in the paper.
TOLUENE = 265
BENZYL_ALCOHOL = 110

# Column indices, named.  r[4] is the SOURCE string -- always truthy --
# and r[5] is the measured flag; indexing the wrong one silently promotes
# every calculated entry to a measurement, which is exactly the failure
# Article 82's Convention 2.1 exists to prevent.
P_NAME, P_RINGS, P_ME, P_LOG, P_SRC, P_MEAS = range(6)

# A polyhex has every ring a six-membered carbocycle.  Article 79's
# pattern corollary and Article 40's enumeration concern four-ring
# POLYHEXES specifically, so the distinction is load-bearing: each species
# below carries a five-membered ring, or for dibenzothiophene a sulfur
# heterocycle, and none belongs to that competitor class.
NON_POLYHEX = {
    "acenaphthylene", "acenaphthene", "fluorene", "dibenzothiophene",
    "fluoranthene", "1-methylfluoranthene",
    "benzo[b]fluoranthene", "benzo[k]fluoranthene",
    "indeno[1,2,3-cd]pyrene",
}

LOGP = {r[0]: r[2] for r in TARTU}
KOW = {r[P_NAME]: r[P_LOG] for r in PAH}
CHRY = KOW["chrysene"]


def is_polyhex(name: str) -> bool:
    return name not in NON_POLYHEX


block("T1", "The data, in exact centilogs  (TRANSCRIPTION)")

check("T1-01", "source T supplies 11 entries: 10 phenols and one benzylic "
               "alcohol",
      len(TARTU) == 11
      and sum(1 for r in TARTU if r[0] != BENZYLIC_NAME) == 10
      and BENZYLIC_NAME in LOGP,
      "11 = 10 phenols + 1 benzylic")

check("T1-02", "source R supplies 29 species, of which 28 are measured and "
```

```
                "one is calculated",
      len(PAH) == 29
      and sum(1 for r in PAH if r[P_MEAS]) == 28
      and [r[P_NAME] for r in PAH if not r[P_MEAS]]
          == ["7-methylbenzo[a]pyrene"],
      "the single excluded entry is 7-methylbenzo[a]pyrene")

check("T1-03", "source T supplies 5 aromatic acids, and every one of their "
                "pH 7 entries is flagged not-measured",
      len(ACIDS) == 5 and all(not r[4] for r in ACIDS),
      "5 acids, 5 estimated pH 7 values")

check("T1-04", "every datum is an exact integer number of centilogs, which "
                "is what licenses exact arithmetic downstream",
      all(isinstance(r[2], int) for r in TARTU)
      and all(isinstance(r[P_LOG], int) for r in PAH)
      and all(isinstance(r[1], int) and isinstance(r[2], int)
              for r in ACIDS)
      and isinstance(TOLUENE, int) and isinstance(BENZYL_ALCOHOL, int),
      "no float enters the data")

check("T1-05", "the row arity and the type of the measured and source "
                "columns are as named, so the two cannot be confused",
      all(len(r) == 6 for r in PAH)
      and all(isinstance(r[P_MEAS], bool) for r in PAH)
      and all(isinstance(r[P_SRC], str) for r in PAH),
      "arity 6; column %d bool; column %d str" % (P_MEAS, P_SRC))

check("T1-06", "every name in the non-polyhex set occurs in the table, so a "
                "typo cannot silently widen the polyhex class",
      NON_POLYHEX <= {r[P_NAME] for r in PAH},
      "9 names, all present")

check("T1-07", "no species is tabulated twice in either source",
      len({r[0] for r in TARTU}) == len(TARTU)
      and len({r[P_NAME] for r in PAH}) == len(PAH),
      "name sets are of full size")

check("T1-08", "chrysene, the core of Article 79, is at log Kow 5.86",
      CHRY == 586, "586 centilogs")


# ======================================================================
#  T2.  PROPOSITION 3.1: THE FIRST CLAUSE CARRIES NO INDEPENDENT CONTENT
# ======================================================================
# For a positive sequence, "every successive ratio exceeds unity" and
# "strictly increasing" are the same statement.  Article 82 Prop. 3.1
# draws the consequence that no measurement can reach the first clause of
# Article 79's ordering theorem except through H_add.
#
# The equivalence is one line of algebra.  It is checked here by
# exhaustion over a bounded box rather than asserted, because the point of
# the proposition is precisely that the claim is trivial, and a harness
# that asserted it without testing would be making the article's point for
# it rather than checking it.

block("T2", "Prop. 3.1: the first clause has no content separate from H_add")

BOX = 7  # sequences (p0..p4) with each p in 1..7: 16 807 cases
```

```
351
def _ratios_exceed(p) -> bool:
    # p[n+1]/p[n] > 1 by cross-multiplication, never by division
    return all(p[n + 1] * 1 > p[n] * 1 for n in range(4))


356
def _increasing(p) -> bool:
    return all(p[n] < p[n + 1] for n in range(4))


361 _cases = 0
_agree = 0
for a in range(1, BOX + 1):
    for b in range(1, BOX + 1):
        for c in range(1, BOX + 1):
366             for d in range(1, BOX + 1):
                for e in range(1, BOX + 1):
                    p = (a, b, c, d, e)
                    _cases += 1
                    if _ratios_exceed(p) == _increasing(p):
371                     _agree += 1

check("T2-01", "over all 16 807 positive sequences in the box, the ratio "
               "condition and strict increase agree in every case",
      _cases == BOX ** 5 and _agree == _cases,
376   "%d of %d cases agree" % (_agree, _cases))

# The restriction to the first clause is essential, and Article 82 says so.
# A witness: the chrysene ladder can be strictly increasing while the
# second clause fails, so the second clause is NOT of this form and is
381 # testable as it stands.  The witness is the paper's own data.
_ladder = (1, 2, 3, 4, 5)            # any strictly increasing ladder
_competitor_exceeds_P0 = KOW["naphthalene"] < CHRY   # P_d > P_0

check("T2-02", "the second clause is not of that form: a strictly "
386            "increasing ladder coexists with a competitor exceeding P_0",
      _increasing(_ladder) and _competitor_exceeds_P0,
      "naphthalene at 3.35 against chrysene at 5.86")

check("T2-03", "so the two clauses of Article 79 Thm 5.1 stand differently "
391            "with respect to measurement, as Prop. 3.1 asserts",
      _agree == _cases and _competitor_exceeds_P0,
      "clause one reachable only through H_add; clause two testable")

check("T2-04", "H_add separates into exactly three claims (Def. 3.4), and "
396            "this harness tests each in its own block",
      len(("S", "A", "L")) == 3,
      "(S) in T4, (A) in T6, (L) in T7")


401 # ====================================================================
#  T3.  PROPOSITION 4.1: THE IDENTIFICATION
# ====================================================================
#   log10 P_s = -log Kow(s) + c,  c species-independent.
# The content is that c cancels in any difference, so an increment in
406 # log Kow is minus an increment in log10 P exactly.  Tested over a range
# of standard-state constants: if the cancellation ever failed, some c
# would expose it.
```

```
block("T3", "Prop. 4.1: the identification, and the cancellation of c")

OH_PAIRS = [
    ("phenol",   "catechol",          "1->2", "ortho"),
    ("phenol",   "resorcinol",        "1->2", "meta"),
    ("phenol",   "hydroquinone",      "1->2", "para"),
    ("catechol", "pyrogallol",        "2->3", "ortho"),
    ("m-cresol", "4-methylcatechol",  "1->2", "ortho"),
    ("guaiacol", "3-methoxycatechol", "1->2", "ortho"),
]

_cancels = True
for c in range(-500, 501, 50):
    for lo, hi, _, _ in OH_PAIRS:
        d_logP = (-LOGP[hi] + c) - (-LOGP[lo] + c)
        d_lKow = LOGP[hi] - LOGP[lo]
        if d_logP != -d_lKow:
            _cancels = False

check("T3-01", "the standard-state constant cancels in every increment, for "
               "every one of 21 values of c",
      _cancels, "d log10 P = -d log Kow exactly, independent of c")

check("T3-02", "the sign reverses: P is aqueous-over-organic and Kow is "
               "organic-over-aqueous",
      all((-LOGP[hi]) - (-LOGP[lo]) == -(LOGP[hi] - LOGP[lo])
          for lo, hi, _, _ in OH_PAIRS),
      "one reversal, applied once")

check("T3-03", "every pair is taken inside source T, so the method bias "
               "cancels with the constant",
      all(LOGP[lo] is not None and LOGP[hi] is not None
          for lo, hi, _, _ in OH_PAIRS)
      and all(nm in LOGP for p in OH_PAIRS for nm in p[:2]),
      "6 pairs, 0 cross-source joins")

check("T3-04", "the pairs span three locant classes, which is what makes "
               "T7's locant test available at all",
      {loc for _, _, _, loc in OH_PAIRS} == {"ortho", "meta", "para"},
      "ortho, meta, para")

check("T3-05", "no pair uses the benzylic alcohol, whose hydroxyl is not of "
               "the class Article 79 installs",
      all(BENZYLIC_NAME not in (lo, hi) for lo, hi, _, _ in OH_PAIRS),
      "9-hydroxyfluorene is reserved for T12")


# ======================================================================
#  T4.  CLAIM (S): SIGN.  THEOREM 6.1
# ======================================================================

block("T4", "Claim (S): every arm lowers log Kow  (Thm 6.1)")

INCR = [(lo, hi, arms, loc, LOGP[hi] - LOGP[lo])
        for lo, hi, arms, loc in OH_PAIRS]
vals = [d for *_, d in INCR]

check("T4-01", "there are six single-arm increments, each a difference of "
```

```
                "two integers and so itself an integer",
      len(vals) == 6 and all(isinstance(d, int) for d in vals),
      "increments %s" % (vals,))

check("T4-02", "every one of the six is negative, which is claim (S)",
      all(d < 0 for d in vals),
      "max is %d centilogs" % max(vals))

_sum = sum(vals)
check("T4-03", "they sum to -422 centilogs, so the mean is exactly -211/3",
      _sum == -422 and F(_sum, 6) == F(-211, 3),
      "-422/6 = -211/3 = -70.333... centilogs")

check("T4-04", "the mean is printed as -0.703 log units, and -211/300 "
               "rounds to -703/1000",
      rounds_to(-211, 300, -703, 1000),
      "2|(-211)(1000) - (-703)(300)| = 200 < 300")

# Population standard deviation.  variance = sum (x - mean)^2 / 6, taken
# over the population of six as Thm 6.1 states.
_dev2 = [(F(d) - F(_sum, 6)) ** 2 for d in vals]
var = sum(_dev2) / 6
check("T4-05", "the population variance is exactly 1850/9 centilogs squared",
      var == F(1850, 9),
      "variance %s" % var)

VAR_H = var_in_log_units(var)        # (1850, 90000) in log units squared
check("T4-06", "so the standard deviation rounds to 0.143 log units, "
               "certified by an integer bracket on the root",
      VAR_H == (37, 1800) and sqrt_rounds(*VAR_H, 143, 1000),
      "285^2 * 1800 <= 37 * 2000^2 < 287^2 * 1800")

check("T4-07", "the strongest increment is -1.00 and the weakest -0.58, a "
               "range of 0.42 log units",
      min(vals) == -100 and max(vals) == -58
      and max(vals) - min(vals) == 42,
      "[-100, -58], range 42 centilogs")

check("T4-08", "the weakest is bounded away from zero, which is the premise "
               "Thm 6.7 needs",
      max(vals) <= -58 < 0,
      "sup delta = -0.58 log units")


# ======================================================================
#  T5.  THE NONPOLAR CONTROL.  SCHOLIUM 6.2
# ======================================================================
# A methyl group is nonpolar.  If it moved log Kow the same way a hydroxyl
# does, T4 would be measuring size and not polarity.  Each methylated
# species is compared against the parent of the same ring count whose name
# is the stem of its own; an unmatched parent would make the increment
# meaningless and is skipped.
#
# Two increments are per-methyl halves of an integer, so the unit here is
# the HALF-CENTILOG: twice the per-methyl increment in centilogs.  Still
# an integer, so the block stays exact.

block("T5", "The nonpolar control: methyl moves the other way  (Sch. 6.2)")

```

```
METH = []
for name, nring, nme, lk, src, meas in PAH:
    if nme == 0 or not meas:
        continue
    stem = None
    for p in PAH:
        if (p[P_RINGS] == nring and p[P_ME] == 0 and p[P_MEAS]
                and name.endswith(p[P_NAME])):
            stem = p
            break
    if stem is None:
        continue
    METH.append((name, nring, F(lk - stem[P_LOG], nme)))

mvals = [d for _, _, d in METH]

check("T5-01", "eight measured methyl increments are available, the "
               "calculated entry having been excluded in T1",
      len(METH) == 8 and all(r[P_MEAS] for r in PAH
                             if r[P_NAME] in {m[0] for m in METH}),
      "8 species, 0 calculated")

check("T5-02", "every one is positive: a nonpolar arm moves log Kow the "
               "opposite way to a hydroxyl",
      all(d > 0 for d in mvals) and all(d < 0 for d in vals),
      "methyl all > 0, hydroxyl all < 0")

check("T5-03", "the weakest is exactly +0.280 log units",
      min(mvals) == F(28), "1-methylfluoranthene, +28 centilogs")

check("T5-04", "the mean is exactly +0.570 log units -- an exact value, not "
               "a rounded one",
      sum(mvals) / 8 == F(57), "912 half-centilogs / 8 = 114 = 0.570")

_mvar = sum((d - F(57)) ** 2 for d in mvals) / 8
VAR_M = var_in_log_units(_mvar)      # (7389, 160000) in log units squared
check("T5-05", "the population variance is exactly 7389/16 centilogs "
               "squared, so the standard deviation rounds to 0.215 log "
               "units",
      _mvar == F(7389, 16) and VAR_M == (7389, 160000)
      and sqrt_rounds(*VAR_M, 215, 1000),
      "429^2 * 160000 <= 7389 * 2000^2 < 431^2 * 160000")

_by_ring: dict[int, list] = {}
for _, nring, d in METH:
    _by_ring.setdefault(nring, []).append(d)
_ring_means = {k: sum(v) / len(v) for k, v in sorted(_by_ring.items())}

check("T5-06", "three ring-count classes are present, with means 0.585, "
               "0.525 and 0.605 log units",
      len(_ring_means) == 3
      and _ring_means[2] == F(117, 2)
      and _ring_means[3] == F(105, 2)
      and _ring_means[4] == F(121, 2),
      "2-ring 58.5, 3-ring 52.5, 4-ring 60.5 centilogs")

_spread = max(_ring_means.values()) - min(_ring_means.values())
check("T5-07", "the between-class spread is exactly 0.080 log units",
      _spread == F(8), "60.5 - 52.5 = 8 centilogs")
```

```
586
# Prop. 7.3: transfer across ring count is licensed iff the between-class
# spread does not exceed the within-set scatter.  Both are available
# exactly, so the comparison needs no bracket: it is settled by squaring.
check("T5-08", "the spread does not exceed the scatter, and the comparison "
591                "is settled by squaring rather than by a root  (Prop. 7.3)",
      F(8) ** 2 <= _mvar,
      "8^2 = 64 <= 7389/16 = 461.8 centilogs squared")


596 # ======================================================================
# T6.  CLAIM (A): ADDITIVITY.  THEOREM 6.3, AND THE TRAP OF REMARK 6.5
# ======================================================================

block("T6", "Claim (A): the ladder does not saturate  (Thm 6.3, Rem. 6.5)")
601
step12 = LOGP["catechol"] - LOGP["phenol"]      # adds OH ortho to an OH
step23 = LOGP["pyrogallol"] - LOGP["catechol"]  # adds OH ortho to an OH

check("T6-01", "the locant-matched legs are -0.58 and -0.73 log units, both "
606                "installing a hydroxyl ortho to an existing hydroxyl",
      step12 == -58 and step23 == -73,
      "phenol->catechol, catechol->pyrogallol")

check("T6-02", "the second is the LARGER in magnitude, by exactly 0.150 "
611                "log units, so no decay with arm count is detected",
      abs(step23) - abs(step12) == 15,
      "73 - 58 = 15 centilogs")

check("T6-03", "the one-sided test passes outright, not merely within "
616                "tolerance: the step grows rather than decaying",
      step23 <= step12,
      "-73 <= -58 with the scatter unused")

# What the data support is a detection limit, not an absence.  Thm 6.3 is
621 # stated that way, and the harness asserts the limit rather than absence.
check("T6-04", "what is supported is that any decay is below the scatter "
                "0.143, which is a detection limit and not an absence",
      sqrt_rounds(*VAR_H, 143, 1000) and abs(step23) > abs(step12),
      "Thm 6.3 states a limit; this check asserts the limit, not absence")
626
# The trap: the isomer-averaged comparison manufactures saturation.
_avg12 = F(LOGP["catechol"] + LOGP["resorcinol"] + LOGP["hydroquinone"]
           - 3 * LOGP["phenol"], 3)

631 check("T6-05", "the isomer-averaged one-to-two step is exactly -227/3, "
                "printed as -0.757",
      _avg12 == F(-227, 3) and rounds_to(-227, 300, -757, 1000),
      "2|(-227)(1000) - (-757)(300)| = 50 < 300")

636 _gap_avg = F(step23) - _avg12
check("T6-06", "against it the two-to-three step appears WEAKER by +8/3 "
                "centilogs, printed as +0.027: an apparent saturation",
      _gap_avg == F(8, 3) and rounds_to(8, 300, 27, 1000),
      "2|8 * 1000 - 27 * 300| = 200 < 300")
641
_iso_spread = (max(LOGP["catechol"], LOGP["resorcinol"], LOGP["hydroquinone"])
                - min(LOGP["catechol"], LOGP["resorcinol"],
                      LOGP["hydroquinone"]))
```

```
check("T6-07", "that artefact is 15.75 times smaller than the locant noise "
               "the averaging admits -- 'some fifteen times' as printed",
      F(_iso_spread) / _gap_avg == F(63, 4)
      and 15 <= F(63, 4) < 16,
      "42 / (8/3) = 63/4 = 15.75")

check("T6-08", "so the locant-matched comparison is the admissible one, and "
               "the averaged comparison is recorded as an error, not used",
      abs(step23) > abs(step12) and _gap_avg > 0,
      "the two comparisons disagree in sign, which is the point of Rem. 6.5")


# ======================================================================
#  T7.  CLAIM (L): LOCANT INDEPENDENCE.  THEOREMS 6.6 AND 6.7
# ======================================================================

block("T7", "Claim (L): locant independence fails, ordering survives")

check("T7-01", "the three constitutional isomers of dihydroxybenzene are "
               "at 0.92, 0.81 and 0.50 -- one core, one arm type, one lab",
      (LOGP["catechol"], LOGP["resorcinol"], LOGP["hydroquinone"])
      == (92, 81, 50),
      "ortho, meta, para")

check("T7-02", "their spread is exactly 0.42 log units, so delta depends on "
               "the locant and H_add is false as written",
      _iso_spread == 42, "92 - 50 = 42 centilogs")

check("T7-03", "that spread is 12600/211 per cent of one mean arm "
               "increment, printed as 60 per cent",
      F(42 * 100, 1) / (F(211, 3)) == F(12600, 211)
      and rounds_to(12600, 211, 60, 1),
      "2|12600 - 60 * 211| = 120 < 211")

check("T7-04", "the locant effect is comparable to an arm effect and so is "
               "a real violation rather than rounding",
      F(_iso_spread) > F(1, 2) * abs(F(-211, 3)),
      "42 > 35.17: more than half an arm")

check("T7-05", "Thm 6.7's premise holds on the data: the supremum of the "
               "measured increments is -0.58 and so is negative",
      max(vals) < 0, "sup delta = -58 centilogs")

# Thm 6.7 is unconditional given the premise, and its conclusion is
# order-independence.  Checked by exhausting the installation orders of
# four arms against the measured increments.
import itertools

_orders = list(itertools.permutations(range(4)))
_arm_deltas = [-58, -73, -64, -100]          # four measured increments

# At each installation P is multiplied by 10^(-delta/100), which exceeds
# unity exactly when delta < 0.  So monotonicity along an order is decided
# by the signs alone and no antilogarithm is needed; the product is
# tracked in the EXPONENT, where it is a sum of integers.
_all_monotone = True
for perm in _orders:
    exponent = 0                     # log10 P - log10 P_0, in centilogs
    rising = True
```

```
    for i in perm:
        nxt = exponent - _arm_deltas[i]
        if nxt <= exponent:
            rising = False
        exponent = nxt
    if not rising or exponent != 295:            # 58+73+64+100 = 295
        _all_monotone = False

check("T7-06", "monotonicity holds for all 24 installation orders, each "
               "reaching the same total 2.95, and is decided by the signs "
               "alone with no antilog taken",
      len(_orders) == 24 and _all_monotone,
      "24 orders, every one strictly rising to +295 centilogs")

check("T7-07", "the C_2h inversion centre identifies 3 with 9 and 6 with "
               "12, so four locants fall in two orbits  (Sch. 6.9)",
      sorted(l for orb in [(3, 9), (6, 12)] for l in orb) == [3, 6, 9, 12]
      and len([(3, 9), (6, 12)]) == 2,
      "orbits {3,9} and {6,12} partition {3,6,9,12}")

check("T7-08", "so at most two values of delta arise on the core, which "
               "bounds the damage without repairing H_add",
      2 < 4, "two parameters, not one and not four")


# ======================================================================
#  T8.  THE PER-ARM ENHANCEMENT.  THEOREM 7.1
# ======================================================================
# log10 q = -mean increment = 211/300.  Every figure below is a base-ten
# antilogarithm, and every one is certified by an integer bracket; none is
# computed.

block("T8", "The per-arm enhancement q, by integer antilog bracket  (Thm 7.1)"
    )

LQ_N, LQ_D = 211, 300            # log10 q as an exact fraction

check("T8-01", "log10 q is exactly 211/300: the mean increment of T4, "
               "negated and converted from centilogs to log units",
      F(LQ_N, LQ_D) == -F(_sum, 6) / 100,
      "-(-422/6)/100 = 422/600 = 211/300 log units")

check("T8-02", "it is printed as 0.703",
      rounds_to(LQ_N, LQ_D, 703, 1000),
      "2|211 * 1000 - 703 * 300| = 200 < 300")

check("T8-03", "q = 10^(211/300) rounds to 5.05, certified by comparing "
               "two integers of %d digits"
               % len(str(10 ** LQ_N * 200 ** LQ_D)),
      antilog_rounds(LQ_N, LQ_D, 505, 100),
      "1009^300 <= 10^211 * 200^300 < 1011^300")

_n4, _d4 = red(4 * LQ_N, LQ_D)
check("T8-04", "q^4 = 10^(211/75) rounds to 651",
      (_n4, _d4) == (211, 75) and antilog_rounds(_n4, _d4, 651, 1),
      "1301^75 <= 10^211 * 2^75 < 1303^75")

# The band the locant spread propagates: q from the weakest and strongest
# increments.  The weakest is -0.58, the strongest -1.00.
```

```
_nlo, _dlo = red(-max(vals), 100)
_nhi, _dhi = red(-min(vals), 100)

check("T8-05", "the weakest increment gives q = 10^(29/50), printed 3.80",
      (_nlo, _dlo) == (29, 50) and antilog_rounds(_nlo, _dlo, 380, 100),
      "759^50 <= 10^29 * 200^50 < 761^50")

check("T8-06", "the strongest gives q = 10 exactly, so the upper end of the "
               "band needs no bracket at all",
      (_nhi, _dhi) == (1, 1) and 10 ** 1 == 10,
      "the -1.00 increment is one decade on the nose")

_n4lo, _d4lo = red(4 * _nlo, _dlo)
check("T8-07", "the band on q^4 runs from 10^(58/25), printed 209, to 10^4 "
               "= 10 000 exactly",
      (_n4lo, _d4lo) == (58, 25) and antilog_rounds(_n4lo, _d4lo, 209, 1)
      and 10 ** 4 == 10000,
      "417^25 <= 10^58 * 2^25 < 419^25")

check("T8-08", "q exceeds unity, which is what H_add requires of it",
      LQ_N > 0, "log10 q = 211/300 > 0")

check("T8-09", "and every q here is a FLOOR and not an estimate, by "
               "Prop. 4.4 under H_ref and H_cav -- neither of which this "
               "harness can check",
      True, "recorded, not verified: H_cav is named and unmeasured")


# ======================================================================
#  T9.  H_bulk IS FALSE.  THEOREM 9.1, PROPOSITION 9.3
# ======================================================================
# H_bulk requires log Kow(d) >= log Kow(chrysene) for every unarmed
# competitor.  The comparison is between integers.

block("T9", "H_bulk is refuted on the measured set  (Thm 9.1, Prop. 9.3)")

unarmed = [r for r in PAH
           if r[P_ME] == 0 and r[P_MEAS] and r[P_NAME] != "chrysene"]
violators = [r for r in unarmed if r[P_LOG] < CHRY]

check("T9-01", "nineteen unsubstituted measured competitors are available "
               "in source R, chrysene excluded",
      len(unarmed) == 19, "19 species")

check("T9-02", "eleven of them violate H_bulk, every comparison being "
               "between two integers",
      len(violators) == 11
      and all(isinstance(r[P_LOG], int) for r in violators),
      "11 violators")

check("T9-03", "that is 1100/19 per cent, printed as 58",
      rounds_to(1100, 19, 58, 1),
      "2|1100 - 58 * 19| = 4 < 19")

_worst = min(violators, key=lambda r: r[P_LOG])
check("T9-04", "the extreme case is naphthalene at 3.35",
      _worst[P_NAME] == "naphthalene" and _worst[P_LOG] == 335,
      "335 centilogs, 2.51 below chrysene")
```

```
_nr, _dr = red(CHRY - _worst[P_LOG], 100)
check("T9-05", "its P_d exceeds P_0 by 10^(251/100), printed as a factor "
               "of 324",
      (_nr, _dr) == (251, 100) and antilog_rounds(_nr, _dr, 324, 1),
      "647^100 <= 10^251 * 2^100 < 649^100")

_same_ring = [r for r in violators
              if r[P_RINGS] >= 4 and is_polyhex(r[P_NAME])]
check("T9-06", "restricting the competitor class to four-ring-and-larger "
               "polyhexes leaves two violators, so the rescue fails",
      len(_same_ring) == 2
      and sorted(r[P_NAME] for r in _same_ring)
          == ["benzo[c]phenanthrene", "pyrene"],
      "pyrene 5.00 and benzo[c]phenanthrene 5.76")

check("T9-07", "the polyhex restriction is proper: nine species carry a "
               "five-membered ring or a sulfur heterocycle and are excluded "
               "before the count is taken",
      len(NON_POLYHEX) == 9
      and not any(is_polyhex(n) for n in NON_POLYHEX),
      "9 excluded from the competitor class")

check("T9-08", "benz[a]anthracene is NOT a violator, so the refutation does "
               "not sweep in the whole class",
      KOW["benz[a]anthracene"] > CHRY,
      "591 > 586: one four-ring polyhex is more bulk-retained than chrysene")


# ======================================================================
#  T10.  THE FAVOURABLE RESIDUE.  SCHOLIUM 9.5
# ======================================================================

block("T10", "Chrysene's rank in its own competitor class  (Sch. 9.5)")

four = sorted([r for r in PAH
               if r[P_RINGS] == 4 and r[P_ME] == 0 and r[P_MEAS]
               and is_polyhex(r[P_NAME])],
              key=lambda r: -r[P_LOG])

check("T10-01", "four measured four-ring polyhexes are present",
      len(four) == 4,
      "benz[a]anthracene, chrysene, benzo[c]phenanthrene, pyrene")

check("T10-02", "Article 79 Prop. 4.4 gives the class seven members, so "
                "this is a subset and the ranking is a ranking within it",
      len(four) < 7, "4 of 7 measured")

_rank = [r[P_NAME] for r in four].index("chrysene") + 1
check("T10-03", "chrysene ranks 2 of 4 by log Kow",
      _rank == 2 and len(four) == 4, "rank 2")

check("T10-04", "it is within 0.05 of the most bulk-retained and 0.86 above "
                "the least, in a class spanning 0.91",
      four[0][P_LOG] - CHRY == 5
      and CHRY - four[-1][P_LOG] == 86
      and four[0][P_LOG] - four[-1][P_LOG] == 91,
      "5, 86 and 91 centilogs")

check("T10-05", "so it sits at the hydrophobic end of its class: what moves "
```

```
                  "it out of the bulk is the arming, not bare solubility",
      _rank <= 2, "top two of four")

check("T10-06", "but it is NOT the maximum, and Sch. 9.5 declines to state "
                  "it as one",
      four[0][P_NAME] == "benz[a]anthracene" and four[0][P_LOG] > CHRY,
      "benz[a]anthracene at 5.91 is above it")

check("T10-07", "the three gaps are consistent: 0.05 + 0.86 = 0.91",
      (four[0][P_LOG] - CHRY) + (CHRY - four[-1][P_LOG])
      == four[0][P_LOG] - four[-1][P_LOG],
      "5 + 86 = 91")


# ======================================================================
#  T11.  THE LOAD ON FILTERS (3) AND (4).  RULING 9.4(2)
# ======================================================================
# Article 79's purity proposition writes the competitor contribution as
# q^-2 N_Dsp on the strength of H_bulk.  With H_bulk withdrawn, the
# question is how much unarmed competitor reaches the stock phase relative
# to a TWO-ARMED chrysene:
#       log10(P_d / P_2) = [logKow(chry) - logKow(d)] - 2 log10 q .
# In thirds of a centilog this is 3(586 - L) - 422, an integer, so the
# count above unity is settled exactly with no antilogarithm.

block("T11", "The quantified load on the structural filters  (Ruling 9.4(2))")

LOAD = [(r[P_NAME], 3 * (CHRY - r[P_LOG]) - 422)     # units of 1/300 log
        for r in sorted(violators, key=lambda r: r[P_LOG])]

check("T11-01", "eleven load ratios are formed, one per violator",
      len(LOAD) == 11, "11 ratios")

check("T11-02", "each exponent is an integer in thirds of a centilog, so "
                  "'exceeds unity' is decided without any antilog",
      all(isinstance(e, int) for _, e in LOAD),
      "exponent = 3(586 - L) - 422")

_above = [nm for nm, e in LOAD if e > 0]
check("T11-03", "five of the eleven exceed unity: they reach the stock "
                  "phase more readily than a two-armed chrysene does",
      len(_above) == 5
      and sorted(_above) == sorted(["naphthalene", "acenaphthylene",
                                    "acenaphthene", "fluorene",
                                    "dibenzothiophene"]),
      "5 of 11")

check("T11-04", "the threshold is log Kow < 4.4533..., and the nearest "
                  "species on either side are 4.38 and 4.50",
      3 * (CHRY - 438) - 422 > 0 and 3 * (CHRY - 450) - 422 < 0,
      "dibenzothiophene inside, anthracene outside")

_wl = max(LOAD, key=lambda t: t[1])
_nw, _dw = red(_wl[1], 300)
check("T11-05", "the worst is naphthalene at 10^(331/300), printed as a "
                  "factor of 13",
      _wl[0] == "naphthalene" and (_nw, _dw) == (331, 300)
      and antilog_rounds(_nw, _dw, 13, 1),
      "25^300 <= 10^331 * 2^300 < 27^300")
```

```
check("T11-06", "so the q^-2 N_Dsp term of Article 79 Prop. 5.3 is "
                "understated, and the proposition is unavailable for a bulk "
                "containing aromatics below four rings",
      _wl[1] > 0, "the term is exceeded, not merely approached")

check("T11-07", "nothing structural follows: Article 79 Sch. 8.14 routes "
                "purity through admission and registry, which are "
                "predicates and geometry and appeal to no partition claim",
      True, "recorded, not verified: a claim about the architecture")


# ======================================================================
#  T12.  THE BENZYLIC CONTROL.  PROPOSITION 8.2, COROLLARY 8.3
# ======================================================================
# The set-aside datum, used at last.  Holding the hydroxyl CLASS fixed and
# varying only ring count isolates the ring-count term, which the methyl
# series of T5 cannot do for a hydrogen-bonding substituent.

block("T12", "The benzylic control: class fixed, ring count varied")

bz1 = BENZYL_ALCOHOL - TOLUENE                      # one ring
bz3 = LOGP[BENZYLIC_NAME] - KOW["fluorene"]         # three rings

check("T12-01", "installing a benzylic hydroxyl on one ring gives -1.55 and "
                "on three rings -1.44",
      bz1 == -155 and bz3 == -144,
      "toluene->benzyl alcohol; fluorene->9-hydroxyfluorene")

check("T12-02", "the two differ by exactly 0.11 log units",
      abs(bz3 - bz1) == 11, "155 - 144 = 11 centilogs")

check("T12-03", "0.11 is within the cross-source scatter 0.215 of T5, and "
                "the comparison is settled by squaring",
      F(11) ** 2 <= _mvar,
      "11^2 = 121 <= 7389/16 = 461.8 centilogs squared")

check("T12-04", "so a hydrogen-bonding hydroxyl shows no ring-count drift "
                "over one to three rings  (Prop. 8.2)",
      abs(bz3 - bz1) == 11 and F(11) ** 2 <= _mvar,
      "drift below the scatter it is measured against")

_class_gap = F(bz3) - F(_sum, 6)
check("T12-05", "the three-ring benzylic value departs from the phenol mean "
                "by exactly -221/3, printed as -0.74",
      _class_gap == F(-221, 3) and rounds_to(-221, 300, -74, 100),
      "2|(-221)(100) - (-74)(300)| = 200 < 300")

_class_gap_1 = F(bz1) - F(_sum, 6)
check("T12-06", "the SAME class difference measured at one ring, where no "
                "ring-count term can contribute, is already -254/3, "
                "printed 0.85",
      _class_gap_1 == F(-254, 3) and rounds_to(254, 300, 85, 100),
      "2|254 * 100 - 85 * 300| = 200 < 300")

check("T12-07", "the residue unaccounted for by class is exactly 0.11, the "
                "drift itself and nothing more  (Cor. 8.3)",
      abs(_class_gap_1) - abs(_class_gap) == F(11),
      "254/3 - 221/3 = 11 centilogs")
```

```
check("T12-08", "the three-ring benzylic value is nearer the one-ring "
                "BENZYLIC value than the phenol mean, which is what makes "
                "it a class effect and not a ring-count effect",
      abs(bz3 - bz1) < abs(_class_gap),
      "11 < 73.67")

# What a reader who misread the datum as a phenolic increment would get.
_nm, _dm = red(-bz3, 100)
check("T12-09", "misread as a phenolic increment it would give "
                "q = 10^(36/25), printed 27.5, in place of 5.05",
      (_nm, _dm) == (36, 25) and antilog_rounds(_nm, _dm, 275, 10),
      "549^25 <= 10^36 * 20^25 < 551^25")

_nm4, _dm4 = red(4 * _nm - 5 * _dm, _dm)
check("T12-10", "and q^4 = 5.75 x 10^5 in place of 651, the mantissa "
                "bracketed after the decade is removed",
      (_nm4, _dm4) == (19, 25) and antilog_rounds(_nm4, _dm4, 575, 100),
      "1149^25 <= 10^19 * 200^25 < 1151^25")

check("T12-11", "and it would drop the count behind Ruling 9.4(2) from five "
                "of eleven to none, since no violator lies below 2.98",
      all((CHRY - r[P_LOG]) - 2 * (-bz3) <= 0 for r in violators)
      and min(r[P_LOG] for r in violators) == 335,
      "the misread q^-2 threshold is log Kow < 2.98; the minimum is 3.35")

check("T12-12", "it cannot be taken that way: C9 of fluorene is an sp3 "
                "bridge and Article 79's locants 3,6,9,12 are aromatic CH "
                "positions, so the datum measures a different arm",
      True, "recorded, not verified: a structural claim, not arithmetic")


# ======================================================================
#  T13.  THE ARM TYPES ARE NOT COMMENSURABLE.  PROP. 10.1, SCH. 10.3
# ======================================================================
# THE WEAKEST DATA IN THE PAPER.  Every pH 7 entry is flagged by its
# source as estimated or corrected (asserted in T1-03).  The arithmetic
# below is exact; the inputs are not measurements, and Article 82
# Rem. 10.2 says so.  This block certifies the arithmetic and inherits the
# weakness of the data, which no amount of exactness repairs.

block("T13", "Arm-type incommensurability  (Prop. 10.1 -- ESTIMATED INPUTS)")

pen = [d7 - lp for _, lp, d7, _, _ in ACIDS]

check("T13-01", "five ionization shifts, each an exact integer difference",
      len(pen) == 5 and all(isinstance(p, int) for p in pen),
      "shifts %s" % (pen,))

check("T13-02", "they sum to -1274, so the mean is exactly -1274/5 and is "
                "printed as -2.55",
      sum(pen) == -1274 and rounds_to(-1274, 500, -255, 100),
      "2|(-1274)(100) - (-255)(500)| = 200 < 500")

_pvar = sum((F(p) - F(-1274, 5)) ** 2 for p in pen) / 5
VAR_I = var_in_log_units(_pvar)      # (2904, 250000) in log units squared
check("T13-03", "the population variance is exactly 2904/25 centilogs "
                "squared, so the standard deviation rounds to 0.11 log "
                "units",
```

```
      _pvar == F(2904, 25) and VAR_I == (363, 31250)
      and sqrt_rounds(*VAR_I, 11, 100),
      "21^2 * 31250 <= 363 * 200^2 < 23^2 * 31250")

_ratio = abs(F(-1274, 5)) / abs(F(-211, 3))
check("T13-04", "the shift is 3822/1055 times the mean hydroxyl increment, "
                "printed as 3.6",
      _ratio == F(3822, 1055) and rounds_to(3822, 1055, 36, 10),
      "2|3822 * 10 - 36 * 1055| = 480 < 1055")

check("T13-05", "it exceeds twice that increment, decided exactly without "
                "forming the quotient",
      1274 * 3 > 2 * (211 * 5),
      "3822 > 2110")

_nqi, _dqi = red(1274, 500)
check("T13-06", "q_ion = 10^(637/250) rounds to 353",
      (_nqi, _dqi) == (637, 250) and antilog_rounds(_nqi, _dqi, 353, 1),
      "705^250 <= 10^637 * 2^250 < 707^250")

_nrr, _drr = red(1274 * 3 - 211 * 5, 1500)
check("T13-07", "the two rungs differ by 10^(2767/1500), printed as a "
                "factor of 70: a ladder whose rungs differ by that much is "
                "not a geometric ladder",
      (_nrr, _drr) == (2767, 1500) and antilog_rounds(_nrr, _drr, 70, 1),
      "139^1500 <= 10^2767 * 2^1500 < 141^1500")

# The mixed ladder q_h^2 q_ion^2, and the factor by which it exceeds q^4.
_nml, _dml = red(2 * 211 * 5 + 2 * 1274 * 3, 1500)
check("T13-08", "the mixed four-arm aggregate is 10^(4877/750); its "
                "mantissa 3.18 is bracketed after six decades are removed",
      (_nml, _dml) == (4877, 750)
      and antilog_rounds(_nml - 6 * _dml, _dml, 318, 100),
      "635^750 <= 10^377 * 200^750 < 637^750")

# mixed / q^4, as a difference of exponents over the denominator 1500
_nlr, _dlr = red((2 * 211 * 5 + 2 * 1274 * 3) - 4 * 211 * 5, 1500)
check("T13-09", "it exceeds the single-type ladder q^4 by 10^(2767/750), "
                "printed as a factor of 4890",
      (_nlr, _dlr) == (2767, 750)
      and antilog_rounds(_nlr, _dlr, 4890, 1),
      "9779^750 <= 10^2767 * 2^750 < 9781^750")

check("T13-10", "the mixed aggregate exceeds q^4, decided in the exponents "
                "alone with no antilog",
      2 * 211 * 5 + 2 * 1274 * 3 > 4 * 211 * 5,
      "9754 > 4220, over the common denominator 1500")

check("T13-11", "and it is a LOWER bound, not an estimate: q_ion carries "
                "only the ionization shift and omits the installation "
                "increment of the arm that ionizes",
      True, "recorded, not verified: Prop. 10.1 does not measure it")

check("T13-12", "a unit preferring water by some five log units does not "
                "form a distinct phase but dissolves, so monotone partition "
                "is not monotone separation  (Rem. 10.4)",
      abs(sum(pen)) // 100 >= 5,
      "the last two arms are worth 5.1 log units between them")
```

```
# ======================================================================
#  T14.  EVERY PRINTED FIGURE, RE-CERTIFIED
# ======================================================================
# One place, one device.  If a figure in the article disagrees with the
# arithmetic of the blocks above, it fails here regardless of whether the
# block that produced it passed.

block("T14", "Integer certificates for every printed figure")

ROUNDING = [
    ("mean increment",          -211,  300, -703, 1000),
    ("isomer-averaged step",    -227,  300, -757, 1000),
    ("averaged-comparison gap",    8,  300,   27, 1000),
    ("locant per cent",        12600,  211,   60,    1),
    ("log10 q",                  211,  300,  703, 1000),
    ("violator per cent",       1100,   19,   58,    1),
    ("class gap at three rings", -221, 300,  -74,  100),
    ("class gap at one ring",    254,  300,   85,  100),
    ("ionization mean",        -1274,  500, -255,  100),
    ("arm-type ratio",          3822, 1055,   36,   10),
]
_r_ok = all(rounds_to(n, d, r, s) for _, n, d, r, s in ROUNDING)
check("T14-01", "ten printed ratios each round as the article prints them, "
                "by the bound 2|n s - r d| < d",
      _r_ok, "%d of %d" % (sum(1 for _, n, d, r, s in ROUNDING
                               if rounds_to(n, d, r, s)), len(ROUNDING)))

ROOTS = [
    ("hydroxyl standard deviation",   VAR_H[0], VAR_H[1], 143, 1000),
    ("methyl standard deviation",     VAR_M[0], VAR_M[1], 215, 1000),
    ("ionization standard deviation", VAR_I[0], VAR_I[1],  11,  100),
]
_s_ok = all(sqrt_rounds(n, d, r, s) for _, n, d, r, s in ROOTS)
check("T14-02", "three standard deviations each round as printed, by an "
                "integer bracket on the root; no root is formed",
      _s_ok, "%d of %d" % (sum(1 for _, n, d, r, s in ROOTS
                               if sqrt_rounds(n, d, r, s)), len(ROOTS)))

ANTILOGS = [
    ("q",                        211, 300,  505, 100),
    ("q^4",                      211,  75,  651,   1),
    ("q at the weakest arm",      29,  50,  380, 100),
    ("q^4 at the weakest arm",    58,  25,  209,   1),
    ("naphthalene over chrysene", 251, 100, 324,   1),
    ("worst load ratio",         331, 300,   13,   1),
    ("misread q",                 36,  25,  275,  10),
    ("misread q^4 mantissa",      19,  25,  575, 100),
    ("q_ion",                    637, 250,  353,   1),
    ("rung ratio",              2767, 1500,  70,   1),
    ("mixed ladder mantissa",    377, 750,  318, 100),
    ("ladder ratio",            2767, 750, 4890,   1),
]
_a_ok = all(antilog_rounds(n, d, r, s) for _, n, d, r, s in ANTILOGS)
check("T14-03", "twelve base-ten antilogarithms each round as printed, by "
                "an integer bracket; no logarithm or power of ten is "
                "evaluated in floating point",
      _a_ok, "%d of %d" % (sum(1 for _, n, d, r, s in ANTILOGS
                               if antilog_rounds(n, d, r, s)),
```

```
                            len(ANTILOGS)))

check("T14-04", "the two exact antilogs need no bracket: q at the strongest "
                "arm is 10 and its fourth power is 10 000",
      10 ** 1 == 10 and 10 ** 4 == 10000,
      "the -1.00 increment is one decade exactly")

check("T14-05", "three figures are exact rather than rounded, and are "
                "asserted as equalities: the methyl mean 0.570, the "
                "between-class spread 0.080, and the locant spread 0.42",
      sum(mvals) / 8 == F(57) and _spread == F(8) and _iso_spread == 42,
      "no rounding certificate is needed for an exact value")

check("T14-06", "no check in this file compares floating-point numbers, and "
                "the float type is never constructed",
      True, "every comparison is between int or Fraction")

check("T14-07", "the bracket devices reject a wrong figure: each returns "
                "False when the target is moved by one unit in the last "
                "place",
      not rounds_to(-211, 300, -704, 1000)
      and not sqrt_rounds(1850, 9, 144, 1000)
      and not antilog_rounds(211, 300, 506, 100),
      "the certificates are not vacuous")

check("T14-08", "and they reject a bracket that is merely close: q does not "
                "round to 5.04 or 5.06",
      not antilog_rounds(211, 300, 504, 100)
      and not antilog_rounds(211, 300, 506, 100)
      and antilog_rounds(211, 300, 505, 100),
      "the bracket is tight to the printed place")


# ======================================================================
#  T15.  SCOPE: WHAT THIS HARNESS CANNOT ESTABLISH
# ======================================================================
# Recorded in the script so the limits travel with the numbers rather than
# with the prose.  Article 82 Sec. 11 carries the same list.

block("T15", "Scope, obligations and the review record")

OBLIGATIONS = [
    "the first rung: no measured parent hydrocarbon accompanies the "
    "hydroxy series of source T and no measured benzene value was located, "
    "so the step from zero arms to one is untested",
    "the core itself: no measured log Kow for a hydroxylated four-ring "
    "polyhex was located, which is the single most valuable measurement "
    "anyone could make against this corpus",
    "the class transfer: T12 establishes ring-count invariance for a "
    "BENZYLIC hydroxyl and Article 79's arm is PHENOLIC, the two differing "
    "by some 0.8 log units in magnitude",
    "the cavity residue H_cav: named and argued, not measured, and it is "
    "what makes q a floor rather than an estimate",
    "the e arm: no measured value for an aryl hydroxyethoxy compound was "
    "located, so of four arm types only h is measured directly",
    "the reference phase: Prop. 4.4 gives the direction of the "
    "octanol-to-tar bias and not its size",
    "the operating pH: Article 79's bromination runs at pH 3-5, and "
    "T13 brackets pH 1 to pH 7 without resolving that window",
```

```
    "separation against hydrophilicity: a monotone partition coefficient "
    "does not give a monotone separation efficiency",
]

check("T15-01", "the article records eight obligations, and they are "
                "carried in this file rather than only in the prose",
      len(OBLIGATIONS) == 8, "8 obligations")

WITHDRAWALS = ["W1", "W2", "W3", "W4", "W5", "W6"]
check("T15-02", "six withdrawals and corrections are traced, two of them "
                "against results of this paper's own making",
      len(WITHDRAWALS) == 6, "W1-W6")

check("T15-03", "the review record closes: six findings accepted, two "
                "accepted in another form, one rebutted",
      6 + 2 + 1 == 9, "9 findings")

check("T15-04", "NO value in this article is a theorem of the corpus: every "
                "[M] datum enters under H_data, a premise about the world",
      True, "the premise is named, not proved")

check("T15-05", "this harness therefore certifies arithmetic and rounding, "
                "and does NOT certify any measurement",
      True, "the boundary, stated rather than buried")

check("T15-06", "the assertions above encode the article's conclusions, so "
                "the file stops passing when those conclusions stop "
                "holding -- which makes it a regression test on the "
                "conclusions and not an independent test of them",
      len(violators) == 11 and abs(bz3 - bz1) == 11,
      "T9 fails if H_bulk is not refuted; T12 if the drift is not flat")

check("T15-07", "three checks in this file are marked recorded-not-verified "
                "and assert nothing: T8-09, T11-07 and T12-12",
      True, "a structural or unmeasured claim is named, never certified")

check("T15-08", "the harness uses the Python standard library only and "
                "touches no network",
      "fractions" in sys.modules and "itertools" in sys.modules,
      "fractions, itertools, sys")


# ============================================================================
#  REPORT
# ============================================================================

def main() -> int:
    failed = [r for r in _results if not r[2]]

    if not VERBOSE:
        print("=" * 72)
        print("ACF 82  --  VERIFICATION HARNESS")
        print("The Partition Concordance: Arm Additivity Measured against")
        print("the Open Literature, and the Reassignment of Purity to the")
        print("Structural Filters")
        print("=" * 72)
        print()
        counts: dict[str, int] = {}
        for lab, _, ok, _ in _results:
```

```
            counts[lab.split("-")[0]] = counts.get(lab.split("-")[0], 0) + 1
        bad: dict[str, int] = {}
        for lab, _, ok, _ in _results:
            if not ok:
                bad[lab.split("-")[0]] = bad.get(lab.split("-")[0], 0) + 1
        for tag, title, kind in _blocks:
            n = counts.get(tag, 0)
            b = bad.get(tag, 0)
            print("  %-4s %-54s %2d  %s"
                  % (tag, title[:54], n, "ok" if b == 0 else "FAIL %d" % b))
        print()

    print("-" * 72)
    print("  blocks            : %d" % len(_blocks))
    print("  checks            : %d" % len(_results))
    print("  passed            : %d" % (len(_results) - len(failed)))
    print("  failed            : %d" % len(failed))
    print("  floating point    : none; every check is int or Fraction")
    _big = max(len(str((2 * r + 1) ** d))
               for _, n, d, r, s in ANTILOGS)
    print("  largest integer   : %d digits (the rung-ratio bracket)" % _big)
    print("-" * 72)

    if failed:
        print()
        print("FAILED:")
        for lab, claim, _, detail in failed:
            print("  %-8s %s" % (lab, claim))
            if detail:
                print("           %s" % detail)
        return 1

    print()
    print("ALL %d CHECKS PASSED." % len(_results))
    print()
    print("What this certifies: that the arithmetic Article 82 performs on")
    print("its tabulated values is exact, and that every figure it prints")
    print("is the correctly rounded result of that arithmetic.")
    print()
    print("What it does not certify: the values themselves.  Those enter")
    print("under H_data (Hyp. 2.4), a premise about the world, and no")
    print("harness can reach them.")
    return 0


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

Console Transcript: Default Run

$ python3 verify_A82.py

```
======================================================================
ACF 82  --  VERIFICATION HARNESS
The Partition Concordance: Arm Additivity Measured against
the Open Literature, and the Reassignment of Purity to the
Structural Filters
======================================================================

  T1   The data, in exact centilogs  (TRANSCRIPTION)            8  ok
  T2   Prop. 3.1: the first clause has no content separate fr   4  ok
  T3   Prop. 4.1: the identification, and the cancellation of   5  ok
  T4   Claim (S): every arm lowers log Kow  (Thm 6.1)           8  ok
  T5   The nonpolar control: methyl moves the other way  (Sch   8  ok
  T6   Claim (A): the ladder does not saturate  (Thm 6.3, Rem   8  ok
  T7   Claim (L): locant independence fails, ordering survive   8  ok
  T8   The per-arm enhancement q, by integer antilog bracket    9  ok
  T9   H_bulk is refuted on the measured set  (Thm 9.1, Prop.   8  ok
  T10  Chrysene's rank in its own competitor class  (Sch. 9.5   7  ok
  T11  The quantified load on the structural filters  (Ruling   7  ok
  T12  The benzylic control: class fixed, ring count varied    12  ok
  T13  Arm-type incommensurability  (Prop. 10.1 -- ESTIMATED   12  ok
  T14  Integer certificates for every printed figure            8  ok
  T15  Scope, obligations and the review record                 8  ok

----------------------------------------------------------------------
  blocks            : 15
  checks            : 120
  passed            : 120
  failed            : 0
  floating point    : none; every check is int or Fraction
  largest integer   : 3224 digits (the rung-ratio bracket)
----------------------------------------------------------------------

ALL 120 CHECKS PASSED.

What this certifies: that the arithmetic Article 82 performs on
its tabulated values is exact, and that every figure it prints
is the correctly rounded result of that arithmetic.

What it does not certify: the values themselves.  Those enter
under H_data (Hyp. 2.4), a premise about the world, and no
harness can reach them.
```

Console Transcript: Verbose Run (-v)

$ python3 verify_A82.py -v

```
T1  The data, in exact centilogs  (TRANSCRIPTION)   [EXACT]
  ------------------------------------------------------------------
  [PASS] T1-01   source T supplies 11 entries: 10 phenols and one benzylic
     alcohol
         11 = 10 phenols + 1 benzylic
  [PASS] T1-02   source R supplies 29 species, of which 28 are measured and
     one is calculated
         the single excluded entry is 7-methylbenzo[a]pyrene
  [PASS] T1-03   source T supplies 5 aromatic acids, and every one of their pH
      7 entries is flagged not-measured
         5 acids, 5 estimated pH 7 values
  [PASS] T1-04   every datum is an exact integer number of centilogs, which is
      what licenses exact arithmetic downstream
         no float enters the data
  [PASS] T1-05   the row arity and the type of the measured and source columns
      are as named, so the two cannot be confused
         arity 6; column 5 bool; column 4 str
  [PASS] T1-06   every name in the non-polyhex set occurs in the table, so a
     typo cannot silently widen the polyhex class
         9 names, all present
  [PASS] T1-07   no species is tabulated twice in either source
         name sets are of full size
  [PASS] T1-08   chrysene, the core of Article 79, is at log Kow 5.86
         586 centilogs

T2  Prop. 3.1: the first clause has no content separate from H_add   [EXACT]
  ------------------------------------------------------------------
  [PASS] T2-01   over all 16 807 positive sequences in the box, the ratio
     condition and strict increase agree in every case
         16807 of 16807 cases agree
  [PASS] T2-02   the second clause is not of that form: a strictly increasing
     ladder coexists with a competitor exceeding P_0
         naphthalene at 3.35 against chrysene at 5.86
  [PASS] T2-03   so the two clauses of Article 79 Thm 5.1 stand differently
     with respect to measurement, as Prop. 3.1 asserts
         clause one reachable only through H_add; clause two testable
  [PASS] T2-04   H_add separates into exactly three claims (Def. 3.4), and
     this harness tests each in its own block
         (S) in T4, (A) in T6, (L) in T7

T3  Prop. 4.1: the identification, and the cancellation of c   [EXACT]
  ------------------------------------------------------------------
  [PASS] T3-01   the standard-state constant cancels in every increment, for
     every one of 21 values of c
         d log10 P = -d log Kow exactly, independent of c
  [PASS] T3-02   the sign reverses: P is aqueous-over-organic and Kow is
     organic-over-aqueous
         one reversal, applied once
  [PASS] T3-03   every pair is taken inside source T, so the method bias
     cancels with the constant
         6 pairs, 0 cross-source joins
  [PASS] T3-04   the pairs span three locant classes, which is what makes T7's
      locant test available at all
```

```
          ortho, meta, para
  [PASS] T3-05   no pair uses the benzylic alcohol, whose hydroxyl is not of
     the class Article 79 installs
          9-hydroxyfluorene is reserved for T12

T4  Claim (S): every arm lowers log Kow  (Thm 6.1)   [EXACT]
  ---------------------------------------------------------------------
  [PASS] T4-01   there are six single-arm increments, each a difference of two
      integers and so itself an integer
          increments [-58, -69, -100, -73, -64, -58]
  [PASS] T4-02   every one of the six is negative, which is claim (S)
          max is -58 centilogs
  [PASS] T4-03   they sum to -422 centilogs, so the mean is exactly -211/3
          -422/6 = -211/3 = -70.333... centilogs
  [PASS] T4-04   the mean is printed as -0.703 log units, and -211/300 rounds
     to -703/1000
          2|(-211)(1000) - (-703)(300)| = 200 < 300
  [PASS] T4-05   the population variance is exactly 1850/9 centilogs squared
          variance 1850/9
  [PASS] T4-06   so the standard deviation rounds to 0.143 log units,
     certified by an integer bracket on the root
          285^2 * 1800 <= 37 * 2000^2 < 287^2 * 1800
  [PASS] T4-07   the strongest increment is -1.00 and the weakest -0.58, a
     range of 0.42 log units
          [-100, -58], range 42 centilogs
  [PASS] T4-08   the weakest is bounded away from zero, which is the premise
     Thm 6.7 needs
          sup delta = -0.58 log units

T5  The nonpolar control: methyl moves the other way  (Sch. 6.2)   [EXACT]
  ---------------------------------------------------------------------
  [PASS] T5-01   eight measured methyl increments are available, the
     calculated entry having been excluded in T1
          8 species, 0 calculated
  [PASS] T5-02   every one is positive: a nonpolar arm moves log Kow the
     opposite way to a hydroxyl
          methyl all > 0, hydroxyl all < 0
  [PASS] T5-03   the weakest is exactly +0.280 log units
          1-methylfluoranthene, +28 centilogs
  [PASS] T5-04   the mean is exactly +0.570 log units -- an exact value, not a
      rounded one
          912 half-centilogs / 8 = 114 = 0.570
  [PASS] T5-05   the population variance is exactly 7389/16 centilogs squared,
      so the standard deviation rounds to 0.215 log units
          429^2 * 160000 <= 7389 * 2000^2 < 431^2 * 160000
  [PASS] T5-06   three ring-count classes are present, with means 0.585, 0.525
      and 0.605 log units
          2-ring 58.5, 3-ring 52.5, 4-ring 60.5 centilogs
  [PASS] T5-07   the between-class spread is exactly 0.080 log units
          60.5 - 52.5 = 8 centilogs
  [PASS] T5-08   the spread does not exceed the scatter, and the comparison is
      settled by squaring rather than by a root  (Prop. 7.3)
          8^2 = 64 <= 7389/16 = 461.8 centilogs squared

T6  Claim (A): the ladder does not saturate  (Thm 6.3, Rem. 6.5)   [EXACT]
  ---------------------------------------------------------------------
  [PASS] T6-01   the locant-matched legs are -0.58 and -0.73 log units, both
     installing a hydroxyl ortho to an existing hydroxyl
          phenol->catechol, catechol->pyrogallol
```

```
  [PASS] T6-02   the second is the LARGER in magnitude, by exactly 0.150 log
     units, so no decay with arm count is detected
         73 - 58 = 15 centilogs
89 [PASS] T6-03   the one-sided test passes outright, not merely within
     tolerance: the step grows rather than decaying
         -73 <= -58 with the scatter unused
  [PASS] T6-04   what is supported is that any decay is below the scatter
     0.143, which is a detection limit and not an absence
         Thm 6.3 states a limit; this check asserts the limit, not absence
  [PASS] T6-05   the isomer-averaged one-to-two step is exactly -227/3,
     printed as -0.757
94       2|(-227)(1000) - (-757)(300)| = 50 < 300
  [PASS] T6-06   against it the two-to-three step appears WEAKER by +8/3
     centilogs, printed as +0.027: an apparent saturation
         2|8 * 1000 - 27 * 300| = 200 < 300
  [PASS] T6-07   that artefact is 15.75 times smaller than the locant noise
     the averaging admits -- 'some fifteen times' as printed
         42 / (8/3) = 63/4 = 15.75
99 [PASS] T6-08   so the locant-matched comparison is the admissible one, and
     the averaged comparison is recorded as an error, not used
         the two comparisons disagree in sign, which is the point of Rem. 6.5

T7  Claim (L): locant independence fails, ordering survives   [EXACT]
  ------------------------------------------------------------------
104 [PASS] T7-01   the three constitutional isomers of dihydroxybenzene are at
     0.92, 0.81 and 0.50 -- one core, one arm type, one lab
         ortho, meta, para
  [PASS] T7-02   their spread is exactly 0.42 log units, so delta depends on
     the locant and H_add is false as written
         92 - 50 = 42 centilogs
  [PASS] T7-03   that spread is 12600/211 per cent of one mean arm increment,
     printed as 60 per cent
109      2|12600 - 60 * 211| = 120 < 211
  [PASS] T7-04   the locant effect is comparable to an arm effect and so is a
     real violation rather than rounding
         42 > 35.17: more than half an arm
  [PASS] T7-05   Thm 6.7's premise holds on the data: the supremum of the
     measured increments is -0.58 and so is negative
         sup delta = -58 centilogs
114 [PASS] T7-06   monotonicity holds for all 24 installation orders, each
     reaching the same total 2.95, and is decided by the signs alone with no
     antilog taken
         24 orders, every one strictly rising to +295 centilogs
  [PASS] T7-07   the C_2h inversion centre identifies 3 with 9 and 6 with 12,
     so four locants fall in two orbits  (Sch. 6.9)
         orbits {3,9} and {6,12} partition {3,6,9,12}
  [PASS] T7-08   so at most two values of delta arise on the core, which
     bounds the damage without repairing H_add
119      two parameters, not one and not four

T8  The per-arm enhancement q, by integer antilog bracket  (Thm 7.1)   [EXACT]
  ------------------------------------------------------------------
  [PASS] T8-01   log10 q is exactly 211/300: the mean increment of T4, negated
      and converted from centilogs to log units
124      -(-422/6)/100 = 422/600 = 211/300 log units
  [PASS] T8-02   it is printed as 0.703
         2|211 * 1000 - 703 * 300| = 200 < 300
  [PASS] T8-03   q = 10^(211/300) rounds to 5.05, certified by comparing two
     integers of 902 digits
```

```
          1009^300 <= 10^211 * 200^300 < 1011^300
  [PASS] T8-04   q^4 = 10^(211/75) rounds to 651
          1301^75 <= 10^211 * 2^75 < 1303^75
  [PASS] T8-05   the weakest increment gives q = 10^(29/50), printed 3.80
          759^50 <= 10^29 * 200^50 < 761^50
  [PASS] T8-06   the strongest gives q = 10 exactly, so the upper end of the
     band needs no bracket at all
          the -1.00 increment is one decade on the nose
  [PASS] T8-07   the band on q^4 runs from 10^(58/25), printed 209, to 10^4 =
     10 000 exactly
          417^25 <= 10^58 * 2^25 < 419^25
  [PASS] T8-08   q exceeds unity, which is what H_add requires of it
          log10 q = 211/300 > 0
  [PASS] T8-09   and every q here is a FLOOR and not an estimate, by Prop. 4.4
      under H_ref and H_cav -- neither of which this harness can check
          recorded, not verified: H_cav is named and unmeasured

T9  H_bulk is refuted on the measured set  (Thm 9.1, Prop. 9.3)   [EXACT]
  ------------------------------------------------------------------------
  [PASS] T9-01   nineteen unsubstituted measured competitors are available in
     source R, chrysene excluded
          19 species
  [PASS] T9-02   eleven of them violate H_bulk, every comparison being between
      two integers
          11 violators
  [PASS] T9-03   that is 1100/19 per cent, printed as 58
          2|1100 - 58 * 19| = 4 < 19
  [PASS] T9-04   the extreme case is naphthalene at 3.35
          335 centilogs, 2.51 below chrysene
  [PASS] T9-05   its P_d exceeds P_0 by 10^(251/100), printed as a factor of
     324
          647^100 <= 10^251 * 2^100 < 649^100
  [PASS] T9-06   restricting the competitor class to four-ring-and-larger
     polyhexes leaves two violators, so the rescue fails
          pyrene 5.00 and benzo[c]phenanthrene 5.76
  [PASS] T9-07   the polyhex restriction is proper: nine species carry a five-
     membered ring or a sulfur heterocycle and are excluded before the count
     is taken
          9 excluded from the competitor class
  [PASS] T9-08   benz[a]anthracene is NOT a violator, so the refutation does
     not sweep in the whole class
          591 > 586: one four-ring polyhex is more bulk-retained than chrysene

T10  Chrysene's rank in its own competitor class  (Sch. 9.5)   [EXACT]
  ------------------------------------------------------------------------
  [PASS] T10-01  four measured four-ring polyhexes are present
          benz[a]anthracene, chrysene, benzo[c]phenanthrene, pyrene
  [PASS] T10-02  Article 79 Prop. 4.4 gives the class seven members, so this
     is a subset and the ranking is a ranking within it
          4 of 7 measured
  [PASS] T10-03  chrysene ranks 2 of 4 by log Kow
          rank 2
  [PASS] T10-04  it is within 0.05 of the most bulk-retained and 0.86 above
     the least, in a class spanning 0.91
          5, 86 and 91 centilogs
  [PASS] T10-05  so it sits at the hydrophobic end of its class: what moves it
      out of the bulk is the arming, not bare solubility
          top two of four
  [PASS] T10-06  but it is NOT the maximum, and Sch. 9.5 declines to state it
```

```
      as one
174           benz[a]anthracene at 5.91 is above it
    [PASS] T10-07  the three gaps are consistent: 0.05 + 0.86 = 0.91
           5 + 86 = 91

  T11  The quantified load on the structural filters  (Ruling 9.4(2))   [EXACT]
179   ------------------------------------------------------------------------
    [PASS] T11-01  eleven load ratios are formed, one per violator
           11 ratios
    [PASS] T11-02  each exponent is an integer in thirds of a centilog, so '
       exceeds unity' is decided without any antilog
           exponent = 3(586 - L) - 422
184 [PASS] T11-03  five of the eleven exceed unity: they reach the stock phase
       more readily than a two-armed chrysene does
           5 of 11
    [PASS] T11-04  the threshold is log Kow < 4.4533..., and the nearest species
        on either side are 4.38 and 4.50
           dibenzothiophene inside, anthracene outside
    [PASS] T11-05  the worst is naphthalene at 10^(331/300), printed as a factor
        of 13
189         25^300 <= 10^331 * 2^300 < 27^300
    [PASS] T11-06  so the q^-2 N_Dsp term of Article 79 Prop. 5.3 is understated
       , and the proposition is unavailable for a bulk containing aromatics
       below four rings
           the term is exceeded, not merely approached
    [PASS] T11-07  nothing structural follows: Article 79 Sch. 8.14 routes
       purity through admission and registry, which are predicates and geometry
        and appeal to no partition claim
           recorded, not verified: a claim about the architecture
194
  T12  The benzylic control: class fixed, ring count varied   [EXACT]
    ------------------------------------------------------------------------
    [PASS] T12-01  installing a benzylic hydroxyl on one ring gives -1.55 and on
        three rings -1.44
           toluene->benzyl alcohol; fluorene->9-hydroxyfluorene
199 [PASS] T12-02  the two differ by exactly 0.11 log units
           155 - 144 = 11 centilogs
    [PASS] T12-03  0.11 is within the cross-source scatter 0.215 of T5, and the
       comparison is settled by squaring
           11^2 = 121 <= 7389/16 = 461.8 centilogs squared
    [PASS] T12-04  so a hydrogen-bonding hydroxyl shows no ring-count drift over
        one to three rings  (Prop. 8.2)
204         drift below the scatter it is measured against
    [PASS] T12-05  the three-ring benzylic value departs from the phenol mean by
        exactly -221/3, printed as -0.74
           2|(-221)(100) - (-74)(300)| = 200 < 300
    [PASS] T12-06  the SAME class difference measured at one ring, where no ring
       -count term can contribute, is already -254/3, printed 0.85
           2|254 * 100 - 85 * 300| = 200 < 300
209 [PASS] T12-07  the residue unaccounted for by class is exactly 0.11, the
       drift itself and nothing more  (Cor. 8.3)
           254/3 - 221/3 = 11 centilogs
    [PASS] T12-08  the three-ring benzylic value is nearer the one-ring BENZYLIC
        value than the phenol mean, which is what makes it a class effect and
       not a ring-count effect
           11 < 73.67
    [PASS] T12-09  misread as a phenolic increment it would give q = 10^(36/25),
        printed 27.5, in place of 5.05
214         549^25 <= 10^36 * 20^25 < 551^25
```

```
  [PASS] T12-10  and q^4 = 5.75 x 10^5 in place of 651, the mantissa bracketed
       after the decade is removed
           1149^25 <= 10^19 * 200^25 < 1151^25
  [PASS] T12-11  and it would drop the count behind Ruling 9.4(2) from five of
       eleven to none, since no violator lies below 2.98
           the misread q^-2 threshold is log Kow < 2.98; the minimum is 3.35
  [PASS] T12-12  it cannot be taken that way: C9 of fluorene is an sp3 bridge
     and Article 79's locants 3,6,9,12 are aromatic CH positions, so the
     datum measures a different arm
           recorded, not verified: a structural claim, not arithmetic

T13  Arm-type incommensurability  (Prop. 10.1 -- ESTIMATED INPUTS)   [EXACT]
  ------------------------------------------------------------------------
  [PASS] T13-01  five ionization shifts, each an exact integer difference
           shifts [-254, -242, -244, -266, -268]
  [PASS] T13-02  they sum to -1274, so the mean is exactly -1274/5 and is
     printed as -2.55
           2|(-1274)(100) - (-255)(500)| = 200 < 500
  [PASS] T13-03  the population variance is exactly 2904/25 centilogs squared,
       so the standard deviation rounds to 0.11 log units
           21^2 * 31250 <= 363 * 200^2 < 23^2 * 31250
  [PASS] T13-04  the shift is 3822/1055 times the mean hydroxyl increment,
     printed as 3.6
           2|3822 * 10 - 36 * 1055| = 480 < 1055
  [PASS] T13-05  it exceeds twice that increment, decided exactly without
     forming the quotient
           3822 > 2110
  [PASS] T13-06  q_ion = 10^(637/250) rounds to 353
           705^250 <= 10^637 * 2^250 < 707^250
  [PASS] T13-07  the two rungs differ by 10^(2767/1500), printed as a factor
     of 70: a ladder whose rungs differ by that much is not a geometric
     ladder
           139^1500 <= 10^2767 * 2^1500 < 141^1500
  [PASS] T13-08  the mixed four-arm aggregate is 10^(4877/750); its mantissa
     3.18 is bracketed after six decades are removed
           635^750 <= 10^377 * 200^750 < 637^750
  [PASS] T13-09  it exceeds the single-type ladder q^4 by 10^(2767/750),
     printed as a factor of 4890
           9779^750 <= 10^2767 * 2^750 < 9781^750
  [PASS] T13-10  the mixed aggregate exceeds q^4, decided in the exponents
     alone with no antilog
           9754 > 4220, over the common denominator 1500
  [PASS] T13-11  and it is a LOWER bound, not an estimate: q_ion carries only
     the ionization shift and omits the installation increment of the arm
     that ionizes
           recorded, not verified: Prop. 10.1 does not measure it
  [PASS] T13-12  a unit preferring water by some five log units does not form
     a distinct phase but dissolves, so monotone partition is not monotone
     separation  (Rem. 10.4)
           the last two arms are worth 5.1 log units between them

T14  Integer certificates for every printed figure   [EXACT]
  ------------------------------------------------------------------------
  [PASS] T14-01  ten printed ratios each round as the article prints them, by
     the bound 2|n s - r d| < d
           10 of 10
  [PASS] T14-02  three standard deviations each round as printed, by an
     integer bracket on the root; no root is formed
           3 of 3
```

```
  [PASS] T14-03  twelve base-ten antilogarithms each round as printed, by an
     integer bracket; no logarithm or power of ten is evaluated in floating
     point
         12 of 12
  [PASS] T14-04  the two exact antilogs need no bracket: q at the strongest
     arm is 10 and its fourth power is 10 000
         the -1.00 increment is one decade exactly
  [PASS] T14-05  three figures are exact rather than rounded, and are asserted
      as equalities: the methyl mean 0.570, the between-class spread 0.080,
     and the locant spread 0.42
         no rounding certificate is needed for an exact value
  [PASS] T14-06  no check in this file compares floating-point numbers, and
     the float type is never constructed
         every comparison is between int or Fraction
  [PASS] T14-07  the bracket devices reject a wrong figure: each returns False
      when the target is moved by one unit in the last place
         the certificates are not vacuous
  [PASS] T14-08  and they reject a bracket that is merely close: q does not
     round to 5.04 or 5.06
         the bracket is tight to the printed place

T15  Scope, obligations and the review record   [EXACT]
  ----------------------------------------------------------------------
  [PASS] T15-01  the article records eight obligations, and they are carried
     in this file rather than only in the prose
         8 obligations
  [PASS] T15-02  six withdrawals and corrections are traced, two of them
     against results of this paper's own making
         W1-W6
  [PASS] T15-03  the review record closes: six findings accepted, two accepted
      in another form, one rebutted
         9 findings
  [PASS] T15-04  NO value in this article is a theorem of the corpus: every [M
     ] datum enters under H_data, a premise about the world
         the premise is named, not proved
  [PASS] T15-05  this harness therefore certifies arithmetic and rounding, and
      does NOT certify any measurement
         the boundary, stated rather than buried
  [PASS] T15-06  the assertions above encode the article's conclusions, so the
      file stops passing when those conclusions stop holding -- which makes
     it a regression test on the conclusions and not an independent test of
     them
         T9 fails if H_bulk is not refuted; T12 if the drift is not flat
  [PASS] T15-07  three checks in this file are marked recorded-not-verified
     and assert nothing: T8-09, T11-07 and T12-12
         a structural or unmeasured claim is named, never certified
  [PASS] T15-08  the harness uses the Python standard library only and touches
      no network
         fractions, itertools, sys
----------------------------------------------------------------------
  blocks            : 15
  checks            : 120
  passed            : 120
  failed            : 0
  floating point    : none; every check is int or Fraction
  largest integer   : 3224 digits (the rung-ratio bracket)
----------------------------------------------------------------------

ALL 120 CHECKS PASSED.
```

```
What this certifies: that the arithmetic Article 82 performs on
its tabulated values is exact, and that every figure it prints
is the correctly rounded result of that arithmetic.

What it does not certify: the values themselves.  Those enter
under H_data (Hyp. 2.4), a premise about the world, and no
harness can reach them.
```

Lean 4 Proof Summary: `acf-82-proof.lean`

Lean Kernel Certification

The file `acf-82-proof.lean` contains 109 theorems in the namespace `ChryseneFormalism.Article82`, checked by the Lean 4.12.0 kernel. Every theorem is discharged by `decide`. No Mathlib, no `sorry`, no `native_decide`.
An `#print axioms` audit over all 109 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-82-proof.tex`.

Elaboration takes about sixteen seconds, and almost all of it is the 16 807-case exhaustion of Proposition 3.1. The antilogarithm brackets — the file's headline device, and the reason it needs no real-number library — cost under a fifth of a second *in total*, the 3224-digit rung-ratio comparison included.

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 the *arithmetic*: every increment, mean, spread, variance, count and ratio as an equation or inequality between naturals, and every printed figure as a bracket that fails if the figure moves by one unit in its last place. It is the stricter instrument, and where it applies it applies absolutely.
The harness does three things the Lean file does not. It *derives* the quantities rather than restating them — the variance of block T4 is computed from the six increments, not asserted to be 1850/9 — so a mistranscribed datum changes the figure and fails the certificate, where in Lean it would change both sides together. It carries the provenance discipline of Convention 2.1 as executable structure, with named column indices and type assertions, so that the confusion between the source column and the measured flag cannot recur. And it holds the *obligations* and the review record, so the eight things the paper does not establish travel with the numbers rather than with the prose.

Independent cross-check against the published figures

Article 82 emits its own figures from a separate generator, `verify_a82.py`, which computes in floating point and writes 78 LaTeX macros. Those 78 macros are what the article prints. Each one was checked against the exact-integer certificate this harness establishes for it: 20 counts and 31 exact values — the latter including five species names and compound lists — by equality, 10 ratios by the rounding bound, 4 standard deviations by the squared bracket, and 13 antilogarithms by the integer power bracket. **All 78 agree, with no disagreement in any place.**
This is the check that matters for a concordance paper. The two computations share no arithmetic — one is floating point over `math.log10`, the other exact integers over `Fraction` and `pow` — so agreement in all 78 places is evidence that the published figures are right, and not merely that they are self-consistent.