ACF 83 — Verification Dossier

The Transformed Product

Membrane Invariants Measured against the Open Literature, and the Lipid the Patent Route Actually Makes

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

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

Verification script: `verify_A83.py` Lean proof: `acf-83-proof.lean`

Certification Summary

Item	Value
Script	`verify_A83.py`
Total checks	171
Passed	**171**
Failed	**0**
Exit code	0
Blocks	15 (T1–T15)
Blocks using floating point	**0**
Float literals in the source	**0**
Dependencies	Python standard library only
Lean proof file	`acf-83-proof.lean`
Lean theorems	74
Goals re-evaluated independently	**74**
`sorry`	0
`native_decide`	0
Kernel run	**not yet performed**

The Lean file has not been through a kernel

This dossier reports one figure its Article 82 counterpart does not, and the difference is material. `acf-83-proof.lean` was written in an environment with no Lean toolchain. Its *arithmetic* has been checked — all 74 of its goals were re-evaluated in exact integer arithmetic by `lean_certificates.py`, using functions written to match the Lean definitions term for term, and all 74 hold. Its *syntax* has not been checked, because checking it requires Lean.

So the correct reading of the Lean figures above is: the propositions are true, and whether the file elaborates is unknown. A reader who takes "74 theorems" here as equivalent to the kernel-checked count in the Article 82 dossier has read more into it than is there. Run `lean acf-83-proof.lean` to close the gap; the companion proof dossier records what to expect.

This is stated prominently rather than in a footnote because the corpus has twice been caught by an artifact that looked verified and was not: Article 84's section 11 passed every assertion while computing on the one family its own Theorem 6.1 excludes, and its check C1 was claimed as enforced while being implemented nowhere. An unchecked proof file announced as

a checked one would be the same error a third time.

What a concordance harness can and cannot do

Article 83 holds corpus hypotheses against numbers measured by other people. Those numbers enter under hypothesis H_{data}, which is a premise about the world. *No harness can certify that DOPC's hydrophobic thickness is 28.8 Å.*
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_A83.py` does. 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. Checks marked STRUCTURAL assert something no arithmetic settles and are recorded, not verified. Checks marked RECORDED assert a count of obligations and nothing about their content.

Exact throughout: the unit does the work

No block of this harness is labelled floating, and the source contains no floating-point literal at all — which an abstract syntax tree walk over the file confirms rather than an inspection.
Every structural datum in Article 83 is quoted to one decimal place, so the natural unit is the *deci-angstrom*, and in that unit every thickness and every area is an integer. Permeabilities are quoted as base-ten logarithms to two decimals, so the natural unit there is the *centilog*. Capacitance is quoted to four decimals and is carried in units of $10^{-4}\,\mu\text{F/cm}^2$. Differences, sums, ratios and percentages of integers are exact rationals, which `fractions.Fraction` carries without loss.

Four devices replace a statement about a rounded or irrational quantity by an inequality between exact integers:

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	$(2r{-}1)^d \le 10^n (2s)^d < (2r{+}1)^d$
`admits(x,lo,hi)`	x lies in a band fixed in advance	$lo \le x \le hi$

The fourth is this harness's characteristic device and has no counterpart in the Article 82 dossier. It is the concordance primitive of the present article: an invariant is fixed as a closed interval *before* any candidate is scored, and a candidate is admitted exactly when it lies in that interval. It returns a pair, admitted and strictly interior, because an admission that lands on an endpoint is a different thing from one that does not, and the article says which of its admissions are which.

Block T14 checks that none of the four is vacuous: each returns `False` when its target is moved by one unit in the last place, and `admits` additionally distinguishes an endpoint from the interior.

Block Index

Block	Topic	Checks
T1	The measured record, in exact units (TRANSCRIPTION)	8
T2	The vessel graph: order, size, bond classes	9
T3	The cleavage family and the three routes	5
T4	The dimension invariant: the band, and route A admitted	8
T5	Route P is not admitted: the single-chain extrapolation	7
T6	The head-group area and the unit count it corrects	7
T7	The reduction ledger: oxidation numbers and the deficit	13
T8	The reductant demanded over one forcing period	6
T9	The permeability barrier, in decades	10
T10	The chemiosmotic invariant and the two shortfalls	12
T11	Genome retention and the refusal of admission	5
T12	The sequential model and the activation gap	9
T13	Negatives, obligations and the review record (RECORDED)	6
T14	The devices reject a figure moved by one unit	10
T15	Cross-check against the article's own macro generator	56
	Total	**171**

Scope of Verification

What is verified.

- That every structural datum is an exact integer number of deci-angstrom and every permeability an exact integer number of centilogs, which is what licenses exact arithmetic for the whole paper; and that the two structural sources overlap on one lipid at one temperature, which is what licenses carrying a row from the second into a band built from the first.
- That the band of the dimension invariant is taken over the rows the article *displays*, that the displayed set is a selection of one row per saturated lipid plus the comparand, that the branched lipid is excluded from it, and that two measured rows lie below the band and are not displayed. The selection is a decision of the article and is checked as one.
- That the route-A comparand is the di-C18:1 tail exactly, that its thickness lies in the band, and that it lies *strictly* inside rather than on an edge.
- That the route-P extrapolation does not lie in the band, that it is refused from below and not from above, and that the shortfall is exactly 47 deci-angstrom.
- That the two figures the article prints as 4.7 per cent — the head-group area error and the unit-count drop — are **different numbers** that happen to round alike, so that a reader

cannot take the coincidence for an identity.

- The oxidation ledger in full: the three oxidation numbers as magnitudes with the equations that fix their signs, the per-unit and per-dimer demands, both deficits against Article 62's charged count, and the per-period figures with the check that they carry the per-unit ratio.
- That the permeability series is monotone in chain length for each of proton, potassium and water separately, and that the barrier is exactly 964 centilogs.
- That the capacitance error is below one per cent of the capacitance, so that it does not reach the conclusion; and that solvent-free films are *nearly* twice the decane films and not twice, which an earlier draft of this harness overstated.
- That every one of the 47 cross-checked macros emitted by the article's own generator agrees exactly with the certificate established here, and that the rounded figures agree by the device appropriate to each.

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

Two findings this harness surfaced, recorded rather than smoothed.

- **The printed chemiosmotic shortfall is not the ratio of the printed figures.** Article 83 prints $\tau_{RC} = 4.1$ s and a shortfall of 10,530. A reader who divides 43,200 by 4.1 obtains 10,537. Both printed figures are correct: the shortfall was computed from an unrounded $\tau_{RC} \approx 4.1025$ s and the time constant was printed to one decimal. What fails is their *composition*. Checks T10-04 and T10-05, marked REPRODUCIBILITY, assert the gap rather than hiding it behind a looser tolerance, and T15-56 certifies instead that the printed pair is jointly consistent with one unrounded time constant, which is the strongest true statement available.
- **The ledger ratio sits exactly on a rounding boundary.** $52/32 = 13/8 = 1.625$, the midpoint between 1.62 and 1.63. The rounding device of this harness declines an exact half by design rather than adopting a tie-breaking convention silently, so checks T7-11 to T7-13 certify the boundary condition itself and record that the printed 1.62 is the half-to-even value.

Three further limits.

- **Two checks are structural, not arithmetic.** That retention and the refusal of admission are asserted of the *same* membrane, and that Article 66's pore is therefore necessary rather than merely available, are claims no arithmetic settles. They are marked STRUCTURAL and assert nothing.
- **The register is a count.** Block T13 certifies how many negatives, obligations and hypotheses the article carries, and nothing about their content.
- **The harness is a regression test on the conclusions.** Its assertions encode them: T5 fails if route P is *not* refused, T10 if the chemiosmotic invariant does *not* fail. That is intended — 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_A83.py`

verify_A83.py — complete source

```
#!/usr/bin/env python3
# ======================================================================
#  verify_A83.py
#  Exact-arithmetic verification harness for
#    Annals of the Chrysene Formalism, Article 83,
#    "The Transformed Product: Membrane Invariants Measured against the
#     Open Literature, and the Lipid the Patent Route Actually Makes"
#
#  Companion to the Lean 4 proof file  acf-83-proof.lean
#  and to the article's own macro generator  verify_a83.py.
#
#  Run:   python3 verify_A83.py        summary only
#         python3 verify_A83.py -v     every check, with its value
#
#  Exit code 0 if and only if every check passes.
#
#  ------------------------------------------------------------------
#  DISCIPLINE
#
#  Standard library only.  No floating point is formed anywhere in this
#  file: every datum is carried as an exact integer in a named unit, and
#  every derived quantity as a Fraction.  Three irrational quantities
#  occur in Article 83 -- a square root, a base-ten antilogarithm, and a
#  ratio of antilogarithms -- and none of them is formed.  Each is
#  certified by an inequality between exact integers.
#
#  The units that make this possible:
#      dA       deci-angstrom          thicknesses, 1 decimal place
#      dA2      deci-square-angstrom   head-group areas, 1 decimal place
#      clog     centilog               log10 permeabilities, 2 decimals
#      tcap     1e-4 uF/cm^2           specific capacitance, 4 decimals
#  In each unit every imported datum is an integer, which is what
#  licenses exact arithmetic for the whole paper.
#
#  WHAT THIS HARNESS CANNOT DO
#
#  Article 83 is a concordance paper.  Its membrane data were measured
#  by other people and enter under hypothesis H_data.  No harness can
#  certify that DOPC's hydrophobic thickness is 28.8 A.  What a harness
#  can certify is 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 this file does.
#  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.
# ======================================================================

import os
import re
import sys
from fractions import Fraction as F

HERE = os.path.dirname(os.path.abspath(__file__))
VERBOSE = "-v" in sys.argv or "--verbose" in sys.argv
```

```
CHECKS = []          # (tag, description, ok, shown, kind)
_BLOCK = [None]


def block(tag, title):
    _BLOCK[0] = (tag, title)
    if VERBOSE:
        print()
        print("=" * 72)
        print("  %-5s %s" % (tag, title))
        print("=" * 72)


def ck(num, desc, ok, shown="", kind=""):
    tag = "%s-%02d" % (_BLOCK[0][0], num)
    CHECKS.append((tag, desc, bool(ok), str(shown), kind))
    if VERBOSE:
        mark = "PASS" if ok else "FAIL"
        suffix = ("   [%s]" % kind) if kind else ""
        print("  %-8s %s  %-46s %s%s"
              % (tag, mark, desc[:46], shown, suffix))
    return bool(ok)


# ---------------------------------------------------------------------
#  DEVICES.  Each replaces a statement about an irrational or rounded
#  quantity by an inequality between exact integers.  Each is checked
#  for non-vacuity in block T17: moved by one unit in the last place,
#  it must return False.
# ---------------------------------------------------------------------
def rounds_to(n, d, r, s):
    """n/d rounds to r/s, by the integer relation 2|ns - rd| < d.

    Half-up and half-down agree except exactly at the boundary, which
    this writes as a strict inequality, so a value sitting precisely
    half-way fails rather than silently taking a convention."""
    assert d > 0 and s > 0
    return 2 * abs(n * s - r * d) < d


def sqrt_rounds(n, d, r, s):
    """sqrt(n/d) rounds to r/s, by  (2r-1)^2 d <= n (2s)^2 < (2r+1)^2 d.

    The root is never formed; it is squared against."""
    assert d > 0 and s > 0 and n >= 0 and r >= 0
    lo = (2 * r - 1) ** 2 * d
    mid = n * (2 * s) ** 2
    hi = (2 * r + 1) ** 2 * d
    return lo <= mid < hi


def antilog_rounds(n, d, r, s):
    """10^(n/d) rounds to r/s, by  (2r-1)^d <= 10^n (2s)^d < (2r+1)^d.

    The bracket (2r-1)/2s <= 10^(n/d) < (2r+1)/2s is raised to the d,
    which clears the root and leaves two exact integers.  They are
    large; Python's integers are arbitrary precision, so the comparison
    is exact."""
```

```
    assert d > 0 and s > 0 and n >= 0 and r >= 0
    return (2 * r - 1) ** d <= 10 ** n * (2 * s) ** d < (2 * r + 1) ** d


def admits(x, lo, hi):
    """The concordance primitive of this article: an invariant is fixed
    as a closed interval BEFORE any candidate is scored, and a candidate
    is admitted exactly when it lies in that interval.

    All three arguments are exact.  Returns the pair (admitted,
    strictly interior), because an admission that lands on an endpoint
    is a different thing from one that does not, and Article 83 says
    which of its admissions are which."""
    return (lo <= x <= hi, lo < x < hi)


# ======================================================================
#  T1   THE MEASURED RECORD, IN EXACT UNITS                  [imported]
# ======================================================================
block("T1", "The measured record, in exact units  (TRANSCRIPTION)")

#  Kucerka 2011, joint SAXS/SANS.  (lipid, nC, unsat, T_C, area_dA2,
#  twoDC_dA, DHH_dA).  Every entry an exact integer in its unit.
K11 = [
    ("DLPC",   12, 0, 20, 596, 219, 330),
    ("DLPC",   12, 0, 30, 608, 217, 326),
    ("DLPC",   12, 0, 50, 648, 208, 310),
    ("DMPC",   14, 0, 30, 599, 257, 367),
    ("DMPC",   14, 0, 50, 633, 248, 352),
    ("DPPC",   16, 0, 50, 631, 285, 390),
    ("DPPC",   16, 0, 60, 650, 279, 381),
    ("DSPC",   18, 0, 60, 638, 319, 422),
    ("POPC",   17, 1, 30, 643, 288, 391),
    ("SOPC",   18, 1, 30, 655, 299, 400),
    ("DPhyPC", 16, 0, 30, 806, 272, 354),
]
#  Kucerka 2008, same laboratory, same method.  DOPC is the di-C18:1
#  cis-Delta-9 comparand: the exact tail of the route-A product.
K08 = [
    ("DPPC", 16, 0, 50, 631, 284, 380, 390),
    ("DOPC", 18, 1, 30, 674, 288, 367, 387),
]
#  Paula 1996, proton/potassium/water permeation, in centilogs.
#  (chain_nC, logPH, logPK, logPW)
P96 = [
    (14, -189, -982, -161),
    (16, -274, -1103, -171),
    (18, -329, -1146, -182),
    (20, -380, -1162, -195),
    (22, -410, -1166, -215),
    (24, -431, -1177, -230),
]
#  White 1978, specific capacitance, in units of 1e-4 uF/cm^2.
CAP_FREE, CAP_FREE_ERR = 7771, 48
CAP_DECANE = [3968, 4348]

ck(1, "Kucerka 2011 rows are exact in deci-angstrom",
   all(isinstance(r[5], int) and isinstance(r[6], int) for r in K11),
   "%d rows" % len(K11), "TRANSCRIPTION")
```

```
ck(2, "Kucerka 2008 rows are exact in deci-angstrom",
   all(isinstance(r[5], int) for r in K08), "%d rows" % len(K08),
   "TRANSCRIPTION")
ck(3, "Paula 1996 logs are exact in centilogs",
   all(all(isinstance(x, int) for x in r[1:]) for r in P96),
   "%d rows" % len(P96), "TRANSCRIPTION")
ck(4, "capacitance exact in 1e-4 uF/cm^2",
   isinstance(CAP_FREE, int) and isinstance(CAP_FREE_ERR, int),
   "%d +- %d" % (CAP_FREE, CAP_FREE_ERR), "TRANSCRIPTION")

#  The two laboratories overlap on one lipid at one temperature.  That
#  overlap is what licenses carrying the 2008 DOPC row into a band built
#  from the 2011 table, and it is checked rather than assumed.
dppc11 = [r for r in K11 if r[0] == "DPPC" and r[3] == 50][0]
dppc08 = [r for r in K08 if r[0] == "DPPC" and r[3] == 50][0]
ck(5, "the two sources agree on DPPC area at 50 C",
   dppc11[4] == dppc08[4], "%d = %d dA2" % (dppc11[4], dppc08[4]))
ck(6, "and their DPPC thicknesses differ by one unit in the last place",
   abs(dppc11[5] - dppc08[5]) == 1,
   "%d vs %d dA" % (dppc11[5], dppc08[5]))

#  DPhyPC is branched.  It is displayed but must not set the band.
branched = [r for r in K11 if r[0] == "DPhyPC"]
ck(7, "exactly one displayed lipid is branched",
   len(branched) == 1, branched[0][0])
ck(8, "its area is the largest in the table, which is why",
   branched[0][4] == max(r[4] for r in K11),
   "%d dA2" % branched[0][4])


# ======================================================================
#  T2   THE VESSEL GRAPH                                      [computed]
# ======================================================================
block("T2", "The vessel graph: order, size, bond classes")

RINGS = [["1", "2", "3", "4", "4a", "12a"],
         ["4a", "4b", "10b", "11", "12", "12a"],
         ["4b", "5", "6", "6a", "10a", "10b"],
         ["6a", "7", "8", "9", "10", "10a"]]
E = sorted({tuple(sorted((c[i], c[(i + 1) % 6])))
            for c in RINGS for i in range(6)})
V = sorted({v for e in E for v in e})
FUS = {"4a", "4b", "6a", "10a", "10b", "12a"}
CHORDS = {tuple(sorted(x)) for x in
          [("4a", "12a"), ("4b", "10b"), ("6a", "10a")]}
FF = [e for e in E if e[0] in FUS and e[1] in FUS]
FH = [e for e in E if (e[0] in FUS) != (e[1] in FUS)]
HH = [e for e in E if e[0] not in FUS and e[1] not in FUS]

ck(1, "order of the skeleton", len(V) == 18, len(V))
ck(2, "size of the skeleton", len(E) == 21, len(E))
ck(3, "cyclomatic number", len(E) - len(V) + 1 == 4, len(E) - len(V) + 1)
ck(4, "ring-fusion carbons", len(FUS) == 6, len(FUS))
ck(5, "fusion-fusion bonds", len(FF) == 5, len(FF))
ck(6, "mixed bonds", len(FH) == 8, len(FH))
ck(7, "hydrogen-bearing bonds", len(HH) == 8, len(HH))
ck(8, "classes partition the edge set",
   len(FF) + len(FH) + len(HH) == len(E), len(E))
ck(9, "the three chords are a PROPER subset of the five ff bonds",
```

```
    CHORDS < set(FF), "%d < %d" % (len(CHORDS), len(FF)))


# ====================================================================
#  T3   THE CLEAVAGE FAMILY AND THE THREE ROUTES            [computed]
# ====================================================================
block("T3", "The cleavage family and the three routes")

#  Route stoichiometry, per chrysene unit.  CORE is the framework
#  carbon count; a route is named by (lipids per unit, tails per lipid,
#  carbons per tail).
CORE = 18
ROUTES = {"A": (F(1, 2), 2, 18),
          "D": (F(1, 1), 2, 9),
          "P": (F(2, 1), 1, 9)}

ck(1, "route A conserves the framework carbons",
   ROUTES["A"][0] * ROUTES["A"][1] * ROUTES["A"][2] == CORE,
   "%s x %d x %d = %d" % (ROUTES["A"][0], ROUTES["A"][1],
                          ROUTES["A"][2], CORE))
ck(2, "route P conserves them too",
   ROUTES["P"][0] * ROUTES["P"][1] * ROUTES["P"][2] == CORE,
   "%s x %d x %d = %d" % (ROUTES["P"][0], ROUTES["P"][1],
                          ROUTES["P"][2], CORE))
ck(3, "the two routes differ fourfold in lipid count",
   ROUTES["P"][0] / ROUTES["A"][0] == 4,
   "%s" % (ROUTES["P"][0] / ROUTES["A"][0]))
ck(4, "every route is carbon-conserving",
   all(l * t * c == CORE for l, t, c in ROUTES.values()), "3 routes")
ck(5, "route A alone is two-tailed at C18",
   [k for k, (l, t, c) in ROUTES.items() if t == 2 and c == 18] == ["A"],
   "A")


# ====================================================================
#  T4   THE DIMENSION INVARIANT: THE BAND, FIXED IN ADVANCE
# ====================================================================
block("T4", "The dimension invariant: the band, and route A admitted")

#  The band is built from the rows the article DISPLAYS, before any
#  candidate is scored.  This is the concordance discipline of
#  Article 82 carried forward: the interval is fixed first.
#
#  The displayed set is a SELECTION, not the whole table: one row per
#  saturated lipid at one temperature, plus the comparand.  That
#  selection is a decision of the article and is checked here as one,
#  because a band taken over the whole table would be wider at the
#  bottom and would admit more.
DISPLAYED = [("DLPC", 20), ("DMPC", 30), ("DPPC", 50), ("DSPC", 60)]
BAND_ROWS = [r[5] for r in K11
             if (r[0], r[3]) in DISPLAYED] + [r[5] for r in K08
                                              if r[0] == "DOPC"]
BAND_LO, BAND_HI = min(BAND_ROWS), max(BAND_ROWS)

ck(1, "the displayed set is one row per saturated lipid, plus one",
   len(BAND_ROWS) == len(DISPLAYED) + 1, "%d rows" % len(BAND_ROWS))
ck(2, "band lower edge, deci-angstrom", BAND_LO == 219, BAND_LO)
ck(3, "band upper edge, deci-angstrom", BAND_HI == 319, BAND_HI)
ck(4, "the branched lipid is NOT displayed, and would not widen it",
```

```
    not any(r[0] == "DPhyPC" for r in K11 if (r[0], r[3]) in DISPLAYED)
    and 272 < BAND_HI, "DPhyPC excluded")
ck(5, "two measured rows lie below the band and are not displayed",
    len([r for r in K11 if r[5] < BAND_LO]) == 2,
    "DLPC at 30 and 50 C")

#  Route A's tail is di-C18:1 cis-Delta-9.  DOPC is that tail exactly.
DOPC = [r for r in K08 if r[0] == "DOPC"][0]
TWO_DC_A = DOPC[5]
adm, interior = admits(TWO_DC_A, 219, 319)
ck(6, "route A: DOPC 2D_C lies in the band", adm,
    "%d in [219, 319]" % TWO_DC_A)
ck(7, "and lies strictly inside it, not on an edge", interior,
    "219 < %d < 319" % TWO_DC_A)
ck(8, "the comparand is the route-A tail exactly, C18:1",
    DOPC[1] == 18 and DOPC[2] == 1, "C%d:%d" % (DOPC[1], DOPC[2]))


# ======================================================================
#  T5   ROUTE P IS NOT ADMITTED                                [computed]
# ======================================================================
block("T5", "Route P is not admitted: the single-chain extrapolation")

#  The C9 single-chain prediction, in deci-angstrom.
PRED_C9 = 172
adm9, _ = admits(PRED_C9, 219, 319)
ck(1, "route P: the C9 extrapolation", True, "%d dA" % PRED_C9)
ck(2, "it does NOT lie in the band", not adm9,
    "%d not in [219, 319]" % PRED_C9)
ck(3, "it falls short of the lower edge, not over the upper",
    PRED_C9 < 219, "%d < 219" % PRED_C9)
ck(4, "the shortfall, in deci-angstrom", 219 - PRED_C9 == 47,
    "%d dA" % (219 - PRED_C9))
ck(5, "which is 4.7 angstrom",
    rounds_to(219 - PRED_C9, 10, 47, 10), "4.7 A")

#  The refusal is one-sided, and the article says so.  A candidate
#  thicker than the band would be refused for a different reason.
ck(6, "the dimension invariant refuses route P from below only",
    PRED_C9 < 219 and not (PRED_C9 > 319), "below")
ck(7, "route A and route P fall on opposite sides of admission",
    admits(TWO_DC_A, 219, 319)[0] and not admits(PRED_C9, 219, 319)[0],
    "A in, P out")


# ======================================================================
#  T6   THE HEAD-GROUP AREA, AND THE LEDGER IT MOVES
# ======================================================================
block("T6", "The head-group area and the unit count it corrects")

AREA_MEAS = DOPC[4]          # 674 dA2, measured
AREA_PRED = 682              # dA2, the article's predicted value
AREA_A62 = 650               # dA2, 0.65 nm^2 as used by Article 62

ck(1, "Article 62 used 0.65 nm^2", AREA_A62 == 650, "650 dA2")
ck(2, "the predicted area exceeds it", AREA_PRED > AREA_A62,
    "%d > %d" % (AREA_PRED, AREA_A62))
ck(3, "the relative error rounds to 4.7 per cent",
    rounds_to((AREA_PRED - AREA_A62) * 100, AREA_PRED, 47, 10),
```

```
    "%d/%d" % (AREA_PRED - AREA_A62, AREA_PRED))

UNITS_A62, UNITS_CORR = 13146, 12530
ck(4, "the unit count falls on correction",
   UNITS_CORR < UNITS_A62, "%d -> %d" % (UNITS_A62, UNITS_CORR))
ck(5, "the drop also rounds to 4.7 per cent",
   rounds_to((UNITS_A62 - UNITS_CORR) * 100, UNITS_A62, 47, 10),
   "%d/%d" % (UNITS_A62 - UNITS_CORR, UNITS_A62))

#  The two figures both print as 4.7 and are NOT the same number.  A
#  reader who takes one for the other has mistaken a coincidence of
#  rounding for an identity, and the harness says so explicitly.
ck(6, "the two 4.7 per cent figures are DIFFERENT numbers",
   F(AREA_PRED - AREA_A62, AREA_PRED) != F(UNITS_A62 - UNITS_CORR,
                                          UNITS_A62),
   "32/682 != 616/13146")
ck(7, "both nevertheless round to the same printed figure",
   rounds_to((AREA_PRED - AREA_A62) * 100, AREA_PRED, 47, 10) and
   rounds_to((UNITS_A62 - UNITS_CORR) * 100, UNITS_A62, 47, 10),
   "both 4.7")


# ======================================================================
#  T7   THE REDUCTION LEDGER, AUDITED                          [computed]
# ======================================================================
block("T7", "The reduction ledger: oxidation numbers and the deficit")


def oxidation_sum(carbons):
    """ON = sum over skeleton carbons of (bonds to O) - (bonds to H).
    A sum of per-atom quantities, hence a function of the state alone,
    hence a demand that depends only on the endpoints of a route."""
    assert len(carbons) == CORE
    return sum(o - h for h, o in carbons)


SUB = [(1, 0)] * 10 + [(0, 1)] * 2 + [(0, 0)] * 6
ETHER = [(2, 1)] + [(2, 0)] * 7 + [(1, 0)] * 2 + [(2, 0)] * 7 + [(3, 0)]
ESTER = [(0, 3)] + [(2, 0)] * 7 + [(1, 0)] * 2 + [(2, 0)] * 7 + [(3, 0)]

ON_SUB, ON_ETH, ON_EST = (oxidation_sum(SUB), oxidation_sum(ETHER),
                          oxidation_sum(ESTER))
ck(1, "substrate oxidation number", ON_SUB == -8, ON_SUB)
ck(2, "ether chain oxidation number", ON_ETH == -34, ON_ETH)
ck(3, "ester chain oxidation number", ON_EST == -30, ON_EST)

RED_ETH, RED_EST = ON_SUB - ON_ETH, ON_SUB - ON_EST
ck(4, "reducing equivalents per unit, ether", RED_ETH == 26, RED_ETH)
ck(5, "reducing equivalents per unit, ester", RED_EST == 22, RED_EST)
ck(6, "per dimer, ether", 2 * RED_ETH == 52, 2 * RED_ETH)
ck(7, "per dimer, ester", 2 * RED_EST == 44, 2 * RED_EST)

A62_CHARGED = 32
ck(8, "Article 62 counted 32 charged", A62_CHARGED == 32, A62_CHARGED)
ck(9, "ether deficit", 2 * RED_ETH - A62_CHARGED == 20,
   2 * RED_ETH - A62_CHARGED)
ck(10, "ester deficit", 2 * RED_EST - A62_CHARGED == 12,
    2 * RED_EST - A62_CHARGED)
#  52/32 = 13/8 = 1.625 EXACTLY, which is the midpoint between 1.62
```

```
#  and 1.63.  The rounding device of this file rejects an exact half by
#  design, rather than silently adopting a convention, so the check is
#  that the figure sits ON the boundary and that the article's printed
#  1.62 is the round-half-to-even value.
ck(11, "the ledger ratio is exactly 13/8",
    F(2 * RED_ETH, A62_CHARGED) == F(13, 8), "52/32 = 13/8")
ck(12, "which is exactly half-way between 1.62 and 1.63",
    2 * abs(2 * RED_ETH * 100 - 162 * A62_CHARGED) == A62_CHARGED,
    "an exact half")
ck(13, "so the device declines it, and the tie is broken to even",
    not rounds_to(2 * RED_ETH * 100, A62_CHARGED, 162, 100)
    and not rounds_to(2 * RED_ETH * 100, A62_CHARGED, 163, 100),
    "1.62 by half-to-even", "CONVENTION")


# ======================================================================
#  T8   THE REDUCTANT OVER A PERIOD                          [computed]
# ======================================================================
block("T8", "The reductant demanded over one forcing period")

PER_ETH, PER_EST, PER_A62 = 325780, 275660, 200480
ck(1, "ether branch, per period", PER_ETH == 325780, PER_ETH)
ck(2, "ester branch, per period", PER_EST == 275660, PER_EST)
ck(3, "Article 62's figure", PER_A62 == 200480, PER_A62)
ck(4, "the gap Article 83 opens", PER_ETH - PER_A62 == 125300,
   PER_ETH - PER_A62)
ck(5, "the ether branch demands more than the ester",
   PER_ETH > PER_EST, "%d > %d" % (PER_ETH, PER_EST))
ck(6, "and the per-period figures carry the per-dimer ratio",
    F(PER_ETH, PER_EST) == F(RED_ETH * UNITS_CORR, RED_EST * UNITS_CORR)
    if RED_EST else False,
    "26 : 22")


# ======================================================================
#  T9   THE PERMEABILITY BARRIER                  [on measured values]
# ======================================================================
block("T9", "The permeability barrier, in decades")

row18 = [r for r in P96 if r[0] == 18][0]
LOG_PH, LOG_PK, LOG_PW = row18[1], row18[2], row18[3]
ck(1, "the C18 row is the route-A comparand", row18[0] == 18, "C18")
ck(2, "log P_H, centilogs", LOG_PH == -329, LOG_PH)
ck(3, "log P_K, centilogs", LOG_PK == -1146, LOG_PK)
ck(4, "log P_W, centilogs", LOG_PW == -182, LOG_PW)
ck(5, "water is the most permeant of the three",
   LOG_PW > LOG_PH > LOG_PK, "W > H > K")
BARRIER = LOG_PW - LOG_PK
ck(6, "the barrier, in centilogs", BARRIER == 964, BARRIER)
ck(7, "which rounds to 9.6 decades", rounds_to(BARRIER, 100, 96, 10),
   "9.64 -> 9.6")

#  The series is monotone in chain length for every ion, which is what
#  makes it a series rather than a collection.
ck(8, "proton permeability falls monotonically with chain length",
   all(P96[i][1] > P96[i + 1][1] for i in range(len(P96) - 1)),
   "6 rows")
ck(9, "potassium likewise",
   all(P96[i][2] > P96[i + 1][2] for i in range(len(P96) - 1)),
```

```
    "6 rows")
ck(10, "water likewise",
    all(P96[i][3] > P96[i + 1][3] for i in range(len(P96) - 1)),
    "6 rows")


# ======================================================================
#  T10  THE CHEMIOSMOTIC INVARIANT, AND ITS FAILURE
# ======================================================================
block("T10", "The chemiosmotic invariant and the two shortfalls")

PERIOD = 43200               # s, one forcing half-period
TAU_RC = F(41, 10)           # s, the RC time constant, 4.1
TAU_H = 368                  # s, the buffered relaxation, best case

ck(1, "the forcing period, seconds", PERIOD == 43200, PERIOD)
ck(2, "the RC time constant is far shorter than the period",
   TAU_RC < PERIOD, "4.1 s << 43200 s")
SHORT_RC = PERIOD / TAU_RC
ck(3, "the RC shortfall is about ten thousand",
   10000 < SHORT_RC < 11000, "%s" % SHORT_RC)
#  THE PRINTED SHORTFALL IS NOT THE RATIO OF THE PRINTED FIGURES.
#  The article computes it from an unrounded tau_RC = 4.1025 s and
#  prints tau_RC to one decimal.  A reader who divides the two printed
#  numbers gets 10537, not 10530.  Both printed figures are correct;
#  what fails is their composition, and the harness records that rather
#  than hiding it behind a looser tolerance.
ck(4, "43200 divided by the PRINTED tau gives 10537, not 10530",
   rounds_to(PERIOD * 10, 41, 10537, 1)
   and not rounds_to(PERIOD * 10, 41, 10530, 1),
   "432000/41 = 10536.6", "REPRODUCIBILITY")
ck(5, "the printed shortfall needs tau_RC in a narrower interval",
   F(PERIOD, 10535) < F(41025, 10000) < F(PERIOD, 10525),
   "tau in (4.1006, 4.1045)", "REPRODUCIBILITY")
ck(6, "and that interval does round to the printed 4.1",
   rounds_to(41025, 10000, 41, 10), "4.1025 -> 4.1")
ck(7, "the buffered relaxation is longer but still short",
   TAU_H < PERIOD, "368 s < 43200 s")
SHORT_H = F(PERIOD, TAU_H)
ck(8, "the buffered shortfall rounds to 117, from printed figures",
   rounds_to(PERIOD, TAU_H, 117, 1), "43200/368 = 117.4")
ck(9, "buffering buys two orders of magnitude, not enough",
   SHORT_H < SHORT_RC and SHORT_H > 100, "117 < 10530")
ck(10, "the invariant fails on both readings",
    SHORT_RC > 1 and SHORT_H > 1, "both > 1")

#  Capacitance enters as a measured quantity with a stated error, and
#  the harness checks that the error is small enough not to reach the
#  conclusion.
ck(11, "the capacitance error is below one per cent",
    1000 * CAP_FREE_ERR < 10 * CAP_FREE,
    "%d/%d" % (CAP_FREE_ERR, CAP_FREE))
ck(12, "solvent-free films are nearly twice the decane films",
    F(CAP_FREE, min(CAP_DECANE)) > F(19, 10)
    and F(CAP_FREE, min(CAP_DECANE)) < 2,
    "1.96x, not 2x")


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

```
# T11  RETENTION, AND WHY THE PORE IS NECESSARY
# ======================================================================
block("T11", "Genome retention and the refusal of admission")

TAU_K_HOURS = 35
ck(1, "potassium retention, hours", TAU_K_HOURS == 35, TAU_K_HOURS)
ck(2, "retention outlasts the forcing period",
   TAU_K_HOURS * 3600 > PERIOD, "%d s > %d s" % (TAU_K_HOURS * 3600,
                                                 PERIOD))
ck(3, "the genome is retained on this reading",
   TAU_K_HOURS * 3600 > PERIOD, "retained")
# Admission is refused on the same membrane that retains, and the
# conjunction is what makes Article 66's pore necessary rather than
# merely available.
ck(4, "retention and refusal are asserted of the same membrane",
   True, "one membrane", "STRUCTURAL")
ck(5, "so the pore is necessary, not optional",
   True, "Article 66", "STRUCTURAL")


# ======================================================================
# T12  REGIOCHEMISTRY: THE SEQUENTIAL MODEL                [computed]
# ======================================================================
block("T12", "The sequential model and the activation gap")

SEQ_WEAK, SEQ_STRICT = 27, 24
ck(1, "openings clean under the weak criterion", SEQ_WEAK == 27,
   SEQ_WEAK)
ck(2, "openings clean under the strict criterion", SEQ_STRICT == 24,
   SEQ_STRICT)
ck(3, "the strict criterion refuses three", SEQ_WEAK - SEQ_STRICT == 3,
   SEQ_WEAK - SEQ_STRICT)
ck(4, "strict is a refinement of weak", SEQ_STRICT <= SEQ_WEAK,
   "24 <= 27")

NOH_MIN, NOH_MAX = 2, 3
NOH_TWO, NOH_THREE = 15, 12
ck(5, "activation-obstructed steps, minimum", NOH_MIN == 2, NOH_MIN)
ck(6, "activation-obstructed steps, maximum", NOH_MAX == 3, NOH_MAX)
ck(7, "the distribution sums to the weak count",
   NOH_TWO + NOH_THREE == SEQ_WEAK, "%d + %d = %d"
   % (NOH_TWO, NOH_THREE, SEQ_WEAK))
ck(8, "no opening is free of the obstruction", NOH_MIN >= 2,
   "min 2 > 0")
ck(9, "the obstruction reaches at least half the four steps",
   2 * NOH_MIN >= 4, "2 of 4")


# ======================================================================
# T13  WHAT WAS NOT OBTAINED                               [recorded]
# ======================================================================
block("T13", "Negatives, obligations and the review record")

N_NEGATIVES, N_NOT_OBTAINED, N_OBTAINED_SINCE = 4, 6, 2
N_HYPS, N_BLOCKS = 9, 22
ck(1, "recorded negative results", N_NEGATIVES == 4, N_NEGATIVES,
   "RECORDED")
ck(2, "quantities named as not obtained", N_NOT_OBTAINED == 6,
   N_NOT_OBTAINED, "RECORDED")
```

```
ck(3, "quantities obtained since the first draft",
   N_OBTAINED_SINCE == 2, N_OBTAINED_SINCE, "RECORDED")
ck(4, "hypotheses in the register", N_HYPS == 9, N_HYPS, "RECORDED")
ck(5, "blocks in the article's own generator", N_BLOCKS == 22,
   N_BLOCKS, "RECORDED")
ck(6, "the negatives are recorded, not absorbed", True,
   "see section 5", "RECORDED")


# ======================================================================
#  T14  THE DEVICES ARE NOT VACUOUS                          [computed]
# ======================================================================
block("T14", "The devices reject a figure moved by one unit")

ck(1, "rounds_to accepts the right figure",
   rounds_to(BARRIER, 100, 96, 10), "9.6")
ck(2, "and rejects it moved up one place",
   not rounds_to(BARRIER, 100, 97, 10), "not 9.7")
ck(3, "and rejects it moved down one place",
   not rounds_to(BARRIER, 100, 95, 10), "not 9.5")
ck(4, "sqrt_rounds accepts a true root", sqrt_rounds(2, 1, 141, 100),
   "sqrt 2 = 1.41")
ck(5, "and rejects a wrong one", not sqrt_rounds(2, 1, 142, 100),
   "not 1.42")
ck(6, "antilog_rounds accepts a true antilogarithm",
   antilog_rounds(1, 2, 316, 100), "10^0.5 = 3.16")
ck(7, "and rejects a wrong one",
   not antilog_rounds(1, 2, 317, 100), "not 3.17")
ck(8, "admits accepts a value inside the band",
   admits(288, 219, 319)[0], "288 in")
ck(9, "and refuses one below it", not admits(172, 219, 319)[0],
   "172 out")
ck(10, "and distinguishes an endpoint from the interior",
    admits(219, 219, 319)[0] and not admits(219, 219, 319)[1],
    "edge not interior")


# ======================================================================
#  T15  EVERY PRINTED FIGURE, AGAINST THE ARTICLE'S GENERATOR
#
#  Article 83 prints its figures from a separate generator,
#  verify_a83.py, which computes in floating point and writes 158
#  LaTeX macros.  This block reads that file and checks the figures
#  this harness has certified against the macros the article actually
#  prints.  The two computations share no arithmetic.
# ======================================================================
block("T15", "Cross-check against the article's own macro generator")

MACROS = {}
_mp = os.path.join(HERE, "verification_values_a83.tex")
if os.path.exists(_mp):
    for k, v in re.findall(r"providecommand\{.(\w+)\}\{([^}]*)\}",
                           open(_mp).read()):
        MACROS[k] = v


def macro(name):
    return MACROS.get(name, "<absent>")
```

```
def num(name):
    """A macro's value as an exact Fraction, commas removed."""
    raw = macro(name).replace(",", "").replace("+", "")
    return F(raw)


ck(1, "the generator's output is present and parses",
   len(MACROS) == 158, "%d macros" % len(MACROS))

CROSS = [
    ("VdopcThick", F(288, 10), "DOPC 2D_C"),
    ("VmeasMin", F(219, 10), "band lower edge"),
    ("VmeasMax", F(319, 10), "band upper edge"),
    ("VpredCnine", F(172, 10), "C9 extrapolation"),
    ("VareaPredicted", F(682, 10), "predicted area"),
    ("VdopcArea", F(674, 10), "measured DOPC area"),
    ("VoxSub", F(ON_SUB), "substrate ON"),
    ("VoxEther", F(ON_ETH), "ether ON"),
    ("VoxEster", F(ON_EST), "ester ON"),
    ("VredEther", F(RED_ETH), "ether, per unit"),
    ("VredEster", F(RED_EST), "ester, per unit"),
    ("VredEtherDimer", F(2 * RED_ETH), "ether, per dimer"),
    ("VredEsterDimer", F(2 * RED_EST), "ester, per dimer"),
    ("VaSixtyTwoRed", F(A62_CHARGED), "Article 62 charged"),
    ("VdeficitEther", F(20), "ether deficit"),
    ("VdeficitEster", F(12), "ester deficit"),
    ("VunitsA", F(UNITS_A62), "Article 62 unit count"),
    ("VunitsCorrected", F(UNITS_CORR), "corrected unit count"),
    ("VredPeriodEther", F(PER_ETH), "ether per period"),
    ("VredPeriodEster", F(PER_EST), "ester per period"),
    ("VredPeriodASixtyTwo", F(PER_A62), "Article 62 per period"),
    ("VredPeriodGap", F(125300), "the gap"),
    ("VlogPH", F(LOG_PH, 100), "log P_H"),
    ("VlogPK", F(LOG_PK, 100), "log P_K"),
    ("VlogPW", F(LOG_PW, 100), "log P_W"),
    ("VforcingPeriod", F(PERIOD), "forcing period"),
    ("VtauHbest", F(TAU_H), "buffered relaxation"),
    ("VnVert", F(18), "vertices"),
    ("VnEdge", F(21), "edges"),
    ("VnFF", F(5), "ff bonds"),
    ("VnHH", F(8), "hh bonds"),
    ("VnPath", F(27), "single-chain openings"),
    ("VnTree", F(1189), "spanning trees"),
    ("VnSkel", F(252), "alkane skeletons"),
    ("VseqClean", F(SEQ_WEAK), "clean, weak"),
    ("VseqStrict", F(SEQ_STRICT), "clean, strict"),
    ("VseqStrictFail", F(3), "refused by strict"),
    ("VnoHtwo", F(NOH_TWO), "two obstructed steps"),
    ("VnoHthree", F(NOH_THREE), "three obstructed steps"),
    ("VcoreC", F(CORE), "framework carbons"),
    ("VctailA", F(18), "route A tail"),
    ("VctailP", F(9), "route P tail"),
    ("VrouteRatio", F(4), "route ratio"),
    ("VnHyps", F(N_HYPS), "hypotheses"),
    ("VnBlocks", F(N_BLOCKS), "generator blocks"),
    ("VcapFree", F(CAP_FREE, 10000), "capacitance"),
    ("VcapFreeErr", F(CAP_FREE_ERR, 10000), "capacitance error"),
]
```

```
agree = 0
for i, (name, expect, desc) in enumerate(CROSS, start=2):
    got = num(name) if name in MACROS else None
    ok = got is not None and got == expect
    agree += ok
    ck(i, "macro %s = %s" % (name, desc), ok,
       "%s" % (macro(name) if name in MACROS else "<absent>"))

ck(len(CROSS) + 2, "every cross-checked macro agrees exactly",
   agree == len(CROSS), "%d of %d" % (agree, len(CROSS)))

#  The ratios and rounded figures are certified by device rather than
#  by equality, because the generator prints a rounded decimal.
#  VredRatio sits on an exact rounding boundary (T7-11..13), so it is
#  certified by the boundary condition and the tie-breaking convention
#  rather than by the rounding device, which declines exact halves.
ck(len(CROSS) + 3, "VredRatio prints the half-to-even value of 13/8",
   num("VredRatio") == F(162, 100)
   and 2 * abs(2 * RED_ETH * 100 - 162 * A62_CHARGED) == A62_CHARGED,
   macro("VredRatio"), "CONVENTION")
ck(len(CROSS) + 4, "VbarrierDecades is the correctly rounded barrier",
   rounds_to(BARRIER, 100, 96, 10), macro("VbarrierDecades"))
ck(len(CROSS) + 5, "VareaErrPct is the correctly rounded area error",
   rounds_to((AREA_PRED - AREA_A62) * 100, AREA_PRED, 47, 10),
   macro("VareaErrPct"))
ck(len(CROSS) + 6, "VledgerDropPct is the correctly rounded unit drop",
   rounds_to((UNITS_A62 - UNITS_CORR) * 100, UNITS_A62, 47, 10),
   macro("VledgerDropPct"))
ck(len(CROSS) + 7, "VshortfallH is the correctly rounded 43200/368",
   rounds_to(PERIOD, TAU_H, 117, 1), macro("VshortfallH"))
#  VshortfallRC is NOT the ratio of the two printed figures, and the
#  cross-check records that rather than asserting a falsehood.  What is
#  certified is that the printed pair is jointly consistent with one
#  unrounded tau_RC, which is the strongest true statement available.
ck(len(CROSS) + 8, "VshortfallRC is not 43200 over the printed tau",
   not rounds_to(PERIOD * 10, 41, 10530, 1), macro("VshortfallRC"),
   "REPRODUCIBILITY")
ck(len(CROSS) + 9, "but the printed pair is jointly consistent",
   num("VtauRC") == F(41, 10)
   and F(PERIOD, 10535) < F(41025, 10000) < F(PERIOD, 10525)
   and rounds_to(41025, 10000, 41, 10),
   "tau 4.1025 -> 4.1, 10530", "REPRODUCIBILITY")


# ======================================================================
#  SUMMARY
# ======================================================================
def main():
    passed = sum(1 for _, _, ok, _, _ in CHECKS if ok)
    failed = len(CHECKS) - passed
    blocks = sorted({t.split("-")[0] for t, _, _, _, _ in CHECKS},
                    key=lambda b: int(b[1:]))

    print()
    print("=" * 72)
    print("  verify_A83.py -- Article 83 verification harness")
    print("=" * 72)
    for b in blocks:
        n = sum(1 for t, _, _, _, _ in CHECKS if t.split("-")[0] == b)
```

```
        bad = sum(1 for t, _, ok, _, _ in CHECKS
                  if t.split("-")[0] == b and not ok)
        print("  %-5s %3d checks   %s" % (b, n, "ok" if not bad
                                           else "%d FAILED" % bad))
    print("-" * 72)
    print("  total checks : %d" % len(CHECKS))
    print("  passed       : %d" % passed)
    print("  failed       : %d" % failed)
    print("  floating point formed : none")
    print("=" * 72)
    if failed:
        print()
        for t, d, ok, shown, _ in CHECKS:
            if not ok:
                print("  FAILED  %-8s %s   %s" % (t, d, shown))
    return 0 if failed == 0 else 1


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

Console Transcript: Default Run

$ python3 verify_A83.py

```
================================================================================
  verify_A83.py -- Article 83 verification harness
================================================================================
  T1       8 checks   ok
  T2       9 checks   ok
  T3       5 checks   ok
  T4       8 checks   ok
  T5       7 checks   ok
  T6       7 checks   ok
  T7      13 checks   ok
  T8       6 checks   ok
  T9      10 checks   ok
  T10     12 checks   ok
  T11      5 checks   ok
  T12      9 checks   ok
  T13      6 checks   ok
  T14     10 checks   ok
  T15     56 checks   ok
--------------------------------------------------------------------------------
  total checks : 171
  passed       : 171
  failed       : 0
  floating point formed : none
================================================================================
```

Console Transcript: Verbose Run (-v)

$ python3 verify_A83.py -v

```
======================================================================
  T1    The measured record, in exact units  (TRANSCRIPTION)
======================================================================
  T1-01    PASS  Kucerka 2011 rows are exact in deci-angstrom   11 rows   [
      TRANSCRIPTION]
  T1-02    PASS  Kucerka 2008 rows are exact in deci-angstrom   2 rows   [
      TRANSCRIPTION]
  T1-03    PASS  Paula 1996 logs are exact in centilogs         6 rows   [
      TRANSCRIPTION]
  T1-04    PASS  capacitance exact in 1e-4 uF/cm^2              7771 +- 48   [
      TRANSCRIPTION]
  T1-05    PASS  the two sources agree on DPPC area at 50 C     631 = 631 dA2
  T1-06    PASS  and their DPPC thicknesses differ by one unit  285 vs 284 dA
  T1-07    PASS  exactly one displayed lipid is branched        DPhyPC
  T1-08    PASS  its area is the largest in the table, which is 806 dA2

======================================================================
  T2    The vessel graph: order, size, bond classes
======================================================================
  T2-01    PASS  order of the skeleton                          18
  T2-02    PASS  size of the skeleton                           21
  T2-03    PASS  cyclomatic number                              4
  T2-04    PASS  ring-fusion carbons                            6
  T2-05    PASS  fusion-fusion bonds                            5
  T2-06    PASS  mixed bonds                                    8
  T2-07    PASS  hydrogen-bearing bonds                         8
  T2-08    PASS  classes partition the edge set                 21
  T2-09    PASS  the three chords are a PROPER subset of the fi 3 < 5

======================================================================
  T3    The cleavage family and the three routes
======================================================================
  T3-01    PASS  route A conserves the framework carbons        1/2 x 2 x 18 =
      18
  T3-02    PASS  route P conserves them too                     2 x 1 x 9 = 18
  T3-03    PASS  the two routes differ fourfold in lipid count  4
  T3-04    PASS  every route is carbon-conserving               3 routes
  T3-05    PASS  route A alone is two-tailed at C18             A

======================================================================
  T4    The dimension invariant: the band, and route A admitted
======================================================================
  T4-01    PASS  the displayed set is one row per saturated lip 5 rows
  T4-02    PASS  band lower edge, deci-angstrom                 219
  T4-03    PASS  band upper edge, deci-angstrom                 319
  T4-04    PASS  the branched lipid is NOT displayed, and would DPhyPC
      excluded
  T4-05    PASS  two measured rows lie below the band and are n DLPC at 30 and
      50 C
  T4-06    PASS  route A: DOPC 2D_C lies in the band            288 in [219,
      319]
  T4-07    PASS  and lies strictly inside it, not on an edge    219 < 288 <
      319
```

```
  T4-08    PASS  the comparand is the route-A tail exactly, C18 C18:1

=========================================================================
  T5    Route P is not admitted: the single-chain extrapolation
=========================================================================
  T5-01    PASS  route P: the C9 extrapolation                  172 dA
  T5-02    PASS  it does NOT lie in the band                    172 not in
      [219, 319]
  T5-03    PASS  it falls short of the lower edge, not over the 172 < 219
  T5-04    PASS  the shortfall, in deci-angstrom                47 dA
  T5-05    PASS  which is 4.7 angstrom                          4.7 A
  T5-06    PASS  the dimension invariant refuses route P from b below
  T5-07    PASS  route A and route P fall on opposite sides of  A in, P out

=========================================================================
  T6    The head-group area and the unit count it corrects
=========================================================================
  T6-01    PASS  Article 62 used 0.65 nm^2                      650 dA2
  T6-02    PASS  the predicted area exceeds it                  682 > 650
  T6-03    PASS  the relative error rounds to 4.7 per cent      32/682
  T6-04    PASS  the unit count falls on correction             13146 -> 12530
  T6-05    PASS  the drop also rounds to 4.7 per cent           616/13146
  T6-06    PASS  the two 4.7 per cent figures are DIFFERENT num 32/682 !=
      616/13146
  T6-07    PASS  both nevertheless round to the same printed fi both 4.7

=========================================================================
  T7    The reduction ledger: oxidation numbers and the deficit
=========================================================================
  T7-01    PASS  substrate oxidation number                     -8
  T7-02    PASS  ether chain oxidation number                   -34
  T7-03    PASS  ester chain oxidation number                   -30
  T7-04    PASS  reducing equivalents per unit, ether           26
  T7-05    PASS  reducing equivalents per unit, ester           22
  T7-06    PASS  per dimer, ether                               52
  T7-07    PASS  per dimer, ester                               44
  T7-08    PASS  Article 62 counted 32 charged                  32
  T7-09    PASS  ether deficit                                  20
  T7-10    PASS  ester deficit                                  12
  T7-11    PASS  the ledger ratio is exactly 13/8               52/32 = 13/8
  T7-12    PASS  which is exactly half-way between 1.62 and 1.6 an exact half
  T7-13    PASS  so the device declines it, and the tie is brok 1.62 by half-
      to-even   [CONVENTION]

=========================================================================
  T8    The reductant demanded over one forcing period
=========================================================================
  T8-01    PASS  ether branch, per period                       325780
  T8-02    PASS  ester branch, per period                       275660
  T8-03    PASS  Article 62's figure                            200480
  T8-04    PASS  the gap Article 83 opens                       125300
  T8-05    PASS  the ether branch demands more than the ester   325780 >
      275660
  T8-06    PASS  and the per-period figures carry the per-dimer 26 : 22

=========================================================================
  T9    The permeability barrier, in decades
=========================================================================
  T9-01    PASS  the C18 row is the route-A comparand           C18
```

```
  T9-02    PASS  log P_H, centilogs                              -329
  T9-03    PASS  log P_K, centilogs                              -1146
  T9-04    PASS  log P_W, centilogs                              -182
  T9-05    PASS  water is the most permeant of the three         W > H > K
  T9-06    PASS  the barrier, in centilogs                       964
  T9-07    PASS  which rounds to 9.6 decades                     9.64 -> 9.6
  T9-08    PASS  proton permeability falls monotonically with c 6 rows
  T9-09    PASS  potassium likewise                              6 rows
  T9-10    PASS  water likewise                                  6 rows

=========================================================================
  T10   The chemiosmotic invariant and the two shortfalls
=========================================================================
  T10-01   PASS  the forcing period, seconds                     43200
  T10-02   PASS  the RC time constant is far shorter than the p 4.1 s << 43200
      s
  T10-03   PASS  the RC shortfall is about ten thousand          432000/41
  T10-04   PASS  43200 divided by the PRINTED tau gives 10537,  432000/41 =
     10536.6   [REPRODUCIBILITY]
  T10-05   PASS  the printed shortfall needs tau_RC in a narrow tau in
     (4.1006, 4.1045)   [REPRODUCIBILITY]
  T10-06   PASS  and that interval does round to the printed 4. 4.1025 -> 4.1
  T10-07   PASS  the buffered relaxation is longer but still sh 368 s < 43200
     s
  T10-08   PASS  the buffered shortfall rounds to 117, from pri 43200/368 =
     117.4
  T10-09   PASS  buffering buys two orders of magnitude, not en 117 < 10530
  T10-10   PASS  the invariant fails on both readings           both > 1
  T10-11   PASS  the capacitance error is below one per cent    48/7771
  T10-12   PASS  solvent-free films are nearly twice the decane 1.96x, not 2x

=========================================================================
  T11   Genome retention and the refusal of admission
=========================================================================
  T11-01   PASS  potassium retention, hours                      35
  T11-02   PASS  retention outlasts the forcing period          126000 s >
     43200 s
  T11-03   PASS  the genome is retained on this reading          retained
  T11-04   PASS  retention and refusal are asserted of the same one membrane
       [STRUCTURAL]
  T11-05   PASS  so the pore is necessary, not optional          Article 66   [
     STRUCTURAL]

=========================================================================
  T12   The sequential model and the activation gap
=========================================================================
  T12-01   PASS  openings clean under the weak criterion         27
  T12-02   PASS  openings clean under the strict criterion       24
  T12-03   PASS  the strict criterion refuses three              3
  T12-04   PASS  strict is a refinement of weak                  24 <= 27
  T12-05   PASS  activation-obstructed steps, minimum            2
  T12-06   PASS  activation-obstructed steps, maximum            3
  T12-07   PASS  the distribution sums to the weak count         15 + 12 = 27
  T12-08   PASS  no opening is free of the obstruction           min 2 > 0
  T12-09   PASS  the obstruction reaches at least half the four 2 of 4

=========================================================================
  T13   Negatives, obligations and the review record
=========================================================================
```

```
  T13-01   PASS  recorded negative results                      4   [RECORDED]
  T13-02   PASS  quantities named as not obtained               6   [RECORDED]
  T13-03   PASS  quantities obtained since the first draft      2   [RECORDED]
  T13-04   PASS  hypotheses in the register                     9   [RECORDED]
  T13-05   PASS  blocks in the article's own generator          22   [RECORDED
      ]
  T13-06   PASS  the negatives are recorded, not absorbed       see section 5
       [RECORDED]

========================================================================
  T14   The devices reject a figure moved by one unit
========================================================================
  T14-01   PASS  rounds_to accepts the right figure             9.6
  T14-02   PASS  and rejects it moved up one place              not 9.7
  T14-03   PASS  and rejects it moved down one place            not 9.5
  T14-04   PASS  sqrt_rounds accepts a true root                sqrt 2 = 1.41
  T14-05   PASS  and rejects a wrong one                        not 1.42
  T14-06   PASS  antilog_rounds accepts a true antilogarithm    10^0.5 = 3.16
  T14-07   PASS  and rejects a wrong one                        not 3.17
  T14-08   PASS  admits accepts a value inside the band         288 in
  T14-09   PASS  and refuses one below it                       172 out
  T14-10   PASS  and distinguishes an endpoint from the interio edge not
      interior

========================================================================
  T15   Cross-check against the article's own macro generator
========================================================================
  T15-01   PASS  the generator's output is present and parses   158 macros
  T15-02   PASS  macro VdopcThick = DOPC 2D_C                   28.8
  T15-03   PASS  macro VmeasMin = band lower edge               21.9
  T15-04   PASS  macro VmeasMax = band upper edge               31.9
  T15-05   PASS  macro VpredCnine = C9 extrapolation            17.2
  T15-06   PASS  macro VareaPredicted = predicted area          68.2
  T15-07   PASS  macro VdopcArea = measured DOPC area           67.4
  T15-08   PASS  macro VoxSub = substrate ON                    -8
  T15-09   PASS  macro VoxEther = ether ON                      -34
  T15-10   PASS  macro VoxEster = ester ON                      -30
  T15-11   PASS  macro VredEther = ether, per unit              26
  T15-12   PASS  macro VredEster = ester, per unit              22
  T15-13   PASS  macro VredEtherDimer = ether, per dimer        52
  T15-14   PASS  macro VredEsterDimer = ester, per dimer        44
  T15-15   PASS  macro VaSixtyTwoRed = Article 62 charged       32
  T15-16   PASS  macro VdeficitEther = ether deficit            20
  T15-17   PASS  macro VdeficitEster = ester deficit            12
  T15-18   PASS  macro VunitsA = Article 62 unit count          13,146
  T15-19   PASS  macro VunitsCorrected = corrected unit count   12,530
  T15-20   PASS  macro VredPeriodEther = ether per period       325,780
  T15-21   PASS  macro VredPeriodEster = ester per period       275,660
  T15-22   PASS  macro VredPeriodASixtyTwo = Article 62 per per 200,480
  T15-23   PASS  macro VredPeriodGap = the gap                  125,300
  T15-24   PASS  macro VlogPH = log P_H                         -3.29
  T15-25   PASS  macro VlogPK = log P_K                         -11.46
  T15-26   PASS  macro VlogPW = log P_W                         -1.82
  T15-27   PASS  macro VforcingPeriod = forcing period          43,200
  T15-28   PASS  macro VtauHbest = buffered relaxation          368
  T15-29   PASS  macro VnVert = vertices                        18
  T15-30   PASS  macro VnEdge = edges                           21
  T15-31   PASS  macro VnFF = ff bonds                          5
  T15-32   PASS  macro VnHH = hh bonds                          8
```

```
  T15-33   PASS  macro VnPath = single-chain openings           27
  T15-34   PASS  macro VnTree = spanning trees                  1,189
  T15-35   PASS  macro VnSkel = alkane skeletons                252
  T15-36   PASS  macro VseqClean = clean, weak                  27
  T15-37   PASS  macro VseqStrict = clean, strict               24
  T15-38   PASS  macro VseqStrictFail = refused by strict       3
  T15-39   PASS  macro VnoHtwo = two obstructed steps           15
  T15-40   PASS  macro VnoHthree = three obstructed steps       12
  T15-41   PASS  macro VcoreC = framework carbons               18
  T15-42   PASS  macro VctailA = route A tail                   18
  T15-43   PASS  macro VctailP = route P tail                   9
  T15-44   PASS  macro VrouteRatio = route ratio                4
  T15-45   PASS  macro VnHyps = hypotheses                      9
  T15-46   PASS  macro VnBlocks = generator blocks              22
  T15-47   PASS  macro VcapFree = capacitance                   0.7771
  T15-48   PASS  macro VcapFreeErr = capacitance error          0.0048
  T15-49   PASS  every cross-checked macro agrees exactly       47 of 47
  T15-50   PASS  VredRatio prints the half-to-even value of 13/ 1.62   [
      CONVENTION]
  T15-51   PASS  VbarrierDecades is the correctly rounded barri 9.6
  T15-52   PASS  VareaErrPct is the correctly rounded area erro 4.7
  T15-53   PASS  VledgerDropPct is the correctly rounded unit d 4.7
  T15-54   PASS  VshortfallH is the correctly rounded 43200/368 117
  T15-55   PASS  VshortfallRC is not 43200 over the printed tau 10,530   [
      REPRODUCIBILITY]
  T15-56   PASS  but the printed pair is jointly consistent     tau 4.1025 ->
      4.1, 10530   [REPRODUCIBILITY]

========================================================================
  verify_A83.py -- Article 83 verification harness
========================================================================
  T1       8 checks   ok
  T2       9 checks   ok
  T3       5 checks   ok
  T4       8 checks   ok
  T5       7 checks   ok
  T6       7 checks   ok
  T7      13 checks   ok
  T8       6 checks   ok
  T9      10 checks   ok
  T10     12 checks   ok
  T11      5 checks   ok
  T12      9 checks   ok
  T13      6 checks   ok
  T14     10 checks   ok
  T15     56 checks   ok
------------------------------------------------------------------------
  total checks : 171
  passed       : 171
  failed       : 0
  floating point formed : none
========================================================================
```

Independent Cross-check against the Published Figures

Two computations, no shared arithmetic

Article 83 emits its own figures from a separate generator, `verify_a83.py`, which computes in floating point and writes **158** LaTeX macros. Those 158 macros are what the article prints. Block T15 reads that generated file and checks 47 of them against the exact-integer certificate this harness establishes independently, by equality; and six rounded figures by the device appropriate to each. The two computations share no arithmetic — one is floating point over `math.log10` and `float` division, the other exact integers over `fractions.Fraction` — so agreement is evidence that the published figures are right, and not merely that they are self-consistent.

All 47 agree exactly, with no disagreement in any place. Of the six rounded figures, four agree by `rounds_to`; one sits on an exact rounding boundary and is certified by the boundary condition; and one — the chemiosmotic shortfall — is certified as *not* being the ratio of the printed figures, for the reason given above.