ACF 83 — Lean 4 Proof 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

Proof file: `acf-83-proof.lean` Verification harness: `verify_A83.py`

Certification Status

Item	Value
Proof file	`acf-83-proof.lean`
Lean version targeted	`4.12.0` (core only, no Mathlib)
Namespace	`ChryseneFormalism.Article83`
Sections	S1–S15
Definitions	27
Theorems	74
Stated with `by decide`	74
`sorry`	**0**
`native_decide`	**0**
`import` lines	**0**
Goals re-evaluated in exact integer arithmetic	**74 of 74**
Goals found to hold	**74**
Kernel run	**not yet performed**
Axiom audit	**pending the kernel run**
Elaboration time	**not measured**

Read the status table before the theorem index

This file has not been through the Lean kernel. It was written in an environment with no Lean toolchain, and no claim is made here that it elaborates.

What *has* been established is narrower and is worth stating precisely, because it is not nothing. Every one of the 74 goals below was re-evaluated in exact integer arithmetic by `lean_certificates.py`, a companion file whose functions are written to match the Lean definitions term for term — including the two-directional `roundsTo`, which avoids `Nat` truncation, and the bracket devices, which avoid forming any irrational quantity. All 74 hold. A proposition that failed there would fail in Lean too.

So: **the propositions are true; whether the file elaborates is unknown.** The three figures that a Lean dossier normally leads with — theorems kernel-checked, axioms depended upon, elaboration time — are marked pending above rather than estimated, and they should stay that way until `lean acf-83-proof.lean` has been run.

The alternative was to quote the figures the Article 82 dossier quotes and let the format imply

they had been measured. That would repeat, in a new register, the two defects this corpus has already had to correct once each: Article 84's section 11, which passed every assertion while computing on the one family its own Theorem 6.1 excludes, and its check C1, which was described as enforced in both directions while being implemented nowhere. A dossier is a claim about what was done, and the claim has to be true.

What this file certifies, once it is checked

Each theorem is stated over `Nat` and discharged by `decide`: the goal is a decidable proposition, the kernel evaluates a verified Boolean decision procedure, and the proof is accepted only when that evaluation yields `true`. No external oracle, no `native_decide`, no `sorry`, no Mathlib import — the last four confirmed by a parse that strips comment blocks first, so that the file's own prose about avoiding them cannot be mistaken for use of them.

And what it will *not* certify. Article 83 is a concordance paper. Its membrane data were measured by other people and enter under hypothesis H_{data}, which is a premise about the world. It is not a theorem of this corpus, it is not an axiom of this file, and no kernel can settle it. The theorems below certify that the arithmetic Article 83 performs on its tabulated values is correct and that every figure it prints is the correctly rounded result of that arithmetic. They do not certify that DOPC's hydrophobic thickness is 28.8 Å.

Encoding

One decimal place makes the structural half an integer problem. Thicknesses and areas in Article 83 are quoted to one decimal, so the natural unit is the *deci-angstrom* and every structural datum is a natural number in it. Permeabilities are quoted as base-ten logarithms to two decimals, so the unit there is the *centilog*. Capacitance is quoted to four decimals and is carried in units of $10^{-4}\,\mu\text{F/cm}^2$. No rational encoding appears in this file at all.

Signs, without negatives. Oxidation numbers and log permeabilities are negative, and `Nat` has none. Each is carried as its *magnitude* together with the equation that fixes its direction: "the ether chain lies 26 below the substrate" is `onSubstrateMag + 26 = onEtherMag`, an equation between naturals, and the sign is the direction of that equation. Truncating subtraction — the standing hazard of `Nat` arithmetic — never appears on either side of a certificate.

One consequence is worth flagging because it inverts a reading. Since a *larger* magnitude is a *smaller* permeability, the ordering theorems of S10 run the opposite way from the ordering in the article's own table. `water_is_more_permeant_than_proton` is stated as $182 < 329$.

Four devices, no irrational formed.

`roundsTo` $n\,d\,r\,s$	n/d rounds to r/s	$2\lvert ns - rd\rvert < d$
`sqrtRounds` $n\,d\,r\,s$	$\sqrt{n/d}$ rounds to r/s	$(2r{-}1)^2 d \le n(2s)^2 < (2r{+}1)^2 d$
`antilogRounds` $n\,d\,r\,s$	$10^{n/d}$ 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 first three are the devices of the Article 82 proof file, reused. The first is written out in two directions rather than with a subtraction, for the reason above.

The admission bracket is this file's characteristic device. `admits` has no counterpart in the Article 82 file and is the formal content of the concordance method as this article uses it: 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. `admitsStrictly` is carried beside it because an admission landing on an endpoint is a different thing from one that does not, and S5 certifies that route A is interior rather than merely admitted.

The device is cheap, which is the point. Article 82's characteristic device compared integers of 902 and 3224 digits; this one compares three naturals below a thousand. A concordance test is not expensive to certify once the band is an integer interval — what is expensive is fixing the band honestly, and no kernel helps with that.

What the antilogarithm bracket is used for here, and why little. Article 83 needs the device far less than Article 82 did, because its permeabilities enter as *differences* of logarithms, which stay in centilogs and never leave `Nat`. S14 exercises the device on the one quantity that does need it, the barrier expressed as a factor rather than as a count of decades, and does so on the usual scientific split: $964 = 9 \cdot 100 + 64$, so the factor is $10^{0.64}$ times 10^9 and only the mantissa needs bracketing. Without the split the device would compare integers of about a thousand digits, which the kernel would settle but which nothing requires it to.

What is not encoded. Three things, each carried by the harness or by the article instead. The *provenance discipline* — which rows are displayed, which source each datum comes from, which values are flagged estimated — is here only as counts, while `verify_A83.py` carries it as executable structure. The *derivation* of the quantities is not reproduced: this file certifies that the barrier is 964 centilogs, where the harness computes it from the two imported logarithms, so a mistranscribed datum would fail there and not here. And the two *structural* claims of the article — that retention and refusal are asserted of the same membrane, and that Article 66's pore is therefore necessary — are not arithmetic and are certified nowhere.

Theorem Index

Section	Name	Statement (informal)
S1	`device_accepts_right_rounding`	the rounding bound accepts the barrier
	`device_rejects_rounding_up`	and rejects it moved up
	`device_rejects_rounding_down`	and moved down
	`device_accepts_true_root`	the squared bracket
	`device_rejects_wrong_root`	is not vacuous either
	`device_accepts_true_antilog`	nor the antilog bracket
	`device_rejects_wrong_antilog`	
	`device_admits_inside_band`	the admission bracket
	`device_refuses_outside_band`	refuses what is outside
	`edge_is_admitted`	an endpoint is admitted
	`edge_is_not_interior`	but is not interior
S2	`dppc_rows_agree_on_area`	the two sources overlap
	`dppc_thicknesses_differ_by_one`	by one in the last place
	`branched_area_is_the_largest`	which is why it is excluded
	`capacitance_error_under_one_percent`	the error does not reach the conclusion
S3	`euler_relation_holds`	$18 + 4 = 21 + 1$
	`bond_classes_partition_the_edges`	$5 + 8 + 8 = 21$
	`chords_are_fewer_than_fusion_bonds`	$3 < 5$
S4	`route_A_conserves_carbon`	two C18 tails per two units
	`route_D_conserves_carbon`	
	`route_P_conserves_carbon`	four C9 chains
	`routes_differ_fourfold`	the lipid count
S5	`band_is_nonempty`	$219 < 319$
	`route_A_is_admitted`	**the dimension invariant**
	`route_A_is_strictly_interior`	and not on an edge
S6	`route_P_is_refused`	the C9 chain is out
	`route_P_is_refused_from_below`	below, not above
	`shortfall_is_47_deci_angstrom`	by 4.7 Å exactly
	`the_two_routes_fall_on_opposite_sides`	one in, one out
S7	`area_gap_is_32`	$650 + 32 = 682$
	`area_error_rounds_to_47`	4.7 per cent
	`unit_drop_is_616`	$12{,}530 + 616 = 13{,}146$
	`unit_drop_rounds_to_47`	also 4.7 per cent
	`the_two_percentages_are_different_numbers`	**and they are not the same number**
S8	`ether_demand_is_26`	by the sign equation
	`ester_demand_is_22`	
	`ether_per_dimer_is_52`	
	`ester_per_dimer_is_44`	
	`ether_deficit_is_20`	against Article 62's 32
	`ester_deficit_is_12`	
	`ledger_ratio_is_thirteen_eighths`	$52/32 = 13/8$
	`ledger_ratio_sits_on_a_rounding_boundary`	**exactly half-way**
	`device_declines_both_neighbours`	so the device declines it
S9	`period_gap_is_125300`	what Article 83 adds
	`ether_exceeds_ester_over_a_period`	
	`period_figures_carry_the_per_unit_ratio`	26 : 22
S10	`water_is_more_permeant_than_proton`	in magnitudes, reversed
	`proton_is_more_permeant_than_potassium`	
	`barrier_is_964_centilogs`	by the sign equation
	`barrier_rounds_to_96_decades`	printed 9.6

Section	Name	Statement (informal)
	`proton_series_is_monotone`	six rows
	`potassium_series_is_monotone`	
	`water_series_is_monotone`	
S11	`buffered_relaxation_is_shorter_than_the_period`	
	`buffered_shortfall_rounds_to_117`	from printed figures
	`printed_pair_does_not_compose`	**43,200/4.1 is 10,537, not 10,530**
	`unrounded_time_constant_rounds_to_41`	
	`unrounded_time_constant_gives_the_printed_shortfall`	so both printed figures are right
S12	`retention_outlasts_the_forcing_period`	35 h against 12 h
	`retention_in_seconds`	
S13	`strict_criterion_refines_the_weak`	$24 \leq 27$
	`strict_criterion_refuses_three`	
	`obstruction_distribution_sums_to_the_weak_count`	$15 + 12 = 27$
	`obstruction_is_never_absent`	the floor is 2
	`obstruction_reaches_half_the_four_steps`	
S14	`barrier_exponent_splits`	$964 = 9 \cdot 100 + 64$
	`barrier_mantissa_rounds_to_437`	$10^{0.64} = 4.37$
	`mantissa_bracket_rejects_its_neighbours`	
	`mantissa_lies_between_one_and_ten`	the split is the usual one
S15	`negatives_recorded`	four
	`quantities_not_obtained`	six
	`quantities_obtained_since`	two
	`hypotheses_in_the_register`	nine
	`blocks_in_the_generator`	twenty-two

Proof Listing: `acf-83-proof.lean`

`acf-83-proof.lean` — complete source

```
/-
  acf-83-proof.lean
  ------------------------------------------------------------------
  Formal certificates 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 verification harness  verify_A83.py.

  Lean 4 core only.  No Mathlib, no `sorry`, no `native_decide`.
  Every theorem is stated over `Nat` and discharged by `decide`: the
  goal is a decidable proposition, the kernel evaluates a verified
  Boolean decision procedure, and the proof is accepted only when that
  evaluation yields `true`.

  To check:   lean acf-83-proof.lean
  A silent run with exit code 0 is the certificate.  There are no
  dependencies beyond the Lean core, so no lakefile and no package
  fetch are required.

  ------------------------------------------------------------------
  WHAT THIS CERTIFIES, AND WHAT IT DOES NOT

  Article 83 is a concordance paper.  Its membrane data were measured
  by other people and enter under hypothesis H_data, which is a premise
  about the world.  No kernel reaches it.  What is certified below is
  that the arithmetic Article 83 performs on its tabulated values is
  correct, and that every figure it prints is the correctly rounded
  result of that arithmetic.  Nothing here certifies that DOPC's
  hydrophobic thickness is 28.8 angstrom.

  A reader who takes a clean axiom audit as settling the paper's
  empirical claims has read the wrong thing into it.

  ------------------------------------------------------------------
  ENCODING

  One decimal place makes the structural half of the paper an integer
  problem.  Thicknesses and areas are quoted to one decimal, so the
  natural unit is the deci-angstrom, and in that unit every datum is a
  natural number.  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 1e-4 uF/cm^2.
  No rational encoding appears in this file at all.

  Signs, without negatives.  Oxidation numbers and log permeabilities
  are negative and `Nat` has none.  Each is carried as its MAGNITUDE
  together with the equation that fixes its direction: "the ether chain
  lies 26 below the substrate" is `onSubMag + 26 = onEtherMag`, an
  equation between naturals, and the sign is the direction of that
  equation.  Truncating subtraction, the standing hazard of `Nat`
  arithmetic, never appears on either side of a certificate.
```

```
-/

set_option maxRecDepth 4000
set_option exponentiation.threshold 2000

namespace ChryseneFormalism.Article83

/- ================================================================
   DEVICES

   Each replaces a statement about a rounded or irrational quantity by
   an inequality between exact naturals.  S1 shows that none of them is
   vacuous: moved by one unit in the last place, each returns false.
   ================================================================ -/

/-- `n/d` rounds to `r/s`.  Written in two directions rather than with
a subtraction, so that `Nat` truncation never arises.  The inequality
is strict, so a value sitting exactly half-way is DECLINED rather than
silently taking a tie-breaking convention; S8 turns on that. -/
def roundsTo (n d r s : Nat) : Prop :=
  (n * s ≥ r * d ∧ 2 * (n * s - r * d) < d) ∨
  (n * s < r * d ∧ 2 * (r * d - n * s) < d)

/-- `sqrt (n/d)` rounds to `r/s`.  The root is never formed; it is
squared against. -/
def sqrtRounds (n d r s : Nat) : Prop :=
  (2 * r - 1) ^ 2 * d ≤ n * (2 * s) ^ 2 ∧
  n * (2 * s) ^ 2 < (2 * r + 1) ^ 2 * d

/-- `10 ^ (n/d)` rounds to `r/s`.  The bracket is raised to the `d`,
which clears the root and leaves two exact naturals. -/
def antilogRounds (n d r s : Nat) : Prop :=
  (2 * r - 1) ^ d ≤ 10 ^ n * (2 * s) ^ d ∧
  10 ^ n * (2 * s) ^ d < (2 * r + 1) ^ d

/-- 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. -/
def admits (x lo hi : Nat) : Prop := lo ≤ x ∧ x ≤ hi

/-- Admission at an endpoint is a different thing from admission in the
interior, and Article 83 says which of its admissions are which. -/
def admitsStrictly (x lo hi : Nat) : Prop := lo < x ∧ x < hi

/- ================================================================
   S1   The devices are not vacuous
   ================================================================ -/

theorem device_accepts_right_rounding : roundsTo 964 100 96 10 := by decide

theorem device_rejects_rounding_up : ¬ roundsTo 964 100 97 10 := by decide

theorem device_rejects_rounding_down : ¬ roundsTo 964 100 95 10 := by decide

theorem device_accepts_true_root : sqrtRounds 2 1 141 100 := by decide

theorem device_rejects_wrong_root : ¬ sqrtRounds 2 1 142 100 := by decide

theorem device_accepts_true_antilog : antilogRounds 1 2 316 100 := by decide
```

```
theorem device_rejects_wrong_antilog : ¬ antilogRounds 1 2 317 100 := by
  decide

theorem device_admits_inside_band : admits 288 219 319 := by decide

theorem device_refuses_outside_band : ¬ admits 172 219 319 := by decide

theorem edge_is_admitted : admits 219 219 319 := by decide

theorem edge_is_not_interior : ¬ admitsStrictly 219 219 319 := by decide

/- ================================================================
   S2   The measured record, in exact units
   Every datum below is a natural number in its named unit, which is
   what licenses exact arithmetic for the whole paper.
   ================================================================ -/

/-- Hydrophobic thickness of DOPC, in deci-angstrom.  This is the
di-C18:1 cis-Delta-9 comparand: the exact tail of the route-A product,
and the one row the dimension test turns on. -/
def dopcTwoDC : Nat := 288

/-- Head-group area of DOPC, in deci-square-angstrom. -/
def dopcArea : Nat := 674

/-- Specific capacitance of a solvent-free film, in 1e-4 uF/cm^2. -/
def capFree : Nat := 7771

/-- Its stated error, in the same unit. -/
def capFreeErr : Nat := 48

theorem dppc_rows_agree_on_area : (631 : Nat) = 631 := by decide

theorem dppc_thicknesses_differ_by_one : (285 : Nat) = 284 + 1 := by
  decide

theorem branched_area_is_the_largest : (806 : Nat) > 674 := by decide

theorem capacitance_error_under_one_percent :
    1000 * capFreeErr < 10 * capFree := by decide

/- ================================================================
   S3   The vessel graph
   ================================================================ -/

theorem euler_relation_holds : (18 : Nat) + 4 = 21 + 1 := by decide

theorem bond_classes_partition_the_edges : (5 : Nat) + 8 + 8 = 21 := by
  decide

theorem chords_are_fewer_than_fusion_bonds : (3 : Nat) < 5 := by decide

/- ================================================================
   S4   The three routes, each carbon-conserving

   Lipids per unit is a half for route A, so every count is carried
   over the common denominator 2 and the equations are between
   naturals.
```

```
   ============================================================== -/

theorem route_A_conserves_carbon : 1 * 2 * 18 = 2 * 18 := by decide

theorem route_D_conserves_carbon : 2 * 2 * 9 = 2 * 18 := by decide

theorem route_P_conserves_carbon : 4 * 1 * 9 = 2 * 18 := by decide

theorem routes_differ_fourfold : (4 : Nat) = 4 * 1 := by decide

/- ==============================================================
   S5   The dimension invariant

   The band is fixed from the displayed rows before any candidate is
   scored.  That ordering is the concordance discipline of Article 82,
   and it is what distinguishes a test from a fit.
   ============================================================== -/

def bandLo : Nat := 219
def bandHi : Nat := 319

theorem band_is_nonempty : bandLo < bandHi := by decide

theorem route_A_is_admitted : admits dopcTwoDC bandLo bandHi := by decide

theorem route_A_is_strictly_interior :
    admitsStrictly dopcTwoDC bandLo bandHi := by decide

/- ==============================================================
   S6   Route P is refused
   ============================================================== -/

/-- The single-chain C9 extrapolation, in deci-angstrom. -/
def predCnine : Nat := 172

theorem route_P_is_refused : ¬ admits predCnine bandLo bandHi := by decide

theorem route_P_is_refused_from_below : predCnine < bandLo := by decide

theorem shortfall_is_47_deci_angstrom : predCnine + 47 = bandLo := by
  decide

theorem the_two_routes_fall_on_opposite_sides :
    admits dopcTwoDC bandLo bandHi ∧ ¬ admits predCnine bandLo bandHi := by
  decide

/- ==============================================================
   S7   The head-group area, and the unit count it corrects

   Two figures of this article both print as 4.7 per cent.  They are
   DIFFERENT numbers, and the last theorem of this section says so, so
   that a reader cannot take a coincidence of rounding for an identity.
   ============================================================== -/

def areaPredicted : Nat := 682
def areaArticle62 : Nat := 650
def unitsArticle62 : Nat := 13146
def unitsCorrected : Nat := 12530
```

```
theorem area_gap_is_32 : areaArticle62 + 32 = areaPredicted := by decide

theorem area_error_rounds_to_47 :
    roundsTo (32 * 100) areaPredicted 47 10 := by decide

theorem unit_drop_is_616 : unitsCorrected + 616 = unitsArticle62 := by
  decide

theorem unit_drop_rounds_to_47 :
    roundsTo (616 * 100) unitsArticle62 47 10 := by decide

theorem the_two_percentages_are_different_numbers :
    32 * unitsArticle62 ≠ 616 * areaPredicted := by decide

/- ================================================================
   S8   The reduction ledger

   Oxidation numbers are negative; each is carried as its magnitude
   with the equation that fixes its direction.
   ================================================================ -/

def onSubstrateMag : Nat := 8
def onEtherMag : Nat := 34
def onEsterMag : Nat := 30

theorem ether_demand_is_26 : onSubstrateMag + 26 = onEtherMag := by decide

theorem ester_demand_is_22 : onSubstrateMag + 22 = onEsterMag := by decide

theorem ether_per_dimer_is_52 : 2 * 26 = 52 := by decide

theorem ester_per_dimer_is_44 : 2 * 22 = 44 := by decide

theorem ether_deficit_is_20 : 32 + 20 = 52 := by decide

theorem ester_deficit_is_12 : 32 + 12 = 44 := by decide

theorem ledger_ratio_is_thirteen_eighths : 52 * 8 = 13 * 32 := by decide

/-- 52/32 is exactly 1.625, the midpoint between 1.62 and 1.63.  The
device declines an exact half by design, so what is certified is the
boundary condition itself. -/
theorem ledger_ratio_sits_on_a_rounding_boundary :
    2 * (52 * 100 - 162 * 32) = 32 := by decide

theorem device_declines_both_neighbours :
    ¬ roundsTo (52 * 100) 32 162 100 ∧ ¬ roundsTo (52 * 100) 32 163 100 := by
  decide

/- ================================================================
   S9   The reductant demanded over one forcing period
   ================================================================ -/

def periodEther : Nat := 325780
def periodEster : Nat := 275660
def periodArticle62 : Nat := 200480

theorem period_gap_is_125300 :
    periodArticle62 + 125300 = periodEther := by decide
```

```
theorem ether_exceeds_ester_over_a_period :
    periodEster < periodEther := by decide

theorem period_figures_carry_the_per_unit_ratio :
    periodEther * 22 = periodEster * 26 := by decide

/- ================================================================
   S10  The permeability barrier

   Log permeabilities are negative; magnitudes carry them, so a LARGER
   magnitude is a SMALLER permeability and the order below is reversed
   from the order in the article's own table.
   ================================================================ -/

def magLogWater : Nat := 182
def magLogProton : Nat := 329
def magLogPotassium : Nat := 1146

theorem water_is_more_permeant_than_proton :
    magLogWater < magLogProton := by decide

theorem proton_is_more_permeant_than_potassium :
    magLogProton < magLogPotassium := by decide

theorem barrier_is_964_centilogs :
    magLogWater + 964 = magLogPotassium := by decide

theorem barrier_rounds_to_96_decades : roundsTo 964 100 96 10 := by decide

/-- The proton series falls monotonically with chain length.  Written
as explicit conjunctions rather than over a list, so that the goal is
decidable without any instance beyond `Nat` comparison. -/
theorem proton_series_is_monotone :
    (189 : Nat) < 274 ∧ (274 : Nat) < 329 ∧ (329 : Nat) < 380 ∧
    (380 : Nat) < 410 ∧ (410 : Nat) < 431 := by decide

theorem potassium_series_is_monotone :
    (982 : Nat) < 1103 ∧ (1103 : Nat) < 1146 ∧ (1146 : Nat) < 1162 ∧
    (1162 : Nat) < 1166 ∧ (1166 : Nat) < 1177 := by decide

theorem water_series_is_monotone :
    (161 : Nat) < 171 ∧ (171 : Nat) < 182 ∧ (182 : Nat) < 195 ∧
    (195 : Nat) < 215 ∧ (215 : Nat) < 230 := by decide

/- ================================================================
   S11  The chemiosmotic invariant, and a reproducibility gap

   This section certifies something the article does not say, and the
   point of certifying it is that a reader would otherwise discover it
   by failing to reproduce a figure.  The printed shortfall 10530 is
   NOT 43200 divided by the printed time constant 4.1; that quotient is
   10537.  Both printed figures are correct.  What fails is their
   composition, because the shortfall was computed from an unrounded
   time constant of about 4.1025 s and the time constant was printed to
   one decimal.
   ================================================================ -/

def forcingPeriod : Nat := 43200
```

```
def bufferedRelaxation : Nat := 368

theorem buffered_relaxation_is_shorter_than_the_period :
    bufferedRelaxation < forcingPeriod := by decide

theorem buffered_shortfall_rounds_to_117 :
    roundsTo forcingPeriod bufferedRelaxation 117 1 := by decide

theorem printed_pair_does_not_compose :
    roundsTo (forcingPeriod * 10) 41 10537 1 ∧
    ¬ roundsTo (forcingPeriod * 10) 41 10530 1 := by decide

theorem unrounded_time_constant_rounds_to_41 :
    roundsTo 41025 10000 41 10 := by decide

theorem unrounded_time_constant_gives_the_printed_shortfall :
    roundsTo (forcingPeriod * 10000) 41025 10530 1 := by decide

/- ================================================================
   S12  Retention
   ================================================================ -/

theorem retention_outlasts_the_forcing_period :
    35 * 3600 > forcingPeriod := by decide

theorem retention_in_seconds : 35 * 3600 = 126000 := by decide

/- ================================================================
   S13  The sequential model and the activation gap
   ================================================================ -/

theorem strict_criterion_refines_the_weak : (24 : Nat) ≤ 27 := by decide

theorem strict_criterion_refuses_three : 24 + 3 = 27 := by decide

theorem obstruction_distribution_sums_to_the_weak_count :
    15 + 12 = 27 := by decide

theorem obstruction_is_never_absent : (2 : Nat) ≥ 2 := by decide

theorem obstruction_reaches_half_the_four_steps :
    2 * 2 ≥ 4 := by decide

/- ================================================================
   S14  An antilogarithm, on the split that keeps the integers small

   The barrier is 9.64 decades, so as a factor it is 10 ^ 9.64.  Split
   in the usual way, 964 = 9*100 + 64, the factor is 10 ^ 0.64 times
   10 ^ 9 and only the mantissa needs a bracket.  Without the split the
   device would compare integers of about a thousand digits, which the
   kernel would settle but which nothing requires it to.
   ================================================================ -/

theorem barrier_exponent_splits : (964 : Nat) = 9 * 100 + 64 := by decide

theorem barrier_mantissa_rounds_to_437 :
    antilogRounds 64 100 437 100 := by decide

theorem mantissa_bracket_rejects_its_neighbours :
```

```
    ¬ antilogRounds 64 100 438 100 ∧ ¬ antilogRounds 64 100 436 100 := by
  decide

theorem mantissa_lies_between_one_and_ten :
    (100 : Nat) ≤ 437 ∧ (437 : Nat) < 1000 := by decide

/- ================================================================
   S15  The register

   Counts only.  The content behind them -- which negatives, which
   obligations -- is carried by the article and by the harness, not
   here, and S15 asserts nothing about it.
   ================================================================ -/

theorem negatives_recorded : (4 : Nat) = 4 := by decide

theorem quantities_not_obtained : (6 : Nat) = 6 := by decide

theorem quantities_obtained_since : (2 : Nat) = 2 := by decide

theorem hypotheses_in_the_register : (9 : Nat) = 9 := by decide

theorem blocks_in_the_generator : (22 : Nat) = 22 := by decide

end ChryseneFormalism.Article83

/-
  ----------------------------------------------------------------
  AXIOM AUDIT

  Uncomment to run.  Each theorem above is discharged by `decide`,
  which reduces the goal to a kernel-evaluated Boolean expression, so
  the expected report for every one of them is

      'theorem' does not depend on any axioms

  and in particular not on `propext`, `Quot.sound`, `funext` or
  `Classical.choice`.

  #print axioms ChryseneFormalism.Article83.route_A_is_admitted
  #print axioms ChryseneFormalism.Article83.route_P_is_refused
  #print axioms ChryseneFormalism.Article83.barrier_is_964_centilogs
  #print axioms ChryseneFormalism.Article83.printed_pair_does_not_compose
-/
```

Arithmetic Validation: `lean_certificates.py`

How the goals were checked without a kernel

`lean_certificates.py` mirrors each Lean definition as a Python function written to match it term for term, and evaluates every goal of `acf-83-proof.lean` over Python's arbitrary-precision integers. The mirrors reproduce the features that matter: `roundsTo` is written in two directions so that no subtraction truncates, the bracket devices form no irrational quantity, and `admits` returns admission and strict interiority separately.

Result: 74 propositions evaluated, 74 hold, none fails. This is a check on the arithmetic and not on the Lean source. It would not catch a syntax error, a misapplied tactic, a name collision, or a definition that elaborates to something other than what its mirror assumes.

`lean_certificates.py` — complete source

```
#!/usr/bin/env python3
"""
lean_certificates.py
--------------------
Validates the arithmetic of every theorem in acf-83-proof.lean.

WHY THIS FILE EXISTS.  The Lean proof file was written in an environment
with no Lean toolchain, so its SYNTAX has not been checked by the Lean
elaborator.  Its ARITHMETIC can be checked here, and is.  Each entry
below carries the Lean theorem's own name and evaluates that theorem's
goal exactly, over Python's arbitrary-precision integers, using mirrors
of the Lean definitions written term for term.  A proposition that fails
here would fail in Lean too.

The file also asserts its own completeness: it reads acf-83-proof.lean,
extracts every `theorem` name, and fails if the set of names checked
here is not exactly the set declared there.  So the claim "every goal
was re-evaluated" is enforced rather than asserted.

WHAT THIS DOES NOT ESTABLISH: that the Lean file elaborates, that its
syntax is well formed, or that the kernel accepts it.  It would not
catch a syntax error, a misapplied tactic, a name collision, or a
definition that elaborates to something other than what its mirror here
assumes.  Those require Lean.  Run `lean acf-83-proof.lean`.

Run:  python3 lean_certificates.py        summary
      python3 lean_certificates.py -v     every proposition
"""

import os
import re
import sys

HERE = os.path.dirname(os.path.abspath(__file__))
LEAN = os.path.join(HERE, "acf-83-proof.lean")
VERBOSE = "-v" in sys.argv or "--verbose" in sys.argv

RESULTS = []          # (lean theorem name, holds, note)
```

```
def goal(name, prop, note=""):
    RESULTS.append((name, bool(prop), note))
    return bool(prop)


# ---------------------------------------------------------------------
#  MIRRORS OF THE LEAN DEFINITIONS, term for term.
# ---------------------------------------------------------------------
def roundsTo(n, d, r, s):
    """Lean:  (n*s >= r*d /\\ 2*(n*s - r*d) < d)
           \\/ (n*s <  r*d /\\ 2*(r*d - n*s) < d)
    Two directions, so that Nat subtraction never truncates."""
    if n * s >= r * d:
        return 2 * (n * s - r * d) < d
    return 2 * (r * d - n * s) < d


def sqrtRounds(n, d, r, s):
    """Lean: (2*r-1)^2 * d <= n*(2*s)^2  /\\  n*(2*s)^2 < (2*r+1)^2 * d"""
    return (2 * r - 1) ** 2 * d <= n * (2 * s) ** 2 < (2 * r + 1) ** 2 * d


def antilogRounds(n, d, r, s):
    """Lean: (2*r-1)^d <= 10^n * (2*s)^d  /\\  10^n*(2*s)^d < (2*r+1)^d"""
    return (2 * r - 1) ** d <= 10 ** n * (2 * s) ** d < (2 * r + 1) ** d


def admits(x, lo, hi):
    """Lean: lo <= x /\\ x <= hi"""
    return lo <= x <= hi


def admitsStrictly(x, lo, hi):
    """Lean: lo < x /\\ x < hi"""
    return lo < x < hi


#  The Lean file's definitions, with the same values.
dopcTwoDC, dopcArea = 288, 674
capFree, capFreeErr = 7771, 48
bandLo, bandHi = 219, 319
predCnine = 172
areaPredicted, areaArticle62 = 682, 650
unitsArticle62, unitsCorrected = 13146, 12530
onSubstrateMag, onEtherMag, onEsterMag = 8, 34, 30
periodEther, periodEster, periodArticle62 = 325780, 275660, 200480
magLogWater, magLogProton, magLogPotassium = 182, 329, 1146
forcingPeriod, bufferedRelaxation = 43200, 368


# =====================================================================
#  S1   The devices are not vacuous
# =====================================================================
goal("device_accepts_right_rounding", roundsTo(964, 100, 96, 10))
goal("device_rejects_rounding_up", not roundsTo(964, 100, 97, 10))
goal("device_rejects_rounding_down", not roundsTo(964, 100, 95, 10))
goal("device_accepts_true_root", sqrtRounds(2, 1, 141, 100))
goal("device_rejects_wrong_root", not sqrtRounds(2, 1, 142, 100))
```

```
goal("device_accepts_true_antilog", antilogRounds(1, 2, 316, 100))
goal("device_rejects_wrong_antilog", not antilogRounds(1, 2, 317, 100))
goal("device_admits_inside_band", admits(288, 219, 319))
goal("device_refuses_outside_band", not admits(172, 219, 319))
goal("edge_is_admitted", admits(219, 219, 319))
goal("edge_is_not_interior", not admitsStrictly(219, 219, 319))

# ============================================================================
#  S2   The measured record, in exact units
# ============================================================================
goal("dppc_rows_agree_on_area", 631 == 631, "both sources, 50 C")
goal("dppc_thicknesses_differ_by_one", 285 == 284 + 1)
goal("branched_area_is_the_largest", 806 > dopcArea, "DPhyPC")
goal("capacitance_error_under_one_percent",
     1000 * capFreeErr < 10 * capFree)

# ============================================================================
#  S3   The vessel graph
# ============================================================================
goal("euler_relation_holds", 18 + 4 == 21 + 1)
goal("bond_classes_partition_the_edges", 5 + 8 + 8 == 21)
goal("chords_are_fewer_than_fusion_bonds", 3 < 5)

# ============================================================================
#  S4   The three routes
# ============================================================================
goal("route_A_conserves_carbon", 1 * 2 * 18 == 2 * 18)
goal("route_D_conserves_carbon", 2 * 2 * 9 == 2 * 18)
goal("route_P_conserves_carbon", 4 * 1 * 9 == 2 * 18)
goal("routes_differ_fourfold", 4 == 4 * 1)

# ============================================================================
#  S5   The dimension invariant
# ============================================================================
goal("band_is_nonempty", bandLo < bandHi)
goal("route_A_is_admitted", admits(dopcTwoDC, bandLo, bandHi))
goal("route_A_is_strictly_interior",
     admitsStrictly(dopcTwoDC, bandLo, bandHi))

# ============================================================================
#  S6   Route P is refused
# ============================================================================
goal("route_P_is_refused", not admits(predCnine, bandLo, bandHi))
goal("route_P_is_refused_from_below", predCnine < bandLo)
goal("shortfall_is_47_deci_angstrom", predCnine + 47 == bandLo)
goal("the_two_routes_fall_on_opposite_sides",
     admits(dopcTwoDC, bandLo, bandHi)
     and not admits(predCnine, bandLo, bandHi))

# ============================================================================
#  S7   The head-group area and the unit count
# ============================================================================
goal("area_gap_is_32", areaArticle62 + 32 == areaPredicted)
goal("area_error_rounds_to_47", roundsTo(32 * 100, areaPredicted, 47, 10))
goal("unit_drop_is_616", unitsCorrected + 616 == unitsArticle62)
goal("unit_drop_rounds_to_47",
     roundsTo(616 * 100, unitsArticle62, 47, 10))
goal("the_two_percentages_are_different_numbers",
     32 * unitsArticle62 != 616 * areaPredicted,
```

```
     "both print as 4.7 per cent")

# ======================================================================
#  S8   The reduction ledger
# ======================================================================
goal("ether_demand_is_26", onSubstrateMag + 26 == onEtherMag)
goal("ester_demand_is_22", onSubstrateMag + 22 == onEsterMag)
goal("ether_per_dimer_is_52", 2 * 26 == 52)
goal("ester_per_dimer_is_44", 2 * 22 == 44)
goal("ether_deficit_is_20", 32 + 20 == 52)
goal("ester_deficit_is_12", 32 + 12 == 44)
goal("ledger_ratio_is_thirteen_eighths", 52 * 8 == 13 * 32)
goal("ledger_ratio_sits_on_a_rounding_boundary",
     2 * (52 * 100 - 162 * 32) == 32, "exactly 1.625")
goal("device_declines_both_neighbours",
     not roundsTo(52 * 100, 32, 162, 100)
     and not roundsTo(52 * 100, 32, 163, 100))

# ======================================================================
#  S9   The reductant over one period
# ======================================================================
goal("period_gap_is_125300", periodArticle62 + 125300 == periodEther)
goal("ether_exceeds_ester_over_a_period", periodEster < periodEther)
goal("period_figures_carry_the_per_unit_ratio",
     periodEther * 22 == periodEster * 26)

# ======================================================================
#  S10  The permeability barrier.  Magnitudes, so the order reverses.
# ======================================================================
goal("water_is_more_permeant_than_proton", magLogWater < magLogProton)
goal("proton_is_more_permeant_than_potassium",
     magLogProton < magLogPotassium)
goal("barrier_is_964_centilogs", magLogWater + 964 == magLogPotassium)
goal("barrier_rounds_to_96_decades", roundsTo(964, 100, 96, 10))
goal("proton_series_is_monotone",
     189 < 274 < 329 < 380 < 410 < 431)
goal("potassium_series_is_monotone",
     982 < 1103 < 1146 < 1162 < 1166 < 1177)
goal("water_series_is_monotone", 161 < 171 < 182 < 195 < 215 < 230)

# ======================================================================
#  S11  The chemiosmotic invariant, and the reproducibility gap
# ======================================================================
goal("buffered_relaxation_is_shorter_than_the_period",
     bufferedRelaxation < forcingPeriod)
goal("buffered_shortfall_rounds_to_117",
     roundsTo(forcingPeriod, bufferedRelaxation, 117, 1))
goal("printed_pair_does_not_compose",
     roundsTo(forcingPeriod * 10, 41, 10537, 1)
     and not roundsTo(forcingPeriod * 10, 41, 10530, 1),
     "43200/4.1 = 10537")
goal("unrounded_time_constant_rounds_to_41",
     roundsTo(41025, 10000, 41, 10))
goal("unrounded_time_constant_gives_the_printed_shortfall",
     roundsTo(forcingPeriod * 10000, 41025, 10530, 1))

# ======================================================================
#  S12  Retention
# ======================================================================
```

```
goal("retention_outlasts_the_forcing_period", 35 * 3600 > forcingPeriod)
goal("retention_in_seconds", 35 * 3600 == 126000)

# ======================================================================
#  S13  The sequential model
# ======================================================================
goal("strict_criterion_refines_the_weak", 24 <= 27)
goal("strict_criterion_refuses_three", 24 + 3 == 27)
goal("obstruction_distribution_sums_to_the_weak_count", 15 + 12 == 27)
goal("obstruction_is_never_absent", 2 >= 2)
goal("obstruction_reaches_half_the_four_steps", 2 * 2 >= 4)

# ======================================================================
#  S14  The antilogarithm, on the scientific split
# ======================================================================
goal("barrier_exponent_splits", 964 == 9 * 100 + 64)
goal("barrier_mantissa_rounds_to_437", antilogRounds(64, 100, 437, 100),
     "10^0.64 = 4.37")
goal("mantissa_bracket_rejects_its_neighbours",
     not antilogRounds(64, 100, 438, 100)
     and not antilogRounds(64, 100, 436, 100))
goal("mantissa_lies_between_one_and_ten", 100 <= 437 < 1000)

# ======================================================================
#  S15  The register
# ======================================================================
goal("negatives_recorded", 4 == 4)
goal("quantities_not_obtained", 6 == 6)
goal("quantities_obtained_since", 2 == 2)
goal("hypotheses_in_the_register", 9 == 9)
goal("blocks_in_the_generator", 22 == 22)


# ======================================================================
#  COMPLETENESS.  The claim that every goal of the Lean file was
#  re-evaluated here is enforced against the Lean source rather than
#  asserted in prose.
# ======================================================================
def completeness():
    if not os.path.exists(LEAN):
        return None, set(), set()
    declared = set(re.findall(r"^theorem ([A-Za-z_0-9]+)",
                              open(LEAN).read(), flags=re.M))
    checked = {n for n, _, _ in RESULTS}
    return declared, declared - checked, checked - declared


def main():
    declared, missing, extra = completeness()
    held = sum(1 for _, ok, _ in RESULTS if ok)
    failed = len(RESULTS) - held

    if VERBOSE:
        for n, ok, note in RESULTS:
            print("  %-4s %-52s %s" % ("ok" if ok else "FAIL", n, note))
        print()

    print("=" * 72)
    print("  lean_certificates.py -- arithmetic behind acf-83-proof.lean")
```

```
    print("=" * 72)
    print("  propositions evaluated : %d" % len(RESULTS))
    print("  holding                : %d" % held)
    print("  failing                : %d" % failed)
    if declared is None:
        print("  completeness           : acf-83-proof.lean not found")
    else:
        print("  theorems in the .lean  : %d" % len(declared))
        print("  declared but unchecked : %d %s"
              % (len(missing), sorted(missing) if missing else ""))
        print("  checked but undeclared : %d %s"
              % (len(extra), sorted(extra) if extra else ""))
    print("-" * 72)
    print("  This validates the ARITHMETIC of the Lean file, not its")
    print("  syntax.  Run `lean acf-83-proof.lean` for the kernel.")
    print("=" * 72)
    if failed:
        for n, ok, note in RESULTS:
            if not ok:
                print("  FAILED  %s  %s" % (n, note))
    bad = failed or (declared is not None and (missing or extra))
    return 1 if bad else 0


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

What Remains To Be Done

To complete this dossier

1. Install Lean 4.12.0 and run `lean acf-83-proof.lean`. There are no dependencies beyond the core, so no `lakefile` and no package fetch are required. A silent run with exit code 0 is the certificate.
2. Run `#print axioms` over each theorem. The expected report for every one of them is *"does not depend on any axioms"*, since `decide` reduces the goal to a kernel-evaluated Boolean expression; but that is a prediction and should be recorded as a measurement.
3. Measure the elaboration time and record it. The expensive constructs of the Article 82 file — a 16,807-case exhaustion, and brackets over integers of 3224 digits — have no counterpart here, so this file should be fast; again, a prediction.
4. Replace the three pending rows of the status table with what was measured, and strike the warning box on page 1.

If the file does *not* elaborate, the arithmetic validation above localises the problem: the propositions are true, so any failure is in the Lean source — syntax, a tactic, an instance, or a definition that does not mean what its mirror assumes — and not in the figures.

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 difference, ratio, count and band membership 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. Where it applies it applies absolutely — once it has been run.
The harness does four things the Lean file does not. It *derives* the quantities rather than restating them — the barrier of block T9 is computed from the two imported logarithms, not asserted to be 964 — 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* as executable structure, including which rows the article displays and which it does not. It performs the *cross-check* against the article's own generator, 47 macros by equality and six by device, which the Lean file cannot do because it cannot read the generated file. And it holds the *obligations* and the review record, so that what the paper does not establish travels with the numbers rather than with the prose.