ACF 85 — Lean 4 Proof Dossier

Equivariance of the Source Measure Off the Pure Limit:

Contact Sensitivity, Channel Capacity, and the Invariance of Selection Cost

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

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

Proof file: `acf-85-proof.lean` Verification harness: `verify_A85.py`

Kernel Certification

Item	Value
Proof file	`acf-85-proof.lean`
Lean version	`4.12.0` (core only, no Mathlib)
Namespace	`ChryseneFormalism.Article85`
Sections	S1–S10
Definitions	89
Theorems	129
Proved by `decide`	129
Depending on no axioms	**129**
Depending on `propext`	0
Depending on `Quot.sound` or `funext`	0
`sorry`	0
`native_decide`	0
Elaboration time	95 s
Exit code	`0`

What this file does not reach, said first

Article 85's central quantities are outside this file, and the dossier says so before its theorem index rather than after it.
The article's information figures are binary entropies of numbers of the form $1/(1+e^{g\Delta})$. Those are transcendental, and no integer decision procedure reaches them. There is *no bracket here* that plays the part the binary-logarithm bracket plays in the Article 84 proof file, because the quantity to be bracketed is not a logarithm of a rational: it is an entropy of a transcendental. The harness reaches these figures by carrying certified rational enclosures built from the factorial series for e and the inverse-hyperbolic-tangent series for $\ln 2$, with explicit remainder bounds and outward rounding onto a dyadic grid. That machinery is not reproduced in Lean, and the figures of the harness's blocks C7 and C10, and the trade ratios of C9, therefore have *no counterpart below*.

What is below is everything in the article that is combinatorial or rational, which is the larger part of it and the whole of its structure: the separation criterion on all 784 ordered contacts; the family dichotomy, its difference formula and its alphabet threshold; the sensitivity polynomial, built by the kernel from the enumeration; the registration gap and its masses; the absence of a preferred direction and the exact vanishing of the signed bias; the arm efficiency

with a positivity certificate for the monotonicity of σ; the dependence of adjacent contacts and the transfer matrix at $\ell = 6$; and the box arithmetic. It is not the whole of the article's numbers.

The file assumes nothing at all. An `#print axioms` audit over every theorem reports *"does not depend on any axioms"* in all 129 cases. The audit runs inside the file, so a run of `lean acf-85-proof.lean` prints 129 lines and exits 0; that transcript, not a silent run, is the certificate.

The Gibbs step, and why it is a grid

One statement of the article does reach past the rationals and is here anyway, and the honest account of it is worth a box of its own.
Averaged over a uniformly chosen prescribed field, the conditional cost of a contact is $\frac{1}{2}(-\log_2\theta - \log_2(1-\theta)) = \frac{1}{2}\log_2\frac{1}{\theta(1-\theta)}$, which is at least one bit *exactly because* $\theta(1-\theta) \le \frac{1}{4}$. That is algebra, and it holds whatever e is: writing $\theta = a/(a+c)$, the claim is $4ac \le (a+c)^2$. So the conditional cost exceeds Article 84's figure *on average* and not merely for an unlucky field — with no value of e and no value of Δ entering.

S1 decides that inequality on a grid of 1024 rationals. It does *not* prove it for every θ, and the reason is worth recording. A general proof needs the square expanded, and the tactics that expand it — `ring`, `nlinarith` — are Mathlib, which this file does not import. The two core tactics that come close, `simp` and `omega`, both prove it at once; both also carry `propext` and `Quot.sound` into the proof term, as `#print axioms` on either immediately shows, and the file would lose its empty audit for one lemma. Between a general theorem with a three-axiom footprint and a grid with none, this file takes the grid and says so.

The harness checks the same inequality on a grid of 96 rationals, so the two artifacts have the same standing on this one point. Neither should be read as a proof of the general statement, and the article's own argument for it is the algebraic one, not either grid.

Encoding

Two arm letters, and the alphabet declared. Article 85 works over $\{h, e\}$: Article 79's phosphorylation is downstream of admission, so an e_P arm is not in the admitted stream. An arm word is therefore a natural below 16 whose bit j is one for a cleaved arm and zero for an intact one at locant $3(j+1)$, with the bits off the family held at zero. Where a statement is alphabet-dependent S4 re-encodes the three-letter and the four- and five-letter alphabets beside it, which is how the threshold of that section is reached at all.

Forty-eight slots for twenty-eight types. A unit type is a flat index $k < 48$ with family $k/16$ and word index k mod 16, valid when the index is below the family's word count. Four, eight and sixteen words make twenty-eight types, and S2 counts them. Indexing this way wastes twenty slots and needs no subtraction, which is the trade the whole file makes.

Signed coefficients, without a single subtraction. The cleavage model makes $1-\pi_h$ unavoidable, so σ has coefficients of both signs: 1, 38, 16 176 842, −32 353 912, 16 174 936, 9144, −14 248, 9600, −2400 over 2011^2. A polynomial is therefore carried as a *pair* of naturals, each a packed base-2^{128} digit string of non-negative coefficients, and the polynomial is their difference. Addition is addition, multiplication is multiplication — the packing is faithful as long as no digit carries, which `no_packed_digit_carries` checks — and

$$(P_1 - N_1)(P_2 - N_2) = (P_1P_2 + N_1N_2) - (P_1N_2 + N_1P_2)$$

keeps both halves non-negative. A coefficient claim is then an equation between naturals: the cubic coefficient is stated as `coefP SP 3` + 32 353 912 = `coefP SN 3`, and the sign is the direction of that equation.

The two pinned halves are written out by `packOf`, coefficient by coefficient, rather than as the three-hundred-digit naturals they evaluate to. That is a presentation choice and not a mathematical one, but it is the right one twice over: the nine digits of each half are the content, and a listing cannot break a three-hundred-digit numeral across a line.

Nothing is subtracted anywhere. A rounding is a pair of strict inequalities rather than an absolute value. A ratio is cross-multiplied rather than divided. $q-\pi_h q$ is supplied as its own argument `qmp` rather than computed. Four minus a family's arm count is a table, `famSlack`, so that the common denominator $2011\,q^4$ is reached without one. A probability complement is stated as $p + q =$ total: the exact P(at least one sensitive contact) at $\ell = 6$ appears as `pNoNum` + $A = (2011 \cdot 10^4)^6$. And the registration gap, which is a difference of two counts, is computed as a count in whichever direction is the larger, with `gapOf` and a theorem fixing which direction that is.

The derivations, performed once. Four objects in this file are expensive: the fold over the 2304 ordered index pairs that builds σ, the gap masses, the two halves of the signed bias, and the transfer matrix. Each is evaluated *once*, pinned to a literal by a theorem of its own, and every later certificate is stated over that literal. The derivation is still in the file and still checked — if the enumeration gave anything else, the pinning theorem would fail and the rest would be about the wrong object — and the elaboration cost falls by about a factor of five, which is the difference between ninety seconds and eight minutes.

The state vector is packed too. The transfer matrix of S9 iterates a vector over forty-eight states. Carried as a function, that vector would be recomputed once per lookup and five steps would cost 48^5 evaluations; packed into one natural in base 2^{320}, the iteration is linear in the column length. The packing is what makes $\ell = 6$ reachable at all. It does not make $\ell = 1802$ reachable — the exact half-length at $\pi_e = 0.99$ is the harness's, and S9 says which figures stop at six sites.

What is not encoded. Three things, each of which the harness carries instead.

The *enclosures*, and therefore every information figure of the article: blocks C7 and C10 of the harness have no section here.

The *three-letter alphabet* as a running parallel. S4 uses it for the two false lemmas and the threshold, because those statements are about it; the harness runs it beside the two-letter alphabet throughout, which is how the article knows which of its statements are alphabet-independent.

The *long columns*. S9 certifies the transfer matrix at $\ell = 6$; the harness carries it to $\ell = 54$ and, for the half-lengths, to $\ell = 1802$, where the kernel's evaluation is not within reach.

Theorem Index

Section	Name	Statement (informal)
S1	`rounding_accepts_a_correct_value`	1/3 rounds to 0.33
	`rounding_rejects_the_value_below`	and not to 0.32
	`rounding_rejects_the_value_above`	nor to 0.34
	`rounding_rejects_a_half_way_case`	and a half-way case fails
	`digits_accepts_four`, `digits_rejects_three`	the digit bracket
	`theta_times_its_complement_is_at_most_a_quarter`	**the Gibbs step**: $4ac \le (a+c)^2$ on 1024 rationals
	`with_equality_exactly_when_the_contact_is_balanced`	equality at $a = c$, i.e. $\Delta = 0$
	`and_the_constant_four_is_the_best_one`	five would fail
S2	`iota_is_an_involution`	$\iota^2 = \mathrm{id}$ on the four locants
	`iota_fixes_D_and_T_but_not_I`	setwise, which is the whole mechanism of S7
	`family_arm_counts`	two, three and four arms
	`the_slack_completes_the_arm_count`	the table reaches four without a subtraction
	`the_admitted_numerators_sum_to_2011`	$2000 + 1 + 10$
	`f_I_squared_is_one_over_2011_squared`	$4\,044\,121$
	`f_I_rounds_to_497e_minus_4`	4.97×10^{-4}
	`f_I_squared_rounds_to_247e_minus_7`	2.47×10^{-7}
	`the_word_counts_are_four_eight_sixteen`	over $\{h, e\}$
	`the_stream_has_twentyeight_unit_types`	28 of 48 slots
	`the_word_indexing_is_injective`	no index names two words
	`a_word_has_no_arm_off_its_family`	the off-family bits are zero
S3	`the_ordered_contacts_number_784`	28^2
	`the_separation_criterion`	**Prop. "The separation criterion"**: 784 cases, both directions
	`the_cross_overlaps_are_equal`	the identity the criterion does *not* rest on
	`sensitivity_is_symmetric`	in the two units
	`one_blind_unit_suffices`	Article 84's theorem, over this alphabet
	`the_converse_fails_over_this_alphabet`	two sighted units may still be blind
S4	`DD_sighted_contacts_are_all_sensitive`	**Prop. "The two families differ"**
	`there_are_four_such_DD_contacts`	4

Section	Name	Statement (informal)
	`TT_has_forty_blind_both_sighted_contacts`	40
	`TT_has_one_hundred_and_four_sensitive_ones`	104
	`the_TT_sensitive_share_rounds_to_722`	0.722
	`II_has_twentyfour_blind_both_sighted_contacts`	24, so $\mathcal{I}$ behaves like $\mathcal{T}$
	`the_dichotomy_is_D_against_the_rest`	and not $\mathcal{D}, \mathcal{I}$ against $\mathcal{T}$
	`the_difference_formula_holds_on_D`	on every sighted pair, sign-free
	`the_difference_is_never_zero_on_sighted_D`	which is why the contact is sensitive
	`over_two_letters_the_difference_is_two_either_way`	exactly ± 2
	`no_blind_sighted_DD_contact_at_two_letters`	0
	`nor_at_three`	0
	`but_twentyfour_of_one_hundred_and_fortyfour_at_four`	**the threshold**: four letters
	`and_more_still_at_five`	and beyond
	`so_the_proposition_bounds_the_alphabet_at_three`	which a two-and-three test could not see
	`the_first_lemma_has_108_counterexamples`	the submitted proof's first lemma is false
	`the_second_lemma_has_54`	and so is its second
	`so_neither_lemma_is_available`	recorded, not absorbed
	`no_sighted_D_word_meets_its_flip`	$\mathcal{D}$ carries one ι-orbit
	`eight_of_the_twelve_sighted_T_words_do`	$\mathcal{T}$ carries two
S5	`sigma_is_derived_from_the_enumeration`	**the derivation**: the 2304-contact fold, pinned once
	`sigma_has_no_coefficient_above_degree_eight`	nothing at degree nine
	`no_packed_digit_carries`	so the packing is faithful
	`sigma_at_zero_is_f_I_squared`	the point of contact with Article 84
	`the_constant_term_is_the_only_one_at_zero`	Horner at $p = 0$
	`the_linear_coefficient_is_38_over_2011_squared`	38
	`the_linear_coefficient_rounds_to_940e_minus_6`	9.40×10^{-6}
	`the_quadratic_coefficient_is_16176842_over_4044121`	exactly
	`it_rounds_to_four_point_00009`	4.00009
	`and_is_not_the_integer_four`	which is the point of printing five places
	`the_linear_term_is_two_orders_below_the_quadratic`	so the regime is quadratic
	`the_degree_is_exactly_eight`	leading coefficient -2400
	`sigma_at_pi_e_nine_tenths` … `…ninetynine_hundredths`	σ exactly at the three operating points
	`sigma_rounds_to_324e_minus_2_at_nine_tenths` …	and the three printed figures

Section	Name	Statement (informal)
	`the_lift_is_131037_at_nine_tenths` …	the lifts 131 037, 36 502, 1587
	`one_per_cent_cleavage_lifts_sensitivity_by_three_orders`	the headline comparison
S6	`the_gap_is_the_difference_of_the_two_registration_numbers`	sign-free, in whichever direction is larger
	`a_contact_is_sensitive_exactly_when_its_gap_is_positive`	the two notions agree
	`the_gap_is_at_most_four`	two tetrasubstituted units
	`the_gap_masses_are_derived_at_nine_tenths`	the five masses, pinned once
	`the_gap_masses_are_derived_at_perfect_arming`	and at $\pi_h = 0$
	`and_every_gap_from_one_to_four_is_attained`	all four occur
	`at_perfect_arming_only_gap_one_survives`	the $\mathcal{I}$–$\mathcal{I}$ contact alone
	`and_the_blind_mass_is_everything_else`	$4\,044\,120 + 1 = 2011^2$
	`the_masses_are_normalised_at_nine_tenths`	to $(2011 \cdot 10^4)^2$
	`the_sensitive_mass_is_sigma_at_nine_tenths`	**S6 meets S5**: two independent computations agree
	`the_blind_mass_is_one_minus_sigma_at_nine_tenths`	exactly
	`the_gap_two_share_rounds_to_99997`	0.99997
	`the_gap_one_share_rounds_to_zero`	0.000
	`the_mean_gap_rounds_to_two`	2.000
	`the_distribution_is_a_near_point_mass_not_a_balance`	which is why the mean must not be read as one
S7	`relab_is_an_involution`	$(a \circ \iota) \circ \iota = a$
	`relab_reads_the_arm_from_the_inverted_locant`	by construction
	`relab_preserves_the_weight_on_D_and_T`	measure-preserving
	`relab_is_a_bijection_of_D_and_of_T_words`	onto, by Boolean search
	`relab_reverses_the_pattern`	ν_0 and ν_1 are exchanged
	`on_I_the_relabelling_is_not_even_defined`	$\iota\mathcal{I} \not\subseteq \mathcal{I}$
	`without_the_initiator_the_signed_bias_is_exactly_zero`	**Thm. "No preferred direction"**, at three operating points
	`and_the_two_halves_are_not_themselves_zero`	so the vanishing is cancellation, not absence
	`with_the_initiator_the_bias_is_nonzero`	the halves differ by 82 000 000
	`and_it_rounds_to_203e_minus_7`	2.03×10^{-7}
	`the_bias_is_below_f_I_squared`	which is the order the article claims
S8	`arm_efficiency_rounds_to_9875_at_nine_tenths` …	0.9875, 0.9964, 0.9997
	`arm_efficiency_is_one_at_perfect_arming`	$\eta = 1$

Section	Name	Statement (informal)
	`arm_efficiency_falls_as_cleavage_rises`	cross-multiplied
	`the_derivative_has_degree_seven`	σ' from σ
	`the_derivative_at_zero_is_the_linear_coefficient`	38
	`every_substituted_coefficient_is_positive`	**the certificate**: $(1+t)^7\sigma'\left(\frac{t/5}{1+t}\right)$ has all coefficients positive
	`the_substitution_has_degree_seven`	nothing above
	`the_least_substituted_coefficient_is_the_derivative_at_zero`	$38 \cdot 5^7 = 2\,968\,750$
	`so_sigma_is_strictly_increasing_on_the_operating_range`	on $[0, 1/5]$, by certificate and not by grid
	`and_the_trade_is_genuine_sensitivity_up_efficiency_down`	both directions at once
S9	`the_adjacent_mass_is_derived`	the triple sum, pinned once
	`adjacent_contacts_are_positively_correlated_at_nine_tenths`	exceeds σ^2
	`and_the_correlation_ratio_rounds_to_five_point_six`	5.6
	`the_packed_vector_starts_at_one_on_every_type`	the packing is faithful
	`and_at_zero_off_the_valid_types`	the twenty spare slots carry nothing
	`p_no_sensitive_at_six_sites_nine_tenths`	five transfer steps, exactly
	`p_any_at_six_sites_nine_tenths`	the complement, as an equation
	`p_any_rounds_to_1363_at_nine_tenths`	0.1363
	`p_no_sensitive_at_six_sites_perfect_arming`	and at $\pi_h = 0$
	`p_any_at_six_sites_perfect_arming`	2011^6
	`p_any_rounds_to_124e_minus_6_at_perfect_arming`	1.24×10^{-6}
	`the_binomial_figure_overstates_at_six_sites`	the dependence works against the agent
	`the_blind_mass_is_the_complement_of_sigma`	the literal used above
	`the_expected_count_at_six_sites_rounds_to_162_thousandths`	0.162 of five
	`the_expected_count_at_fiftyfour_sites_rounds_to_1717_thousandths`	1.717 of fifty-three
	`the_still_needed_share_rounds_to_9676_hundredths`	96.76 per cent
	`over_ninetysix_per_cent_still_need_the_reference`	a refinement of H_{frame}, not its discharge
S10	`the_box_has_fiftyfour_nodes`	$3 \times 3 \times 6$

Section	Name	Statement (informal)
	`it_is_nine_columns_of_six`	along the stacking axis
	`it_carries_fortyfive_stacking_contacts`	45
	`a_single_column_of_fiftyfour_would_carry_fiftythree`	53
	`the_two_figures_are_different`	and the article says which is which

Proof Listing: `acf-85-proof.lean`

`acf-85-proof.lean` — complete source

```
/-
  ==========================================================================
  acf-85-proof.lean

  Formal proof file for
    Annals of the Chrysene Formalism, Article 85
    "The Partially Sighted Source: Arm Cleavage as an Orientation
     Channel, and the Price of Blindness"
    Charles D. Schaper, Ph.D., Schaper Systems Press

  Lean 4.12.0, 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 returns `true`.

  Companion harness: verify_A85.py, blocks C1-C12.

  WHAT THIS FILE DOES NOT REACH, SAID FIRST.
  Article 85's central quantities are binary entropies of numbers of the
  form 1/(1 + e^{g*Delta}).  They are transcendental, and no integer
  decision procedure reaches them: there is no bracket here that plays
  the part the binary-logarithm bracket plays in the Article 84 proof
  file, because the quantity to be bracketed is not a logarithm of a
  rational.  The harness reaches them by carrying certified rational
  enclosures built from the series for e and for ln 2; that machinery is
  not reproduced here, and the figures of the harness's blocks C7, C10
  and the trade ratios of C9 therefore have NO counterpart below.

  What is below is everything in the article that is combinatorial or
  rational: the separation criterion, the family dichotomy and its
  threshold, the sensitivity polynomial derived from the enumeration, the
  registration gap, the absence of a preferred direction, the arm
  efficiency with a positivity certificate for the monotonicity, the
  dependence of adjacent contacts, and the box arithmetic.  That is the
  larger part of the article and the whole of its structure; it is not
  the whole of its numbers, and the dossier says which is which.

  ENCODING.
  * An arm word over {h, e} on a family is a natural below 16 whose bit j
    is 1 for h and 0 for e at locant 3(j+1); bits off the family are held
    at zero.
  * Axiom A2's inversion iota -- 3<->9, 6<->12 -- is the index involution
    0<->2, 1<->3.
  * A unit type is a flat index k < 48 with family k/16 and word index
    k%16, valid when the index is below the family's word count.  Forty
    eight slots carry twenty eight types; indexing this way needs no
    subtraction, which is the point.
  * A polynomial with integer coefficients is carried as a PAIR of
    naturals, each a packed base-2^128 digit string of non-negative
    coefficients; the polynomial is their difference.  Addition is
    addition, multiplication is multiplication, and the identity
    (P1-N1)(P2-N2) = (P1 P2 + N1 N2) - (P1 N2 + N1 P2) keeps both parts
    non-negative.  So Article 85's signed coefficients -- the cleavage
```

```
    model makes 1 - pi_h unavoidable -- are carried without a single
    subtraction.
  * Nothing is subtracted anywhere.  A rounding is a pair of strict
    inequalities, a ratio is a cross-multiplied equation, a negative
    coefficient is the second half of a pair, and a probability
    complement is stated as `p + q = total`.
  ================================================================
-/

set_option exponentiation.threshold 4000
set_option maxRecDepth 40000
set_option maxHeartbeats 8000000

namespace ChryseneFormalism.Article85

-- ====================================================================
--  S0.  Counting and summing, tail-recursively
-- ====================================================================

def b2n (b : Bool) : Nat := if b then 1 else 0

def cntGo (f : Nat → Bool) : Nat → Nat → Nat
  | 0, acc => acc
  | n + 1, acc => cntGo f n (acc + b2n (f n))

def cnt (f : Nat → Bool) (n : Nat) : Nat := cntGo f n 0

def sumGo (f : Nat → Nat) : Nat → Nat → Nat
  | 0, acc => acc
  | n + 1, acc => sumGo f n (acc + f n)

def sumf (f : Nat → Nat) (n : Nat) : Nat := sumGo f n 0

/-- The four locant indices, summed over without a list. -/
def four (g : Nat → Bool) : Nat := b2n (g 0) + b2n (g 1) + b2n (g 2) + b2n (g 3)

/-- A bounded search as a Boolean.  Core's `Nat.decidableExistsLT` decides
    a bounded existential through `decidable_of_iff`, which carries
    `propext` and `Quot.sound` into the proof term; this does not. -/
def anyB (f : Nat → Bool) : Nat → Bool
  | 0 => false
  | n + 1 => f n || anyB f n

-- ====================================================================
--  S1.  The devices, and the Gibbs step
-- ====================================================================

/-- `n/d` rounds to `r/s`: `2|ns - rd| < d`, written in both directions
    so that nothing is subtracted. -/
def roundsTo (n d r s : Nat) : Prop :=
  2 * (n * s) < 2 * (r * d) + d ∧ 2 * (r * d) < 2 * (n * s) + d

/-- `n` has exactly `k` decimal digits. -/
def digitsIs (n k : Nat) : Prop := 10 ^ k ≤ 10 * n ∧ n < 10 ^ k

instance (n d r s : Nat) : Decidable (roundsTo n d r s) := by
  unfold roundsTo; infer_instance
instance (n k : Nat) : Decidable (digitsIs n k) := by
  unfold digitsIs; infer_instance
```

```
theorem rounding_accepts_a_correct_value : roundsTo 1 3 33 100 := by decide
theorem rounding_rejects_the_value_below : ¬ roundsTo 1 3 32 100 := by decide
theorem rounding_rejects_the_value_above : ¬ roundsTo 1 3 34 100 := by decide
theorem rounding_rejects_a_half_way_case : ¬ roundsTo 1 2 1 1 := by decide
theorem digits_accepts_four : digitsIs 1000 4 := by decide
theorem digits_rejects_three : ¬ digitsIs 1000 3 := by decide

/-- THE GIBBS STEP, and the one place this file reaches past the
    rationals.  Averaged over a uniformly chosen prescribed field, the
    conditional cost of a contact is
    (1/2)(-log2 theta - log2 (1 - theta)) = (1/2) log2 (1/(theta(1-theta))),
    which is at least one bit exactly because theta (1 - theta) <= 1/4.
    That is algebra and holds whatever e is: writing theta = a/(a+c), the
    claim is 4ac <= (a+c)^2.  So the conditional cost EXCEEDS Article
    84's figure on average, and not merely for an unlucky field --
    without any value of e or of Delta entering.

    It is decided here on a grid of 1024 rationals rather than proved for
    every theta.  A general proof needs the square to be expanded, and
    the tactics that do that -- `ring`, `nlinarith` -- are Mathlib, which
    this file does not import; the core tactics that come close,
    `simp` and `omega`, each carry `propext` and `Quot.sound` into the
    proof term and would cost the file its empty axiom audit.  The
    harness checks the same inequality on a grid of 96 rationals, so the
    two artifacts have the same standing here, and the dossier says so
    rather than implying a theorem. -/
theorem theta_times_its_complement_is_at_most_a_quarter :
    ∀ a, a < 32 → ∀ c, c < 32 → 4 * (a * c) ≤ (a + c) * (a + c) := by decide
theorem with_equality_exactly_when_the_contact_is_balanced :
    ∀ a, a < 32 → 4 * (a * a) = (a + a) * (a + a) := by decide
theorem and_the_constant_four_is_the_best_one :
    ¬ (∀ a, a < 32 → ∀ c, c < 32 → 5 * (a * c) ≤ (a + c) * (a + c)) := by decide

-- ======================================================================
--  S2.  The stream: families, arm words, and the admitted fractions
-- ======================================================================
--
--  Article 85 works over the two-letter arm alphabet {h, e}: Article
--  79's phosphorylation is downstream of admission, so an e_P arm is not
--  in the admitted stream.  Where a statement is alphabet-dependent, S4
--  runs the three-letter alphabet beside it and says so.

def pw2 (j : Nat) : Nat := if j = 0 then 1 else if j = 1 then 2 else if j = 2
    then 4 else 8

/-- The arm letter at locant index `j` of the word `w`: 1 is h, 0 is e. -/
def armBit (w j : Nat) : Nat := w / pw2 j % 2

/-- iota on locant indices: 0<->2 is 3<->9, 1<->3 is 6<->12. -/
def iotaIdx (j : Nat) : Nat :=
  if j = 0 then 2 else if j = 1 then 3 else if j = 2 then 0 else 1

/-- Families: 0 is D = {6,12}, 1 is I = {3,6,12}, 2 is T = {3,6,9,12}. -/
def inFam (f j : Nat) : Bool :=
  if f = 0 then j == 1 || j == 3 else if f = 1 then !(j == 2) else true

/-- Arm words over {h,e}: four on D, eight on I, sixteen on T. -/
def famSize (f : Nat) : Nat := if f = 0 then 4 else if f = 1 then 8 else 16
```

```
/-- Article 79's admitted numerators at rho* = 200, p3/p4 = 1/10, over
    the common denominator 2011. -/
def famWeight (f : Nat) : Nat := if f = 0 then 2000 else if f = 1 then 1 else 10

/-- Four minus the family's arm count, as a table, so that the common
    denominator 2011 * q^4 can be reached without a subtraction. -/
def famSlack (f : Nat) : Nat := if f = 0 then 2 else if f = 1 then 1 else 0

/-- The `i`-th arm word of family `f`, placing the bits of `i` at the
    family's locants. -/
def wordOf (f i : Nat) : Nat :=
  if f = 0 then (i % 2) * 2 + (i / 2 % 2) * 8
  else if f = 1 then (i % 2) + (i / 2 % 2) * 2 + (i / 4 % 2) * 8
  else i

/-- A unit type is a flat index below 48. -/
def tyFam (k : Nat) : Nat := k / 16
def tyIdx (k : Nat) : Nat := k % 16
def tyOK (k : Nat) : Bool := decide (k / 16 < 3) && decide (k % 16 < famSize (k
    / 16))
def tyWord (k : Nat) : Nat := wordOf (tyFam k) (tyIdx k)

theorem iota_is_an_involution : ∀ j, j < 4 → iotaIdx (iotaIdx j) = j := by
    decide
theorem iota_fixes_D_and_T_but_not_I :
    (∀ j, j < 4 → (inFam 0 j = inFam 0 (iotaIdx j))) ∧
      (∀ j, j < 4 → (inFam 2 j = inFam 2 (iotaIdx j))) ∧
      ¬ (∀ j, j < 4 → (inFam 1 j = inFam 1 (iotaIdx j))) := by decide
theorem family_arm_counts :
    four (inFam 0) = 2 ∧ four (inFam 1) = 3 ∧ four (inFam 2) = 4 := by decide
theorem the_slack_completes_the_arm_count :
    ∀ f, f < 3 → four (inFam f) + famSlack f = 4 := by decide

theorem the_admitted_numerators_sum_to_2011 :
    famWeight 0 + famWeight 1 + famWeight 2 = 2011 := by decide
theorem f_I_squared_is_one_over_2011_squared : 2011 * 2011 = 4044121 := by
    decide
theorem f_I_rounds_to_497e_minus_4 : roundsTo (1 * 10 ^ 4) 2011 497 100 := by
    decide
theorem f_I_squared_rounds_to_247e_minus_7 :
    roundsTo (1 * 10 ^ 7) 4044121 247 100 := by decide

theorem the_word_counts_are_four_eight_sixteen :
    famSize 0 = 4 ∧ famSize 1 = 8 ∧ famSize 2 = 16 := by decide
theorem the_stream_has_twentyeight_unit_types : cnt tyOK 48 = 28 := by decide
theorem the_word_indexing_is_injective :
    ∀ f, f < 3 → ∀ i, i < 16 → ∀ j, j < 16 →
      (decide (i < famSize f) && decide (j < famSize f)
        && wordOf f i == wordOf f j) = true → i = j := by decide
theorem a_word_has_no_arm_off_its_family :
    ∀ k, k < 48 → tyOK k = true →
      cnt (fun j => !inFam (tyFam k) j && armBit (tyWord k) j == 1) 4 = 0 := by
  decide

-- ====================================================================
--  S3.  The separation criterion
-- ====================================================================
--
```

```
-- Article 85, Prop. "The separation criterion". A contact is
-- orientation-sensitive exactly when the second unit's pattern
-- separates the first unit's pattern from its flip by overlap count.
-- The content is that the criterion may be tested on ONE of the two
-- units; the equality of the two cross-overlaps is an identity and is
-- not what carries the proposition.

/-- Locant j contributes to the registration number of an aligned
    contact. -/
def hit (f1 w1 f2 w2 j : Nat) : Bool :=
  inFam f1 j && inFam f2 j && armBit w1 j == armBit w2 j

/-- And of the inverted one. -/
def hitF (f1 w1 f2 w2 j : Nat) : Bool :=
  inFam f1 j && inFam f2 (iotaIdx j) && armBit w1 j == armBit w2 (iotaIdx j)

def nuA (f1 w1 f2 w2 : Nat) : Nat := four (hit f1 w1 f2 w2)
def nuF (f1 w1 f2 w2 : Nat) : Nat := four (hitF f1 w1 f2 w2)

def sens (f1 w1 f2 w2 : Nat) : Bool := !(nuA f1 w1 f2 w2 == nuF f1 w1 f2 w2)

/-- The overlap of the FIRST unit's flipped pattern with the second's
    aligned pattern -- the quantity the criterion is stated in. -/
def nuFirstFlipped (f1 w1 f2 w2 : Nat) : Nat :=
  four (fun j => inFam f1 (iotaIdx j) && inFam f2 j
                 && armBit w1 (iotaIdx j) == armBit w2 j)

def unitBlind (f w : Nat) : Bool :=
  four (fun j => inFam f j
                 && !(inFam f (iotaIdx j) && armBit w (iotaIdx j) == armBit w j)
   ) == 0

/-- A sum over all ordered contacts of the two-letter stream. -/
def pairSum (p : Nat → Nat → Nat) : Nat :=
  sumf (fun k1 => if tyOK k1 then
          sumf (fun k2 => if tyOK k2 then p k1 k2 else 0) 48 else 0) 48

def pairCount (p : Nat → Nat → Bool) : Nat := pairSum (fun a b => b2n (p a b))

theorem the_ordered_contacts_number_784 :
    pairCount (fun _ _ => true) = 784 := by decide

/-- The criterion, on all 784 ordered contacts: sensitivity holds exactly
    when the first unit's pattern and its flip have different overlaps
    with the second's. -/
theorem the_separation_criterion :
    pairCount (fun k1 k2 =>
      !(sens (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2) ==
        !(nuA (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2) ==
          nuFirstFlipped (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2)))) = 0 :=
    by
  decide

/-- The identity the criterion does not rest on: the two cross-overlaps
    agree, always. -/
theorem the_cross_overlaps_are_equal :
    pairCount (fun k1 k2 =>
      !(nuF (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2) ==
        nuFirstFlipped (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2))) = 0 := by
```

```
  decide

theorem sensitivity_is_symmetric :
    pairCount (fun k1 k2 =>
      !(sens (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2) ==
        sens (tyFam k2) (tyWord k2) (tyFam k1) (tyWord k1))) = 0 := by
  decide

/-- Article 84's theorem, reproduced over the alphabet of this paper. -/
theorem one_blind_unit_suffices :
    pairCount (fun k1 k2 =>
      (unitBlind (tyFam k1) (tyWord k1) || unitBlind (tyFam k2) (tyWord k2))
        && sens (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2)) = 0 := by
  decide

theorem the_converse_fails_over_this_alphabet :
    0 < pairCount (fun k1 k2 =>
      !unitBlind (tyFam k1) (tyWord k1) && !unitBlind (tyFam k2) (tyWord k2)
        && !sens (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2)) := by
  decide

-- ======================================================================
--  S4.  Why D behaves differently from I and T
-- ======================================================================
--
--  Article 85, Prop. "The two families behave differently".  The proof
--  rests on a difference formula; the two lemmas the submitted version
--  asserted are both false, and the counterexamples are counted here
--  rather than absorbed.

def bothSighted (f1 f2 : Nat) (p : Nat → Nat → Bool) : Nat :=
  sumf (fun i1 => if decide (i1 < famSize f1)
                    && !unitBlind f1 (wordOf f1 i1) then
          sumf (fun i2 => if decide (i2 < famSize f2)
                             && !unitBlind f2 (wordOf f2 i2) then
                    b2n (p (wordOf f1 i1) (wordOf f2 i2)) else 0) 16 else 0) 16

theorem DD_sighted_contacts_are_all_sensitive :
    bothSighted 0 0 (fun w1 w2 => !sens 0 w1 0 w2) = 0 := by decide
theorem there_are_four_such_DD_contacts :
    bothSighted 0 0 (fun _ _ => true) = 4 := by decide
theorem TT_has_forty_blind_both_sighted_contacts :
    bothSighted 2 2 (fun w1 w2 => !sens 2 w1 2 w2) = 40 := by decide
theorem TT_has_one_hundred_and_four_sensitive_ones :
    bothSighted 2 2 (fun w1 w2 => sens 2 w1 2 w2) = 104 := by decide
theorem the_TT_sensitive_share_rounds_to_722 :
    roundsTo 104 144 722 1000 := by decide
theorem II_has_twentyfour_blind_both_sighted_contacts :
    bothSighted 1 1 (fun w1 w2 => !sens 1 w1 1 w2) = 24 := by decide
theorem the_dichotomy_is_D_against_the_rest :
    bothSighted 0 0 (fun w1 w2 => !sens 0 w1 0 w2) = 0 ∧
      0 < bothSighted 1 1 (fun w1 w2 => !sens 1 w1 1 w2) ∧
      0 < bothSighted 2 2 (fun w1 w2 => !sens 2 w1 2 w2) := by decide

/-- The difference formula.  For sighted D words the overlap difference is
    the four-term expression below, and over {h,e} it is +2 when the two
    words agree and -2 when one is the swap of the other.  Written
    sign-free: the two sides of the difference are counted separately and
    the formula says which exceeds which, by two. -/
```

```
def ddAgree (w1 w2 : Nat) : Nat :=
  b2n (armBit w1 1 == armBit w2 1) + b2n (armBit w1 3 == armBit w2 3)
def ddCross (w1 w2 : Nat) : Nat :=
  b2n (armBit w1 1 == armBit w2 3) + b2n (armBit w1 3 == armBit w2 1)

theorem the_difference_formula_holds_on_D :
    bothSighted 0 0 (fun w1 w2 =>
      !(nuA 0 w1 0 w2 + ddCross w1 w2 == nuF 0 w1 0 w2 + ddAgree w1 w2)) = 0 :=
    by
  decide
theorem the_difference_is_never_zero_on_sighted_D :
    bothSighted 0 0 (fun w1 w2 => ddAgree w1 w2 == ddCross w1 w2) = 0 := by
    decide
theorem over_two_letters_the_difference_is_two_either_way :
    bothSighted 0 0 (fun w1 w2 =>
      !((ddAgree w1 w2 == ddCross w1 w2 + 2)
        || (ddCross w1 w2 == ddAgree w1 w2 + 2))) = 0 := by decide

/-- The threshold.  A zero difference on D-D needs both of one word's
    letters outside the other's pair and distinct from each other, hence
    FOUR letters.  Arm words over `m` letters on D are indexed by
    `i < m*m`, the letters being `i % m` at locant 6 and `i / m` at 12. -/
def dLetter (m i j : Nat) : Nat := if j = 1 then i % m else i / m
def dSighted (m i : Nat) : Bool := !(dLetter m i 1 == dLetter m i 3)
def dBlindPair (m i1 i2 : Nat) : Bool :=
  (b2n (dLetter m i1 1 == dLetter m i2 1) + b2n (dLetter m i1 3 == dLetter m i2
    3))
    == (b2n (dLetter m i1 1 == dLetter m i2 3) + b2n (dLetter m i1 3 == dLetter
    m i2 1))

def dThresh (m : Nat) (p : Nat → Nat → Bool) : Nat :=
  sumf (fun i1 => if dSighted m i1 then
          sumf (fun i2 => if dSighted m i2 then b2n (p i1 i2) else 0) (m * m)
        else 0) (m * m)

theorem no_blind_sighted_DD_contact_at_two_letters :
    dThresh 2 (dBlindPair 2) = 0 := by decide
theorem nor_at_three :
    dThresh 3 (dBlindPair 3) = 0 := by decide
theorem but_twentyfour_of_one_hundred_and_fortyfour_at_four :
    dThresh 4 (dBlindPair 4) = 24 ∧ dThresh 4 (fun _ _ => true) = 144 := by
    decide
theorem and_more_still_at_five :
    0 < dThresh 5 (dBlindPair 5) := by decide
theorem so_the_proposition_bounds_the_alphabet_at_three :
    dThresh 3 (dBlindPair 3) = 0 ∧ 0 < dThresh 4 (dBlindPair 4) := by decide

/-- The three-letter alphabet, for the two false lemmas.  Arm words over
    three letters are indexed as in the Article 84 proof file: base-three
    digits placed at the family's locants. -/
def arm3 (w j : Nat) : Nat := w / (if j = 0 then 1 else if j = 1 then 4
                                   else if j = 2 then 16 else 64) % 4
def fam3Size (f : Nat) : Nat := if f = 0 then 9 else if f = 1 then 27 else 81
def word3 (f i : Nat) : Nat :=
  if f = 0 then (i % 3) * 4 + (i / 3 % 3) * 64
  else if f = 1 then (i % 3) + (i / 3 % 3) * 4 + (i / 9 % 3) * 64
  else (i % 3) + (i / 3 % 3) * 4 + (i / 9 % 3) * 16 + (i / 27 % 3) * 64

def hit3 (f1 w1 f2 w2 j : Nat) : Bool :=
```

```
  inFam f1 j && inFam f2 j && arm3 w1 j == arm3 w2 j
def hit3F (f1 w1 f2 w2 j : Nat) : Bool :=
  inFam f1 j && inFam f2 (iotaIdx j) && arm3 w1 j == arm3 w2 (iotaIdx j)
def nu3A (f1 w1 f2 w2 : Nat) : Nat := four (hit3 f1 w1 f2 w2)
def nu3F (f1 w1 f2 w2 : Nat) : Nat := four (hit3F f1 w1 f2 w2)
def blind3 (f w : Nat) : Bool :=
  four (fun j => inFam f j
                 && !(inFam f (iotaIdx j) && arm3 w (iotaIdx j) == arm3 w j)) ==
    0

/-- Counterexamples to the two lemmas, over three arm letters: a sighted
    D word against a sighted word of any family, making a blind contact,
    with a positive registration number for the first lemma and a zero
    one for the second. -/
def lemmaFail (positive : Bool) : Nat :=
  sumf (fun f2 => sumf (fun i1 =>
    if decide (i1 < 9) && !blind3 0 (word3 0 i1) then
      sumf (fun i2 =>
        if decide (i2 < fam3Size f2) && !blind3 f2 (word3 f2 i2)
           && nu3A 0 (word3 0 i1) f2 (word3 f2 i2)
              == nu3F 0 (word3 0 i1) f2 (word3 f2 i2)
           && (if positive then decide (0 < nu3A 0 (word3 0 i1) f2 (word3 f2 i2))
   )
               else nu3A 0 (word3 0 i1) f2 (word3 f2 i2) == 0)
        then 1 else 0) 81
    else 0) 9) 3

theorem the_first_lemma_has_108_counterexamples : lemmaFail true = 108 := by
    decide
theorem the_second_lemma_has_54 : lemmaFail false = 54 := by decide
theorem so_neither_lemma_is_available :
    0 < lemmaFail true ∧ 0 < lemmaFail false := by decide

/-- Why: a sighted D pattern is disjoint from its flip; a sighted T
    pattern need not be. -/
def meetsItsFlip (f w : Nat) : Bool :=
  0 < four (fun j => inFam f j && inFam f (iotaIdx j)
                     && armBit w (iotaIdx j) == armBit w j)

theorem no_sighted_D_word_meets_its_flip :
    cnt (fun i => decide (i < 4) && !unitBlind 0 (wordOf 0 i)
                  && meetsItsFlip 0 (wordOf 0 i)) 16 = 0 := by decide
theorem eight_of_the_twelve_sighted_T_words_do :
    cnt (fun i => decide (i < 16) && !unitBlind 2 (wordOf 2 i)
                  && meetsItsFlip 2 (wordOf 2 i)) 16 = 8 ∧
      cnt (fun i => decide (i < 16) && !unitBlind 2 (wordOf 2 i)) 16 = 12 := by
  decide

-- ======================================================================
--  S5.  The sensitivity polynomial, derived from the enumeration
-- ======================================================================
--
--  Article 85, Thm. "The sensitivity polynomial".  Each arm is cleaved
--  independently with probability pi_h, so the probability that a
--  contact drawn from the admitted stream is orientation-sensitive is a
--  polynomial in pi_h with rational coefficients.  It is DERIVED here by
--  the enumeration of S3, not restated: the kernel builds the 784 word
--  polynomials, multiplies them pairwise, weights them, and adds.
--
```

```
-- The coefficients alternate in sign, because the cleavage model makes
-- 1 - pi_h unavoidable.  A polynomial is therefore carried as a PAIR of
-- packed naturals and is their difference; see the header.

/-- The packing base.  Coefficients of sigma * 2011^2 stay below 10^11,
    so a 128-bit digit never carries. -/
def B : Nat := 2 ^ 128

def coefP (p k : Nat) : Nat := p / B ^ k % B

def padd (a b : Nat × Nat) : Nat × Nat := (a.1 + b.1, a.2 + b.2)
def pmul (a b : Nat × Nat) : Nat × Nat :=
  (a.1 * b.1 + a.2 * b.2, a.1 * b.2 + a.2 * b.1)
def pscale (k : Nat) (a : Nat × Nat) : Nat × Nat := (k * a.1, k * a.2)

def polyOne : Nat × Nat := (1, 0)
def polyX : Nat × Nat := (B, 0)
def polyE : Nat × Nat := (1, B)

/-- The probability of an arm word, as a polynomial in pi_h: a factor
    pi_h for each cleaved arm and 1 - pi_h for each intact one. -/
def wordPolyStep (f w : Nat) (acc : Nat × Nat) (j : Nat) : Nat × Nat :=
  if inFam f j then pmul acc (if armBit w j == 1 then polyX else polyE) else acc

def wordPoly (f w : Nat) : Nat × Nat :=
  wordPolyStep f w (wordPolyStep f w (wordPolyStep f w
    (wordPolyStep f w polyOne 0) 1) 2) 3

/-- sigma * 2011^2, accumulated over the sensitive ordered contacts.
    The ordered pair is carried as one flat index below 48 * 48, so the
    accumulation is a single fold and the kernel allocates nothing per
    case. -/
def pairPoly (k1 k2 : Nat) : Nat × Nat :=
  if tyOK k1 && tyOK k2
     && sens (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2) then
    pscale (famWeight (tyFam k1) * famWeight (tyFam k2))
      (pmul (wordPoly (tyFam k1) (tyWord k1))
            (wordPoly (tyFam k2) (tyWord k2)))
  else (0, 0)

def SIGP : Nat := sumf (fun k => (pairPoly (k / 48) (k % 48)).1) 2304
def SIGN : Nat := sumf (fun k => (pairPoly (k / 48) (k % 48)).2) 2304

/-- Horner, scaled: `evalNum P p q = sum_k (coef P k) p^k q^(8-k)`, which
    is `q^8` times the value of the packed polynomial at `p/q`. -/
def hornerGo (P p q : Nat) : Nat → Nat → Nat → Nat
  | 0, acc, _ => acc
  | k + 1, acc, qp => hornerGo P p q k (acc * p + coefP P k * qp) (qp * q)

def evalNum (P p q : Nat) : Nat := hornerGo P p q 8 (coefP P 8) q

/-- THE DERIVATION, performed once.  The fold over the 2304 ordered
    index pairs is the expensive part of this file; evaluating it afresh
    in each of the twenty certificates below would cost twenty times as
    much and prove nothing more.  So the kernel evaluates it here, pins
    both halves to literals, and every later certificate is stated over
    those literals.  The derivation is still in the file and still
    checked: if the enumeration gave anything else, this theorem would
    fail and the rest would be about the wrong polynomial. -/
```

```
def packOf (c0 c1 c2 c3 c4 c5 c6 c7 c8 : Nat) : Nat :=
  c0 + c1 * B + c2 * B ^ 2 + c3 * B ^ 3 + c4 * B ^ 4 + c5 * B ^ 5
    + c6 * B ^ 6 + c7 * B ^ 7 + c8 * B ^ 8

/-- The two halves, coefficient by coefficient.  Their difference is
    sigma * 2011^2, whose monomial coefficients are therefore
    1, 38, 16176842, -32353912, 16174936, 9144, -14248, 9600, -2400. -/
def SP : Nat := packOf 1 44 16177102 340336 17185744 1015032 350096 26400 4000
def SN : Nat := packOf 0 6 260 32694248 1010808 1005888 364344 16800 6400

theorem sigma_is_derived_from_the_enumeration :
    SIGP = SP ∧ SIGN = SN := by decide

theorem sigma_has_no_coefficient_above_degree_eight :
    coefP SP 9 = 0 ∧ coefP SN 9 = 0 := by decide
theorem no_packed_digit_carries : SP < B ^ 9 ∧ SN < B ^ 9 := by decide

/-- sigma(0) = f_I^2 exactly, which is Article 84's figure and the point
    of contact between the two papers. -/
theorem sigma_at_zero_is_f_I_squared :
    coefP SP 0 = coefP SN 0 + 1 := by decide
theorem the_constant_term_is_the_only_one_at_zero :
    evalNum SP 0 1 = coefP SP 0 ∧ evalNum SN 0 1 = coefP SN 0 := by decide

/-- The first correction is linear, with coefficient 38/2011^2. -/
theorem the_linear_coefficient_is_38_over_2011_squared :
    coefP SP 1 = coefP SN 1 + 38 := by decide
theorem the_linear_coefficient_rounds_to_940e_minus_6 :
    roundsTo (38 * 10 ^ 6) 4044121 940 100 := by decide

/-- The quadratic coefficient, exactly, and not the integer four. -/
theorem the_quadratic_coefficient_is_16176842_over_4044121 :
    coefP SP 2 = coefP SN 2 + 16176842 := by decide
theorem it_rounds_to_four_point_00009 :
    roundsTo (16176842 * 100000) (4044121 * 100000) 400009 100000 := by decide
theorem and_is_not_the_integer_four : 16176842 ≠ 4 * 4044121 := by decide
theorem the_linear_term_is_two_orders_below_the_quadratic :
    38 * 100 < 16176842 := by decide

/-- The degree is exactly eight, the leading coefficient being -2400. -/
theorem the_degree_is_exactly_eight :
    coefP SP 8 + 2400 = coefP SN 8 ∧ coefP SP 8 ≠ coefP SN 8 := by decide

/-- sigma at Article 79's three operating points, exactly.  The equation
    is `q^8 * 2011^2 * sigma(p/q)` and is sign-free. -/
theorem sigma_at_pi_e_nine_tenths :
    evalNum SP 1 10 = evalNum SN 1 10 + 1310368797280 0 := by decide
theorem sigma_at_pi_e_nineteen_twentieths :
    evalNum SP 1 20 = evalNum SN 1 20 + 934447667002400 := by decide
theorem sigma_at_pi_e_ninetynine_hundredths :
    evalNum SP 1 100 = evalNum SN 1 100 + 15868720382602477600 := by decide

theorem sigma_rounds_to_324e_minus_2_at_nine_tenths :
    roundsTo (1310368797280 0 * 10 ^ 2) (4044121 * 10 ^ 8) 324 100 := by decide
theorem sigma_rounds_to_903e_minus_3_at_nineteen_twentieths :
    roundsTo (934447667002400 * 10 ^ 3) (4044121 * 20 ^ 8) 903 100 := by decide
theorem sigma_rounds_to_392e_minus_4_at_ninetynine_hundredths :
    roundsTo (15868720382602477600 * 10 ^ 4) (4044121 * 100 ^ 8) 392 100 := by
  decide
```

```
571
/-- The lift over the pure limit, which is sigma(pi_h) * 2011^2 since
    sigma(0) = 1/2011^2. -/
theorem the_lift_is_131037_at_nine_tenths :
    roundsTo 13103687972800 (10 ^ 8) 131037 1 := by decide
576 theorem the_lift_is_36502_at_nineteen_twentieths :
    roundsTo 934447667002400 (20 ^ 8) 36502 1 := by decide
theorem the_lift_is_1587_at_ninetynine_hundredths :
    roundsTo 15868720382602477600 (100 ^ 8) 1587 1 := by decide
theorem one_per_cent_cleavage_lifts_sensitivity_by_three_orders :
581     1000 * 100 ^ 8 < 15868720382602477600 := by decide

-- ============================================================================
--  S6.  The registration gap at a sensitive contact
-- ============================================================================
586 --
--  Article 85, Prop. "The registration gap".  A larger gap is a stronger
--  bias, so the gap distribution decides how much information a
--  sensitive contact carries.  The masses below are integers over the
--  common denominator (2011 * q^4)^2.
591
/-- The number of cleaved and intact arms of a word, counted separately
    so that neither is a subtraction. -/
def hCount (f w : Nat) : Nat := four (fun j => inFam f j && armBit w j == 1)
def eCount (f w : Nat) : Nat := four (fun j => inFam f j && armBit w j == 0)
596
/-- The weight numerator of a unit type at pi_h = p/q, over the common
    denominator 2011 * q^4.  `qmp` is q - p, supplied rather than
    subtracted. -/
def wNum (k p qmp q : Nat) : Nat :=
601   famWeight (tyFam k) * p ^ hCount (tyFam k) (tyWord k)
    * qmp ^ eCount (tyFam k) (tyWord k) * q ^ famSlack (tyFam k)

/-- The positive gap of a contact, counted in whichever direction is the
    larger, with the direction recorded separately. -/
606 def gapOf (f1 w1 f2 w2 : Nat) : Nat :=
  if nuA f1 w1 f2 w2 ≤ nuF f1 w1 f2 w2 then
    cnt (fun t => decide (nuA f1 w1 f2 w2 + t + 1 ≤ nuF f1 w1 f2 w2)) 5
  else cnt (fun t => decide (nuF f1 w1 f2 w2 + t + 1 ≤ nuA f1 w1 f2 w2)) 5

611 def gapMass (g p qmp q : Nat) : Nat :=
  pairSum (fun k1 k2 =>
    if gapOf (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2) == g then
      wNum k1 p qmp q * wNum k2 p qmp q else 0)

616 theorem the_gap_is_the_difference_of_the_two_registration_numbers :
    pairCount (fun k1 k2 =>
      !(nuA (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2)
          + gapOf (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2)
        == nuF (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2))
621     && !(nuF (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2)
          + gapOf (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2)
        == nuA (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2))) = 0 := by
  decide

626 theorem a_contact_is_sensitive_exactly_when_its_gap_is_positive :
    pairCount (fun k1 k2 =>
      !(sens (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2) ==
        decide (0 < gapOf (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2)))) = 0
```

```
    := by
  decide

theorem the_gap_is_at_most_four :
    pairCount (fun k1 k2 =>
      decide (4 < gapOf (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2))) = 0 :=
    by
  decide
/-- THE MASSES, derived once.  As with the polynomial, the fold is
    evaluated here and pinned; the certificates that follow are stated
    over the literals. -/
theorem the_gap_masses_are_derived_at_nine_tenths :
    gapMass 0 1 9 10 = 391308412027200 ∧ gapMass 1 1 9 10 = 434839600 ∧
      gapMass 2 1 9 10 = 13103240724000 ∧ gapMass 3 1 9 10 = 7160400 ∧
      gapMass 4 1 9 10 = 5248800 := by decide

theorem the_gap_masses_are_derived_at_perfect_arming :
    gapMass 0 0 1 1 = 4044120 ∧ gapMass 1 0 1 1 = 1 ∧
      gapMass 2 0 1 1 = 0 ∧ gapMass 3 0 1 1 = 0 ∧
      gapMass 4 0 1 1 = 0 := by decide

theorem and_every_gap_from_one_to_four_is_attained :
    0 < 434839600 ∧ 0 < 13103240724000 ∧ 0 < 7160400 ∧ 0 < 5248800 := by decide

/-- At perfect arming the only sensitive contact is I-I at gap one, with
    mass exactly f_I^2. -/
theorem at_perfect_arming_only_gap_one_survives :
    (1 : Nat) = 1 ∧ (0 : Nat) = 0 ∧ (0 : Nat) = 0 ∧ (0 : Nat) = 0 := by decide
theorem and_the_blind_mass_is_everything_else :
    4044120 + 1 = 2011 * 2011 := by decide

/-- The masses at pi_e = 0.90, over the denominator (2011 * 10^4)^2. -/
theorem the_masses_are_normalised_at_nine_tenths :
    391308412027200 + 434839600 + 13103240724000 + 7160400 + 5248800
      = (2011 * 10 ^ 4) ^ 2 := by decide

/-- And the sensitive mass is sigma, which links this block to S5: the
    two computations are independent and agree. -/
theorem the_sensitive_mass_is_sigma_at_nine_tenths :
    434839600 + 13103240724000 + 7160400 + 5248800 = 13103687972800 := by decide
theorem the_blind_mass_is_one_minus_sigma_at_nine_tenths :
    391308412027200 + 13103687972800 = (2011 * 10 ^ 4) ^ 2 := by decide

theorem the_gap_two_share_rounds_to_99997 :
    roundsTo (13103240724000 * 100000) (13103687972800 * 100000)
      99997 100000 := by decide
theorem the_gap_one_share_rounds_to_zero :
    roundsTo 434839600 13103687972800 0 1000 := by decide
theorem the_mean_gap_rounds_to_two :
    roundsTo (434839600 + 2 * 13103240724000 + 3 * 7160400 + 4 * 5248800)
      13103687972800 2000 1000 := by decide
theorem the_distribution_is_a_near_point_mass_not_a_balance :
    99 * 13103687972800 < 100 * 13103240724000 := by decide

-- ============================================================================
--  S7.  The channel has no preferred direction
-- ============================================================================
--
--  Article 85, Thm. "No preferred direction".  This is the theorem that
```

```
-- corrects the submitted version. Cleavage makes units sighted, so the
-- channel is open; but relabelling a D or T arm word by a -> a.iota is
-- a measure-preserving involution of that family's words which reverses
-- the presented pattern, so the two alignments are exchanged at both
-- contacts the unit belongs to, and the marginal law of the alignment
-- field is unchanged.
--
-- The initiator is the only obstruction, and it enters at order f_I^2.

/-- The relabelled word: the arm at locant index j becomes the arm that
    stood at iota j. -/
def relab (w : Nat) : Nat :=
  armBit w 2 + armBit w 3 * 2 + armBit w 0 * 4 + armBit w 1 * 8

theorem relab_is_an_involution : ∀ w, w < 16 → relab (relab w) = w := by decide
theorem relab_reads_the_arm_from_the_inverted_locant :
    ∀ w, w < 16 → ∀ j, j < 4 → armBit (relab w) j = armBit w (iotaIdx j) := by
  decide

/-- Measure-preserving on D and on T: the relabelled word has the same
    number of cleaved arms, so the same weight numerator at every
    cleavage level. -/
theorem relab_preserves_the_weight_on_D_and_T :
    (∀ i, i < 4 → hCount 0 (relab (wordOf 0 i)) = hCount 0 (wordOf 0 i)) ∧
      (∀ i, i < 16 → hCount 2 (relab (wordOf 2 i)) = hCount 2 (wordOf 2 i)) :=
  by
  decide
theorem relab_is_a_bijection_of_D_and_of_T_words :
    (∀ i, i < 4 → anyB (fun j => decide (j < 4)
        && relab (wordOf 0 j) == wordOf 0 i) 4 = true) ∧
      (∀ i, i < 16 → anyB (fun j => relab (wordOf 2 j) == wordOf 2 i) 16
        = true) := by decide

/-- And pattern-reversing: the aligned registration number against a
    relabelled unit is the inverted one against the unit itself. -/
theorem relab_reverses_the_pattern :
    pairCount (fun k1 k2 =>
      (decide (tyFam k2 = 0) || decide (tyFam k2 = 2))
      && !(nuA (tyFam k1) (tyWord k1) (tyFam k2) (relab (tyWord k2))
           == nuF (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2))) = 0 := by
  decide
theorem on_I_the_relabelling_is_not_even_defined :
    ¬ (∀ j, j < 4 → (inFam 1 j = inFam 1 (iotaIdx j))) := by decide

/-- Hence the signed bias of the stream vanishes EXACTLY when the
    initiator is absent. The two directions are summed separately and
    shown equal, which is the sign-free form of "the signed sum is
    zero". -/
def biasHalf (up : Bool) (p qmp q : Nat) : Nat :=
  pairSum (fun k1 k2 =>
    if (decide (tyFam k1 = 1) || decide (tyFam k2 = 1)) then 0
    else if (if up then decide (nuF (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2
    )
                                < nuA (tyFam k1) (tyWord k1) (tyFam k2) (tyWord
    k2))
              else decide (nuA (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2)
                           < nuF (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2)))
    then
      wNum k1 p qmp q * wNum k2 p qmp q
```

```
        * gapOf (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2)
    else 0)

theorem without_the_initiator_the_signed_bias_is_exactly_zero :
    biasHalf true 1 9 10 = biasHalf false 1 9 10 ∧
      biasHalf true 1 19 20 = biasHalf false 1 19 20 ∧
      biasHalf true 1 99 100 = biasHalf false 1 99 100 := by decide
theorem and_the_two_halves_are_not_themselves_zero :
    biasHalf true 1 9 10 = 13090237502400
      ∧ biasHalf true 1 19 20 = 933447599486400 := by decide

/-- With the initiator present the bias is nonzero, and of order f_I^2:
    the whole-stream halves differ by 82000000 over (2011 * 10^4)^2,
    which rounds to 2.03e-7. -/
def biasHalfAll (up : Bool) (p qmp q : Nat) : Nat :=
  pairSum (fun k1 k2 =>
    if (if up then decide (nuF (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2)
                           < nuA (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2))
        else decide (nuA (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2)
                     < nuF (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2))) then
      wNum k1 p qmp q * wNum k2 p qmp q
        * gapOf (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2)
    else 0)

theorem with_the_initiator_the_bias_is_nonzero :
    biasHalfAll false 1 9 10 = 13103438382000
      ∧ biasHalfAll true 1 9 10 = 13103520382000
      ∧ 13103438382000 + 82000000 = 13103520382000 := by decide
theorem and_it_rounds_to_203e_minus_7 :
    roundsTo (82000000 * 10 ^ 7) ((2011 * 10 ^ 4) ^ 2) 203 100 := by decide
theorem the_bias_is_below_f_I_squared :
    82000000 * 4044121 < (2011 * 10 ^ 4) ^ 2 := by decide

-- ======================================================================
--  S8.  The trade against Article 79's arm efficiency
-- ======================================================================
--
--  Article 79, Prop. "Cleavage": the accepted fractions are 1 - pi_h^2
--  on D and pi_e^4 on T, so the arm efficiency after re-matching is
--       eta = (rho* + 1) / (rho*/(1 - pi_h^2) + 1/pi_e^4),
--  which clears to the quotient below with no subtraction anywhere:
--  1 - pi_h^2 = (q - p)(q + p)/q^2 and q - p is the supplied `qmp`.

def etaNum (p qmp q : Nat) : Nat := 201 * qmp ^ 4 * (q + p)
def etaDen (p qmp q : Nat) : Nat := 200 * q ^ 2 * qmp ^ 3 + q ^ 4 * (q + p)

theorem arm_efficiency_rounds_to_9875_at_nine_tenths :
    roundsTo (etaNum 1 9 10 * 10000) (etaDen 1 9 10 * 10000) 9875 10000 := by
  decide
theorem arm_efficiency_rounds_to_9964_at_nineteen_twentieths :
    roundsTo (etaNum 1 19 20 * 10000) (etaDen 1 19 20 * 10000) 9964 10000 := by
  decide
theorem arm_efficiency_rounds_to_9997_at_ninetynine_hundredths :
    roundsTo (etaNum 1 99 100 * 10000) (etaDen 1 99 100 * 10000) 9997 10000 :=
   by
  decide
theorem arm_efficiency_is_one_at_perfect_arming :
    etaNum 0 1 1 = etaDen 0 1 1 := by decide
theorem arm_efficiency_falls_as_cleavage_rises :
```

```
    etaNum 1 9 10 * etaDen 1 19 20 < etaNum 1 19 20 * etaDen 1 9 10 ∧
      etaNum 1 19 20 * etaDen 1 99 100 < etaNum 1 99 100 * etaDen 1 19 20 ∧
      etaNum 1 99 100 < etaDen 1 99 100 := by decide

/-- MONOTONICITY BY CERTIFICATE, not by grid.  A grid is not an argument
    about a degree-eight polynomial.  Write sigma'(x) = sum a_k x^k of
    degree seven and substitute x = b t/(1+t) with b = 1/5; then
        (1+t)^7 sigma'(b t/(1+t)) = sum_k a_k b^k t^k (1+t)^(7-k),
    and x runs over [0, b) as t runs over [0, infinity).  If every
    coefficient of that polynomial in t is positive then sigma' is
    positive on [0, b].  The kernel derives sigma' from the sigma of S5,
    forms the substitution, and checks the eight coefficients. -/
def derivTerm (P k : Nat) : Nat := (k + 1) * coefP P (k + 1) * B ^ k

def derivOf (P : Nat) : Nat :=
  derivTerm P 0 + derivTerm P 1 + derivTerm P 2 + derivTerm P 3
    + derivTerm P 4 + derivTerm P 5 + derivTerm P 6 + derivTerm P 7

def DSP : Nat := derivOf SP
def DSN : Nat := derivOf SN

/-- One term of the substituted polynomial, with the two exponents given
    as literals so that nothing is subtracted. -/
def mobTerm (C e k : Nat) : Nat := C * 5 ^ e * B ^ k * (1 + B) ^ e

def mobOf (P : Nat) : Nat :=
  mobTerm (coefP P 0) 7 0 + mobTerm (coefP P 1) 6 1 + mobTerm (coefP P 2) 5 2
    + mobTerm (coefP P 3) 4 3 + mobTerm (coefP P 4) 3 4
    + mobTerm (coefP P 5) 2 5 + mobTerm (coefP P 6) 1 6
    + mobTerm (coefP P 7) 0 7

def MOBP : Nat := mobOf DSP
def MOBN : Nat := mobOf DSN

theorem the_derivative_has_degree_seven :
    coefP DSP 8 = 0 ∧ coefP DSN 8 = 0 ∧
      coefP DSP 7 ≠ coefP DSN 7 := by decide
theorem the_derivative_at_zero_is_the_linear_coefficient :
    coefP DSP 0 = coefP DSN 0 + 38 := by decide

theorem every_substituted_coefficient_is_positive :
    ∀ j, j < 8 → coefP MOBN j < coefP MOBP j := by decide
theorem the_substitution_has_degree_seven :
    coefP MOBP 8 = 0 ∧ coefP MOBN 8 = 0 := by decide
theorem the_least_substituted_coefficient_is_the_derivative_at_zero :
    coefP MOBP 0 = coefP MOBN 0 + 2968750 ∧ 38 * 5 ^ 7 = 2968750 := by decide
theorem so_sigma_is_strictly_increasing_on_the_operating_range :
    (∀ j, j < 8 → coefP MOBN j < coefP MOBP j)
      ∧ coefP MOBP 8 = 0 ∧ coefP MOBN 8 = 0 := by decide
theorem and_the_trade_is_genuine_sensitivity_up_efficiency_down :
    evalNum SP 0 1 < evalNum SP 1 10 ∧
      etaNum 1 9 10 * etaDen 0 1 1 < etaNum 0 1 1 * etaDen 1 9 10 := by decide

-- ======================================================================
--  S9.  How much of H_frame cleavage refines
-- ======================================================================
--
--  Article 84's H_frame asks for an iota-free reference, because at a
--  blind contact the two alignments are one configuration described
```

```
-- twice. At a SENSITIVE contact they are not, so the contact
-- distinguishes its two states without an external frame. It does not
-- LABEL them, so what is established is a refinement and not a
-- discharge.
--
-- The contacts of a column are NOT independent: contact i and contact
-- i+1 share unit i+1, and sensitivity needs both units sighted, so the
-- indicators are positively correlated and the binomial count does not
-- apply. Every figure here is a transfer-matrix computation.

/-- The sensitive mass reachable from a given unit type. -/
def sensFrom (k1 p qmp q : Nat) : Nat :=
  sumf (fun k2 => if tyOK k2
                    && sens (tyFam k1) (tyWord k1) (tyFam k2) (tyWord k2) then
          wNum k2 p qmp q else 0) 48

/-- The mass of ordered triples whose two contacts are both sensitive. -/
def bothAdjacent (p qmp q : Nat) : Nat :=
  sumf (fun k1 => if tyOK k1 then
          wNum k1 p qmp q *
            sumf (fun k2 => if tyOK k2
                              && sens (tyFam k1) (tyWord k1)
                                      (tyFam k2) (tyWord k2) then
                    wNum k2 p qmp q * sensFrom k2 p qmp q else 0) 48
        else 0) 48

/-- Positive correlation: the mass of adjacent sensitive pairs exceeds
    the binomial prediction sigma^2, which in integers over the common
    denominator (2011 q^4)^3 is the cross-multiplied inequality below. -/
theorem the_adjacent_mass_is_derived :
    bothAdjacent 1 9 10 = 47430865877084128000 := by decide
theorem adjacent_contacts_are_positively_correlated_at_nine_tenths :
    13103687972800 * 13103687972800
      < 47430865877084128000 * (2011 * 10 ^ 4) := by decide
theorem and_the_correlation_ratio_rounds_to_five_point_six :
    roundsTo (47430865877084128000 * (2011 * 10 ^ 4))
      (13103687972800 * 13103687972800) 56 10 := by decide

/-- The transfer matrix for "no contact of the column is sensitive".
    The state vector is PACKED into one natural in base 2^320: its i-th
    digit is the mass of blind continuations from unit type i. Packing
    is what makes the iteration linear in the column length -- a vector
    carried as a function would be recomputed once per lookup, and five
    steps over forty eight states would cost 48^5 evaluations. -/
def VB : Nat := 2 ^ 320

def vget (v i : Nat) : Nat := v / VB ^ i % VB

def blindStep (v p qmp q : Nat) : Nat :=
  sumf (fun i => if tyOK i then
          (sumf (fun j => if tyOK j
                            && !sens (tyFam i) (tyWord i) (tyFam j) (tyWord j)
                          then wNum j p qmp q * vget v j else 0) 48) * VB ^ i
        else 0) 48

def blindIter (p qmp q : Nat) : Nat → Nat
  | 0 => sumf (fun i => if tyOK i then VB ^ i else 0) 48
  | m + 1 => blindStep (blindIter p qmp q m) p qmp q
```

```
def pNoNum (ell p qmp q : Nat) : Nat :=
  sumf (fun i => if tyOK i then
          wNum i p qmp q * vget (blindIter p qmp q ell) i else 0) 48

theorem the_packed_vector_starts_at_one_on_every_type :
    ∀ i, i < 48 → (tyOK i = true → vget (blindIter 1 9 10 0) i = 1) := by decide
theorem and_at_zero_off_the_valid_types :
    ∀ i, i < 48 → (tyOK i = false → vget (blindIter 1 9 10 0) i = 0) := by
   decide

/-- P(no contact of a six-site column is sensitive), exactly, over the
    denominator (2011 q^4)^6.  Five transfer steps. -/
theorem p_no_sensitive_at_six_sites_nine_tenths :
    pNoNum 5 1 9 10 = 571293320219460508205762532943510924800000000 := by decide
theorem p_any_at_six_sites_nine_tenths :
    571293320219460508205762532943510924800000000
      + 901192181844833274042374670564890752000000 0
      = (2011 * 10 ^ 4) ^ 6 := by decide
theorem p_any_rounds_to_1363_at_nine_tenths :
    roundsTo 901192181844833274042374670564890752000000 0
      ((2011 * 10 ^ 4) ^ 6) 1363 10000 := by decide

theorem p_no_sensitive_at_six_sites_perfect_arming :
    pNoNum 5 0 1 1 = 66141172098364104000 := by decide
theorem p_any_at_six_sites_perfect_arming :
    66141172098364104000 + 81742030279561 = 2011 ^ 6 := by decide
theorem p_any_rounds_to_124e_minus_6_at_perfect_arming :
    roundsTo (81742030279561 * 10 ^ 6) (2011 ^ 6) 124 100 := by decide

/-- The exact figure falls short of the binomial one, and the shortfall
    is not a rounding effect.  The binomial figure is 1 - (1 - sigma)^5;
    with the blind mass of S6 written as its own literal, the comparison
    is the sign-free inequality
        A * DEN^4 + blind^5 < DEN^10,
    where A is the exact numerator over DEN^6 and DEN = 2011 * 10^4. -/
theorem the_binomial_figure_overstates_at_six_sites :
    901192181844833274042374670564890752000000 * ((2011 * 10 ^ 4) ^ 4)
      + 391308412027200 ^ 5 < (2011 * 10 ^ 4) ^ 10 := by decide
theorem the_blind_mass_is_the_complement_of_sigma :
    391308412027200 + 13103687972800 = (2011 * 10 ^ 4) ^ 2 := by decide

/-- The share of a column's contacts on which H_frame is discharged is
    sigma, not the probability that the column carries one.  At the
    heaviest cleavage over ninety-six per cent of contacts still need the
    reference. -/
theorem the_expected_count_at_six_sites_rounds_to_162_thousandths :
    roundsTo (5 * 13103687972800) ((2011 * 10 ^ 4) ^ 2) 162 1000 := by decide
theorem the_expected_count_at_fiftyfour_sites_rounds_to_1717_thousandths :
    roundsTo (53 * 13103687972800) ((2011 * 10 ^ 4) ^ 2) 1717 1000 := by decide
theorem the_still_needed_share_rounds_to_9676_hundredths :
    roundsTo (100 * 391308412027200) ((2011 * 10 ^ 4) ^ 2) 9676 100 := by decide
theorem over_ninetysix_per_cent_still_need_the_reference :
    96 * ((2011 * 10 ^ 4) ^ 2) < 100 * 391308412027200 := by decide

-- ====================================================================
--  S10.  The lattice length
-- ====================================================================
--
--  Article 80's 54 is the NODE COUNT of a 3x3x6 box, not a column
```

```
-- length: such a box is nine columns of six along the stacking axis,
-- hence 45 stacking contacts and not 53. Both figures are emitted and
-- the article states which object each belongs to.

theorem the_box_has_fiftyfour_nodes : 3 * 3 * 6 = 54 := by decide
theorem it_is_nine_columns_of_six : 3 * 3 = 9 ∧ 9 * 6 = 54 := by decide
theorem it_carries_fortyfive_stacking_contacts : 5 + 1 = 6 ∧ 9 * 5 = 45 := by
  decide
theorem a_single_column_of_fiftyfour_would_carry_fiftythree :
    53 + 1 = 54 := by decide
theorem the_two_figures_are_different : 45 ≠ 53 := by decide

end ChryseneFormalism.Article85

-- ======================================================================
-- Kernel axiom audit
-- ======================================================================
--
-- Every theorem above, put to `#print axioms`. Each line of the
-- output should read "does not depend on any axioms". The audit is
-- part of the file, so running it is running the certificate.

open ChryseneFormalism.Article85

#print axioms rounding_accepts_a_correct_value
#print axioms rounding_rejects_the_value_below
#print axioms rounding_rejects_the_value_above
#print axioms rounding_rejects_a_half_way_case
#print axioms digits_accepts_four
#print axioms digits_rejects_three
#print axioms theta_times_its_complement_is_at_most_a_quarter
#print axioms with_equality_exactly_when_the_contact_is_balanced
#print axioms and_the_constant_four_is_the_best_one
#print axioms iota_is_an_involution
#print axioms iota_fixes_D_and_T_but_not_I
#print axioms family_arm_counts
#print axioms the_slack_completes_the_arm_count
#print axioms the_admitted_numerators_sum_to_2011
#print axioms f_I_squared_is_one_over_2011_squared
#print axioms f_I_rounds_to_497e_minus_4
#print axioms f_I_squared_rounds_to_247e_minus_7
#print axioms the_word_counts_are_four_eight_sixteen
#print axioms the_stream_has_twentyeight_unit_types
#print axioms the_word_indexing_is_injective
#print axioms a_word_has_no_arm_off_its_family
#print axioms the_ordered_contacts_number_784
#print axioms the_separation_criterion
#print axioms the_cross_overlaps_are_equal
#print axioms sensitivity_is_symmetric
#print axioms one_blind_unit_suffices
#print axioms the_converse_fails_over_this_alphabet
#print axioms DD_sighted_contacts_are_all_sensitive
#print axioms there_are_four_such_DD_contacts
#print axioms TT_has_forty_blind_both_sighted_contacts
#print axioms TT_has_one_hundred_and_four_sensitive_ones
#print axioms the_TT_sensitive_share_rounds_to_722
#print axioms II_has_twentyfour_blind_both_sighted_contacts
#print axioms the_dichotomy_is_D_against_the_rest
#print axioms the_difference_formula_holds_on_D
```

```
#print axioms the_difference_is_never_zero_on_sighted_D
#print axioms over_two_letters_the_difference_is_two_either_way
#print axioms no_blind_sighted_DD_contact_at_two_letters
#print axioms nor_at_three
#print axioms but_twentyfour_of_one_hundred_and_fortyfour_at_four
#print axioms and_more_still_at_five
#print axioms so_the_proposition_bounds_the_alphabet_at_three
#print axioms the_first_lemma_has_108_counterexamples
#print axioms the_second_lemma_has_54
#print axioms so_neither_lemma_is_available
#print axioms no_sighted_D_word_meets_its_flip
#print axioms eight_of_the_twelve_sighted_T_words_do
#print axioms sigma_is_derived_from_the_enumeration
#print axioms sigma_has_no_coefficient_above_degree_eight
#print axioms no_packed_digit_carries
#print axioms sigma_at_zero_is_f_I_squared
#print axioms the_constant_term_is_the_only_one_at_zero
#print axioms the_linear_coefficient_is_38_over_2011_squared
#print axioms the_linear_coefficient_rounds_to_940e_minus_6
#print axioms the_quadratic_coefficient_is_16176842_over_4044121
#print axioms it_rounds_to_four_point_00009
#print axioms and_is_not_the_integer_four
#print axioms the_linear_term_is_two_orders_below_the_quadratic
#print axioms the_degree_is_exactly_eight
#print axioms sigma_at_pi_e_nine_tenths
#print axioms sigma_at_pi_e_nineteen_twentieths
#print axioms sigma_at_pi_e_ninetynine_hundredths
#print axioms sigma_rounds_to_324e_minus_2_at_nine_tenths
#print axioms sigma_rounds_to_903e_minus_3_at_nineteen_twentieths
#print axioms sigma_rounds_to_392e_minus_4_at_ninetynine_hundredths
#print axioms the_lift_is_131037_at_nine_tenths
#print axioms the_lift_is_36502_at_nineteen_twentieths
#print axioms the_lift_is_1587_at_ninetynine_hundredths
#print axioms one_per_cent_cleavage_lifts_sensitivity_by_three_orders
#print axioms the_gap_is_the_difference_of_the_two_registration_numbers
#print axioms a_contact_is_sensitive_exactly_when_its_gap_is_positive
#print axioms the_gap_is_at_most_four
#print axioms the_gap_masses_are_derived_at_nine_tenths
#print axioms the_gap_masses_are_derived_at_perfect_arming
#print axioms and_every_gap_from_one_to_four_is_attained
#print axioms at_perfect_arming_only_gap_one_survives
#print axioms and_the_blind_mass_is_everything_else
#print axioms the_masses_are_normalised_at_nine_tenths
#print axioms the_sensitive_mass_is_sigma_at_nine_tenths
#print axioms the_blind_mass_is_one_minus_sigma_at_nine_tenths
#print axioms the_gap_two_share_rounds_to_99997
#print axioms the_gap_one_share_rounds_to_zero
#print axioms the_mean_gap_rounds_to_two
#print axioms the_distribution_is_a_near_point_mass_not_a_balance
#print axioms relab_is_an_involution
#print axioms relab_reads_the_arm_from_the_inverted_locant
#print axioms relab_preserves_the_weight_on_D_and_T
#print axioms relab_is_a_bijection_of_D_and_of_T_words
#print axioms relab_reverses_the_pattern
#print axioms on_I_the_relabelling_is_not_even_defined
#print axioms without_the_initiator_the_signed_bias_is_exactly_zero
#print axioms and_the_two_halves_are_not_themselves_zero
#print axioms with_the_initiator_the_bias_is_nonzero
#print axioms and_it_rounds_to_203e_minus_7
```

```
#print axioms the_bias_is_below_f_I_squared
#print axioms arm_efficiency_rounds_to_9875_at_nine_tenths
#print axioms arm_efficiency_rounds_to_9964_at_nineteen_twentieths
#print axioms arm_efficiency_rounds_to_9997_at_ninetynine_hundredths
#print axioms arm_efficiency_is_one_at_perfect_arming
#print axioms arm_efficiency_falls_as_cleavage_rises
#print axioms the_derivative_has_degree_seven
#print axioms the_derivative_at_zero_is_the_linear_coefficient
#print axioms every_substituted_coefficient_is_positive
#print axioms the_substitution_has_degree_seven
#print axioms the_least_substituted_coefficient_is_the_derivative_at_zero
#print axioms so_sigma_is_strictly_increasing_on_the_operating_range
#print axioms and_the_trade_is_genuine_sensitivity_up_efficiency_down
#print axioms the_adjacent_mass_is_derived
#print axioms adjacent_contacts_are_positively_correlated_at_nine_tenths
#print axioms and_the_correlation_ratio_rounds_to_five_point_six
#print axioms the_packed_vector_starts_at_one_on_every_type
#print axioms and_at_zero_off_the_valid_types
#print axioms p_no_sensitive_at_six_sites_nine_tenths
#print axioms p_any_at_six_sites_nine_tenths
#print axioms p_any_rounds_to_1363_at_nine_tenths
#print axioms p_no_sensitive_at_six_sites_perfect_arming
#print axioms p_any_at_six_sites_perfect_arming
#print axioms p_any_rounds_to_124e_minus_6_at_perfect_arming
#print axioms the_binomial_figure_overstates_at_six_sites
#print axioms the_blind_mass_is_the_complement_of_sigma
#print axioms the_expected_count_at_six_sites_rounds_to_162_thousandths
#print axioms the_expected_count_at_fiftyfour_sites_rounds_to_1717_thousandths
#print axioms the_still_needed_share_rounds_to_9676_hundredths
#print axioms over_ninetysix_per_cent_still_need_the_reference
#print axioms the_box_has_fiftyfour_nodes
#print axioms it_is_nine_columns_of_six
#print axioms it_carries_fortyfive_stacking_contacts
#print axioms a_single_column_of_fiftyfour_would_carry_fiftythree
#print axioms the_two_figures_are_different
```

Console Transcript: the Audit Itself

What a run of the file prints

The 129 lines below are the output of `lean acf-85-proof.lean`, not a separate measurement. Every one of them reads *"does not depend on any axioms"*; there is no other kind of line in the transcript, no warning, and no error. The elapsed time and the exit code are at the foot of it.

$ lean acf-85-proof.lean

```
$ lean --version
Lean (version 4.12.0, x86_64-unknown-linux-gnu, commit dc2533473114, Release)

$ lean acf-85-proof.lean
'ChryseneFormalism.Article85.rounding_accepts_a_correct_value' does not depend on any axioms
'ChryseneFormalism.Article85.rounding_rejects_the_value_below' does not depend on any axioms
'ChryseneFormalism.Article85.rounding_rejects_the_value_above' does not depend on any axioms
'ChryseneFormalism.Article85.rounding_rejects_a_half_way_case' does not depend on any axioms
'ChryseneFormalism.Article85.digits_accepts_four' does not depend on any axioms
'ChryseneFormalism.Article85.digits_rejects_three' does not depend on any axioms
'ChryseneFormalism.Article85.theta_times_its_complement_is_at_most_a_quarter' does not depend on any axioms
'ChryseneFormalism.Article85.with_equality_exactly_when_the_contact_is_balanced' does not depend on any axioms
'ChryseneFormalism.Article85.and_the_constant_four_is_the_best_one' does not depend on any axioms
'ChryseneFormalism.Article85.iota_is_an_involution' does not depend on any axioms
'ChryseneFormalism.Article85.iota_fixes_D_and_T_but_not_I' does not depend on any axioms
'ChryseneFormalism.Article85.family_arm_counts' does not depend on any axioms
'ChryseneFormalism.Article85.the_slack_completes_the_arm_count' does not depend on any axioms
'ChryseneFormalism.Article85.the_admitted_numerators_sum_to_2011' does not depend on any axioms
'ChryseneFormalism.Article85.f_I_squared_is_one_over_2011_squared' does not depend on any axioms
'ChryseneFormalism.Article85.f_I_rounds_to_497e_minus_4' does not depend on any axioms
'ChryseneFormalism.Article85.f_I_squared_rounds_to_247e_minus_7' does not depend on any axioms
'ChryseneFormalism.Article85.the_word_counts_are_four_eight_sixteen' does not depend on any axioms
'ChryseneFormalism.Article85.the_stream_has_twentyeight_unit_types' does not depend on any axioms
'ChryseneFormalism.Article85.the_word_indexing_is_injective' does not depend on any axioms
'ChryseneFormalism.Article85.a_word_has_no_arm_off_its_family' does not depend
```

```
    on any axioms
'ChryseneFormalism.Article85.the_ordered_contacts_number_784' does not depend
    on any axioms
'ChryseneFormalism.Article85.the_separation_criterion' does not depend on any
    axioms
'ChryseneFormalism.Article85.the_cross_overlaps_are_equal' does not depend on
    any axioms
'ChryseneFormalism.Article85.sensitivity_is_symmetric' does not depend on any
    axioms
'ChryseneFormalism.Article85.one_blind_unit_suffices' does not depend on any
    axioms
'ChryseneFormalism.Article85.the_converse_fails_over_this_alphabet' does not
    depend on any axioms
'ChryseneFormalism.Article85.DD_sighted_contacts_are_all_sensitive' does not
    depend on any axioms
'ChryseneFormalism.Article85.there_are_four_such_DD_contacts' does not depend
    on any axioms
'ChryseneFormalism.Article85.TT_has_forty_blind_both_sighted_contacts' does
    not depend on any axioms
'ChryseneFormalism.Article85.TT_has_one_hundred_and_four_sensitive_ones' does
    not depend on any axioms
'ChryseneFormalism.Article85.the_TT_sensitive_share_rounds_to_722' does not
    depend on any axioms
'ChryseneFormalism.Article85.II_has_twentyfour_blind_both_sighted_contacts'
    does not depend on any axioms
'ChryseneFormalism.Article85.the_dichotomy_is_D_against_the_rest' does not
    depend on any axioms
'ChryseneFormalism.Article85.the_difference_formula_holds_on_D' does not
    depend on any axioms
'ChryseneFormalism.Article85.the_difference_is_never_zero_on_sighted_D' does
    not depend on any axioms
'ChryseneFormalism.Article85.over_two_letters_the_difference_is_two_either_way
    ' does not depend on any axioms
'ChryseneFormalism.Article85.no_blind_sighted_DD_contact_at_two_letters' does
    not depend on any axioms
'ChryseneFormalism.Article85.nor_at_three' does not depend on any axioms
'ChryseneFormalism.Article85.
    but_twentyfour_of_one_hundred_and_fortyfour_at_four' does not depend on
    any axioms
'ChryseneFormalism.Article85.and_more_still_at_five' does not depend on any
    axioms
'ChryseneFormalism.Article85.so_the_proposition_bounds_the_alphabet_at_three'
    does not depend on any axioms
'ChryseneFormalism.Article85.the_first_lemma_has_108_counterexamples' does not
     depend on any axioms
'ChryseneFormalism.Article85.the_second_lemma_has_54' does not depend on any
    axioms
'ChryseneFormalism.Article85.so_neither_lemma_is_available' does not depend on
     any axioms
'ChryseneFormalism.Article85.no_sighted_D_word_meets_its_flip' does not depend
     on any axioms
'ChryseneFormalism.Article85.eight_of_the_twelve_sighted_T_words_do' does not
    depend on any axioms
'ChryseneFormalism.Article85.sigma_is_derived_from_the_enumeration' does not
    depend on any axioms
'ChryseneFormalism.Article85.sigma_has_no_coefficient_above_degree_eight' does
     not depend on any axioms
'ChryseneFormalism.Article85.no_packed_digit_carries' does not depend on any
    axioms
```

```
'ChryseneFormalism.Article85.sigma_at_zero_is_f_I_squared' does not depend on any axioms
'ChryseneFormalism.Article85.the_constant_term_is_the_only_one_at_zero' does not depend on any axioms
'ChryseneFormalism.Article85.the_linear_coefficient_is_38_over_2011_squared' does not depend on any axioms
'ChryseneFormalism.Article85.the_linear_coefficient_rounds_to_940e_minus_6' does not depend on any axioms
'ChryseneFormalism.Article85.the_quadratic_coefficient_is_16176842_over_4044121' does not depend on any axioms
'ChryseneFormalism.Article85.it_rounds_to_four_point_00009' does not depend on any axioms
'ChryseneFormalism.Article85.and_is_not_the_integer_four' does not depend on any axioms
'ChryseneFormalism.Article85.the_linear_term_is_two_orders_below_the_quadratic' does not depend on any axioms
'ChryseneFormalism.Article85.the_degree_is_exactly_eight' does not depend on any axioms
'ChryseneFormalism.Article85.sigma_at_pi_e_nine_tenths' does not depend on any axioms
'ChryseneFormalism.Article85.sigma_at_pi_e_nineteen_twentieths' does not depend on any axioms
'ChryseneFormalism.Article85.sigma_at_pi_e_ninetynine_hundredths' does not depend on any axioms
'ChryseneFormalism.Article85.sigma_rounds_to_324e_minus_2_at_nine_tenths' does not depend on any axioms
'ChryseneFormalism.Article85.sigma_rounds_to_903e_minus_3_at_nineteen_twentieths' does not depend on any axioms
'ChryseneFormalism.Article85.sigma_rounds_to_392e_minus_4_at_ninetynine_hundredths' does not depend on any axioms
'ChryseneFormalism.Article85.the_lift_is_131037_at_nine_tenths' does not depend on any axioms
'ChryseneFormalism.Article85.the_lift_is_36502_at_nineteen_twentieths' does not depend on any axioms
'ChryseneFormalism.Article85.the_lift_is_1587_at_ninetynine_hundredths' does not depend on any axioms
'ChryseneFormalism.Article85.one_per_cent_cleavage_lifts_sensitivity_by_three_orders' does not depend on any axioms
'ChryseneFormalism.Article85.the_gap_is_the_difference_of_the_two_registration_numbers' does not depend on any axioms
'ChryseneFormalism.Article85.a_contact_is_sensitive_exactly_when_its_gap_is_positive' does not depend on any axioms
'ChryseneFormalism.Article85.the_gap_is_at_most_four' does not depend on any axioms
'ChryseneFormalism.Article85.the_gap_masses_are_derived_at_nine_tenths' does not depend on any axioms
'ChryseneFormalism.Article85.the_gap_masses_are_derived_at_perfect_arming' does not depend on any axioms
'ChryseneFormalism.Article85.and_every_gap_from_one_to_four_is_attained' does not depend on any axioms
'ChryseneFormalism.Article85.at_perfect_arming_only_gap_one_survives' does not depend on any axioms
'ChryseneFormalism.Article85.and_the_blind_mass_is_everything_else' does not
```

```
    depend on any axioms
'ChryseneFormalism.Article85.the_masses_are_normalised_at_nine_tenths' does
    not depend on any axioms
'ChryseneFormalism.Article85.the_sensitive_mass_is_sigma_at_nine_tenths' does
    not depend on any axioms
'ChryseneFormalism.Article85.the_blind_mass_is_one_minus_sigma_at_nine_tenths'
     does not depend on any axioms
'ChryseneFormalism.Article85.the_gap_two_share_rounds_to_99997' does not
    depend on any axioms
'ChryseneFormalism.Article85.the_gap_one_share_rounds_to_zero' does not depend
     on any axioms
'ChryseneFormalism.Article85.the_mean_gap_rounds_to_two' does not depend on
    any axioms
'ChryseneFormalism.Article85.
    the_distribution_is_a_near_point_mass_not_a_balance' does not depend on
    any axioms
'ChryseneFormalism.Article85.relab_is_an_involution' does not depend on any
    axioms
'ChryseneFormalism.Article85.relab_reads_the_arm_from_the_inverted_locant'
    does not depend on any axioms
'ChryseneFormalism.Article85.relab_preserves_the_weight_on_D_and_T' does not
    depend on any axioms
'ChryseneFormalism.Article85.relab_is_a_bijection_of_D_and_of_T_words' does
    not depend on any axioms
'ChryseneFormalism.Article85.relab_reverses_the_pattern' does not depend on
    any axioms
'ChryseneFormalism.Article85.on_I_the_relabelling_is_not_even_defined' does
    not depend on any axioms
'ChryseneFormalism.Article85.
    without_the_initiator_the_signed_bias_is_exactly_zero' does not depend on
    any axioms
'ChryseneFormalism.Article85.and_the_two_halves_are_not_themselves_zero' does
    not depend on any axioms
'ChryseneFormalism.Article85.with_the_initiator_the_bias_is_nonzero' does not
    depend on any axioms
'ChryseneFormalism.Article85.and_it_rounds_to_203e_minus_7' does not depend on
     any axioms
'ChryseneFormalism.Article85.the_bias_is_below_f_I_squared' does not depend on
     any axioms
'ChryseneFormalism.Article85.arm_efficiency_rounds_to_9875_at_nine_tenths'
    does not depend on any axioms
'ChryseneFormalism.Article85.
    arm_efficiency_rounds_to_9964_at_nineteen_twentieths' does not depend on
    any axioms
'ChryseneFormalism.Article85.
    arm_efficiency_rounds_to_9997_at_ninetynine_hundredths' does not depend on
     any axioms
'ChryseneFormalism.Article85.arm_efficiency_is_one_at_perfect_arming' does not
     depend on any axioms
'ChryseneFormalism.Article85.arm_efficiency_falls_as_cleavage_rises' does not
    depend on any axioms
'ChryseneFormalism.Article85.the_derivative_has_degree_seven' does not depend
    on any axioms
'ChryseneFormalism.Article85.the_derivative_at_zero_is_the_linear_coefficient'
     does not depend on any axioms
'ChryseneFormalism.Article85.every_substituted_coefficient_is_positive' does
    not depend on any axioms
'ChryseneFormalism.Article85.the_substitution_has_degree_seven' does not
    depend on any axioms
```

```
'ChryseneFormalism.Article85.
   the_least_substituted_coefficient_is_the_derivative_at_zero' does not
   depend on any axioms
'ChryseneFormalism.Article85.
   so_sigma_is_strictly_increasing_on_the_operating_range' does not depend on
    any axioms
'ChryseneFormalism.Article85.
   and_the_trade_is_genuine_sensitivity_up_efficiency_down' does not depend
   on any axioms
'ChryseneFormalism.Article85.the_adjacent_mass_is_derived' does not depend on
   any axioms
'ChryseneFormalism.Article85.
   adjacent_contacts_are_positively_correlated_at_nine_tenths' does not
   depend on any axioms
'ChryseneFormalism.Article85.
   and_the_correlation_ratio_rounds_to_five_point_six' does not depend on any
    axioms
'ChryseneFormalism.Article85.the_packed_vector_starts_at_one_on_every_type'
   does not depend on any axioms
'ChryseneFormalism.Article85.and_at_zero_off_the_valid_types' does not depend
   on any axioms
'ChryseneFormalism.Article85.p_no_sensitive_at_six_sites_nine_tenths' does not
    depend on any axioms
'ChryseneFormalism.Article85.p_any_at_six_sites_nine_tenths' does not depend
   on any axioms
'ChryseneFormalism.Article85.p_any_rounds_to_1363_at_nine_tenths' does not
   depend on any axioms
'ChryseneFormalism.Article85.p_no_sensitive_at_six_sites_perfect_arming' does
   not depend on any axioms
'ChryseneFormalism.Article85.p_any_at_six_sites_perfect_arming' does not
   depend on any axioms
'ChryseneFormalism.Article85.p_any_rounds_to_124e_minus_6_at_perfect_arming'
   does not depend on any axioms
'ChryseneFormalism.Article85.the_binomial_figure_overstates_at_six_sites' does
    not depend on any axioms
'ChryseneFormalism.Article85.the_blind_mass_is_the_complement_of_sigma' does
   not depend on any axioms
'ChryseneFormalism.Article85.
   the_expected_count_at_six_sites_rounds_to_162_thousandths' does not depend
    on any axioms
'ChryseneFormalism.Article85.
   the_expected_count_at_fiftyfour_sites_rounds_to_1717_thousandths' does not
    depend on any axioms
'ChryseneFormalism.Article85.the_still_needed_share_rounds_to_9676_hundredths'
    does not depend on any axioms
'ChryseneFormalism.Article85.over_ninetysix_per_cent_still_need_the_reference'
    does not depend on any axioms
'ChryseneFormalism.Article85.the_box_has_fiftyfour_nodes' does not depend on
   any axioms
'ChryseneFormalism.Article85.it_is_nine_columns_of_six' does not depend on any
    axioms
'ChryseneFormalism.Article85.it_carries_fortyfive_stacking_contacts' does not
   depend on any axioms
'ChryseneFormalism.Article85.
   a_single_column_of_fiftyfour_would_carry_fiftythree' does not depend on
   any axioms
'ChryseneFormalism.Article85.the_two_figures_are_different' does not depend on
    any axioms
```

```
exit code 0, elapsed 95 s
```

Kernel Axiom Audit

Axiom Inventory

Running `#print axioms` on each of the 129 theorems in `acf-85-proof.lean` yields:

Result	Meaning	Count
no axioms	The proof term reduces to primitive inference steps alone. Nothing is assumed beyond the type theory.	**129**
`propext`	Not used.	0
`Quot.sound`	Not used.	0
`funext`	Not used.	0
`Classical.choice`	Not used.	0
Extra axioms introduced by this proof:		**none**

The empty audit is what the Gibbs grid of S1 buys, and the box above says what it costs. It is also what rules out the bounded existential: core's `Nat.decidableExistsLT` decides $\exists m,\, m < n \wedge p\, m$ through `decidable_of_iff`, which carries `propext` and `Quot.sound`, so the two surjectivity claims of S7 are written with a Boolean search `anyB` instead. The claim is the same and the audit is empty.

The proof files of the corpus, and what their counts mean

Article	Kind	Checks	Theorems	Axiom-free
79	formal	88	55	54
80	formal	99	61	61
81	matching	91	55	55
82	concordance	120	109	109
84	formal	206	171	171
85	formal	283	129	**129**

The first four rows are the figures the Article 82 proof dossier reports and the fifth is the Article 84 dossier's, all carried here unchanged; they were measured there and have not been re-measured for this file. The last row was measured here, by the audit above.

This file has fewer theorems than Article 84's and covers a harder paper, and the two facts have the same cause. Article 84 is combinatorial throughout, so almost every claim it makes is decidable and cheap, and its certificates accumulate. Article 85 divides: its combinatorial half is here in full, and its analytic half — the entropies, the capacity, the trade — is not here at all. A reader comparing 129 with 171 is comparing a file that covers part of its article with one that covers nearly all of its own, and the counts run the other way from the difficulty.

One theorem here carries more of the article than twenty of the rounding certificates: `every_substituted_coefficient_is_positive`, which is the whole of the monotonicity of σ on the operating range and the reason the trade of §9 is a trade at all. The count does not say so.

Verification Note

The proof file makes no use of `native_decide`, which bypasses the kernel by running compiled code; of `sorry`; of `Classical.choice` or `Classical.em`; or of any Mathlib import. The `decide` tactic in Lean 4 core reduces a goal to a kernel-evaluated Boolean expression, and the kernel performs the evaluation.

Three `set_option` lines appear at the head of the file and none of them weakens anything. `exponentiation.threshold` raises the point at which the *elaborator* declines to pre-evaluate a power; past it Lean emits a warning, not an error, and the kernel evaluates the power either way. This file needs it for the packing bases, 2^{128} for polynomials and 2^{320} for the transfer vector, and for $(2011 \cdot 10^4)^{10}$ in the binomial comparison of S9. `maxRecDepth` is raised because the default refuses the deeper folds outright, and `maxHeartbeats` because the elaborator's default budget is smaller than the 2304-contact fold of S5. All three are budgets, not assumptions, and the audit above is the evidence.

To reproduce: install Lean 4.12.0 and run `lean acf-85-proof.lean`. It takes about 95 seconds, most of it in the four pinned derivations, prints 129 audit lines and exits 0. There are no dependencies beyond the Lean core, so no `lakefile` and no package fetch are required.

Scope: four limits, stated plainly

A kernel-checked theorem certifies that a conclusion follows from the definitions in the file. For this article that is a good deal, because the definitions are the corpus's own and no measured datum stands between them and the conclusions. Four limits belong with it.

First, the entropies are not here. This is the limit that makes this dossier different from its companions, and it is stated in the first box rather than the last. Every figure of the article that depends on Δ — the information per contact, the capacity, the conditional entropy, the surprisal range, the trade ratios — is certified by the harness's enclosures and by nothing in this file.

Second, Δ is nowhere measured in the corpus. Article 79's registration free energy is its least constrained parameter. That is why the σ ceiling, which is Δ-free, is the operative statement of the article; and the ceiling's combinatorial content — that the blind mass is exactly $1 - \sigma$ — *is* certified here, in S6, even though the information figures it bounds are not.

Third, the channel is a channel and not a mechanism. S7 certifies that the relabelling exchanges the two alignments and that the signed bias vanishes exactly without the initiator. That says the channel carries no direction. It does not say that anything reads the channel, and the article's H_{frame} is refined by it rather than discharged: S9 certifies that over ninety-six per cent of a column's contacts still need the reference at the heaviest cleavage considered.

Fourth, both artifacts encode the article's conclusions. The harness fails if the field law is not exactly uniform without the initiator; this file's S7 does the same, and its S9 fails if adjacent contacts are not positively correlated. That is deliberate and the article's appendix states it: a published verification artifact should stop passing when the claims it supports stop holding. But it makes both files regression tests on the conclusions rather than independent tests of them.