Spec Category

Dr. Tamás Nagy

Dr. Tamás Nagy

tamas@thelatent.space

Skeleton

Abstract

Spec Category — The Proof Category of Spectral Transfers.

This paper presents 39 machine-verified theorems building on 16 established facts and 8 hypotheses. All results are formally verified in the Platonic proof kernel (118 verification units, 2 proved statements) and exportable to Lean 4.

---

1. Introduction

2. Further Results

Theorem (rmt_two_step_high_gain). *Rmt Two Step High Gain.* [Platonic: rmt_two_step_high_gain, domain: spec_category]

Theorem (gap_ordering). *Gap Ordering.* [Platonic: gap_ordering, domain: spec_category]

Theorem (delta_comp_formula). *Delta Comp Formula.* [Platonic: delta_comp_formula, domain: spec_category]

Theorem (delta_comp_ge_d1). *Delta Comp Ge D1.* [Platonic: delta_comp_ge_d1, domain: spec_category]

Theorem (delta_comp_ge_d2). *Delta Comp Ge D2.* [Platonic: delta_comp_ge_d2, domain: spec_category]

Theorem (delta_comp_le_one). *Delta Comp Le One.* [Platonic: delta_comp_le_one, domain: spec_category]

Theorem (delta_comp_symmetric). *Delta Comp Symmetric.* [Platonic: delta_comp_symmetric, domain: spec_category]

Theorem (delta_comp_with_zero). *Delta Comp With Zero.* [Platonic: delta_comp_with_zero, domain: spec_category]

Theorem (delta_comp_with_one). *Delta Comp With One.* [Platonic: delta_comp_with_one, domain: spec_category]

Theorem (delta_comp_monotone_left). *Delta Comp Monotone Left.* [Platonic: delta_comp_monotone_left, domain: spec_category]

Theorem (delta_comp_three). *Delta Comp Three.* [Platonic: delta_comp_three, domain: spec_category]

Theorem (circuit_two_step). *Circuit Two Step.* [Platonic: circuit_two_step, domain: spec_category]

Theorem (circuit_residue_decreasing). *Circuit Residue Decreasing.* [Platonic: circuit_residue_decreasing, domain: spec_category]

Theorem (circuit_two_ge_one). *Circuit Two Ge One.* [Platonic: circuit_two_ge_one, domain: spec_category]

Theorem (circuit_perfect_edge). *Circuit Perfect Edge.* [Platonic: circuit_perfect_edge, domain: spec_category]

Theorem (circuit_no_finite_exact). *Circuit No Finite Exact.* [Platonic: circuit_no_finite_exact, domain: spec_category]

Theorem (threshold_solvable). *Threshold Solvable.* [Platonic: threshold_solvable, domain: spec_category]

Theorem (cft_algebraic_exact). *Cft Algebraic Exact.* [Platonic: cft_algebraic_exact, domain: spec_category]

Theorem (wiles_flt_circuit). *Wiles Flt Circuit.* [Platonic: wiles_flt_circuit , domain: spec_category]

Theorem (langlands_gap_theorem). *Langlands Gap Theorem.* [Platonic: langlands_gap_theorem, domain: spec_category]

Theorem (langlands_combined_additive_zero). *Langlands Combined Additive Zero.* [Platonic: langlands_combined_additive_zero, domain: spec_category]

Theorem (chebotarev_circuit). *Chebotarev Circuit.* [Platonic: chebotarev_circuit, domain: spec_category]

Theorem (euler_product_additive_invisible). *Euler Product Additive Invisible.* [Platonic: euler_product_additive_invisible, domain: spec_category]

Theorem (sieve_parity_barrier). *Sieve Parity Barrier.* [Platonic: sieve_parity_barrier, domain: spec_category]

Theorem (sieve_plus_fourier). *Sieve Plus Fourier.* [Platonic: sieve_plus_fourier, domain: spec_category]

Theorem (twin_prime_gap). *Twin Prime Gap.* [Platonic: twin_prime_gap, domain: spec_category]

Theorem (sieve_iterative_two). *Sieve Iterative Two.* [Platonic: sieve_iterative_two, domain: spec_category]

Theorem (rmt_improves_sieve). *Rmt Improves Sieve.* [Platonic: rmt_improves_sieve, domain: spec_category]

Theorem (rmt_threshold_break). *Rmt Threshold Break.* [Platonic: rmt_threshold_break, domain: spec_category]

Theorem (explicit_formula_preserves). *Explicit Formula Preserves.* [Platonic: explicit_formula_preserves, domain: spec_category]

Theorem (rmt_unconditional_weak). *Rmt Unconditional Weak.* [Platonic: rmt_unconditional_weak, domain: spec_category]

Theorem (ternary_goldbach_solved). *Ternary Goldbach Solved.* [Platonic: ternary_goldbach_solved, domain: spec_category]

Theorem (binary_goldbach_near_threshold). *Binary Goldbach Near Threshold.* [Platonic: binary_goldbach_near_threshold, domain: spec_category]

Theorem (additive_algebraic_gap). *Additive Algebraic Gap.* [Platonic: additive_algebraic_gap, domain: spec_category]

Theorem (classification_solved). *Classification Solved.* [Platonic: classification_solved , domain: spec_category]

Theorem (non_langlands_needed). *Non Langlands Needed.* [Platonic: non_langlands_needed, domain: spec_category]

3. Bounds and Estimates

Theorem (twin_prime_circuit_bound). *Twin Prime Circuit Bound.* [Platonic: twin_prime_circuit_bound, domain: spec_category]

Theorem (sieve_iter_bounded). *Sieve Iter Bounded.* [Platonic: sieve_iter_bounded, domain: spec_category]

Theorem (rmt_plus_sieve_bound). *Rmt Plus Sieve Bound.* [Platonic: rmt_plus_sieve_bound, domain: spec_category]

4. Formal Framework

Hypotheses

- H_sieve_bound: Sieve Bound
- H_sieve_upper: Sieve Upper
- H_fourier_bound: Fourier Bound
- H_fourier_upper: Fourier Upper
- H_tp_fourier: Tp Fourier
- H_tp_sieve_lo: Tp Sieve Lo
- H_tp_sieve_hi: Tp Sieve Hi
- H_rmt_bounded: Rmt Bounded

Established Facts

- F_cft_value: Cft Value
- F_mod_value: Mod Value
- F_bc_bound: Bc Bound
- F_bc_upper: Bc Upper
- F_langlands_additive_zero: Langlands Additive Zero

- F_ep_local: Ep Local
- F_ep_additive_blind: Ep Additive Blind
- F_gpy_value_lower: Gpy Value Lower
- F_gpy_value_upper: Gpy Value Upper
- F_rmt_unconditional_nonneg: Rmt Unconditional Nonneg
- F_explicit_nonneg: Explicit Nonneg
- F_explicit_upper: Explicit Upper
- F_ternary_value: Ternary Value
- F_ternary_upper: Ternary Upper
- F_binary_conditional: Binary Conditional
- F_binary_upper: Binary Upper

5. Proof Architecture

All proofs are implemented in the Platonic kernel (elysium/fields/spec_category/).

File	Role
spec_category_proof.py	

6. Discussion

References