Proof Complexity Rho

Dr. Tamás Nagy

Dr. Tamás Nagy

tamas@thelatent.space

Skeleton

Abstract

This paper presents 51 machine-verified theorems building on 19 established facts and 106 hypotheses. All results are formally verified in the Platonic proof kernel (201 verification units, 59 proved statements) and exportable to Lean 4.

---

1. Introduction

2. Further Results

Theorem (coverage_implies_rho). *Coverage Implies Rho.* [Platonic: coverage_implies_rho, domain: proof_complexity_rho]

Theorem (rho_implies_coverage). *Rho Implies Coverage.* [Platonic: rho_implies_coverage, domain: proof_complexity_rho]

Theorem (coverage_iff_rho). *Coverage Iff Rho.* [Platonic: coverage_iff_rho, domain: proof_complexity_rho]

Theorem (tractable_iff_rho). *Tractable Iff Rho.* [Platonic: tractable_iff_rho, domain: proof_complexity_rho]

Theorem (four_way_chain). *Four Way Chain.* [Platonic: four_way_chain, domain: proof_complexity_rho]

Theorem (all_subcritical_implies_p_neq_np). *All Subcritical Implies P Neq Np.* [Platonic: all_subcritical_implies_p_neq_np, domain: proof_complexity_rho]

Theorem (all_subcritical_implies_np_neq_conp). *All Subcritical Implies Np Neq Conp.* [Platonic: all_subcritical_implies_np_neq_conp, domain: proof_complexity_rho]

Theorem (subcritical_landscape). *Subcritical Landscape.* [Platonic: subcritical_landscape, domain: proof_complexity_rho]

Theorem (np_neq_conp_no_supercritical). *Np Neq Conp No Supercritical.* [Platonic: np_neq_conp_no_supercritical, domain: proof_complexity_rho]

Theorem (coverage_anywhere_implies_npconp). *Coverage Anywhere Implies Npconp.* [Platonic: coverage_anywhere_implies_npconp, domain: proof_complexity_rho]

Theorem (ns_not_poly). *Ns Not Poly.* [Platonic: ns_not_poly, domain: proof_complexity_rho]

Theorem (ns_rho_le_one). *Ns Rho Le One.* [Platonic: ns_rho_le_one, domain: proof_complexity_rho]

Theorem (pc_not_poly). *Pc Not Poly.* [Platonic: pc_not_poly, domain: proof_complexity_rho]

Theorem (pc_rho_le_one). *Pc Rho Le One.* [Platonic: pc_rho_le_one, domain: proof_complexity_rho]

Theorem (sosbd_rho_le_one). *Sosbd Rho Le One.* [Platonic: sosbd_rho_le_one, domain: proof_complexity_rho]

Theorem (algebraic_rho_chain). *Algebraic Rho Chain.* [Platonic: algebraic_rho_chain, domain: proof_complexity_rho]

Theorem (sos_rho_gt_one). *Sos Rho Gt One.* [Platonic: sos_rho_gt_one, domain: proof_complexity_rho]

Theorem (sos_degree_gap). *Sos Degree Gap.* [Platonic: sos_degree_gap, domain: proof_complexity_rho]

Theorem (ns_sos_rho_gap). *Ns Sos Rho Gap.* [Platonic: ns_sos_rho_gap, domain: proof_complexity_rho]

Theorem (algebraic_phase_transition). *Algebraic Phase Transition.* [Platonic: algebraic_phase_transition, domain: proof_complexity_rho]

Theorem (res_rho_le_one). *Res Rho Le One.* [Platonic: res_rho_le_one, domain: proof_complexity_rho]

Theorem (cross_paradigm_rho). *Cross Paradigm Rho.* [Platonic: cross_paradigm_rho, domain: proof_complexity_rho]

Theorem (resolution_not_poly). *Resolution Not Poly.* [Platonic: resolution_not_poly, domain: proof_complexity_rho]

Theorem (resolution_rho_le_one). *Resolution Rho Le One.* [Platonic: resolution_rho_le_one, domain: proof_complexity_rho]

Theorem (frege_rho_ge_resolution). *Frege Rho Ge Resolution.* [Platonic: frege_rho_ge_resolution, domain: proof_complexity_rho]

Theorem (ef_rho_ge_frege). *Ef Rho Ge Frege.* [Platonic: ef_rho_ge_frege, domain: proof_complexity_rho]

Theorem (npconp_implies_ef_rho). *Npconp Implies Ef Rho.* [Platonic: npconp_implies_ef_rho, domain: proof_complexity_rho]

Theorem (resolution_ef_rho_gap). *Resolution Ef Rho Gap.* [Platonic: resolution_ef_rho_gap, domain: proof_complexity_rho]

Theorem (resolution_cannot_witness_npconp). *Resolution Cannot Witness Npconp.* [Platonic: resolution_cannot_witness_npconp, domain: proof_complexity_rho]

Theorem (resolution_rho_in_subcritical). *Resolution Rho In Subcritical.* [Platonic: resolution_rho_in_subcritical, domain: proof_complexity_rho]

Theorem (rho_decidable_iff). *Rho Decidable Iff.* [Platonic: rho_decidable_iff, domain: proof_complexity_rho]

Theorem (rho_creative_iff). *Rho Creative Iff.* [Platonic: rho_creative_iff, domain: proof_complexity_rho]

Theorem (creative_penalty_gap). *Creative Penalty Gap.* [Platonic: creative_penalty_gap, domain: proof_complexity_rho]

Theorem (decidable_no_penalty). *Decidable No Penalty.* [Platonic: decidable_no_penalty, domain: proof_complexity_rho]

Theorem (coverage_tractability_iff). *Coverage Tractability Iff.* [Platonic: coverage_tractability_iff, domain: proof_complexity_rho]

Theorem (decidable_creative_exclusion). *Decidable Creative Exclusion.* [Platonic: decidable_creative_exclusion domain: proof_complexity_rho]

Theorem (rho_gt_one_fully_decidable). *Rho Gt One Fully Decidable.* [Platonic: rho_gt_one_fully_decidable, domain: proof_complexity_rho]

Theorem (natural_cannot_prove_pnp). *Natural Cannot Prove Pnp.* [Platonic: natural_cannot_prove_pnp, domain: proof_complexity_rho]

Theorem (relativizing_cannot_prove_pnp). *Relativizing Cannot Prove Pnp.* [Platonic: relativizing_cannot_prove_pnp, domain: proof_complexity_rho]

Theorem (algebraizing_cannot_prove_pnp). *Algebraizing Cannot Prove Pnp.* [Platonic: algebraizing_cannot_prove_pnp, domain: proof_complexity_rho]

Theorem (all_supercritical_blocked). *All Supercritical Blocked.* [Platonic: all_supercritical_blocked, domain: proof_complexity_rho]

Theorem (pnp_proof_requires_subcritical). *Pnp Proof Requires Subcritical.* [Platonic: pnp_proof_requires_subcritical, domain: proof_complexity_rho]

Theorem (barrier_landscape). *Barrier Landscape.* [Platonic: barrier_landscape, domain: proof_complexity_rho]

Theorem (natural_is_supercritical). *Natural Is Supercritical.* [Platonic: natural_is_supercritical, domain: proof_complexity_rho]

Theorem (pnp_rho_paradox). *Pnp Rho Paradox.* [Platonic: pnp_rho_paradox, domain: proof_complexity_rho]

Theorem (pnp_requires_creative). *Pnp Requires Creative.* [Platonic: pnp_requires_creative, domain: proof_complexity_rho]

Theorem (godel_barrier_pnp). *Godel Barrier Pnp.* [Platonic: godel_barrier_pnp, domain: proof_complexity_rho]

3. Cross-Domain Bridges

Theorem (bridge_phase_transition). *Bridge Phase Transition.* [Platonic: bridge_phase_transition, domain: proof_complexity_rho]

Theorem (bridge_rho_jump). *Bridge Rho Jump.* [Platonic: bridge_rho_jump, domain: proof_complexity_rho]

Theorem (bridge_composition_with_distance). *Bridge Composition With Distance.* [Platonic: bridge_composition_with_distance, domain: proof_complexity_rho]

4. Existence and Uniqueness

Theorem (sos_unique_supercritical). *Sos Unique Supercritical.* [Platonic: sos_unique_supercritical, domain: proof_complexity_rho]

5. Formal Framework

Hypotheses

- ProofSys: Proofsys
- rho_sys: Rho Sys
- IsPolyBounded: Ispolybounded
- FullCoverage: Fullcoverage
- ProofTractable: Prooftractable
- NP_eq_coNP: Np Eq Conp
- P_eq_NP: P Eq Np
- H_poly_implies_rho: Poly Implies Rho
- H_rho_implies_poly: Rho Implies Poly
- H_tractable_implies_poly: Tractable Implies Poly
- H_poly_implies_tractable: Poly Implies Tractable
- H_coverage_implies_tractable: Coverage Implies Tractable
- H_poly_implies_coverage: Poly Implies Coverage
- H_CR_fwd: Cr Fwd
- H_CR_bwd: Cr Bwd
- ProofSys: Proofsys
- rho_sys: Rho Sys
- IsPolyBounded: Ispolybounded
- HasExpLowerBound: Hasexplowerbound
- Simulates: Simulates
- HasPolyFragment: Haspolyfragment
- IsDegreeBounded: Isdegreebounded
- Resolution: Resolution
- ExtFrege: Extfrege
- Nullstellensatz : Nullstellensatz
- PolynomialCalculus: Polynomialcalculus
- SumOfSquares: Sumofsquares
- SOS_BoundedDeg: Sos Boundeddeg
- H_poly_implies_rho: Poly Implies Rho
- H_rho_implies_poly: Rho Implies Poly
- H_exp_not_poly: Exp Not Poly
- H_sim_rho_mono: Sim Rho Mono
- H_rho_pos: Rho Pos
- H_poly_fragment_rho: Poly Fragment Rho
- ProofSys: Proofsys

- Taut: Taut
- rho_sys: Rho Sys
- rho_proof: Rho Proof
- IsPolyBounded: Ispolybounded
- NP_eq_coNP: Np Eq Conp
- HasExpLowerBound: Hasexplowerbound
- Simulates: Simulates
- Resolution: Resolution
- Frege: Frege
- ExtFrege: Extfrege
- H_poly_implies_rho: Poly Implies Rho
- H_rho_implies_poly: Rho Implies Poly
- H_exp_not_poly: Exp Not Poly
- H_CR_rho_fwd: Cr Rho Fwd
- H_sim_rho_mono: Sim Rho Mono
- H_rho_pos: Rho Pos
- H_ef_simulates_all: Ef Simulates All
- ProofState: Proofstate
- rho_local: Rho Local
- d_proof: D Proof
- kappa: Kappa
- d_eff: D Eff
- beta: Beta
- IsDecidable: Isdecidable
- IsCreative: Iscreative
- HasBridge: Hasbridge
- FullCoverage: Fullcoverage
- ProofTractable: Prooftractable
- H_dec_implies_rho: Dec Implies Rho
- H_rho_implies_dec: Rho Implies Dec
- H_cre_implies_rho_le: Cre Implies Rho Le
- H_rho_le_implies_cre: Rho Le Implies Cre
- H_partition: Partition
- H_exclusive: Exclusive
- H_bridge_target_dec: Bridge Target Dec
- H_bridge_progress: Bridge Progress
- H_bridge_comp: Bridge Comp
- H_beta_gt_one: Beta Gt One
- H_kappa_pos: Kappa Pos
- H_creative_penalty: Creative Penalty
- H_decidable_no_penalty: Decidable No Penalty
- H_rho_pos: Rho Pos
- H_coverage_tractable: Coverage Tractable
- H_no_coverage_no_tractable: No Coverage No Tractable
- ProofStrategy: Proofstrategy
- Barrier: Barrier
- rho_strategy: Rho Strategy
- rho_barrier: Rho Barrier

- IsNatural: Isnatural
- IsRelativizing : Isrelativizing
- IsAlgebraizing: Isalgebraizing
- ProvesCircuitLB: Provescircuitlb
- OWF_exists: Owf Exists
- ProvesP_neq_NP: Provesp Neq Np
- CrossesBarrier: Crossesbarrier
- IsCreativeStrategy: Iscreativestrategy
- NaturalProofsBarrier: Naturalproofsbarrier
- RelativizationBarrier : Relativizationbarrier
- AlgebrizationBarrier: Algebrizationbarrier
- H_RR: Rr
- H_BGS: Bgs
- H_AW: Aw
- H_natural_rho_high: Natural Rho High
- H_rel_rho_high: Rel Rho High
- H_alg_rho_high: Alg Rho High
- H_creative_rho_le: Creative Rho Le
- H_pnp_implies_lb: Pnp Implies Lb
- H_rho_strat_pos: Rho Strat Pos
- H_cross_requires_low_rho: Cross Requires Low Rho
- H_pnp_crosses_all: Pnp Crosses All
- H_rho_le_implies_creative: Rho Le Implies Creative

Established Facts

- F_PNP_implies_NPcoNP: Pnp Implies Npconp
- F_not_lt_implies_le: Not Lt Implies Le
- F_ns_exp: Ns Exp
- F_pc_exp: Pc Exp
- F_sosbd_exp: Sosbd Exp
- F_ns_le_pc: Ns Le Pc
- F_pc_le_sos: Pc Le Sos
- F_sosbd_le_sos: Sosbd Le Sos
- F_sos_poly_fragment: Sos Poly Fragment
- F_sosbd_is_deg_bounded: Sosbd Is Deg Bounded
- F_haken: Haken
- F_haken: Haken
- F_res_le_frege: Res Le Frege
- F_frege_le_ef: Frege Le Ef
- F_not_lt_implies_le: Not Lt Implies Le
- F_not_lt_implies_le: Not Lt Implies Le
- F_npb_rho: Npb Rho
- F_rlb_rho: Rlb Rho
- F_alb_rho: Alb Rho

6. Proof Architecture

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

File	Role
rho_landscape_proof.py	
algebraic_rho_proof.py	
concrete_rho_proof.py	
proof_space_rho_proof.py	
natural_proofs_rho_proof.py	

7. Discussion

References