# Constraint Lifecycle

*Dr. Tamás Nagy*

Dr. Tamás Nagy

tamas@thelatent.space

Skeleton

## Abstract

Constraint Lifecycle — clean ProofEnv proof.

This paper presents 52 machine-verified theorems. All results are formally verified in the Platonic proof kernel (112 verification units, 53 proved statements) and exportable to Lean 4.

---

## 1. Introduction

## 2. Further Results

**Theorem** (lifecycle_weight_two_nonneg). *Lifecycle Weight Two Nonneg.* [Platonic: lifecycle_weight_two_nonneg, domain: constraint_lifecycle ]

**Theorem** (lifecycle_weight_le_kelly). *Lifecycle Weight Le Kelly.* [Platonic: lifecycle_weight_le_kelly, domain: constraint_lifecycle ]

**Theorem** (interval_positive_width). *Interval Positive Width.* [Platonic: interval_positive_width, domain: constraint_lifecycle ]

**Theorem** (kelly_diff_propagation). *Kelly Diff Propagation.* [Platonic: kelly_diff_propagation, domain: constraint_lifecycle ]

**Theorem** (lifecycle_weight_four_nonneg). *Lifecycle Weight Four Nonneg.* [Platonic: lifecycle_weight_four_nonneg, domain: constraint_lifecycle ]

**Theorem** (lifecycle_weight_four_le_kelly). *Lifecycle Weight Four Le Kelly.* [Platonic: lifecycle_weight_four_le_kelly, domain: constraint_lifecycle ]

**Theorem** (kelly_gamma_product_diff). *Kelly Gamma Product Diff.* [Platonic: kelly_gamma_product_diff, domain: constraint_lifecycle ]

**Theorem** (lifecycle_weight_zero_safety). *Lifecycle Weight Zero Safety.* [Platonic: lifecycle_weight_zero_safety, domain: constraint_lifecycle ]

**Theorem** (constrained_lifecycle_pos). *Constrained Lifecycle Pos.* [Platonic: constrained_lifecycle_pos, domain: constraint_lifecycle ]

**Theorem** (lifecycle_three_nonneg). *Lifecycle Three Nonneg.* [Platonic: lifecycle_three_nonneg, domain: constraint_lifecycle ]

**Theorem** (lifecycle_three_le_kelly). *Lifecycle Three Le Kelly.* [Platonic: lifecycle_three_le_kelly, domain: constraint_lifecycle]

**Theorem** (unit_interval_product_le_one). *Unit Interval Product Le One.* [Platonic: unit_interval_product_le_one, domain: constraint_lifecycle]

**Theorem** (two_factor_mul_pos). *Two Factor Mul Pos.* [Platonic: two_factor_mul_pos, domain: constraint_lifecycle]

**Theorem** (kelly_weight_mono_wk). *Kelly Weight Mono Wk.* [Platonic: kelly_weight_mono_wk, domain: constraint_lifecycle]

**Theorem** (interval_width_nonneg_from_strict). *Interval Width Nonneg From Strict.* [Platonic: interval_width_nonneg_from_strict, domain: constraint_lifecycle]

**Theorem** (lifecycle_zero_bayes). *Lifecycle Zero Bayes.* [Platonic: lifecycle_zero_bayes, domain: constraint_lifecycle]

**Theorem** (cl_lifecycle_as_vec1_norm_sq). *Cl Lifecycle As Vec1 Norm Sq.* [Platonic: cl_lifecycle_as_vec1_norm_sq, domain: constraint_lifecycle]

**Theorem** (cl_finset_sum_two_factors_nonneg). *Cl Finset Sum Two Factors Nonneg.* [Platonic: cl_finset_sum_two_factors_nonneg, domain: constraint_lifecycle]

**Theorem** (cl_w4_fourfold_mul_reassoc). *Cl W4 Fourfold Mul Reassoc.* [Platonic: cl_w4_fourfold_mul_reassoc, domain: constraint_lifecycle]

**Theorem** (cl_w4_square_nonneg). *Cl W4 Square Nonneg.* [Platonic: cl_w4_square_nonneg, domain: constraint_lifecycle]

**Theorem** (cl_w4_lifecycle_le_prod_unit_caps). *Cl W4 Lifecycle Le Prod Unit Caps.* [Platonic: cl_w4_lifecycle_le_prod_unit_caps, domain: constraint_lifecycle]

**Theorem** (cl_w4_two_factor_le_left). *Cl W4 Two Factor Le Left.* [Platonic: cl_w4_two_factor_le_left, domain: constraint_lifecycle]

**Theorem** (cl_allocation_lipschitz). *Cl Allocation Lipschitz.* [Platonic: cl_allocation_lipschitz, domain: constraint_lifecycle]

**Theorem** (cl_glide_path_decreasing). *Cl Glide Path Decreasing.* [Platonic: cl_glide_path_decreasing, domain: constraint_lifecycle]

**Theorem** (cl_lifecycle_zero_at_floor). *Cl Lifecycle Zero At Floor.* [Platonic: cl_lifecycle_zero_at_floor, domain: constraint_lifecycle]

**Theorem** (cl_lifecycle_mono_kelly). *Cl Lifecycle Mono Kelly.* [Platonic: cl_lifecycle_mono_kelly, domain: constraint_lifecycle]

**Theorem** (lean_cl_feasible_implies_positive). *Lean Cl Feasible Implies Positive.* [Platonic: lean_cl_feasible_implies_positive, domain: constraint_lifecycle]

**Theorem** (lean_cl_surplus_positive). *Lean Cl Surplus Positive.* [Platonic: lean_cl_surplus_positive, domain: constraint_lifecycle]

**Theorem** (lean_cl_surplus_ne_zero). *Lean Cl Surplus Ne Zero.* [Platonic: lean_cl_surplus_ne_zero, domain: constraint_lifecycle]

**Theorem** (lean_cl_feasibility_monotone). *Lean Cl Feasibility Monotone.* [Platonic: lean_cl_feasibility_monotone, domain: constraint_lifecycle ]

**Theorem** (lean_cl_witness_feasible). *Lean Cl Witness Feasible.* [Platonic: lean_cl_witness_feasible, domain: constraint_lifecycle ]

**Theorem** (lean_cl_strict_interior_surplus_one). *Lean Cl Strict Interior Surplus One.* [Platonic: lean_cl_strict_interior_surplus_one, domain: constraint_lifecycle ]

**Theorem** (lean_cl_floor_set_convex_combo). *Lean Cl Floor Set Convex Combo.* [Platonic: lean_cl_floor_set_convex_combo, domain: constraint_lifecycle ]

**Theorem** (lean_cl_cap_set_convex_combo). *Lean Cl Cap Set Convex Combo.* [Platonic: lean_cl_cap_set_convex_combo, domain: constraint_lifecycle ]

**Theorem** (lean_cl_interval_convex_lo). *Lean Cl Interval Convex Lo.* [Platonic: lean_cl_interval_convex_lo, domain: constraint_lifecycle ]

**Theorem** (lean_cl_interval_convex_hi). *Lean Cl Interval Convex Hi.* [Platonic: lean_cl_interval_convex_hi, domain: constraint_lifecycle ]

**Theorem** (lean_cl_midpoint_interval_lb). *Lean Cl Midpoint Interval Lb.* [Platonic: lean_cl_midpoint_interval_lb, domain: constraint_lifecycle ]

**Theorem** (lean_cl_midpoint_interval_ub). *Lean Cl Midpoint Interval Ub.* [Platonic: lean_cl_midpoint_interval_ub, domain: constraint_lifecycle ]

**Theorem** (lean_cl_midpoint_strict_lo). *Lean Cl Midpoint Strict Lo.* [Platonic: lean_cl_midpoint_strict_lo, domain: constraint_lifecycle ]

**Theorem** (lean_cl_midpoint_strict_hi). *Lean Cl Midpoint Strict Hi.* [Platonic: lean_cl_midpoint_strict_hi, domain: constraint_lifecycle ]

**Theorem** (lean_cu_midpoint_pos). *Lean Cu Midpoint Pos.* [Platonic: lean_cu_midpoint_pos, domain: constraint_lifecycle ]

**Theorem** (lean_cu_eq_of_sub_zero). *Lean Cu Eq Of Sub Zero.* [Platonic: lean_cu_eq_of_sub_zero, domain: constraint_lifecycle ]

**Theorem** (lean_gamma_mono_floor_crossmul). *Lean Gamma Mono Floor Crossmul.* [Platonic: lean_gamma_mono_floor_crossmul, domain: constraint_lifecycle ]

**Theorem** (lean_gamma_double_mono_floor_crossmul). *Lean Gamma Double Mono Floor Crossmul.* [Platonic: lean_gamma_double_mono_floor_crossmul, domain: constraint_lifecycle ]

**Theorem** (lean_gamma_decreasing_wealth_crossmul). *Lean Gamma Decreasing Wealth Crossmul.* [Platonic: lean_gamma_decreasing_wealth_crossmul, domain: constraint_lifecycle ]

**Theorem** (lean_sensitivity_crra_order_crossmul). *Lean Sensitivity Crra Order Crossmul.* [Platonic: lean_sensitivity_crra_order_crossmul, domain: constraint_lifecycle ]

**Theorem** (lean_sensitivity_alloc_num_neg). *Lean Sensitivity Alloc Num Neg.* [Platonic: lean_sensitivity_alloc_num_neg, domain: constraint_lifecycle ]

**Theorem** (lean_sensitivity_alloc_denom_pos). *Lean Sensitivity Alloc Denom Pos.* [Platonic: lean_sensitivity_alloc_denom_pos, domain: constraint_lifecycle ]

## 3. Cross-Domain Bridges

**Theorem** (lean_bridge_implied_gamma_lt_crossmul). *Lean Bridge Implied Gamma Lt Crossmul.* [Platonic: lean_bridge_implied_gamma_lt_crossmul, domain: constraint_lifecycle ]

**Theorem** (lean_bridge_three_factor_le_one). *Lean Bridge Three Factor Le One.* [Platonic: lean_bridge_three_factor_le_one, domain: constraint_lifecycle ]

## 4. Bounds and Estimates

**Theorem** (cl_gamma_perturbation_bounded). *Cl Gamma Perturbation Bounded.* [Platonic: cl_gamma_perturbation_bounded, domain: constraint_lifecycle ]

## 5. Existence and Uniqueness

**Theorem** (cl_strict_concavity_unique). *Cl Strict Concavity Unique.* [Platonic: cl_strict_concavity_unique, domain: constraint_lifecycle ]

## 6. Proof Architecture

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

| File | Role |
|---|---|
| constraint_lifecycle_proof.py | |

## 7. Discussion

## References