Human review packet: GCG24UserItemFairness

Generated from byte-pinned source excerpts, the expanded PaperInterface specifications, and hash-bound raw-source-to-Spec screening records.

Paper: GCG24UserItemFairness
Paper id: GCG24UserItemFairness
Source version: NeurIPS 2024 paper; byte-pinned publication source record
Source record: audit/paper_statement_map.json
Rows: 23
Generated: 2026-09-06
Source URL: official source

How to use this packet

For each source claim, compare the exact source excerpts with the expanded PaperInterface specification. Mark up this PDF or fill the blank reviewer box in the TeX source; return the annotated file for reconciliation into the paper-local human-review log.

Open the interactive dashboard

The interactive dashboard is an optional alternative to using this PDF; it presents the same review surface rather than an additional review requirement. After forking and cloning (or pulling) AppliedModelingLib, run the following from the repository root. The cache command is needed only the first time in a new clone.

```
cd <your-repository-checkout>
lake exe cache get
python3 scripts/review_dashboard.py --paper GCG24UserItemFairness --serve
```

Open http://127.0.0.1:8765/ in a browser and keep that terminal running while reviewing. The dashboard records saved annotations in this paper's local reviewer-owned review record.

Contents

- Governing Lean declarations (119; supporting context)
- Paper-specific semantic prerequisites (25; not paper claims)
  - GCG24UserItemFairness.RecommendationModel.rawUserUtility
  - GCG24UserItemFairness.RecommendationModel.normalizedUserUtility
  - GCG24UserItemFairness.RecommendationModel.userFairness
  - GCG24UserItemFairness.EstimatedRecommendationModel.priceOfMisestimation
  - GCG24UserItemFairness.ProofBridge.problem11EqualityFeasibleSet
  - GCG24UserItemFairness.ProofBridge.problem11BasicFeasible
  - GCG24UserItemFairness.ProofBridge.recommendationUtility
  - GCG24UserItemFairness.RecommendationModel.IsOptimalAtLevel
  - GCG24UserItemFairness.RecommendationModel.itemNormalizer
  - GCG24UserItemFairness.RecommendationModel.rawItemUtility
  - GCG24UserItemFairness.RecommendationModel.normalizedItemUtility
  - GCG24UserItemFairness.RecommendationModel.itemFairness
  - GCG24UserItemFairness.RecommendationModel.priceOfFairness
  - GCG24UserItemFairness.assumption_positive_recommendation_utilities
  - GCG24UserItemFairness.assumption_proposition2_utility_rows_determine_types
  - GCG24UserItemFairness.assumption_theorem3_first_half_alpha_domain
  - GCG24UserItemFairness.assumption_theorem3_opposing_type_model
  - GCG24UserItemFairness.assumption_theorem3_second_half_alpha_domain
  - GCG24UserItemFairness.assumption_theorem4_at_least_three_items
  - GCG24UserItemFairness.assumption_theorem4_cold_start_type_wiring
  - GCG24UserItemFairness.assumption_theorem4_displayed_reduced_models
  - GCG24UserItemFairness.assumption_theorem4_estimated_model_reduction
  - GCG24UserItemFairness.assumption_theorem4_parameter_domain
  - GCG24UserItemFairness.assumption_theorem4_true_model_reduction
  - GCG24UserItemFairness.assumption_theorem4_universal_value_vector

- Main-text source claims (2)
  - proposition1_symmetric_lp_reductionSpec
  - proposition2_symmetric_optimum_existsSpec
- Appendix and supplement source claims (16)
  - appendix_c_lemma1_item_fairness_positiveSpec
  - appendix_d_lemma10_active_optimizer_pivot_at_or_before_midpointSpec
  - appendix_d_lemma11_optimal_item_fairness_mono_on_active_pivot_intervalSpec
  - appendix_d_lemma3_unconstrained_baselineSpec
  - appendix_d_lemma4_problem6_unique_sparse_solutionSpec
  - appendix_d_lemma5_problem6_active_optimizer_closed_formSpec
  - appendix_d_lemma6_optimal_policy_type_utility_orderSpec
  - appendix_d_lemma7_active_optimizer_pivot_monotonicitySpec
  - appendix_d_lemma8_selected_pivot_stitchingSpec
  - appendix_d_lemma9_pair_share_algebraSpec
  - appendix_e_lemma12_symmetrized_policySpec
  - appendix_e_lemma13_every_optimal_basic_solution_has_pivot_supportSpec
  - appendix_e_lemma14_problem11_has_unique_optimal_solutionSpec
  - appendix_e_lemma15_optimal_solution_full_closed_formSpec
  - appendix_e_lemma16_midpoint_order_algebraSpec
  - appendix_e_lemma17_every_optimal_solution_has_no_right_half_supportSpec
- Main-text source claims (continued) (3)
  - theorem3_price_decreases_first_halfSpec
  - theorem3_price_increases_second_halfSpec
  - theorem4_misestimation_without_fairness_universalSpec
- Appendix and supplement source claims (continued) (1)
  - appendix_c_lemma2_item_fairness_equality_lp_solution_setSpec
- Main-text source claims (continued 2) (1)
  - theorem4_misestimation_tradeoffSpec

Governing Lean declarations

These exact graph-bound declaration bodies are retained by the displayed specifications or reviewed prerequisites. They are supporting context and do not add source-result rows or semantic judgments.

AppliedModelingLib.PMFKernel

Declaration location: reusable library

Lean declaration body

```
type:
Type u_1 → Type u_2 → Type (max u_1 u_2)

value:
fun (Input : Type u_1) (Output : Type u_2) => Input → PMF Output
```

AppliedModelingLib.Policy

Declaration location: reusable library

Lean declaration body

```
type:
Type u_1 → Type u_2 → Type (max u_1 u_2)

value:
fun (α : Type u_1) (β : Type u_2) => AppliedModelingLib.PMFKernel α β
```

Direct retained dependencies

- AppliedModelingLib.PMFKernel

AppliedModelingLib.Policy.FiberRepresentatives

Declaration location: reusable library

Lean declaration body

```
field repr:
{α : Type u_1} → {κ : Type u_2} → {τ : α → κ} → AppliedModelingLib.Policy.FiberRepresentatives τ → κ → α

field repr_spec:
∀ {α : Type u_1} {κ : Type u_2} {τ : α → κ} (self : AppliedModelingLib.Policy.FiberRepresentatives τ) (k : κ),
  τ (self.repr k) = k
```

AppliedModelingLib.Policy.FiberRepresentatives.ofSurjective

Declaration location: reusable library

Lean declaration body

```
type:
{α : Type u_1} → {κ : Type u_2} → {τ : α → κ} → Function.Surjective τ → AppliedModelingLib.Policy.FiberRepresentatives τ

value:
fun {α : Type u_1} {κ : Type u_2} {τ : α → κ} (hτ : Function.Surjective τ) =>
  { repr := fun (k : κ) => Classical.choose (hτ k),
    repr_spec := AppliedModelingLib.Policy.FiberRepresentatives.ofSurjective._proof_1 hτ }
```

Direct retained dependencies

- AppliedModelingLib.Policy.FiberRepresentatives

AppliedModelingLib.Policy.IsClasswise

Declaration location: reusable library

Lean declaration body

type:
{α : Type u_1} → {κ : Type u_2} → {β : Type u_3} → (α → κ) → AppliedModelingLib.Policy α β → Prop

value:
fun {α : Type u_1} {κ : Type u_2} {β : Type u_3} (τ : α → κ) (ρ : AppliedModelingLib.Policy α β) =>
  ∀ (a a' : α), τ a = τ a' → ρ a = ρ a'

Direct retained dependencies

- AppliedModelingLib.Policy

AppliedModelingLib.Policy.activePairs

Declaration location: reusable library

Lean declaration body

type:
{α : Type u_1} →
  {β : Type u_2} →
    [Fintype α] → [DecidableEq α] → [Fintype β] → [DecidableEq β] → AppliedModelingLib.Policy α β → Finset (α × β)

value:
fun {α : Type u_1} {β : Type u_2} [Fintype α] [DecidableEq α] [Fintype β] [DecidableEq β]
    (ρ : AppliedModelingLib.Policy α β) =>
  {p : α × β | (ρ p.1) p.2 ≠ 0}

Direct retained dependencies

- AppliedModelingLib.Policy

AppliedModelingLib.Policy.activePairsCard

Declaration location: reusable library

Lean declaration body

type:
{α : Type u_1} →
  {β : Type u_2} → [Fintype α] → [DecidableEq α] → [Fintype β] → [DecidableEq β] → AppliedModelingLib.Policy α β → ℕ

value:
fun {α : Type u_1} {β : Type u_2} [Fintype α] [DecidableEq α] [Fintype β] [DecidableEq β]
    (ρ : AppliedModelingLib.Policy α β) =>
  ρ.activePairs.card

Direct retained dependencies

- AppliedModelingLib.Policy
- AppliedModelingLib.Policy.activePairs

AppliedModelingLib.Policy.descendAlong

Declaration location: reusable library

Lean declaration body

type:
{α : Type u_1} →
  {κ : Type u_2} →
    {β : Type u_3} →
      {τ : α → κ} → AppliedModelingLib.Policy.FiberRepresentatives τ → AppliedModelingLib.Policy α β → κ → PMF β

value:
fun {α : Type u_1} {κ : Type u_2} {β : Type u_3} {τ : α → κ} (reps : AppliedModelingLib.Policy.FiberRepresentatives τ)
    (ρ : AppliedModelingLib.Policy α β) (a : κ) =>
  ρ (reps.repr a)

Direct retained dependencies

- AppliedModelingLib.Policy
- AppliedModelingLib.Policy.FiberRepresentatives

AppliedModelingLib.Policy.multiAssignedActions

Declaration location: reusable library

Lean declaration body

```
type:
{α : Type u_1} →
  {β : Type u_2} →
    [Fintype α] → [DecidableEq α] → [Fintype β] → [DecidableEq β] → AppliedModelingLib.Policy α β → Finset β

value:
fun {α : Type u_1} {β : Type u_2} [Fintype α] [DecidableEq α] [Fintype β] [DecidableEq β]
    (ρ : AppliedModelingLib.Policy α β) =>
  {b : β | ∃ (a : α) (a' : α), a ≠ a' ∧ (ρ a) b ≠ 0 ∧ (ρ a') b ≠ 0}
```

Direct retained dependencies

- AppliedModelingLib.Policy

AppliedModelingLib.finiteMax

Declaration location: reusable library

Lean declaration body

```
type:
{α : Type u_1} → [Fintype α] → [Nonempty α] → (α → ℝ) → ℝ

value:
fun {α : Type u_1} [Fintype α] [Nonempty α] (f : α → ℝ) => Finset.univ.sup' Finset.univ_nonempty f
```

AppliedModelingLib.finiteMin

Declaration location: reusable library

Lean declaration body

```
type:
{α : Type u_1} → [Fintype α] → [Nonempty α] → (α → ℝ) → ℝ

value:
fun {α : Type u_1} [Fintype α] [Nonempty α] (f : α → ℝ) => Finset.univ.inf' Finset.univ_nonempty f
```

AppliedModelingLib.pmfExp

Declaration location: reusable library

Lean declaration body

```
type:
{α : Type u_1} → [Fintype α] → PMF α → (α → ℝ) → ℝ

value:
fun {α : Type u_1} [Fintype α] (μ : PMF α) (f : α → ℝ) => ∑ a : α, (μ a).toReal * f a
```

AppliedModelingLib.Policy.agentScore

Declaration location: reusable library

Lean declaration body

```
type:
{α : Type u_1} → {β : Type u_2} → [Fintype β] → [DecidableEq β] → AppliedModelingLib.Policy α β → (α → β → ℝ) → α → ℝ

value:
fun {α : Type u_1} {β : Type u_2} [Fintype β] [DecidableEq β] (ρ : AppliedModelingLib.Policy α β) (score : α → β → ℝ)
    (a : α) =>
  AppliedModelingLib.pmfExp (ρ a) (score a)
```

Direct retained dependencies

- AppliedModelingLib.Policy
- AppliedModelingLib.pmfExp

GCG24UserItemFairness.Item

Declaration location: paper-local

Lean declaration body

type:
ℕ → Type

value:
fun (n : ℕ) => Fin n

GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex

Declaration location: paper-local

Lean declaration body

type:
{n : ℕ} → (GCG24UserItemFairness.Item n → ℝ) → Prop

value:
fun {n : ℕ} (v : GCG24UserItemFairness.Item n → ℝ) => ∀ {{i j : GCG24UserItemFairness.Item n}}, ↑i < ↑j → v j < v i

Direct retained dependencies

- GCG24UserItemFairness.Item

GCG24UserItemFairness.OpposingTypes.pmfOfRealVector

Declaration location: paper-local

Lean declaration body

type:
{n : ℕ} →
  (x : GCG24UserItemFairness.Item n → ℝ) →
    (∀ (j : GCG24UserItemFairness.Item n), 0 ≤ x j) →
      ∑ j : GCG24UserItemFairness.Item n, x j = 1 → PMF (GCG24UserItemFairness.Item n)

value:
fun {n : ℕ} (x : GCG24UserItemFairness.Item n → ℝ) (hnonneg : ∀ (j : GCG24UserItemFairness.Item n), 0 ≤ x j)
    (hsum : ∑ j : GCG24UserItemFairness.Item n, x j = 1) =>
  PMF.ofFintype (fun (j : GCG24UserItemFairness.Item n) => ENNReal.ofReal (x j))
    (GCG24UserItemFairness.OpposingTypes.pmfOfRealVector._proof_1 x hnonneg hsum)

Direct retained dependencies

- GCG24UserItemFairness.Item

GCG24UserItemFairness.OpposingTypes.reverseItem

Declaration location: paper-local

Lean declaration body

type:
{n : ℕ} → GCG24UserItemFairness.Item n → GCG24UserItemFairness.Item n

value:
fun {n : ℕ} (j : GCG24UserItemFairness.Item n) =>
  ⟨n - 1 - ↑j, GCG24UserItemFairness.OpposingTypes.reverseItem._proof_2 j⟩

Direct retained dependencies

- GCG24UserItemFairness.Item

GCG24UserItemFairness.OpposingTypes.theorem4SmallValueVector

Declaration location: paper-local

Lean declaration body

type:
{n : ℕ} → ℝ → GCG24UserItemFairness.Item n → ℝ

value:
fun {n : ℕ} (eps : ℝ) (a : GCG24UserItemFairness.Item n) => min (1 / 2) (eps / (2 * ↑n)) ^ ↑a

Direct retained dependencies

- GCG24UserItemFairness.Item

GCG24UserItemFairness.OpposingTypes.typeOneShare

Declaration location: paper-local

Lean declaration body

type:
ℝ → ℝ → ℝ → ℝ

value:
fun (alpha left right : ℝ) => alpha * left / (alpha * left + (1 - alpha) * right)

GCG24UserItemFairness.OpposingTypes.pairShare

Declaration location: paper-local

Lean declaration body

type:
{n : ℕ} → ℝ → (GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.Item n → ℝ

value:
fun {n : ℕ} (alpha : ℝ) (v : GCG24UserItemFairness.Item n → ℝ) (j : GCG24UserItemFairness.Item n) =>
  GCG24UserItemFairness.OpposingTypes.typeOneShare alpha (v j) (v (GCG24UserItemFairness.OpposingTypes.reverseItem j))

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.reverseItem
- GCG24UserItemFairness.OpposingTypes.typeOneShare

GCG24UserItemFairness.OpposingTypes.problem6LeftSum

Declaration location: paper-local

Lean declaration body

type:
{n : ℕ} → ℝ → (GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.Item n → ℝ

value:
fun {n : ℕ} (alpha : ℝ) (v : GCG24UserItemFairness.Item n → ℝ) (t : GCG24UserItemFairness.Item n) =>
  ∑ j : GCG24UserItemFairness.Item n, if ↑j < ↑t then (GCG24UserItemFairness.OpposingTypes.pairShare alpha v j)⁻¹ else 0

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.pairShare

GCG24UserItemFairness.OpposingTypes.problem6RightSum

Declaration location: paper-local

Lean declaration body

type:
{n : ℕ} → ℝ → (GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.Item n → ℝ

value:
fun {n : ℕ} (alpha : ℝ) (v : GCG24UserItemFairness.Item n → ℝ) (t : GCG24UserItemFairness.Item n) =>
  ∑ j : GCG24UserItemFairness.Item n,
    if ↑t < ↑j then (1 - GCG24UserItemFairness.OpposingTypes.pairShare alpha v j)⁻¹ else 0

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.pairShare

GCG24UserItemFairness.OpposingTypes.problem6ClosedDenominator

Declaration location: paper-local

Lean declaration body

```
type:
{n : ℕ} → ℝ → (GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.Item n → ℝ

value:
fun {n : ℕ} (alpha : ℝ) (v : GCG24UserItemFairness.Item n → ℝ) (t : GCG24UserItemFairness.Item n) =>
  1 +
    GCG24UserItemFairness.OpposingTypes.pairShare alpha v t *
      GCG24UserItemFairness.OpposingTypes.problem6LeftSum alpha v t +
  (1 - GCG24UserItemFairness.OpposingTypes.pairShare alpha v t) *
    GCG24UserItemFairness.OpposingTypes.problem6RightSum alpha v t
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.pairShare
- GCG24UserItemFairness.OpposingTypes.problem6LeftSum
- GCG24UserItemFairness.OpposingTypes.problem6RightSum

GCG24UserItemFairness.OpposingTypes.problem6ClosedValue

Declaration location: paper-local

Lean declaration body

```
type:
{n : ℕ} → ℝ → (GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.Item n → ℝ

value:
fun {n : ℕ} (alpha : ℝ) (v : GCG24UserItemFairness.Item n → ℝ) (t : GCG24UserItemFairness.Item n) =>
  1 / GCG24UserItemFairness.OpposingTypes.problem6ClosedDenominator alpha v t
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.problem6ClosedDenominator

GCG24UserItemFairness.OpposingTypes.problem6ClosedX

Declaration location: paper-local

Lean declaration body

```
type:
{n : ℕ} → ℝ → (GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.Item n → GCG24UserItemFairness.Item n → ℝ

value:
fun {n : ℕ} (alpha : ℝ) (v : GCG24UserItemFairness.Item n → ℝ) (t j : GCG24UserItemFairness.Item n) =>
  if ↑j < ↑t then
    GCG24UserItemFairness.OpposingTypes.problem6ClosedValue alpha v t /
      GCG24UserItemFairness.OpposingTypes.pairShare alpha v j
  else
    if j = t then
      1 -
        GCG24UserItemFairness.OpposingTypes.problem6ClosedValue alpha v t *
          GCG24UserItemFairness.OpposingTypes.problem6LeftSum alpha v t
    else 0
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.pairShare
- GCG24UserItemFairness.OpposingTypes.problem6ClosedValue
- GCG24UserItemFairness.OpposingTypes.problem6LeftSum

GCG24UserItemFairness.OpposingTypes.problem6ClosedY

Declaration location: paper-local

Lean declaration body

```
type:
{n : ℕ} → ℝ → (GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.Item n → GCG24UserItemFairness.Item n → ℝ

value:
fun {n : ℕ} (alpha : ℝ) (v : GCG24UserItemFairness.Item n → ℝ) (t j : GCG24UserItemFairness.Item n) =>
  if ↑j < ↑t then 0
  else
    if j = t then
      1 -
        GCG24UserItemFairness.OpposingTypes.problem6ClosedValue alpha v t *
          GCG24UserItemFairness.OpposingTypes.problem6RightSum alpha v t
    else
      GCG24UserItemFairness.OpposingTypes.problem6ClosedValue alpha v t /
        (1 - GCG24UserItemFairness.OpposingTypes.pairShare alpha v j)
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.pairShare
- GCG24UserItemFairness.OpposingTypes.problem6ClosedValue
- GCG24UserItemFairness.OpposingTypes.problem6RightSum

GCG24UserItemFairness.OpposingTypes.Problem6ClosedNonnegativePivots

Declaration location: paper-local

Lean declaration body

```
field x_pivot_nonneg:
∀ {n : ℕ} {alpha : ℝ} {v : GCG24UserItemFairness.Item n → ℝ} {t : GCG24UserItemFairness.Item n},
  GCG24UserItemFairness.OpposingTypes.Problem6ClosedNonnegativePivots alpha v t →
    0 ≤ GCG24UserItemFairness.OpposingTypes.problem6ClosedX alpha v t t

field y_pivot_nonneg:
∀ {n : ℕ} {alpha : ℝ} {v : GCG24UserItemFairness.Item n → ℝ} {t : GCG24UserItemFairness.Item n},
  GCG24UserItemFairness.OpposingTypes.Problem6ClosedNonnegativePivots alpha v t →
    0 ≤ GCG24UserItemFairness.OpposingTypes.problem6ClosedY alpha v t t
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.problem6ClosedX
- GCG24UserItemFairness.OpposingTypes.problem6ClosedY

GCG24UserItemFairness.OpposingTypes.problem6PivotGap

Declaration location: paper-local

Lean declaration body

```
type:
{n : ℕ} → ℝ → (GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.Item n → ℝ

value:
fun {n : ℕ} (alpha : ℝ) (v : GCG24UserItemFairness.Item n → ℝ) (t : GCG24UserItemFairness.Item n) =>
  GCG24UserItemFairness.OpposingTypes.problem6LeftSum alpha v t -
    GCG24UserItemFairness.OpposingTypes.problem6RightSum alpha v t
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.problem6LeftSum
- GCG24UserItemFairness.OpposingTypes.problem6RightSum

GCG24UserItemFairness.OpposingTypes.problem6PivotCrossingSet

Declaration location: paper-local

Lean declaration body

```
type:
{n : ℕ} → ℝ → (GCG24UserItemFairness.Item n → ℝ) → Finset (GCG24UserItemFairness.Item n)

value:
fun {n : ℕ} (alpha : ℝ) (v : GCG24UserItemFairness.Item n → ℝ) =>
  {t : GCG24UserItemFairness.Item n |
    -(GCG24UserItemFairness.OpposingTypes.pairShare alpha v t)⁻¹ ≤
      GCG24UserItemFairness.OpposingTypes.problem6PivotGap alpha v t}
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.pairShare
- GCG24UserItemFairness.OpposingTypes.problem6PivotGap

GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPivot

Declaration location: paper-local

Lean declaration body

```
type:
{n : ℕ} →
  [NeZero n] →
    (alpha : ℝ) →
      (v : GCG24UserItemFairness.Item n → ℝ) →
        0 < alpha → alpha < 1 → (∀ (j : GCG24UserItemFairness.Item n), 0 < v j) → GCG24UserItemFairness.Item n

value:
fun {n : ℕ} [NeZero n] (alpha : ℝ) (v : GCG24UserItemFairness.Item n → ℝ) (halpha0 : 0 < alpha) (halpha1 : alpha < 1)
    (hpos : ∀ (j : GCG24UserItemFairness.Item n), 0 < v j) =>
  Classical.choose (GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPivot._proof_1 alpha v halpha0 halpha1 hpos)
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.problem6PivotCrossingSet

GCG24UserItemFairness.OpposingTypes.theorem4Problem11ItemValue

Declaration location: paper-local

Lean declaration body

```
type:
{n : ℕ} →
  ℝ →
    (GCG24UserItemFairness.Item n → ℝ) →
      (GCG24UserItemFairness.Item n → ℝ) → (GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.Item n → ℝ

value:
fun {n : ℕ} (beta : ℝ) (v x z : GCG24UserItemFairness.Item n → ℝ) (j : GCG24UserItemFairness.Item n) =>
  2 * beta *
      (GCG24UserItemFairness.OpposingTypes.pairShare (1 / 2) v j * x j +
        (1 - GCG24UserItemFairness.OpposingTypes.pairShare (1 / 2) v j) *
          x (GCG24UserItemFairness.OpposingTypes.reverseItem j)) +
    (1 - 2 * beta) * z j
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.pairShare
- GCG24UserItemFairness.OpposingTypes.reverseItem

GCG24UserItemFairness.OpposingTypes.theorem4Problem11LeftSum

Declaration location: paper-local

Lean declaration body

type:
{n : ℕ} → (GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.Item n → ℝ

value:
fun {n : ℕ} (v : GCG24UserItemFairness.Item n → ℝ) (t : GCG24UserItemFairness.Item n) =>
  ∑ j : GCG24UserItemFairness.Item n,
    if ↑j < ↑t then (GCG24UserItemFairness.OpposingTypes.pairShare (1 / 2) v j)⁻¹ else 0

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.pairShare

GCG24UserItemFairness.OpposingTypes.paper_lemma15_problem11_leftSum

Declaration location: paper-local

Lean declaration body

type:
{n : ℕ} → (GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.Item n → ℝ

value:
fun {n : ℕ} (v : GCG24UserItemFairness.Item n → ℝ) (t : GCG24UserItemFairness.Item n) =>
  GCG24UserItemFairness.OpposingTypes.theorem4Problem11LeftSum v t

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.theorem4Problem11LeftSum

GCG24UserItemFairness.ProofBridge.Problem11Point

Declaration location: paper-local

Lean declaration body

type:
ℕ → Type

value:
fun (n : ℕ) => ((GCG24UserItemFairness.Item n → ℝ) × (GCG24UserItemFairness.Item n → ℝ)) × ℝ

Direct retained dependencies

- GCG24UserItemFairness.Item

GCG24UserItemFairness.RecommendationModel.PolicyLinearComparison

Declaration location: paper-local

Lean declaration body

type:
Type

constructor GCG24UserItemFairness.RecommendationModel.PolicyLinearComparison.eq:
GCG24UserItemFairness.RecommendationModel.PolicyLinearComparison

constructor GCG24UserItemFairness.RecommendationModel.PolicyLinearComparison.le:
GCG24UserItemFairness.RecommendationModel.PolicyLinearComparison

GCG24UserItemFairness.User

Declaration location: paper-local

Lean declaration body

type:
ℕ → Type

value:
fun (m : ℕ) => Fin m

GCG24UserItemFairness.Policy

Declaration location: paper-local

Lean declaration body

type:
ℕ → ℕ → Type

value:
fun (m n : ℕ) => AppliedModelingLib.Policy (GCG24UserItemFairness.User m) (GCG24UserItemFairness.Item n)

Direct retained dependencies

- AppliedModelingLib.Policy
- GCG24UserItemFairness.Item
- GCG24UserItemFairness.User

GCG24UserItemFairness.RecommendationModel

Declaration location: paper-local

Lean declaration body

field utility:
{m n : ℕ} →
  GCG24UserItemFairness.RecommendationModel m n → GCG24UserItemFairness.User m → GCG24UserItemFairness.Item n → ℝ

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.User

GCG24UserItemFairness.EstimatedRecommendationModel

Declaration location: paper-local

Lean declaration body

field trueModel:
{m n : ℕ} → GCG24UserItemFairness.EstimatedRecommendationModel m n → GCG24UserItemFairness.RecommendationModel m n

field estimatedUtility:
{m n : ℕ} →
  GCG24UserItemFairness.EstimatedRecommendationModel m n →
    GCG24UserItemFairness.User m → GCG24UserItemFairness.Item n → ℝ

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.User

GCG24UserItemFairness.EstimatedRecommendationModel.estimatedModel

Declaration location: paper-local

Lean declaration body

type:
{m n : ℕ} → GCG24UserItemFairness.EstimatedRecommendationModel m n → GCG24UserItemFairness.RecommendationModel m n

value:
fun {m n : ℕ} (E : GCG24UserItemFairness.EstimatedRecommendationModel m n) => { utility := E.estimatedUtility }

Direct retained dependencies

- GCG24UserItemFairness.EstimatedRecommendationModel
- GCG24UserItemFairness.RecommendationModel

GCG24UserItemFairness.RecommendationModel.Positive

Declaration location: paper-local

Lean declaration body

type:
{m n : ℕ} → GCG24UserItemFairness.RecommendationModel m n → Prop

value:
fun {m n : ℕ} (W : GCG24UserItemFairness.RecommendationModel m n) =>
  ∀ (u : GCG24UserItemFairness.User m) (j : GCG24UserItemFairness.Item n), 0 < W.utility u j

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.User

GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Variable

Declaration location: paper-local

Lean declaration body

type:
ℕ → ℕ → Type

value:
fun (m n : ℕ) => GCG24UserItemFairness.User m × GCG24UserItemFairness.Item n ⊕ Unit

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.User

GCG24UserItemFairness.RecommendationModel.UserTypeAssignment

Declaration location: paper-local

Lean declaration body

field toType:
{m K : ℕ} → GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K → GCG24UserItemFairness.User m → Fin K

Direct retained dependencies

- GCG24UserItemFairness.User

GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP

Declaration location: paper-local

Lean declaration body

```
field model:
{m n K : ℕ} →
  GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP m n K →
    GCG24UserItemFairness.RecommendationModel m n

field types:
{m n K : ℕ} →
  GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP m n K →
    GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K
```

Direct retained dependencies

- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment

GCG24UserItemFairness.RecommendationModel.UserTypeAssignment.UtilityRowsDetermineTypes

Declaration location: paper-local

Lean declaration body

```
type:
{m n K : ℕ} →
  GCG24UserItemFairness.RecommendationModel m n →
    GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K → Prop

value:
fun {m n K : ℕ} (W : GCG24UserItemFairness.RecommendationModel m n)
    (τ : GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K) =>
  ∀ (u u' : GCG24UserItemFairness.User m), W.utility u = W.utility u' → τ.toType u = τ.toType u'
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.User

GCG24UserItemFairness.RecommendationModel.UtilitiesAgreeWithinTypes

Declaration location: paper-local

Lean declaration body

```
type:
{m n K : ℕ} →
  GCG24UserItemFairness.RecommendationModel m n →
    GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K → Prop

value:
fun {m n K : ℕ} (W : GCG24UserItemFairness.RecommendationModel m n)
    (τ : GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K) =>
  ∀ (u u' : GCG24UserItemFairness.User m), τ.toType u = τ.toType u' → W.utility u = W.utility u'
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.User

GCG24UserItemFairness.RecommendationModel.SymmetricData

Declaration location: paper-local

Lean declaration body

field model:
{m n K : ℕ} →
  GCG24UserItemFairness.RecommendationModel.SymmetricData m n K → GCG24UserItemFairness.RecommendationModel m n

field types:
{m n K : ℕ} →
  GCG24UserItemFairness.RecommendationModel.SymmetricData m n K →
    GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K

field agreeWithinTypes:
∀ {m n K : ℕ} (self : GCG24UserItemFairness.RecommendationModel.SymmetricData m n K),
  self.model.UtilitiesAgreeWithinTypes self.types

Direct retained dependencies

- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.RecommendationModel.UtilitiesAgreeWithinTypes

GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.ofSymmetricData

Declaration location: paper-local

Lean declaration body

type:
{m n K : ℕ} →
  GCG24UserItemFairness.RecommendationModel.SymmetricData m n K →
    GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP m n K

value:
fun {m n K : ℕ} (S : GCG24UserItemFairness.RecommendationModel.SymmetricData m n K) =>
  { model := S.model, types := S.types }

Direct retained dependencies

- GCG24UserItemFairness.RecommendationModel.SymmetricData
- GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP

GCG24UserItemFairness.RecommendationModel.UtilityRowSymmetric

Declaration location: paper-local

Lean declaration body

type:
{m n : ℕ} → GCG24UserItemFairness.RecommendationModel m n → GCG24UserItemFairness.Policy m n → Prop

value:
fun {m n : ℕ} (W : GCG24UserItemFairness.RecommendationModel m n) (ρ : GCG24UserItemFairness.Policy m n) =>
  ∀ (u u' : GCG24UserItemFairness.User m), W.utility u = W.utility u' → ρ u = ρ u'

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.User

GCG24UserItemFairness.RecommendationModel.bestItemUtility

Declaration location: paper-local

Lean declaration body

type:
{m n : ℕ} → [NeZero n] → GCG24UserItemFairness.RecommendationModel m n → GCG24UserItemFairness.User m → ℝ

value:
fun {m n : ℕ} [NeZero n] (W : GCG24UserItemFairness.RecommendationModel m n) (u : GCG24UserItemFairness.User m) =>
  AppliedModelingLib.finiteMax (W.utility u)

Direct retained dependencies

- AppliedModelingLib.finiteMax
- GCG24UserItemFairness.Item
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.User

GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.itemEqualityNormal

Declaration location: paper-local

Lean declaration body

```
type:
{m n K : ℕ} →
  GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP m n K →
    GCG24UserItemFairness.Item n → GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Variable m n → ℝ

value:
fun {m n K : ℕ} (L : GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP m n K)
    (j : GCG24UserItemFairness.Item n)
    (a : GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Variable m n) =>
  match a with
  | Sum.inl (u, j') => if j' = j then L.model.utility u j / L.model.itemNormalizer j else 0
  | Sum.inr val => -1
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP
- GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Variable
- GCG24UserItemFairness.RecommendationModel.itemNormalizer
- GCG24UserItemFairness.User

GCG24UserItemFairness.RecommendationModel.policyLinearExpression

Declaration location: paper-local

Lean declaration body

```
type:
{m n : ℕ} → (GCG24UserItemFairness.User m → GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.Policy m n → ℝ

value:
fun {m n : ℕ} (a : GCG24UserItemFairness.User m → GCG24UserItemFairness.Item n → ℝ)
    (ρ : GCG24UserItemFairness.Policy m n) =>
  ∑ u : GCG24UserItemFairness.User m, ∑ j : GCG24UserItemFairness.Item n, a u j * ((ρ u) j).toReal
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.User

GCG24UserItemFairness.RecommendationModel.policyLinearConstraintHolds

Declaration location: paper-local

Lean declaration body

```
type:
{m n : ℕ} →
  (GCG24UserItemFairness.User m → GCG24UserItemFairness.Item n → ℝ) →
    ℝ → GCG24UserItemFairness.RecommendationModel.PolicyLinearComparison → GCG24UserItemFairness.Policy m n → Prop

value:
fun {m n : ℕ} (a : GCG24UserItemFairness.User m → GCG24UserItemFairness.Item n → ℝ) (b : ℝ)
    (rel : GCG24UserItemFairness.RecommendationModel.PolicyLinearComparison) (ρ : GCG24UserItemFairness.Policy m n) =>
  match rel with
  | GCG24UserItemFairness.RecommendationModel.PolicyLinearComparison.eq =>
    GCG24UserItemFairness.RecommendationModel.policyLinearExpression a ρ = b
  | GCG24UserItemFairness.RecommendationModel.PolicyLinearComparison.le =>
    GCG24UserItemFairness.RecommendationModel.policyLinearExpression a ρ ≤ b
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel.PolicyLinearComparison
- GCG24UserItemFairness.RecommendationModel.policyLinearExpression
- GCG24UserItemFairness.User

GCG24UserItemFairness.RecommendationModel.FiniteLinearPolicySetDescription

Declaration location: paper-local

Lean declaration body

```
field ConstraintIndex:
{m n : ℕ} →
  {S : Set (GCG24UserItemFairness.Policy m n)} →
    GCG24UserItemFairness.RecommendationModel.FiniteLinearPolicySetDescription S → Type

field constraintFintype:
{m n : ℕ} →
  {S : Set (GCG24UserItemFairness.Policy m n)} →
    (self : GCG24UserItemFairness.RecommendationModel.FiniteLinearPolicySetDescription S) → Fintype self.ConstraintIndex

field coefficient:
{m n : ℕ} →
  {S : Set (GCG24UserItemFairness.Policy m n)} →
    (self : GCG24UserItemFairness.RecommendationModel.FiniteLinearPolicySetDescription S) →
      self.ConstraintIndex → GCG24UserItemFairness.User m → GCG24UserItemFairness.Item n → ℝ

field rhs:
{m n : ℕ} →
  {S : Set (GCG24UserItemFairness.Policy m n)} →
    (self : GCG24UserItemFairness.RecommendationModel.FiniteLinearPolicySetDescription S) → self.ConstraintIndex → ℝ

field comparison:
{m n : ℕ} →
  {S : Set (GCG24UserItemFairness.Policy m n)} →
    (self : GCG24UserItemFairness.RecommendationModel.FiniteLinearPolicySetDescription S) →
      self.ConstraintIndex → GCG24UserItemFairness.RecommendationModel.PolicyLinearComparison

field mem_iff:
∀ {m n : ℕ} {S : Set (GCG24UserItemFairness.Policy m n)}
  (self : GCG24UserItemFairness.RecommendationModel.FiniteLinearPolicySetDescription S)
  (ρ : GCG24UserItemFairness.Policy m n),
  ρ ∈ S ↔
    ∀ (c : self.ConstraintIndex),
      GCG24UserItemFairness.RecommendationModel.policyLinearConstraintHolds (self.coefficient c) (self.rhs c)
        (self.comparison c) ρ
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel.PolicyLinearComparison
- GCG24UserItemFairness.RecommendationModel.policyLinearConstraintHolds
- GCG24UserItemFairness.User

GCG24UserItemFairness.RecommendationModel.FiniteLinearPolicySetDescription.Satisfies

Declaration location: paper-local

Lean declaration body

```
type:
{m n : ℕ} →
  {S : Set (GCG24UserItemFairness.Policy m n)} →
    GCG24UserItemFairness.RecommendationModel.FiniteLinearPolicySetDescription S →
      GCG24UserItemFairness.Policy m n → Prop

value:
fun {m n : ℕ} {S : Set (GCG24UserItemFairness.Policy m n)}
    (D : GCG24UserItemFairness.RecommendationModel.FiniteLinearPolicySetDescription S)
    (ρ : GCG24UserItemFairness.Policy m n) =>
  ∀ (c : D.ConstraintIndex),
    GCG24UserItemFairness.RecommendationModel.policyLinearConstraintHolds (D.coefficient c) (D.rhs c) (D.comparison c) ρ
```

Direct retained dependencies

- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel.FiniteLinearPolicySetDescription
- GCG24UserItemFairness.RecommendationModel.policyLinearConstraintHolds

GCG24UserItemFairness.RecommendationModel.propositionOneItemEqualityExpression

Declaration location: paper-local

Lean declaration body

```
type:
{m n : ℕ} →
  GCG24UserItemFairness.RecommendationModel m n →
    GCG24UserItemFairness.Item n → GCG24UserItemFairness.Policy m n → ℝ → ℝ

value:
fun {m n : ℕ} (W : GCG24UserItemFairness.RecommendationModel m n) (j : GCG24UserItemFairness.Item n)
    (ρ : GCG24UserItemFairness.Policy m n) (ell : ℝ) =>
  ∑ u : GCG24UserItemFairness.User m, W.utility u j / W.itemNormalizer j * ((ρ u) j).toReal - ell
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.itemNormalizer
- GCG24UserItemFairness.User

GCG24UserItemFairness.RecommendationModel.PropositionOneLPFeasible

Declaration location: paper-local

Lean declaration body

```
type:
{m n : ℕ} →
  {S : Set (GCG24UserItemFairness.Policy m n)} →
    GCG24UserItemFairness.RecommendationModel m n →
      GCG24UserItemFairness.RecommendationModel.FiniteLinearPolicySetDescription S →
        GCG24UserItemFairness.Policy m n → ℝ → Prop

value:
fun {m n : ℕ} {S : Set (GCG24UserItemFairness.Policy m n)} (W : GCG24UserItemFairness.RecommendationModel m n)
    (D : GCG24UserItemFairness.RecommendationModel.FiniteLinearPolicySetDescription S)
    (ρ : GCG24UserItemFairness.Policy m n) (ell : ℝ) =>
  D.Satisfies ρ ∧ ∀ (j : GCG24UserItemFairness.Item n), W.propositionOneItemEqualityExpression j ρ ell = 0
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.FiniteLinearPolicySetDescription
- GCG24UserItemFairness.RecommendationModel.FiniteLinearPolicySetDescription.Satisfies
- GCG24UserItemFairness.RecommendationModel.propositionOneItemEqualityExpression

GCG24UserItemFairness.RecommendationModel.attainableItemFairnessSet

Declaration location: paper-local

Lean declaration body

```
type:
{m n : ℕ} → [NeZero n] → GCG24UserItemFairness.RecommendationModel m n → ℝ → Prop

value:
fun {m n : ℕ} [NeZero n] (W : GCG24UserItemFairness.RecommendationModel m n) (a : ℝ) =>
  {r : ℝ | ∃ (ρ : GCG24UserItemFairness.Policy m n), r = W.itemFairness ρ a
```

Direct retained dependencies

- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.itemFairness

GCG24UserItemFairness.RecommendationModel.itemFairnessEqualityLPFeasible

Declaration location: paper-local

Lean declaration body

type:
{m n : ℕ} → GCG24UserItemFairness.RecommendationModel m n → GCG24UserItemFairness.Policy m n → ℝ → Prop

value:
fun {m n : ℕ} (W : GCG24UserItemFairness.RecommendationModel m n) (ρ : GCG24UserItemFairness.Policy m n) (ell : ℝ) =>
  ∀ (j : GCG24UserItemFairness.Item n), W.normalizedItemUtility ρ j = ell

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.normalizedItemUtility

GCG24UserItemFairness.RecommendationModel.optimalItemFairness

Declaration location: paper-local

Lean declaration body

type:
{m n : ℕ} → [NeZero n] → GCG24UserItemFairness.RecommendationModel m n → ℝ

value:
fun {m n : ℕ} [NeZero n] (W : GCG24UserItemFairness.RecommendationModel m n) => sSup W.attainableItemFairnessSet

Direct retained dependencies

- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.attainableItemFairnessSet

GCG24UserItemFairness.RecommendationModel.feasibleAtLevel

Declaration location: paper-local

Lean declaration body

type:
{m n : ℕ} → [NeZero n] → GCG24UserItemFairness.RecommendationModel m n → ℝ → GCG24UserItemFairness.Policy m n → Prop

value:
fun {m n : ℕ} [NeZero n] (W : GCG24UserItemFairness.RecommendationModel m n) (γ : ℝ)
    (ρ : GCG24UserItemFairness.Policy m n) =>
  γ * W.optimalItemFairness ≤ W.itemFairness ρ

Direct retained dependencies

- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.itemFairness
- GCG24UserItemFairness.RecommendationModel.optimalItemFairness

GCG24UserItemFairness.RecommendationModel.attainableUserFairnessAtLevel

Declaration location: paper-local

Lean declaration body

type:
{m n : ℕ} → [NeZero m] → [NeZero n] → GCG24UserItemFairness.RecommendationModel m n → ℝ → ℝ → Prop

value:
fun {m n : ℕ} [NeZero m] [NeZero n] (W : GCG24UserItemFairness.RecommendationModel m n) (γ a : ℝ) =>
  {r : ℝ | ∃ (ρ : GCG24UserItemFairness.Policy m n), W.feasibleAtLevel γ ρ ∧ r = W.userFairness ρ} a

Direct retained dependencies

- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.feasibleAtLevel
- GCG24UserItemFairness.RecommendationModel.userFairness

GCG24UserItemFairness.RecommendationModel.optimalUserFairnessAtLevel

Declaration location: paper-local

Lean declaration body

type:
{m n : ℕ} → [NeZero m] → [NeZero n] → GCG24UserItemFairness.RecommendationModel m n → ℝ → ℝ

value:
fun {m n : ℕ} [NeZero m] [NeZero n] (W : GCG24UserItemFairness.RecommendationModel m n) (γ : ℝ) =>
  sSup (W.attainableUserFairnessAtLevel γ)

Direct retained dependencies

- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.attainableUserFairnessAtLevel

GCG24UserItemFairness.EstimatedRecommendationModel.SolvesEstimatedProblem

Declaration location: paper-local

Lean declaration body

type:
{m n : ℕ} →
  [NeZero m] →
    [NeZero n] → GCG24UserItemFairness.EstimatedRecommendationModel m n → ℝ → GCG24UserItemFairness.Policy m n → Prop

value:
fun {m n : ℕ} [NeZero m] [NeZero n] (E : GCG24UserItemFairness.EstimatedRecommendationModel m n) (γ : ℝ)
    (ρ : GCG24UserItemFairness.Policy m n) =>
  E.estimatedModel.IsOptimalAtLevel γ ρ

Direct retained dependencies

- GCG24UserItemFairness.EstimatedRecommendationModel
- GCG24UserItemFairness.EstimatedRecommendationModel.estimatedModel
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel.IsOptimalAtLevel

GCG24UserItemFairness.RecommendationModel.priceOfFairnessAt

Declaration location: paper-local

Lean declaration body

type:
{m n : ℕ} → [NeZero m] → [NeZero n] → GCG24UserItemFairness.RecommendationModel m n → ℝ → ℝ

value:
fun {m n : ℕ} [NeZero m] [NeZero n] (W : GCG24UserItemFairness.RecommendationModel m n) (γ : ℝ) =>
  have base : ℝ := W.optimalUserFairnessAtLevel 0;
  if h : base = 0 then 0 else (base - W.optimalUserFairnessAtLevel γ) / base

Direct retained dependencies

- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.optimalUserFairnessAtLevel

GCG24UserItemFairness.UserType

Declaration location: paper-local

Lean declaration body

```
type:
ℕ → Type

value:
fun (K : ℕ) => Fin K
```

GCG24UserItemFairness.OpposingTypes.problem6ClosedPolicy

Declaration location: paper-local

Lean declaration body

```
type:
{n : ℕ} →
  (alpha : ℝ) →
    (v : GCG24UserItemFairness.Item n → ℝ) →
      (t : GCG24UserItemFairness.Item n) →
        0 < alpha →
          alpha < 1 →
            (∀ (j : GCG24UserItemFairness.Item n), 0 < v j) →
              GCG24UserItemFairness.OpposingTypes.Problem6ClosedNonnegativePivots alpha v t →
                GCG24UserItemFairness.UserType 2 → PMF (GCG24UserItemFairness.Item n)

value:
fun {n : ℕ} (alpha : ℝ) (v : GCG24UserItemFairness.Item n → ℝ) (t : GCG24UserItemFairness.Item n) (halpha0 : 0 < alpha)
    (halpha1 : alpha < 1) (hpos : ∀ (j : GCG24UserItemFairness.Item n), 0 < v j)
    (hpivot : GCG24UserItemFairness.OpposingTypes.Problem6ClosedNonnegativePivots alpha v t)
    (a : GCG24UserItemFairness.UserType 2) =>
  if a = 0 then
    GCG24UserItemFairness.OpposingTypes.pmfOfRealVector (GCG24UserItemFairness.OpposingTypes.problem6ClosedX alpha v t)
      (GCG24UserItemFairness.OpposingTypes.problem6ClosedX_nonneg halpha0 halpha1 hpos hpivot)
      (GCG24UserItemFairness.OpposingTypes.problem6ClosedX_sum_eq_one alpha v t)
  else
    GCG24UserItemFairness.OpposingTypes.pmfOfRealVector (GCG24UserItemFairness.OpposingTypes.problem6ClosedY alpha v t)
      (GCG24UserItemFairness.OpposingTypes.problem6ClosedY_nonneg halpha0 halpha1 hpos hpivot)
      (GCG24UserItemFairness.OpposingTypes.problem6ClosedY_sum_eq_one alpha v t)
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.Problem6ClosedNonnegativePivots
- GCG24UserItemFairness.OpposingTypes.pmfOfRealVector
- GCG24UserItemFairness.OpposingTypes.problem6ClosedX
- GCG24UserItemFairness.OpposingTypes.problem6ClosedY
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy

Declaration location: paper-local

Lean declaration body

```
type:
{n : ℕ} →
  [NeZero n] →
    (alpha : ℝ) →
      (v : GCG24UserItemFairness.Item n → ℝ) →
        0 < alpha →
          alpha < 1 →
            (∀ (j : GCG24UserItemFairness.Item n), 0 < v j) →
              GCG24UserItemFairness.UserType 2 → PMF (GCG24UserItemFairness.Item n)

value:
fun {n : ℕ} [NeZero n] (alpha : ℝ) (v : GCG24UserItemFairness.Item n → ℝ) (halpha0 : 0 < alpha) (halpha1 : alpha < 1)
```

```
  (hpos : ∀ (j : GCG24UserItemFairness.Item n), 0 < v j) (a : GCG24UserItemFairness.UserType 2) =>
(let t : GCG24UserItemFairness.Item n :=
   GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPivot alpha v halpha0 halpha1 hpos;
 have hpivot : GCG24UserItemFairness.OpposingTypes.Problem6ClosedNonnegativePivots alpha v t :=
   GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy._proof_1 alpha v halpha0 halpha1 hpos;
 GCG24UserItemFairness.OpposingTypes.problem6ClosedPolicy alpha v t halpha0 halpha1 hpos hpivot)
 a
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.Problem6ClosedNonnegativePivots
- GCG24UserItemFairness.OpposingTypes.problem6ClosedPolicy
- GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPivot
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.RecommendationModel.UserTypeAssignment.typeCard

Declaration location: paper-local

Lean declaration body

type:
```
{m K : ℕ} → GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K → GCG24UserItemFairness.UserType K → ℕ
```

value:
```
fun {m K : ℕ} (τ : GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K)
  (k : GCG24UserItemFairness.UserType K) =>
 {u : GCG24UserItemFairness.User m | τ.toType u = k}.card
```

Direct retained dependencies

- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.User
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.RecommendationModel.UserTypeAssignment.typeWeight

Declaration location: paper-local

Lean declaration body

type:
```
{m K : ℕ} → GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K → GCG24UserItemFairness.UserType K → ℝ
```

value:
```
fun {m K : ℕ} (τ : GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K)
  (k : GCG24UserItemFairness.UserType K) =>
 ↑(τ.typeCard k) / ↑m
```

Direct retained dependencies

- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment.typeCard
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.TypePolicy

Declaration location: paper-local

Lean declaration body

type:
```
ℕ → ℕ → Type
```

value:
```
fun (K n : ℕ) => AppliedModelingLib.Policy (GCG24UserItemFairness.UserType K) (GCG24UserItemFairness.Item n)
```

Direct retained dependencies

- AppliedModelingLib.Policy
- GCG24UserItemFairness.Item
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy

Declaration location: paper-local

Lean declaration body

```
type:
{n : ℕ} → GCG24UserItemFairness.TypePolicy 3 n → Prop

value:
fun {n : ℕ} (ρ : GCG24UserItemFairness.TypePolicy 3 n) =>
  (∀ (j : GCG24UserItemFairness.Item n), (ρ 1) (GCG24UserItemFairness.OpposingTypes.reverseItem j) = (ρ 0) j) ∧
    ∀ (j : GCG24UserItemFairness.Item n), (ρ 2) (GCG24UserItemFairness.OpposingTypes.reverseItem j) = (ρ 2) j
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.reverseItem
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.OpposingTypes.problem6LPFeasible

Declaration location: paper-local

Lean declaration body

```
type:
{n : ℕ} → ℝ → (GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.TypePolicy 2 n → ℝ → Prop

value:
fun {n : ℕ} (alpha : ℝ) (v : GCG24UserItemFairness.Item n → ℝ) (ρ : GCG24UserItemFairness.TypePolicy 2 n) (ell : ℝ) =>
  ∀ (j : GCG24UserItemFairness.Item n),
    ell ≤
      GCG24UserItemFairness.OpposingTypes.pairShare alpha v j * ((ρ 0) j).toReal +
        (1 - GCG24UserItemFairness.OpposingTypes.pairShare alpha v j) * ((ρ 1) j).toReal
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.pairShare
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.OpposingTypes.theorem4Problem11PolicyItemValue

Declaration location: paper-local

Lean declaration body

```
type:
{n : ℕ} →
  ℝ → (GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.TypePolicy 3 n → GCG24UserItemFairness.Item n → ℝ

value:
fun {n : ℕ} (beta : ℝ) (v : GCG24UserItemFairness.Item n → ℝ) (ρ : GCG24UserItemFairness.TypePolicy 3 n)
    (j : GCG24UserItemFairness.Item n) =>
  GCG24UserItemFairness.OpposingTypes.theorem4Problem11ItemValue beta v
    (fun (l : GCG24UserItemFairness.Item n) => ((ρ 0) l).toReal)
    (fun (l : GCG24UserItemFairness.Item n) => ((ρ 2) l).toReal) j
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.theorem4Problem11ItemValue
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.OpposingTypes.theorem4Problem11LPFeasible

Declaration location: paper-local

Lean declaration body

type:
{n : ℕ} → ℝ → (GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.TypePolicy 3 n → ℝ → Prop

value:
fun {n : ℕ} (beta : ℝ) (v : GCG24UserItemFairness.Item n → ℝ) (ρ : GCG24UserItemFairness.TypePolicy 3 n) (ell : ℝ) =>
  ∀ (j : GCG24UserItemFairness.Item n),
    ell ≤ GCG24UserItemFairness.OpposingTypes.theorem4Problem11PolicyItemValue beta v ρ j

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.theorem4Problem11PolicyItemValue
- GCG24UserItemFairness.TypePolicy

GCG24UserItemFairness.OpposingTypes.theorem4Problem11TypeZeroActiveItems

Declaration location: paper-local

Lean declaration body

type:
{n : ℕ} → GCG24UserItemFairness.TypePolicy 3 n → Finset (GCG24UserItemFairness.Item n)

value:
fun {n : ℕ} (ρ : GCG24UserItemFairness.TypePolicy 3 n) => {j : GCG24UserItemFairness.Item n | (ρ 0) j ≠ 0}

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero

Declaration location: paper-local

Lean declaration body

type:
{n : ℕ} → [NeZero n] → GCG24UserItemFairness.TypePolicy 3 n → GCG24UserItemFairness.Item n

value:
fun {n : ℕ} [NeZero n] (ρ : GCG24UserItemFairness.TypePolicy 3 n) =>
  (GCG24UserItemFairness.OpposingTypes.theorem4Problem11TypeZeroActiveItems ρ).max'
    (GCG24UserItemFairness.OpposingTypes.theorem4Problem11TypeZeroActiveItems_nonempty ρ)

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.theorem4Problem11TypeZeroActiveItems
- GCG24UserItemFairness.TypePolicy

GCG24UserItemFairness.OpposingTypes.theorem4SymmetricX

Declaration location: paper-local

Lean declaration body

type:
{n : ℕ} → GCG24UserItemFairness.TypePolicy 3 n → GCG24UserItemFairness.Item n → ℝ

value:
fun {n : ℕ} (ρ : GCG24UserItemFairness.TypePolicy 3 n) (j : GCG24UserItemFairness.Item n) =>
  (((ρ 0) j).toReal + ((ρ 1) (GCG24UserItemFairness.OpposingTypes.reverseItem j)).toReal) / 2

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.reverseItem
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.OpposingTypes.theorem4SymmetricY

Declaration location: paper-local

Lean declaration body

type:
{n : ℕ} → GCG24UserItemFairness.TypePolicy 3 n → GCG24UserItemFairness.Item n → ℝ

value:
fun {n : ℕ} (ρ : GCG24UserItemFairness.TypePolicy 3 n) (j : GCG24UserItemFairness.Item n) =>
  (((ρ 1) j).toReal + ((ρ 0) (GCG24UserItemFairness.OpposingTypes.reverseItem j)).toReal) / 2

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.reverseItem
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.OpposingTypes.theorem4SymmetricZ

Declaration location: paper-local

Lean declaration body

type:
{n : ℕ} → GCG24UserItemFairness.TypePolicy 3 n → GCG24UserItemFairness.Item n → ℝ

value:
fun {n : ℕ} (ρ : GCG24UserItemFairness.TypePolicy 3 n) (j : GCG24UserItemFairness.Item n) =>
  (((ρ 2) j).toReal + ((ρ 2) (GCG24UserItemFairness.OpposingTypes.reverseItem j)).toReal) / 2

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.reverseItem
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.OpposingTypes.theorem4SymmetrizedPolicy

Declaration location: paper-local

Lean declaration body

type:
{n : ℕ} → GCG24UserItemFairness.TypePolicy 3 n → GCG24UserItemFairness.UserType 3 → PMF (GCG24UserItemFairness.Item n)

value:
```
fun {n : ℕ} (ρ : GCG24UserItemFairness.TypePolicy 3 n) (a : GCG24UserItemFairness.UserType 3) =>
  if a = 0 then
    GCG24UserItemFairness.OpposingTypes.pmfOfRealVector (GCG24UserItemFairness.OpposingTypes.theorem4SymmetricX ρ)
      (GCG24UserItemFairness.OpposingTypes.theorem4SymmetricX_nonneg ρ)
      (GCG24UserItemFairness.OpposingTypes.theorem4SymmetricX_sum_eq_one ρ)
  else
    if a = 1 then
      GCG24UserItemFairness.OpposingTypes.pmfOfRealVector (GCG24UserItemFairness.OpposingTypes.theorem4SymmetricY ρ)
        (GCG24UserItemFairness.OpposingTypes.theorem4SymmetricY_nonneg ρ)
        (GCG24UserItemFairness.OpposingTypes.theorem4SymmetricY_sum_eq_one ρ)
    else
      GCG24UserItemFairness.OpposingTypes.pmfOfRealVector (GCG24UserItemFairness.OpposingTypes.theorem4SymmetricZ ρ)
        (GCG24UserItemFairness.OpposingTypes.theorem4SymmetricZ_nonneg ρ)
        (GCG24UserItemFairness.OpposingTypes.theorem4SymmetricZ_sum_eq_one ρ)
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.pmfOfRealVector
- GCG24UserItemFairness.OpposingTypes.theorem4SymmetricX
- GCG24UserItemFairness.OpposingTypes.theorem4SymmetricY
- GCG24UserItemFairness.OpposingTypes.theorem4SymmetricZ
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.ProofBridge.problem11PointOfPolicy

Declaration location: paper-local

Lean declaration body

type:
{n : ℕ} → GCG24UserItemFairness.TypePolicy 3 n → ℝ → GCG24UserItemFairness.ProofBridge.Problem11Point n

value:
```
fun {n : ℕ} (ρ : GCG24UserItemFairness.TypePolicy 3 n) (ell : ℝ) =>
  ((fun (j : GCG24UserItemFairness.Item n) => ((ρ 0) j).toReal, fun (j : GCG24UserItemFairness.Item n) =>
      ((ρ 2) j).toReal),
    ell)
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.ProofBridge.Problem11Point
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.TypePolicy.activeTypeItemPairsCard

Declaration location: paper-local

Lean declaration body

type:
{K n : ℕ} → GCG24UserItemFairness.TypePolicy K n → ℕ

value:
```
fun {K n : ℕ} (ρ : GCG24UserItemFairness.TypePolicy K n) => AppliedModelingLib.Policy.activePairsCard ρ
```

Direct retained dependencies

- AppliedModelingLib.Policy.activePairsCard
- GCG24UserItemFairness.Item
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.TypePolicy.ActivePairsBound

Declaration location: paper-local

Lean declaration body

type:
{K n : ℕ} → GCG24UserItemFairness.TypePolicy K n → Prop

value:
fun {K n : ℕ} (ρ : GCG24UserItemFairness.TypePolicy K n) => ρ.activeTypeItemPairsCard ≤ n + K - 1

Direct retained dependencies

- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.TypePolicy.activeTypeItemPairsCard

GCG24UserItemFairness.TypePolicy.sharedItems

Declaration location: paper-local

Lean declaration body

type:
{K n : ℕ} → GCG24UserItemFairness.TypePolicy K n → Finset (GCG24UserItemFairness.Item n)

value:
fun {K n : ℕ} (ρ : GCG24UserItemFairness.TypePolicy K n) => AppliedModelingLib.Policy.multiAssignedActions ρ

Direct retained dependencies

- AppliedModelingLib.Policy.multiAssignedActions
- GCG24UserItemFairness.Item
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.TypePolicy.SharedItemsBound

Declaration location: paper-local

Lean declaration body

type:
{K n : ℕ} → GCG24UserItemFairness.TypePolicy K n → Prop

value:
fun {K n : ℕ} (ρ : GCG24UserItemFairness.TypePolicy K n) => ρ.sharedItems.card ≤ K - 1

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.TypePolicy.sharedItems

GCG24UserItemFairness.TypePolicy.typeZeroActiveItems

Declaration location: paper-local

Lean declaration body

type:
{n : ℕ} → GCG24UserItemFairness.TypePolicy 2 n → Finset (GCG24UserItemFairness.Item n)

value:
fun {n : ℕ} (ρ : GCG24UserItemFairness.TypePolicy 2 n) => {j : GCG24UserItemFairness.Item n | (ρ 0) j ≠ 0}

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.TypePolicy.lastActiveTypeZero

Declaration location: paper-local

Lean declaration body

```
type:
{n : ℕ} → [NeZero n] → GCG24UserItemFairness.TypePolicy 2 n → GCG24UserItemFairness.Item n

value:
fun {n : ℕ} [NeZero n] (ρ : GCG24UserItemFairness.TypePolicy 2 n) =>
  ρ.typeZeroActiveItems.max' (GCG24UserItemFairness.TypePolicy.typeZeroActiveItems_nonempty ρ)
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.TypePolicy.typeZeroActiveItems

GCG24UserItemFairness.TypeWeightedRecommendationModel

Declaration location: paper-local

Lean declaration body

```
field utility:
{K n : ℕ} →
  GCG24UserItemFairness.TypeWeightedRecommendationModel K n →
    GCG24UserItemFairness.UserType K → GCG24UserItemFairness.Item n → ℝ

field weight:
{K n : ℕ} → GCG24UserItemFairness.TypeWeightedRecommendationModel K n → GCG24UserItemFairness.UserType K → ℝ
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.OpposingTypes.theorem4EstimatedReducedModel

Declaration location: paper-local

Lean declaration body

```
type:
{n : ℕ} → ℝ → (GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.TypeWeightedRecommendationModel 3 n

value:
fun {n : ℕ} (beta : ℝ) (v : GCG24UserItemFairness.Item n → ℝ) =>
  {
    utility := fun (k : GCG24UserItemFairness.UserType 3) (j : GCG24UserItemFairness.Item n) =>
      if k = 0 then v j
      else
        if k = 1 then v (GCG24UserItemFairness.OpposingTypes.reverseItem j)
        else (v j + v (GCG24UserItemFairness.OpposingTypes.reverseItem j)) / 2,
    weight := fun (k : GCG24UserItemFairness.UserType 3) =>
      if k = 0 then beta else if k = 1 then beta else 1 - 2 * beta }
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.reverseItem
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel

Declaration location: paper-local

Lean declaration body

```
type:
{n : ℕ} → ℝ → (GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.TypeWeightedRecommendationModel 2 n

value:
fun {n : ℕ} (alpha : ℝ) (v : GCG24UserItemFairness.Item n → ℝ) =>
  {
    utility := fun (k : GCG24UserItemFairness.UserType 2) (j : GCG24UserItemFairness.Item n) =>
      if k = 0 then v j else v (GCG24UserItemFairness.OpposingTypes.reverseItem j),
    weight := fun (k : GCG24UserItemFairness.UserType 2) => if k = 0 then alpha else 1 - alpha }
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.reverseItem
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.ReductionWitness

Declaration location: paper-local

Lean declaration body

```
field data:
{m n K : ℕ} →
  GCG24UserItemFairness.ReductionWitness m n K → GCG24UserItemFairness.RecommendationModel.SymmetricData m n K

field reduced:
{m n K : ℕ} → GCG24UserItemFairness.ReductionWitness m n K → GCG24UserItemFairness.TypeWeightedRecommendationModel K n

field utility_agrees:
∀ {m n K : ℕ} (self : GCG24UserItemFairness.ReductionWitness m n K) (u : GCG24UserItemFairness.User m)
  (j : GCG24UserItemFairness.Item n), self.data.model.utility u j = self.reduced.utility (self.data.types.toType u) j

field weight_eq_typeWeight:
∀ {m n K : ℕ} (self : GCG24UserItemFairness.ReductionWitness m n K) (k : GCG24UserItemFairness.UserType K),
  self.reduced.weight k = self.data.types.typeWeight k
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.SymmetricData
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment.typeWeight
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.User
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.TypeWeightedRecommendationModel.bestItemUtility

Declaration location: paper-local

Lean declaration body

```
type:
{K n : ℕ} →
  [NeZero n] → GCG24UserItemFairness.TypeWeightedRecommendationModel K n → GCG24UserItemFairness.UserType K → ℝ

value:
fun {K n : ℕ} [NeZero n] (T : GCG24UserItemFairness.TypeWeightedRecommendationModel K n)
    (k : GCG24UserItemFairness.UserType K) =>
  AppliedModelingLib.finiteMax (T.utility k)
```

Direct retained dependencies

- AppliedModelingLib.finiteMax
- GCG24UserItemFairness.Item
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.TypeWeightedRecommendationModel.itemNormalizer

Declaration location: paper-local

Lean declaration body

```
type:
{K n : ℕ} → GCG24UserItemFairness.TypeWeightedRecommendationModel K n → GCG24UserItemFairness.Item n → ℝ

value:
fun {K n : ℕ} (T : GCG24UserItemFairness.TypeWeightedRecommendationModel K n) (j : GCG24UserItemFairness.Item n) =>
  ∑ k : GCG24UserItemFairness.UserType K, T.weight k * T.utility k j
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.TypeWeightedRecommendationModel.rawItemUtility

Declaration location: paper-local

Lean declaration body

```
type:
{K n : ℕ} →
  GCG24UserItemFairness.TypeWeightedRecommendationModel K n →
    GCG24UserItemFairness.TypePolicy K n → GCG24UserItemFairness.Item n → ℝ

value:
fun {K n : ℕ} (T : GCG24UserItemFairness.TypeWeightedRecommendationModel K n) (ρ : GCG24UserItemFairness.TypePolicy K n)
    (j : GCG24UserItemFairness.Item n) =>
  ∑ k : GCG24UserItemFairness.UserType K, T.weight k * T.utility k j * ((ρ k) j).toReal
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.TypeWeightedRecommendationModel.normalizedItemUtility

Declaration location: paper-local

Lean declaration body

```
type:
{K n : ℕ} →
  GCG24UserItemFairness.TypeWeightedRecommendationModel K n →
    GCG24UserItemFairness.TypePolicy K n → GCG24UserItemFairness.Item n → ℝ

value:
fun {K n : ℕ} (T : GCG24UserItemFairness.TypeWeightedRecommendationModel K n) (ρ : GCG24UserItemFairness.TypePolicy K n)
    (j : GCG24UserItemFairness.Item n) =>
  have denom : ℝ := T.itemNormalizer j;
  if h : denom = 0 then 0 else T.rawItemUtility ρ j / denom
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.TypeWeightedRecommendationModel.itemNormalizer
- GCG24UserItemFairness.TypeWeightedRecommendationModel.rawItemUtility

GCG24UserItemFairness.TypeWeightedRecommendationModel.itemFairness

Declaration location: paper-local

Lean declaration body

```
type:
{K n : ℕ} →
  [NeZero n] → GCG24UserItemFairness.TypeWeightedRecommendationModel K n → GCG24UserItemFairness.TypePolicy K n → ℝ

value:
fun {K n : ℕ} [NeZero n] (T : GCG24UserItemFairness.TypeWeightedRecommendationModel K n)
    (ρ : GCG24UserItemFairness.TypePolicy K n) =>
  AppliedModelingLib.finiteMin (T.normalizedItemUtility ρ)
```

Direct retained dependencies

- AppliedModelingLib.finiteMin
- GCG24UserItemFairness.Item
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.TypeWeightedRecommendationModel.normalizedItemUtility

GCG24UserItemFairness.TypeWeightedRecommendationModel.attainableItemFairnessSet

Declaration location: paper-local

Lean declaration body

```
type:
{K n : ℕ} → [NeZero n] → GCG24UserItemFairness.TypeWeightedRecommendationModel K n → ℝ → Prop

value:
fun {K n : ℕ} [NeZero n] (T : GCG24UserItemFairness.TypeWeightedRecommendationModel K n) (a : ℝ) =>
  {r : ℝ | ∃ (ρ : GCG24UserItemFairness.TypePolicy K n), r = T.itemFairness ρ} a
```

Direct retained dependencies

- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.TypeWeightedRecommendationModel.itemFairness

GCG24UserItemFairness.TypeWeightedRecommendationModel.optimalItemFairness

Declaration location: paper-local

Lean declaration body

```
type:
{K n : ℕ} → [NeZero n] → GCG24UserItemFairness.TypeWeightedRecommendationModel K n → ℝ

value:
fun {K n : ℕ} [NeZero n] (T : GCG24UserItemFairness.TypeWeightedRecommendationModel K n) =>
  sSup T.attainableItemFairnessSet
```

Direct retained dependencies

- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.TypeWeightedRecommendationModel.attainableItemFairnessSet

GCG24UserItemFairness.TypeWeightedRecommendationModel.feasibleAtLevel

Declaration location: paper-local

Lean declaration body

```
type:
{K n : ℕ} →
  [NeZero n] →
    GCG24UserItemFairness.TypeWeightedRecommendationModel K n → ℝ → GCG24UserItemFairness.TypePolicy K n → Prop

value:
fun {K n : ℕ} [NeZero n] (T : GCG24UserItemFairness.TypeWeightedRecommendationModel K n) (γ : ℝ)
    (ρ : GCG24UserItemFairness.TypePolicy K n) =>
  γ * T.optimalItemFairness ≤ T.itemFairness ρ
```

Direct retained dependencies

- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.TypeWeightedRecommendationModel.itemFairness
- GCG24UserItemFairness.TypeWeightedRecommendationModel.optimalItemFairness

GCG24UserItemFairness.TypeWeightedRecommendationModel.rawTypeUtility

Declaration location: paper-local

Lean declaration body

```
type:
{K n : ℕ} →
  GCG24UserItemFairness.TypeWeightedRecommendationModel K n →
    GCG24UserItemFairness.TypePolicy K n → GCG24UserItemFairness.UserType K → ℝ

value:
fun {K n : ℕ} (T : GCG24UserItemFairness.TypeWeightedRecommendationModel K n) (ρ : GCG24UserItemFairness.TypePolicy K n)
    (k : GCG24UserItemFairness.UserType K) =>
  AppliedModelingLib.Policy.agentScore ρ T.utility k
```

Direct retained dependencies

- AppliedModelingLib.Policy.agentScore
- GCG24UserItemFairness.Item
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.TypeWeightedRecommendationModel.normalizedTypeUtility

Declaration location: paper-local

Lean declaration body

```
type:
{K n : ℕ} →
  [NeZero n] →
    GCG24UserItemFairness.TypeWeightedRecommendationModel K n →
      GCG24UserItemFairness.TypePolicy K n → GCG24UserItemFairness.UserType K → ℝ

value:
fun {K n : ℕ} [NeZero n] (T : GCG24UserItemFairness.TypeWeightedRecommendationModel K n)
    (ρ : GCG24UserItemFairness.TypePolicy K n) (k : GCG24UserItemFairness.UserType K) =>
  T.rawTypeUtility ρ k / T.bestItemUtility k
```

Direct retained dependencies

- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.TypeWeightedRecommendationModel.bestItemUtility
- GCG24UserItemFairness.TypeWeightedRecommendationModel.rawTypeUtility
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.TypeWeightedRecommendationModel.typeFairness

Declaration location: paper-local

Lean declaration body

```
type:
{K n : ℕ} →
  [NeZero K] →
    [NeZero n] → GCG24UserItemFairness.TypeWeightedRecommendationModel K n → GCG24UserItemFairness.TypePolicy K n → ℝ

value:
fun {K n : ℕ} [NeZero K] [NeZero n] (T : GCG24UserItemFairness.TypeWeightedRecommendationModel K n)
    (ρ : GCG24UserItemFairness.TypePolicy K n) =>
  AppliedModelingLib.finiteMin (T.normalizedTypeUtility ρ)
```

Direct retained dependencies

- AppliedModelingLib.finiteMin
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.TypeWeightedRecommendationModel.normalizedTypeUtility
- GCG24UserItemFairness.UserType

GCG24UserItemFairness.TypeWeightedRecommendationModel.attainableTypeFairnessAtLevel

Declaration location: paper-local

Lean declaration body

```
type:
{K n : ℕ} → [NeZero K] → [NeZero n] → GCG24UserItemFairness.TypeWeightedRecommendationModel K n → ℝ → ℝ → Prop

value:
fun {K n : ℕ} [NeZero K] [NeZero n] (T : GCG24UserItemFairness.TypeWeightedRecommendationModel K n) (γ a : ℝ) =>
  {r : ℝ | ∃ (ρ : GCG24UserItemFairness.TypePolicy K n), T.feasibleAtLevel γ ρ ∧ r = T.typeFairness ρ} a
```

Direct retained dependencies

- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.TypeWeightedRecommendationModel.feasibleAtLevel
- GCG24UserItemFairness.TypeWeightedRecommendationModel.typeFairness

GCG24UserItemFairness.TypeWeightedRecommendationModel.optimalTypeFairnessAtLevel

Declaration location: paper-local

Lean declaration body

```
type:
{K n : ℕ} → [NeZero K] → [NeZero n] → GCG24UserItemFairness.TypeWeightedRecommendationModel K n → ℝ → ℝ

value:
fun {K n : ℕ} [NeZero K] [NeZero n] (T : GCG24UserItemFairness.TypeWeightedRecommendationModel K n) (γ : ℝ) =>
  sSup (T.attainableTypeFairnessAtLevel γ)
```

Direct retained dependencies

- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.TypeWeightedRecommendationModel.attainableTypeFairnessAtLevel

GCG24UserItemFairness.TypeWeightedRecommendationModel.IsOptimalAtLevel

Declaration location: paper-local

Lean declaration body

type:
```
{K n : ℕ} →
  [NeZero K] →
    [NeZero n] →
      GCG24UserItemFairness.TypeWeightedRecommendationModel K n → ℝ → GCG24UserItemFairness.TypePolicy K n → Prop
```

value:
```
fun {K n : ℕ} [NeZero K] [NeZero n] (T : GCG24UserItemFairness.TypeWeightedRecommendationModel K n) (γ : ℝ)
    (ρ : GCG24UserItemFairness.TypePolicy K n) =>
  T.feasibleAtLevel γ ρ ∧ T.typeFairness ρ = T.optimalTypeFairnessAtLevel γ
```

Direct retained dependencies

- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.TypeWeightedRecommendationModel.feasibleAtLevel
- GCG24UserItemFairness.TypeWeightedRecommendationModel.optimalTypeFairnessAtLevel
- GCG24UserItemFairness.TypeWeightedRecommendationModel.typeFairness

GCG24UserItemFairness.UserTypeAssignment.IsTypeSymmetric

Declaration location: paper-local

Lean declaration body

type:
```
{m n K : ℕ} → GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K → GCG24UserItemFairness.Policy m n → Prop
```

value:
```
fun {m n K : ℕ} (τ : GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K)
    (ρ : GCG24UserItemFairness.Policy m n) =>
  AppliedModelingLib.Policy.IsClasswise τ.toType ρ
```

Direct retained dependencies

- AppliedModelingLib.Policy.IsClasswise
- GCG24UserItemFairness.Item
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.User

GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Feasible

Declaration location: paper-local

Lean declaration body

type:
```
{m n K : ℕ} →
  GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP m n K → GCG24UserItemFairness.Policy m n → ℝ → Prop
```

value:
```
fun {m n K : ℕ} (L : GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP m n K)
    (ρ : GCG24UserItemFairness.Policy m n) (ell : ℝ) =>
  (∀ (u : GCG24UserItemFairness.User m), ∑ j : GCG24UserItemFairness.Item n, ((ρ u) j).toReal = 1) ∧
    (∀ (u : GCG24UserItemFairness.User m) (j : GCG24UserItemFairness.Item n), 0 ≤ ((ρ u) j).toReal) ∧
      GCG24UserItemFairness.UserTypeAssignment.IsTypeSymmetric L.types ρ ∧
        ∀ (j : GCG24UserItemFairness.Item n), L.model.propositionOneItemEqualityExpression j ρ ell = 0
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP
- GCG24UserItemFairness.RecommendationModel.propositionOneItemEqualityExpression
- GCG24UserItemFairness.User
- GCG24UserItemFairness.UserTypeAssignment.IsTypeSymmetric

GCG24UserItemFairness.UserTypeAssignment.TypeRepresentatives

Declaration location: paper-local

Lean declaration body

```
type:
{m K : ℕ} → GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K → Type

value:
fun {m K : ℕ} (τ : GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K) =>
  AppliedModelingLib.Policy.FiberRepresentatives τ.toType
```

Direct retained dependencies

- AppliedModelingLib.Policy.FiberRepresentatives
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.User

GCG24UserItemFairness.RecommendationModel.SymmetricData.canonicalReduction

Declaration location: paper-local

Lean declaration body

```
type:
{m n K : ℕ} →
  (S : GCG24UserItemFairness.RecommendationModel.SymmetricData m n K) →
    GCG24UserItemFairness.UserTypeAssignment.TypeRepresentatives S.types → GCG24UserItemFairness.ReductionWitness m n K

value:
fun {m n K : ℕ} (S : GCG24UserItemFairness.RecommendationModel.SymmetricData m n K)
    (reps : GCG24UserItemFairness.UserTypeAssignment.TypeRepresentatives S.types) =>
  { data := S,
    reduced :=
      {
        utility := fun (k : GCG24UserItemFairness.UserType K) (j : GCG24UserItemFairness.Item n) =>
          S.model.utility (reps.repr k) j,
        weight := S.types.typeWeight },
    utility_agrees := GCG24UserItemFairness.RecommendationModel.SymmetricData.canonicalReduction._proof_1 S reps,
    weight_eq_typeWeight := GCG24UserItemFairness.RecommendationModel.SymmetricData.canonicalReduction._proof_2 S reps }
```

Direct retained dependencies

- AppliedModelingLib.Policy.FiberRepresentatives
- GCG24UserItemFairness.Item
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.SymmetricData
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment.typeWeight
- GCG24UserItemFairness.ReductionWitness
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.User
- GCG24UserItemFairness.UserType
- GCG24UserItemFairness.UserTypeAssignment.TypeRepresentatives

GCG24UserItemFairness.RecommendationModel.SymmetricData.canonicalReductionOfSurjective

Declaration location: paper-local

Lean declaration body

```
type:
{m n K : ℕ} →
  (S : GCG24UserItemFairness.RecommendationModel.SymmetricData m n K) →
    Function.Surjective S.types.toType → GCG24UserItemFairness.ReductionWitness m n K

value:
fun {m n K : ℕ} (S : GCG24UserItemFairness.RecommendationModel.SymmetricData m n K)
    (hTypes : Function.Surjective S.types.toType) =>
  S.canonicalReduction (AppliedModelingLib.Policy.FiberRepresentatives.ofSurjective hTypes)
```

Direct retained dependencies

- AppliedModelingLib.Policy.FiberRepresentatives.ofSurjective
- GCG24UserItemFairness.RecommendationModel.SymmetricData
- GCG24UserItemFairness.RecommendationModel.SymmetricData.canonicalReduction
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.ReductionWitness
- GCG24UserItemFairness.User

GCG24UserItemFairness.UserTypeAssignment.TypeSymmetryLinearConstraintIndex

Declaration location: paper-local

Lean declaration body

```
type:
ℕ → ℕ → Type

value:
fun (m n : ℕ) => GCG24UserItemFairness.User m × GCG24UserItemFairness.User m × GCG24UserItemFairness.Item n
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.User

GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Constraint

Declaration location: paper-local

Lean declaration body

```
type:
ℕ → ℕ → ℕ → Type

value:
fun (m n K : ℕ) =>
  ((GCG24UserItemFairness.Item n ⊕ GCG24UserItemFairness.User m) ⊕
     GCG24UserItemFairness.UserTypeAssignment.TypeSymmetryLinearConstraintIndex m n) ⊕
    GCG24UserItemFairness.User m × GCG24UserItemFairness.Item n
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.User
- GCG24UserItemFairness.UserTypeAssignment.TypeSymmetryLinearConstraintIndex

GCG24UserItemFairness.UserTypeAssignment.descendTypePolicy

Declaration location: paper-local

Lean declaration body

```
type:
{m n K : ℕ} →
  (τ : GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K) →
    GCG24UserItemFairness.UserTypeAssignment.TypeRepresentatives τ →
      GCG24UserItemFairness.Policy m n → GCG24UserItemFairness.UserType K → PMF (GCG24UserItemFairness.Item n)

value:
fun {m n K : ℕ} (τ : GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K)
    (reps : GCG24UserItemFairness.UserTypeAssignment.TypeRepresentatives τ) (ρ : GCG24UserItemFairness.Policy m n)
    (a : GCG24UserItemFairness.UserType K) =>
  AppliedModelingLib.Policy.descendAlong reps ρ a
```

Direct retained dependencies

- AppliedModelingLib.Policy.descendAlong
- GCG24UserItemFairness.Item
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.User
- GCG24UserItemFairness.UserType
- GCG24UserItemFairness.UserTypeAssignment.TypeRepresentatives

GCG24UserItemFairness.UserTypeAssignment.typeSymmetryLinearCoefficient

Declaration location: paper-local

Lean declaration body

```
type:
{m n K : ℕ} →
  GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K →
    GCG24UserItemFairness.UserTypeAssignment.TypeSymmetryLinearConstraintIndex m n →
      GCG24UserItemFairness.User m → GCG24UserItemFairness.Item n → ℝ

value:
fun {m n K : ℕ} (τ : GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K)
    (c : GCG24UserItemFairness.UserTypeAssignment.TypeSymmetryLinearConstraintIndex m n)
    (u : GCG24UserItemFairness.User m) (j : GCG24UserItemFairness.Item n) =>
  if τ.toType c.1 = τ.toType c.2.1 then
    (if u = c.1 then if j = c.2.2 then 1 else 0 else 0) - if u = c.2.1 then if j = c.2.2 then 1 else 0 else 0
  else 0
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.User
- GCG24UserItemFairness.UserTypeAssignment.TypeSymmetryLinearConstraintIndex

GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.constraintNormal

Declaration location: paper-local

Lean declaration body

```
type:
{m n K : ℕ} →
  GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP m n K →
    GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Constraint m n K →
      GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Variable m n → ℝ

value:
fun {m n K : ℕ} (L : GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP m n K)
    (a : GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Constraint m n K)
    (a_1 : GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Variable m n) =>
  match a, a_1 with
  | Sum.inl (Sum.inl (Sum.inl j)), v => L.itemEqualityNormal j v
  | Sum.inl (Sum.inl (Sum.inr u)), Sum.inl (u', _j) => if u' = u then 1 else 0
  | Sum.inl (Sum.inl (Sum.inr _u)), Sum.inr val => 0
  | Sum.inl (Sum.inr c), Sum.inl (u, j) =>
    GCG24UserItemFairness.UserTypeAssignment.typeSymmetryLinearCoefficient L.types c u j
  | Sum.inl (Sum.inr _c), Sum.inr val => 0
  | Sum.inr (u, j), Sum.inl (u', j') => if u' = u ∧ j' = j then 1 else 0
  | Sum.inr (_u, _j), Sum.inr val => 0
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP
- GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Constraint
- GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Variable

- GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.itemEqualityNormal
- GCG24UserItemFairness.User
- GCG24UserItemFairness.UserTypeAssignment.TypeSymmetryLinearConstraintIndex
- GCG24UserItemFairness.UserTypeAssignment.typeSymmetryLinearCoefficient

GCG24UserItemFairness.UserTypeAssignment.typeSymmetryLinearExpression

Declaration location: paper-local

Lean declaration body

```
type:
{m n K : ℕ} →
  GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K →
    GCG24UserItemFairness.UserTypeAssignment.TypeSymmetryLinearConstraintIndex m n →
      GCG24UserItemFairness.Policy m n → ℝ

value:
fun {m n K : ℕ} (τ : GCG24UserItemFairness.RecommendationModel.UserTypeAssignment m K)
    (c : GCG24UserItemFairness.UserTypeAssignment.TypeSymmetryLinearConstraintIndex m n)
    (ρ : GCG24UserItemFairness.Policy m n) =>
  if τ.toType c.1 = τ.toType c.2.1 then ((ρ c.1) c.2.2).toReal - ((ρ c.2.1) c.2.2).toReal else 0
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.User
- GCG24UserItemFairness.UserTypeAssignment.TypeSymmetryLinearConstraintIndex

GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.ConstraintActive

Declaration location: paper-local

Lean declaration body

```
type:
{m n K : ℕ} →
  GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP m n K →
    GCG24UserItemFairness.Policy m n →
      ℝ → GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Constraint m n K → Prop

value:
fun {m n K : ℕ} (L : GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP m n K)
    (ρ : GCG24UserItemFairness.Policy m n) (ell : ℝ)
    (a : GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Constraint m n K) =>
  match a with
  | Sum.inl (Sum.inl (Sum.inl j)) => L.model.propositionOneItemEqualityExpression j ρ ell = 0
  | Sum.inl (Sum.inl (Sum.inr u)) => ∑ j : GCG24UserItemFairness.Item n, ((ρ u) j).toReal = 1
  | Sum.inl (Sum.inr c) => GCG24UserItemFairness.UserTypeAssignment.typeSymmetryLinearExpression L.types c ρ = 0
  | Sum.inr (u, j) => (ρ u) j = 0
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP
- GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Constraint
- GCG24UserItemFairness.RecommendationModel.propositionOneItemEqualityExpression
- GCG24UserItemFairness.User
- GCG24UserItemFairness.UserTypeAssignment.TypeSymmetryLinearConstraintIndex
- GCG24UserItemFairness.UserTypeAssignment.typeSymmetryLinearExpression

GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.ActiveBasis

Declaration location: paper-local

Lean declaration body

field feasible:
∀ {m n K : ℕ} {L : GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP m n K}
  {ρ : GCG24UserItemFairness.Policy m n} {ell : ℝ} (self : L.ActiveBasis ρ ell), L.Feasible ρ ell

field basis:
{m n K : ℕ} →
  {L : GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP m n K} →
    {ρ : GCG24UserItemFairness.Policy m n} →
      {ell : ℝ} →
        L.ActiveBasis ρ ell →
          Finset (GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Constraint m n K)

field basis_card:
∀ {m n K : ℕ} {L : GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP m n K}
  {ρ : GCG24UserItemFairness.Policy m n} {ell : ℝ} (self : L.ActiveBasis ρ ell), self.basis.card = m * n + 1

field basis_independent:
∀ {m n K : ℕ} {L : GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP m n K}
  {ρ : GCG24UserItemFairness.Policy m n} {ell : ℝ} (self : L.ActiveBasis ρ ell),
  LinearIndependent ℝ fun (c : ↥self.basis) => L.constraintNormal ↑c

field basis_active:
∀ {m n K : ℕ} {L : GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP m n K}
  {ρ : GCG24UserItemFairness.Policy m n} {ell : ℝ} (self : L.ActiveBasis ρ ell),
  ∀ c ∈ self.basis, L.ConstraintActive ρ ell c

Direct retained dependencies

- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP
- GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Constraint
- GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.ConstraintActive
- GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Feasible
- GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.Variable
- GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.constraintNormal

Paper-specific semantic prerequisites

GCG24UserItemFairness.RecommendationModel.rawUserUtility

Paper-local declaration:
GCG24UserItemFairness.RecommendationModel.rawUserUtility
Source locator: cited publication, lines 126-132

Verbatim paper-source connection

utility from
P
P
P
using the platform is j ρij wij ; item j's expected utility is i ρij wij , where j ρij = 1 for each
user i.
Fairness desiderata. We suppose that the platform uses as a benchmark, for each user or item,

Lean-expanded paper semantic target

type:
```
{m n : ℕ} →
  GCG24UserItemFairness.RecommendationModel m n → GCG24UserItemFairness.Policy m n → GCG24UserItemFairness.User m → ℝ
```

value:
```
fun {m n : ℕ} (W : GCG24UserItemFairness.RecommendationModel m n) (ρ : GCG24UserItemFairness.Policy m n)
    (u : GCG24UserItemFairness.User m) =>
  AppliedModelingLib.Policy.agentScore ρ W.utility u
```

Direct retained dependencies

- AppliedModelingLib.Policy.agentScore
- GCG24UserItemFairness.Item
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.User

Recorded source-to-declaration screening **Verdict:** matches
Reason: agentScore expands through pmfExp to sum_j rho_ij*w_ij, exactly matching the source raw user utility.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.RecommendationModel.normalizedUserUtility

Paper-local declaration:
GCG24UserItemFairness.RecommendationModel.normalizedUserUtility
Source locator: cited publication, lines 134-145

Verbatim paper-source connection

recommendation policy ρ, let Ui (ρ, w) be user i's utility from ρ, normalized by the utility maxj wij they would receive from being recommended their best match. Let Ij (ρ, w)Pbe item j's utility from ρ, normalized by the utility j receives if it is recommended to every user, i wij . The normalized utilities are:
P
P
j ρij wij
i ρij wij
Ui (ρ, w) =
, Ij (ρ, w) = P
.
maxj wij

Lean-expanded paper semantic target

type:
```
{m n : ℕ} →
  [NeZero n] →
    GCG24UserItemFairness.RecommendationModel m n → GCG24UserItemFairness.Policy m n → GCG24UserItemFairness.User m → ℝ
```

value:
```
fun {m n : ℕ} [NeZero n] (W : GCG24UserItemFairness.RecommendationModel m n) (ρ : GCG24UserItemFairness.Policy m n)
    (u : GCG24UserItemFairness.User m) =>
  W.rawUserUtility ρ u / W.bestItemUtility u
```

Direct retained dependencies

- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.bestItemUtility
- GCG24UserItemFairness.RecommendationModel.rawUserUtility
- GCG24UserItemFairness.User

Recorded source-to-declaration screening **Verdict:** matches

Reason: Expanding rawUserUtility and bestItemUtility gives sum_j rho_ij*w_ij divided by max_j w_ij, exactly the source U_i formula.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.RecommendationModel.userFairness

Paper-local declaration:
GCG24UserItemFairness.RecommendationModel.userFairness
Source locator: cited publication, lines 149-150

Verbatim paper-source connection

Given a recommendation policy ρ, user fairness is quantified as the minimum normalized user utility
Umin (ρ, w) = mini Ui (ρ, w), and item fairness analogously as Imin (ρ, w) = minj Ij (ρ, w).

Lean-expanded paper semantic target

type:
```
{m n : ℕ} →
  [NeZero m] → [NeZero n] → GCG24UserItemFairness.RecommendationModel m n → GCG24UserItemFairness.Policy m n → ℝ
```

value:
```
fun {m n : ℕ} [NeZero m] [NeZero n] (W : GCG24UserItemFairness.RecommendationModel m n)
    (ρ : GCG24UserItemFairness.Policy m n) =>
  AppliedModelingLib.finiteMin (W.normalizedUserUtility ρ)
```

Direct retained dependencies

- AppliedModelingLib.finiteMin
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.normalizedUserUtility
- GCG24UserItemFairness.User

Recorded source-to-declaration screening **Verdict:** matches

Reason: Lean takes the finite minimum of normalized user utility over users, matching U_min(rho,w) = min_i U_i(rho,w).
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.EstimatedRecommendationModel.priceOfMisestimation

Paper-local declaration:
GCG24UserItemFairness.EstimatedRecommendationModel.priceOfMisestimation
Source locator: cited publication, lines 199-220

Verbatim paper-source connection

Similarly, we investigate the price of misestimation. Let ŵij denote the platform’s estimate of the utility of recommending i to j; let ρ̂(γ) be a policy that solves the optimization problem above with 3

[form-feed in source extraction]
∗
misestimated utilities, that is, ρ̂(γ) attains Umin
(γ, ŵ).1 Then we define the price of misestimation
M
on user utility (“price of misestimation”) πU as
M
(γ, w, ŵ) :=
πU

∗
Umin
(γ, w) − Umin (ρ̂(γ), w)
,
∗ (γ, w)
Umin

that is, the decrease in true user fairness due to optimizing with estimated utilities. In particular, in Section 5 we will examine whether item fairness exacerbates the price of misestimation by comparing

Lean-expanded paper semantic target

```
type:
{m n : ℕ} →
  [NeZero m] →
    [NeZero n] → GCG24UserItemFairness.EstimatedRecommendationModel m n → ℝ → GCG24UserItemFairness.Policy m n → ℝ

value:
fun {m n : ℕ} [NeZero m] [NeZero n] (E : GCG24UserItemFairness.EstimatedRecommendationModel m n) (γ : ℝ)
    (ρhat : GCG24UserItemFairness.Policy m n) =>
  have base : ℝ := E.trueModel.optimalUserFairnessAtLevel γ;
  if h : base = 0 then 0 else (base - E.trueModel.userFairness ρhat) / base
```

Direct retained dependencies

- GCG24UserItemFairness.EstimatedRecommendationModel
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel.optimalUserFairnessAtLevel
- GCG24UserItemFairness.RecommendationModel.userFairness

Recorded source-to-declaration screening **Verdict:** matches

Reason: Lean computes the true constrained optimum minus the true fairness of the policy chosen under estimated utilities, divided by the true optimum. SolvesEstimatedProblem supplies the source requirement that the policy optimizes the estimated problem; the zero-base branch totalizes an undefined source case.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.ProofBridge.problem11EqualityFeasibleSet

Paper-local declaration:
GCG24UserItemFairness.ProofBridge.problem11EqualityFeasibleSet
Source locator: cited publication, lines 3375-3400

Verbatim paper-source connection

S ′ = {x, y, z ∈ Ssymm : xj = yn−j+1 , zj = zn−j+1 for all j}.
Since y is completely determined by x, we can identify S ′ with Δ2n−1 and can identify x, y, z ∈ S ′
simply by x, z.
∗
Lemma 12. If a policy ρ = (x, y, z) ∈ Ssymm solves Umin (ρ) = Umin
(1), then there is some policy
∗
ρ′ = x′ , z ′ ∈ S ′ that solves Umin (ρ′) = Umin
(1).

Applying the framework in Proposition 1 to this proof with S = S ′ and using ŵ as the user and item
values, we reduce the problem to finding
ϕ = arg

max

x,z∈Δ2n−1 ,λ

λ
(11)

subject to lj (x, z) = λ

∀j,

and showing that this ϕ is unique.

Lean-expanded paper semantic target

type:
{n : ℕ} → ℝ → (GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.ProofBridge.Problem11Point n → Prop

value:
```
fun {n : ℕ} (beta : ℝ) (v : GCG24UserItemFairness.Item n → ℝ)
    (a : GCG24UserItemFairness.ProofBridge.Problem11Point n) =>
  {p : GCG24UserItemFairness.ProofBridge.Problem11Point n |
     (∀ (j : GCG24UserItemFairness.Item n), 0 ≤ p.1.1 j) ∧
       (∀ (j : GCG24UserItemFairness.Item n), 0 ≤ p.1.2 j) ∧
         ∑ j : GCG24UserItemFairness.Item n, p.1.1 j = 1 ∧
           ∑ j : GCG24UserItemFairness.Item n, p.1.2 j = 1 ∧
             (∀ (j : GCG24UserItemFairness.Item n),
                p.1.2 (GCG24UserItemFairness.OpposingTypes.reverseItem j) = p.1.2 j) ∧
               ∀ (j : GCG24UserItemFairness.Item n),
                 GCG24UserItemFairness.OpposingTypes.theorem4Problem11ItemValue beta v p.1.1 p.1.2 j = p.2}
  a
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.reverseItem
- GCG24UserItemFairness.OpposingTypes.theorem4Problem11ItemValue
- GCG24UserItemFairness.ProofBridge.Problem11Point

Recorded source-to-declaration screening **Verdict:** matches

Reason: Lean gives x and z nonnegative simplex rows, imposes z_j = z_reverse(j), and equates every corrected Problem 11 item value to lambda. This matches the source after eliminating y via y_j = x_reverse(j).
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.ProofBridge.problem11BasicFeasible

Paper-local declaration:
GCG24UserItemFairness.ProofBridge.problem11BasicFeasible
Source locator: cited publication, lines 3401-3412

Verbatim paper-source connection

First, we use an analogue of the sparsity result Proposition 2 to show that the optimal x, z has the
following sparse structure.
Lemma 13. Let x, z be an optimal basic feasible solution to the linear program Problem 11. There
′
′
′
′
is some j ≤ n+1
2 such that for all j > j, xj = 0 and for all j < j, zj = zn−j +1 = 0. Let the
pivot index t denote the minimum such j.
This sparse structure implies uniqueness.

Lean-expanded paper semantic target

type:
{n : ℕ} → ℝ → (GCG24UserItemFairness.Item n → ℝ) → GCG24UserItemFairness.TypePolicy 3 n → ℝ → Prop

value:
fun {n : ℕ} (beta : ℝ) (v : GCG24UserItemFairness.Item n → ℝ) (ρ : GCG24UserItemFairness.TypePolicy 3 n) (ell : ℝ) =>
  GCG24UserItemFairness.ProofBridge.problem11PointOfPolicy ρ ell ∈
    Set.extremePoints ℝ (GCG24UserItemFairness.ProofBridge.problem11EqualityFeasibleSet beta v)

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.ProofBridge.Problem11Point
- GCG24UserItemFairness.ProofBridge.problem11EqualityFeasibleSet
- GCG24UserItemFairness.ProofBridge.problem11PointOfPolicy
- GCG24UserItemFairness.TypePolicy

Recorded source-to-declaration screening **Verdict:** matches

Reason: Lean defines a basic feasible solution as an extreme point of the finite Problem 11 equality-feasible polytope. The pinned Lemma 13 use separately assumes optimality, matching the source phrase ’optimal basic feasible solution.’
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.ProofBridge.recommendationUtility

Paper-local declaration:
GCG24UserItemFairness.ProofBridge.recommendationUtility
Source locator: cited publication, lines 116-120

Verbatim paper-source connection

Users, items, and utilities. There is a finite population of m users and n items. Let wij > 0 be
the utility of recommending item j to user i; this could represent a click-through rate or purchase
probability. We suppose that user-item utilities are symmetric: each of the user i and item j receives
utility wij from being recommended item j.
Let Δn−1 denote the simplex in Rn . The platform's task is to choose a recommendation policy

Lean-expanded paper semantic target

type:
```
{m n : ℕ} →
  GCG24UserItemFairness.RecommendationModel m n → GCG24UserItemFairness.User m → GCG24UserItemFairness.Item n → ℝ
```

value:
```
fun {m n : ℕ} (W : GCG24UserItemFairness.RecommendationModel m n) (u : GCG24UserItemFairness.User m)
    (j : GCG24UserItemFairness.Item n) =>
  W.utility u j
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.User

Recorded source-to-declaration screening **Verdict:** matches

Reason: The source utility matrix entry is the recommendation value assigned to a user-item pair, which is the paper-local recommendationUtility definition.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.RecommendationModel.IsOptimalAtLevel

Paper-local declaration:
GCG24UserItemFairness.RecommendationModel.IsOptimalAtLevel
Source locator: cited publication, lines 152-174

Verbatim paper-source connection

fairness desiderata as follows. At the extremes, the platform could choose a maximally fair solution
∗
for one side, ignoring the other. Denote the optimal user fair utility as Umin
(w) := maxρ Umin (ρ, w)
(achieved by giving each user their favorite item deterministically), and the optimal item fair utility
∗
as Imin
(w) := maxρ Imin (ρ, w).
Finally, we cast the two-sided fair optimization – for the optimal γ-constrained user fair solution – as
∗
Umin
(γ, w) = max Umin (ρ, w)
ρ

(1)

∗
subject to Imin (ρ, w) ≥ γImin
(w),

i.e., we maximize the minimum normalized user utility, subject to the minimum normalized item
utility being at least a fraction γ of the optimal item fair solution.
Research questions: price of fairness and misestimation. We can now define the price of item

Lean-expanded paper semantic target

type:
{m n : ℕ} →
[NeZero m] → [NeZero n] → GCG24UserItemFairness.RecommendationModel m n → ℝ → GCG24UserItemFairness.Policy m n → Prop

value:
fun {m n : ℕ} [NeZero m] [NeZero n] (W : GCG24UserItemFairness.RecommendationModel m n) (γ : ℝ)
(ρ : GCG24UserItemFairness.Policy m n) =>
W.feasibleAtLevel γ ρ ∧ W.userFairness ρ = W.optimalUserFairnessAtLevel γ

Direct retained dependencies

- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.feasibleAtLevel
- GCG24UserItemFairness.RecommendationModel.optimalUserFairnessAtLevel
- GCG24UserItemFairness.RecommendationModel.userFairness

Recorded source-to-declaration screening **Verdict:** matches

Reason: Lean requires itemFairness(rho) at least gamma times optimalItemFairness and userFairness(rho) to equal the supremum over that feasible set, exactly expressing source Problem 1.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.RecommendationModel.itemNormalizer

Paper-local declaration:
GCG24UserItemFairness.RecommendationModel.itemNormalizer
Source locator: cited publication, lines 134-145

Verbatim paper-source connection

recommendation policy ρ, let Ui (ρ, w) be user i's utility from ρ, normalized by the utility maxj wij they would receive from being recommended their best match. Let Ij (ρ, w)Pbe item j's utility from ρ, normalized by the utility j receives if it is recommended to every user, i wij . The normalized utilities are:
P
P
j ρij wij
i ρij wij
Ui (ρ, w) =
, Ij (ρ, w) = P
.
maxj wij

Lean-expanded paper semantic target

type:
{m n : ℕ} → GCG24UserItemFairness.RecommendationModel m n → GCG24UserItemFairness.Item n → ℝ

value:
fun {m n : ℕ} (W : GCG24UserItemFairness.RecommendationModel m n) (j : GCG24UserItemFairness.Item n) =>
  ∑ u : GCG24UserItemFairness.User m, W.utility u j

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.User

Recorded source-to-declaration screening **Verdict:** matches

Reason: Lean sums w_ij over users for fixed item j, exactly matching the source denominator for normalized item utility.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.RecommendationModel.rawItemUtility

Paper-local declaration:
GCG24UserItemFairness.RecommendationModel.rawItemUtility
Source locator: cited publication, lines 126-132

Verbatim paper-source connection

utility from
P
P
P
using the platform is j ρij wij ; item j's expected utility is i ρij wij , where j ρij = 1 for each
user i.
Fairness desiderata. We suppose that the platform uses as a benchmark, for each user or item,

Lean-expanded paper semantic target

type:
```
{m n : ℕ} →
  GCG24UserItemFairness.RecommendationModel m n → GCG24UserItemFairness.Policy m n → GCG24UserItemFairness.Item n → ℝ
```

value:
```
fun {m n : ℕ} (W : GCG24UserItemFairness.RecommendationModel m n) (ρ : GCG24UserItemFairness.Policy m n)
    (j : GCG24UserItemFairness.Item n) =>
  ∑ u : GCG24UserItemFairness.User m, W.utility u j * ((ρ u) j).toReal
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.User

Recorded source-to-declaration screening **Verdict:** matches

Reason: Lean sums w_ij*rho_ij over users for fixed item j, exactly matching the source raw item utility.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.RecommendationModel.normalizedItemUtility

Paper-local declaration:
GCG24UserItemFairness.RecommendationModel.normalizedItemUtility
Source locator: cited publication, lines 134-145

Verbatim paper-source connection

recommendation policy ρ, let Ui (ρ, w) be user i's utility from ρ, normalized by the utility maxj wij they would receive from being recommended their best match. Let Ij (ρ, w)Pbe item j's utility from ρ, normalized by the utility j receives if it is recommended to every user, i wij . The normalized utilities are:
P
P
j ρij wij
i ρij wij
Ui (ρ, w) =
, Ij (ρ, w) = P
.
maxj wij

Lean-expanded paper semantic target

type:
```
{m n : ℕ} →
  GCG24UserItemFairness.RecommendationModel m n → GCG24UserItemFairness.Policy m n → GCG24UserItemFairness.Item n → ℝ
```

value:
```
fun {m n : ℕ} (W : GCG24UserItemFairness.RecommendationModel m n) (ρ : GCG24UserItemFairness.Policy m n)
    (j : GCG24UserItemFairness.Item n) =>
  have denom : ℝ := W.itemNormalizer j;
  if h : denom = 0 then 0 else W.rawItemUtility ρ j / denom
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.itemNormalizer
- GCG24UserItemFairness.RecommendationModel.rawItemUtility

Recorded source-to-declaration screening **Verdict:** matches

Reason: Lean divides sum_i w_ij*rho_ij by sum_i w_ij, matching the source I_j formula. Its explicit zero-denominator value totalizes an undefined source case outside the positive-denominator domain.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.RecommendationModel.itemFairness

Paper-local declaration:
GCG24UserItemFairness.RecommendationModel.itemFairness
Source locator: cited publication, lines 149-150

Verbatim paper-source connection

Given a recommendation policy ρ, user fairness is quantified as the minimum normalized user utility
Umin (ρ, w) = mini Ui (ρ, w), and item fairness analogously as Imin (ρ, w) = minj Ij (ρ, w).

Lean-expanded paper semantic target

type:
{m n : ℕ} → [NeZero n] → GCG24UserItemFairness.RecommendationModel m n → GCG24UserItemFairness.Policy m n → ℝ

value:
fun {m n : ℕ} [NeZero n] (W : GCG24UserItemFairness.RecommendationModel m n) (ρ : GCG24UserItemFairness.Policy m n) =>
  AppliedModelingLib.finiteMin (W.normalizedItemUtility ρ)

Direct retained dependencies

- AppliedModelingLib.finiteMin
- GCG24UserItemFairness.Item
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.normalizedItemUtility

Recorded source-to-declaration screening **Verdict:** matches
Reason: Lean takes the finite minimum of normalized item utility over all items, matching I_min(rho,w) = min_j I_j(rho,w).
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.RecommendationModel.priceOfFairness

Paper-local declaration:
GCG24UserItemFairness.RecommendationModel.priceOfFairness
Source locator: cited publication, lines 174-199

Verbatim paper-source connection

Research questions: price of fairness and misestimation. We can now define the price of item
F
fairness on user fairness πU
|I (“price of fairness”) as the decrease in user fairness with maximal item
fairness constraints:
∗
∗
Umin
(w) − Umin
(γ = 1, w)
F
πU
.
|I (w) :=
∗ (w)
Umin
F
We ask how πU
|I changes with the utility matrix w. We note that while this question (in terms
of solutions to Problem (1)) can be simply stated, as we detail in Section 3, finding a closed
∗
form expression for Umin
(1, w) – the optimal minimum normalized user utility given item fairness
constraints – in terms of w is theoretically challenging.

Similarly, we investigate the price of misestimation. Let ŵij denote the platform’s estimate of the

Lean-expanded paper semantic target

type:
{m n : ℕ} → [NeZero m] → [NeZero n] → GCG24UserItemFairness.RecommendationModel m n → ℝ

value:
fun {m n : ℕ} [NeZero m] [NeZero n] (W : GCG24UserItemFairness.RecommendationModel m n) => W.priceOfFairnessAt 1

Direct retained dependencies

- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.priceOfFairnessAt

Recorded source-to-declaration screening **Verdict:** matches

Reason: Expanding priceOfFairnessAt at gamma = 1 gives the source decrease from unconstrained optimal user fairness to optimal user fairness under maximal item fairness, normalized by the unconstrained optimum. The zero-base branch totalizes an undefined source case.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.assumption_positive_recommendation_utilities

Paper-local declaration:
GCG24UserItemFairness.assumption_positive_recommendation_utilities
Source locator: cited publication, lines 116-120

Verbatim paper-source connection

Users, items, and utilities. There is a finite population of m users and n items. Let wij > 0 be
the utility of recommending item j to user i; this could represent a click-through rate or purchase
probability. We suppose that user-item utilities are symmetric: each of the user i and item j receives
utility wij from being recommended item j.
Let Δn−1 denote the simplex in Rn . The platform's task is to choose a recommendation policy

[Next verbatim source excerpt]

can be captured by a single parameter. Consider the following utility matrix structure. Let v1 >
v2 > ... > vn > 0. Suppose that there are two user types: a user i either has utility wij = vj or
wij = vn−j+1 , and say that user i is of type 1 or 2 respectively. In words, the two types have opposite
preferences, but the preferences can otherwise be generic and be for any number of items. Now, the
direct solution by symmetry for the example no longer directly holds – the items in the middle, not

Lean-expanded paper semantic target

type:
{m n : ℕ} → GCG24UserItemFairness.RecommendationModel m n → Prop

value:
fun {m n : ℕ} (W : GCG24UserItemFairness.RecommendationModel m n) => W.Positive

Direct retained dependencies

- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.Positive

Recorded source-to-declaration screening **Verdict:** matches

Reason: The source works under strictly positive recommendation utilities; the paper-local premise states the same positivity condition over every user-item pair.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.assumption_proposition2_utility_rows_determine_types

Paper-local declaration:
GCG24UserItemFairness.assumption_proposition2_utility_rows_determine_types
Source locator: cited publication, lines 312-316

Verbatim paper-source connection

symmetric structure. Suppose users come in K types, where a user of type k shares a utility vector.
Then, consider Ssymm = {ρ : ρi = ρi′ if wi = wi′ } ⊆ Δm
n−1 , the set of policies where users of the
same type are given the same recommendation probabilities. Note that, in the extreme case, each user
is their own type. Furthermore, let ρk denote the recommendation policy for type k, for ρ ∈ Ssymm .

[Next verbatim source excerpt]

We first define a piece of convenient notation: if two users i, i′ share the same utility vector wi = wi′ ,
we say that they have the same type τ (i) = τ (i′). If τ (i) = τ (i′) := τ , we write wij = wi′ j := wτ j

Lean-expanded paper semantic target

type:
{m n K : ℕ} → GCG24UserItemFairness.RecommendationModel.SymmetricData m n K → Prop

value:
fun {m n K : ℕ} (S : GCG24UserItemFairness.RecommendationModel.SymmetricData m n K) =>
  GCG24UserItemFairness.RecommendationModel.UserTypeAssignment.UtilityRowsDetermineTypes S.model S.types

Direct retained dependencies

- GCG24UserItemFairness.RecommendationModel.SymmetricData
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment.UtilityRowsDetermineTypes

Recorded source-to-declaration screening **Verdict:** matches

Reason: The source defines common types by equality of full utility rows. The premise preserves that equality-defined convention; deterministic tie handling only makes the type assignment total and does not add an economic restriction.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.assumption_theorem3_first_half_alpha_domain

Paper-local declaration:
GCG24UserItemFairness.assumption_theorem3_first_half_alpha_domain
Source locator: cited publication, lines 378-398

Verbatim paper-source connection

fairness as,
∗
∗
Umin
(α) − Umin
(γ = 1, α)
F
πU
.
|I (α) :=
∗
Umin (α)
F
Given this structure, Theorem 3 states that the price of fairness πU
|I (α) increases in the homogeneity
of the users – heterogeneous user populations are less affected by incorporating item fairness
constraints.
F
Theorem 3. πU
|I (α) is decreasing in α for 0 < α ≤ 1/2, and increasing in α for 1/2 ≤ α < 1.
Proof sketch for Theorem 3. We show that when in a population with two opposing types as described

Lean-expanded paper semantic target

type:
ℝ → ℝ → Prop

value:
fun (alpha alpha' : ℝ) =>
  0 < alpha ∧ alpha < 1 ∧ 0 < alpha' ∧ alpha' < 1 ∧ alpha ≤ alpha' ∧ alpha ≤ 1 / 2 ∧ alpha' ≤ 1 / 2

Recorded source-to-declaration screening **Verdict:** matches

Reason: The source's first Theorem 3 clause is explicitly on the alpha domain from zero through one half, which this premise states.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.assumption_theorem3_opposing_type_model

Paper-local declaration:
GCG24UserItemFairness.assumption_theorem3_opposing_type_model
Source locator: cited publication, lines 368-398

Verbatim paper-source connection

can be captured by a single parameter. Consider the following utility matrix structure. Let v1 >
v2 > ... > vn > 0. Suppose that there are two user types: a user i either has utility wij = vj or
wij = vn−j+1 , and say that user i is of type 1 or 2 respectively. In words, the two types have opposite
preferences, but the preferences can otherwise be generic and be for any number of items. Now, the
direct solution by symmetry for the example no longer directly holds – the items in the middle, not
necessarily preferred by either user type, may be binding in terms of item fairness constraints.
Let α be the proportion of type 1 users in the population, out of a fixed population of m users.
Parameter α thus controls the population heterogeneity; if α is near 0 or 1, the population is highly
homogeneous, dominated by users of the same type. If α = 21 , the population is split evenly between
the two types and is highly heterogeneous. Since we parametrize w by α, we may write the price of
fairness as,
∗
∗
Umin
(α) − Umin
(γ = 1, α)
F
πU
.
|I (α) :=
∗
Umin (α)
F
Given this structure, Theorem 3 states that the price of fairness πU
|I (α) increases in the homogeneity
of the users – heterogeneous user populations are less affected by incorporating item fairness
constraints.
F
Theorem 3. πU
|I (α) is decreasing in α for 0 < α ≤ 1/2, and increasing in α for 1/2 ≤ α < 1.
Proof sketch for Theorem 3. We show that when in a population with two opposing types as described

[Next verbatim source excerpt]

Recall Theorem 3 and its setting. We let v1 > v2 > ... > vn and suppose that there are two user
types. A user i either has value wij = vj for all j or wij = vn−j+1 for all j, corresponding to types
1 and 2 respectively. That is, the two types have opposite preferences. We let α be the proportion of
type 1 users in the population, out of a fixed population of m users.
F
Theorem 3. πU

Lean-expanded paper semantic target

```
type:
{m n : ℕ} →
  (S S' : GCG24UserItemFairness.RecommendationModel.SymmetricData m n 2) →
    Function.Surjective S.types.toType →
      Function.Surjective S'.types.toType → ℝ → ℝ → (GCG24UserItemFairness.Item n → ℝ) → Prop

value:
fun {m n : ℕ} (S S' : GCG24UserItemFairness.RecommendationModel.SymmetricData m n 2)
    (hTypes : Function.Surjective S.types.toType) (hTypes' : Function.Surjective S'.types.toType) (alpha alpha' : ℝ)
    (v : GCG24UserItemFairness.Item n → ℝ) =>
  (S.canonicalReductionOfSurjective hTypes).reduced = GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha v ∧
    (S'.canonicalReductionOfSurjective hTypes').reduced =
        GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha' v ∧
      (∀ (j : GCG24UserItemFairness.Item n), 0 < v j) ∧ GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v
```

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex
- GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel
- GCG24UserItemFairness.RecommendationModel.SymmetricData
- GCG24UserItemFairness.RecommendationModel.SymmetricData.canonicalReductionOfSurjective
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.ReductionWitness
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.User

Recorded source-to-declaration screening **Verdict:** matches

Reason: The canonical reductions use typeCard/m as population shares, so equality with weights alpha and 1-alpha encodes the source proportions. The two utility rows are v and its reversal, and Lean requires the source’s positivity and strict decrease of v.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

GCG24UserItemFairness.assumption_theorem3_second_half_alpha_domain

Paper-local declaration:
GCG24UserItemFairness.assumption_theorem3_second_half_alpha_domain
Source locator: cited publication, lines 378-398

Verbatim paper-source connection

fairness as,
∗
∗
Umin
(α) − Umin
(γ = 1, α)
F
πU
.
|I (α) :=
∗
Umin (α)
F
Given this structure, Theorem 3 states that the price of fairness πU
|I (α) increases in the homogeneity
of the users – heterogeneous user populations are less affected by incorporating item fairness
constraints.
F
Theorem 3. πU
|I (α) is decreasing in α for 0 < α ≤ 1/2, and increasing in α for 1/2 ≤ α < 1.
Proof sketch for Theorem 3. We show that when in a population with two opposing types as described

Lean-expanded paper semantic target

type:
ℝ → ℝ → Prop

value:
fun (alpha alpha' : ℝ) =>
  0 < alpha ∧ alpha < 1 ∧ 0 < alpha' ∧ alpha' < 1 ∧ alpha ≤ alpha' ∧ 1 / 2 ≤ alpha ∧ 1 / 2 ≤ alpha'

Recorded source-to-declaration screening

Verdict: matches
Reason: The source's second Theorem 3 clause is explicitly on the alpha domain from one half to one, which this premise states.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.assumption_theorem4_at_least_three_items

Paper-local declaration:
GCG24UserItemFairness.assumption_theorem4_at_least_three_items
Source locator: cited publication, lines 456-470

Verbatim paper-source connection

Theorem 4. If β > n1 and w and ŵ are as described above, then fairness constraints can arbitrarily worsen the price of misestimation.
• The price of misestimation without fairness constraints is low: for all {vj }, there is a
∗
recommendation policy ρ̂ that solves the misestimated problem Umin
(0, ŵ) so that
M
πU
(0, w, ŵ) ≤

1
.
2

• The price of misestimation with fairness constraints can be arbitrarily large: ∀ϵ, there exists

[Next verbatim source excerpt]

Lemma 14. Problem 11 has a unique optimal solution.
We can now describe what this solution (x, z) looks like.
Lemma 15. Let (x, z) be the optimal solution to Problem 12, and let t be the pivot element of (x, z); suppose that ɾ= n+1
2 . Define
X 1
Lt :=
.
q
j<t j
Then
λ=
[U+F8F1 delimiter glyph]

Lean-expanded paper semantic target

type:
ℕ → Prop

value:
fun (n : ℕ) => 2 < n

Recorded source-to-declaration screening **Verdict:** matches

Reason: The Theorem 4 construction uses the displayed multi-item regime; this premise records its source-required at-least-three-items domain.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.assumption_theorem4_cold_start_type_wiring

Paper-local declaration:
GCG24UserItemFairness.assumption_theorem4_cold_start_type_wiring
Source locator: cited publication, lines 444-455

Verbatim paper-source connection

Section 4, suppose that there are 2 types of users, with opposing preferences (i.e., with values
v1 , v2 ..., vn and vn , vn−1 , ..., v1 , respectively) – and the platform has correctly estimated these
preferences. However, now, these two types only make up a proportion β of the population each.
Now, we suppose that there is a fraction 1 − β of the user population who are “new” users. We assume
that these users are drawn from the same distribution as the remaining users, but the platform does
not know their preferences. It thus constructs a prior by averaging over the known users’ preferences
v +v
– (mis)estimating the users’ utility for each item j as j n−j+1
.
2
6

[Next verbatim source excerpt]

The idea follows the above intuition: without item fairness constraints and assuming cold start users
follow the same distribution as existing users, the platform’s price of misestimation is low because
the platform treats cold start users as the average of the existing population. Fairness constraints,
however, can make this cost arbitrarily high, as in expectation cold start users relatively enjoy items
other users do not. Such effects suggest that a more careful treatment of uncertainty and fairness

Lean-expanded paper semantic target

type:
```
{m n : ℕ} →
  GCG24UserItemFairness.RecommendationModel.SymmetricData m n 2 →
    GCG24UserItemFairness.RecommendationModel.SymmetricData m n 3 → GCG24UserItemFairness.User m → Prop
```

value:
```
fun {m n : ℕ} (Strue : GCG24UserItemFairness.RecommendationModel.SymmetricData m n 2)
    (Sest : GCG24UserItemFairness.RecommendationModel.SymmetricData m n 3) (u : GCG24UserItemFairness.User m) =>
  (∀ (u : GCG24UserItemFairness.User m), Sest.types.toType u = 0 → Strue.types.toType u = 0) ∧
    (∀ (u : GCG24UserItemFairness.User m), Sest.types.toType u = 1 → Strue.types.toType u = 1) ∧
      (Strue.types.toType u = 0 ∨ Strue.types.toType u = 1) ∧ Sest.types.toType u = 2
```

Direct retained dependencies

- GCG24UserItemFairness.RecommendationModel.SymmetricData
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.User

Recorded source-to-declaration screening **Verdict:** matches

Reason: Lean makes estimated known types 0 and 1 retain their corresponding true types and makes the selected cold-start user estimated type 2 with true opposing type 0 or 1, matching the source classification.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.assumption_theorem4_displayed_reduced_models

Paper-local declaration:
GCG24UserItemFairness.assumption_theorem4_displayed_reduced_models
Source locator: cited publication, lines 444-456

Verbatim paper-source connection

Section 4, suppose that there are 2 types of users, with opposing preferences (i.e., with values
v1 , v2 ..., vn and vn , vn−1 , ..., v1 , respectively) – and the platform has correctly estimated these
preferences. However, now, these two types only make up a proportion β of the population each.
Now, we suppose that there is a fraction 1 − β of the user population who are “new” users. We assume
that these users are drawn from the same distribution as the remaining users, but the platform does
not know their preferences. It thus constructs a prior by averaging over the known users’ preferences
v +v
– (mis)estimating the users’ utility for each item j as j n−j+1
.
2
6

[form-feed in source extraction]
Theorem 4. If β > n1 and w and ŵ are as described above, then fairness constraints can arbitrarily

[Next verbatim source excerpt]

Item fairness constraints with misestimation. First, notice that in the misestimated value matrix
ŵ there are three distinct user value types, and thus we can identify Ssymm with Δ3n−1 and a policy
ρ ∈ Ssymm with vectors x, y, z, where x is the recommendation policy for users with value ŵij = vj ,
y is the recommendation policy for users with value ŵij = vn−j+1 , and z is the recommendation
policy for the mis-estimated users.
Let

S ′ = {x, y, z ∈ Ssymm : xj = yn−j+1 , zj = zn−j+1 for all j}.
Since y is completely determined by x, we can identify S ′ with Δ2n−1 and can identify x, y, z ∈ S ′
simply by x, z.
∗
Lemma 12. If a policy ρ = (x, y, z) ∈ Ssymm solves Umin (ρ) = Umin
(1), then there is some policy
∗
ρ′ = x′ , z ′ ∈ S ′ that solves Umin (ρ′) = Umin
(1).

Applying the framework in Proposition 1 to this proof with S = S ′ and using ŵ as the user and item
values, we reduce the problem to finding
ϕ = arg

max

x,z∈Δ2n−1 ,λ

λ
(11)

subject to Ij (x, z) = λ

∀j,

and showing that this ϕ is unique.
First, we use an analogue of the sparsity result Proposition 2 to show that the optimal x, z has the
following sparse structure.
Lemma 13. Let x, z be an optimal basic feasible solution to the linear program Problem 11. There

[Next verbatim source excerpt]

In terms of x, z,
v +vn−j+1
zj
2
vj +vn−j+1
βvj + βvn−j+1 + (1 − 2β)
2
v +v
βvj xj + βvn−j+1 xn−j+1 + (1 − 2β) j n−j+1
zj
2
=
vj +vn−j+1
2

Ij (x, z) =

βvj xj + βvn−j+1 xn−j+1 + (1 − 2β) j

[U+0012 control character]
= 2β

vj
vn−j+1
xj +
xn−j+1
vj + vn−j+1
vj + vn−j+1

41

[U+0013 control character]
+ (1 − 2β)zj

[form-feed in source extraction]
v

j
. Note that qj does not depend on β, and qj is decreasing in j. Then
Let qj = vj +vn−j+1

Ij (x, y, z) = 2β (qj xj + (1 − qj)yj) + (1 − 2β)zj .

Settled source reading For Theorem 4, the normalized three-type construction uses masses beta, beta, and 1 - 2 beta; the stated beta range keeps all three masses positive.

Lean-expanded paper semantic target

```
type:
{m n : ℕ} →
  (Strue : GCG24UserItemFairness.RecommendationModel.SymmetricData m n 2) →
    (Sest : GCG24UserItemFairness.RecommendationModel.SymmetricData m n 3) →
      Function.Surjective Strue.types.toType → Function.Surjective Sest.types.toType → ℝ → ℝ → Prop

value:
fun {m n : ℕ} (Strue : GCG24UserItemFairness.RecommendationModel.SymmetricData m n 2)
    (Sest : GCG24UserItemFairness.RecommendationModel.SymmetricData m n 3)
    (hTypesTrue : Function.Surjective Strue.types.toType) (hTypesEst : Function.Surjective Sest.types.toType)
    (beta eps : ℝ) =>
  (Strue.canonicalReductionOfSurjective hTypesTrue).reduced =
      GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel (1 / 2)
        (GCG24UserItemFairness.OpposingTypes.theorem4SmallValueVector eps) ∧
    (Sest.canonicalReductionOfSurjective hTypesEst).reduced =
      GCG24UserItemFairness.OpposingTypes.theorem4EstimatedReducedModel beta
        (GCG24UserItemFairness.OpposingTypes.theorem4SmallValueVector eps)
```

Direct retained dependencies

- GCG24UserItemFairness.OpposingTypes.theorem4EstimatedReducedModel
- GCG24UserItemFairness.OpposingTypes.theorem4SmallValueVector
- GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel
- GCG24UserItemFairness.RecommendationModel.SymmetricData
- GCG24UserItemFairness.RecommendationModel.SymmetricData.canonicalReductionOfSurjective
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.ReductionWitness
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.User

Recorded source-to-declaration screening **Verdict:** matches
Reason: Using normalized typeCard/m weights, Lean’s true reduction has opposing rows with masses 1/2 and 1/2. Its estimated reduction has known opposing rows with masses beta and beta and the averaged cold-start row with mass 1-2*beta, matching approved context GCG24-T4-POPULATION-MASS-01.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.assumption_theorem4_estimated_model_reduction

Paper-local declaration:
GCG24UserItemFairness.assumption_theorem4_estimated_model_reduction
Source locator: cited publication, lines 444-455

Verbatim paper-source connection

Section 4, suppose that there are 2 types of users, with opposing preferences (i.e., with values
v1 , v2 ..., vn and vn , vn−1 , ..., v1 , respectively) – and the platform has correctly estimated these
preferences. However, now, these two types only make up a proportion β of the population each.
Now, we suppose that there is a fraction 1 − β of the user population who are “new” users. We assume
that these users are drawn from the same distribution as the remaining users, but the platform does
not know their preferences. It thus constructs a prior by averaging over the known users’ preferences
v +v
– (mis)estimating the users’ utility for each item j as j n−j+1
.
2
6

[Next verbatim source excerpt]

Item fairness constraints with misestimation. First, notice that in the misestimated value matrix
ŵ there are three distinct user value types, and thus we can identify Ssymm with Δ3n−1 and a policy
ρ ∈ Ssymm with vectors x, y, z, where x is the recommendation policy for users with value ŵij = vj ,
y is the recommendation policy for users with value ŵij = vn−j+1 , and z is the recommendation
policy for the mis-estimated users.
Let

S ′ = {x, y, z ∈ Ssymm : xj = yn−j+1 , zj = zn−j+1 for all j}.
Since y is completely determined by x, we can identify S ′ with Δ2n−1 and can identify x, y, z ∈ S ′
simply by x, z.
∗
Lemma 12. If a policy ρ = (x, y, z) ∈ Ssymm solves Umin (ρ) = Umin
(1), then there is some policy
∗
ρ′ = x′ , z ′ ∈ S ′ that solves Umin (ρ′) = Umin
(1).

Applying the framework in Proposition 1 to this proof with S = S ′ and using ŵ as the user and item

Lean-expanded paper semantic target

```
type:
{m n : ℕ} →
  GCG24UserItemFairness.EstimatedRecommendationModel m n →
    GCG24UserItemFairness.RecommendationModel.SymmetricData m n 3 → Prop

value:
fun {m n : ℕ} (E : GCG24UserItemFairness.EstimatedRecommendationModel m n)
    (Sest : GCG24UserItemFairness.RecommendationModel.SymmetricData m n 3) =>
  E.estimatedModel = Sest.model
```

Direct retained dependencies

- GCG24UserItemFairness.EstimatedRecommendationModel
- GCG24UserItemFairness.EstimatedRecommendationModel.estimatedModel
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.SymmetricData

Recorded source-to-declaration screening **Verdict:** matches

Reason: The source’s estimated-utility construction is represented by the declared reduction of the estimated model, with no added behavioral or empirical assumption.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.assumption_theorem4_parameter_domain

Paper-local declaration:
GCG24UserItemFairness.assumption_theorem4_parameter_domain
Source locator: cited publication, lines 456-470

Verbatim paper-source connection

Theorem 4. If β > n1 and w and ŵ are as described above, then fairness constraints can arbitrarily worsen the price of misestimation.
• The price of misestimation without fairness constraints is low: for all {vj }, there is a
∗
recommendation policy ρ̂ that solves the misestimated problem Umin
(0, ŵ) so that
M
πU
(0, w, ŵ) ≤

1
.
2

• The price of misestimation with fairness constraints can be arbitrarily large: ∀ϵ, there exists

[Next verbatim source excerpt]

Recall Theorem 4.
Theorem 4. If β > n1 and w and ŵ are as described above, then fairness constraints can arbitrarily worsen the price of misestimation.
• The price of misestimation without fairness constraints is low: for all {vj }, there is a
∗
recommendation policy ρ̂ that solves the misestimated problem Umin
(0, ŵ) so that
1
M
πU
(0, w, ŵ) ≤ .
2
37

[form-feed in source extraction]
• The price of misestimation with fairness constraints can be arbitrarily large: ∀ϵ, there exists

Settled source reading For Theorem 4, the normalized three-type construction uses masses beta, beta, and 1 - 2 beta; the stated beta range keeps all three masses positive.

Lean-expanded paper semantic target

type:
ℕ → ℝ → ℝ → Prop

value:
fun (n : ℕ) (beta eps : ℝ) => 0 < eps ∧ (↑n)⁻¹ < beta ∧ beta < 1 / 2

Recorded source-to-declaration screening **Verdict:** matches
Reason: The source requires the stated beta and epsilon parameter inequalities for its Theorem 4 construction; this premise records that same domain.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.assumption_theorem4_true_model_reduction

Paper-local declaration:
GCG24UserItemFairness.assumption_theorem4_true_model_reduction
Source locator: cited publication, lines 444-455

Verbatim paper-source connection

Section 4, suppose that there are 2 types of users, with opposing preferences (i.e., with values
v1 , v2 ..., vn and vn , vn−1 , ..., v1 , respectively) – and the platform has correctly estimated these
preferences. However, now, these two types only make up a proportion β of the population each.
Now, we suppose that there is a fraction 1 − β of the user population who are “new” users. We assume
that these users are drawn from the same distribution as the remaining users, but the platform does
not know their preferences. It thus constructs a prior by averaging over the known users’ preferences
v +v
– (mis)estimating the users’ utility for each item j as j n−j+1
.
2
6

[Next verbatim source excerpt]

Item fairness constraints with misestimation. First, notice that in the misestimated value matrix
ŵ there are three distinct user value types, and thus we can identify Ssymm with Δ3n−1 and a policy
ρ ∈ Ssymm with vectors x, y, z, where x is the recommendation policy for users with value ŵij = vj ,
y is the recommendation policy for users with value ŵij = vn−j+1 , and z is the recommendation
policy for the mis-estimated users.
Let

S ′ = {x, y, z ∈ Ssymm : xj = yn−j+1 , zj = zn−j+1 for all j}.
Since y is completely determined by x, we can identify S ′ with Δ2n−1 and can identify x, y, z ∈ S ′
simply by x, z.
∗
Lemma 12. If a policy ρ = (x, y, z) ∈ Ssymm solves Umin (ρ) = Umin
(1), then there is some policy
∗
ρ′ = x′ , z ′ ∈ S ′ that solves Umin (ρ′) = Umin
(1).

Applying the framework in Proposition 1 to this proof with S = S ′ and using ŵ as the user and item

Lean-expanded paper semantic target

```
type:
{m n : ℕ} →
  GCG24UserItemFairness.EstimatedRecommendationModel m n →
    GCG24UserItemFairness.RecommendationModel.SymmetricData m n 2 → Prop

value:
fun {m n : ℕ} (E : GCG24UserItemFairness.EstimatedRecommendationModel m n)
    (Strue : GCG24UserItemFairness.RecommendationModel.SymmetricData m n 2) =>
  E.trueModel = Strue.model
```

Direct retained dependencies

- GCG24UserItemFairness.EstimatedRecommendationModel
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.SymmetricData

Recorded source-to-declaration screening **Verdict:** matches

Reason: The source’s true-utility side of the construction reduces to the displayed two-type model; this premise records that source-model reduction.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GCG24UserItemFairness.assumption_theorem4_universal_value_vector

Paper-local declaration:
GCG24UserItemFairness.assumption_theorem4_universal_value_vector
Source locator: cited publication, lines 368-372

Verbatim paper-source connection

can be captured by a single parameter. Consider the following utility matrix structure. Let v1 >
v2 > ... > vn > 0. Suppose that there are two user types: a user i either has utility wij = vj or
wij = vn−j+1 , and say that user i is of type 1 or 2 respectively. In words, the two types have opposite
preferences, but the preferences can otherwise be generic and be for any number of items. Now, the
direct solution by symmetry for the example no longer directly holds – the items in the middle, not

[Next verbatim source excerpt]

[form-feed in source extraction]
Theorem 4. If β > n1 and w and ŵ are as described above, then fairness constraints can arbitrarily
worsen the price of misestimation.
• The price of misestimation without fairness constraints is low: for all {vj }, there is a
∗
recommendation policy ρ̂ that solves the misestimated problem Umin
(0, ŵ) so that
M
πU
(0, w, ŵ) ≤

1
.
2

• The price of misestimation with fairness constraints can be arbitrarily large: ∀ϵ, there exists

Lean-expanded paper semantic target

type:
{n : ℕ} → (GCG24UserItemFairness.Item n → ℝ) → Prop

value:
fun {n : ℕ} (v : GCG24UserItemFairness.Item n → ℝ) =>
  (∀ (j : GCG24UserItemFairness.Item n), 0 < v j) ∧ GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v

Direct retained dependencies

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex

Recorded source-to-declaration screening **Verdict:** matches

Reason: The no-fairness clause quantifies over the source's admissible value vectors; this premise captures the same universal vector condition.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

Main-text source claims

Review row 1

Source locator: cited publication, lines 292-316

Verbatim source input

Proposition 1. Suppose that for a set of recommendation policies S ⊆ Δm
n−1 ,
(i) S can be described by a finite set of linear constraints
(ii) There exists an optimal solution ρ∗ to Problem (2) such that ρ∗ ∈ S
(iii) ρ∗ is the unique feasible solution to Problem (2) in S
∗
Multiple policies ρ̂ may optimize Umin
(γ, ŵ), and different policies ρ̂ may have different values of
Umin (ρ̂, w). In our experimental results, we use the policy ρ̂ found by the convex optimization solver.
1

4

[form-feed in source extraction]
Then, finding an optimal solution ρ∗ to Problem (2) can be reduced to solving a linear program L.
With Proposition 1, the key technical challenges become: (a) given utility matrix w, finding S
satisfying the above conditions; (b) given S and w, finding a closed-form expression for ρ∗ ; (c) given
∗
a closed form for ρ∗ in terms of w, reasoning about the properties of item fair solution Umin
(1, w).
To do this, for our main results, we construct S as a set of recommendations with a particular
symmetric structure. Suppose users come in K types, where a user of type k shares a utility vector.
Then, consider Ssymm = {ρ : ρi = ρi′ if wi = wi′ } ⊆ Δm
n−1 , the set of policies where users of the
same type are given the same recommendation probabilities. Note that, in the extreme case, each user
is their own type. Furthermore, let ρk denote the recommendation policy for type k, for ρ ∈ Ssymm .

Expanded PaperInterface specification

```
∀ {m n : ℕ} [inst : NeZero m] [inst_1 : NeZero n] (W : GCG24UserItemFairness.RecommendationModel m n),
  GCG24UserItemFairness.assumption_positive_recommendation_utilities W →
    ∀ (S : Set (GCG24UserItemFairness.Policy m n))
      (D : GCG24UserItemFairness.RecommendationModel.FiniteLinearPolicySetDescription S)
      (ρstar : GCG24UserItemFairness.Policy m n),
      W.IsOptimalAtLevel 1 ρstar →
        ρstar ∈ S →
          (∀ ρ ∈ S, W.itemFairness ρ = W.optimalItemFairness → ρ = ρstar) →
            (W.PropositionOneLPFeasible D ρstar W.optimalItemFairness ∧
                ∀ (ρ' : GCG24UserItemFairness.Policy m n) (ell' : ℝ),
                  W.PropositionOneLPFeasible D ρ' ell' → ell' ≤ W.optimalItemFairness) ∧
              ∀ (ρ : GCG24UserItemFairness.Policy m n) (ell : ℝ),
                (W.PropositionOneLPFeasible D ρ ell ∧
                    ∀ (ρ' : GCG24UserItemFairness.Policy m n) (ell' : ℝ),
                      W.PropositionOneLPFeasible D ρ' ell' → ell' ≤ ell) →
                  ρ = ρstar ∧ ell = W.optimalItemFairness
```

Retained governing declarations

- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.FiniteLinearPolicySetDescription
- GCG24UserItemFairness.RecommendationModel.IsOptimalAtLevel
- GCG24UserItemFairness.RecommendationModel.PropositionOneLPFeasible
- GCG24UserItemFairness.RecommendationModel.itemFairness
- GCG24UserItemFairness.RecommendationModel.optimalItemFairness
- GCG24UserItemFairness.assumption_positive_recommendation_utilities

Lean-elaborated claim roles

1. parameter: ℕ
2. parameter: ℕ
3. assumption: NeZero m
4. assumption: NeZero n
5. parameter: GCG24UserItemFairness.RecommendationModel m n
6. assumption: GCG24UserItemFairness.assumption_positive_recommendation_utilities W
7. parameter: Set (GCG24UserItemFairness.Policy m n)
8. parameter: GCG24UserItemFairness.RecommendationModel.FiniteLinearPolicySetDescription S
9. parameter: GCG24UserItemFairness.Policy m n
10. assumption: W.IsOptimalAtLevel 1 ρstar
11. assumption: ρstar ∈ S

12. assumption: ∀ ρ ∈ S, W.itemFairness ρ = W.optimalItemFairness → ρ = ρstar
13. conclusion: (W.PropositionOneLPFeasible D ρstar W.optimalItemFairness ∧
    ∀ (ρ' : GCG24UserItemFairness.Policy m n) (ell' : ℝ),
      W.PropositionOneLPFeasible D ρ' ell' → ell' ≤ W.optimalItemFairness) ∧
  ∀ (ρ : GCG24UserItemFairness.Policy m n) (ell : ℝ),
    (W.PropositionOneLPFeasible D ρ ell ∧
        ∀ (ρ' : GCG24UserItemFairness.Policy m n) (ell' : ℝ), W.PropositionOneLPFeasible D ρ' ell' → ell' ≤ ell) →
      ρ = ρstar ∧ ell = W.optimalItemFairness

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.proposition1_symmetric_lp_reduction

Recorded source-to-Spec screening **Verdict:** matches

Reason: Proposition 1’s finite-linear-set, existence, and unique-in-set conditions reducing the fairness problem to an equality LP are retained in the Spec.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 2

Source locator: cited publication, lines 312-338

Verbatim source input

symmetric structure. Suppose users come in K types, where a user of type k shares a utility vector. Then, consider Ssymm = {ρ : ρi = ρi′ if wi = wi′ } ⊆ Δm
n−1 , the set of policies where users of the
same type are given the same recommendation probabilities. Note that, in the extreme case, each user is their own type. Furthermore, let ρk denote the recommendation policy for type k, for ρ ∈ Ssymm .
Proposition 2. Ssymm satisfies conditions (i) and (ii) in Proposition 1. Furthermore, solutions ρ ∈ Ssymm to Problem (2) have a sparse structure:
• If ρ is a basic feasible solution to the linear program L in Proposition 1, then there are at most n + K − 1 type-item pairs (k, j) such that ρkj > 0 (out of nK possible pairs).
• If ρ is also optimal, then there are at most K − 1 items that are ever recommended to more than one type of user, i.e., where ρkj , ρk′ j > 0 for k′= k ′ .
For our main results, we will leverage the sparsity structure in Proposition 2 to show, with further restrictions on utility matrix w, that Ssymm also satisfies (iii). We will then derive closed-form expressions for that solution, and show how it changes as a function of w.
The sparsity structure in Proposition 2 is also interesting in its own right. Without item fairness constraints, solutions will be highly sparse: each user will be recommended their most preferred item deterministically; each other item will have a recommendation probability of zero. Once we add item fairness constraints, solutions do not necessarily remain sparse. However, Proposition 2 shows that sparse solutions still arise under item fairness constraints in settings with symmetric solutions.

4

User preference diversity and fairness tradeoffs

We now use the theoretical framework developed in the above section to understand the effect of the structure of user utilities on the price of fairness. In particular, we identify “free fairness,” i.e., the price of fairness is low, when preferences are sufficiently diverse.

[Next verbatim source excerpt]

Proof. We will first prove that Ssymm satisfies conditions (i) and (ii), and then show sparsity.
Part 1. Recall that for fixed w,
\

Ssymm = {ρ : ρi = ρi′ if wi = wi′ } =

{ρ : ρi = ρi′ }

i,i′ :wi =wi′

which is a finite set of linear constraints, so condition (i) holds.
Now, we show condition (ii), that always there is some ϕ∗ ∈ Ssymm that is an optimal solution to Problem 1. Let ρ be an arbitrary optimal solution to Problem 1. Then ρ is also feasible: Imin (ρ) =
∗
Imin
.
We first define a piece of convenient notation: if two users i, i′ share the same utility vector wi = wi′ , we say that they have the same type τ (i) = τ (i′). If τ (i) = τ (i′) := τ , we write wij = wi′ j := wτ j

[Next verbatim source excerpt]

symmetric structure. Suppose users come in K types, where a user of type k shares a utility vector. Then, consider Ssymm = {ρ : ρi = ρi′ if wi = wi′ } ⊆ Δm
n−1 , the set of policies where users of the
same type are given the same recommendation probabilities. Note that, in the extreme case, each user is their own type. Furthermore, let ρk denote the recommendation policy for type k, for ρ ∈ Ssymm .

[Next verbatim source excerpt]

We first define a piece of convenient notation: if two users i, i′ share the same utility vector wi = wi′ , we say that they have the same type τ (i) = τ (i′). If τ (i) = τ (i′) := τ , we write wij = wi′ j := wτ j

Expanded PaperInterface specification

```
∀ {m n K : ℕ} [inst : NeZero m] [inst_1 : NeZero n] [NeZero K]
  (S : GCG24UserItemFairness.RecommendationModel.SymmetricData m n K) (hTypes : Function.Surjective S.types.toType),
  GCG24UserItemFairness.assumption_positive_recommendation_utilities S.model →
    GCG24UserItemFairness.assumption_proposition2_utility_rows_determine_types S →
      ∀ (ρsrc : GCG24UserItemFairness.Policy m n) (ell : ℝ)
        (hsource :
          (GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.ofSymmetricData S).ActiveBasis ρsrc ell),
        (∀ (ρ : GCG24UserItemFairness.Policy m n),
            S.model.UtilityRowSymmetric ρ ↔
              ∀ (c : GCG24UserItemFairness.UserTypeAssignment.TypeSymmetryLinearConstraintIndex m n),
                GCG24UserItemFairness.RecommendationModel.policyLinearConstraintHolds
                  (GCG24UserItemFairness.UserTypeAssignment.typeSymmetryLinearCoefficient S.types c) 0
                  GCG24UserItemFairness.RecommendationModel.PolicyLinearComparison.eq ρ) ∧
          (∃ (ρsym : GCG24UserItemFairness.Policy m n),
            S.model.UtilityRowSymmetric ρsym ∧ S.model.IsOptimalAtLevel 1 ρsym) ∧
            (GCG24UserItemFairness.UserTypeAssignment.descendTypePolicy S.types
                (AppliedModelingLib.Policy.FiberRepresentatives.ofSurjective hTypes) ρsrc).ActivePairsBound ∧
              (ell = (S.canonicalReductionOfSurjective hTypes).reduced.optimalItemFairness →
                (GCG24UserItemFairness.UserTypeAssignment.descendTypePolicy S.types
                    (AppliedModelingLib.Policy.FiberRepresentatives.ofSurjective hTypes) ρsrc).SharedItemsBound)
```

Retained governing declarations

- AppliedModelingLib.Policy.FiberRepresentatives.ofSurjective
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel.IsOptimalAtLevel
- GCG24UserItemFairness.RecommendationModel.PolicyLinearComparison
- GCG24UserItemFairness.RecommendationModel.SymmetricData
- GCG24UserItemFairness.RecommendationModel.SymmetricData.canonicalReductionOfSurjective
- GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.ActiveBasis
- GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.ofSymmetricData
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.RecommendationModel.UtilityRowSymmetric
- GCG24UserItemFairness.RecommendationModel.policyLinearConstraintHolds
- GCG24UserItemFairness.ReductionWitness
- GCG24UserItemFairness.TypePolicy.ActivePairsBound
- GCG24UserItemFairness.TypePolicy.SharedItemsBound
- GCG24UserItemFairness.TypeWeightedRecommendationModel.optimalItemFairness
- GCG24UserItemFairness.User
- GCG24UserItemFairness.UserTypeAssignment.TypeSymmetryLinearConstraintIndex
- GCG24UserItemFairness.UserTypeAssignment.descendTypePolicy
- GCG24UserItemFairness.UserTypeAssignment.typeSymmetryLinearCoefficient
- GCG24UserItemFairness.assumption_positive_recommendation_utilities
- GCG24UserItemFairness.assumption_proposition2_utility_rows_determine_types

Lean-elaborated claim roles

1. parameter: ℕ
2. parameter: ℕ
3. parameter: ℕ
4. assumption: NeZero m
5. assumption: NeZero n
6. assumption: NeZero K
7. parameter: GCG24UserItemFairness.RecommendationModel.SymmetricData m n K
8. assumption: Function.Surjective S.types.toType
9. assumption: GCG24UserItemFairness.assumption_positive_recommendation_utilities S.model
10. assumption: GCG24UserItemFairness.assumption_proposition2_utility_rows_determine_types S
11. parameter: GCG24UserItemFairness.Policy m n
12. parameter: ℝ
13. parameter: (GCG24UserItemFairness.RecommendationModel.UserSymmetricEqualityLP.ofSymmetricData S).ActiveBasis ρsrc ell
14. conclusion: (∀ (ρ : GCG24UserItemFairness.Policy m n),
    S.model.UtilityRowSymmetric ρ ↔
      ∀ (c : GCG24UserItemFairness.UserTypeAssignment.TypeSymmetryLinearConstraintIndex m n),
        GCG24UserItemFairness.RecommendationModel.policyLinearConstraintHolds
          (GCG24UserItemFairness.UserTypeAssignment.typeSymmetryLinearCoefficient S.types c) 0
          GCG24UserItemFairness.RecommendationModel.PolicyLinearComparison.eq ρ) ∧
  (∃ (ρsym : GCG24UserItemFairness.Policy m n), S.model.UtilityRowSymmetric ρsym ∧ S.model.IsOptimalAtLevel 1 ρsym) ∧
    (GCG24UserItemFairness.UserTypeAssignment.descendTypePolicy S.types
        (AppliedModelingLib.Policy.FiberRepresentatives.ofSurjective hTypes) ρsrc).ActivePairsBound ∧
      (ell = (S.canonicalReductionOfSurjective hTypes).reduced.optimalItemFairness →
        (GCG24UserItemFairness.UserTypeAssignment.descendTypePolicy S.types
          (AppliedModelingLib.Policy.FiberRepresentatives.ofSurjective hTypes) ρsrc).SharedItemsBound)

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.proposition2_symmetric_optimum_exists

Recorded source-to-Spec screening **Verdict:** matches

Reason: Under approved context GCG24-P2-EQUALITY-DEFINED-TYPE-PARTITION-01, agreeWithinTypes plus UtilityRowsDetermineTypes makes declared types exactly the utility-row classes. The target gives the finite linear characterization of S_symm, existence of a symmetric gamma = 1 optimum, the n+K-1 active type-item bound for a source-coordinate basic feasible solution, and the K-1 shared-item bound when its objective is optimal.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Appendix and supplement source claims

Review row 3

Source locator: cited publication, lines 1742-1745

Verbatim source input

Lemma 1. Since wij > 0 for all j, lmin
> 0.
∗
= maxρ lmin (ρ) > 0.

Expanded PaperInterface specification

```
∀ {m n : ℕ} [NeZero m] [inst : NeZero n] (W : GCG24UserItemFairness.RecommendationModel m n),
  W.Positive → 0 < W.optimalItemFairness
```

Retained governing declarations

- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.Positive
- GCG24UserItemFairness.RecommendationModel.optimalItemFairness

Lean-elaborated claim roles

1. parameter: ℕ
2. parameter: ℕ
3. assumption: NeZero m
4. assumption: NeZero n
5. parameter: GCG24UserItemFairness.RecommendationModel m n
6. assumption: W.Positive
7. conclusion: Real.lt✝ 0 W.optimalItemFairness

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.appendix_c_lemma1_item_fairness_positive

Recorded source-to-Spec screening **Verdict:** matches

Reason: The source asserts strict positivity of optimal item fairness under positive utilities; the expanded Spec states that same conclusion under the displayed positive-model premise.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 4

Source locator: cited publication, lines 3071-3078

Verbatim source input

Lemma 10. If α ≤ 1/2, then t(α) ≤ (n + 1)/2.
Now, for 1 ≤ t ≤ n, define A(t) = {α : t(α) = t}. Since t(α) is weakly increasing in α by Lemma
7, A(t) is an interval in (0, 1), and [A(1), A(2), ..., A(n)] forms a consecutive partition of (0, 1). We
∗
will now show that lmin
(α) is increasing on each of the first ⌊(n + 1)/2⌋ intervals, which by Lemma
10 contain all α ∈ (0, 1/2].
∗

Expanded PaperInterface specification

```
∀ {n : ℕ} [inst : NeZero n] {alpha : ℝ} {v : GCG24UserItemFairness.Item n → ℝ} (halpha0 : 0 < alpha)
  (halpha1 : alpha < 1),
  alpha ≤ 1 / 2 →
    ∀ (hpos : ∀ (j : GCG24UserItemFairness.Item n), 0 < v j),
      GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v →
        have t : GCG24UserItemFairness.Item n :=
          (GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy alpha v halpha0 halpha1
              hpos).lastActiveTypeZero;
        ↑t ≤ ↑(GCG24UserItemFairness.OpposingTypes.reverseItem t)
```

Retained governing declarations

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex
- GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy
- GCG24UserItemFairness.OpposingTypes.reverseItem
- GCG24UserItemFairness.TypePolicy.lastActiveTypeZero

Lean-elaborated claim roles

1. parameter: ℕ
2. assumption: NeZero n
3. parameter: ℝ
4. parameter: GCG24UserItemFairness.Item n → ℝ
5. assumption: 0 < alpha
6. assumption: alpha < 1
7. assumption: alpha ≤ 1 / 2
8. assumption: ∀ (j : GCG24UserItemFairness.Item n), 0 < v j
9. assumption: GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v
10. conclusion: (↑(GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy alpha v halpha0 halpha1 hpos).lastActiveTypeZero).le
  ↑(GCG24UserItemFairness.OpposingTypes.reverseItem
    (GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy alpha v halpha0 halpha1 hpos).lastActiveTypeZero)

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.appendix_d_lemma10_active_optimizer_pivot_at_or_before_midpoint

Recorded source-to-Spec screening **Verdict:** matches

Reason: Lemma 10's alpha-at-most-one-half pivot bound is preserved as the selected active pivot lying at or before its reverse-item midpoint.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 5

Source locator: cited publication, lines 3079-3100

Verbatim source input

Lemma 11. Imin
(α) is increasing on A(t) for t ≤ (n + 1)/2.
∗
Finally, we show that this implies Imin
(α) is increasing on 0 < α ≤ 1/2. Notice that Ij (x, y, α)
is continuous in α for each fixed x, y, so Imin (x, y, α) = minj Ij (x, y, α) is continuous in α for
∗
each fixed x, y. Recall that Imin
(α) = maxx,y∈Δn−1 Imin (x, y, α). As the supremum of a set of
∗
continuous functions on a compact set must itself be continuous , this means that Imin
(α) must be
∗
continuous in α, since Δn−1 × Δn−1 is compact. Since Imin (α) is continuousSand is increasing on
∗
each interval {A(t)}t≤(n+1)/2 by Lemma 11, Imin
(α) must be increasing on t≤(n+1)/2 A(t). By
S
∗
Lemma 10, t≤(n+1)/2 A(t) ⊃ (0, 1/2]. So Imin (α) is increasing on (0, 1/2].

Now, we show Lemmas 10 and 11.

Expanded PaperInterface specification

```
∀ {n : ℕ} [inst : NeZero n] {alpha alpha' : ℝ} {v : GCG24UserItemFairness.Item n → ℝ} (halpha0 : 0 < alpha)
  (halpha1 : alpha < 1) (halpha0' : 0 < alpha') (halpha1' : alpha' < 1),
  alpha ≤ alpha' →
    ∀ (hpos : ∀ (j : GCG24UserItemFairness.Item n), 0 < v j),
      GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v →
        (GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy alpha v halpha0 halpha1
              hpos).lastActiveTypeZero =
            (GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy alpha' v halpha0' halpha1'
              hpos).lastActiveTypeZero →
          (have t : GCG24UserItemFairness.Item n :=
              (GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy alpha v halpha0 halpha1
                hpos).lastActiveTypeZero;
            ↑t ≤ ↑(GCG24UserItemFairness.OpposingTypes.reverseItem t)) →
            (GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha v).optimalItemFairness ≤
              (GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha' v).optimalItemFairness
```

Retained governing declarations

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex
- GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy
- GCG24UserItemFairness.OpposingTypes.reverseItem
- GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel
- GCG24UserItemFairness.TypePolicy.lastActiveTypeZero
- GCG24UserItemFairness.TypeWeightedRecommendationModel.optimalItemFairness

Lean-elaborated claim roles

1. parameter: ℕ
2. assumption: NeZero n
3. parameter: ℝ
4. parameter: ℝ
5. parameter: GCG24UserItemFairness.Item n → ℝ
6. assumption: 0 < alpha
7. assumption: alpha < 1
8. assumption: 0 < alpha'
9. assumption: alpha' < 1
10. assumption: alpha ≤ alpha'
11. assumption: ∀ (j : GCG24UserItemFairness.Item n), 0 < v j
12. assumption: GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v
13. assumption: (GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy alpha v halpha0 halpha1 hpos).lastActiveTypeZero =
  (GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy alpha' v halpha0' halpha1' hpos).lastActiveTypeZero
14. assumption: have t : GCG24UserItemFairness.Item n :=
  (GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy alpha v halpha0 halpha1 hpos).lastActiveTypeZero;
↑t ≤ ↑(GCG24UserItemFairness.OpposingTypes.reverseItem t)
15. conclusion: Real.le† (GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha v).optimalItemFairness
  (GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha' v).optimalItemFairness

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.appendix_d_lemma11_optimal_item_fairness_mono_on_active_pivot_interval

Recorded source-to-Spec screening **Verdict:** matches

Reason: The source’s monotonicity on a fixed active-pivot interval is represented by ordered alpha values with the same pivot and the stated midpoint condition implying nondecreasing optimum item fairness.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 6

Source locator: cited publication, lines 2098-2182

Verbatim source input

Lemma 3. For all 0 < α < 1, Umin
(α) = 1.
∗
∗
F
(Umin −Umin (1,α))/U ∗
By Lemma 3, Umin
(α) := Umin
is constant in α, so πU
min and it is
|l (α) =
∗
enough to show Umin (1, α) increases in α.
∗

∗

Recall the definition of Ssymm :
{ρ : ρi = ρi′ if wi = wi′ }.
By Proposition 2, we know that Ssymm is a satisfies conditions (i) and (ii) on S for Proposition 1.
Thus by Proposition 1, it is sufficient to find ϕ such that
ϕ ∈ arg

max

ρ∈S∩Δm
n−1 ,λ

λ

subject to lj (ρ) = λ

∀j,

and show that this ϕ is unique, where the linear program is derived from the proof of Proposition 1.
Since each user in our population only has one of two possible value vectors, wi = (v1 , ..., vn)
or wi = (vn , ..., v1), we can identify Ssymm with Δ2n−1 , and identify ρ ∈ Ssymm with a pair x, y.
Expanding the expression for lj (ρ) and making explicit the dependence on α, we see that
P
i ρij wij (α)
lj (ρ, α) = P
wij (α)
P i
P
i:τ (i)=1 ρij vj +
i:τ (i)=2 ρij vn−j+1
P
P
=
i:τ (i)=1 vj +
i:τ (i)=2 vn−j+1
mαxj vj + m(1 − α)yj vn−j+1
mαvj + m(1 − α)vn−j+1
= qj (α)xj + (1 − qj (α))yj

=

where
qj (α) :=

αvj
.
αvj + (1 − α)vn−j+1

Note that for all j and 0 < α < 1, we have 0 < qj , 1 − qj < 1. We can thus simplify
ϕ = arg

max

x,y∈Δ2n−1 ,λ

λ
(6)

subject to qj (α)xj + (1 − qj (α))yj = λ

∀j,

First, we will show uniqueness and find a closed form to Problem 6 in Lemmas 4 and 5. Then, we
∗
will use this closed forms to show that Umin
(1, α) is increasing for α < 1/2. By symmetry, this
∗
implies that Umin
(1, α) must be decreasing for α > 1/2 and we will be done.

Expanded PaperInterface specification

```
∀ {n : ℕ} [inst : NeZero n] {alpha : ℝ} {v : GCG24UserItemFairness.Item n → ℝ},
  0 < alpha →
    alpha < 1 →
      (∀ (j : GCG24UserItemFairness.Item n), 0 < v j) →
        GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v →
          (GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha v).optimalTypeFairnessAtLevel 0 = 1
```

Retained governing declarations

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex
- GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel
- GCG24UserItemFairness.TypeWeightedRecommendationModel.optimalTypeFairnessAtLevel

Lean-elaborated claim roles

1. parameter: ℕ
2. assumption: NeZero n
3. parameter: ℝ
4. parameter: GCG24UserItemFairness.Item n → ℝ
5. assumption: 0 < alpha
6. assumption: alpha < 1
7. assumption: ∀ (j : GCG24UserItemFairness.Item n), 0 < v j
8. assumption: GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v
9. conclusion: (GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha v).optimalTypeFairnessAtLevel 0 = 1

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.appendix_d_lemma3_unconstrained_baseline

Recorded source-to-Spec screening **Verdict:** matches

Reason: The target states that the unconstrained optimum at gamma = 0 equals 1 for the positive, strictly decreasing two-type opposing model with 0 < alpha < 1. Its normalized type utilities and minimum over the two represented types are the reduced-model form of the source U_min*(alpha) = 1.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 7

Source locator: cited publication, lines 2183-2188

Verbatim source input

Lemma 4. Problem 6 has a unique solution (x, y, λ). Moreover, the solution is sparse: there is some
1 ≤ t ≤ n such that for j > t, xj = 0, and for j < t, yj = 0.
Thus, to solve our original problem (Problem 1) we merely need to evaluate the user fairness Umin of
x, y solving Problem 6 and show that this is increasing. We will do this in the remainder of the proof.
For each α, let x, y be the solution to the corresponding Problem 6, and define7
t(α) := max{j : xj (α) > 0}.

Expanded PaperInterface specification

```
∀ {n : ℕ} [NeZero n] {alpha : ℝ} {v : GCG24UserItemFairness.Item n → ℝ},
  0 < alpha →
    alpha < 1 →
      (∀ (j : GCG24UserItemFairness.Item n), 0 < v j) →
        GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v →
          ∃ (t : GCG24UserItemFairness.Item n) (ρstar : GCG24UserItemFairness.TypePolicy 2 n) (ellstar : ℝ),
            (∀ (j : GCG24UserItemFairness.Item n),
               GCG24UserItemFairness.OpposingTypes.pairShare alpha v j * ((ρstar 0) j).toReal +
                  (1 - GCG24UserItemFairness.OpposingTypes.pairShare alpha v j) * ((ρstar 1) j).toReal =
                ellstar) ∧
              (∀ (ρ : GCG24UserItemFairness.TypePolicy 2 n) (ell : ℝ),
                 GCG24UserItemFairness.OpposingTypes.problem6LPFeasible alpha v ρ ell → ell ≤ ellstar) ∧
                (∀ {j : GCG24UserItemFairness.Item n}, ↑t < ↑j → ((ρstar 0) j).toReal = 0) ∧
                  (∀ {j : GCG24UserItemFairness.Item n}, ↑j < ↑t → ((ρstar 1) j).toReal = 0) ∧
                    ∀ (ρ : GCG24UserItemFairness.TypePolicy 2 n) (ell : ℝ),
                      (∀ (j : GCG24UserItemFairness.Item n),
                         GCG24UserItemFairness.OpposingTypes.pairShare alpha v j * ((ρ 0) j).toReal +
                            (1 - GCG24UserItemFairness.OpposingTypes.pairShare alpha v j) * ((ρ 1) j).toReal =
                          ell) →
                        (∀ (ρ' : GCG24UserItemFairness.TypePolicy 2 n) (ell' : ℝ),
                           GCG24UserItemFairness.OpposingTypes.problem6LPFeasible alpha v ρ' ell' → ell' ≤ ell) →
                          ρ = ρstar ∧ ell = ellstar
```

Retained governing declarations

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex
- GCG24UserItemFairness.OpposingTypes.pairShare
- GCG24UserItemFairness.OpposingTypes.problem6LPFeasible
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.UserType

Lean-elaborated claim roles

1. parameter: ℕ
2. assumption: NeZero n
3. parameter: ℝ
4. parameter: GCG24UserItemFairness.Item n → ℝ
5. assumption: 0 < alpha
6. assumption: alpha < 1
7. assumption: ∀ (j : GCG24UserItemFairness.Item n), 0 < v j
8. assumption: GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v
9. conclusion: ∃ (t : GCG24UserItemFairness.Item n) (ρstar : GCG24UserItemFairness.TypePolicy 2 n) (ellstar : ℝ),
  (∀ (j : GCG24UserItemFairness.Item n),
     GCG24UserItemFairness.OpposingTypes.pairShare alpha v j * ((ρstar 0) j).toReal +
        (1 - GCG24UserItemFairness.OpposingTypes.pairShare alpha v j) * ((ρstar 1) j).toReal =
      ellstar) ∧
    (∀ (ρ : GCG24UserItemFairness.TypePolicy 2 n) (ell : ℝ),
       GCG24UserItemFairness.OpposingTypes.problem6LPFeasible alpha v ρ ell → ell ≤ ellstar) ∧
      (∀ {j : GCG24UserItemFairness.Item n}, ↑t < ↑j → ((ρstar 0) j).toReal = 0) ∧
        (∀ {j : GCG24UserItemFairness.Item n}, ↑j < ↑t → ((ρstar 1) j).toReal = 0) ∧
          ∀ (ρ : GCG24UserItemFairness.TypePolicy 2 n) (ell : ℝ),
            (∀ (j : GCG24UserItemFairness.Item n),
               GCG24UserItemFairness.OpposingTypes.pairShare alpha v j * ((ρ 0) j).toReal +
                  (1 - GCG24UserItemFairness.OpposingTypes.pairShare alpha v j) * ((ρ 1) j).toReal =
                ell) →
              (∀ (ρ' : GCG24UserItemFairness.TypePolicy 2 n) (ell' : ℝ),
                 GCG24UserItemFairness.OpposingTypes.problem6LPFeasible alpha v ρ' ell' → ell' ≤ ell) →
                ρ = ρstar ∧ ell = ellstar

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.appendix_d_lemma4_problem6_unique_sparse_solution

Recorded source-to-Spec screening **Verdict:** matches
Reason: The source’s unique Problem 6 optimizer and its left/right sparse pivot structure are both present in the Spec’s existential optimizer, support, optimality, and uniqueness clauses.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

Review row 8

Source locator: cited publication, lines 2190-2256

Verbatim source input

Lemma 5. Let 0 < α < 1, t := t(α). Define
X 1
Lt =
,
q
j<t j

Rt =

7

1
.
1 − qj
j>t

X

While there is a single optimal solution x, y to Problem 6 for each α, there may be two t that satisfy Lemma 4. In particular, this occurs when for each j, xj = 0 or yj = 0. By contrast, t(α) is unique.

28

[form-feed in source extraction]
Then

1
,
1 + qt Lt + (1 − qt)Rt

∗
lmin
=

and

[U+F8F1 delimiter glyph] l∗
min
[U+F8F4 delimiter glyph]
j<t
[U+F8F2 delimiter glyph] qj ,
Lt
xj = 1 − 1+q L +(1−q
, j=t,
t t
t)Rt
[U+F8F4 delimiter glyph]
[U+F8F3 delimiter glyph]
0,
j>t

[U+F8F1 delimiter glyph]
[U+F8F4 delimiter glyph]
[U+F8F2 delimiter glyph]0,
Rt
,
yj = 1 − 1+qt Lt +(1−q
t)Rt
∗
[U+F8F4 delimiter glyph]
[U+F8F3 delimiter glyph] lmin ,
1−qj

j<t
j=t.
j>t

Note that for t′ = t(α), x, y above may not be valid probability vectors.

Expanded PaperInterface specification

```
∀ {n : ℕ} [inst : NeZero n] {alpha : ℝ} {v : GCG24UserItemFairness.Item n → ℝ},
  0 < alpha →
    alpha < 1 →
      (∀ (j : GCG24UserItemFairness.Item n), 0 < v j) →
        GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v →
          ∃ (ρstar : GCG24UserItemFairness.TypePolicy 2 n) (ellstar : ℝ),
            (∀ (j : GCG24UserItemFairness.Item n),
                GCG24UserItemFairness.OpposingTypes.pairShare alpha v j * ((ρstar 0) j).toReal +
                    (1 - GCG24UserItemFairness.OpposingTypes.pairShare alpha v j) * ((ρstar 1) j).toReal =
                  ellstar) ∧
              (∀ (ρ : GCG24UserItemFairness.TypePolicy 2 n) (ell : ℝ),
                  GCG24UserItemFairness.OpposingTypes.problem6LPFeasible alpha v ρ ell → ell ≤ ellstar) ∧
                (∀ (ρ : GCG24UserItemFairness.TypePolicy 2 n) (ell : ℝ),
                    (∀ (j : GCG24UserItemFairness.Item n),
                        GCG24UserItemFairness.OpposingTypes.pairShare alpha v j * ((ρ 0) j).toReal +
```

```
      (1 - GCG24UserItemFairness.OpposingTypes.pairShare alpha v j) * ((ρ 1) j).toReal =
    ell) →
  (∀ (ρ' : GCG24UserItemFairness.TypePolicy 2 n) (ell' : ℝ),
    GCG24UserItemFairness.OpposingTypes.problem6LPFeasible alpha v ρ' ell' → ell' ≤ ell) →
  ρ = ρstar ∧ ell = ellstar) ∧
have t : GCG24UserItemFairness.Item n := ρstar.lastActiveTypeZero;
ellstar = GCG24UserItemFairness.OpposingTypes.problem6ClosedValue alpha v t ∧
 (∀ {j : GCG24UserItemFairness.Item n},
   ↑j < ↑t →
    ((ρstar 0) j).toReal = ellstar / GCG24UserItemFairness.OpposingTypes.pairShare alpha v j) ∧
 ((ρstar 0) t).toReal =
   1 - ellstar * GCG24UserItemFairness.OpposingTypes.problem6LeftSum alpha v t ∧
  (∀ {j : GCG24UserItemFairness.Item n}, ↑t < ↑j → ((ρstar 0) j).toReal = 0) ∧
   (∀ {j : GCG24UserItemFairness.Item n}, ↑j < ↑t → ((ρstar 1) j).toReal = 0) ∧
    ((ρstar 1) t).toReal =
      1 - ellstar * GCG24UserItemFairness.OpposingTypes.problem6RightSum alpha v t ∧
     ∀ {j : GCG24UserItemFairness.Item n},
       ↑t < ↑j →
        ((ρstar 1) j).toReal =
          ellstar / (1 - GCG24UserItemFairness.OpposingTypes.pairShare alpha v j)
```

Retained governing declarations

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex
- GCG24UserItemFairness.OpposingTypes.pairShare
- GCG24UserItemFairness.OpposingTypes.problem6ClosedValue
- GCG24UserItemFairness.OpposingTypes.problem6LPFeasible
- GCG24UserItemFairness.OpposingTypes.problem6LeftSum
- GCG24UserItemFairness.OpposingTypes.problem6RightSum
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.TypePolicy.lastActiveTypeZero
- GCG24UserItemFairness.UserType

Lean-elaborated claim roles

```
1. parameter: ℕ
2. assumption: NeZero n
3. parameter: ℝ
4. parameter: GCG24UserItemFairness.Item n → ℝ
5. assumption: 0 < alpha
6. assumption: alpha < 1
7. assumption: ∀ (j : GCG24UserItemFairness.Item n), 0 < v j
8. assumption: GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v
9. conclusion: ∃ (ρstar : GCG24UserItemFairness.TypePolicy 2 n) (ellstar : ℝ),
  (∀ (j : GCG24UserItemFairness.Item n),
     GCG24UserItemFairness.OpposingTypes.pairShare alpha v j * ((ρstar 0) j).toReal +
       (1 - GCG24UserItemFairness.OpposingTypes.pairShare alpha v j) * ((ρstar 1) j).toReal =
     ellstar) ∧
   (∀ (ρ : GCG24UserItemFairness.TypePolicy 2 n) (ell : ℝ),
     GCG24UserItemFairness.OpposingTypes.problem6LPFeasible alpha v ρ ell → ell ≤ ellstar) ∧
    (∀ (ρ : GCG24UserItemFairness.TypePolicy 2 n) (ell : ℝ),
      (∀ (j : GCG24UserItemFairness.Item n),
         GCG24UserItemFairness.OpposingTypes.pairShare alpha v j * ((ρ 0) j).toReal +
           (1 - GCG24UserItemFairness.OpposingTypes.pairShare alpha v j) * ((ρ 1) j).toReal =
         ell) →
       (∀ (ρ' : GCG24UserItemFairness.TypePolicy 2 n) (ell' : ℝ),
         GCG24UserItemFairness.OpposingTypes.problem6LPFeasible alpha v ρ' ell' → ell' ≤ ell) →
       ρ = ρstar ∧ ell = ellstar) ∧
     have t : GCG24UserItemFairness.Item n := ρstar.lastActiveTypeZero;
     ellstar = GCG24UserItemFairness.OpposingTypes.problem6ClosedValue alpha v t ∧
      (∀ {j : GCG24UserItemFairness.Item n},
        ↑j < ↑t → ((ρstar 0) j).toReal = ellstar / GCG24UserItemFairness.OpposingTypes.pairShare alpha v j) ∧
      ((ρstar 0) t).toReal = 1 - ellstar * GCG24UserItemFairness.OpposingTypes.problem6LeftSum alpha v t ∧
       (∀ {j : GCG24UserItemFairness.Item n}, ↑t < ↑j → ((ρstar 0) j).toReal = 0) ∧
        (∀ {j : GCG24UserItemFairness.Item n}, ↑j < ↑t → ((ρstar 1) j).toReal = 0) ∧
         ((ρstar 1) t).toReal = 1 - ellstar * GCG24UserItemFairness.OpposingTypes.problem6RightSum alpha v t ∧
          ∀ {j : GCG24UserItemFairness.Item n},
            ↑t < ↑j →
             ((ρstar 1) j).toReal = ellstar / (1 - GCG24UserItemFairness.OpposingTypes.pairShare alpha v j)
```

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.appendix_d_lemma5_problem6_active_optimizer_closed_form

Recorded source-to-Spec screening **Verdict:** matches
Reason: The Spec formalizes the source's selected-pivot closed-form value and policy coordinates under the same interior-alpha, positive, and decreasing-value conditions.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

Review row 9

Source locator: cited publication, lines 2257-2313

Verbatim source input

Lemma 6. For 0 < α < 1/2, if ρ∗ is the optimal recommendation policy, then Ui (ρ∗ , α) ≥
Ui′ (ρ∗ , α) for τ (i) = 1, τ (i′) = 2.
∗
By Lemma 6, Umin
(α) = Ui (α) for i with τ (i) = 2. Then
∗
Umin
(1, α) = Ui (α)
n
X
1
yj vn−j+1
maxj vj j=1
[U+F8EB delimiter glyph]
[U+F8F6 delimiter glyph]
X
1
[U+F8ED delimiter glyph]yt(α) vn−t(α)+1 +
yj vn−j+1 [U+F8F8 delimiter glyph]
=
maxj vj
j>t(α)
[U+F8EB delimiter glyph][U+F8EB delimiter glyph]
[U+F8F6 delimiter glyph]
[U+F8F6 delimiter glyph]
X I ∗ (α)
X I ∗ (α)
1
min
min
[U+F8ED delimiter glyph][U+F8ED delimiter glyph]1 −
[U+F8F8 delimiter glyph] vn−t(α)+1 +
=
vn−j+1 [U+F8F8 delimiter glyph]
maxj vj
1 − qj (α)
1 − qj (α)
j>t(α)
j>t(α)
[U+F8EB delimiter glyph]
[U+F8F6 delimiter glyph]
X I ∗ (α)
1
min
[U+F8ED delimiter glyph]vn−t(α)+1 +
=
(vn−j+1 − vn−t(α)+1)[U+F8F8 delimiter glyph]
maxj vj
1 − qj (α)

=

j>t(α)

∗
The following results will enable us to conclude that Umin
(1, α) is increasing for α < 1/2.

Expanded PaperInterface specification

```
∀ {n : ℕ} [inst : NeZero n] {alpha : ℝ} {v : GCG24UserItemFairness.Item n → ℝ} (halpha0 : 0 < alpha)
  (halpha_half : alpha < 1 / 2) (hpos : ∀ (j : GCG24UserItemFairness.Item n), 0 < v j),
  GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v →
    (GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha v).normalizedTypeUtility
        (GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy alpha v halpha0
          (GCG24UserItemFairness.PaperInterface.appendix_d_lemma6_optimal_policy_type_utility_orderSpec._proof_2
            halpha_half)
          hpos)
        1 ≤
      (GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha v).normalizedTypeUtility
        (GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy alpha v halpha0
          (GCG24UserItemFairness.PaperInterface.appendix_d_lemma6_optimal_policy_type_utility_orderSpec._proof_2
            halpha_half)
          hpos)
        0
```

Retained governing declarations

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex
- GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy

- GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel
- GCG24UserItemFairness.TypeWeightedRecommendationModel.normalizedTypeUtility
- GCG24UserItemFairness.UserType

Lean-elaborated claim roles

1. parameter: ℕ
2. assumption: NeZero n
3. parameter: ℝ
4. parameter: GCG24UserItemFairness.Item n → ℝ
5. assumption: 0 < alpha
6. assumption: alpha < 1 / 2
7. assumption: ∀ (j : GCG24UserItemFairness.Item n), 0 < v j
8. assumption: GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v
9. conclusion: Real.le✝
```
 ((GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha v).normalizedTypeUtility
   (GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy alpha v halpha0
     (GCG24UserItemFairness.PaperInterface.appendix_d_lemma6_optimal_policy_type_utility_orderSpec._proof_2
       halpha_half)
     hpos)
   1)
 ((GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha v).normalizedTypeUtility
   (GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy alpha v halpha0
     (GCG24UserItemFairness.PaperInterface.appendix_d_lemma6_optimal_policy_type_utility_orderSpec._proof_2
       halpha_half)
     hpos)
   0)
```

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.appendix_d_lemma6_optimal_policy_type_utility_order

Recorded source-to-Spec screening **Verdict:** matches

Reason: Lemma 6’s ordering of normalized utility across the two opposing user types for alpha below one half is the conclusion of the selected Spec.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 10

Source locator: cited publication, lines 2315-2316

Verbatim source input

Lemma 7. t(α) is weakly increasing.
*

Expanded PaperInterface specification

```
∀ {n : ℕ} [inst : NeZero n] {alpha alpha' : ℝ} {v : GCG24UserItemFairness.Item n → ℝ} (halpha0 : 0 < alpha)
  (halpha1 : alpha < 1) (halpha0' : 0 < alpha') (halpha1' : alpha' < 1),
  alpha ≤ alpha' →
    ∀ (hpos : ∀ (j : GCG24UserItemFairness.Item n), 0 < v j),
      GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v →
        ↑(GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy alpha v halpha0 halpha1
            hpos).lastActiveTypeZero ≤
          ↑(GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy alpha' v halpha0' halpha1'
            hpos).lastActiveTypeZero
```

Retained governing declarations

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex
- GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy
- GCG24UserItemFairness.TypePolicy.lastActiveTypeZero

Lean-elaborated claim roles

1. parameter: ℕ
2. assumption: NeZero n
3. parameter: ℝ
4. parameter: ℝ
5. parameter: GCG24UserItemFairness.Item n → ℝ
6. assumption: 0 < alpha
7. assumption: alpha < 1
8. assumption: 0 < alpha'
9. assumption: alpha' < 1
10. assumption: alpha ≤ alpha'
11. assumption: ∀ (j : GCG24UserItemFairness.Item n), 0 < v j
12. assumption: GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v
13. conclusion: (↑(GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy alpha v halpha0 halpha1 hpos).lastActiveTypeZero).le
  ↑(GCG24UserItemFairness.OpposingTypes.problem6FirstClosedPolicy alpha' v halpha0' halpha1' hpos).lastActiveTypeZero

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.appendix_d_lemma7_active_optimizer_pivot_monotonicity

Recorded source-to-Spec screening **Verdict:** matches

Reason: The source states that the selected pivot t(alpha) is weakly increasing; the Spec gives the corresponding ordered-alpha pivot inequality.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 11

Source locator: cited publication, lines 2317-2318

Verbatim source input

Lemma 8. Imin
(α) is increasing for α ≤ 1/2.

Expanded PaperInterface specification

```
∀ {n : ℕ} [inst : NeZero n] {alpha alpha' : ℝ} {v : GCG24UserItemFairness.Item n → ℝ},
  0 < alpha →
    alpha < 1 →
      0 < alpha' →
        alpha' < 1 →
          alpha ≤ alpha' →
            alpha ≤ 1 / 2 →
              alpha' ≤ 1 / 2 →
                (∀ (j : GCG24UserItemFairness.Item n), 0 < v j) →
                  GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v →
                    (GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha v).optimalItemFairness ≤
                      (GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha' v).optimalItemFairness
```

Retained governing declarations

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex
- GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel
- GCG24UserItemFairness.TypeWeightedRecommendationModel.optimalItemFairness

Lean-elaborated claim roles

1. parameter: ℕ
2. assumption: NeZero n
3. parameter: ℝ
4. parameter: ℝ
5. parameter: GCG24UserItemFairness.Item n → ℝ
6. assumption: 0 < alpha
7. assumption: alpha < 1
8. assumption: 0 < alpha'
9. assumption: alpha' < 1
10. assumption: alpha ≤ alpha'
11. assumption: alpha ≤ 1 / 2
12. assumption: alpha' ≤ 1 / 2
13. assumption: ∀ (j : GCG24UserItemFairness.Item n), 0 < v j
14. assumption: GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v
15. conclusion: Real.le† (GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha v).optimalItemFairness
  (GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha' v).optimalItemFairness

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.appendix_d_lemma8_selected_pivot_stitching

Recorded source-to-Spec screening **Verdict:** matches

Reason: The source's stitched first-half monotonicity statement is represented by the Spec's ordered-alpha nondecreasing optimal item-fairness conclusion on that domain.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 12

Source locator: cited publication, lines 2320-2339

Verbatim source input

Lemma 9. qj (α) strictly increases as α increases and strictly decreases as j increases.
∗
Since t(α) is increasing and vj decreases in j, vn−t(α)+1 is increasing in α. Also, Imin
(α) is
l∗

(α)

min
increasing for α < 1/2, and qj (α) is increasing, so 1−q
is increasing. For j > t(α), vn−j+1 −
j (α)
∗
vn−t(α)+1 > 0. Thus, Umin (1, α) is increasing.

D.2

Supporting lemma proofs

We will first prove Lemma 9, as it is a simple result that we will use extensively.

Expanded PaperInterface specification

```
∀ {n : ℕ} {alpha alpha' : ℝ} {v : GCG24UserItemFairness.Item n → ℝ} {i j : GCG24UserItemFairness.Item n},
  0 < alpha →
    alpha < 1 →
      0 < alpha' →
        alpha' < 1 →
          alpha < alpha' →
            (∀ (j : GCG24UserItemFairness.Item n), 0 < v j) →
              GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v →
                ↑i < ↑j →
                  GCG24UserItemFairness.OpposingTypes.pairShare alpha v j <
                      GCG24UserItemFairness.OpposingTypes.pairShare alpha' v j ∧
                    GCG24UserItemFairness.OpposingTypes.pairShare alpha v j <
                      GCG24UserItemFairness.OpposingTypes.pairShare alpha v i
```

Retained governing declarations

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex
- GCG24UserItemFairness.OpposingTypes.pairShare

Lean-elaborated claim roles

1. parameter: ℕ
2. parameter: ℝ
3. parameter: ℝ
4. parameter: GCG24UserItemFairness.Item n → ℝ
5. parameter: GCG24UserItemFairness.Item n
6. parameter: GCG24UserItemFairness.Item n
7. assumption: 0 < alpha
8. assumption: alpha < 1
9. assumption: 0 < alpha'
10. assumption: alpha' < 1
11. assumption: alpha < alpha'
12. assumption: ∀ (j : GCG24UserItemFairness.Item n), 0 < v j
13. assumption: GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v
14. assumption: ↑i < ↑j
15. conclusion: GCG24UserItemFairness.OpposingTypes.pairShare alpha v j < GCG24UserItemFairness.OpposingTypes.pairShare alpha' v j ∧
    GCG24UserItemFairness.OpposingTypes.pairShare alpha v j < GCG24UserItemFairness.OpposingTypes.pairShare alpha v i

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.appendix_d_lemma9_pair_share_algebra

Recorded source-to-Spec screening **Verdict:** matches

Reason: The source claims pair shares rise with alpha and fall with item index; the Spec records exactly those two strict algebraic monotonicity directions.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 13

Source locator: cited publication, lines 3379-3402

Verbatim source input

Lemma 12. If a policy ρ = (x, y, z) ∈ Ssymm solves Umin (ρ) = Umin
(1), then there is some policy
∗
ρ′ = x′ , z ′ ∈ S ′ that solves Umin (ρ′) = Umin
(1).

Applying the framework in Proposition 1 to this proof with S = S ′ and using ŵ as the user and item values, we reduce the problem to finding
ϕ = arg

max

x,z∈Δ2n−1 ,λ

λ
(11)

subject to Ij (x, z) = λ

∀j,

and showing that this ϕ is unique.
First, we use an analogue of the sparsity result Proposition 2 to show that the optimal x, z has the following sparse structure.

Expanded PaperInterface specification

```
∀ {n : ℕ} [inst : NeZero n] {beta : ℝ} {v : GCG24UserItemFairness.Item n → ℝ},
  (∀ (j : GCG24UserItemFairness.Item n), 0 < v j) →
    ∀ {ρ : GCG24UserItemFairness.TypePolicy 3 n},
      (GCG24UserItemFairness.OpposingTypes.theorem4EstimatedReducedModel beta v).IsOptimalAtLevel 1 ρ →
        (GCG24UserItemFairness.OpposingTypes.theorem4EstimatedReducedModel beta v).IsOptimalAtLevel 1
            (GCG24UserItemFairness.OpposingTypes.theorem4SymmetrizedPolicy ρ) ∧
          GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy
            (GCG24UserItemFairness.OpposingTypes.theorem4SymmetrizedPolicy ρ)
```

Retained governing declarations

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy
- GCG24UserItemFairness.OpposingTypes.theorem4EstimatedReducedModel
- GCG24UserItemFairness.OpposingTypes.theorem4SymmetrizedPolicy
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.TypeWeightedRecommendationModel.IsOptimalAtLevel

Lean-elaborated claim roles

1. parameter: ℕ
2. assumption: NeZero n
3. parameter: ℝ
4. parameter: GCG24UserItemFairness.Item n → ℝ
5. assumption: ∀ (j : GCG24UserItemFairness.Item n), 0 < v j
6. parameter: GCG24UserItemFairness.TypePolicy 3 n
7. assumption: (GCG24UserItemFairness.OpposingTypes.theorem4EstimatedReducedModel beta v).IsOptimalAtLevel 1 ρ
8. conclusion: (GCG24UserItemFairness.OpposingTypes.theorem4EstimatedReducedModel beta v).IsOptimalAtLevel 1
   (GCG24UserItemFairness.OpposingTypes.theorem4SymmetrizedPolicy ρ) ∧
  GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy
   (GCG24UserItemFairness.OpposingTypes.theorem4SymmetrizedPolicy ρ)

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.appendix_e_lemma12_symmetrized_policy

Recorded source-to-Spec screening **Verdict:** matches

Reason: Lemma 12's existence of a symmetrized optimal policy in the equality-defined subset is the direct conclusion of the selected Spec.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 14

Source locator: cited publication, lines 3403-3412

Verbatim source input

Lemma 13. Let x, z be an optimal basic feasible solution to the linear program Problem 11. There
′
′
′
′
′
is some j ≤ n+1
2 such that for all j > j, xj = 0 and for all j < j, zj = zn−j +1 = 0. Let the
pivot index t denote the minimum such j.
This sparse structure implies uniqueness.

[Next verbatim source excerpt]

S ′ = {x, y, z ∈ Ssymm : xj = yn−j+1 , zj = zn−j+1 for all j}.
Since y is completely determined by x, we can identify S ′ with Δ2n−1 and can identify x, y, z ∈ S ′
simply by x, z.
∗
Lemma 12. If a policy ρ = (x, y, z) ∈ Ssymm solves Umin (ρ) = Umin
(1), then there is some policy
∗
ρ′ = x′ , z ′ ∈ S ′ that solves Umin (ρ′) = Umin
(1).

Applying the framework in Proposition 1 to this proof with S = S ′ and using ŵ as the user and item
values, we reduce the problem to finding
ϕ = arg

max

x,z∈Δ2n−1 ,λ

λ
(11)

subject to lj (x, z) = λ

∀j,

and showing that this ϕ is unique.

[Next verbatim source excerpt]

Lemma 17. Let (x, z) be an optimal solution to Problem 11. For j > n+1
2 , xj = 0.
In particular, if we let h = ⌊ n+1
2 ⌋ indicate the index halfway through the set of items, this means that
we can simplify the problem above to
arg x max
,...,x ,
1

λ

h

z1 ,...,zh ,
λ

subject to lj (x, z) = λ, 1 ≤ j ≤ h
X
X
X
xj =
zj +
zn−j+1 = 1
j≤h

j≤h

(12)

h<j≤n

xj , zj ≥ 0.
Now this is a linear program with 2h + 1 variables and h + 2 constraints, so by an argument analogous
to Problem 12, we find that every basic feasible solution x1:h , z1:h must share h − 1 zeros. This means
that there is a single index t such that xt > 0, zt > 0; if there were more than one, by the pigeonhole
principle there must be some t′ ≤ h such that xt′ = zt′ = xn−t′ +1 = 0, and lt′ (x, z) = 0. Then
∗
lmin (x, z) = 0; however, by Lemma 1, lmin (x, z) = lmin
> 0, so we have a contradiction.

Expanded PaperInterface specification

∀ {n : ℕ} [inst : NeZero n] {beta : ℝ} {v : GCG24UserItemFairness.Item n → ℝ} {ρ : GCG24UserItemFairness.TypePolicy 3 n}
  {ell : ℝ},
  2 < n →

```
(↑n)⁻¹ < beta →
  beta < 1 / 2 →
    (∀ (j : GCG24UserItemFairness.Item n), 0 < v j) →
      GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v →
        GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρ →
          (∀ (j : GCG24UserItemFairness.Item n),
             GCG24UserItemFairness.OpposingTypes.theorem4Problem11PolicyItemValue beta v ρ j = ell) →
            (∀ (ρ' : GCG24UserItemFairness.TypePolicy 3 n) (ell' : ℝ),
               GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρ' →
                 (∀ (j : GCG24UserItemFairness.Item n),
                    GCG24UserItemFairness.OpposingTypes.theorem4Problem11PolicyItemValue beta v ρ' j = ell') →
                   ell' ≤ ell) →
              GCG24UserItemFairness.ProofBridge.problem11BasicFeasible beta v ρ ell →
                have t : GCG24UserItemFairness.Item n :=
                  GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ;
                ↑t ≤ ↑(GCG24UserItemFairness.OpposingTypes.reverseItem t) ∧
                  (∀ (j : GCG24UserItemFairness.Item n), ↑t < ↑j → (ρ 0) j = 0) ∧
                    (∀ (j : GCG24UserItemFairness.Item n),
                       ↑j < ↑t → (ρ 2) j = 0 ∧ (ρ 2) (GCG24UserItemFairness.OpposingTypes.reverseItem j) = 0) ∧
                      ∀ (s : GCG24UserItemFairness.Item n),
                        (∀ (j : GCG24UserItemFairness.Item n), ↑s < ↑j → (ρ 0) j = 0) →
                          (∀ (j : GCG24UserItemFairness.Item n),
                             ↑j < ↑s →
                               (ρ 2) j = 0 ∧ (ρ 2) (GCG24UserItemFairness.OpposingTypes.reverseItem j) = 0) →
                            ↑t ≤ ↑s
```

Retained governing declarations

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex
- GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy
- GCG24UserItemFairness.OpposingTypes.reverseItem
- GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero
- GCG24UserItemFairness.OpposingTypes.theorem4Problem11PolicyItemValue
- GCG24UserItemFairness.ProofBridge.problem11BasicFeasible
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.UserType

Lean-elaborated claim roles

```
1. parameter: ℕ
2. assumption: NeZero n
3. parameter: ℝ
4. parameter: GCG24UserItemFairness.Item n → ℝ
5. parameter: GCG24UserItemFairness.TypePolicy 3 n
6. parameter: ℝ
7. assumption: 2 < n
8. assumption: (↑n)⁻¹ < beta
9. assumption: beta < 1 / 2
10. assumption: ∀ (j : GCG24UserItemFairness.Item n), 0 < v j
11. assumption: GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v
12. assumption: GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρ
13. assumption: ∀ (j : GCG24UserItemFairness.Item n),
  GCG24UserItemFairness.OpposingTypes.theorem4Problem11PolicyItemValue beta v ρ j = ell
14. assumption: ∀ (ρ' : GCG24UserItemFairness.TypePolicy 3 n) (ell' : ℝ),
  GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρ' →
    (∀ (j : GCG24UserItemFairness.Item n),
       GCG24UserItemFairness.OpposingTypes.theorem4Problem11PolicyItemValue beta v ρ' j = ell') →
      ell' ≤ ell
15. assumption: GCG24UserItemFairness.ProofBridge.problem11BasicFeasible beta v ρ ell
16. conclusion: ↑(GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ) ≤
    ↑(GCG24UserItemFairness.OpposingTypes.reverseItem
       (GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ)) ∧
  (∀ (j : GCG24UserItemFairness.Item n),
     ↑(GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ) < ↑j → (ρ 0) j = 0) ∧
    (∀ (j : GCG24UserItemFairness.Item n),
       ↑j < ↑(GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ) →
         (ρ 2) j = 0 ∧ (ρ 2) (GCG24UserItemFairness.OpposingTypes.reverseItem j) = 0) ∧
      ∀ (s : GCG24UserItemFairness.Item n),
        (∀ (j : GCG24UserItemFairness.Item n), ↑s < ↑j → (ρ 0) j = 0) →
          (∀ (j : GCG24UserItemFairness.Item n),
             ↑j < ↑s → (ρ 2) j = 0 ∧ (ρ 2) (GCG24UserItemFairness.OpposingTypes.reverseItem j) = 0) →
            ↑(GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ) ≤ ↑s
```

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.appendix_e_lemma13_every_optimal_basic_solution_has_pivot_support

Recorded source-to-Spec screening **Verdict:** matches
Reason: The source’s basic-feasible Problem 11 pivot-support structure, including the midpoint bound and zero pattern, is retained in the Spec.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

Review row 15

Source locator: cited publication, lines 3413-3414

Verbatim source input

Lemma 14. Problem 11 has a unique optimal solution.
We can now describe what this solution (x, z) looks like.

Expanded PaperInterface specification

```
∀ {n : ℕ} [NeZero n] {beta : ℝ} {v : GCG24UserItemFairness.Item n → ℝ},
  2 < n →
    (↑n)⁻¹ < beta →
      beta < 1 / 2 →
        (∀ (j : GCG24UserItemFairness.Item n), 0 < v j) →
          GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v →
            ∃ (ρstar : GCG24UserItemFairness.TypePolicy 3 n) (ellstar : ℝ),
              GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρstar ∧
                (∀ (j : GCG24UserItemFairness.Item n),
                   GCG24UserItemFairness.OpposingTypes.theorem4Problem11PolicyItemValue beta v ρstar j = ellstar) ∧
                  (∀ (ρ : GCG24UserItemFairness.TypePolicy 3 n) (ell : ℝ),
                     GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρ →
                       GCG24UserItemFairness.OpposingTypes.theorem4Problem11LPFeasible beta v ρ ell → ell ≤ ellstar) ∧
                    ∀ (ρ : GCG24UserItemFairness.TypePolicy 3 n) (ell : ℝ),
                      GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρ →
                        GCG24UserItemFairness.OpposingTypes.theorem4Problem11LPFeasible beta v ρ ell →
                          (∀ (ρ' : GCG24UserItemFairness.TypePolicy 3 n) (ell' : ℝ),
                             GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρ' →
                               GCG24UserItemFairness.OpposingTypes.theorem4Problem11LPFeasible beta v ρ' ell' →
                                 ell' ≤ ell) →
                            ρ = ρstar ∧ ell = ellstar
```

Retained governing declarations

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex
- GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy
- GCG24UserItemFairness.OpposingTypes.theorem4Problem11LPFeasible
- GCG24UserItemFairness.OpposingTypes.theorem4Problem11PolicyItemValue
- GCG24UserItemFairness.TypePolicy

Lean-elaborated claim roles

1. parameter: ℕ
2. assumption: NeZero n
3. parameter: ℝ
4. parameter: GCG24UserItemFairness.Item n → ℝ
5. assumption: 2 < n
6. assumption: (↑n)⁻¹ < beta
7. assumption: beta < 1 / 2
8. assumption: ∀ (j : GCG24UserItemFairness.Item n), 0 < v j
9. assumption: GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v
10. conclusion: ∃ (ρstar : GCG24UserItemFairness.TypePolicy 3 n) (ellstar : ℝ),
  GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρstar ∧
    (∀ (j : GCG24UserItemFairness.Item n),
       GCG24UserItemFairness.OpposingTypes.theorem4Problem11PolicyItemValue beta v ρstar j = ellstar) ∧
      (∀ (ρ : GCG24UserItemFairness.TypePolicy 3 n) (ell : ℝ),
         GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρ →
           GCG24UserItemFairness.OpposingTypes.theorem4Problem11LPFeasible beta v ρ ell → ell ≤ ellstar) ∧
        ∀ (ρ : GCG24UserItemFairness.TypePolicy 3 n) (ell : ℝ),
          GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρ →
            GCG24UserItemFairness.OpposingTypes.theorem4Problem11LPFeasible beta v ρ ell →
              (∀ (ρ' : GCG24UserItemFairness.TypePolicy 3 n) (ell' : ℝ),
                 GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρ' →
                   GCG24UserItemFairness.OpposingTypes.theorem4Problem11LPFeasible beta v ρ' ell' → ell' ≤ ell) →
                ρ = ρstar ∧ ell = ellstar

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.appendix_e_lemma14_problem11_has_unique_optimal_solution

Recorded source-to-Spec screening **Verdict:** matches
Reason: Lemma 14 explicitly asserts a unique optimum for Problem 11, which is the selected Spec's uniqueness conclusion under the displayed model conditions.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

Review row 16

Source locator: cited publication, lines 3415-3472

Verbatim source input

Lemma 15. Let (x, z) be the optimal solution to Problem 12, and let t be the pivot element of (x, z);
suppose that ⁄= n+1
2 . Define
X 1
Lt :=
.
q
j<t j
Then
λ=
[U+F8F1 delimiter glyph]
[U+F8F4 delimiter glyph]
[U+F8F2 delimiter glyph]0,[U+0010 control character]

zj = 12
[U+F8F4 delimiter glyph]
[U+F8F3 delimiter glyph]

2βqt + 1/2(1 − 2β)
,
1 + qt Lt + 1/2(n − 2t)
j<t

λ
1 − (n − 2t) 1−2β

[U+0011 control character]

,

λ
1−2β ,

j ∈ {t, n − t + 1} .
t<j <n−t+1

Finally,
[U+F8F1 delimiter glyph]
λ
[U+F8F4 delimiter glyph]
[U+F8F2 delimiter glyph] 2βqj ,
λ
xj = 1 − 2β
Lt ,
[U+F8F4 delimiter glyph]
[U+F8F3 delimiter glyph]0,
38

j<t
j=t.
j>t

[form-feed in source extraction]
If t = n+1
2 , then x remains the same, but zt = 1 and
2βqt + (1 − 2β)
λ=
.
1 + qt Lt

Formalized review target

The archival source above is not asserted equivalent to this formalized target.

Appendix E, Lemma 15 corrected target: for the equality-form LP solved in the lemma, the printed noncenter lambda, x, and z coordinate formulas hold; at the center pivot z_t = 1 and lambda = 1 / (1 + L_t).

Archival source anchor: cited publication, lines 3415-3472

Expanded PaperInterface specification

```
∀ {n : ℕ} [inst : NeZero n] {beta : ℝ} {v : GCG24UserItemFairness.Item n → ℝ} {ρ : GCG24UserItemFairness.TypePolicy 3 n}
  {ell : ℝ},
  2 < n →
    (↑n)⁻¹ < beta →
      beta < 1 / 2 →
        (∀ (j : GCG24UserItemFairness.Item n), 0 < v j) →
          GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v →
            GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρ →
              (∀ (j : GCG24UserItemFairness.Item n),
                  GCG24UserItemFairness.OpposingTypes.theorem4Problem11PolicyItemValue beta v ρ j = ell) →
                (∀ (ρ' : GCG24UserItemFairness.TypePolicy 3 n) (ell' : ℝ),
                    GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρ' →
                      GCG24UserItemFairness.OpposingTypes.theorem4Problem11LPFeasible beta v ρ' ell' → ell' ≤ ell) →
                  have t : GCG24UserItemFairness.Item n :=
```

```
 GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ;
↑t ≤ ↑(GCG24UserItemFairness.OpposingTypes.reverseItem t) ∧
 (∀ (j : GCG24UserItemFairness.Item n),
    ↑j < ↑t →
     ((ρ 0) j).toReal =
       ell / (2 * beta * GCG24UserItemFairness.OpposingTypes.pairShare (1 / 2) v j)) ∧
   ((ρ 0) t).toReal =
     1 -
       ell / (2 * beta) * GCG24UserItemFairness.OpposingTypes.paper_lemma15_problem11_leftSum v t ∧
  (∀ (j : GCG24UserItemFairness.Item n), ↑t < ↑j → ((ρ 0) j).toReal = 0) ∧
   (∀ (j : GCG24UserItemFairness.Item n),
      ↑j < ↑t →
       ((ρ 2) j).toReal = 0 ∧
        ((ρ 2) (GCG24UserItemFairness.OpposingTypes.reverseItem j)).toReal = 0) ∧
    (∀ (j : GCG24UserItemFairness.Item n),
       ↑t < ↑j →
        ↑j < ↑(GCG24UserItemFairness.OpposingTypes.reverseItem t) →
         ((ρ 2) j).toReal = ell / (1 - 2 * beta)) ∧
     (↑t < ↑(GCG24UserItemFairness.OpposingTypes.reverseItem t) ∧
        ell =
          (2 * beta * GCG24UserItemFairness.OpposingTypes.pairShare (1 / 2) v t +
             1 / 2 * (1 - 2 * beta)) /
           (1 +
              GCG24UserItemFairness.OpposingTypes.pairShare (1 / 2) v t *
                GCG24UserItemFairness.OpposingTypes.paper_lemma15_problem11_leftSum v t +
             1 / 2 * (↑n - 2 * (↑↑t + 1))) ∧
       ((ρ 2) t).toReal = 1 / 2 * (1 - (↑n - 2 * (↑↑t + 1)) * ell / (1 - 2 * beta)) ∧
        ((ρ 2) (GCG24UserItemFairness.OpposingTypes.reverseItem t)).toReal =
          1 / 2 * (1 - (↑n - 2 * (↑↑t + 1)) * ell / (1 - 2 * beta)) ∨
      ↑t = ↑(GCG24UserItemFairness.OpposingTypes.reverseItem t) ∧
       ((ρ 2) t).toReal = 1 ∧
         ell =
           1 / (1 + GCG24UserItemFairness.OpposingTypes.paper_lemma15_problem11_leftSum v t))
```

Retained governing declarations

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex
- GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy
- GCG24UserItemFairness.OpposingTypes.pairShare
- GCG24UserItemFairness.OpposingTypes.paper_lemma15_problem11_leftSum
- GCG24UserItemFairness.OpposingTypes.reverseItem
- GCG24UserItemFairness.OpposingTypes.theorem4Problem11LPFeasible
- GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero
- GCG24UserItemFairness.OpposingTypes.theorem4Problem11PolicyItemValue
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.UserType

Lean-elaborated claim roles

1. parameter: ℕ
2. assumption: NeZero n
3. parameter: ℝ
4. parameter: GCG24UserItemFairness.Item n → ℝ
5. parameter: GCG24UserItemFairness.TypePolicy 3 n
6. parameter: ℝ
7. assumption: 2 < n
8. assumption: (↑n)⁻¹ < beta
9. assumption: beta < 1 / 2
10. assumption: ∀ (j : GCG24UserItemFairness.Item n), 0 < v j
11. assumption: GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v
12. assumption: GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρ
13. assumption: ∀ (j : GCG24UserItemFairness.Item n),
  GCG24UserItemFairness.OpposingTypes.theorem4Problem11PolicyItemValue beta v ρ j = ell
14. assumption: ∀ (ρ' : GCG24UserItemFairness.TypePolicy 3 n) (ell' : ℝ),
  GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρ' →
    GCG24UserItemFairness.OpposingTypes.theorem4Problem11LPFeasible beta v ρ' ell' → ell' ≤ ell
15. conclusion: ↑(GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ) ≤
    ↑(GCG24UserItemFairness.OpposingTypes.reverseItem
        (GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ)) ∧
  (∀ (j : GCG24UserItemFairness.Item n),
     ↑j < ↑(GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ) →
      ((ρ 0) j).toReal = ell / (2 * beta * GCG24UserItemFairness.OpposingTypes.pairShare (1 / 2) v j)) ∧
   ((ρ 0) (GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ)).toReal =
     1 -
       ell / (2 * beta) *
         GCG24UserItemFairness.OpposingTypes.paper_lemma15_problem11_leftSum v
           (GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ) ∧
    (∀ (j : GCG24UserItemFairness.Item n),
       ↑(GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ) < ↑j → ((ρ 0) j).toReal = 0) ∧


```
```

```
(∀ (j : GCG24UserItemFairness.Item n),
    ↑j < ↑(GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ) →
      ((ρ 2) j).toReal = 0 ∧ ((ρ 2) (GCG24UserItemFairness.OpposingTypes.reverseItem j)).toReal = 0) ∧
  (∀ (j : GCG24UserItemFairness.Item n),
      ↑(GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ) < ↑j →
        ↑j <
          ↑(GCG24UserItemFairness.OpposingTypes.reverseItem
              (GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ)) →
        ((ρ 2) j).toReal = ell / (1 - 2 * beta)) ∧
    (↑(GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ) <
        ↑(GCG24UserItemFairness.OpposingTypes.reverseItem
          (GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ)) ∧
      ell =
        (2 * beta *
            GCG24UserItemFairness.OpposingTypes.pairShare (1 / 2) v
              (GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ) +
          1 / 2 * (1 - 2 * beta)) /
          (1 +
            GCG24UserItemFairness.OpposingTypes.pairShare (1 / 2) v
                (GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ) *
              GCG24UserItemFairness.OpposingTypes.paper_lemma15_problem11_leftSum v
                (GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ) +
            1 / 2 *
              (↑n -
                2 * (↑↑(GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ) + 1))) ∧
      ((ρ 2) (GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ)).toReal =
        1 / 2 *
          (1 -
            (↑n -
                2 *
                  (↑↑(GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ) + 1)) *
              ell /
                (1 - 2 * beta)) ∧
        ((ρ 2)
            (GCG24UserItemFairness.OpposingTypes.reverseItem
              (GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ))).toReal =
          1 / 2 *
            (1 -
              (↑n -
                  2 *
                    (↑↑(GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ) + 1)) *
                ell /
                  (1 - 2 * beta)) ∨
      ↑(GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ) =
        ↑(GCG24UserItemFairness.OpposingTypes.reverseItem
          (GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ)) ∧
       ((ρ 2) (GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ)).toReal = 1 ∧
        ell =
          1 /
            (1 +
              GCG24UserItemFairness.OpposingTypes.paper_lemma15_problem11_leftSum v
                (GCG24UserItemFairness.OpposingTypes.theorem4Problem11LastActiveTypeZero ρ)))
```

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.appendix_e_lemma15_optimal_solution_full_closed_form

Recorded source-to-Spec screening

Verdict: matches_approved_corrected_target
Reason: The archival source formula differs in the recorded Appendix E typo; the expanded Spec matches the separately approval-pinned corrected target, exactly as the user-approved correction record requires.
Reviewer check: ☐ Matches formalized target
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

Review row 17

Source locator: cited publication, lines 3713-3735

Verbatim source input

Lemma 16. If j < n+1
2 , then qj > 1/2; if j > 2 , then qj < 1/2. If j = 2 , qj = 1/2.

Proof. If j < n+1
2 , then vj > vn−j+1 and
qj =

vj
1
1
1
= ;
=
vn−j+1 >
vj + vn−j+1
1+1
2
1 + vj

if j < n+1
2 , then vj < vn−j+1 and a symmetric argument holds.
If j = n+1
2 , then vj = vn−j+1 and thus qj = 1/2.

Expanded PaperInterface specification

```
∀ {n : ℕ} {v : GCG24UserItemFairness.Item n → ℝ} (j : GCG24UserItemFairness.Item n),
  (∀ (j : GCG24UserItemFairness.Item n), 0 < v j) →
    GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v →
      (↑j < ↑(GCG24UserItemFairness.OpposingTypes.reverseItem j) →
         1 / 2 < GCG24UserItemFairness.OpposingTypes.pairShare (1 / 2) v j) ∧
       (↑(GCG24UserItemFairness.OpposingTypes.reverseItem j) < ↑j →
          GCG24UserItemFairness.OpposingTypes.pairShare (1 / 2) v j < 1 / 2) ∧
        (↑j = ↑(GCG24UserItemFairness.OpposingTypes.reverseItem j) →
          GCG24UserItemFairness.OpposingTypes.pairShare (1 / 2) v j = 1 / 2)
```

Retained governing declarations

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex
- GCG24UserItemFairness.OpposingTypes.pairShare
- GCG24UserItemFairness.OpposingTypes.reverseItem

Lean-elaborated claim roles

1. parameter: ℕ
2. parameter: GCG24UserItemFairness.Item n → ℝ
3. parameter: GCG24UserItemFairness.Item n
4. assumption: ∀ (j : GCG24UserItemFairness.Item n), 0 < v j
5. assumption: GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v
6. conclusion: (↑j < ↑(GCG24UserItemFairness.OpposingTypes.reverseItem j) →
   1 / 2 < GCG24UserItemFairness.OpposingTypes.pairShare (1 / 2) v j) ∧
  (↑(GCG24UserItemFairness.OpposingTypes.reverseItem j) < ↑j →
     GCG24UserItemFairness.OpposingTypes.pairShare (1 / 2) v j < 1 / 2) ∧
   (↑j = ↑(GCG24UserItemFairness.OpposingTypes.reverseItem j) →
     GCG24UserItemFairness.OpposingTypes.pairShare (1 / 2) v j = 1 / 2)

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.appendix_e_lemma16_midpoint_order_algebra

Recorded source-to-Spec screening **Verdict:** matches

Reason: Lemma 16's three-way pair-share ordering around the midpoint is preserved by the Spec's left, right, and midpoint algebraic cases.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 18

Source locator: cited publication, lines 3845-3925

Verbatim source input

Lemma 17. Let (x, z) be an optimal solution to Problem 11. For j > n+1
2 , xj = 0.
In particular, if we let h = ⌊ n+1
2 ⌋ indicate the index halfway through the set of items, this means that
we can simplify the problem above to
arg x max
,...,x ,
1

λ

h

z1 ,...,zh ,
λ

subject to lj (x, z) = λ, 1 ≤ j ≤ h
X
X
X
xj =
zj +
zn−j+1 = 1
j≤h

j≤h

(12)

h<j≤n

xj , zj ≥ 0.
Now this is a linear program with 2h + 1 variables and h + 2 constraints, so by an argument analogous
to Problem 12, we find that every basic feasible solution x1:h , z1:h must share h − 1 zeros. This means
that there is a single index t such that xt > 0, zt > 0; if there were more than one, by the pigeonhole
principle there must be some t′ ≤ h such that xt′ = zt′ = xn−t′ +1 = 0, and lt′ (x, z) = 0. Then
∗
lmin (x, z) = 0; however, by Lemma 1, lmin (x, z) = lmin
> 0, so we have a contradiction.
In terms of x, z,
v +vn−j+1
zj
2
vj +vn−j+1
βvj + βvn−j+1 + (1 − 2β)
2
v +v
βvj xj + βvn−j+1 xn−j+1 + (1 − 2β) j n−j+1
zj
2
=
vj +vn−j+1
2

lj (x, z) =

βvj xj + βvn−j+1 xn−j+1 + (1 − 2β) j

[U+0012 control character]
= 2β

vj
vn−j+1
xj +
xn−j+1
vj + vn−j+1
vj + vn−j+1

41

[U+0013 control character]
+ (1 − 2β)zj

[form-feed in source extraction]
v

j
. Note that qj does not depend on β, and qj is decreasing in j. Then
Let qj = vj +vn−j+1

lj (x, y, z) = 2β (qj xj + (1 − qj)yj) + (1 − 2β)zj .
Part 2. Moreover, the set of j such that xj = 0 is contiguous; similarly for zj .

Expanded PaperInterface specification

∀ {n : ℕ} [NeZero n] {beta : ℝ} {v : GCG24UserItemFairness.Item n → ℝ} {ρ : GCG24UserItemFairness.TypePolicy 3 n}
  {ell : ℝ},

```
2 < n →
  (↑n)⁻¹ < beta →
    beta < 1 / 2 →
      (∀ (j : GCG24UserItemFairness.Item n), 0 < v j) →
        GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v →
          GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρ →
            (∀ (j : GCG24UserItemFairness.Item n),
               GCG24UserItemFairness.OpposingTypes.theorem4Problem11PolicyItemValue beta v ρ j = ell) →
              (∀ (ρ' : GCG24UserItemFairness.TypePolicy 3 n) (ell' : ℝ),
                 GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρ' →
                   GCG24UserItemFairness.OpposingTypes.theorem4Problem11LPFeasible beta v ρ' ell' → ell' ≤ ell) →
                ∀ (j : GCG24UserItemFairness.Item n),
                  ↑(GCG24UserItemFairness.OpposingTypes.reverseItem j) < ↑j → (ρ 0) j = 0
```

Retained governing declarations

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex
- GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy
- GCG24UserItemFairness.OpposingTypes.reverseItem
- GCG24UserItemFairness.OpposingTypes.theorem4Problem11LPFeasible
- GCG24UserItemFairness.OpposingTypes.theorem4Problem11PolicyItemValue
- GCG24UserItemFairness.TypePolicy
- GCG24UserItemFairness.UserType

Lean-elaborated claim roles

1. parameter: ℕ
2. assumption: NeZero n
3. parameter: ℝ
4. parameter: GCG24UserItemFairness.Item n → ℝ
5. parameter: GCG24UserItemFairness.TypePolicy 3 n
6. parameter: ℝ
7. assumption: 2 < n
8. assumption: (↑n)⁻¹ < beta
9. assumption: beta < 1 / 2
10. assumption: ∀ (j : GCG24UserItemFairness.Item n), 0 < v j
11. assumption: GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v
12. assumption: GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρ
13. assumption: ∀ (j : GCG24UserItemFairness.Item n),
    GCG24UserItemFairness.OpposingTypes.theorem4Problem11PolicyItemValue beta v ρ j = ell
14. assumption: ∀ (ρ' : GCG24UserItemFairness.TypePolicy 3 n) (ell' : ℝ),
    GCG24UserItemFairness.OpposingTypes.Theorem4MirrorSymmetricPolicy ρ' →
      GCG24UserItemFairness.OpposingTypes.theorem4Problem11LPFeasible beta v ρ' ell' → ell' ≤ ell
15. parameter: GCG24UserItemFairness.Item n
16. assumption: ↑(GCG24UserItemFairness.OpposingTypes.reverseItem j) < ↑j
17. conclusion: (ρ 0) j = 0

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.appendix_e_lemma17_every_optimal_solution_has_no_right_half_support

Recorded source-to-Spec screening **Verdict:** matches

Reason: The source's no-right-half support statement for every optimal Problem 11 solution is the direct support conclusion of the Spec.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Main-text source claims (continued)

Review row 19

Source locator: cited publication, lines 396-397

Verbatim source input

Theorem 3. πU
|l (α) is decreasing in α for 0 < α ≤ 1/2, and increasing in α for 1/2 ≤ α < 1.

Expanded PaperInterface specification

```
∀ {m n : ℕ} [inst : NeZero m] [inst_1 : NeZero n] (S S' : GCG24UserItemFairness.RecommendationModel.SymmetricData m n 2)
  (hTypes : Function.Surjective S.types.toType) (hTypes' : Function.Surjective S'.types.toType) {alpha alpha' : ℝ}
  {v : GCG24UserItemFairness.Item n → ℝ},
  (S.canonicalReductionOfSurjective hTypes).reduced = GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha v →
    (S'.canonicalReductionOfSurjective hTypes').reduced =
      GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha' v →
      0 < alpha →
        alpha < 1 →
          0 < alpha' →
            alpha' < 1 →
              alpha ≤ alpha' →
                alpha ≤ 1 / 2 →
                  alpha' ≤ 1 / 2 →
                    (∀ (j : GCG24UserItemFairness.Item n), 0 < v j) →
                      GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v →
                        S'.model.priceOfFairness ≤ S.model.priceOfFairness
```

Retained governing declarations

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex
- GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel
- GCG24UserItemFairness.RecommendationModel.SymmetricData
- GCG24UserItemFairness.RecommendationModel.SymmetricData.canonicalReductionOfSurjective
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.RecommendationModel.priceOfFairness
- GCG24UserItemFairness.ReductionWitness
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.User

Lean-elaborated claim roles

1. parameter: ℕ
2. parameter: ℕ
3. assumption: NeZero m
4. assumption: NeZero n
5. parameter: GCG24UserItemFairness.RecommendationModel.SymmetricData m n 2
6. parameter: GCG24UserItemFairness.RecommendationModel.SymmetricData m n 2
7. assumption: Function.Surjective S.types.toType
8. assumption: Function.Surjective S'.types.toType
9. parameter: ℝ
10. parameter: ℝ
11. parameter: GCG24UserItemFairness.Item n → ℝ
12. assumption: (S.canonicalReductionOfSurjective hTypes).reduced = GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha v
13. assumption: (S'.canonicalReductionOfSurjective hTypes').reduced = GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha' v
14. assumption: 0 < alpha
15. assumption: alpha < 1
16. assumption: 0 < alpha'
17. assumption: alpha' < 1
18. assumption: alpha ≤ alpha'
19. assumption: alpha ≤ 1 / 2
20. assumption: alpha' ≤ 1 / 2
21. assumption: ∀ (j : GCG24UserItemFairness.Item n), 0 < v j
22. assumption: GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v
23. conclusion: Real.le† S'.model.priceOfFairness S.model.priceOfFairness

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.theorem3_price_decreases_first_half

Recorded source-to-Spec screening **Verdict:** matches
Reason: This is Theorem 3’s first displayed clause: price of fairness decreases on the alpha interval from zero through one half.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

Review row 20

Source locator: cited publication, lines 396-397

Verbatim source input

Theorem 3. πU
|l (α) is decreasing in α for 0 < α ≤ 1/2, and increasing in α for 1/2 ≤ α < 1.

Expanded PaperInterface specification

```
∀ {m n : ℕ} [inst : NeZero m] [inst_1 : NeZero n] (S S' : GCG24UserItemFairness.RecommendationModel.SymmetricData m n 2)
  (hTypes : Function.Surjective S.types.toType) (hTypes' : Function.Surjective S'.types.toType) {alpha alpha' : ℝ}
  {v : GCG24UserItemFairness.Item n → ℝ},
  (S.canonicalReductionOfSurjective hTypes).reduced = GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha v →
    (S'.canonicalReductionOfSurjective hTypes').reduced =
      GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha' v →
    0 < alpha →
      alpha < 1 →
        0 < alpha' →
          alpha' < 1 →
            alpha ≤ alpha' →
              1 / 2 ≤ alpha →
                1 / 2 ≤ alpha' →
                  (∀ (j : GCG24UserItemFairness.Item n), 0 < v j) →
                    GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v →
                      S.model.priceOfFairness ≤ S'.model.priceOfFairness
```

Retained governing declarations

- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex
- GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel
- GCG24UserItemFairness.RecommendationModel.SymmetricData
- GCG24UserItemFairness.RecommendationModel.SymmetricData.canonicalReductionOfSurjective
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.RecommendationModel.priceOfFairness
- GCG24UserItemFairness.ReductionWitness
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.User

Lean-elaborated claim roles

1. parameter: ℕ
2. parameter: ℕ
3. assumption: NeZero m
4. assumption: NeZero n
5. parameter: GCG24UserItemFairness.RecommendationModel.SymmetricData m n 2
6. parameter: GCG24UserItemFairness.RecommendationModel.SymmetricData m n 2
7. assumption: Function.Surjective S.types.toType
8. assumption: Function.Surjective S'.types.toType
9. parameter: ℝ
10. parameter: ℝ
11. parameter: GCG24UserItemFairness.Item n → ℝ
12. assumption: (S.canonicalReductionOfSurjective hTypes).reduced = GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha v
13. assumption: (S'.canonicalReductionOfSurjective hTypes').reduced = GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel alpha' v
14. assumption: 0 < alpha
15. assumption: alpha < 1
16. assumption: 0 < alpha'
17. assumption: alpha' < 1
18. assumption: alpha ≤ alpha'
19. assumption: 1 / 2 ≤ alpha
20. assumption: 1 / 2 ≤ alpha'
21. assumption: ∀ (j : GCG24UserItemFairness.Item n), 0 < v j
22. assumption: GCG24UserItemFairness.OpposingTypes.StrictlyDecreasingByIndex v
23. conclusion: Real.le† S.model.priceOfFairness S'.model.priceOfFairness

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.theorem3_price_increases_second_half

Recorded source-to-Spec screening **Verdict:** matches
Reason: This is Theorem 3’s second displayed clause: price of fairness increases on the alpha interval from one half to one.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

Review row 21

Source locator: cited publication, lines 456-476

Verbatim source input

Theorem 4. If β > n1 and w and ŵ are as described above, then fairness constraints can arbitrarily worsen the price of misestimation.
• The price of misestimation without fairness constraints is low: for all {vj }, there is a
∗
recommendation policy ρ̂ that solves the misestimated problem Umin
(0, ŵ) so that
M
πU
(0, w, ŵ) ≤

1
.
2

• The price of misestimation with fairness constraints can be arbitrarily large: ∀ϵ, there exists
∗
{vj } and a recommendation policy ρ̂ that solves the problem Umin
(1, ŵ) such that
M
πU
(1, w, ŵ) > 1 − ϵ.

[Next verbatim source excerpt]

fairness desiderata as follows. At the extremes, the platform could choose a maximally fair solution
∗
for one side, ignoring the other. Denote the optimal user fair utility as Umin
(w) := maxρ Umin (ρ, w)
(achieved by giving each user their favorite item deterministically), and the optimal item fair utility
∗
as Imin
(w) := maxρ Imin (ρ, w).
Finally, we cast the two-sided fair optimization – for the optimal γ-constrained user fair solution – as
∗
Umin
(γ, w) = max Umin (ρ, w)
ρ

(1)

∗
subject to Imin (ρ, w) ≥ γImin
(w),

i.e., we maximize the minimum normalized user utility, subject to the minimum normalized item utility being at least a fraction γ of the optimal item fair solution.
Research questions: price of fairness and misestimation. We can now define the price of item

[Next verbatim source excerpt]

Similarly, we investigate the price of misestimation. Let ŵij denote the platform's estimate of the utility of recommending i to j; let ρ̂(γ) be a policy that solves the optimization problem above with
3

[form-feed in source extraction]
∗
misestimated utilities, that is, ρ̂(γ) attains Umin
(γ, ŵ).1 Then we define the price of misestimation
M
on user utility ("price of misestimation") πU as
M
(γ, w, ŵ) :=
πU

∗
Umin
(γ, w) − Umin (ρ̂(γ), w)
,
∗ (γ, w)
Umin

that is, the decrease in true user fairness due to optimizing with estimated utilities. In particular, in Section 5 we will examine whether item fairness exacerbates the price of misestimation by comparing
M
M
πU
(0, w, ŵ) and πU
(1, w, ŵ), particularly in the case of cold start users.

[Next verbatim source excerpt]

[form-feed in source extraction]
We now formalize and generalize this example, when the level of heterogeneity in the population can be captured by a single parameter. Consider the following utility matrix structure. Let v1 > v2 > ... > vn > 0. Suppose that there are two user types: a user i either has utility wij = vj or wij = vn−j+1 , and say that user i is of type 1 or 2 respectively. In words, the two types have opposite

preferences, but the preferences can otherwise be generic and be for any number of items. Now, the direct solution by symmetry for the example no longer directly holds – the items in the middle, not necessarily preferred by either user type, may be binding in terms of item fairness constraints.
Let α be the proportion of type 1 users in the population, out of a fixed population of m users.

[Next verbatim source excerpt]

We now formalize the above argument, building on the analysis in the previous section. As in Section 4, suppose that there are 2 types of users, with opposing preferences (i.e., with values v1 , v2 ..., vn and vn , vn−1 , ..., v1 , respectively) – and the platform has correctly estimated these preferences. However, now, these two types only make up a proportion β of the population each. Now, we suppose that there is a fraction 1 − β of the user population who are “new” users. We assume that these users are drawn from the same distribution as the remaining users, but the platform does not know their preferences. It thus constructs a prior by averaging over the known users’ preferences
v +v
– (mis)estimating the users’ utility for each item j as j n−j+1
.

Settled source reading For Theorem 4, the normalized three-type construction uses masses beta, beta, and 1 - 2 beta; the stated beta range keeps all three masses positive.

Expanded PaperInterface specification

```
∀ {m n : ℕ} [inst : NeZero m] [inst_1 : NeZero n] (E : GCG24UserItemFairness.EstimatedRecommendationModel m n)
  (Strue : GCG24UserItemFairness.RecommendationModel.SymmetricData m n 2)
  (Sest : GCG24UserItemFairness.RecommendationModel.SymmetricData m n 3)
  (hTypesTrue : Function.Surjective Strue.types.toType) (hTypesEst : Function.Surjective Sest.types.toType) {beta : ℝ}
  {v : GCG24UserItemFairness.Item n → ℝ},
  GCG24UserItemFairness.assumption_theorem4_true_model_reduction E Strue →
    GCG24UserItemFairness.assumption_theorem4_estimated_model_reduction E Sest →
      (Strue.canonicalReductionOfSurjective hTypesTrue).reduced =
          GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel (1 / 2) v →
        (Sest.canonicalReductionOfSurjective hTypesEst).reduced =
            GCG24UserItemFairness.OpposingTypes.theorem4EstimatedReducedModel beta v →
          (∀ (u : GCG24UserItemFairness.User m), Sest.types.toType u = 0 → Strue.types.toType u = 0) →
            (∀ (u : GCG24UserItemFairness.User m), Sest.types.toType u = 1 → Strue.types.toType u = 1) →
              (↑n)⁻¹ < beta →
                GCG24UserItemFairness.assumption_theorem4_universal_value_vector v →
                  ∃ (ρ : GCG24UserItemFairness.Policy m n),
                    E.SolvesEstimatedProblem 0 ρ ∧ E.priceOfMisestimation 0 ρ ≤ 1 / 2
```

Retained governing declarations

- GCG24UserItemFairness.EstimatedRecommendationModel
- GCG24UserItemFairness.EstimatedRecommendationModel.SolvesEstimatedProblem
- GCG24UserItemFairness.EstimatedRecommendationModel.priceOfMisestimation
- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.theorem4EstimatedReducedModel
- GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel.SymmetricData
- GCG24UserItemFairness.RecommendationModel.SymmetricData.canonicalReductionOfSurjective
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.ReductionWitness
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.User
- GCG24UserItemFairness.assumption_theorem4_estimated_model_reduction
- GCG24UserItemFairness.assumption_theorem4_true_model_reduction
- GCG24UserItemFairness.assumption_theorem4_universal_value_vector

Lean-elaborated claim roles

1. parameter: ℕ
2. parameter: ℕ
3. assumption: NeZero m
4. assumption: NeZero n
5. parameter: GCG24UserItemFairness.EstimatedRecommendationModel m n
6. parameter: GCG24UserItemFairness.RecommendationModel.SymmetricData m n 2
7. parameter: GCG24UserItemFairness.RecommendationModel.SymmetricData m n 3
8. assumption: Function.Surjective Strue.types.toType
9. assumption: Function.Surjective Sest.types.toType
10. parameter: ℝ
11. parameter: GCG24UserItemFairness.Item n → ℝ
12. assumption: GCG24UserItemFairness.assumption_theorem4_true_model_reduction E Strue
13. assumption: GCG24UserItemFairness.assumption_theorem4_estimated_model_reduction E Sest
14. assumption: (Strue.canonicalReductionOfSurjective hTypesTrue).reduced =
  GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel (1 / 2) v
15. assumption: (Sest.canonicalReductionOfSurjective hTypesEst).reduced =
  GCG24UserItemFairness.OpposingTypes.theorem4EstimatedReducedModel beta v
16. assumption: ∀ (u : GCG24UserItemFairness.User m), Sest.types.toType u = 0 → Strue.types.toType u = 0
17. assumption: ∀ (u : GCG24UserItemFairness.User m), Sest.types.toType u = 1 → Strue.types.toType u = 1
18. assumption: (↑n)⁻¹ < beta
19. assumption: GCG24UserItemFairness.assumption_theorem4_universal_value_vector v
20. conclusion: ∃ (ρ : GCG24UserItemFairness.Policy m n), E.SolvesEstimatedProblem 0 ρ ∧ E.priceOfMisestimation 0 ρ ≤ 1 / 2

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.theorem4_misestimation_without_fairness_universal

Recorded source-to-Spec screening **Verdict:** matches

Reason: For selected source_item theorem4_misestimation_without_fairness_universal, the source's 'for all {v_j}' ranges over the preceding v1 > ... > vn > 0 opposing-preference family, and the expanded target correspondingly takes an arbitrary v satisfying positivity and strict decrease. Its reduction equalities, surjective type maps, and hknown0/ hknown1 premises instantiate exactly the source's opposing true rows, known beta masses, and averaged cold-start estimated row, using the approved normalized cold-start mass 1 - 2 * beta. It then existentially supplies a policy solving the estimated gamma = 0 problem and concludes the same price-of-misestimation bound <= 1 / 2. The displayed optimization and price dependencies match the source's U_min^*(gamma, w-hat) and normalized true-utility loss formulas. Although beta < 1 / 2 is not a separate binder in this row, the surjective three-type realization and equality of its normalized type weights to beta, beta, and 1 - 2 * beta force the corrected cold-start weight to be positive, so the target does not admit models outside the approved domain. Positive utilities also make the auxiliary zero-denominator and zero-base branches immaterial.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Appendix and supplement source claims (continued)

Review row 22

Source locator: cited publication, lines 1759-1800

Verbatim source input

Lemma 2. Let F denote the feasible set of Problem 1,
F ≜ arg max
Imin (ρ) ≜ arg max
min Ij (ρ).
m
m
ρ∈Δn−1

ρ∈Δn−1

Then F is the solution set of the linear program
F = arg max
m

ρ∈Δn−1 ,λ

j

(3)

λ
(4)

subject to Ij (ρ) = λ

∀j.

Now, condition (iii) on S implies that there is a unique ρ∗ in S ∩ F. Thus
ρ∗ = F ∩ S = arg
max
λ
m
ρ∈Δn−1 ∩S,λ

.
(5)
subject to Ij (ρ) = λ ∀j,
We can add S as a constraint in Problem 4 because we know that there is at least one solution to Problem 4 in S – namely, ρ∗ – so this additional constraint will not change the optimal objective value, but will constrain the set of optimal solutions to be in S, as desired.
By condition (ii), ρ∗ is an optimal solution for Problem 1; by condition (i), S can be described by a finite set of linear constraints, and thus Problem 5 is a linear program; call this L.

Expanded PaperInterface specification

```
∀ {m n : ℕ} [NeZero m] [inst : NeZero n] (W : GCG24UserItemFairness.RecommendationModel m n),
  W.Positive →
    ∀ (rho : GCG24UserItemFairness.Policy m n),
      W.itemFairness rho = W.optimalItemFairness ↔
        ∃ (ell : ℝ),
          W.itemFairnessEqualityLPFeasible rho ell ∧
            ∀ (rho' : GCG24UserItemFairness.Policy m n) (ell' : ℝ),
              W.itemFairnessEqualityLPFeasible rho' ell' → ell' ≤ ell
```

Retained governing declarations

- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel
- GCG24UserItemFairness.RecommendationModel.Positive
- GCG24UserItemFairness.RecommendationModel.itemFairness
- GCG24UserItemFairness.RecommendationModel.itemFairnessEqualityLPFeasible
- GCG24UserItemFairness.RecommendationModel.optimalItemFairness

Lean-elaborated claim roles

1. parameter: ℕ
2. parameter: ℕ
3. assumption: NeZero m
4. assumption: NeZero n
5. parameter: GCG24UserItemFairness.RecommendationModel m n
6. assumption: W.Positive
7. parameter: GCG24UserItemFairness.Policy m n
8. conclusion: W.itemFairness rho = W.optimalItemFairness ↔
   ∃ (ell : ℝ),
     W.itemFairnessEqualityLPFeasible rho ell ∧
       ∀ (rho' : GCG24UserItemFairness.Policy m n) (ell' : ℝ), W.itemFairnessEqualityLPFeasible rho' ell' → ell' ≤ ell

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.appendix_c_lemma2_item_fairness_equality_lp_solution_set

Recorded source-to-Spec screening **Verdict:** matches

Reason: The source identifies the item-fairness optimum with the equality linear-program solution set; the Spec gives that equivalence and its maximizing equality-feasibility condition.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Main-text source claims (continued 2)

Review row 23

Source locator: cited publication, lines 456-476

Verbatim source input

Theorem 4. If β > n1 and w and ŵ are as described above, then fairness constraints can arbitrarily worsen the price of misestimation.
• The price of misestimation without fairness constraints is low: for all {vj }, there is a
∗
recommendation policy ρ̂ that solves the misestimated problem Umin
(0, ŵ) so that
M
πU
(0, w, ŵ) ≤

1
.
2

• The price of misestimation with fairness constraints can be arbitrarily large: ∀ϵ, there exists
∗
{vj } and a recommendation policy ρ̂ that solves the problem Umin
(1, ŵ) such that
M
πU
(1, w, ŵ) > 1 − ϵ.

[Next verbatim source excerpt]

fairness desiderata as follows. At the extremes, the platform could choose a maximally fair solution
∗
for one side, ignoring the other. Denote the optimal user fair utility as Umin
(w) := maxρ Umin (ρ, w)
(achieved by giving each user their favorite item deterministically), and the optimal item fair utility
∗
as Imin
(w) := maxρ Imin (ρ, w).
Finally, we cast the two-sided fair optimization – for the optimal γ-constrained user fair solution – as
∗
Umin
(γ, w) = max Umin (ρ, w)
ρ

(1)

∗
subject to Imin (ρ, w) ≥ γImin
(w),

i.e., we maximize the minimum normalized user utility, subject to the minimum normalized item utility being at least a fraction γ of the optimal item fair solution.
Research questions: price of fairness and misestimation. We can now define the price of item

[Next verbatim source excerpt]

Similarly, we investigate the price of misestimation. Let ŵij denote the platform's estimate of the utility of recommending i to j; let ρ̂(γ) be a policy that solves the optimization problem above with
3

[form-feed in source extraction]
∗
misestimated utilities, that is, ρ̂(γ) attains Umin
(γ, ŵ).1 Then we define the price of misestimation
M
on user utility ("price of misestimation") πU as
M
(γ, w, ŵ) :=
πU

∗
Umin
(γ, w) − Umin (ρ̂(γ), w)
,
∗ (γ, w)
Umin

that is, the decrease in true user fairness due to optimizing with estimated utilities. In particular, in Section 5 we will examine whether item fairness exacerbates the price of misestimation by comparing
M
M
πU
(0, w, ŵ) and πU
(1, w, ŵ), particularly in the case of cold start users.

[Next verbatim source excerpt]

[form-feed in source extraction]
We now formalize and generalize this example, when the level of heterogeneity in the population

can be captured by a single parameter. Consider the following utility matrix structure. Let v1 > v2 > ... > vn > 0. Suppose that there are two user types: a user i either has utility wij = vj or wij = vn−j+1 , and say that user i is of type 1 or 2 respectively. In words, the two types have opposite preferences, but the preferences can otherwise be generic and be for any number of items. Now, the direct solution by symmetry for the example no longer directly holds – the items in the middle, not necessarily preferred by either user type, may be binding in terms of item fairness constraints. Let α be the proportion of type 1 users in the population, out of a fixed population of m users.

[Next verbatim source excerpt]

We now formalize the above argument, building on the analysis in the previous section. As in Section 4, suppose that there are 2 types of users, with opposing preferences (i.e., with values v1 , v2 ..., vn and vn , vn−1 , ..., v1 , respectively) – and the platform has correctly estimated these preferences. However, now, these two types only make up a proportion β of the population each. Now, we suppose that there is a fraction 1 − β of the user population who are “new” users. We assume that these users are drawn from the same distribution as the remaining users, but the platform does not know their preferences. It thus constructs a prior by averaging over the known users’ preferences v +v
– (mis)estimating the users’ utility for each item j as j n−j+1
.

Settled source reading For Theorem 4, the normalized three-type construction uses masses beta, beta, and 1 - 2 beta; the stated beta range keeps all three masses positive.

Expanded PaperInterface specification

```
∀ {m n : ℕ} [inst : NeZero m] [inst_1 : NeZero n] {beta eps : ℝ},
  GCG24UserItemFairness.assumption_theorem4_parameter_domain n beta eps →
    ∃ (v : GCG24UserItemFairness.Item n → ℝ),
      GCG24UserItemFairness.assumption_theorem4_universal_value_vector v ∧
        ∀ (E : GCG24UserItemFairness.EstimatedRecommendationModel m n)
          (Strue : GCG24UserItemFairness.RecommendationModel.SymmetricData m n 2)
          (Sest : GCG24UserItemFairness.RecommendationModel.SymmetricData m n 3)
          (hTypesTrue : Function.Surjective Strue.types.toType) (hTypesEst : Function.Surjective Sest.types.toType),
          GCG24UserItemFairness.assumption_theorem4_true_model_reduction E Strue →
            GCG24UserItemFairness.assumption_theorem4_estimated_model_reduction E Sest →
              (Strue.canonicalReductionOfSurjective hTypesTrue).reduced =
                GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel (1 / 2) v →
                (Sest.canonicalReductionOfSurjective hTypesEst).reduced =
                  GCG24UserItemFairness.OpposingTypes.theorem4EstimatedReducedModel beta v →
                  (∀ (u : GCG24UserItemFairness.User m), Sest.types.toType u = 0 → Strue.types.toType u = 0) →
                    (∀ (u : GCG24UserItemFairness.User m), Sest.types.toType u = 1 → Strue.types.toType u = 1) →
                      ∃ (ρ : GCG24UserItemFairness.Policy m n),
                        E.SolvesEstimatedProblem 1 ρ ∧ 1 - eps < E.priceOfMisestimation 1 ρ
```

Retained governing declarations

- GCG24UserItemFairness.EstimatedRecommendationModel
- GCG24UserItemFairness.EstimatedRecommendationModel.SolvesEstimatedProblem
- GCG24UserItemFairness.EstimatedRecommendationModel.priceOfMisestimation
- GCG24UserItemFairness.Item
- GCG24UserItemFairness.OpposingTypes.theorem4EstimatedReducedModel
- GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel
- GCG24UserItemFairness.Policy
- GCG24UserItemFairness.RecommendationModel.SymmetricData
- GCG24UserItemFairness.RecommendationModel.SymmetricData.canonicalReductionOfSurjective
- GCG24UserItemFairness.RecommendationModel.UserTypeAssignment
- GCG24UserItemFairness.ReductionWitness
- GCG24UserItemFairness.TypeWeightedRecommendationModel
- GCG24UserItemFairness.User
- GCG24UserItemFairness.assumption_theorem4_estimated_model_reduction
- GCG24UserItemFairness.assumption_theorem4_parameter_domain
- GCG24UserItemFairness.assumption_theorem4_true_model_reduction
- GCG24UserItemFairness.assumption_theorem4_universal_value_vector

Lean-elaborated claim roles

1. parameter: ℕ
2. parameter: ℕ
3. assumption: NeZero m
4. assumption: NeZero n
5. parameter: ℝ
6. parameter: ℝ
7. assumption: GCG24UserItemFairness.assumption_theorem4_parameter_domain n beta eps
8. conclusion: ∃ (v : GCG24UserItemFairness.Item n → ℝ),
  GCG24UserItemFairness.assumption_theorem4_universal_value_vector v ∧
    ∀ (E : GCG24UserItemFairness.EstimatedRecommendationModel m n)
      (Strue : GCG24UserItemFairness.RecommendationModel.SymmetricData m n 2)
      (Sest : GCG24UserItemFairness.RecommendationModel.SymmetricData m n 3)
      (hTypesTrue : Function.Surjective Strue.types.toType) (hTypesEst : Function.Surjective Sest.types.toType),
      GCG24UserItemFairness.assumption_theorem4_true_model_reduction E Strue →
        GCG24UserItemFairness.assumption_theorem4_estimated_model_reduction E Sest →
          (Strue.canonicalReductionOfSurjective hTypesTrue).reduced =
              GCG24UserItemFairness.OpposingTypes.twoTypeReducedModel (1 / 2) v →
            (Sest.canonicalReductionOfSurjective hTypesEst).reduced =
                GCG24UserItemFairness.OpposingTypes.theorem4EstimatedReducedModel beta v →
              (∀ (u : GCG24UserItemFairness.User m), Sest.types.toType u = 0 → Strue.types.toType u = 0) →
                (∀ (u : GCG24UserItemFairness.User m), Sest.types.toType u = 1 → Strue.types.toType u = 1) →
                  ∃ (ρ : GCG24UserItemFairness.Policy m n),
                    E.SolvesEstimatedProblem 1 ρ ∧ 1 - eps < E.priceOfMisestimation 1 ρ

Lean proof endpoint (built separately)

GCG24UserItemFairness.PaperInterface.theorem4_misestimation_tradeoff

Recorded source-to-Spec screening **Verdict:** matches

Reason: For selected source_item theorem4_misestimation_tradeoff, GCG24UserItemFairness.txt:456-476 states that for every tolerance eps there exist values {v_j} and a policy solving the estimated gamma = 1 problem whose price of misestimation exceeds 1 - eps. The bound source excerpts fix v1 > ... > vn > 0, the two opposing true utility rows, two known groups of mass beta, the cold-start averaged estimated row, the gamma-constrained max-min objective, and price as normalized loss in true minimum user utility. The expanded target preserves the binder order and conclusion: under 0 < eps, 1 / n < beta, and beta < 1 / 2 it chooses a positive strictly decreasing v and, for every finite realization of the described true and estimated models, produces an estimated-optimal gamma = 1 policy with 1 - eps < priceOfMisestimation. GCG24-T4-POSITIVE-TOLERANCE-01 supplies the source-faithful 0 < eps reading from the Appendix E requirement v2 < (v1 / n) * eps; GCG24-T4-POPULATION-MASS-01 governs the internally normalized beta, beta, 1 - 2 * beta population weights. The exact reductions, type-weight normalization, known-type implications, PMF policy, fairness extrema, and normalized-loss dependency match those source conditions. Positive values and nonempty finite domains make the displayed zero-denominator and zero-base branches unreachable, so they do not alter the claim.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation