Human review packet: GS62CollegeAdmissions

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

Paper: GS62CollegeAdmissions
Paper id: GS62CollegeAdmissions
Source version: American Mathematical Monthly 69(1), January 1962, printed pages 9-15
Source record: audit/paper_statement_map.json
Rows: 6
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 GS62CollegeAdmissions --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

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.FiniteChoice.linearTopQChoice

Declaration location: reusable library

Lean declaration body

```
type:
{α : Type u_1} → [DecidableEq α] → [LinearOrder α] → ℕ → Finset α → Finset α

value:
fun {α : Type u_1} [DecidableEq α] [LinearOrder α] (q : ℕ) (a : Finset α) =>
  if h : q ≤ a.card then
    Finset.image
      (fun (i : Fin q) =>
        ↑((a.orderIsoOfFin (AppliedModelingLib.FiniteChoice.linearTopQChoice._proof_1 a))
            (↑i, AppliedModelingLib.FiniteChoice.linearTopQChoice._proof_2 q a h i)))
      Finset.univ
  else a
```

AppliedModelingLib.Matching.Assignment

Declaration location: reusable library

Lean declaration body

```
field m_match:
{M : Type u_1} → {W : Type u_2} → AppliedModelingLib.Matching.Assignment M W → M → Option W

field w_match:
{M : Type u_1} → {W : Type u_2} → AppliedModelingLib.Matching.Assignment M W → W → Option M

field consistent_m:
∀ {M : Type u_1} {W : Type u_2} (self : AppliedModelingLib.Matching.Assignment M W) (m : M) (w : W),
  self.m_match m = some w ↔ self.w_match w = some m
```

AppliedModelingLib.Matching.ManyToOne.valApplicant

Declaration location: reusable library

Lean declaration body

```
type:
{Applicants : Type u_1} → {Colleges : Type u_2} → (Applicants → Colleges → ℝ) → Applicants → Option Colleges → ℝ

value:
fun {Applicants : Type u_1} {Colleges : Type u_2} (val_applicant : Applicants → Colleges → ℝ) (a : Applicants)
    (c : Option Colleges) =>
  match c with
  | none => 0
  | some c' => val_applicant a c'
```

AppliedModelingLib.Matching.ManyToOneAssignment

Declaration location: reusable library

Lean declaration body

```
field app_match:
{Applicants : Type u_1} →
  {Colleges : Type u_2} →
    AppliedModelingLib.Matching.ManyToOneAssignment Applicants Colleges → Applicants → Option Colleges

field college_roster:
{Applicants : Type u_1} →
  {Colleges : Type u_2} →
    AppliedModelingLib.Matching.ManyToOneAssignment Applicants Colleges → Colleges → Finset Applicants

field consistent:
∀ {Applicants : Type u_1} {Colleges : Type u_2}
  (self : AppliedModelingLib.Matching.ManyToOneAssignment Applicants Colleges) (a : Applicants) (c : Colleges),
  self.app_match a = some c ↔ a ∈ self.college_roster c
```

AppliedModelingLib.Matching.valM

Declaration location: reusable library

Lean declaration body

type:
```
{M : Type u_1} → {W : Type u_2} → (M → W → ℝ) → M → Option W → ℝ
```

value:
```
fun {M : Type u_1} {W : Type u_2} (val : M → W → ℝ) (m : M) (w : Option W) =>
  match w with
  | none => 0
  | some w' => val m w'
```

AppliedModelingLib.Matching.valW

Declaration location: reusable library

Lean declaration body

type:
```
{M : Type u_1} → {W : Type u_2} → (W → M → ℝ) → W → Option M → ℝ
```

value:
```
fun {M : Type u_1} {W : Type u_2} (val : W → M → ℝ) (w : W) (m : Option M) =>
  match m with
  | none => 0
  | some m' => val w m'
```

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState

Declaration location: paper-local

Lean declaration body

field applications:
```
{Applicants : Type u_3} →
  {Colleges : Type u_4} →
    [inst : DecidableEq Applicants] →
      GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges →
        Colleges → Finset Applicants
```

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.collegeOrder

Declaration location: paper-local

Lean declaration body

type:
```
{Applicants : Type u_1} →
  {Colleges : Type u_2} →
    (val_college : Colleges → Applicants → ℝ) →
      (∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a') →
        Colleges → LinearOrder Applicants
```

value:
```
fun {Applicants : Type u_1} {Colleges : Type u_2} (val_college : Colleges → Applicants → ℝ)
    (hcollegeStrict : ∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a')
    (c : Colleges) =>
  LinearOrder.lift' (fun (a : Applicants) => OrderDual.toDual (val_college c a))
    fun {{a b : Applicants}}
      (h :
        (fun (a : Applicants) => OrderDual.toDual (val_college c a)) a =
          (fun (a : Applicants) => OrderDual.toDual (val_college c a)) b) =>
    hcollegeStrict c a b h
```

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.collegeTopQ

Declaration location: paper-local

Lean declaration body

type:
```
{Applicants : Type u_1} →
  {Colleges : Type u_2} →
    [DecidableEq Applicants] →
      (Colleges → ℕ) →
        (val_college : Colleges → Applicants → ℝ) →
          (∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a') →
            Colleges → Finset Applicants → Finset Applicants
```

value:
```
fun {Applicants : Type u_1} {Colleges : Type u_2} [DecidableEq Applicants] (quota : Colleges → ℕ)
    (val_college : Colleges → Applicants → ℝ)
    (hcollegeStrict : ∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a') (c : Colleges)
    (pool : Finset Applicants) =>
  AppliedModelingLib.FiniteChoice.linearTopQChoice (quota c) pool
```

Direct retained dependencies

- AppliedModelingLib.FiniteChoice.linearTopQChoice
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.collegeOrder

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.initialSourceState

Declaration location: paper-local

Lean declaration body

```
type:
{Applicants : Type u_1} →
  {Colleges : Type u_2} →
    [inst : DecidableEq Applicants] →
      GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges

value:
fun {Applicants : Type u_1} {Colleges : Type u_2} [DecidableEq Applicants] =>
  { applications := fun (x : Colleges) => ∅ }
```

Direct retained dependencies

- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceEligible

Declaration location: paper-local

Lean declaration body

```
type:
{Applicants : Type u_1} →
  {Colleges : Type u_2} → (Applicants → Colleges → ℝ) → (Colleges → Applicants → ℝ) → Applicants → Colleges → Prop

value:
fun {Applicants : Type u_1} {Colleges : Type u_2} (val_applicant : Applicants → Colleges → ℝ)
    (val_college : Colleges → Applicants → ℝ) (a : Applicants) (c : Colleges) =>
  0 < val_applicant a c ∧ 0 < val_college c a
```

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.untriedEligibleColleges

Declaration location: paper-local

Lean declaration body

```
type:
{Applicants : Type u_1} →
  {Colleges : Type u_2} →
    [Fintype Colleges] →
      [inst : DecidableEq Applicants] →
        (Applicants → Colleges → ℝ) →
          (Colleges → Applicants → ℝ) →
            GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges →
              Applicants → Finset Colleges

value:
fun {Applicants : Type u_1} {Colleges : Type u_2} [Fintype Colleges] [DecidableEq Applicants]
    (val_applicant : Applicants → Colleges → ℝ) (val_college : Colleges → Applicants → ℝ)
    (s : GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges)
    (a : Applicants) =>
  {c : Colleges |
    GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceEligible val_applicant val_college a c ∧
      a ∉ s.applications c}
```

Direct retained dependencies

- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceEligible

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.untriedEligiblePairs

Declaration location: paper-local

Lean declaration body

```
type:
{Applicants : Type u_1} →
  {Colleges : Type u_2} →
    [Fintype Applicants] →
      [Fintype Colleges] →
        [inst : DecidableEq Applicants] →
          (Applicants → Colleges → ℝ) →
            (Colleges → Applicants → ℝ) →
              GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges →
                Finset (Applicants × Colleges)

value:
fun {Applicants : Type u_1} {Colleges : Type u_2} [Fintype Applicants] [Fintype Colleges] [DecidableEq Applicants]
    (val_applicant : Applicants → Colleges → ℝ) (val_college : Colleges → Applicants → ℝ)
    (s : GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges) =>
  {p : Applicants × Colleges |
    GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceEligible val_applicant val_college p.1 p.2 ∧
      p.1 ∉ s.applications p.2}
```

Direct retained dependencies

- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceEligible

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.waitingList

Declaration location: paper-local

Lean declaration body

```
type:
{Applicants : Type u_1} →
  {Colleges : Type u_2} →
    [inst : DecidableEq Applicants] →
      (Colleges → ℕ) →
        (val_college : Colleges → Applicants → ℝ) →
          (∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a') →
            GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges →
              Colleges → Finset Applicants

value:
fun {Applicants : Type u_1} {Colleges : Type u_2} [DecidableEq Applicants] (quota : Colleges → ℕ)
    (val_college : Colleges → Applicants → ℝ)
    (hcollegeStrict : ∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a')
    (s : GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges)
    (c : Colleges) =>
  GS62CollegeAdmissions.ExactCollegeBatchedProcedure.collegeTopQ quota val_college hcollegeStrict c (s.applications c)
```

Direct retained dependencies

- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.collegeTopQ

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceStateInvariant

Declaration location: paper-local

Lean declaration body

```
type:
{Applicants : Type u_1} →
  {Colleges : Type u_2} →
    [inst : DecidableEq Applicants] →
      (Colleges → ℕ) →
        (Applicants → Colleges → ℝ) →
          (val_college : Colleges → Applicants → ℝ) →
            (∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a') →
              GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges → Prop

value:
fun {Applicants : Type u_1} {Colleges : Type u_2} [DecidableEq Applicants] (quota : Colleges → ℕ)
    (val_applicant : Applicants → Colleges → ℝ) (val_college : Colleges → Applicants → ℝ)
    (hcollegeStrict : ∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a')
    (s : GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges) =>
  (∀ (a : Applicants) (c d : Colleges),
     a ∈ GS62CollegeAdmissions.ExactCollegeBatchedProcedure.waitingList quota val_college hcollegeStrict s c →
       a ∈ GS62CollegeAdmissions.ExactCollegeBatchedProcedure.waitingList quota val_college hcollegeStrict s d →
         c = d) ∧
    (∀ (a : Applicants) (c : Colleges),
       a ∈ s.applications c →
```

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceEligible val_applicant val_college a c) ∧
∀ (a : Applicants) (c d : Colleges),
a ∈ s.applications c →
GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceEligible val_applicant val_college a d →
a ∉ s.applications d → val_applicant a d ≤ val_applicant a c

Direct retained dependencies

- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceEligible
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.waitingList

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.assignedCollege

Declaration location: paper-local

Lean declaration body

```
type:
{Applicants : Type u_1} →
  {Colleges : Type u_2} →
    [Fintype Colleges] →
      [inst : DecidableEq Applicants] →
        (Colleges → ℕ) →
          (val_college : Colleges → Applicants → ℝ) →
            (∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a') →
              GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges →
                Applicants → Option Colleges

value:
fun {Applicants : Type u_1} {Colleges : Type u_2} [Fintype Colleges] [DecidableEq Applicants] (quota : Colleges → ℕ)
    (val_college : Colleges → Applicants → ℝ)
    (hcollegeStrict : ∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a')
    (s : GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges)
    (a : Applicants) =>
  if h :
      ∃ (c : Colleges),
        a ∈ GS62CollegeAdmissions.ExactCollegeBatchedProcedure.waitingList quota val_college hcollegeStrict s c then
    some (Classical.choose h)
  else none
```

Direct retained dependencies

- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.waitingList

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceActive

Declaration location: paper-local

Lean declaration body

```
type:
{Applicants : Type u_1} →
  {Colleges : Type u_2} →
    [Fintype Colleges] →
      [inst : DecidableEq Applicants] →
        (Colleges → ℕ) →
          (Applicants → Colleges → ℝ) →
            (val_college : Colleges → Applicants → ℝ) →
              (∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a') →
                GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges →
                  Applicants → Prop

value:
fun {Applicants : Type u_1} {Colleges : Type u_2} [Fintype Colleges] [DecidableEq Applicants] (quota : Colleges → ℕ)
    (val_applicant : Applicants → Colleges → ℝ) (val_college : Colleges → Applicants → ℝ)
    (hcollegeStrict : ∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a')
    (s : GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges)
    (a : Applicants) =>
  GS62CollegeAdmissions.ExactCollegeBatchedProcedure.assignedCollege quota val_college hcollegeStrict s a = none ∧
    (GS62CollegeAdmissions.ExactCollegeBatchedProcedure.untriedEligibleColleges val_applicant val_college s a).Nonempty
```

Direct retained dependencies

- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.assignedCollege
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.untriedEligibleColleges

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.nextCollege

Declaration location: paper-local

Lean declaration body

```
type:
{Applicants : Type u_1} →
  {Colleges : Type u_2} →
    [Fintype Applicants] →
      [Fintype Colleges] →
        [inst : DecidableEq Applicants] →
          [DecidableEq Colleges] →
            (Colleges → ℕ) →
              (Applicants → Colleges → ℝ) →
                (val_college : Colleges → Applicants → ℝ) →
                  (∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a') →
                    GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges →
                      Applicants → Option Colleges

value:
fun {Applicants : Type u_1} {Colleges : Type u_2} [Fintype Applicants] [Fintype Colleges] [DecidableEq Applicants]
    [DecidableEq Colleges] (quota : Colleges → ℕ) (val_applicant : Applicants → Colleges → ℝ)
    (val_college : Colleges → Applicants → ℝ)
    (hcollegeStrict : ∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a')
    (s : GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges)
    (a : Applicants) =>
  if hactive :
      GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceActive quota val_applicant val_college hcollegeStrict s
        a then
    some
      (Classical.choose
        (GS62CollegeAdmissions.ExactCollegeBatchedProcedure.nextCollege._proof_1 quota val_applicant val_college
          hcollegeStrict s a hactive))
  else none
```

Direct retained dependencies

- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceActive
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.untriedEligibleColleges

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.newApplications

Declaration location: paper-local

Lean declaration body

```
type:
{Applicants : Type u_1} →
  {Colleges : Type u_2} →
    [Fintype Applicants] →
      [Fintype Colleges] →
        [inst : DecidableEq Applicants] →
          [DecidableEq Colleges] →
            (Colleges → ℕ) →
              (Applicants → Colleges → ℝ) →
                (val_college : Colleges → Applicants → ℝ) →
                  (∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a') →
                    GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges →
                      Colleges → Finset Applicants

value:
fun {Applicants : Type u_1} {Colleges : Type u_2} [Fintype Applicants] [Fintype Colleges] [DecidableEq Applicants]
    [DecidableEq Colleges] (quota : Colleges → ℕ) (val_applicant : Applicants → Colleges → ℝ)
    (val_college : Colleges → Applicants → ℝ)
    (hcollegeStrict : ∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a')
    (s : GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges)
    (c : Colleges) =>
  {a : Applicants |
    GS62CollegeAdmissions.ExactCollegeBatchedProcedure.nextCollege quota val_applicant val_college hcollegeStrict s a =
      some c}
```

Direct retained dependencies

- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.nextCollege

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceAssignmentFromState

Declaration location: paper-local

Lean declaration body

```
type:
{Applicants : Type u_1} →
  {Colleges : Type u_2} →
    [Fintype Applicants] →
      [Fintype Colleges] →
        [inst : DecidableEq Applicants] →
          [DecidableEq Colleges] →
            (quota : Colleges → ℕ) →
              (val_applicant : Applicants → Colleges → ℝ) →
                (val_college : Colleges → Applicants → ℝ) →
                  (hcollegeStrict : ∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a') →
                    (s :
                        GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges) →
                      GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceStateInvariant quota val_applicant
                          val_college hcollegeStrict s →
                        AppliedModelingLib.Matching.ManyToOneAssignment Applicants Colleges

value:
fun {Applicants : Type u_1} {Colleges : Type u_2} [Fintype Applicants] [Fintype Colleges] [DecidableEq Applicants]
    [DecidableEq Colleges] (quota : Colleges → ℕ) (val_applicant : Applicants → Colleges → ℝ)
    (val_college : Colleges → Applicants → ℝ)
    (hcollegeStrict : ∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a')
    (s : GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges)
    (hinv :
      GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceStateInvariant quota val_applicant val_college
        hcollegeStrict s) =>
  { app_match := GS62CollegeAdmissions.ExactCollegeBatchedProcedure.assignedCollege quota val_college hcollegeStrict s,
    college_roster := GS62CollegeAdmissions.ExactCollegeBatchedProcedure.waitingList quota val_college hcollegeStrict s,
    consistent := fun (a : Applicants) (c : Colleges) =>
      GS62CollegeAdmissions.ExactCollegeBatchedProcedure.assignedCollege_eq_some_iff quota val_applicant val_college
        hcollegeStrict s hinv a c }
```

Direct retained dependencies

- AppliedModelingLib.Matching.ManyToOneAssignment
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceStateInvariant
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.assignedCollege
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.waitingList

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceStep

Declaration location: paper-local

Lean declaration body

```
type:
{Applicants : Type u_1} →
  {Colleges : Type u_2} →
    [Fintype Applicants] →
      [Fintype Colleges] →
        [inst : DecidableEq Applicants] →
          [DecidableEq Colleges] →
            (Colleges → ℕ) →
              (Applicants → Colleges → ℝ) →
                (val_college : Colleges → Applicants → ℝ) →
                  (∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a') →
                    GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges →
                      GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges

value:
fun {Applicants : Type u_1} {Colleges : Type u_2} [Fintype Applicants] [Fintype Colleges] [DecidableEq Applicants]
    [DecidableEq Colleges] (quota : Colleges → ℕ) (val_applicant : Applicants → Colleges → ℝ)
    (val_college : Colleges → Applicants → ℝ)
    (hcollegeStrict : ∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a')
    (s : GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges) =>
  {
    applications := fun (c : Colleges) =>
      s.applications c ∪
        GS62CollegeAdmissions.ExactCollegeBatchedProcedure.newApplications quota val_applicant val_college
          hcollegeStrict s c }
```

Direct retained dependencies

- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.newApplications

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceRunToTerminal

Declaration location: paper-local

Lean declaration body

```
type:
{Applicants : Type u_1} →
  {Colleges : Type u_2} →
    [Fintype Applicants] →
      [Fintype Colleges] →
        [inst : DecidableEq Applicants] →
          [DecidableEq Colleges] →
            (Colleges → ℕ) →
              (Applicants → Colleges → ℝ) →
                (val_college : Colleges → Applicants → ℝ) →
                  (∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a') →
                    GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges →
                      GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges

value:
fun {Applicants : Type u_1} {Colleges : Type u_2} [Fintype Applicants] [Fintype Colleges] [DecidableEq Applicants]
    [DecidableEq Colleges] (quota : Colleges → ℕ) (val_applicant : Applicants → Colleges → ℝ)
    (val_college : Colleges → Applicants → ℝ)
    (hcollegeStrict : ∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a')
    (a : GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges) =>
  WellFounded.Nat.fix
    (fun (x : GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges) =>
      (GS62CollegeAdmissions.ExactCollegeBatchedProcedure.untriedEligiblePairs val_applicant val_college x).card)
    (fun (a : GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges)
        (a_1 :
          (y : GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges) →
            InvImage (fun (x1 x2 : ℕ) => x1 < x2)
              (fun
                  (x :
                    GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges) =>
                (GS62CollegeAdmissions.ExactCollegeBatchedProcedure.untriedEligiblePairs val_applicant val_college
                    x).card)
              y a →
            GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges) =>
      let s : GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges := a;
      if hactive :
          ∃ (a : Applicants),
            GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceActive quota val_applicant val_college
              hcollegeStrict s a then
        a_1
          (GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceStep quota val_applicant val_college hcollegeStrict
            s)
          (GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceRunToTerminal._proof_1 quota val_applicant
            val_college hcollegeStrict a hactive)
      else s)
    a
```

Direct retained dependencies

- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceActive
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceStep
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.untriedEligiblePairs

Paper-specific semantic prerequisites

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalState

Paper-local declaration:
GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalState
Source locator: cited publication, lines 232-247

Verbatim paper-source connection

4. Stable assignments and the admissions problem. The extension of our
"deferred-acceptance"procedureto the problemofcollegeadmissionsis straight-
forward.For conveniencewe will assume that ifa college is not willingto accept
a student underany circumstances,as describedin Section 2, then that student
will not even be permittedto apply to the college. With this understandingthe
procedure follows: First, all students apply to the college of their firstchoice.
A college with a quota of q then places on its waiting list the q applicants who
rank highest,or all applicants if there are fewerthan q, and rejects the rest.
Rejected applicants then apply to their second choice and again each college
selects the top q fromamong the new applicants and those on its waiting list,
puts these on its new waitinglist,and rejects the rest.The procedureterminates
when every applicant is either on a waiting list or has been rejected by every
college to which he is willingand permittedto apply. At this point each college

[form-feed in source extraction]
14 COLLEGE ADMISSIONS AND STABILITY OF MARRIAGE [January

admits everyoneon its waitinglist and the stable assignmenthas been achieved.

Lean-expanded paper semantic target

type:
```
{Applicants : Type u_1} →
  {Colleges : Type u_2} →
    [Fintype Applicants] →
      [Fintype Colleges] →
        [inst : DecidableEq Applicants] →
          [DecidableEq Colleges] →
            (Colleges → ℕ) →
              (val_applicant : Applicants → Colleges → ℝ) →
                (val_college : Colleges → Applicants → ℝ) →
                  (∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a') →
                    (∀ (a : Applicants) (c c' : Colleges), val_applicant a c = val_applicant a c' → c = c') →
                      GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState Applicants Colleges
```

value:
```
fun {Applicants : Type u_1} {Colleges : Type u_2} [Fintype Applicants] [Fintype Colleges] [DecidableEq Applicants]
    [DecidableEq Colleges] (quota : Colleges → ℕ) (val_applicant : Applicants → Colleges → ℝ)
    (val_college : Colleges → Applicants → ℝ)
    (hcollegeStrict : ∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a')
    (happlicantStrict : ∀ (a : Applicants) (c c' : Colleges), val_applicant a c = val_applicant a c' → c = c') =>
  GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceRunToTerminal quota val_applicant val_college hcollegeStrict
    GS62CollegeAdmissions.ExactCollegeBatchedProcedure.initialSourceState
```

Direct retained dependencies

- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceWaitingListState
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.initialSourceState
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceRunToTerminal

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

Reason: The expanded value is the terminal waiting-list state obtained from the empty history by simultaneous rounds: unmatched applicants choose their best untried mutually eligible college, and each college retains its top quota-many applicants. It halts precisely when no such applicant-college pair remains, matching the Section 4 termination condition. The approved strict-list numeric extension warrants both strict ranking premises, while the two-sided positive filter faithfully represents willingness and permission pair by pair.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalAssignment

Paper-local declaration:
GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalAssignment
Source locator: cited publication, lines 232-247

Verbatim paper-source connection

4. Stable assignments and the admissions problem. The extension of our
"deferred-acceptance"procedureto the problemofcollegeadmissionsis straight-
forward.For conveniencewe will assume that ifa college is not willingto accept
a student underany circumstances,as describedin Section 2, then that student
will not even be permittedto apply to the college. With this understandingthe
procedure follows: First, all students apply to the college of their firstchoice.
A college with a quota of q then places on its waiting list the q applicants who
rank highest,or all applicants if there are fewerthan q, and rejects the rest.
Rejected applicants then apply to their second choice and again each college
selects the top q fromamong the new applicants and those on its waiting list,
puts these on its new waitinglist,and rejects the rest.The procedureterminates
when every applicant is either on a waiting list or has been rejected by every
college to which he is willingand permittedto apply. At this point each college

[form-feed in source extraction]
14 COLLEGE ADMISSIONS AND STABILITY OF MARRIAGE [January

admits everyoneon its waitinglist and the stable assignmenthas been achieved.

Lean-expanded paper semantic target

type:
```
{Applicants : Type u_1} →
  {Colleges : Type u_2} →
    [Fintype Applicants] →
      [Fintype Colleges] →
        [DecidableEq Applicants] →
          [DecidableEq Colleges] →
            (Colleges → ℕ) →
              (val_applicant : Applicants → Colleges → ℝ) →
                (val_college : Colleges → Applicants → ℝ) →
                  (∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a') →
                    (∀ (a : Applicants) (c c' : Colleges), val_applicant a c = val_applicant a c' → c = c') →
                      AppliedModelingLib.Matching.ManyToOneAssignment Applicants Colleges
```

value:
```
fun {Applicants : Type u_1} {Colleges : Type u_2} [Fintype Applicants] [Fintype Colleges] [DecidableEq Applicants]
    [DecidableEq Colleges] (quota : Colleges → ℕ) (val_applicant : Applicants → Colleges → ℝ)
    (val_college : Colleges → Applicants → ℝ)
    (hcollegeStrict : ∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a')
    (happlicantStrict : ∀ (a : Applicants) (c c' : Colleges), val_applicant a c = val_applicant a c' → c = c') =>
  GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceAssignmentFromState quota val_applicant val_college
    hcollegeStrict
    (GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalState quota val_applicant val_college
      hcollegeStrict happlicantStrict)
    (GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalState_invariant quota val_applicant
      val_college hcollegeStrict happlicantStrict)
```

Direct retained dependencies

- AppliedModelingLib.Matching.ManyToOneAssignment
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceStateInvariant
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceAssignmentFromState
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalState

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

Reason: The expanded value starts from the empty application history, repeatedly lets each unmatched applicant apply to the best untried pair that is positive for both sides, keeps each college’s top quota-many cumulative applicants, and converts the terminal waiting lists into the stated assignment. This is the Section 4 procedure and its terminal admission step. The approved strict-list numeric extension warrants both strict ranking premises; pairwise eligibility does not add the rejected global implication.
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 93-98

Verbatim source input

DEFINITION. An assignment ofapplicantstocolleges
willbecalledunstableif
thereare twoapplicantsa and ,Bwhoare assignedtocolleges
A andB, respectively,
although A toB and A prefers: toa.
,3prefers

Expanded PaperInterface specification

```
∀ {Applicants : Type u_1} {Colleges : Type u_2} (val_applicant : Applicants → Colleges → ℝ)
  (val_college : Colleges → Applicants → ℝ) (mu : AppliedModelingLib.Matching.ManyToOneAssignment Applicants Colleges),
  ∃ (alpha : Applicants) (beta : Applicants) (A : Colleges) (B : Colleges),
    mu.app_match alpha = some A ∧
      mu.app_match beta = some B ∧
        val_applicant beta B < val_applicant beta A ∧ val_college A alpha < val_college A beta
```

Retained governing declarations

- AppliedModelingLib.Matching.ManyToOneAssignment

Lean-elaborated claim roles

1. parameter: Type u_1
2. parameter: Type u_2
3. parameter: Applicants → Colleges → ℝ
4. parameter: Colleges → Applicants → ℝ
5. parameter: AppliedModelingLib.Matching.ManyToOneAssignment Applicants Colleges
6. conclusion: ∃ (alpha : Applicants) (beta : Applicants) (A : Colleges) (B : Colleges),
  mu.app_match alpha = some A ∧
    mu.app_match beta = some B ∧ val_applicant beta B < val_applicant beta A ∧ val_college A alpha < val_college A beta

Lean proof endpoint (built separately)

GS62CollegeAdmissions.PaperInterface.unstableCollegeAssignment_iff_source_definition

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

Reason: The target is exactly the printed replacement-pair criterion: two assigned applicants, one preferring the other's college, and that college preferring the applicant over its assignee.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 2

Source locator: cited publication, lines 115-117

Verbatim source input

DEFINITION. A stableassignment applicantis at least
is calledoptimalif every
as welloffunderit as underanyotherstableassignment.

Expanded PaperInterface specification

```
∀ {Applicants : Type u_1} {Colleges : Type u_2} (quota : Colleges → ℕ) (val_applicant : Applicants → Colleges → ℝ)
  (val_college : Colleges → Applicants → ℝ) (mu : AppliedModelingLib.Matching.ManyToOneAssignment Applicants Colleges),
  (∀ (c : Colleges), (mu.college_roster c).card ≤ quota c) ∧
    (¬∃ (alpha : Applicants) (beta : Applicants) (A : Colleges) (B : Colleges),
        mu.app_match alpha = some A ∧
          mu.app_match beta = some B ∧
            val_applicant beta B < val_applicant beta A ∧ val_college A alpha < val_college A beta) ∧
      ∀ (nu : AppliedModelingLib.Matching.ManyToOneAssignment Applicants Colleges),
        (∀ (c : Colleges), (nu.college_roster c).card ≤ quota c) →
          (¬∃ (alpha : Applicants) (beta : Applicants) (A : Colleges) (B : Colleges),
              nu.app_match alpha = some A ∧
                nu.app_match beta = some B ∧
                  val_applicant beta B < val_applicant beta A ∧ val_college A alpha < val_college A beta) →
            ∀ (a : Applicants),
              (match nu.app_match a with
                | none => 0
                | some c => val_applicant a c) ≤
              match mu.app_match a with
              | none => 0
              | some c => val_applicant a c
```

Retained governing declarations

- AppliedModelingLib.Matching.ManyToOneAssignment

Lean-elaborated claim roles

1. parameter: Type u_1
2. parameter: Type u_2
3. parameter: Colleges → ℕ
4. parameter: Applicants → Colleges → ℝ
5. parameter: Colleges → Applicants → ℝ
6. parameter: AppliedModelingLib.Matching.ManyToOneAssignment Applicants Colleges
7. conclusion: (∀ (c : Colleges), (mu.college_roster c).card ≤ quota c) ∧
   (¬∃ (alpha : Applicants) (beta : Applicants) (A : Colleges) (B : Colleges),
       mu.app_match alpha = some A ∧
         mu.app_match beta = some B ∧
           val_applicant beta B < val_applicant beta A ∧ val_college A alpha < val_college A beta) ∧
     ∀ (nu : AppliedModelingLib.Matching.ManyToOneAssignment Applicants Colleges),
       (∀ (c : Colleges), (nu.college_roster c).card ≤ quota c) →
         (¬∃ (alpha : Applicants) (beta : Applicants) (A : Colleges) (B : Colleges),
             nu.app_match alpha = some A ∧
               nu.app_match beta = some B ∧
                 val_applicant beta B < val_applicant beta A ∧ val_college A alpha < val_college A beta) →
           ∀ (a : Applicants),
             (match nu.app_match a with
               | none => 0
               | some c => val_applicant a c) ≤
             match mu.app_match a with
             | none => 0
             | some c => val_applicant a c

Lean proof endpoint (built separately)

GS62CollegeAdmissions.PaperInterface.literalApplicantOptimalCollegeAssignment_iff_source_definition

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

Reason: The surrounding Section 2 text fixes each college’s quota and explains the displayed instability pair as a replacement that stays within that quota. The target requires the candidate roster to satisfy card <= quota and applies the same condition to every comparison assignment before treating it as stable. Both no-pair clauses literally require alpha at A, beta at B, beta preferring A to B, and A preferring beta to alpha; it then requires every applicant to be at least as well off under the candidate as under each such stable comparison assignment.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 3

Source locator: cited publication, lines 133-139

Verbatim source input

A certain communityconsists of n men and n women. Each person ranks
those of the opposite sex in accordance with his or her preferencesfora marriage
partner.We seek a satisfactoryway of marryingoffall membersof the com-
munity.Imitatingour earlierdefinition,we call a set of marriagesunstable(and
here the suitabilityof the termis quite clear) if under it thereare a man and a
woman who are not marriedto each other but prefereach other to theiractual
mates.

Expanded PaperInterface specification

```
∀ {M : Type u_1} {W : Type u_2} (val_m : M → W → ℝ) (val_w : W → M → ℝ)
  (mu : AppliedModelingLib.Matching.Assignment M W),
  ((∀ (m : M), ∃ (w : W), mu.m_match m = some w) ∧ ∀ (w : W), ∃ (m : M), mu.w_match w = some m) ∧
    ∃ (m : M) (w : W),
      AppliedModelingLib.Matching.valM val_m m (mu.m_match m) < val_m m w ∧
        AppliedModelingLib.Matching.valW val_w w (mu.w_match w) < val_w w m
```

Retained governing declarations

- AppliedModelingLib.Matching.Assignment
- AppliedModelingLib.Matching.valM
- AppliedModelingLib.Matching.valW

Lean-elaborated claim roles

1. parameter: Type u_1
2. parameter: Type u_2
3. parameter: M → W → ℝ
4. parameter: W → M → ℝ
5. parameter: AppliedModelingLib.Matching.Assignment M W
6. conclusion: ((∀ (m : M), ∃ (w : W), mu.m_match m = some w) ∧ ∀ (w : W), ∃ (m : M), mu.w_match w = some m) ∧
   ∃ (m : M) (w : W),
     AppliedModelingLib.Matching.valM val_m m (mu.m_match m) < val_m m w ∧
       AppliedModelingLib.Matching.valW val_w w (mu.w_match w) < val_w w m

Lean proof endpoint (built separately)

GS62CollegeAdmissions.PaperInterface.unstableMarriage_iff_source_definition

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

Reason: The target requires a complete marriage and a man and woman who each strictly prefer the other to their assigned mate; strict preference makes them necessarily not married to each other, matching the source definition.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 4

Source locator: cited publication, lines 127-139

Verbatim source input

3. Stable assignments and a marriage problem. In trying to settle the question of the existence of stable assignmentswe were led to look firstat a special case, in whichthereare the same numberof applicants as colleges and all quotas are unity. This situation is, of course, highlyunnatural in the context of college admissions, but there is another "story" into which it fits quite readily.
A certain communityconsists of n men and n women. Each person ranks those of the opposite sex in accordance with his or her preferencesfora marriage partner.We seek a satisfactoryway of marryingoffall membersof the community.Imitatingour earlierdefinition,we call a set of marriagesunstable(and here the suitabilityof the termis quite clear) if under it thereare a man and a woman who are not marriedto each other but prefereach other to theiractual mates.

[Next verbatim source excerpt]

THEOREM 1. Therealwaysexistsa stablesetofmarriages.

[Next verbatim source excerpt]

3. Stable assignments and a marriage problem. In trying to settle the question of the existence of stable assignmentswe were led to look firstat a special case, in whichthereare the same numberof applicants as colleges and all quotas are unity. This situation is, of course, highlyunnatural in the context of college admissions, but there is another "story" into which it fits quite readily.

[Next verbatim source excerpt]

A certain communityconsists of n men and n women. Each person ranks those of the opposite sex in accordance with his or her preferencesfora marriage partner.We seek a satisfactoryway of marryingoffall membersof the community.Imitatingour earlierdefinition,we call a set of marriagesunstable(and here the suitabilityof the termis quite clear) if under it thereare a man and a woman who are not marriedto each other but prefereach other to theiractual mates.

Expanded PaperInterface specification

```
∀ {M : Type u_1} {W : Type u_2} [inst : Fintype M] [inst_1 : Fintype W] [DecidableEq M] [DecidableEq W]
  (val_m : M → W → ℝ) (val_w : W → M → ℝ),
  Fintype.card M = Fintype.card W →
    (∀ (m : M) (w w' : W), val_m m w = val_m m w' → w = w') →
      (∀ (w : W) (m m' : M), val_w w m = val_w w m' → m = m') →
        (∀ (m : M) (w : W), 0 < val_m m w) →
          (∀ (w : W) (m : M), 0 < val_w w m) →
            ∃ (mu : AppliedModelingLib.Matching.Assignment M W),
              ¬(((∀ (m : M), ∃ (w : W), mu.m_match m = some w) ∧ ∀ (w : W), ∃ (m : M), mu.w_match w = some m) ∧
                  ∃ (m : M) (w : W),
                    AppliedModelingLib.Matching.valM val_m m (mu.m_match m) < val_m m w ∧
                      AppliedModelingLib.Matching.valW val_w w (mu.w_match w) < val_w w m) ∧
                (∀ (m : M), ∃ (w : W), mu.m_match m = some w) ∧ ∀ (w : W), ∃ (m : M), mu.w_match w = some m
```

Retained governing declarations

- AppliedModelingLib.Matching.Assignment
- AppliedModelingLib.Matching.valM
- AppliedModelingLib.Matching.valW

Lean-elaborated claim roles

```
1. parameter: Type u_1
2. parameter: Type u_2
3. parameter: Fintype M
4. parameter: Fintype W
5. parameter: DecidableEq M
6. parameter: DecidableEq W
7. parameter: M → W → ℝ
8. parameter: W → M → ℝ
9. assumption: Fintype.card M = Fintype.card W
10. assumption: ∀ (m : M) (w w' : W), val_m m w = val_m m w' → w = w'
11. assumption: ∀ (w : W) (m m' : M), val_w w m = val_w w m' → m = m'
12. assumption: ∀ (m : M) (w : W), 0 < val_m m w
13. assumption: ∀ (w : W) (m : M), 0 < val_w w m
14. conclusion: ∃ (mu : AppliedModelingLib.Matching.Assignment M W),
  ¬(((∀ (m : M), ∃ (w : W), mu.m_match m = some w) ∧ ∀ (w : W), ∃ (m : M), mu.w_match w = some m) ∧
      ∃ (m : M) (w : W),
        AppliedModelingLib.Matching.valM val_m m (mu.m_match m) < val_m m w ∧
          AppliedModelingLib.Matching.valW val_w w (mu.w_match w) < val_w w m) ∧
    (∀ (m : M), ∃ (w : W), mu.m_match m = some w) ∧ ∀ (w : W), ∃ (m : M), mu.w_match w = some m
```

Lean proof endpoint (built separately)

GS62CollegeAdmissions.PaperInterface.theorem1_stable_marriage_exists

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

Reason: For equal finite complete strict lists, the target returns a perfect matching and excludes a mutually preferred nonmatching pair, which is exactly the source's stable set of marriages.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 5

Source locator: cited publication, lines 253-255

Verbatim source input

THEOREM 2. Everyapplicant is at least as well offundertheassignmentgiven
by thedeferredacceptanceprocedureas he would be underany otherstableassign-
ment.

[Next verbatim source excerpt]

4. Stable assignments and the admissions problem. The extension of our
"deferred-acceptance"procedureto the problemofcollegeadmissionsis straight-
forward.For conveniencewe will assume that ifa college is not willingto accept
a student underany circumstances,as describedin Section 2, then that student
will not even be permittedto apply to the college. With this understandingthe
procedure follows: First, all students apply to the college of their firstchoice.
A college with a quota of q then places on its waiting list the q applicants who
rank highest,or all applicants if there are fewerthan q, and rejects the rest.
Rejected applicants then apply to their second choice and again each college
selects the top q fromamong the new applicants and those on its waiting list,
puts these on its new waitinglist,and rejects the rest.The procedureterminates
when every applicant is either on a waiting list or has been rejected by every
college to which he is willingand permittedto apply. At this point each college

[form-feed in source extraction]
14 COLLEGE ADMISSIONS AND STABILITY OF MARRIAGE [January

[Next verbatim source excerpt]

admits everyoneon its waitinglist and the stable assignmenthas been achieved.
The proofthat the assignmentis stable is entirelyanalogous to the proofgiven
forthe marriageproblemand is leftto the reader.
5. Optimality.We now show that the "deferredacceptance" procedurejust
described yields not only a stable but an optimal assignment of applicants.
That is,

Settled source reading The result-specific conditions and corrections are stated in the [clarification memo](docs/SOURCE_CLARIFICATIONS.md).

Expanded PaperInterface specification

```
∀ {Applicants : Type u_1} {Colleges : Type u_2} [inst : Fintype Applicants] [inst_1 : Fintype Colleges]
  [inst_2 : DecidableEq Applicants] [inst_3 : DecidableEq Colleges] (quota : Colleges → ℕ)
  (val_applicant : Applicants → Colleges → ℝ) (val_college : Colleges → Applicants → ℝ)
  (happlicant_strict :
    (∀ (a : Applicants) (c c' : Colleges), val_applicant a c = val_applicant a c' → c = c') ∧
      ∀ (a : Applicants) (c : Colleges), val_applicant a c ≠ 0)
  (hcollege_strict :
    (∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a') ∧
      ∀ (c : Colleges) (a : Applicants), val_college c a ≠ 0)
  (nu : AppliedModelingLib.Matching.ManyToOneAssignment Applicants Colleges),
  ((∀ (c : Colleges), (nu.college_roster c).card ≤ quota c) ∧
      (∀ (a : Applicants), 0 ≤ AppliedModelingLib.Matching.ManyToOne.valApplicant val_applicant a (nu.app_match a)) ∧
        (∀ (c : Colleges), ∀ a ∈ nu.college_roster c, 0 ≤ val_college c a) ∧
          ∀ (a : Applicants) (c : Colleges),
            AppliedModelingLib.Matching.ManyToOne.valApplicant val_applicant a (nu.app_match a) < val_applicant a c →
              (0 < val_college c a ∧ (nu.college_roster c).card < quota c ∨
                  ∃ a' ∈ nu.college_roster c, val_college c a' < val_college c a) →
                False) →
    ∀ (a : Applicants),
      AppliedModelingLib.Matching.ManyToOne.valApplicant val_applicant a (nu.app_match a) ≤
        AppliedModelingLib.Matching.ManyToOne.valApplicant val_applicant a
          ((GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalAssignment quota val_applicant
              val_college
              (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_1 val_college
                hcollege_strict)
              (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_2
                val_applicant happlicant_strict)).app_match
            a)
```

Retained governing declarations

- AppliedModelingLib.Matching.ManyToOne.valApplicant
- AppliedModelingLib.Matching.ManyToOneAssignment
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalAssignment

Lean-elaborated claim roles

1. parameter: Type u_1
2. parameter: Type u_2
3. parameter: Fintype Applicants

```
4. parameter: Fintype Colleges
5. parameter: DecidableEq Applicants
6. parameter: DecidableEq Colleges
7. parameter: Colleges → ℕ
8. parameter: Applicants → Colleges → ℝ
9. parameter: Colleges → Applicants → ℝ
10. assumption: (∀ (a : Applicants) (c c' : Colleges), val_applicant a c = val_applicant a c' → c = c') ∧
  ∀ (a : Applicants) (c : Colleges), val_applicant a c ≠ 0
11. assumption: (∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a') ∧
  ∀ (c : Colleges) (a : Applicants), val_college c a ≠ 0
12. parameter: AppliedModelingLib.Matching.ManyToOneAssignment Applicants Colleges
13. assumption: (∀ (c : Colleges), (nu.college_roster c).card ≤ quota c) ∧
  (∀ (a : Applicants), 0 ≤ AppliedModelingLib.Matching.ManyToOne.valApplicant val_applicant a (nu.app_match a)) ∧
    (∀ (c : Colleges), ∀ a ∈ nu.college_roster c, 0 ≤ val_college c a) ∧
      ∀ (a : Applicants) (c : Colleges),
        AppliedModelingLib.Matching.ManyToOne.valApplicant val_applicant a (nu.app_match a) < val_applicant a c →
          (0 < val_college c a ∧ (nu.college_roster c).card < quota c ∨
              ∃ a' ∈ nu.college_roster c, val_college c a' < val_college c a) →
            False
14. parameter: Applicants
15. conclusion: AppliedModelingLib.Matching.ManyToOne.valApplicant val_applicant a (nu.app_match a) ≤
  AppliedModelingLib.Matching.ManyToOne.valApplicant val_applicant a
    ((GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalAssignment quota val_applicant
        val_college
        (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_1 val_college
          hcollege_strict)
        (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_2 val_applicant
          happlicant_strict)).app_match
      a)
```

Lean proof endpoint (built separately)

GS62CollegeAdmissions.PaperInterface.theorem2_applicant_optimality

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

Reason: The expanded target quantifies every assignment satisfying quota feasibility, both individual-rationality clauses, and the same vacancy-or-displacement no-block condition, then for every applicant compares its value weakly below the value in sourceWaitingListFinalAssignment. That is the source theorem's applicant-wise weak preference for the direct Section 4 procedure outcome over every other stable assignment.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 6

Source locator: cited publication, lines 232-249

Verbatim source input

admits everyoneon its waitinglist and the stable assignmenthas been achieved.
The proofthat the assignmentis stable is entirelyanalogous to the proofgiven
forthe marriageproblemand is leftto the reader.

Settled source reading The result-specific conditions and corrections are stated in the [clarification memo](docs/SOURCE_CLARIFICATIONS.md).

Expanded PaperInterface specification

```
∀ {Applicants : Type u_1} {Colleges : Type u_2} [inst : Fintype Applicants] [inst_1 : Fintype Colleges]
  [inst_2 : DecidableEq Applicants] [inst_3 : DecidableEq Colleges] (quota : Colleges → ℕ)
  (val_applicant : Applicants → Colleges → ℝ) (val_college : Colleges → Applicants → ℝ)
  (happlicant_strict :
    (∀ (a : Applicants) (c c' : Colleges), val_applicant a c = val_applicant a c' → c = c') ∧
      ∀ (a : Applicants) (c : Colleges), val_applicant a c ≠ 0)
  (hcollege_strict :
    (∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a') ∧
      ∀ (c : Colleges) (a : Applicants), val_college c a ≠ 0),
  (¬∃ (a : Applicants),
        GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceActive quota val_applicant val_college
          (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_1 val_college
            hcollege_strict)
          (GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalState quota val_applicant
            val_college
            (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_1 val_college
              hcollege_strict)
            (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_2 val_applicant
              happlicant_strict))
          a) ∧
    (∀ (c : Colleges),
       ((GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalAssignment quota val_applicant
              val_college
              (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_1
                val_college hcollege_strict)
              (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_2
                val_applicant happlicant_strict)).college_roster
            c).card ≤
          quota c) ∧
      (∀ (a : Applicants),
         0 ≤
          AppliedModelingLib.Matching.ManyToOne.valApplicant val_applicant a
            ((GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalAssignment quota val_applicant
                val_college
                (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_1
                  val_college hcollege_strict)
                (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_2
                  val_applicant happlicant_strict)).app_match
              a)) ∧
        (∀ (c : Colleges),
           ∀
             a ∈
               (GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalAssignment quota val_applicant
                    val_college
                    (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_1
                      val_college hcollege_strict)
                    (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_2
                      val_applicant happlicant_strict)).college_roster
                 c,
             0 ≤ val_college c a) ∧
          ∀ (a : Applicants) (c : Colleges),
            AppliedModelingLib.Matching.ManyToOne.valApplicant val_applicant a
                ((GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalAssignment quota
                    val_applicant val_college
                    (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_1
                      val_college hcollege_strict)
                    (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_2
                      val_applicant happlicant_strict)).app_match
                  a) <
              val_applicant a c →
            (0 < val_college c a ∧
                ((GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalAssignment quota
                      val_applicant val_college
                      (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_1
                        val_college hcollege_strict)
                      (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_2
                        val_applicant happlicant_strict)).college_roster
                    c).card <
                quota c ∨
              ∃
                a' ∈
                  (GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalAssignment quota
                      val_applicant val_college
                      (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_1
                        val_college hcollege_strict)
```

(GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_2
val_applicant happlicant_strict)).college_roster
c,
val_college c a' < val_college c a) →
False

Retained governing declarations

- AppliedModelingLib.Matching.ManyToOne.valApplicant
- AppliedModelingLib.Matching.ManyToOneAssignment
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceActive
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalAssignment
- GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalState

Lean-elaborated claim roles

1. parameter: Type u_1
2. parameter: Type u_2
3. parameter: Fintype Applicants
4. parameter: Fintype Colleges
5. parameter: DecidableEq Applicants
6. parameter: DecidableEq Colleges
7. parameter: Colleges → ℕ
8. parameter: Applicants → Colleges → ℝ
9. parameter: Colleges → Applicants → ℝ
10. assumption: (∀ (a : Applicants) (c c' : Colleges), val_applicant a c = val_applicant a c' → c = c') ∧
∀ (a : Applicants) (c : Colleges), val_applicant a c ≠ 0
11. assumption: (∀ (c : Colleges) (a a' : Applicants), val_college c a = val_college c a' → a = a') ∧
∀ (c : Colleges) (a : Applicants), val_college c a ≠ 0
12. conclusion: (¬∃ (a : Applicants),
GS62CollegeAdmissions.ExactCollegeBatchedProcedure.SourceActive quota val_applicant val_college
(GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_1 val_college
hcollege_strict)
(GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalState quota val_applicant val_college
(GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_1 val_college
hcollege_strict)
(GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_2 val_applicant
happlicant_strict))
a) ∧
(∀ (c : Colleges),
((GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalAssignment quota val_applicant
val_college
(GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_1 val_college
hcollege_strict)
(GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_2
val_applicant happlicant_strict)).college_roster
c).card ≤
quota c) ∧
(∀ (a : Applicants),
0 ≤
AppliedModelingLib.Matching.ManyToOne.valApplicant val_applicant a
((GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalAssignment quota val_applicant
val_college
(GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_1
val_college hcollege_strict)
(GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_2
val_applicant happlicant_strict)).app_match
a)) ∧
(∀ (c : Colleges),
∀
a ∈
(GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalAssignment quota val_applicant
val_college
(GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_1
val_college hcollege_strict)
(GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_2
val_applicant happlicant_strict)).college_roster
c,
0 ≤ val_college c a) ∧
∀ (a : Applicants) (c : Colleges),
AppliedModelingLib.Matching.ManyToOne.valApplicant val_applicant a
((GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalAssignment quota
val_applicant val_college
(GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_1
val_college hcollege_strict)
(GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_2
val_applicant happlicant_strict)).app_match
a) <
val_applicant a c →
(0 < val_college c a ∧
((GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalAssignment quota
val_applicant val_college
(GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_1
val_college hcollege_strict)
(GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_2
val_applicant happlicant_strict)).college_roster
c).card <
quota c ∨

```
∃
  a' ∈
    (GS62CollegeAdmissions.ExactCollegeBatchedProcedure.sourceWaitingListFinalAssignment quota
         val_applicant val_college
         (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_1
           val_college hcollege_strict)
         (GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stabilitySpec._proof_2
           val_applicant happlicant_strict)).college_roster
       c,
   val_college c a' < val_college c a) →
False
```

Lean proof endpoint (built separately)

GS62CollegeAdmissions.PaperInterface.section4_waiting_list_terminal_stability

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

Reason: The expanded conclusion says the direct final state has no SourceActive applicant (which expands to being unmatched with an untried mutually eligible college) and that its returned assignment is quota-feasible, applicant- and college-individually rational, and has no strictly preferred mutually eligible vacancy-or-displacement block. This is the stated terminal waiting-list outcome under the approved completed stability reading.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation