Structure and Projection Theorems: Complete Exercise Guide with ALL Exam Examples

Core Theoretical Foundation

Structure Theorem (CRITICAL THEOREM)

Definition: Let $P(\vec{x}) \subseteq \mathbb{N}^k$ be a predicate. Then $P(\vec{x})$ is semi-decidable if and only if there exists a decidable predicate $Q(t,\vec{x}) \subseteq \mathbb{N}^{k+1}$ such that $P(\vec{x}) = \exists t.Q(t,\vec{x})$.

Complete Proof: **(⇒)** If $P(\vec{x})$ is semi-decidable:

- sc_P is computable, so $\exists e \in \mathbb{N}$ such that sc_P $= \varphi_e^{(k)}$
- $P(\vec{x})$ iff sc_P$(\vec{x}) = 1$ iff sc_P$(\vec{x}) \downarrow$ iff $\varphi_e^{(k)}(\vec{x}) \downarrow$ iff $\exists t.H^{(k)}(e,\vec{x},t)$
- Set $Q(t,\vec{x}) = H^{(k)}(e,\vec{x},t)$ (decidable since H is decidable)

(⇐) If $P(\vec{x}) = \exists t.Q(t,\vec{x})$ with Q decidable:

- sc_P$(\vec{x}) = 1(\mu t.|\chi$_Q$(t,\vec{x}) - 1|)$ is computable
- Therefore P is semi-decidable

Projection Theorem (CRITICAL THEOREM)

Definition: Let $P(x,\vec{y}) \subseteq \mathbb{N}^{k+1}$ be semi-decidable. Then $R(\vec{y}) \equiv \exists x.P(x,\vec{y})$ is semi-decidable.

Complete Proof:

1. P semi-decidable $\Longrightarrow$ by Structure Theorem: $P(x,\vec{y}) \equiv \exists t.Q(t,x,\vec{y})$ with Q decidable
2. $R(\vec{y}) \equiv \exists x.P(x,\vec{y}) \equiv \exists x.\exists t.Q(t,x,\vec{y}) \equiv \exists w.Q((w)_1,(w)_2,\vec{y})$
3. Since Q decidable and existential quantification preserves semi-decidability $\Longrightarrow$ R semi-decidable

Complete Closure Properties Analysis

Closure Under Logical Operations (MEMORIZE THIS TABLE)

Operation	Decidable	Semi-Decidable	Proof Method
Negation ¬	✓ CLOSED	✗ NOT CLOSED	$K \in K$ vs $\bar{K} \notin K$
Conjunction ∧	✓ CLOSED	✓ CLOSED	Structure + encoding
Disjunction ∨	✓ CLOSED	✓ CLOSED	Structure + encoding
Existential ∃	✗ NOT CLOSED	✓ CLOSED	Projection Theorem
Universal ∀	✗ NOT CLOSED	✗ NOT CLOSED	Can make r.e. → non-r.e.

Detailed Closure Proofs

Conjunction Closure: If $P(\vec{x})$, $Q(\vec{x})$ semi-decidable, then $P(\vec{x}) \wedge Q(\vec{x})$ semi-decidable.

```
Proof:
- P(x⃗) ≡ ∃t.P'(t,x⃗), Q(x⃗) ≡ ∃t.Q'(t,x⃗) (Structure Theorem)
- P ∧ Q ≡ ∃w.(P'((w)₁,x⃗) ∧ Q'((w)₂,x⃗))
- Since conjunction of decidable predicates is decidable, apply Structure Theorem
```

Negation Non-Closure: Semi-decidable predicates NOT closed under negation.

```
Counterexample:
- Q(x) ≡ "x ∈ K" ≡ "ϕₓ(x) ↓" (semi-decidable)
- ¬Q(x) ≡ "x ∉ K" ≡ "ϕₓ(x) ↑" (NOT semi-decidable)
```

ALL EXAM EXERCISES WITH COMPLETE SOLUTIONS

Type 1: Direct Structure/Projection Application

Exam 2023-02-01 (without assuming theorems) *Exercise*: Show that if $Q(\vec{x},y)$ is semi-decidable then $P(\vec{x}) = \exists y.Q(\vec{x},y)$ is semi-decidable. Does the converse hold?

Complete Solution:

1. **Forward Direction**:
   - Q semi-decidable $\Longrightarrow$ sc_Q computable with index e
   - $Q(\vec{x},y)$ iff $\varphi_e^{(k+1)}(\vec{x},y)\downarrow$ iff $\exists t.H^{(k+1)}(e,\langle\vec{x},y\rangle,t)$
   - $P(\vec{x}) = \exists y.\exists t.H^{(k+1)}(e,\langle\vec{x},y\rangle,t) = \exists w.H^{(k+1)}(e,\langle\vec{x},\langle w\rangle_1\rangle,\langle w\rangle_2)$
   - Therefore $sc\_P(\vec{x}) = 1(\mu w.|\chi\_H^{(k+1)}(e,\langle\vec{x},\langle w\rangle_1\rangle,\langle w\rangle_2) - 1|)$ computable
2. **Converse FALSE**: $Q(x,y) \equiv$ "$\varphi_y(x)\uparrow$" (not semi-decidable), but $P(x) \equiv \exists y.$"$\varphi_y(x)\uparrow$" (always true, hence decidable with e_0)

Exam 2022-06-17 (Structure Theorem Bidirectional) *Exercise*: Show $P(\vec{x})$ is semi-decidable iff ∃ decidable $Q(\vec{x},y)$ such that $P(\vec{x}) = \exists y.Q(\vec{x},y)$.

Complete Solution: This IS the Structure Theorem. Use complete proof above.

Type 2: Function Equality Predicates

Exam 2024-02-16 *Exercise*: Given f: ℕ → ℕ, define Q_f(x,y) ≡ "f(x) = y". Show f computable iff Q_f semi-decidable.

Complete Solution:

1. **(⇒)** f computable:
   - If e is index for f, then $Q\_f(x,y) \equiv \exists t.S(e,x,y,t)$
   - Since S decidable, Q_f semi-decidable by Structure Theorem
2. **(⇐)** Q_f semi-decidable:
   - To compute f(x), search y such that Q_f(x,y) holds
   - $f(x) = \langle \mu w.S(e,x,\langle w\rangle_1,\langle w\rangle_2)\rangle_1$ where e is index for sc_Q_f

Exam 2020-06-30 *Exercise*: Given f,g: $\mathbb{N} \to \mathbb{N}$ with f total, define Q_fg(x) = "f(x) = g(x)". Show that if f,g computable, then Q_fg semi-decidable.

Complete Solution:

- Let e_1, e_2 be indices such that $f = \varphi_{e1}$, $g = \varphi_{e2}$
- $sc\_Q\_fg(x) = 1(\mu w.|f(x) - g(x)|)$ is computable
- Therefore Q_fg semi-decidable

Type 3: Projection with Counterexamples

Exam 2015-07-16 *Exercise*: Show that if $P(x,\vec{y})$ semi-decidable, then $\exists x.P(x,\vec{y})$ semi-decidable. Does converse hold?

Complete Solution:

1. **Forward**: Direct application of Projection Theorem
2. **Converse FALSE**:
   - **Standard**: P(x,y) = "$x \notin W_x$" (not semi-decidable), Q(y) = $\exists x.P(x,y)$ (always true, decidable)
   - **Less degenerate**: $P(x,y) = (y > x) \wedge (y \notin W_x)$, then $Q(y) = \exists x.P(x,y)$ decidable for $y > e_0$

Type 4: Closure Under Logical Operations

Universal vs Existential Quantification Analysis

- **Existential**: Decidable + ∃ = Semi-decidable (Structure Theorem)
- **Universal**: Decidable + ∀ = Can become non-semi-decidable

Example:

- $R(t,x) \equiv \neg H(x,x,t)$ (decidable)
- $\forall t.R(t,x) \equiv$ "$x \notin K$" (not semi-decidable)

STANDARD COUNTEREXAMPLE PATTERNS (MEMORIZE THESE)

Pattern A: Always Undefined Function

```
Let e₀ be index for always undefined function
For P(x,y) involving "ϕᵧ(x) ↑":
∃y.P(x,y) becomes universally true by taking y = e₀
```

Pattern B: Halting Set K

```
K = {x | x ∈ Wₓ} = {x | ϕₓ(x) ↓} (semi-decidable, not decidable)
K̄ = {x | x ∉ Wₓ} = {x | ϕₓ(x) ↑} (not semi-decidable)
```

Pattern C: Degenerate Dependency

```
P(x,y) = (y = constant) ∧ (property of x)
Makes ∃x.P(x,y) depend only on y, often decidable
But P inherits decidability/semi-decidability from property of x
```

Pattern D: Non-Semi-Decidable via Composition

```
If P(x,y) not semi-decidable, show:
- If some projection were semi-decidable
- Then composition would make non-semi-decidable predicate semi-decidable
- Contradiction
```

WORKED EXAMPLES FROM EXERCISES

Example 1: Projection Theorem Without Assuming It

Problem: Prove ∃ quantification preserves semi-decidability without using theorems.

Solution Template:

```
Given: Q(x⃗,y) semi-decidable
Want: P(x⃗) = ∃y.Q(x⃗,y) semi-decidable

1. Q semi-decidable ⟹ sc_Q computable with index e
2. Q(x⃗,y) iff sc_Q(x⃗,y) ↓ iff ∃t.H⁽ᵏ⁺¹⁾(e,⟨x⃗,y⟩,t)
3. P(x⃗) iff ∃y.∃t.H⁽ᵏ⁺¹⁾(e,⟨x⃗,y⟩,t)
4. Use pairing: P(x⃗) iff ∃w.H⁽ᵏ⁺¹⁾(e,⟨x⃗,(w)₁⟩,(w)₂)
5. sc_P(x⃗) = 1(μw.|χ_H⁽ᵏ⁺¹⁾(e,⟨x⃗,(w)₁⟩,(w)₂) - 1|) computable
```

Example 2: Bidirectional Structure Theorem

Problem: Show P semi-decidable ⟺ ∃ decidable Q such that $P(\vec{x}) = \exists y.Q(\vec{x},y)$.

Solution:

- **(⇒)**: Use halting predicate H as witness
- **(⇐)**: Use μ-operator to construct semi-characteristic function

Example 3: Function Computability via Predicates

Problem: Show function f computable iff its equality predicate is semi-decidable.

Key Insight: Semi-decidability allows "search" for correct output value.

STRATEGIC APPROACH FOR EXAMS

Step 1: Identify Exercise Type

1. **Direct application**: Apply Structure/Projection theorem
2. **Converse question**: Construct counterexample using patterns
3. **Closure properties**: Use table above
4. **Function-predicate relationship**: Use semi-decidability for search

Step 2: Apply Appropriate Template

- **Semi-decidability proof**: Find decidable Q, apply Structure Theorem
- **Non-semi-decidability proof**: Reduce from $\bar{K}$ or use closure contradiction
- **Counterexample**: Use Pattern A, B, C, or D from above

Step 3: Use Standard Notation

- $H^{(k)}(e,\vec{x},t)$ for halting predicate
- $\langle w\rangle_1$, $\langle w\rangle_2$ for pairing projections
- μ-operator for semi-characteristic functions
- $\varphi_e^{(k)}$ for k-ary partial recursive functions

Step 4: Critical Details

- Always specify indices e for computable functions
- Use correct encoding for multiple quantifiers
- State which functions/predicates are computable/decidable
- Apply Structure Theorem explicitly when using existential quantification

FORMULA FOR SUCCESS

Structure/Projection Exercise = Pattern Recognition + Template Application + Standard Counterexamples + Precise Encoding

These theorems appear in nearly every computability exam. Master the four patterns above, memorize the standard counterexamples, and you can solve any variation.