Exercise 1 Universal Guide: All Terminologies and Exam Patterns

CRITICAL CORRESPONDENCES (MEMORIZE THIS)

Equivalent Terminologies

Concept	Terminology 1	Terminology 2	Function
Decidable	Decidable Predicate	Recursive Set	Characteristic χ_A
Semi-Decidable	Semi-Decidable Predicate	Recursively Enumerable (r.e.) Set	Semi-Characteristic sc_A
Computable	Computable Function	Recursive Function	-

UNIVERSAL RULE: Exercise 1 always tests these fundamental concepts regardless of terminology used.

STANDARD STRUCTURE (Always Parts a, b, c)

VARIATION A: Predicate Terminology

Part a: Definition of decidable predicate

Part b: Definition of semi-decidable predicate

Part c: Theoretical question about predicates

VARIATION B: Set Terminology

Part a: Definition of recursive set

Part b: Definition of recursively enumerable (r.e.) set

Part c: Theoretical question about sets

VARIATION C: Mixed Terminology

Part a: Any of the above

Part b: Any of the above

Part c: Questions mixing predicates, sets, and functions

UNIVERSAL DEFINITIONS (Master ALL Variations)

Decidable Predicate / Recursive Set

Predicate Version: A predicate P(x⃗) ⊆ ℕᵏ is decidable if the characteristic function χ_P : ℕᵏ → ℕ defined by

```
χ_P(x⃗) = {
  1 if P(x⃗)
  0 otherwise
}
```

is computable.

Set Version: A set A ⊆ ℕ is recursive if the characteristic function χ_A : ℕ → ℕ defined by

```
χ_A(x) = {
  1 if x ∈ A
  0 otherwise
}
```

is computable.

Semi-Decidable Predicate / Recursively Enumerable Set

Predicate Version: A predicate $P(\vec{x}) \subseteq \mathbb{N}^k$ is semi-decidable if the semi-characteristic function $sc\_P : \mathbb{N}^k \to \mathbb{N}$ defined by

```
sc_P(x⃗) = {
  1 if P(x⃗)
  ↑ otherwise
}
```

is computable.

Set Version: A set $A \subseteq \mathbb{N}$ is recursively enumerable (r.e.) if the semi-characteristic function $sc\_A : \mathbb{N} \to \mathbb{N}$ defined by

```
sc_A(x) = {
  1 if x ∈ A
  ↑ otherwise
}
```

is computable.

COMPLETE PART C PATTERN ANALYSIS

Pattern 1: Structure/Projection Theorem Applications

Applies to: Both predicates and sets
Common Questions:

- "Show if Q semi-decidable then $\exists y.Q(\vec{x},y)$ semi-decidable"
- "Show if A r.e. then $\{f(x) \mid x \in A\}$ r.e. for computable f"

Solution Framework:

1. Apply Structure Theorem (semi-decidable $\iff \exists t.H(\ldots)$)
2. Use Projection Theorem (closure under $\exists$)
3. Construct appropriate semi-characteristic function

Pattern 2: Closure Properties

Applies to: Both predicates and sets

Key Questions:

- Logical operations: ∧, ∨, ¬, ⟹
- Set operations: ∪, ∩, complement
- Quantification: ∃, ∀

Universal Closure Table:

Operation	Recursive/Decidable	R.E./Semi-Decidable
Union/Disjunction ∪/∨	✓ CLOSED	✓ CLOSED
Intersection/Conjunction ∩/∧	✓ CLOSED	✓ CLOSED
Complement/Negation c/¬	✓ CLOSED	✗ NOT CLOSED
Existential Quantification ∃	✗ NOT CLOSED	✓ CLOSED
Universal Quantification ∀	✗ NOT CLOSED	✗ NOT CLOSED

Pattern 3: Set Operations and Transformations

Exam Examples: 2024-02-02, 2023-07-21, 2022-02-04

Question Types:

- "If A recursive and B = A ∩ P, is B recursive?" (P = even numbers)
- "If A r.e. and B = $\{z^2 \mid z \in A\}$, is B r.e.?"
- "If A and Ā both r.e., is A recursive?"

Solution Template:

```
For Set Transformations:
1. Identify the transformation T: A → B
2. Show T preserves computability properties
3. Construct characteristic/semi-characteristic function for B
4. Use composition, bounded search, or encoding as needed

For Closure Under Operations:
1. Apply closure properties from table above
2. Construct explicit functions when needed
3. Use counterexamples for non-closure (K and K̄)
```

Pattern 4: Function-Set Relationships

Exam Examples: Function computability ⟺ Set properties

Question Types:

- "f computable iff Q_f(x,y) = 'f(x) = y' semi-decidable"
- "Show there exists non-computable f with specific image properties"

Solution Strategy:

1. Use semi-decidability for "search" operations
2. Apply diagonalization for non-computability
3. Use encoding for image characterization

Pattern 5: Recursive vs R.E. Characterizations

Exam Examples: 2022-02-04

Key Theorem: A ∩ Ā both r.e. ⟹ A recursive

Solution Template:

```
Given: A and Ā both r.e.
Want: A recursive

Proof:
1. sc_A and sc_Ā computable with indices e₁, e₀
2. Define: 1 - sc_Ā(x) = {0 if x ∈ A, ↑ otherwise}
3. χ_A(x) = ⟨μw.(S(e₀,x,⟨w⟩₁,⟨w⟩₂) ∨ S(e₁,x,⟨w⟩₁,⟨w⟩₂))⟩₁
4. This is computable, therefore A recursive
```

Pattern 6: Advanced Set Classifications

Exam Examples: Using Rice's theorem, Rice-Shapiro theorem

Question Types:

- "Classify A = {x | W_x = E_x} from recursiveness viewpoint"
- "Establish if A and Ā are recursive/r.e."

Solution Framework:

1. Check if set is saturated (functional property)
2. Apply Rice's theorem (not recursive)
3. Apply Rice-Shapiro theorem (r.e. classification)
4. Use reduction from K when needed

COMPREHENSIVE SOLUTION TEMPLATES

Template A: Set Intersection/Union with Recursive Sets

```
Given: A recursive, P recursive (e.g., P = even numbers)
Want: B = A ∩ P recursive

Proof:
χ_B(x) = χ_A(x) · χ_P(x) is computable
Therefore B recursive

Counterexample for converse:
A = {2x+1 | x ∈ K}, P = even numbers
B = A ∩ P = ∅ (recursive)
But A not recursive (K ≤_m A via f(x) = 2x+1)
```

Template B: Transformation Preservation

```
Given: A r.e., T(A) = {f(x) | x ∈ A} for computable f
Want: T(A) r.e.

Proof:
Let e₁, e₂ be indices for sc_A, f respectively
sc_T(A)(y) = 1(μ⟨x,t⟩.H(e₁,x,t) ∧ S(e₂,x,y,t))
This is computable, therefore T(A) r.e.
```

Template C: Non-Closure Counterexamples

```
For Complement/Negation:
- K = {x | x ∈ W_x} is r.e.
- K̄ = {x | x ∉ W_x} is not r.e.
- If K̄ were r.e., then K recursive (contradiction)

For Universal Quantification:
- R(t,x) = ¬H(x,x,t) is decidable
- ∀t.R(t,x) = "x ∉ K" is not semi-decidable
```

EXAM-SPECIFIC VARIATIONS

2024-02-02: Set Operations

Parts a,b: Recursive set, r.e. set definitions

Part c: A recursive, B = A ∩ P ⟹ B recursive? Converse?

2023-07-21: Set Transformations

Parts a,b: Recursive set, r.e. set definitions

Part c: A recursive, $B = \{z^2 \mid z \in A\}$ ⟹ B recursive? Generalize to r.e.?

2022-02-04: Fundamental Characterization

Parts a,b: Recursive set, r.e. set definitions

Part c: A and $\bar{A}$ both r.e. $\Longrightarrow$ A recursive

2024-06-20: Predicate Logic

Parts a,b: Decidable predicate, semi-decidable predicate

Part c: Logical implication closure

2025-01-28: Quantification

Parts a,b: Semi-decidable predicate definition (note: missing part a)

Part c: Universal quantification implications

STRATEGIC APPROACH FOR ANY EXERCISE 1

Step 1: Terminology Recognition (30 seconds)

1. Identify if using predicate or set terminology
2. Note any mixed terminology
3. Recall corresponding definitions

Step 2: Pattern Classification (30 seconds)

1. Structure/Projection application
2. Closure properties
3. Set operations/transformations
4. Function-set relationships
5. Recursive vs r.e. characterization
6. Advanced set classification

Step 3: Template Application (4-6 minutes)

1. Apply appropriate solution template
2. Use correct terminology consistently
3. Include all technical details

Step 4: Counterexample Construction (if needed)

1. Use K and $\bar{K}$ for non-closure
2. Use diagonalization for non-computability
3. Use reduction techniques when appropriate

UNIVERSAL SUCCESS CHECKLIST

Definitions Mastery

- [ ] All four definition variations (predicate/set × decidable/semi-decidable)
- [ ] Characteristic vs semi-characteristic functions
- [ ] Terminology correspondences

Theoretical Tools

- [ ] Structure and Projection theorems
- [ ] Closure properties table
- [ ] Rice's and Rice-Shapiro theorems
- [ ] Reduction techniques

Technical Skills

- [ ] Function composition and encoding
- [ ] Semi-characteristic function construction
- [ ] Counterexample patterns
- [ ] Index notation and step predicates

Common Mistakes to Avoid

1. **Mixing terminologies** within same answer
2. **Forgetting converse questions** - always check if bidirectional
3. **Incomplete closure analysis** - check all relevant operations
4. **Wrong characteristic function** - χ vs sc distinction
5. **Missing technical details** - indices, computability statements

FINAL FORMULA FOR UNIVERSAL SUCCESS

Exercise 1 Mastery = Terminology Fluency + Pattern Recognition + Template Library + Technical Precision

This guide covers ALL possible Exercise 1 variations across different computability courses and exam styles. Master these patterns and you can handle any Exercise 1 regardless of terminology or specific theoretical focus.