Overview

This lecture covers fundamental results in computability theory regarding recursively enumerable (r.e.) sets and program properties, focusing on the Rice-Shapiro theorem and its applications.

Key Concepts

Recursively Enumerable Sets

A set $A \subseteq N$ is recursively enumerable (r.e.) if its semi-characteristic function is computable:

sc_A(x) = {
1 if $x \in A$
↑ otherwise
}

Structure Theorem for Semi-Decidable Predicates

A predicate P(x□) is semi-decidable if and only if there exists a decidable predicate Q(t,x□) such that:

P(x□) ≡ □t.Q(t,x□)

This characterizes semi-decidable predicates as existential quantifications of decidable predicates.

Rice-Shapiro Theorem

Statement

For a set of computable functions $A \subseteq C$, if the corresponding set of indices $A = \{x \mid \varphi_x \in A\}$ is r.e., then:

$\forall f\ (f \in A \Leftrightarrow \exists \theta \subseteq f,\ \theta \text{ finite},\ \theta \in A)$

Interpretation

The theorem states that membership in a recursively enumerable set of functions can be determined by examining finite portions of the functions' behavior.

Application Strategy

The theorem provides two ways to prove a set is not r.e.:

1. Find $f \notin A$ with a finite $\theta \subseteq f$ where $\theta \in A$
2. Find $f \in A$ where all finite $\theta \subseteq f$ satisfy $\theta \notin A$

Example Applications

Case Study 1: Total Functions

- Set $T = \{x \mid \varphi_x \text{ total}\}$
- T is not r.e.
- Proof: For any total $f \in T$, all finite $\theta \subseteq f$ are partial

Case Study 2: Injective Functions

- Set $I = \{x \mid \varphi_x \text{ injective}\}$
- I is not r.e.
- Proof: $id \notin I$ but has finite $\theta \subseteq id$ with $\theta \in I$

S-m-n Theorem Applications

The s-m-n theorem allows construction of functions with specific domain/range properties:

- Given $m,n \geq 1$, there exists computable total $s_{m,n}: N^{m+1} \to N$
- For all $e \,\square\, N$, $x\square \,\square\, N^m$, $y\square \,\square\, N^n$:
  $\varphi_e^{(m+n)}(x\square, y\square) = \varphi s_{m,n}(e, x\square)^{(n)}(y\square)$

Practical Applications

- Understanding program properties
- Analyzing computational limits
- Proving undecidability results
- Characterizing tractable verification problems

Key Takeaways

1. Program properties depending on infinite behavior are typically not r.e.
2. Only "finitary" properties of programs can be semi-decidable
3. The s-m-n theorem enables construction of functions with specified properties
4. Reduction to known non-r.e. sets (like K) proves non-recursive enumerability