1. Definition by Cases

Core Concept

A function can be defined through cases using decidable predicates and known computable functions:

```
f(x̄) = {
  f₁(x̄) if Q₁(x̄)
  f₂(x̄) if Q₂(x̄)
  ...
  fₙ(x̄) if Qₙ(x̄)
}
```

where:

- Each Q_i is a decidable predicate
- Predicates are mutually exclusive
- Each f_i is computable

Characteristic Functions

For a set $A \subseteq N$, its characteristic function is:

```
χA(x) = {
  1 if x ∈ A
  0 if x ∉ A
}
```

2. Algebra of Decidability

Basic Operations

For decidable predicates Q_1, Q_2:

1. Negation (¬Q):

```
X₍¬Q₎(x) = sg(XQ(x))
```

2. Conjunction ($Q_1 \land Q_2$):

```
X₍Q₁∧Q₂₎(x) = XQ₁(x) · XQ₂(x)
```

3. Disjunction ($Q_1 \lor Q_2$):

```
X₍Q₁∨Q₂₎(x) = max{XQ₁(x), XQ₂(x)}
```

Closure Properties

If $Q_1,\ldots,Q_n$ are decidable predicates and $f:\{0,1\}^n \to \{0,1\}$, then:

- The predicate Q corresponding to $f(XQ_1,\ldots,XQ_n)$ is decidable

3. Bounded Operations

Bounded Sum

For $f:N^{k+1} \to N$ computable:

```
g(x̄,y) = Σz<y f(x̄,z)

Defined by primitive recursion:
g(x̄,0) = 0
g(x̄,y+1) = g(x̄,y) + f(x̄,y)
```

Bounded Product

Similarly:

```
h(x̄,y) = Πz<y f(x̄,z)

Defined by:
h(x̄,0) = 1
h(x̄,y+1) = h(x̄,y) · f(x̄,y)
```

Bounded Quantification

For decidable predicate Q:

1. Universal:

```
Q₁(x̄,y) = ∀z<y.Q(x̄,z)
XQ₁(x̄,y) = Πz<y XQ(x̄,z)
```

2. Existential:

```
Q₂(x̄,y) = ∃z<y.Q(x̄,z)
XQ₂(x̄,y) = sg(Σz<y XQ(x̄,z))
```

4. Bounded Minimalization

Definition

For $f:N^{k+1}\to N$ total computable:

```
h(x̄,y) = μz<y.f(x̄,z) = {
  min{z<y : f(x̄,z)=0} if such z exists
  y otherwise
}
```

Key Properties

1. Always terminates (bounded search)
2. Total when f is total
3. Computable through bounded operations:

```
h(x̄,y) = Σz<y Πw≤z sg(f(x̄,w))
```

Examples

1. Integer square root:

```
⌊√x⌋ = max{y≤x : y²≤x}
     = min{y≤x : (y+1)²>x}
```

2. Greatest common divisor:

```
gcd(x,y) = min{z≤min(x,y) : z|x ∧ z|y}
```

5. Applications to Arithmetic Functions

The following functions can be proven computable using bounded operations:

1. Number of divisors D(x)
2. Primality test Pr(x)
3. nth prime number p_n
4. Prime power decomposition $(x)_i$

These concepts form the foundation for understanding more complex computability results and provide essential tools for proving functions computable through bounded operations.