1. Primitive Recursive Functions

1.1 Basic Functions

The base functions in PR are:

1. Zero function: $Z(x) = 0$
2. Successor function: $S(x) = x + 1$
3. Projection functions: $U_i^n(x_1, \ldots, x_n) = x_i$

1.2 Construction Methods

PR functions can be built using:

1. Composition
2. Primitive Recursion

```
h(x̄,0) = f(x̄)
h(x̄,y+1) = g(x̄,y,h(x̄,y))
```

1.3 Template for Proving Function is PR

```
To prove h is PR:
1. Express h using primitive recursion:
   - Identify base case h(x̄,0)
   - Define step case h(x̄,y+1)
2. Show f and g are PR
3. Conclude h is PR by primitive recursion
```

Example:

```
Proving addition is PR:
plus(x,0) = x
plus(x,y+1) = S(plus(x,y))

where:
- f(x) = x is PR (projection)
- g(x,y,z) = S(z) is PR (composition of successor)
```

2. Ackermann's Function

2.1 Properties

```
ψ(0,y) = y + 1
ψ(x+1,0) = ψ(x,1)
ψ(x+1,y+1) = ψ(x,ψ(x+1,y))
```

2.2 Proving Properties

To prove ψ is:

1. Total: Use well-founded induction on ($\mathbb{N}^2$,≤_lex)
2. Computable: Show it's in R
3. Not PR: Show it grows faster than any PR function

3. Function Analysis Template

3.1 Determining if Function is PR

1. Try to define using only:
   - Basic functions
   - Composition
   - Primitive recursion
2. If impossible, prove it's not PR:
   - Show it grows faster than any PR function
   - Reduce from Ackermann's function

3.2 Determining if Function is Recursive

1. Try to prove computability:
   - Write explicit algorithm
   - Use closure properties
   - Apply recursion theorems
2. If not computable:
   - Use diagonalization
   - Reduce from known non-computable function