1. Universal Function

1.1 Definition

The universal function is defined as:

```
Ψ_U: ℕ² → ℕ
Ψ_U(x,y) = ϕ_x(y)
```

General k-ary version:

```
Ψ_U^(k): ℕ^(k+1) → ℕ
Ψ_U^(k)(e,x̄) = ϕ_e^(k)(x̄)
```

1.2 Key Properties

1. Computability predicates:

```
H(e,x,t): "P_e(x)↓ in t or fewer steps"
S(e,x,y,t): "P_e(x)↓y in t or fewer steps"
```

2. Kleene Normal Form:

```
ϕ_e^(k)(x) = (μz.|χ_S^k(e,x̄,(z)_1,(z)_2) - 1|)_1
```

2. Second Recursion Theorem (SRT)

2.1 Statement

For any total computable function f, there exists e_0 such that:

```
ϕ_e_0 = ϕ_f(e_0)
```

2.2 Application Template

```
1. Define suitable computable function f
2. Apply SRT to get e_0
3. Use the fact that ϕ_e_0 = ϕ_f(e_0) to reach conclusion
```

2.3 Common Applications

1. Proving K is not recursive:

```
Define f(x) = {
    e_0 if x ∈ K
    e_1 if x ∉ K
}
```

2. Proving a set is not saturated:

```
Use SRT to construct e where:
ϕ_e(y) = {
    0 if y = e
    ↑ otherwise
}
```

3. Exercise Solving Strategies

3.1 For Universal Function Problems

1. Configuration tracking:

```
Consider:
- Initial configuration
- Step-by-step execution
- Final configuration
```

2. Using predicates H and S:

```
To check if computation halts:
use H(e,x,t)

To check specific output:
use S(e,x,y,t)
```

3.2 For SRT Problems

1. Function construction:

```
- Define f based on desired property
- Ensure f is total and computable
```

```
- Apply SRT to get fixed point
```

2. Contradiction proofs:

```
- Assume property holds
- Define f to break property
- Use fixed point to reach contradiction
```

4. Common Problem Types

4.1 Fixed Point Problems

```
Goal: Find e where ϕ_e has special property P

1. Define f to enforce P
2. Apply SRT
3. Show fixed point has P
```

4.2 Impossibility Proofs

```
Goal: Show no program can have property P

1. Assume program exists
2. Define f to create contradiction
3. Use fixed point to show impossibility
```

4.3 Non-Computability Proofs

```
Goal: Show function g is not computable

1. Assume g computable
2. Define f using g
3. Use fixed point to reach contradiction
```

5. Exercise Examples

5.1 Self-Referential Programs

```
Goal: Find program that prints its own index

1. Define f(x) = "program that outputs x"
```

```
2. Apply SRT to get e_0
3. Show e_0 outputs e_0
```

5.2 Impossibility Results

```
Goal: Show no program can determine if another
      program outputs its own index

1. Assume such program exists
2. Define f to contradict it
3. Use fixed point to reach contradiction
```