

Fixed Table

complete and Disjoint

Fixed Table

Complete and Disjoint

Temperature (°C)	Humidity (%)	Action
< 0	*	Heat + Humidify
[0-20)	< 40	Heat + Humidify
[0-20)	[40, 60]	Heat
[0-20)	> 60	Heat
[20-25)	< 40	Humidify
[20-25)	[40, 60]	OFF
[20-25)	> 60	Dehumidify
≥ 25	≤ 60	Cool
≥ 25	> 60	Cool + Dehumidify

- Every temperature/humidity combination is covered (complete)
- No overlapping ranges (disjoint)
- Clear boundary definitions using [inclusive, exclusive)
- Realistic HVAC logic: cold air needs humidity, hot air gets dehumidified naturally by cooling
- Verifiable by a domain expert in a format they can understand

Mapping to Swift

Basic Structure

```
enum UserRole {
    case admin, user, guest
}

enum ResourceType {
    case publicResource, sharedResource, privateResource
}

enum Action {
    case read, write, delete
}

enum AccessResult {
    case allow, deny

    var symbol: String { ... }
}

// Complete and exhaustive implementation
func checkAccess(
    role: UserRole,
    resource: ResourceType,
    action: Action,
    isOwner: Bool
) -> AccessResult {
    switch (role, resource, action, isOwner) { ... }
}
```