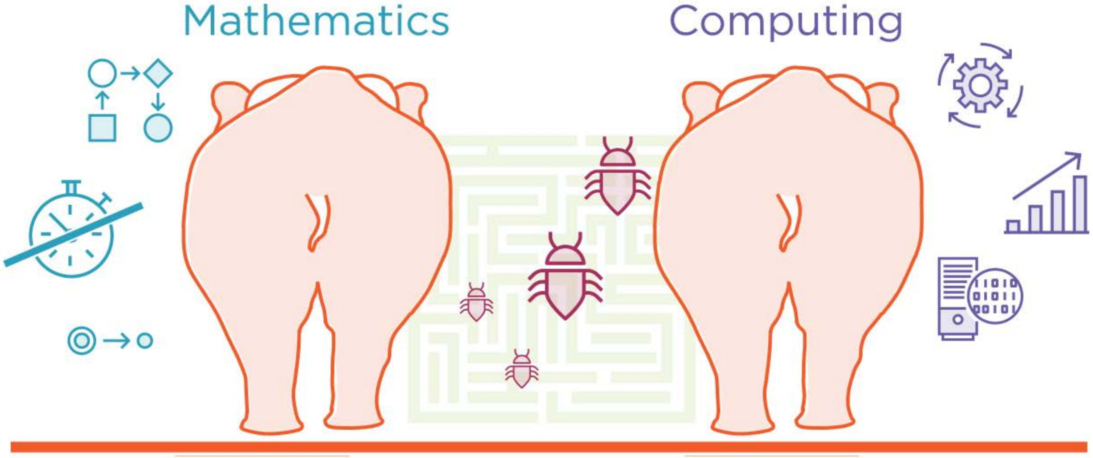
Controlling Execution Flow with Pattern Matching

Zoran Horvat
CEO AT CODING HELMET

@zoranh75 http://csharpmentor.com

The Secret Life of Functions



The Secret Life of Functions

Pattern matching

The Secret Life of Functions

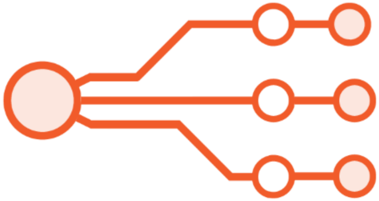
Pattern matching

The Secret Life of Functions

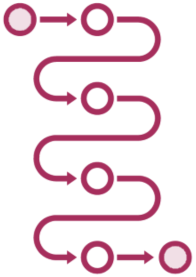
Pattern matching

The Secret Life of Functions

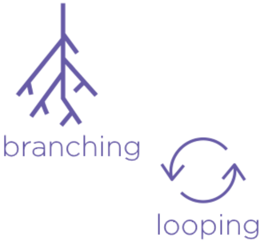
Pattern matching



Function composition (pipelining)

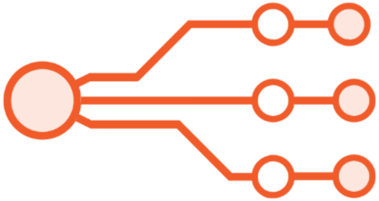


Imperative control flow

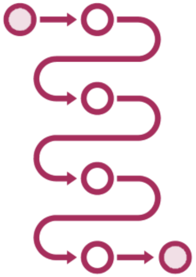


The Secret Life of Functions

Pattern matching



Function composition (pipelining)



Imperative control flow

branching
looping

Defining a Function

```
static void Subtract(
  this Amount from,
  Amount amount)
```

Fail

from.Currency ≠ amount.Currency

Succeed partially

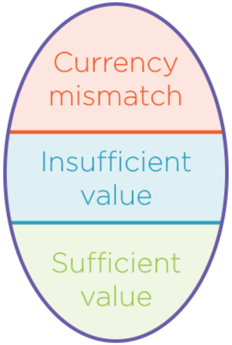
from.Currency = amount.Currency

from.Value < amount.Value

Succeed

from.Currency = amount.Currency

from.Value ≥ amount.Value



Mutually exclusive
Intersections
of cases are empty

Complete
Union of all cases
is the entire domain

Defining a Function

```
static void Subtract(
  this Amount from,
  Amount amount)
```

Fail

from.Currency ≠ amount.Currency

Succeed partially

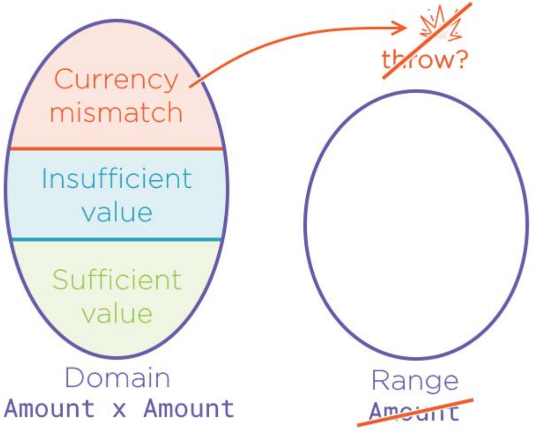
from.Currency = amount.Currency

from.Value < amount.Value

Succeed

from.Currency = amount.Currency

from.Value ≥ amount.Value



Defining a Function

```
static void Subtract(
  this Amount from,
  Amount amount)
```

Fail

from.Currency ≠ amount.Currency

Succeed partially

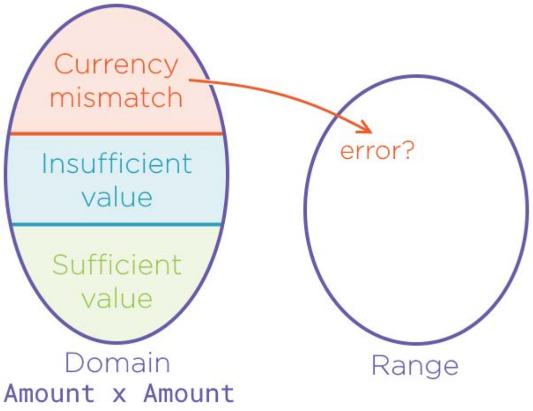
from.Currency = amount.Currency

from.Value < amount.Value

Succeed

from.Currency = amount.Currency

from.Value ≥ amount.Value



Defining a Function

```
static void Subtract(
  this Amount from,
  Amount amount)
```

Fail

from.Currency ≠ amount.Currency

Succeed partially

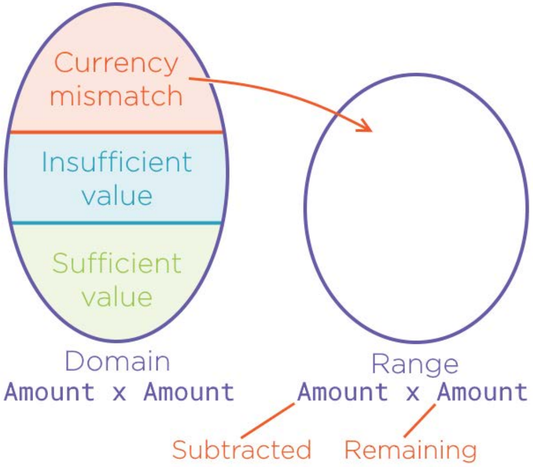
from.Currency = amount.Currency

from.Value < amount.Value

Succeed

from.Currency = amount.Currency

from.Value ≥ amount.Value



Defining a Function

```
static void Subtract(
  this Amount from,
  Amount amount)
```

Fail

from.Currency ≠ amount.Currency

Succeed partially

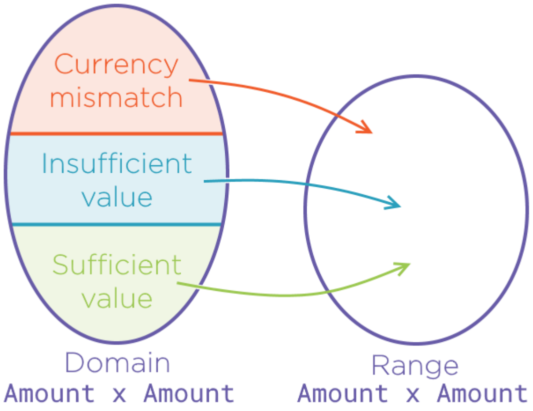
from.Currency = amount.Currency

from.Value < amount.Value

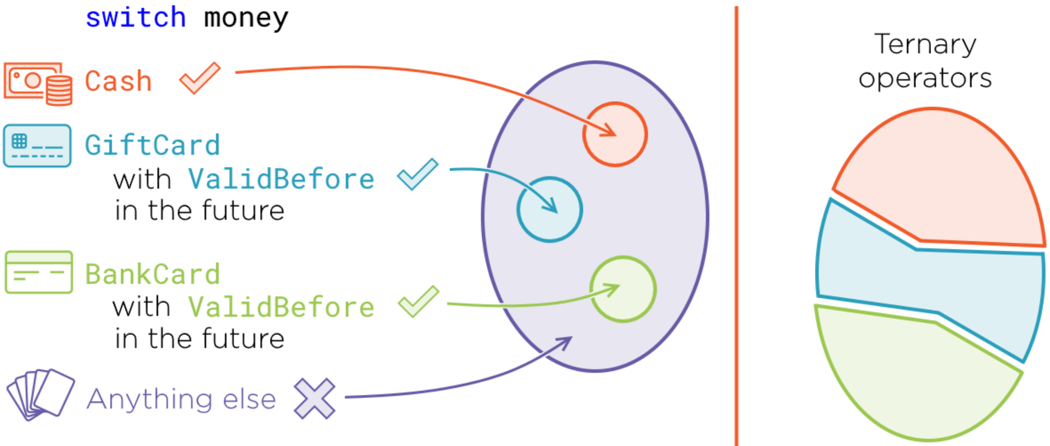
Succeed

from.Currency = amount.Currency

from.Value ≥ amount.Value



`switch` vs. Ternary Operator



Summary

Equality tests and pattern matching

- Patterns often call `Equals()`
- Patterns often test runtime type

Test type and set variable (C# 7)

- Used to mimic pattern matching

Chain of ternary operators

- Progressively constrains the domain

Switch instruction with type matching

- Declares patterns from specific to general
- Compiler handles evaluation

Summary

Pattern matching as a control flow

- The only form of branching

Applicable to some functions

- When a function looks like a "union" of several functions
- Pattern matching used to select one of the functions/mappings

Next module:

Working with Sequences in Functional Way