

- Given values v_1 and v_2

if $v_1 : t_1$ and $v_2 : t_2$

then $(v_1, v_2) : t_1 * t_2$

Pair

Product type

$(v_1, v_2, v_3) : t_1 * t_2 * t_3$

Triple

$(v_1, v_2, \dots, v_k) : t_1 * t_2 * \dots * t_k$

Tuples

- Philip Wadler: Theorems for Free

Types are often the only thing you need to know about the function.

- Destruct larger tuples using pattern matching

let $t = (2, 3, 4)$

match t with

$(a, b, c) \rightarrow a + b - c ;$

- Variants in OCaml

Enum in Java/C++

Inheritance in Java/C++ : open world.

abstract class DayOfWeek

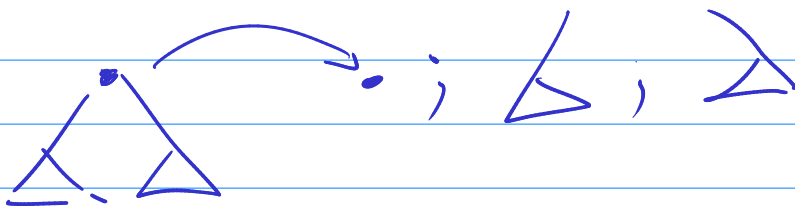
} closed world. Much greater control/error checking

- Let's talk about binary trees.

Algebraic
Data
Types

Variants can have infinitely many members.
Enums cannot.

- Pre order traversals



- Shapes of values: fixed.

Easy to define new functions.

Variants

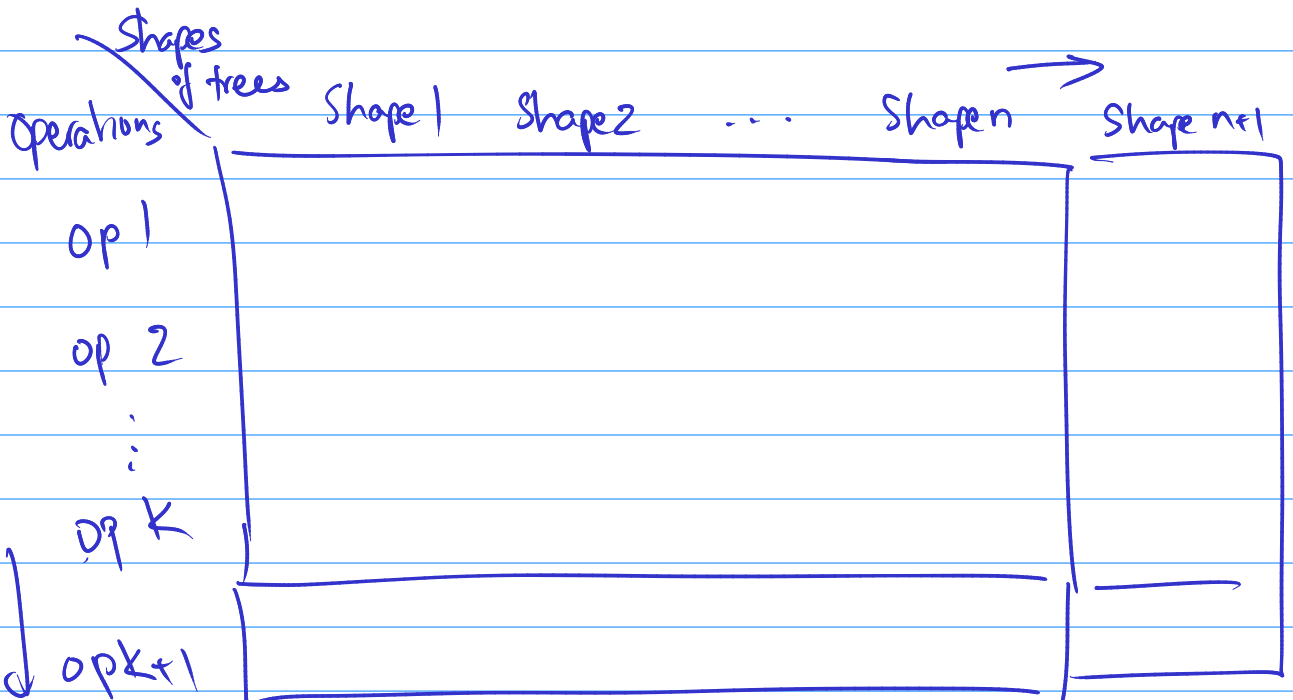
- Shapes of values: open

class NewClass extends OldClass

Inheritance /
ad hoc polymorphism

Binary Tree → To add a function
to the Binary Tree
class, we need
to add an implement
to every inheriting
subclass.

preorder, root, find BST, post order, ...



let op k+1 t =

Case 1 →
Case 2 →
...
Case n → n

```
class 1 extends Base {  
  op k+1  
}  
class 2 extends Base {  
  op k+1  
  ...  
}
```

- Pattern matching similar to the Visitor pattern.

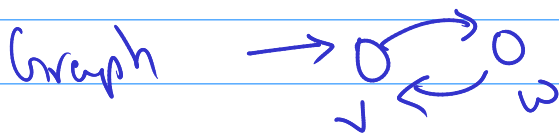
```
abstract class TreeVisitor
```

```
    int Leaf() { }  
    int Node1() { }  
    int Node2() { }  
    int Node3() { }
```

```
class Preorder extends  
    TreeVisitor {
```

```
    int Leaf() { }  
    int Node1() { }  
    int Node2() { }  
}
```

- Limitation: Difficult to define cyclic structures



(V, E)

type edge = (vertex, vertex);

type graph = (vertex set, edge set);

type graph = (vertex set, (vertex, set vertex) map)