

RECURSIVE DATATYPES

David Kauchak
CS52 – Spring 2017

Recursive datatype

```
datatype 'a binTree =
  Empty
  | Node of 'a binTree * 'a * 'a binTree;
```

- Defines a type variable for use in the datatype constructors
- Still just defines a new type called "binTree"

Recursive datatype

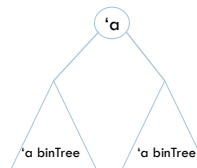
```
datatype 'a binTree =
  Empty
  | Node of 'a binTree * 'a * 'a binTree;
```

What is this?

Recursive datatype

```
datatype 'a binTree =
  Empty
  | Node of 'a binTree * 'a * 'a binTree;
```

Binary Tree!



A binary tree is a recursive data structure where each node in the tree consists of a value and then two other binary trees.

Recursive datatype

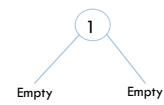
```
datatype 'a binTree =
  Empty
  | Node of 'a binTree * 'a * 'a binTree;
```

Node(Empty, 1, Empty); **What does this look like?**

Recursive datatype

```
datatype 'a binTree =
  Empty
  | Node of 'a binTree * 'a * 'a binTree;
```

Node(Empty, 1, Empty);



Recursive datatype

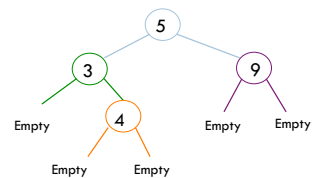
```
datatype 'a binTree =
  Empty
  | Node of 'a binTree * 'a * 'a binTree;
```

Node(Node(Empty, 3, Node(Empty, 4, Empty)), 5, Node(Empty, 9, Empty));

What does this look like?

Recursive datatype

```
datatype 'a binTree =
  Empty
  | Node of 'a binTree * 'a * 'a binTree;
Node(Node(Empty, 3, Node(Empty, 4, Empty)), 5, Node(Empty, 9, Empty));
```



Recursive datatype

```
datatype 'a binTree =
  Empty
  | Node of 'a binTree * 'a * 'a binTree;
```

```
Node(Node(Empty, "apple", Node(Empty, "banana", Empty)),
      "carrot",
      Node(Empty, "rhubarb", Empty));
```

What does this look like?

Recursive datatype

```
datatype 'a binTree =
  Empty
  | Node of 'a binTree * 'a * 'a binTree;
```

```
Node(Node(Empty, "apple", Node(Empty, "banana", Empty)),
      "carrot",
      Node(Empty, "rhubarb", Empty));
```

