

A Note on Parameterized Type Definitions

David Walker

COS 326

```
type ('key, 'val) tree =  
  Leaf  
  | Node of 'key * 'val * ('key, 'val) tree * ('key, 'val) tree
```

```
type 'a stree = (string, 'a) tree
```

```
type sitree = int stree
```

General form:

definition:

```
type 'x f = body
```

use:

```
arg f
```

A Better Notation:

definition:

```
type f x = body
```

use:

```
f arg
```

Take-home Message

- Think of parameterized types like functions:
 - a function that take a type as an argument
 - produces a type as a result
- Theoretical basis:
 - System F-omega
 - a typed lambda calculus with general type-level functions as well as value-level functions