

Parametric Polymorphism Examples

type (<list type params>) <identifier> = <type expr>

Example 1- pair

-type 'a pair = 'a * 'a;
*type 'a pair = 'a * 'a*

-(1,2): int pair;
val it = (1,2): int pair

Example 2- word count

- type ('d,'r) mapping = ('d * 'r) list;
*type ('a, 'b) mapping = ('a * 'b) list*

-val wc = ("in",5), ("a",1)]: (string, int) mapping;
val wc = [("in",5), ("a",1)] : (string, int) mapping

Equality Types and 'a versus `a

= and <> are **equality operators**

ML defines a class of types called equality types, which are types that allow equality to be tested. Most basic types are equality types – integer, boolean, character and string, **not** functions

One can form more equality types by forming tuples or lists of equality types.

If a function uses equality comparison, it restricts the type to an equality type, as illustrated in the examples below.

The following examples are from [Ullman, 1998, pg. 153]

```
1 fun rev1(L) =  
2     if L = nil then nil  
3     else rev1(tl(L) @ [hd(L)];  
   val rev1 = fun : 'a list -> 'a list
```

Reversal using an equality comparison

```
4 fun rev2(nil) = nil  
5 |   rev2(x::xs) = rev2(xs) @ [x]  
   val rev2 = fun : 'a list -> 'a list
```

Reversal *without* an equality comparison