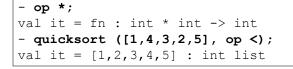


The **op** keyword

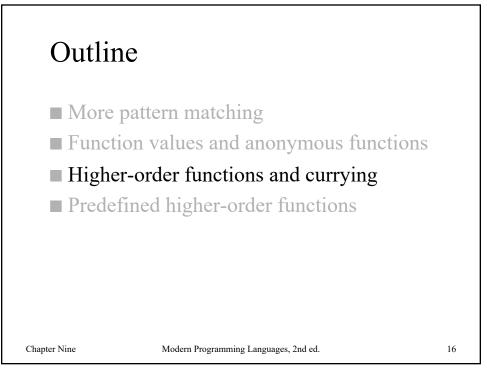


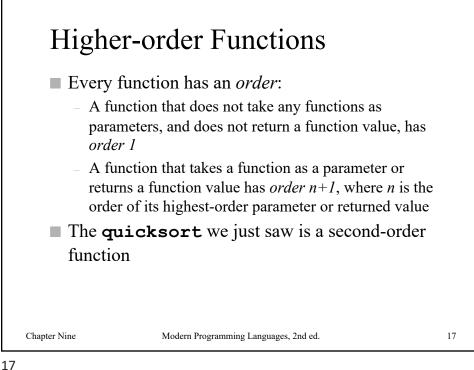
- Binary operators are special functions
- Sometimes you want to treat them like plain functions: to pass <, for example, as an argument of type int * int -> bool
- The keyword **op** before an operator gives you the underlying function

```
Chapter Nine
```

Modern Programming Languages, 2nd ed.

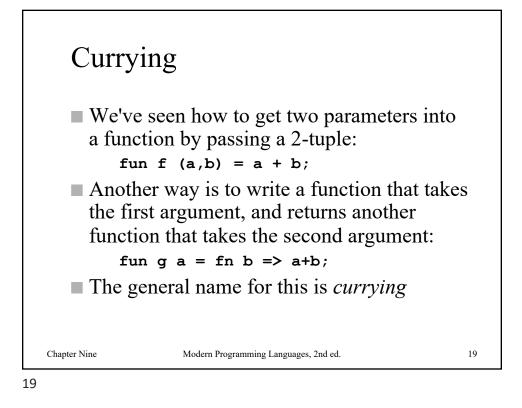
15





1/

<section-header><text><text><equation-block><equation-block><equation-block><equation-block><equation-block><table-row><equation-block><table-row><equation-block>

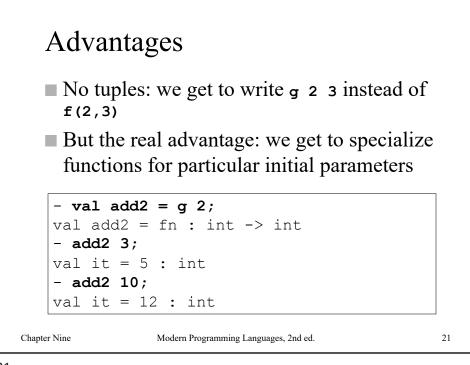


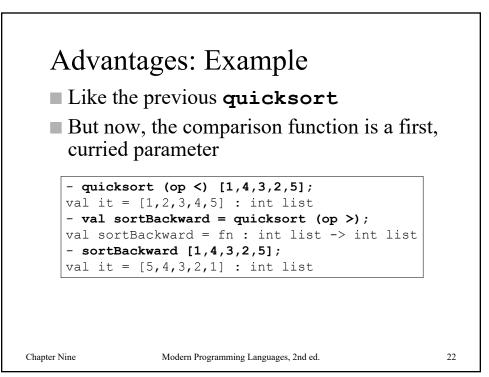
Curried Addition

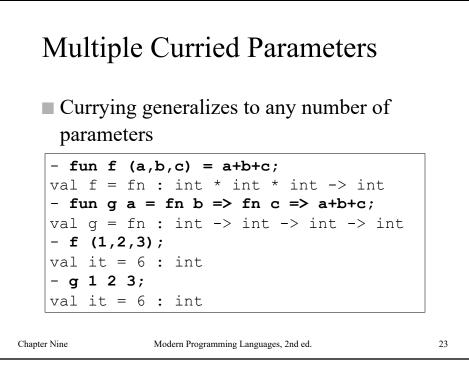
f fun f (a,b) = a+b;
val f = fn : int * int -> int
fun g a = fn b => a+b;
val g = fn : int -> int -> int
f(2,3);
val it = 5 : int
d it = 5 : int

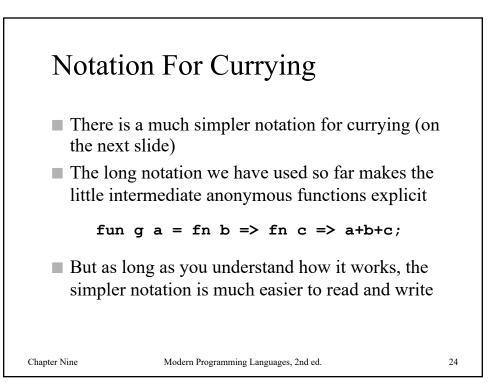
Benember that function application is left
associative

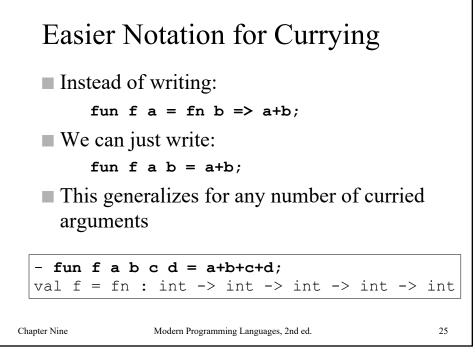
Bo g 2 3 means ((g 2) 3)

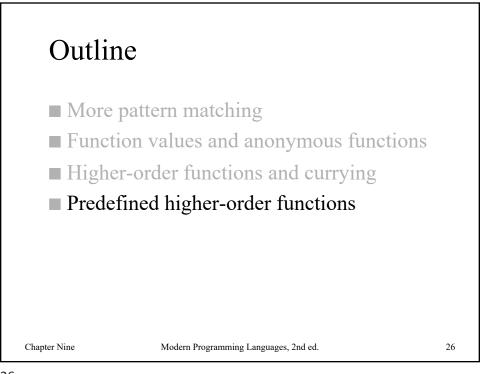


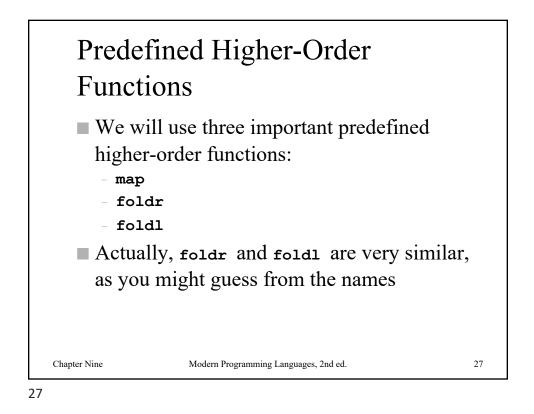








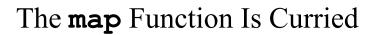


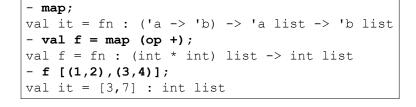


The map Function
Used to apply a function to every element
of a list, and collect a list of results - map ~ [1,2,3,4];
val it = [~1,~2,~3,~4] : int list
- map (fn x => x+1) [1,2,3,4];
val it = [2,3,4,5] : int list
- map (fn x => x mod 2 = 0) [1,2,3,4];
val it = [false,true,false,true] : bool list
- map (op +) [(1,2),(3,4),(5,6)];
val it = [3,7,11] : int list

Modern Programming Languages, 2nd ed.

Chapter Nine





Chapter Nine

Modern Programming Languages, 2nd ed.

