

- *Syntax*

$$e ::= x$$
$$| \lambda x \rightarrow e$$
$$| e_1 e_2$$

- Programs are *expressions* or  $\lambda$ -terms
- *Variable*:  $x, y, z$
- *Abstraction*: (aka nameless function definition)  
 $\lambda x \rightarrow e$  means “for any  $x$ , compute  $e$ ”;  $x$  is the *formal parameter*,  $e$  is the *body*
- *Application*: (aka function call)  $e_1 e_2$  means “apply  $e_1$  to  $e_2$ ”;  $e_1$  is the *function* and  $e_2$  is the **argument**
- *Syntactic Sugar*: convenient notation used as a short-hand for valid syntax

— *instead of:*  $\lambda x \rightarrow (\lambda y \rightarrow (\lambda z \rightarrow e))$  *we write:*  $\lambda x \rightarrow \lambda y \rightarrow \lambda z \rightarrow e$   
 $\lambda x \rightarrow \lambda y \rightarrow \lambda z \rightarrow e$   $\lambda x y z \rightarrow e$   
 $((e_1 e_2) e_3) e_4$   $e_1 e_2 e_3 e_4$

- *Scope of a variable* The part of a program where a *variable is visible*
- In the expression  $\lambda x \rightarrow e$ 
  - $x$  is the newly-introduced variable
  - $e$  is the *scope* of  $x$
  - Any occurrence of  $x$  in  $\lambda x \rightarrow e$  is *bound* (by the *binder*  $\lambda x$ )
- *Rewrite rules*:
  - $\alpha$ -step: renaming formals
  - $\beta$ -step: aka function call

Var	Desc
$B$	the number of data pages
$R$	number of records per page
$D$	average time to read or write a disk page
$F$	average fanout for a non-leaf page

		Scan	Equality		Range		Insert		Delete
$()$									
Heap		$BD$	$0.5BD$		$BD$		$2D$		Search + $D$
Sorted		$BD$	$D \log_2 B$		$D(\log_2 B + \# \text{ matching pages})$		Search + $BD$		Search + $BD$
Clustered		$1.5BD$	$D \log_F 1.5B$		$D(\log_F 1.5B + \# \text{ matching pages})$		Search + $D$		Search + $D$
Unclust. Tree		$BD(R + 0.15)$	$D(1 + \log_F 0.15B)$		$D(\log_F 0.15B + \# \text{ matching pages})$		Search + $2D$		Search + $2D$
Unclust. Hash		$BD(R + 0.125)$	$2D$		$BD$		Search + $2D$		Search + $2D$