**Haskell, List Comprehension**

Syntax of “list comprehension”

- [ elementExpression | comma-separated-clauses ]

- clause ::= var <- list

boolean condition

- elementExpression can refer to all variables defined in clauses.

- every clause can refer to variables defined in earlier clauses.

pythsLM = pyths (Monad)

[()] – single element list

[] – an empty list

To be able to use monad – import Control.Monad

[1, 4, 6] >>= (\x -> take x (repeat x)))

->[1 4 4 4 4 6 6 6 6 6 6]

-bind to a function which takes x & take list of possibilities

Type interference, example for haskell - <http://en.wikipedia.org/wiki/Type_inference>

[ [……]

[……]

[……]

[……] ]

Foldr – fold right

Foldl – fold left