CMSC105 : Fundamentals of Computer Programming

Vikas Sindhwani
University of Chicago
Department of Computer Science
vikass@cs.uchicago.edu

Nov 03 2003

Abstract
Locals and Scoping
All about LOCAL

LOCAL is used to organize function collections.

Syntax:

<exp> = (local (<def-1> ...<def-N>) <exp>)

The local definitions are used for the evaluation of the body expression. They have only local semantics i.e are not visible outside the local expression.

How to evaluate a local expression?
Step 1: Redefine local definitions with names that dont clash with top-level
definitions. Modify body expression appropriately. Step 2: Treat redefined local definitions as top-level definitions and evaluate the modified expression.

Group Auxiliary functions and the Main function using local; Put the local expression encapsulating this collection of functions in a new function definition; The body of local should apply the main function to the arguments of the new function.
Insertion Sort

(define (sort alon)
  (local (
    (define (sort alon)
      (cond
        ((empty? alon) empty)
        ((cons? alon) (insert (first alon) (sort (rest alon)))))))
    (define (insert an alon)
      (cond
        ((empty? alon) (list an))
        (else (cond
          ((> an (first alon)) (cons an alon))
          (else (cons (first alon) (insert an (rest alon))))))))
    (sort alon))))

(define (merge list1 list2)
  (sort (append list1 list2)))
Summary

Each definition introduces two types of names: names of functions and names of parameters.

When these names appear in the definition, they are called binding occurrences.

When these appear elsewhere they are called bound occurrences.

Each occurrence is embedded within a scope.
Scope of a function name: local or global

Scope of a parameter: Function body

Bound occurrences receive their value or meaning from binding occurrences in that scope.
Traversing Graphs

Traversing Family Tree nodes : Exercise 18.1.13

Representing and Traversing directed graphs : Exercises in Back Tracking