CMSC105 : Fundamentals of Computer Programming

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

Oct 27 2003

Abstract
Mutually Referential Data Definitions
Trees

Represent a collection of objects of arbitrary number where one needs to encode dependencies.

Such dependencies may involve a fixed number of objects (Structures in Structures e.g representing people in an ancestor family trees; each person is related two persons (a father and a mother)) or an arbitrary number of objects (Lists in Structures e.g Descendant Family trees; each person is related to a list of persons (children))

Lists in Structures require two definitions: one for structures and one for lists; these refer to each other.
Mutual Referentiality: Lists in Structures

Descendant Family Trees:
(define-struct parent (children name date eyes))

parent (dft)
: (make-parent alloc na d e)

list-of-children (loc): EMPTY OR (cons aparent loc)

Design Recipe: For interrelated classes of data, construct templates for each definition and develop them in parallel. Developing programs for mutual reference
(define (fun-for-parent .. aparent)
  (cond
    ... (parent-name aparent))
    ... (parent-date aparent))
    ... (parent-eyes aparent))
  ..(fun-for-parent (parent-children aparent))

(define (fun-for-loc .. aloc)
  (cond
    ((empty? aloc) ...)
    (else
      ... (fun-for-parent→descendants ...(first aloc)) ...
      ((fun-for-loc .. (rest aloc))))))

Count Descendants, Web-Pages