CMSC105 : Fundamentals of Computer Programming

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

8 Oct 2003

Abstract
Moving Circles, Mixed Data, Scheme Syntax and Semantics
Moving Circles

- Simple Scheme graphics: Exercises 6.6.1-6.6.6
Mixed Data

- Predicates in Scheme: `number? boolean? symbol? struct?`

- Exercise 7.1.3 Develop representations for circles and squares. Develop the function `area` which consumes either a circle or a square, and computes the area.

- Design Recipe: Provide a mixed data definition; Write a template that uses a `cond` expression together with type-predicates to identify data types; Use auxiliary functions in the body of the program for each answer expression.
Scheme Vocabulary Review

- \(<\text{var}>\) = x, area – of – disk, perimeter, ...

- \(<\text{con}>\) = true, false, ′a,′ doll, ..., 1, –1, 3/5, 1.22, ...

- \(<\text{prm}>\) = +, –, ...

Keywords: define, cond, else, and, not, or..
Scheme Grammar Review

- **< def >** = (define(< var >< var > ... < var >) < exp >)  
  (define < var >< exp >)  
  (define - struct < var0 > (< var1 >< var2 > ... < varN >))

- **< exp >** = < var >  
  < con >  
  (< prm >< exp > ... < exp >)  
  (< var >< exp > ... < exp >)  
  (cond(< exp >< exp >)...(< exp >< exp >))  
  (cond(< exp >< exp >)...(else < exp >))  
  (and < exp >< exp >)  
  (or < exp >< exp >)  
  (not < exp >)

- The grammar defines the class of syntactically legal sentences. Ex 8.2.?
Scheme Semantics Review

- Scheme follows Evaluation Rules to compute the value of a scheme expression. `<var>` and `<con>` evaluate to their obvious values; `(<prm><exp> ... <exp>)` and `(<var><exp> ... <exp>)` are evaluated by computing the values of the left subexpressions; pasting these values (replacing the argument variables in the definition by these values) in the body of the definition of the operation, and evaluating the body. Special forms have special evaluation rules.
Exercise 8.3.2
Structures: Exercises 8.7.?