PROCEDURAL REPRESENTATION

CMPSC 460 – Principles of Programming Languages

Representation Strategies for Data Types

- The Symbol Table data type
  - Create an empty symbol table
  - Retrieve a value from the symbol table
  - Add a new variable with a value to the symbol table

Representation Strategies for Data Types

- Symbol Table: {(a, 2), (b, 4), (c,8)}

- Data structure representation:
  1. `(a 2 (b 4 (c 8 ())))`
  2. `(((a b) (2 4)) ((c) (8)))`

Data Structure Representation

```
(define empty-ST
   (lambda () null))

(define add-ST
   (lambda (var val st)
     (list  var val st)))

(define search-ST
   (lambda (search-var st)
            (cond
         ((null? st)
            (error search-var "not found"))
         ((pair? st)
                   (let ((saved-var (car st))
                                    (saved-val (cadr st))
                                    (saved-st (caddr st)))
                                (if (eq? search-var saved-var)
                                    saved-val
                                    (search-ST search-var saved-st))))
                (else (error "Invalid environment")))))
```

Data Structure Representation

- How do you define the following symbol table:

```
{(d, 6), (y, 8),(x, 7), (y,14)}
```

Data Structure Representation

```
(define st1
    (add-ST 'd 6
      (add-ST 'y 8
        (add-ST 'x 7
          (add-ST 'y 14
            (empty-ST))))))
```

Data Structure Representation

- What are the advantages of this implementation?
- What are the disadvantages of this implementation?

Procedural Representation

- Data and behavior
- Two type of behavior:
  - Observers
  - Constructors

Procedural Representation

```
(define empty-ST
    (lambda ()
      (lambda (search-var)
        (error search-var "not found"))))

(define add-ST
    (lambda (var val saved-st)
      (lambda (search-var)
        (if (eq? search-var var)
          val
          (search-ST search-var saved-st)))))

 (define search-ST
    (lambda (search-var st)
      (st search-var)))
```

Procedural Representation

- Exercise:

A list is either a pair or an empty list. Write a procedural representation of lists. You have to implement the following functions: *null*, *car*, *cdr*, *cons* and *null?*.

References

- Harold Abelson and Gerald Jay Sussman. *Structure and Interpretation of Computer Programs*. 2nd edition, 1996, MIT press.