MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.001 Structure and Interpretation of Computer Programs
Spring, 2007

Recitation 6, Wed. February 28

Data Abstraction Notes   Dr. Kimberle Koile

Main idea: hide implementation details

Example: primes

 representing an integer as the product of its prime factors

 40 = 2 * 2 * 2 * 5

What are possible representations?

e.g.,
 (2 2 2 5)

 (2 5 2 2)   order doesn't matter

 ((2 3) (5 1))

 (40 (2 3) (5 1))

 (40 2 2 2 5)

Choose a representation and write get-number, which takes a factorization and returns the number that was factored. Here's an example (mnre elaborate than we did in class):

multiply-factors: list(numbers) → number
make-factorization: list(numbers) → factorization
get-factors: factorization → list(numbers)
get-number: factorization → number

(define (multiply-factors factors)
 ;; assume factors is a list
 (if (null? factors)
      1   ;; the number is prime
     (* (car factors) (multiply-factors (cdr factors))))))

or (define (multiply-factors factors)
    (define (helper rest-of-factors product)
       (if (null? rest-of-factors)
           product
           (helper (cdr rest-of-factors) (* (car factors) product)))
    (helper factors 1))

or (define (multiply-factors factors)
      (apply * factors))

```
(define (make-factorization factors)
  ;; assume factors is a list
  ;; represent a factorization as the number and the list of factors
   (cons  (multiply-factors factors)   factors)))

 (define (get-factors f)
  ;; returns the list of factors in a factorization
    (cdr f)

(define (get-number f)
   ;; returns the number represented by the factorization
  (car f)))
```

Alternate representation:

```
(define (make-factorization factors)
  ;; assume factors is a list
  ;; represent a factorization as  the list of factors
   factors)

(define (get-factors f)
  ;; returns the list of factors in a factorization
   f)

(define (get-number f)
   ;; returns the number represented by the factorization
  (multiply-factors f))
```

What's the type of get-number?
    (Remember the arrow!)

factorization → number