Main idea: hide implementation details

Example: primes

representing an integer as the product of its prime factors

\[ 40 = 2 \times 2 \times 2 \times 5 \]

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 (more elaborate than we did in class):

\[
\text{multiply-factors}: \text{list(numbers)} \rightarrow \text{number} \\
\text{make-factorization}: \text{list(numbers)} \rightarrow \text{factorization} \\
\text{get-factors}: \text{factorization} \rightarrow \text{list(numbers)} \\
\text{get-number}: \text{factorization} \rightarrow \text{number}
\]

\[
\begin{align*}
\text{(define (multiply-factors factors)} & \\
& ;; \text{assume factors is a list} \\
& (\text{if (null? factors)} & \\
& \text{1} & ;; \text{the number is prime} & \\
& (* (car factors) (multiply-factors (cdr factors))))))
\end{align*}
\]

or

\[
\begin{align*}
\text{(define (multiply-factors factors)} & \\
& (\text{define (helper rest-of-factors product)} & \\
& (\text{if (null? rest-of-factors)} & \\
& \text{product} & \\
& \text{helper (cdr rest-of-factors) (* (car factors) product)} & \\
& \text{(helper factors 1))}) & \\
\text{or (define (multiply-factors factors)} & \\
& (\text{apply * factors})
\end{align*}
\]
(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