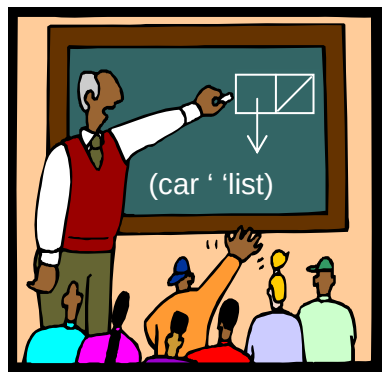
6.001 recitation 3/16/07

- tags
- stacks and queues



Dr. Kimberle Koile

tags

---

what are tags?

How are they useful?

what is an example of a tagged data structure?

stacks

- **constructor:**
  (make-stack)

- **selectors:**
  (top stack)

  (stack-elements stack)

- **operations:**

  (stack? stack)

  (empty-stack? stack)

  (insert-stack-elt stack elt)

  (delete-stack-elt stack)

stacks

(define (tagged-list? tag l) (and (pair? l) (eq? tag (car l))))

(define *stack-tag* 'stack)

- **constructor:**
  (define (make-stack)

- **selectors:**
  (define (top stack)

- **operations:**

  (define (stack? stack)

  (define (empty-stack? stack)

stack problems

---

1. Fill in the code for insert-stack-elt (aka push) for a stack.

```
(define (insert-stack-elt  element stack)
  (if (stack? stack)
      (cons [                    ]
            (cons [              ] [                    ]
            )
      (error "Insert on a non-stack")))
```

stack problems

2. Write delete-stack-elt (aka pop) for a stack. This version of pop should return a new stack that contains all elements except the top. (Don't forget the two error checks.)

```
(define delete-stack-elt  (stack)




)
```

queues

(define (tagged-list? tag l) (and (pair? l) (eq? tag (car l))))

(define *queue-tag* 'queue)

- **constructor:**
  (make-queue)

- **selectors:**
  (front-queue queue)

  (queue-elements queue)

- **operations:**

  (queue? queue)

  (empty-queue? queue)

  (insert-queue-elt queu elt)

  (delete-queue-elt queue)

queue problems

---

3. Write insert-queue-elt for a queue. (Don't forget an error check.)

```
(define insert-queue-elt  (queue)




)
```

queue problems

4. Write delete-queue-elt for a queue. (Don't forget an error check.)

```
(define delete-queue-elt  (queue)

)
```

stacks and queues

- **constructor:**
  (make-it)

- **selectors:**
  (first-elt s-or-q)

  (elements s-or-q)

- **operations:**

  (is-type? s-or-q)

  (empty? s-or-q)

  (insert-element s-or-q elt)

  (delete-element s-or-q)

5. Write a delete-elt procedure that works on either stacks or queues.

```
(define delete-elt s-or-q)

)
```

What if we could change the data structures rather than copying them?